GOVP1199711665

제 3 차년도
최종 보고서

국어정보처리기술 개발

통합 국어정보베이스

Integrated Korean Information Base

연구 기관

한 국 과 학 기 술 원

과 학 기 술 처

제 출 문

과 학 기 술 처 장 관 귀하

본 보고서를 '국어정보처리기술 개발에 관한 연구' 과제의 통합 국어정보 베이스에 관한 연구의 3 차년도 최종보고서로 제출합니다.

1997 년 8 월 20 일

주관연구기관명: 한 국 과 학 기 술 원

총괄연구책임자: 최 기선

연 구 원: 강 정구, 남 기춘, 박 영찬
이 재성, 이 운재, 서 광준,
최 병진, 이 승미, 강 병주,
이 공주, 장 병규, 최 용석,
허 욱, 박 준식, 김 남일,
김 선배, 이 주호, 천 정훈,
황 금하, 김 남경, 김 은화

위탁연구기관명: 고려대학교

위탁연구책임자: 이 성환

위탁연구기관명: 오롬테크

위탁연구책임자: 김 문호

위탁연구기관명: 원광대학교

위탁연구책임자: 이 용주

위탁연구기관명: 동국대학교

위탁연구책임자: 변 정용

위탁연구기관명: 숙명여자대학교

위탁연구책임자: 김 성혁

위탁연구기관명 전북대학교

위탁연구책임자: 안 동언

위탁연구기관명 우송산업대학교

위탁연구책임자: 이 창조

여 백

요 약 문

I. 제목

통합 국어정보 베이스

II. 연구개발의 목표 및 중요성

컴퓨터의 일반적 사용은 언어에 기반을 두고 있으며, 대부분의 소프트웨어들은 다양한 정보를 표현하는 언어의 처리에 점차 비중을 두어가고 있다. 따라서, 다양하고 고도화된 실용적인 언어 처리 시스템의 구성을 위해서는 다양한 사용자 언어의 분석을 위한 대규모 언어정보베이스 구축, 생산적•효율적인 언어정보베이스 구축을 위한 언어자료 가공도구 개발 및 표준규격 마련 등의 연구기반이 필요하다.

본 연구과제인 통합 국어정보 베이스는 효율적인 국어공학 연구에 필요한 대용량 국어자료의 구축•관리•통합•보급•활용을 통하여 국어정보처리를 촉진하고, 국어정보의 표준화를 유도하여 호환성을 향상시키며, 국어공학 발전의 촉진과 이용편의를 도모하여 우리말 컴퓨터 개발에 필요한 기반 기술의 개발을 목표로 한다.

III. 연구개발의 내용 및 범위

(0) 국어정보베이스 통합환경 및 지원관리 시스템 개발

- 표준 모듈 인터페이스(쉘)
- 텍스트 코퍼스 및 전자사전 개발 관리 시스템:

표준 텍스트 코퍼스 및 전자사전 형식의 정의, 텍스트 코퍼스 및 전자사전 관리 도구 개발

- 한국어 형태태깅 시스템 : 한국어 어절 태깅 모델에 기반한 자동 태거와 태깅 워크벤치 구성
- 한국어 구문태깅 시스템 : 구문태깅 환경 시스템 구축
- 한국어/영어 정렬 시스템: 한/영 정렬 시스템의 완성 및 워크벤치 개발

(1) 국어정보베이스 표준규격화

- 전문용어 개발관리 시스템

기본 TEI-K 및 DTD 개발, SGML-K 보완, 대응 에디터 및 파서 개발

전문용어 사전 구축 지침서 작성, 샘플 전문용어 사전 구축

- 한국어 입출력 표준환경:

한글 및 국어정보처리용 기본 공통 루틴 및 도구 개발

- 균형화 코퍼스 구축 표준방법론 및 시범패키지:

300 만 어절 규모의 균형 코퍼스 및 태그주석 코퍼스 시범패키지 구축

- 품사 사전 규칙과 시범패키지:

품사분류의 표준화 및 품사 분류 변환 및 검증 시범패키지 구축

(2) 국어기반 데이터베이스 구축

- 6 만 문장 규모의 구문구조 부착 코퍼스 구축
- 문장레벨의 낭독 음성 DB 보급판 구축
- 사용 빈도순 상위 1,500 자 규모의 글씨 DB 구축

IV. 연구개발결과 및 활용에 대한 건의

1. 주요 연구개발결과

(0) 국어정보베이스 통합환경 및 지원관리 시스템 개발

- 표준 모듈 인터페이스(쉘):

국어정보베이스 서비스 체계 확립, 전용 WWW 서버 구축(http://kibs.kaist.ac.kr/), 100 여개의 WWW 페이지 및 20 여개의 CGI 프로그램 작성

- 텍스트 코퍼스 및 전자사전 개발관리 시스템:

표준 사전/텍스트 코퍼스 형식의 정의(SDML version 1.0), TDMS(Text Corpus and Dictionary Management System) 기본규격 작성, TDMS Client-Server 시스템 및 Stand-alone 시스템 개발

- 한국어 형태태깅 시스템:

새로운 형태·통사 태그집합의 설정에 따른 2 차년도 시스템 수정·보완, 태그주석 교정지원도구 개발

- 한국어 구문태깅 시스템:

구문태그 설정과 트리주석 코퍼스의 재설계, 한국어 부분 구문 분석기 개발, 한국어 구문 규칙 개발, 트리주석 교정지원도구 개발

- 한국어/영어 정렬 시스템:

한국어/영어 병렬코퍼스에 대한 단어단위 및 구단위 정렬 모델 설계 및 시스템 원형 개발, 2 만 문장 규모의 학습용 병렬코퍼스 구축, 대역 사전 구축 워크벤치 구성 및 개발

(1) 국어정보베이스 표준 규격화

- 전문용어 개발 관리 시스템 및 시범 패키지:

TEI 의 분석과 한글문헌 구조분석을 통한 TEI-K 개발, 전문용어 사전을 위한 기본 DTD 개발, SGML-K 보완, TEI-K 를 위한 관리 도구 개발

TIF(Terminology Interchange Format)에 기반한 전문용어 교환형식 규격 작성, TIF 형식에 따른 전문용어 사전 구축지침서 작성, 정보학 분야와 전산학 분야의 각 용어 100 개로 구성된 전문용어 사전 시범패키지 구축

- 한국어 입출력 표준환경:

코퍼스 입력코드 규격 작성, 1 차년도 한글 글자꼴 보완, 70 개의 함수로 구성된 국어정보처리용 C 라이브러리 구축, 정음형 모자익 개발

- 균형화 코퍼스 구축 표준 방법론 및 시범패키지:

균형 코퍼스 구축 방법론 개발, 300 만 어절 규모의 균형 원문 코퍼스 및 태그주석 코퍼스 구축

- 품사 사전 규칙과 시범패키지:

62,164 개의 표제어를 갖는 품사변환 사전 작성, 세부과제 시스템 간의 품사변환기 개발, 54 개의 형태•통사 태그집합 설정

(2) 국어기반 데이터베이스 구축

- 6 만 문장 규모의 트리주석 코퍼스 구축
- 70 명분의 단어레벨의 낭독음성 DB 보급판 구축:

단독 숫자 41 종 및 4 연 숫자 35 종, 단문 1 종, PBW 452 어절, 고빈도 2,000 어절의 음성 DB 구축, 문장 레벨의 음성 DB 구축

- 사용 빈도순 상위 한글 1,500 자에 대한 글씨 DB 구축:

1,000 자 x1,000 벌(정서체 500 벌, 자유필체 500 벌)의 8 비트 명암영상 글씨 DB 구축

2. 활용에 대한 건의

- 우리말 정보처리를 위한 기초연구의 기본 플랫폼
- 우리말 인터페이스 도구 개발
- 정보 확산 및 교환의 매개체
- 개발된 연구결과를 TECH-MART 의 형태로 정기적으로 공개하여 기업체 참여 유도
- 개발된 연구결과의 베타판 공개에 의한 응용제품 연구개발의 조기 활성화 유도

Summary

I. Title

Integrated Korean Information Base

II. The Objects and the Importance of the Research and Development

Computer processing nowadays is generally based on a language or languages, and most software programs have focused on the proper handling of the languages that represent various kinds of information. Therefore, in order to construct various sorts of advanced and practical language processing systems, it is necessary to provide a research platform that includes the construction of large volume of language information base for deep linguistic analyses, the development of language tools for productive and efficient construction of the language information base, and the preparation of standards and specifications.

The following are the objects of this research, the integrated Korean information base. First, it aims to expedite the progress of Korean information processing through the construction, management, integration, distribution, and the practical use of large-volumed Korean information which is essential for the efficient and effective research on Korean language engineering. Secondly, it is to improve portability by the guidance and the standardization of Korean information. And finally, based on the progresses of the Korean information and the acquaintances of the users to the language engineering, the research sets a goal to develop the profound and basic technologies on demand for the development of our own-language computers.

III. The contents and the range of the research and development

(0) The development of an integrated, environment and support management system for Korean information base

- a standard module interface (shell)
- a text corpus and electronic dictionary development and management system:

the definition of standard forms for texts and electronic dictionaries, and the development of a prototype for the text and electronic dictionary development and management system.

- Korean part-of-speech tagging system:

the enlargement of the first-year system functionality and the establishment of the environments for the tagging.

- Korean syntactic tagging system:

the enlargement of the first-year system functionality and the establishment of the environments for the tagging.

- Korean/English alignment system:

the enhancement of the first-year model and the development of the prototype alignment system.

(1) The standardization and the specification for Korean information base

- a terminological data base development and management system:

the development of basic TEI-K and DTDs, the upgrade of SGML-K, and the development of corresponding editors and parsers.

the preparation of the guidelines for the establishment of the terminological dictionaries, and the construction of a sample dictionary.

- a standard Korean input/output environment:

the development of basic, common routines and tools for Hangul and Korean information processing.

- a standardized methodology for the construction of a balanced corpus, and an example package:

a balanced corpus and a tagged (annotated) corpus of three million word phrases.

- part-of-speech transfer dictionary rules and an example package:

the standardization of part-of-speech classification, and the development of an example package for the conversion and verification among different part-of-speech classifications

(2) The construction of Korean information base

- the construction of bracketted corpus of one hundred thousand sentences.
- the construction of word-level narrative speech data base for distribution.
- the collection of one thousand hand-written letter scripts based on the frequency of use.

IV. Research results and recommendations for future applications

1. Major results of the research and the development

(0) The development of an integrated, environment and support management system for Korean Information Base

- a standard module interface (shell):

modelled the conceptual diagram of Korean information base, constructed a web server for the information base (http://kibs.kaist.ac.kr/), integrated information bases and tools through more than one hundred web pages and twenty CGI programs.

- text corpus and electronic dictionary development and management system:

defined a standard dictionary/text markup language (SDML version 1.0), developed the preliminary specification for TDMS(Text Corpus and Dictionary Management System) , and developed a prototype of TDMS.

- Korean part-of-speech tagging system:

updated the first-year system according to the upgraded part-of-speech tag set, and developed a tool for correcting the tagged corpus. Tagging Workbench is developed.

- Korean syntactic tree-tagging system:

defined syntactic tags and redesigned tree-tagged corpus, developed Korean partial parser and Korean syntactic rules, and developed a tool for correcting tree-tagged corpus.

- Korean/English alignment system:

designed and built a prototype for word- and phrase-level alignment system for Korean/English parallel corpus, and constructed a parallel corpus of twenty thousand sentences for the training purpose. Translation Dictionary Workbench is developed.

(1) The Standardization and Specification for Korean Information Base

- a terminology development and management system:

- developed TEI-K by analyzing TEI and the structures of Korean documentation, designed DTDs for the terminology dictionaries, upgraded SGML-K, and developed a management tool for TEI-K.
- developed a specification on the interchange format of the technical vocabularies based on TIF(Terminology Interchange Format) and prepared a guideline on the construction of terminological dictionaries, and developed an example dictionary with sample entries of one hundred each from information processing domain and computer science domain

- a standard Korean input/output environment:

devised an input code scheme for corpus handling, upgraded the first-year Hangul fonts, developed about seventy C libraries for Korean information processing, and developed Jung-Eum Mosaic browser.

- a standardized methodology and a prototype for the balanced corpus:

designed a methodology for the construction of the balanced corpus, and constructed the raw and the tagged, balanced corpus of three million word phrases.

- part-of-speech transfer dictionnary rules and a prototype:

constructed a part-of-speech tag conversion dictionary with 62,164 entries, developed a tag converter, and defined a standard lexico-syntactic tag set of 54 tags.

- an example package for the terminological data base:

(2) The Construction of Korean Information Base

- developed a tree-tagged corpus of one hundred thousand sentences.

- developed a word-level narrative speech data base from seventy sources:

41 entries for one-digit numbers, 35 four-digit numbers, one single paragraph, 452 phonetically balanced words (PBWs), and two thousand words of high frequency. Sentence-level speech database.

- developed one thousand hand-written Hangul scripts of high frequency:

developed 8-bit, black/white scripts data base for one thousand characters from one thousand different sources

2. Recommendations for future applications

- serves as a basic platform for the profound researches for Korean information processing.
- develops interfacing tools for Korean language.
- serves as a medium for information dissemination and interchange.
- guides the involvement of industries through the periodic opening of research products and results in Tech-Mart form.
- guides rapid activation of applications research and development through the release of previously developed beta results.

Contents

목차

1. 통합 국어정보베이스의 개요

한국과학기술원
최기선

여 백

1. 통합 국어정보베이스의 개요

1장. 연구개발의 목표 및 내용

통합국어정보베이스는 대용량의 국어자료를 구축, 통합, 관리하고 보급하는 것을 주 목적으로 한다. 이러한 노력은 효율적인 국어공학의 연구에 필요한 정보를 구축하여 여러 연구자간의 자료 공유와 학제적 연구의 활성화를 목표로 하며 국어정보처리의 촉진과 국어공학의 발전 이용의 편의 도모를 위함이다. 이러한 노력외에 국어정보의 표준화를 유도하여 호환성을 향상하고 우리말 컴퓨터 개발에 필요한 기반 기술의 개발을 병행한다.

이러한 목표를 위하여 당해년도인 3 차년도(1996.9-1997.8)에는 1 차년도와 2 차년도에 구축, 개발된 자료와 도구를 통합하고 평가하여 부족한 점을 보완하여 실용화를 꾀하는데 주력하였다. 1 차년도의 연구는 단어중심의 기본 모델 설계 및 기반 기술의 원형 구축을 위한 연구를 행하였으며, 2 차년도에는 기본적으로 구축된 자료와 도구를 개발하고 통합 표준안 개발 및 대량의 데이터베이스 구축 연구를 행하였다. 당해년도의 주된 연구는 국어자료의 대용화화를 추진하여 통합적 자료로서 모습을 갖추게 하고, 각종 도구의 성능향상과 실용화에 역점을 두었다. 이러한 목표를 위한 세부 연구 내용은 다음과 같다.

1. 국어기반 데이터베이스 구축

((1). (2)의 연구결과는 원문 코퍼스 1,000 만 어절을 제외한 나머지는 전적으로 문화체육부 지원에 의해 구축되고 있음.)

(1) 대규모 원문 코퍼스

한국어의 언어적 행태와 특이성을 알아보고 연구의 기초자료로 사용되는 원문 코퍼스를 구축한다. 1,2 차년도를 통하여 누계 4,300 만 어절을 구축하였으며 3 차년도는 누계 7,800 만 어절 구축을 목표로 연구 진행되고 있다. (1997.12. 구축 완료 예정: 문체부 지원)

(2) 품사부착 코퍼스 구축

한국어의 품사적 특이성과 자동 태거의 모델 학습, 구문파싱의 기초자료로 사용이 되는 품사부착 코퍼스는 1,2 차년도 누계 500 만 어절이 구축되었으면 3 차년도의 연구는 누계 1,500 만 어절을 목표로 연구 진행되고 있다 (1997.12 구축 완료 예정 : 문체부 지원)

(3) 구문구조 부착 코퍼스 구축

한국어 구문구조를 부착한 코퍼스로서 2 차년도 누계 10 만 문장이 구축되어 있다. 3 차년도의 연구는 6 만 문장의 구문트리 부착 코퍼스의 구축되었다.

(4) 음성 데이터베이스 구축

한국어의 음성 DB 를 구축하는 작업으로 단어음성 DB 로 70 명의 4 회 발성분이 구축되어 있고 문장 DB 로 100 문장의 50 명 발성이 구축되었다. 또한 음소레이블링 작업으로 PBW 10 명분이 구축완료 되었다.

(5) 필기체 글씨 데이터베이스 구축

KSC 완성형 한글 사용 빈도순 상위 1,500 자 1000 벌이 구축 완료되었다. 이러한 연구는 데이터베이스 구성 방식의 설계를 근간으로 하며 데이터베이스 검증 및 보완 도구의 개발도 포함한다.

2. 국어정보베이스 통합 환경 및 지원관리 시스템 개발

(1) 표준 모듈 인터페이스 셀

사전, 코퍼스, 음성 및 필키체 데이터베이스등 국어공학관련 정보베이스를 통합적으로 관리 및 유지할 수 있는 환경의 개발로서 국어공학 관련 정보의 제공, 관리 및 유지의 일관성 및 편의성을 도모하도록 하는 통합구조를 구축하였다.

(2) 텍스트 및 사전 데이터베이스 개발 관리 시스템 (TDMS)

중복된 사전 및 사전 관리 프로그램의 개발등의 중복된 작업의 개선을 위하여 SGML 에 기반한 표준 사전 기술 언어 (SDML)을 기본으로 하여 여러 응용프로그램에서 사전을 공유할 수 있도록 관리시스템을 개발하였다. 본 관리시스템은 응용프로그램에 종속적이던 사전 관리 개념을 유연하고 독립적인 개념으로 전환하였으며 사전의 표준화를 가능하도록 하게 한다.

(3) 형태태깅 시스템 (품사부착 워크벤치)

형태소 분석기의 결과로부터 자동으로 품사를 부착하는 기능을 주된 목적으로 한다. 형태태깅을 위한 확률적 한국어 모형을 HMM 을 이용하여 모델링 하였다. 이러한 기능 외에 자동 품사 추정의 오류를 보정하고 사용자의 수정기록을 통한 수정 후보 제시의 기능을 제공함으로써 품사 부착 코퍼스의 구축을 효율적으로 행하며 구축방법론의 확립에 기여한다. 이러한 버전은 UNIX 와 Win95 버전을 동시에 개발 완료하였다.

(4) 구문태깅 시스템 (구문구조 부착시스템)

구문트리 코퍼스의 기본 제약 조건을 위반하는 문장에 대한 자동 검증 및 수정 기능을 개발한다. 본 시스템은 구문트리 태깅의 기본 규칙을 자동 학습할 수 있으며 품사 부착된 문장을 효율적으로 구문 구조 부착하여 구문태그 부착 코퍼스 구축에 효율 및 일관성을 유지할 수 있다.

(5) 한국어/영어 정렬 시스템

한국어와 영어로 구성된 코퍼스에서 서로 대응되는 대역 어휘를 자동 추출하는 시스템으로 다량의 코퍼스에서 두 언어의 언어모델을 확률적으로 자동 학습하여 대역 어휘를 자동 생성한다. 이러한 방법론은 번역 용례 및 용어의 검증을 위한 워크벤치로 확장 가능하며 한국어에 대한 통계적 기계번역 방식의 기초 기술을 제공 한다.

3. 국어정보베이스 표준 규격

(1) 문서 구조 표현을 위한 표준화 규격

TEI 에 기반하여 TEI-K 의 DTD 를 개발한다. 이러한 개발된 표준화 패키지에 기반하여 용어 데이터베이스 구축을 위한 DTD 를 제작하여 앞으로의 문서구축의 가이드라인을 제공한다. 본 연구에서 샘플용어 데이터베이스를 시범적으로 구축하여 표준화 규격의 적합성을 검증하였으며, SGML 문헌 처리 기반기술을 제공한다.

(2) 한국어 입출력 표준환경

IOS 10647 에 근거한, 한국어 처리에 적합한 다수바이트 정음형 한글 코드를 개발하였다. 단순한 코드의 개발이 아닌 코드의 보급과 활용을 위하여 한글 환경 터미널의 hunterm 과 한글 편집기 Hunvi 를 개발하였다. 이러한 도구는 정음형을 축으로 하는 통합 코드 변환기를 개발하는 것을 포함하며 PC 용 정음형 기본 입력 루틴의 library 도 개발 하였다.

(3) 품사사전 규칙과 시범 패키지

한국어 품사 태그의 연구자간의 다양한 품사 태그 패키지를 일원화하고 표준화를 유도한다. 62,164 개의 표제어를 갖는 품사 변환 사전을 작성 하였고 각 세부 시스템간의 품사 변환기를 개발하였다. 품사변환기는 54 개의 형태, 통사 태그 집합을 설정하였고, 확장 품사 사전 규칙을 통하여 변환을 행한다. 이러한 연구는 국어정보베이스 통합에 따른 시너지 효과를 기대할 수 있으며 세부 개발 시스템의 호환성과 가치를 증대시키는 역할을 한다.

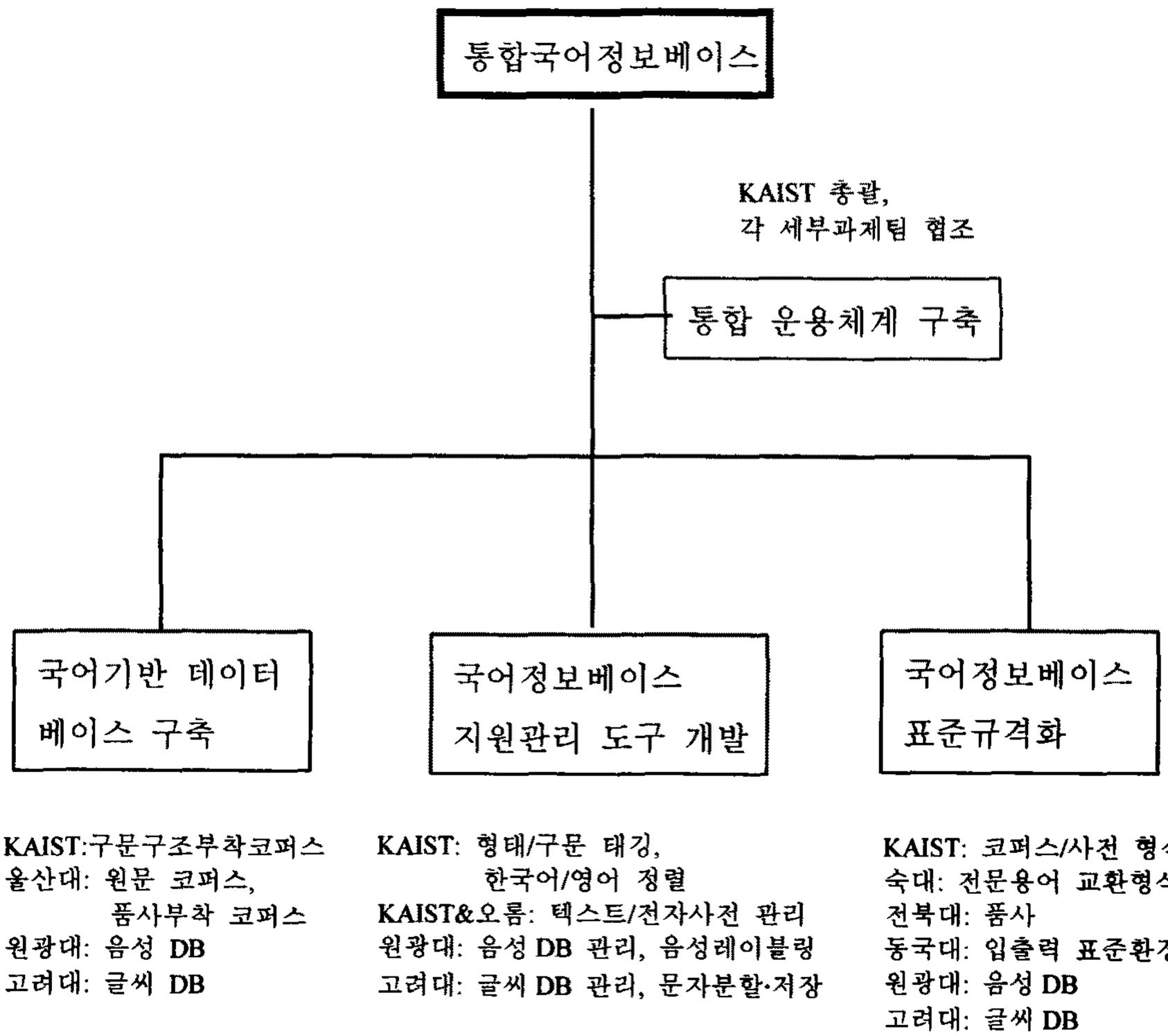
2장. 연구 수행 방법

1절. 연구추진체계

- 독단적인 연구를 막고 협동적인 연구를 유도하기 위한 목적 이외에 위탁연구 관리의 통합관리 운영체제를 확립하기 위하여 [그림 1.1]과 같이 연구개발체계를 조직화함.

- 문체부 지원 영역과 상호보완관계가 되도록 연구·조직화함. 즉, 문체부 지원 부분인 원문 코퍼스, 품사부착 코퍼스와 그에 해당하는 지원관리 도구는 분리·개발하도록 함.

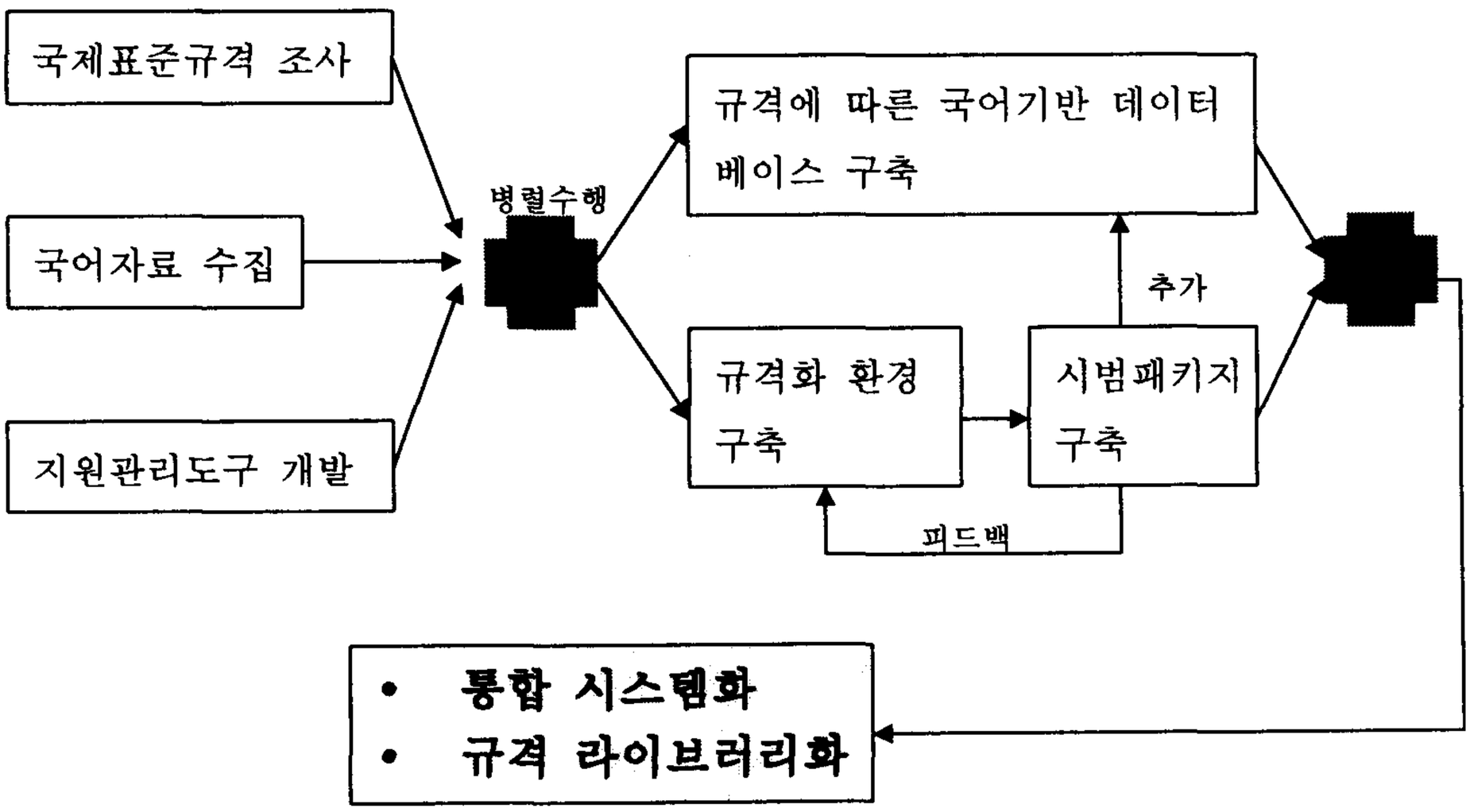
- 통합국어정보베이스의 효율적인 보급과 홍보를 위하여 각 세부과제팀들의 적극적인 협조를 바탕으로 과제 총괄 기관인 한국과학기술원(KAIST)에서 WWW 서버를 구축, 관리, 운영함.



[그림 1.1] 통합국어정보베이스 연구추진체계

2 절. 연구추진방법

- 통합국어정보베이스의 목표를 효율적으로 달성하기 위하여 표준규격화, 국어 자료 수집 및 국어정보베이스 구축관련 지원관리 도구 개발 작업을 [그림 1.2]에서와 같이 병렬로 진행함.
- 표준규격의 검증과 작업 공정의 개선을 위하여 국어정보베이스 표준규격화 관련 세부과제들 각각에 대하여 시범패키지를 구축함.
- 베타 실험용 보급을 위하여 세부과제의 시작품들을 통합하고 표준규격을 라이브러리화함.



[그림 1.2] 통합국어정보베이스 연구추진방법

3절. 연구개발 추진 경과

1. 네트워크를 통한 분산 체제 개발

- 인터넷 WWW 이용
- 연구결과물 교환
- 통합 표준 개발 관리 시스템 TDMS에 의한 규격화
- WWW Home Page 구축

2. 정기적 워크샵 개최

- 96/3/14 - 6/11 : 품사태그 표준규격 소위원회 1~8차, KAIST, 대전
- 96/4/26 - 4/27 : 통합국어정보베이스 중간 발표, 대덕롯데호텔, 대전
- 96/6/14 : 통합국어정보베이스 표준규격 워크샵, 과총회관, 서울
- 96/6/22 : 확대 표준규격 위원회, KAIST, 대전
- 96/7/11 : 제 1회 우리말 정보처리 규격 심포지움, 국립중앙도서관, 서울
- 97/6/28 : 제 2회 우리말 정보처리 규격 심포지움, 고려대, 서울
- 97/5/30-5/31 : 국어정보베이스 표준화 전문가 회의, Expo 호텔, 대전

3. 연구진행 문서집

- 당해연도 중간/최종 보고서
- 워크샵 발표집 (4/26-27, 6/14, 6/22, 7/11)

3장. 연구개발 결과

1절. 연구 결과

1. 국어정보베이스 통합환경 및 지원 관리 시스템

계획	실행
ㄱ. 표준모듈 인터페이스 - 종합환경으로의 발전 - 모듈간의 표준 API 제공	ㄱ.ㄴ.ㄷ. 통합표준 개발 관리시스템으로 일원화 - TDMS (텍스트사전 개발 관리시스템) 표준접속 표준정보 마크업 기술언어 사용자 인터페이스 자동 생성 DBMS 자동 연결 - (1 ㄱ) 전문용어 개발관리시스템의 TDMS로 일원화
ㄴ. 텍스트 데이타베이스 개발관리시스템 - 표준텍스트 화일 포멧 정의 - 텍스트데이터베이스 관리도구	
ㄷ. 전자사전 개발 관리 시스템 사양 - 표준전자사전 포멧 정의 - 전자사전 관리도구 개발	
ㄹ. 형태태깅 시스템 - 태깅시스템 확장(자동태깅알고리즘) - 문장태깅 - 워크벤치 개발	- 계획대로 진행 - 태깅 워크벤치로의 발전
ㅁ. 구문태깅 시스템 - 언어이론에 독립적, 일반적, 통계적 구문태깅 - 간결하고 확장성 있는 저장형식 - 품사부착코퍼스-구문구조부착 코퍼스 자동 생성	- 계획대로 진행
ㅂ. 한국어/영어 정렬 시스템 - 대역어휘 구조 정렬 모델 개선 및 구현	- 계획대로 진행

2. 국어정보베이스 표준 규격

계획	실행
ㄱ. 전문용어 개발 관리시스템 및 시범 패키지 - SGML 및 TEI 기반 전문용어 관리 시스템 개발 - TEI-K 에디터 및 파서 제공 - 전문용어 교환포맷 (TIF) 개발 - TIF 에 기반한 박물관 전문용어 사전 시범구축	(0) ㄱ,ㄴ,ㄷ 통합표준개발관리 시스템으로 일원화 - 표준 SGML-K, TEI-K 계획대로 진행 - [변경] 전문용어 영역변경 : 박물관 ⟶ 컴퓨터 - 정보학 분야 용어 100 개로 된 전문용어 사전 시범 패키지 구축
ㄴ. 한국어 입출력 표준환경 - 한글창제 원리에 입각한 표준 라이브러리 - 한국어 정보처리에 최적코드 개발 및 기본 프로그램	- 계획대로 진행 - 목적을 국어원본 자료의 원형 보존용으로 잠정적 국한 (규격위원회 및 우리말정보처리 규격 심포지움 건의 사항) - UNIX 환경용 에디터, 코드변환기 개발
ㄷ. 균형화 코퍼스 구축 표준 방법론 및 시범패키지 - 균형코퍼스 구축방법론 - 300 만 어절 시범 패키지	- 계획대로 진행
ㄹ. 품사사전 규칙과 시범 패키지 - 표준 품사분류 기준 및 다른 품사 규칙간의 변환기 개발	- 계획대로 진행

3. 국어기반 데이터베이스 구축

계획	실행
ㄱ. 텍스트 데이타베이스 (문체부지원)	- 계획대로 진행
ㄴ. 품사부착 텍스트 데이타베이스 (문체부 지원)	- 계획대로 진행

ㄷ. 구문구조 부착 텍스트 데이타베이스 - 10 만 문장 규모 구축	- 계획대로 진행 (정교한 수정본은 차기년도에 완료함) - 표준구문태그 완료
ㄹ. 음성 데이타베이스 - 70 명분의 단어레벨의 낭독음성 DB 보급판 구축, 문장레벨 음성 DB 구축	- 계획대로 진행
ㅁ. 필기체 데이타베이스 - 사용빈도순 상위 한글 1,500 자 글씨 DB 구축	- 계획대로 진행

(1) 주요 연구 성과 및 보급 현황

1) 특허

- 텍스트 및 전자사전 관리 도구 (출원중) : 응용프로그램에 독립적인 전자사전 및 코퍼스 관리 시스템
- 표준사전 표기 언어 (SDML) (출원중) : SGML 에 기반한 표준 사전 표기 언어

2) 기술 이전

- 텍스트 및 전자사전 개발 관리 도구 : 전자 사전 및 텍스트 마크업 기술 규격

3) 연구 성과 보급

- 국어정보베이스 CD-ROM

  보급 일시 : 1997.5.30 국어정보베이스 표준화 전문가 회의

  본 과제에서 개발한 자료, 도구, 환경에 대한 시험판 공개

  TDMS, 텍스트코퍼스, 품사부착코퍼스, 구문구조 부착코퍼스

● 대한민국 국어정보베이스 (시험판)

보급 일시 : 97.6.28 제 2 회 우리말 정보처리 규격 심포지움

TDMS, 시험용 언어자료, 음성 DB,, 필기체 DB 의 시험 보급판

여백

2. 텍스트코퍼스 및 전자사전 관리시스템 (TDMS)

한국과학기술원
최기선

여 백

2. 텍스트코퍼스 및 전자사전 관리시스템(TDMS)

1장. 서론

자연언어처리 프로그램들은 대개 많은 양의 문법 정보, 의미 정보, 용례 등을 필요로 한다. 이러한 정보는 전자 사전을 통해 제공이 되며, 제공되는 정보의 양과 질에 따라 프로그램의 성능도 매우 영향을 받는다.

그러나 전자 사전을 작성하는 일은 매우 노동 집약적이면서도 전문지식을 필요로 하며, 이러한 사전은 일단 완성이 되었더라도, 계속 새로운 단어를 첨가해야 하고, 경우에 따라서는 새로운 필드나, 정의 등이 추가됨으로써, 계속 새로운 형태의 사전을 만들어야 하는 경우가 많다.

효율적으로 사전을 구축하기 위해서는 사전 구축 과정 중에도 이미 입력한 단어나 내용을 쉽게 확인하고 참고할 수 있어야 하며, 다른 사전의 내용을 참조하거나, 다른 사전의 내용을 복사할 수 있으면 좋을 것이다. 특히 이미 만들어진 여러 종류의 사전들과 텍스트 코퍼스 등에서 필요한 정보를 자동으로 추출할 수 있다면, 새로운 사전의 생성이나 사전의 갱신들을 효과적으로 할 수 있을 것이다. 요즈음 컴퓨터에 입력된 종이 사전에서 사전 정보를 추출해 사용하는 사례들도 발표되고 있다.

텍스트코퍼스 및 사전 관리 시스템(TDMS)은 다양한 형태의 사전으로부터 필요한 정보를 추출, 변형하고, 편집할 수 있는 환경을 제공해 준다. 이를 위해서는 다양한 형태의 사전을 기술할 수 있는 방법과 이를 운용할 수 있는 방법이 있어야 하는데, TDMS 에서는 표준 사전 표기 언어(SDML: Standard Dictionary Markup Language)를 정의하여 사용하고 있다.

SDML 은 ISO 표준인 표준 일반 표기 언어(SGML)에서 TDMS 에 필요한 일부분의 기능만을 사용한 것이며, SGML 과 관련된 기술인 TEI(Text Encoding Initiative), TIF(Terminology Interchange Format), DSSSL(Document Style Semantics and Specification Language), HyTime 등의 일부 정의들도 역시 사용하였다.

TDMS 의 대상으로는 우선 자연언어 처리 프로그램 사전을 고려하고 있다. 즉, 형태소 해석기용 사전, 구문 해석기용 사전, 기계 번역 시스템의 해석, 변환, 생성용 사전 등을 포괄적으로 지원할 수 있는 시스템을 고려한다. 자연 언어 처리 프로그램용 사전은 명확한 필드의 구분과 사용 방법이 있으므로 정의하기 편리하지만, 자질값과 같은 표준화되지 않은 속성을 처리해야 하는 어려움이 있다.

일반 국어 사전의 경우, 그 형식이 다양하고 그 표시 형식에 따라 의미하는 바가 다를 수도 있는 등 일반 사전을 그대로 전자사전 형태로 표현하는 데는 한계가 있을 수 있다. 따라서 TDMS 에서는 구분 가능한 형태의 일반 사전을 표현한다.

TDMS 에서 텍스트 코퍼스에 대한 처리는 현재 단순한 관리 기능만을 제공한다. 즉, 각 텍스트 화일의 관리정보를 정형화시켜서 필드로 정의하고, 실제의 텍스트는 가변 크기의 필드로 임의의 텍스트가 포함될 수 있도록 한다. 각 텍스트 화일의 관리정보는 텍스트 헤더 정보로 저장되어 필요한 코퍼스 텍스트를 찾는 키로 이용될 수 있다. 실제 텍스트 필드에 대한 처리는 거의하지 않고 단순한 텍스트로만 인식하여 처리한다.

본 프로젝트의 1 차년도에는 관련연구와 프로토타입을 구현하였고, 2 차년도에는 1 차년도의 결과를 기본으로 하여 표준사전 형식을 정의하고, TDMS 의 규격 버전 1.0 을 작성하였다. TDMS 번전 1.0 은 Windows95 에서 작동하는 Stand-alone 버전과 Win/NT 에 Server 를 두고 Windows95 의 Client 로 연결하여 사용하는 Server/Client 버전으로 나누어져 있다. 3 차년도인 금년에는 개선된 배포용 Stand-alone 버전 1.0 을 제작하였고 기존의 TDMS 를 13 개 부분 모듈로 나누어 재구성한 TDMS Server/Client 버전 2.0 을 구현하였다.

2 장. 기존 사전 형식 연구

1 절. 일반 사전 형식

1. 국어사전의 마크로 구조와 마이크로 구조

(1) 마크로 구조

사전을 편찬할 때에 표제어가 어떠한 순서에 의해 배열되어지는가에 대한 구조가

마크로 구조이다. 즉 표제어의 배열에 있어서 자모의 종류와 이들 사이에 어떠한 차례가 정해져야 하는가가 바로 마크로 구조를 어떻게 이루어야 하는지의 문제이다. 여기서는 마크로 구조가 어떻게 이루어져야 하는가는 우선 문제가 아니므로, 국어사전에 나와있는 마크로 구조를 간단히 살펴보기만 한다.

현재 사용되고 있는 국어사전의 초성, 중성, 종성의 종류와 순서는 다음과 같다.

ㄱ. 초성

ㄱ ㄲ ㄴ ㄵ ㄷ ㄸ ㄹ ㄺ ㅁ ㅂ ㅃ ㅄ ㅅ ㅆ ㅇ ㅈ ㅉ ㅊ ㅋ ㅌ ㅍ ㅎ

ㄴ. 중성

ㅏ ㅐ ㅑ ㅒ ㅓ ㅔ ㅕ ㅖ ㅗ ㅘ ㅙ ㅚ ㅛ ㅜ ㅝ ㅞ ㅟ ㅠ ㅡ ㅢ ㅣ ㆍ

ㄷ. 종성

ㄱ ㄲ ㄳ ㄴ ㄵ ㄶ ㄷ ㄹ ㄺ ㄻ ㄼ ㄽ ㄾ (ㄿ) ㅀ ㅁ ㅂ ㅄ ㅅ ㅆ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ

또한 표제어가 동일한 경우(HOMOGRAPH)에는 어법에 따라, 체언→용언→수식사→조사의 순서로 배열하고, 또 어법이 같은 경우에는 우리말→한자어→외국어의 순서대로 배열하였다. 또한 모든 조건이 같은 경우에는 현대어→옛말, 일반어→전문어의 순서대로 배열하였고, 이들은 각각 어깨번호 ^{1}, ^{2}, ^{3} 등으로 표시하였다.

(2) 마이크로 구조

개별 표제어에 대하여 어떠한 내용이 어떤 순서에 의하여 배열되는가의 문제는 바로 마이크로 구조의 문제이다. 기존의 국어사전에는 각 표제어에 대한 정보들이 다음과 같은 구조를 지니고 있다.

표제어

표제어 번호 (동철자인 경우: HOMOGRAPH)

한자* | 로마자

발음

품사*

활용형

고유어 어원|외래어 어원

전문어 표시

용법

의미*

표제어Alternative

표제어Alternative 구분: 준말, 늘임말, 동의어,상위어 등

수치

주의

용례

출처

속담

성구/관용구

조어확장자: 확장가능 표시: 복합어

파생어

품사 --- 명사 + 하다, 되다

히

동사 --> 부사

대부분의 국어사전이 이와 같은 구조 속에서 필요한 정보들을 제공하고 있다. 앞으로의 과제는 이러한 마이크로 구조를 다시 검토하여 이를 보다 보편 타당하게 나타내는데 있다.

2 . Longman Dictionary of Contemporary English (LDOCE)

(1) 마크로 구조

표제어는 알파벳 순서에 의해 배열되어 있다. 또한 하나 이상의 단어로 이루어진 구의 경우에는 그 구(phrase)에 대한 표제어가 따로 있게 된다. 그러나 idiom 의 경우에는 특정구가 따로 표제어로 나타나지 않고, 다른 표제어의 내부에 idiom 으로서 특수하게 표시되어 있다. 또한 동철자어(homograph)의 경우에는 품사에 따라 다르게 발음되어 지는데, 이 경우에는 표제어가 어깨번호 ^{1}, ^{2}, ^{3} ...과 함께 따로따로 나타난다. 단 표제어가 두개의 품사를 지니고, 의미의 설명에 있어서 하나로 충분할 경우에는 따로 표제어로 등록하지 않는다.

(2) 마이크로 구조

표제어: 음절의 구분이 표시되어 있다. 표제어의 변이형이 나타난다.

예) **caf-tan, kaftan; gen-e-ral-ize, -ise**

표제어 번호Entry number (동철자인 경우: HOMOGRAPH)

발음: 국제음성부호(International Phonetic Alphabet: IPA)를 사용하여 표기하며, 영국식 발음과 미국식 발음이 다를 경우 영국식을 먼저 표기하고, 미국식을 표기한다. 주강세(stress)와 부강세(secondary stress)가 표시된다. 외래어 발음의 표기시에는, '(...)'안에 어원을 표시한다.

표제어 Alternative: 예를 들어 ***autumn*** 은 AmE (미국식 영어)의 경우 ***fall***

관련어: 동의어, 반의어 등 품사

활용형: 품사의 활용형이 불규칙이거나, 혼란을 야기시킬 가능성이 있을 때 표기한다.

Pronunciation of Inflections: 활용형의 발음이 불규칙하여 애매한 경우에 표기한다.

다의어 번호: 다의어인 경우에 표시

문법적 정보:

표제어의 문법적 정보는 '[...]'안에 알파벳 대문자와 숫자의 코드로 나타내는데, 이 코드를 위한 도표가 따로 마련되어 있어 여기에 모든 문법적 정보가 총망라되어 있다. 즉 개별 단어가 가질 수 있는

문법적 특성을 전부 모아, 이것으로부터 일반화된 특징들을 분류하여, 각각의 문법적 특징에 대해 특정 코드로 표시한 것이다. 그럼으로써, 표제어가 가지고 있는 통사적인 특징들이 간편하게 나타내어 질 수 있고, 잉여적인 정보들이 불필요하게 반복될 필요가 없게 된다. 여기서 알파벳 대문자와 숫자는 각각 일정한 문법적 특성에 대응하게 되어 있다.

예) [Wn1], [Wa2], [Wv3]. [Wp2] ...

설명

용법

용례: 이태릭체로 쓰여있고, 콜론(:) 뒤에 표시된다.

숙어

공기정보: 숙어는 아니지만 어느 정도 고정되어 결합하여 사용되는 어구는 'in the phr.'라는 설명 뒤에 나타난다.

2절. 자연어 처리용 전자 사전 형식

1. MATES/EK 기계번역 사전 형식

MATES/EK 사전의 구성은 크게 해석 사전과 대역 사전으로 나눌 수 있다. 해석 사전에는 형태소, 구문, 의미 정보가 들어 있으며 대역 사전에는 역어 선택 정보, 생성 정보 등이 들어 있다. 또한 각 사전은 품사별로 나뉘어 관리되며 실제 사용에서는 가상 접근 유틸리티에 의하여 하나의 사전처럼 사용할 수 있다

사전의 구성과 종류

구성

영어 해석 사전

형태소, 구문, 의미 해석을 위한 정보가 모두 하나의 사전에 기술된다.

영한 대역 사전

역어 선택을 위한 정보는 물론 한국어 생성에 필요한 형

태소, 구문 정보를 포함한다.

종류

영어 해석 사전

영어 명사 사전

영어 동사 사전

영어 형용사 사전

영어 부사 사전

영어 대명사 사전

영어 한정사 사전

영어 조동사 사전

영어 전치사 사전

영어 접속사 사전

영한 대역 사전

영한 명사 사전

영한 동사 사전

영한 형용사 사전

영한 부사 사전

MATES/EK 의 사전 형식은 해석 사전과 대역 사전이 다르고, 각 품사별로도 약간씩 차이가 난다.

사전의 형식

영어 해석 사전

전품사 공통 정보

1) Lexical-Unit : 해당 품사의 표제어

2) Usage-ID-No. : 동일한 Lexical-Unit가 여러 가지 용법으로 쓰이는 경우, 각각에 대해 별도의 Usage-ID-No.로 나누어 구분

3) Multi-Lex : 숙어를 기술하는 경우 Head만 1)에 기입하고 나머지는 Multi-Lex에 기술한다.

4) LDOCE-UID : Longman Dictionary of Contempory English에서 해당 표제어가 갖는 용법의 번호를 기입한다.

5) Multi-POS : 동일한 Lexical-Unit가 해당 표제어의 품사 이외에 다른 품사로도 쓰이는 다품사어인 경우 그 품사를 모두 기한다.

명사

1) Sub-Category : 명사 세분류
(고유명사, 보통명사, 수량사, ...)

2) Inflection : 명사의 복수형 활용 변화를 표시

3) Properties : 어휘가 갖는 의미적 특성을 있는 대로 복수개 기입(부사성, 동작성, 구체성, ...)

4) Semantic Marker : 명사의 의미 분류표를 참조하여 해당 의미 코드를 기입한다.

5) Number : 영어 명사 수 정보

6) Article : N, A, T 중 하나로 표기

7) Position : 전치/후치 명사

8) Gender :

9) Relation to Prepositional Phrase : 후접 전치사구에 대한 구문 의미적 관계를 표시

10) Relation to Clause : 명사 다음에 오는 절과의 구문, 의미적 관계를 표시

11) Constituents of Prepositional Phrase : 명사가 복합 전치사구를 이루는 경우에 표시

동사

1) Inflection : 표제어인 동사가 3인칭 현재형, 과거형, 현재분사형, 등으로 활용할 때의 변화를 표기

2) LDOCE Verb Types : LDOCE에서 사용하는 동사 패턴의 코드를 그대로 표시

3) Verb Type : 2) 중에서 하나의 동사 패턴만을 기술,

여러 개의 동사 패턴이 있으면 sheet를 나누게 된다.

4) Situation Type : 영어 동사 상황 의미표

5) Voice : 능동/수동태로 사용될 가능성 표시

6) Modality : 표제어가 Modality의 속성을 지니는 경우

7) SUBJ-D-Verb Passive : 이중 목적어를 취하는 동사의 수동태 변환시, 주어가 될 수 있는 목적어를 기술

8) AGT of to-inf : 표제어의 동사 패턴이 V3인 경우에 표기

9) Case Frame : 표제어가 취하는 격구조의 구문 의미 정보를 기술

형용사

1) Inflection : 활용형의 코드 기입

2) Sub-Category : 표제어의 해당 LDOCE코드를 참조하여 기입

3) Properties : 영어 형용사 특성

4) Modality : 표제어가 Modality 속성을 지니는 경우

5) Semantic Types in THE+ADJ : 표제어인 형용사 앞에 THE가 붙어, 명사처럼 쓰이는 경우 의미 범주

6) Case Frame : be동사와 함께 서술 용법으로 쓰이는 경우, 좌우에 위치한 격이 갖는 구문,의미 정보를 기술

7) Semantic Role of subj in to-inf : to부정사와 함께 쓰이는 경우에 기술

부사

1) Sub-Category :

2) Inflection : 활용형 코드

3) Modality : 표제어가 Modality속성을 지니는 경우

4) Properties : 부정성을 띄는 것, 비교급/최상급이 없는 것 등

5) Position : 동사를 중심으로 한 위치

6) Semantic Category : 영어 부사 의미 분류

대명사

1) Sub-Category : 대명사 세분류

2) Central Pronoun : 인칭대명사에 대하여 Person, Gender, Number, Case Function을 표시한다.

3) Relative Pronoun : 관계 대명사에 대하여 표시

4) Interrogative Pronoun: 의문 대명사로 분류된 표제어

5) Indefinite Pronoun : 부정 대명사에 대하여

조동사

1) Inflection : do, be, have 등의 활용 표시

2) Sub-Category : 조동사 세분류

3) Tense Restriction : 표제어가 시제에 대한 제한성을 가지는 경우에 표시

4) Semantic Category of Modal : 조동사 의미 분류

한정사

1) Inflection : 형용사, 부사와 같음

2) Sub-Class : 한정사 세분류

3) Following Noun Type : 수식 받는 명사에 대한 제한성 표시

4) Properties : 한정사 특성 표시

접속사

1) Sub-Category :

2) Semantic Cat.(Deep Case) : 표제어의 Sub-Category가 Subordinator인 경우 접속사 의미 분류

전치사

1) Candidate Deep Case : 후보 심층격

영한 대역 사전

전 품사 공통 정보

1) EK_LEX : 영어 해석 사전의 표제어

2) EK_UID : EK_LEX와 함께 유일 Key가 되는 정보

3) E_UID : 영어 해석 사전에 기입된 해당 표제어의 E-ID
를 똑같이 적는다.

4) E_POS : 영어 표제어의 품사를 영어 해석 사전에 기입
된 정보를 보고 그대로 기입한다.

5) K_POS : 한국어 역어의 품사(POS)를 기입한다.

6) K_LEX_COMPONENT :

apple -> 사과

effect -> 영향을 미치다.

no one -> 아무도 .. 않다.

...

명사

1) K_COUNT_NOUN : 셀 수 있는 명사의 경우 단위를 기입

2) 형용사와의 공기 정보

E_ADJ_LEX : 공기하는 영어 형용사

K_LEX : 대응하는 한국어 역어

K_POS : 대역어의 품사

K_LEX_CHANGE : [COA/REM]

3) 동사와의 공기 정보

E_VERB_LEX : 공기하는 영어 동사

K_LEX_COMPONENT : 동사와 공기하여 선택된 새
로운

한국어 대역어

4) 전치사와의 공기 정보

후접 전치사와의 공기 정보

E_PREP_LEX

E_SYN_FORM

EK_DEEP_CASE

K_SEMANTIC_MK

K_SURF_LEX_COMPONENT

전접 전치사와의 공기 정보

E_PREP_LEX

EK_DEEP_CASE

K_SURF_LEX_COMPONENT

전 후접 전치사와의 공기 정보

E_PRECEDING_PREP

E_FOLLOWING_PREP

EK_DEEP_CASE

K_SEMANTIC_MK

K_SURF_LEX_COMPONENT

동사

1) E_VP : 동사 해석 사전의 Verb_Type
2) E_LEX_COMPONENT : 영어 표제어에 대한 한국어 대역어
3) EK_CASE_FRAME : 동사 영어 해석 사전과 같음
4) K_SURF_LEX_COMPONENT : 심층격에 대응하는 한국어 표층격 조사
5) K_PASS_POS : K_PASS_LEX_COMPONENT의 품사
6) K_PASS_LEX_COMPONENT : 대역어가 능동사인 경우 피동형 어간을 표시한다.

형용사

1) E_AP : 형용사의 용법(한정/서술)
2) EK_CASE_FRAME ,K_SURF_LEX_COMPONENT : 서술적 용법인 경우, 동사 사전 참조
3) K_VERB_ASPECT : 영어 형용사의 대역어인 한국어 동사가 갖는 ASPECT속성
4) K_PASS_POS, K_PASS_LEX_COMPONENT :
5) COLLOCATION VERB INFORMATION : 동사와 함께 쓰이는 경우

부사

1) K_SUBCAT : 한국어 부사의 세분류

2) K_LEX_COMPONENT : 한국어 부사

2. CAT2 기계번역 사전 형식

CAT2 는 유럽의 여러 나라가 공동으로 연구하는 EUROTRA 라는 기계번역 프로젝트에서 나온 기계번역 시스템이다. 이 시스템은 interlingua 방식을 사용하기 때문에 각 언어별 사전의 형태가 동일하며 사전 정보에 의존도가 높다. 다음은 CAT2 에 사용된 사전의 개략적인 구조와 예를 설명한다.

(1) 사전의 일반 적인 구조

```
string=STRING,
lex=LEX,
phon=PH,
role=ROLE,
head={cat=CAT,
      pform=PF,
      restr=RESTR,
      ehead={   cat=CAT,
        num=NUM,
        sem=SEM,
        pvalue=PV}},
frame=FRAME
```

(2) 사전 항목의 예

쓰다

```
lex={lex=ssuta,morph={first=yes,last=no},
     head={cat=v,
           ehead={cat=v,sem={abstract={temp={akt=active}}}}},
     frame={arg1={role=agent,oblig=yes,
                  head={ehead={cat=n,case=nom,sem={anim~=nil}}}},
```

```
                    arg2={role=theme,oblig=yes,
                        head={ehead={cat=n,case=acc,sem={anim=nil,abstract=nil}
                }}}}.[].
    lex={lex=ssuta,morph={first=yes,last=no},
        head={cat=v,
            ehead={cat=v,sem={abstract={temp={akt=stative}}}}},
        frame={arg1={role=agent,oblig=yes,
                        head={ehead={cat=n,case=nom,sem={anim~=nil}}}},
            arg2={role=theme,oblig=yes,
                        head={ehead={cat=n,case=acc,sem={anim~=nil}}}},
            arg3={role=goal,oblig=yes,
                        head={ehead={cat=n,case=nil,sem={anim~=nil},
                                                    pvalue=funct}}}}}.[
        ].
로
    lex={string=lo,lex=p,morph={first=no},pos=post,phon={lp=v},role=funct,
        head={cat=p,p=adv,pform=ulo,restr={head={cat=v}},
                ehead={cat=n,case=nil}},
        frame={arg1={pos=pre,morph={first=yes,last=no},phon={lp=v},
                        head={cat=(n;d;pl;card),ehead={cat=n}}},
            arg2=nil,arg3=nil,arg4=nil,arg5=nil}} >>

    ({head={ehead={pvalue=(dir ; tool)}},
        frame={arg1={head={ehead={sem={anim=nil,abstract=nil}}}}}} ;

    {head={ehead={pvalue=(language;material)}},
        frame={arg1={head={ehead={sem={anim=nil,abstract={temp=nil}}}
    }}}} ;

    {head={ehead={pvalue=funct}},
```

```
                frame={arg1={head={ehead={sem={anim~=nil,abstract=nil}}}}}).
               [].
```

write, employ

```
lex={lex=write,
        head={cat=v,
                ehead={cat=v,sem={abstract={temp={akt=active}}}}},
      frame={arg1={role=agent,oblig=yes,
                head={ehead={cat=n,case=nom,sem={anim~=nil}}}},
            arg2={role=theme,oblig=no,
                head={ehead={cat=n,case=acc,
                        sem={anim=nil,abstract=nil}}}}}.[].
lex={lex=employ,
        head={cat=v,
                ehead={cat=v,sem={abstract={temp={akt=stative}}}}},
      frame={arg1={role=agent,oblig=yes,
                head={ehead={cat=n,case=nom,sem={anim~=nil}}}},
            arg2={role=theme,oblig=yes,
                head={ehead={cat=n,case=acc,sem={anim~=nil}}}},
            arg3={role=goal,oblig=no,
                head={pform=as,ehead={cat=n,case=nil,sem={anim~=n
      il},
                                                    pvalue=funct}}
      }}}.[].
```

walk

```
lex={lex=walk,
        head={cat=n,
            ehead={cat=n,sem={abstract={temp={akt=active}}}}},
    frame={type=static,
            arg1={role=agent,oblig=yes,
                head={ehead={cat=n,case=nom,sem={anim~=nil}}}},
            arg2={role=direction,oblig=yes,
```

```
                                head={ehead={cat=n,case=nil,pvalue=dir,sem={anim=
        nil}}}},
                        pred={head={ehead={cat=v,lex=take}}}}}.[].
```

take

```
lex={lex=kata,morph={first=yes,last=no},
        head={cat=v,
              ehead={cat=v,vsup=yes,sem={abstract={temp={akt=active}}}
        }},
        frame={arg1=ARG1,arg1={head={ehead={case=nom}}},
               arg2={role=npred,
                     head={ehead={case=acc}},
                     frame={type=static,
                            arg1=ARG1,arg2=ARG3,arg3=nil,arg4=nil,arg5
        =nil}},
                arg3=ARG3,arg3={head={ehead={case=nil,pvalue=dir}}}}}.
        [].
```

with

```
lex={string=with,lex=p,role=funct,pos=pre,min=yes,
        head={cat=p,lex=p,ehead={cat=n,case=acc,pform=with,neg=no,coor
        d=no}},
        head=({ehead={pvalue= tool},
               restr={head={ehead={sem={abstract={temp~=nil}}}}}} ;
              {ehead={pvalue=temp_closed,type=abs,sem={abstract={temp=
        time}}},
               restr={head={ehead={sem={abstract={temp~=nil}}}}}})}.[].
```

in

```
lex={string=in,lex=p,role=funct,pos=pre,min=yes,
        head={cat=p,lex=p,ehead={cat=n,case=acc,pform=in,neg=no,coord=
        no},
             ehead=({pvalue=(space_point;language;material)} ;
```

```
                                {pvalue=temp_simul,sem={abstract={temp=time}}})}}.
        [].
as
lex={string=as,lex=p,role=funct,pos=pre,min=yes,
        head={cat=p,lex=p,ehead={cat=n,case=acc,pform=as,coord=no,pval
        ue=funct}},
but
lex={string=but,lex=coord,role=funct,pos=mid,max=no,min=yes,
        head={cat=coord,lex=coord,coord=yes,
                ehead={index=coord,coord=adversative}},
        frame={arg1={head={coord=no,ehead={cat~=v}}},
                arg2={max=yes,head={coord=no,

          ehead={cat~=n,coord=adversative}}}}}.[].
```

(1) CAT2 사전에 쓰이는 자질 구조

CAT2 사전에 사용되는 자질 구조는 <표면 자질 구조>, <의미격 자질 구조>, <머리 자질 구조>, <하위 범주화 자질 구조>, <피수식어 자질 구조>, <확장 머리 자질 구조>, <어휘 의미 자질 구조>, <범주 유형 자질>, <수식 관계 자질>로 나뉜다.

CAT2 시스템은 사전의 구조가 복잡하기 때문에 이러한 자질 구조를 여기에 모두 설명 하기 어렵다.

3. EDR 자연어 처리용 전자 사전 형식

(1) (1) 일반적 특징

EDR 전자 사전은 다음과 같은 원칙을 바탕으로 개발되었다.

1) EDR 사전에는 표층 구조 차원의 형태소 및 통사적 정보와 심층 구조 차원의 의미적 정보가 각각 분리되어 다루어지고 있다. 즉 개별 언어에 따라 각각 차이가 있는 표층적인 정보는 단어 사전에, 그리

고 의미적 정보는 개념 사전에 각각 저장되어 있다.

2) 단어 사전에는 표제어와 이 표제어가 나타내는 개념을 지시하는 concept identifier간에 대응 관계가 이루어져야 한다.

3) 특수한 문법 규칙이나 알고리즘에 의존적인 정보들은 배제되어야 하며,. 단어들과 이 단어들이 나타내는 개념들에 관한 정보들은 대량의 텍스트에 근거를 두어야 한다.

과 같은 원칙은 많은 양의 의미 정보를 개별 언어들의 사전에 공유하게 하기 위한 것이다. 그리고 2)의 원칙은 개별 언어 사전에 concept identifier 를 연결시키기 위한 매개체로서, 개별 단어 사전으로부터, 개념 사전에 접근하기 위한 방법을 제공하기 위함이다. 또한 이러한 원칙을 바탕으로 한다면, 한 언어의 사전이 다른 언어와는 무관하게 개발되어 질 수 있다는 것이다. 이런 식으로, 개별 언어에 의존적인 표층 정보가 단어 사전에 저장되며, 반면에 개념 사전의 형태로 존재하는 의미 정보는 서로 다른 여러 언어의 사전에 의해 공유되어진다.

공기 정보와 대역어의 대응 관계에 대한 정보는 단어 사전과는 별도로 다루어지며, 독립적인 공기 사전과 대역 사전을 만드는데 사용된다. 3)의 원칙은 개발된 전자 사전이 특정 시스템에 제한되기보다는 범용적으로 사용하기 위함이며, 사전이 앞으로의 응용과 발전을 위해 폭넓은 영역을 마련하기 위함이다. 동시에 이러한 원칙들은 대규모의 사전을 개발하는 동안 정보의 일관성과 정확성을 유지시키는 어려움을 감소하는데 기여한다.

이러한 원칙에 입각하여 만들어진 EDR 전자서전은 다음과 같은 특징을 지닌다.

1) 일상 문장에 사용된 모든 어휘를 cover할 만큼 대규모이다.

2) 특별한 응용 시스템이나 알고리즘을 위해 설계된 것이 아니고, 일반적인 응용 목적을 가지고 개발되었다.

3) 의미 분석에 필요한 지식 베이스를 제공한다.

4) 대량의 텍스트를 근거로 이루어졌기 때문에 상당히 객관성이 높다.

5) 여러 언어나, 전문 분야에 걸쳐 상당히 일반화된 근본적인 내용을 포함한다.

(2) EDR 사전의 구조

단어 사전 -	일반 어휘	일본어 20만 단어
		영어 20만 단어
	기술 용어	일본어 10만 단어
		영어 10만 단어
개념 사전 -	개념 분류 사전	40만 개념
	개념 설명 사전	40만 개념
공기 사전 -	일본어 공기	30만 단어
	영어 공기	30만 단어
대역 사전 -	일-영 사전	30만 단어
	영-일 사전	30만 단어
말뭉치 -	영어 말뭉치	25만 문장
	일어 말뭉치	25만 문장

(3) EDR 전자 사전의 역할

1) 단어 사전의 역할

자연언어 처리시 다음과 같은 정보를 제공한다.

ㄱ) 형태소 분석과 생성시 필요한 정보: 표제어와 형태소의 연결 관계 정보(접속정보)

ㄴ) 통사 구조분석과 생성시 필요한 정보:문법적 정보, 품사, 표층격 정보, 통사 지배 정보

ㄷ) 의미 정보의 제공을 위하여, 단어 사전의 표제어를 개념 사전의 개념과

연결시키는 concept identifier의 정보를 제공한다.

2) 개념 사전의 역할

개념 사전은 문장에 나타나는 의미적 내용이나, 개념들을 전산처리하기 위해 필요한 정보들을 제공한다. 다시 말하면,

ㄱ) 문장에 대한 적절한 의미 표현을 생성하는데 필요한 지식들을 보여준다.

ㄴ) 의미 내용의 유사성(동치성)을 규정하는데 필요한 지식들을 제공한다.

ㄷ) 의미 내용을 유사한(동치의) 내용으로 바꾸는데 필요한 지식들을 포함한다.

3) 공기 사전의 역할

공기 사전은 어떻게 단어들이 사용되는지에 대한 정보를 제공한다. 이러한 정보는 어떤 상황을 나타내기 위해 문장을 이루는 단어 결합의 유형으로 이루어진다. 즉 주어진 단어가 다른 단어와 함께 나타나는 관계와, 사용되어지는 단어의 유형을 포함하고 있다. 이러한 정보는 다음과 같은 경우에 유익하다. 즉, 어떤 언어의 특정 단어는 다른 언어에서는 여러 가지로 번역될 수 있다. 그러나 그 단어가 일정한 단어들과 어울려 나타나는 공기 정보를 알고 있다면, 이에 해당하는 언어를 다른 언어에서 찾는 것은 용이하다.

4) 대역 사전의 역할

대역 사전은 서로 다른 언어의 단어들간의 대응 관계를 보여준다. 이 사전은 자연언어 처리하는데 사용되어지도록 디자인되어졌고, 다음과 같은 3가지 기능을 제공한다.

ㄱ) 대응어는 전산처리를 위해 적절한 단어들이 주어졌는지가 고려되었다.

ㄴ) 표제어와 타언어의 해당 단어간의 대응 관계를 명확하게 하기 위해, 설명이 동의어, 상위어, 그리고 하위어의 관계에 의해 이루어졌다.

ㄷ) 이 사전은 표제어와 대응어의 관계를 보충하는데 필요한 정보도 유지한다.

5) 말뭉치의 역할

EDR 말뭉치는 선별된 문장들, 구성 성분 형태소 정보, 통사 구조(트리)와 개념 관계 표현의 정보들로 이루어졌다. 개념 사전과 공기 사전의 data는

EDR말뭉치의 통사 구조와 개념 관계 표현으로부터 추출된 것이다. EDR 말뭉치는 EDR 전자 사전의 개발에만 제한된 것이 아니고, 자연언어처리의 연구 분야에서 다양하게 응용할 수 있다.

EDR말뭉치는 다음과 같은 과정을 통해 완성되었다.

ㄱ) 텍스트 수집: 영어와 일어 각각 20만 문장, 신문, 백과사전, 교육용 텍스트, 참고문헌

ㄴ) KWIC data 완성

ㄷ) 문장 선택: keyword로 단어 사전에 기록된 어휘를 사용하여 선택.

ㄹ) 문장 분석: 문장의 통사적, 의미적 구조를 생성한다.

(4) 단어 사전의 유형

단어 사전: 일반 사전(General Dictionary) 과 기술 용어 사전(Technical Terminology Dictionary)으로 이루어졌다.

ㄱ) 일반 사전: 일상생활에서 쓰이는 단어, 일상 쓰이는 기술 용어 (technical Term), 고유명사, 약자, 숙어 및 숙어적 표현.

ㄴ) 기술 용어 사전: 특수 분야의 정보처리에 사용되는 단어 및 숙어

1) 일본어 단어 사전의 마이크로 구조

Headword information -	Headword	- 표제어
	Constituent(s)	- 구성 성분
	Notation	- 표기
	Adjacency attributes	- 접속정보
	Kana notation	- 가나 표기
	Pronunciation	- 발음
Grammatical Information -	Part of speech	- 품사
	Syntactic tree	- 구문 트리
	Conjugation	- 활용
	Surface cases	- 표층격
	Aspect information	- 양태
	Function word information	- 기능

어정보

Semantic Information -	Concept identifier	- 개념
	Concept illustration	- 개념 설명
Supplementary Information -	Usage	- 용례
	Frequency	- 빈도수

2) 영어 단어 사전의 마이크로 구조

Headword information -	Headword	- 표제어
	Constituent(s)	- 구성 성분
	Notation	- 표기
	Adjacency attributes	- 접속정보
	Syllable division	- 음절 표시
	Pronunciation	- 발음
Grammatical Information -	Part of speech	- 품사
	Syntactic tree	- 구문 트리
	Inflection	- 활용
	Grammatical attributes	- 문법적 정

보

	Function word information	- 기능

어정보

Semantic Information -	Concept identifier	- 개념
	Concept illustration	- 개념 설명
Supplementary Information -	Usage	- 용례
	Frequency	- 빈도수

(3) 마이크로 구조에 나타난 정보의 내용

ㄱ) Headword Information

- Headwords는 형태소 분석을 위해 문장에 나타나는 단어를 확인하기 위해 사전검색 (Dictionary look-up) 을 할 때 사용된다. 문장 생성에서는 생성 형태소에 의한 출력 문장을 형성하

는 데에도 사용된다.

- 표기는 실제 문장에 쓰이는 단어의 스트링이다.
- 접속 정보는 인접어나 문장에서의 구성 성분에 대해 형태소적 제한 정보를 제공한다.
- 가나식 표기법과 발음은 개별 단어에만 주어진다.

ㄴ) Grammatical Information

- 이곳의 정보는 통사 분석에서 문장의 통사 구조를 찾아내는데 사용되거나, 통사 형태의 생성시에 문장 구조를 규정해주는데 사용
- 하나 이상의 단어로 이루어진 표제어에 대해서는 통사 구조(syntactic tree)가 제공된다.
- 술어(동사, 형용사, 형용사적 명사 등)와 조동사에 대해서는 활용형의 정보가 제공된다.
- 표층격의 정보는 술어에 제공된다.
- 양태정보는 동사에 제공된다.
- 기능어 정보는 불변화사, 조동사, 형식명사, 수사와 접속사에 제공된다.

ㄷ) Semantic Information

- 개념은 단어의 의미적 중의성을 구별해주는데 사용되는 정보이다.

  개념은 language independent 하다.
- 이곳에 개념 사전의 가장 중요한 요소인 개념과의 link를 제공한다.
- 개념은 Concept identifier와 Concept illustration으로 표현된다.

  Concept Identifier 는 개념을 지시하는 수치적 표시이다.

ㄹ) Supplementary Information

- Usage: 어떤 단어는 특수한 상황에서, 특정한 방법으로 사용되는 경향이 있다. 그러한 단어를 포함하는 문장을 작업할 때, 주요 key 가 된다.
- Frequency: EDR 데이타 베이스에 나타나는 단어의 빈도수를 보여준다.

3절. 기존의 사전 관리 시스템

사전은 저장 구조가 일반 텍스트와 달라서 사전을 관리하기 위한 별도의 프로그램이 있다. 간단한 사전의 경우라면 텍스트 에디터를 통하여 사전을 입력하거나 수정하고 이것을 필요한 구조로 바꾸어 저장하는 도구(예를 들면 사전 컴파일러)만 있으면 기본 적인 관리가 가능하다.

그러나 사전의 구조가 복잡하면 복잡할수록 사전 관리 기능의 요구 사항이 많아지게 되고 상당히 복잡한 구조의 사전 관리기가 필요하게 된다. 이 절에서는 일반 사전과 자연언어 처리용 사전을 관리하는 기존의 사전 관리 시스템을 간단히 소개한다.

1. 일반 사전 관리 시스템

(1) 국립 국어 연구원의 사전 편찬 지원 시스템

일반 사전 관리 시스템 중 잘 알려져 있는 것은 최근에 개발된(현재 개발 진행중) 국립 국어 연구원의 사전 편찬 지원 시스템이 있다. 이 시스템의 목적은 기존의 사전을 포함하는 큰 규모의 새로운 사전을 편찬하는 것을 지원하는 데 있다. 내부적으로는 데이터베이스를 사용하고 있으며 검색과 편집 기능이 뛰어나다. 다음은 사전 편찬 지원 시스템을 간략히 설명한 것이다.

1. 사전 편찬 지원 시스템의 목적

   국립 국어 연구원의 사전 편찬에 활용

   각 어휘들의 사용 실태 조사 연구에 관한 자료의 정리 및 통계 처리

   방대한 양의 자료를 효율적으로 관리

자료집 발간

축적된 자료들을 효율적으로 관리

컴퓨터 통신망을 이용한 자료 서비스

2. 사전 편찬 지원 시스템의 구성

사전 데이터베이스 시스템

문헌 데이터베이스 시스템

도서 관리

유틸리티

코드 체계 및 폰트

검색 및 소트

자료 변환

어휘 분석

전자 게시판 전자 우편함

에뮬레이터

기타

3. 사전의 구조

1) 표제어 테이블

표제어 번호, 표제어, 어깨번호, 표제어IC, 표제어 형태, 구표제어번호

2) 부표제어 테이블

표제어번호, 부표제어번호

3) 표제어 상태 테이블

표제어 번호, 어원상태, 원집필자, 발음상태, 발음집필자, 뜻풀이상태, 뜻풀이집필자, 관용구상태, 관용구집필자, 속담상태, 속담집필자(상태=완료/보류)

4) 작업자 현황 테이블

표제어번호, 원고입력자, 원고상태, 교열자1, 교열자2, 교열자3, 교열상태, 교정자1, 교정자2, 교정자3, 교정상태

4-1) 관련어목록 테이블

표제어번호, 표제어, 지정, 참조

5) 원어 테이블

사전, 원어코드값, 표제어번호, 원어내용

6) 발음 테이블

사전, 발음코드값, 표제어번호, 발음내용

7) 활용정보 테이블

사전, 활용정보코드값, 표제어번호, 활용정보내용

8) 어원 테이블

사전, 어원코드값, 표제어번호, 어원내용

9) 뜻풀이 테이블

사전, 뜻풀이코드값, 표제어번호, 뜻풀이내용

10) 관용구 테이블

사전, 관용구코드값, 표제어번호, 관용구내용

11) 관용구 풀이 테이블

사전, 관용구풀이코드값, 표제어번호, 관용구풀이내용

12) 속담 테이블

사전, 속담코드값, 표제어번호, 속담내용

13) 속담풀이 테이블

사전, 속담풀이코드값, 표제어번호, 속담풀이내용

* 품사 코드표

명사	N1
수사	N2
대명사	N3
동사	V1
형용사	V2
관형사	D1
부사	D2
감탄사	A
조사	E1
어미	E2
서술격조사	I
접사	S

** 그 외의 코드표

어간의 기본형 제시	-
분석기호	]
서술격조사가 생략된 경우	%
문장부호	M1
문장끝	$
아라비아숫자앞	#1
영문자앞	#2
의미가 있는 단위로 사용되는 기호	#3
한자를 한글로	()
동음이의어	01,02,...
고유명사(인명)	*1
고유명사(지명)	*2
고유명사(기타)	*3
오자, 문장오류	@
미등재어	#
활용접사가 모음만 남는 경우	₩
하나의 단위로 취급해야 하는 것들	+

4. 사전 제작 과정

1) 6개 사전에 대한 표제어 카드를 만든다.

2) 표제어 목록을 입력하여 데이터베이스에 저장한다.

3) 미등제 표제어를 확인하고 새 사전의 표제어를 확정한다.

4) 관련된 데이터(기존의 사전, 용례 데이터 등)을 이용하여 새 사전의 표제어 항목을 완성해 나간다. (뜻풀이, 발음, 용례, 관용구, 속담 등..)

5. 특징

고어와 한자 등을 수용하기 위한 4byte 코드와 폰트 사용

윈도우 인터페이스 환경

사전의 모든 필드를 미리 정하여 사용

다양한 검색 기능 제공
통계 정보 제공
여러 사람이 동시에 사용 가능

6. 장단점

사전 편찬 전용 시스템으로서 가치가 있다.
데이터베이스에서 작업을 하므로 빠르고 견고하다.
새로운 필드의 추가시 프로그램이 바뀌어야 한다.
다른 시스템(자연언어 처리용)으로 데이터를 옮기기 어렵다.

이 사전 편찬 지원 시스템은 특별한 목적을 가지고 설계된 것이어서 검색과 편집에 있어서는 다양한 방법을 제공한다. 그러나 새로운 형식의 사전을 만들기 위해서는 프로그램을 고치는 노력이 필요하다. 또한 다른 종류의 사전을 병합하는 방법이 마련되어 있지 않다.

2. 자연언어 처리용 사전 관리 시스템

(1) 형태소 해석기용 사전 관리기

형태소 해석기용 사전 관리기는 사전의 저장 구조와 밀접한 관련이 있다. 일반적으로 형태소 해석기에 사용되는 사전은 dbm file(hashing), btree, trie 등을 사용하고 있으며, trie 사전은 on-line update 가 어렵다. 형태소 해석용 사전의 내용으로는 (단어: 품사 1, 품사 2, ...) 의 간단한 형태이다.

1) 포항공대의 형태소 해석용 사전 관리기

Insert, Delete, Update의 간단한 기능
Debugging 기능

2) 한국과학기술원의 형태소 해석용 사전 관리기

dbm file형식
batch mode
사전에서 text로 저장하는 기능

text 사전으로 읽어들이는 기능
text에 있는 내용을 사전에서 지우는 기능
interactive mode(insert, delete, search)

이러한 사전 관리기는 형태소 해석기의 성능 향상을 위한 간단한 사전 갱신과 DBMS를 사용할 수 없는 경우에 유용하다.

(2) 기계번역용 사전 관리기 -- MATES/EK

기계번역에 사용되는 사전의 종류는 형태소 해석 사전, 구문 해석 사전, 의미 해석 사전, 대역 사전, 생성 사전 등과 별도의 정보(연어 정보 또는 공기 정보 등)를 가지는 사전으로 나누어 볼 수 있다. 그러나 기계번역의 방법에 따라 필요한 사전의 종류와 수는 약간씩 차이가 있다.

MATES/EK 사전 관리기는 일반어 해석 사전 11 개(품사별), 일반어 대역사전 4 개, 전문어 해석 사전 4 개, 전문어 대역사전 4 개, 총 23 개 파일을 관리한다.

일반어-명사-해석 : 영어 사전
일반어-동사-해석
일반어-형용사-해석
일반어-부사-해석
일반어-접속사-해석
일반어-한정사-해석
일반어-전치사-해석
일반어-조동사-해석
일반어-대명사-해석
일반어-BE-동사-해석
일반어-BE-조동사-해석
전문어-명사-해석
전문어-동사-해석
전문어-형용사-해석
전문어-부사-해석

일반어-명사-대역 : 영한 사전

일반어-동사-대역

일반어-형용사-대역

일반어-부사-대역

전문어-명사-대역

전문어-동사-대역

전문어-형용사-대역

전문어-부사-대역

MATES/EK 사전 관리기의 기능

도움말

online 환경 - menubar, pulldown menu

key search

editing 취소 기능

batch 처리

다음은 MATES/EK 사전의 내용을 볼수 있는 Work Sheet Form 이다.

영어 동사 사전 WSF - i (Verb)

Lexical_unit			Usage_ID_No.	
Multi_Lex				
LDOCE_UID				
Inflection (code) (irregular)	0(none)/1(- -> s,d,ing)/2(- -> s,ed,ing)/ 3(- ->es,ed,ing)/4(e -> es,es,ing)/5(y -> ies,ied,ying) 6(ie -> ies,ied,ying)/7(- -> s,복자음+ed,복자음+ing)/ 8(- -> es, 복자음+ed, 복자음+ing)/9(irregular)			
	3^{rd}_sing		Past	
	Past_Part		Pres_Part	
Multi_POS	[N, ADJ, ADV, DET, PRO, AUX, CON, PRE]			
LDOCE_Type	[* LDOCE 코드 참조]			
Verb_Types				
Situation_type	[* Verb Situation Table 참조]			

Voice	[A(active only) / P(passive only)/ B(both)]					
Modality						
Subj_D_verb	Passive | [D(direct)/I(indirect)/B(both)]					
AGT of to-inf	[N1 (Subject) /N2(Object)/B(both)					
Case Frame	Suf	심층격	Co-occur LEX	Syn_Form	Semantic Maker	OBG

사전 데이타의 on-line 처리는 사전 EDITOR 에 의하여 실행되며, batch 처리는 기본적 사전 접근 라이브러리(Volume Raw Access Library)를 이용한 사전 재구성 유틸리티로 이루어 지고, 가상 접근 유틸리티는 on-line 및 batch 처리 모두 가능하다.

(3) 가상 접근 유틸리티

예를 들면, 사전을 형태소 사전, 구문 정보 사전, 의미 정보 사전으로 분리할 수도 있고, 하나의 레코드로 관리할 수도 있다. 하나의 레코드로 형태소, 구문, 의미 정보를 모두 관리하는 것은 사전의 변경을 어렵게 한다. 가상 접근 유틸리티는 여러 개의 Volume 으로 이루어진 이러한 레코드를 하나의 레코드에 접근하는 것처럼 사용할 수 있게 한다.

사전 재구성 유틸리티는 하나의 Volume 을 둘로 나누거나 여러 개의 Volume 을 하나로 병합하는 방법을 제공하며 사용자는 여러 개의 Volume 들 간의 link 를 임의로 정의하여 사용할 수 있다.

MATES/EK 에서는 사전을 품사별로 관리하므로 다품사어에 대한 특별한 색인이 필요하다. 각 품사별 사전은 Private Index 를 가지고 있고 전체를 통합하여 Master Index 를 가지고 있다. 우선 Master Index 를 찾아 link 를 따라가면 Private Index 에 도달 할 수 있다.

MATES/EK 사전 관리기의 특성

on-line/batch 처리에서 편리하고 다양한 기능의 제공

품사별 사전 분리와 다품사어에 대한 통합 색인 기능

사전의 정보의 종류에 따라 분할 통합하는 기능과 사용자에 의한

볼륨간 link의 정의
가상 접근으로 여러 개의 사전을 하나의 사전처럼 관리
편집 화면을 사용자가 재구성 가능

MATES/EK 사전 관리기는 기계번역 시스템을 위한 것이었기 때문에 기계번역 시스템(특히 MATES/EK)에 촛점이 맞추어 졌다. 따라서 다른 시스템에 응용하려는 적극적인 시도가 없었다. MATES/EK 사전 관리기는 몇 가지 문제를 보완한다면 자연언어 처리의 여러 응용에 사용될 수 있을 것이다.

그러나 MATES/EK 사전 관리기가 범용성을 가진 사전 관리기로 사용되기에는 문제가 있다. 그 이유는 MATES/EK 사전 관리기가 Tree 구조의 사전을 관리 하기에 적합하지 않으며 사전 검색 기능이 미약하다. 여러 가지 용도의 사전을 통합하고 관리 하기 위해서는 상당히 많은 사전 관리 개념들이 수정되어야 할 것이다.

3장. 기존의 사전 형식 표준화 동향

만일 언어적 data를 컴퓨터로 처리해야 한다면, 이 언어적 data를 컴퓨터가 처리할 수 있는 기호 표시의 체계로 바꾸어야 한다. 컴퓨터가 처리할 수 있는 기호는, 언어적 data에 비해 적은 수이므로, 자연언어처리의 초기 시절부터 특별한 mark-up 체계(system)을 개발하려고 노력하였다. 즉 이 mark-up 체계를 통해서 제한된 기호를 가지고, 복잡한 과제를 설명할 수 있도록 하기 위함이었다. 이러한 표시 체계의 개발과 더불어 항시 문제시된 것이 바로 표준화의 문제이다. 그래서 자연언어처리의 발달과 함께 표준화된 코드 체계를 만들기 위해 노력하였고, 80년대 초에 SGML-Standard (Standardized Generalized Markup Language)가 생겨난 것이다.

먼저 자연언어처리를 위한 표준화된 코드 체계의 발전을 위한 노력과 시도에 대해 살펴보고, 또한 SGML에 기반을 두고있는 TEI(Textual Encoding Initiative) 시스템에 대해 살펴본다.

1 절. SGML(Standard Generalized Markup Language)

SGML 은 약 1980 년부터 IBM-Standard 인 GML 을 바탕으로, Charles F. Goldfarb 와 ISO (International Organization for Standardization)의 한 작업 그룹에 의해 개발되었다. SGML 의 원칙은 내용상 다른 여러 텍스트 부분들이 지시자나 태그를 통하여 표시되어 질 수 있다는 것이다. 내용상 다른 텍스트 부분들이란 예를 들면, 제목, 단락, 언어학적 설명, 인용, 주, 도표, 수식, 특수 fonts 등과 같은 것을 말한다. 그밖에도 텍스트 밖에 놓여있는 대상을 언급하는 지시자도 정의 되어질 수 있다.

태그는 기타 텍스트와 구분되도록 '<' 과 '>'에 의해 정의된 임의의 string 을 말한다. 표시의 끝은 '</>'으로 나타낸다. 텍스트의 표시를 위해 사용된 태그들은 document 의 특별한 부분에 DTD (Document Type Definition)라는 이름으로 정의된다. DTD 는 document 에 사용된 모든 태그들과 경우에 따라서는 그들의 계층구조적인 관계를 포함한다. 다음은 문서의 logical structure 를 정의하는 DTD 의 한 예이다.

```
<!DOCTYPE memo          [
<!ELEMENT memo          - 0     (from, to, subject, contents)       >
<!ELEMENT from          0 0     (person)+                           >
<!ELEMENT to            0 0     (person)+                           >
<!ELEMENT person        0 0     (nickname | (forname?, surname) )   >
<!ELEMENT nickname      - -     (#PCDATA)                           >
<!ELEMENT forname       - -     (#PCDATA)                           >
<!ELEMENT surname       - -     (#PCDATA)                           >
<!ELEMENT subject       0 0     (#PCDATA)                           >
<!ELEMENT contents      0 0     (#PCDATA)                           >
]>
```

위의 DTD 를 바탕으로 임의의 텍스트(여기서는 memo)를 SGML 형태로 표현할 수 있다. 다음의 예는 이 DTD 를 바탕으로 작성된 memo 의 한 예이다.

```
<!DOCTYPE memo PUBLIC >
<memo>
<from> Brian
<to> Martin
<subject> Bord Meeting
<contents> Unfortunately  I have had to reschedule the board meeting
for 10:00 on Tuesday. I hope this does not inconvenience you too much.
</memo>
```

SGML 은 일반적인 틀로서 여러 가지로 사용될 수 있어서, 현재는 수많은 양의 data 를 표준화하여 코드화하는데 사용되고 있다. 예를 들면, Oxford Advanced Learner's Dictionary of Current English 와 CD-ROM 에 기계 가독형태로 저장된 Oxford English Dictionary 가 바로 이러한 SGML 형태로 구조를 이루고 있다. 또한 British National Corpus 와 같은 대량의 코퍼스도 SGML 을 바탕으로 하는 코드체계를 이루고 있다.

2 절. Text Encoding Initiative

자연언어 data 의 기계적 처리를 위해서 사용된 SGML 의 특별한 형태가 1987 년에 Text Encoding Initiative (TEI)라는 것을 발전시켰다. data 의 양이 점점 많아지고, 이에 따른 정보의 검색도 필수적이 되어가자, 기존의 텍스트나 텍스트 코퍼스, 그리고 사전과 같이 표준화되어 있지 않은 다양한 형태의 양식들이 표준화되어져야 할 필요성이 생겨났다.

이에 부응하여 TEI 는 SGML 의 일반적인 특징 이외에

1) 기계가독형 텍스트를 위한 보편적인 교환 포맷을 세분화해야 한다
2) 새로운 텍스트의 코딩을 위하여, 어떠한 종류의 feature들이 텍스트를 위해 사용되어야 하며, 또한 이들이 어떻게 코딩되어야 하는지의 방법도 제시해야 한다.
3) 기존의 아주 중요한 코딩 원칙들을 문서화하고, 이를 설명하기 위한 메

타 언어를 개발한다. (cf. Hockey 1992)

TEI 의 코딩스키마는 SGML 의 특별한 응용형태이기 때문에, TEI 의 원칙에 입각해서 코드화된 문서들은 SGML 과는 완전히 compatible 하여, SGML 필터와 editor 로 작업이 가능하다. SGML 에 대한 TEI 의 특징은, 텍스트의 작업에 있어서 어떤 자질들과 정보가 고려되어져야 할지, 그리고 이것들이 어떻게 코딩되어야할 지가 제시된다는 것이다. 따라서 TEI 는 자연언어처리시에 나타나는 텍스트 specification 의 가능성을 제시하고, 이를 제시된 형태에 따라 나타낼 수 있다.

TEI 에서는 모든 텍스트 문서가 우선 크게 header 와 body 로 이루어진다. header 는 텍스트에 대한 일반적인 정보, 특히, 저자나 제목, 날짜, 출판 장소 그리고 텍스트의 종류에 대한 정보를 포함하고 있다. 그밖에 코딩한 사람이나 문서가 관리되는 기관의 이름도 여기에 포함된다. (사용 조건 및 텍스트 작업 날짜와 프로그램의 이름 등도) 텍스트의 body 는 개별정보 요소들로 이루어지는데, 이러한 정보요소들의 코딩은 TEI 가 제시한다. 이러한 제시들은 어떻게 특정 문서의 data 구조가 SGML 로 표현될 수 있는지, 그리고 어떤 종류의 feature 집합들이 언어학적으로 변별적인 개개의 data 종류에 사용돼야 할 지를 보여준다.

header 와 body 라는 표지는 SGML 의 규정에 따라, Document Type Definition (DTD)에 선언되어져야 한다.

TEI 에서는 운문, 산문, 드라마, 구어체 코퍼스, 일반사전, 그리고 용어 data base 등과 같이 다양한 문서들에 대해 기본 tag set 을 제시하고 있다.

예) 66 - 74: 김성혁 교수: 인코딩 포맷에 관한 연구

1. 산문을 위한 base tag set
   - default text structure, core tag set 사용

2. 운문, 시를 위한 base tag set
   - <lg> ; core tag set으로 type 속성을 포함하고 와 유사

<caesura> ; 시행의 segmentation을 위하여 사용

- <lg1>, <lg2>

<seg> 정렬과 분절을 위하여 사용

한 행의 하부 구성요소를 구별하기 위하여 사용

- 추가 속성

met ; 관습적인 운율구조를 표현

real ; 운율구조가 실현될 때 나타나는 현상

rhyme ; 시행의 그룹에 적용하는 운

이러한 속성들의 값은 이용자 정의.

3. 드라마를 위한 base tag set

- 영화 대본, 라디오 스크립트, 공연 출판물 등

3.1 front, back

- <performance> ; 연극무대나 스크린에 상영될 때 그 상영정보들을 그룹화

<prologue>

<epilogue>

<set> front에 나타나는 setting, 시간 등을 기술

<castList> 단일 출연자 리스트

<castGroup> cast list에 <castItem>이 나타날 때 사용

<castItem> 배역이나 출연진에 관계된 정보

<role> 배역명

<roleDesc> 배역의 역할 기술

<actor>

3.2 body

- 나 <div0>, <div1> 등을 사용
- <sp> 각각의 speech를 태깅. 일반적으로 산문이나 운문의 형태

<speaker> heading이나 label의 특수한 형태로 speaker의 이름을 제공

- 연극대본은 다면적 구조

  예 ; 시가 낭독된다면, 시 자체와 낭독을 분리하여 수록해야 함.

- 무대 지시 사항

  <stage>

  <move> 등장인물들의 등장과 퇴장을 태깅

  who

  type ; entrance, exit, onstage

  where ; L, R, C, perf

- speech ; 산문이나 운문의 형태

  <p> <lb> 새로운 라인의 시작

  <l> <lg>

- 동시 동작 ; 여러 사람이 동시에 대사나 동작을 하는 경우.

  corresp 속성 사용

  <stage id=D1 corresp="S1 S2 S3 S4">

- 다른 기술정보

  <view> 관찰자의 입장에서 스크린내 시각적 내용을 묘사

  <camera> camear angle이나 view point

  <caption> 자막

  <sound> 음향효과

4. Transcription of Speech

- 언어적, 음향적 대상물이며 시간과 밀접한 관계가 있으므로 내부적 구조의 시작과 종결을 구분짓기 어렵다.

- spoken text

  공식 ; 강연, 회의 등

  비공식 ; 어떤 집단의 구성원들간의 대화

- 구성요소

말, 발화
구절, 단락
음성적 이지만 어휘적이지는 않은 현상
(예; 기침)
동작
비 언어적 현상 (사건)
강연 도중의 필기 사항
음성의 변화

- 사용하여 태깅
- spoken text에서 특징적인 요소
  <u> 침묵, 연사가 바뀌기 전이나 후와 같은 speech의 연장
  <pause> 발화 내에서의 휴지
  <vocal> 음성적 현상
  <kinesic> 음성적이지 않은 의사소통 수단 (몸짓 등)
  <event> 어떤 현상이나 사건(잡음 등으로 인하여 대화에 영향을 주는 요소)
  <writing> speech 중간에 나타나는 필기 text
  <shift> 연사가 바뀜으로 발생하는 일련의 발화에 대한 준 언어학적 요소

5. 인쇄된 사전
   - 사전 엔트리의 구조가 사전들이나 사전 내에서 매우 다양하게 나타남. 따라서 모든 사전에 공통적인 구조를 찾아내어 인코딩에 적용.
   - <entry> 모든 사전에 공통적인 구조를 표현
     <entryFree> 동일 요소이나 융통성을 갖는다.
   - 사전에 수록된 정보는 함축적이고 요약적이다. 따라서 사전 원본의 정확한 인쇄 형태와 정보가 표현된 기반 구조 표현이 가능해

야 한다.

5.1 전체적 구조

<text> <front> <body> <back>

<div0> <div1> 양국어 사전에서 사용

<entry> <entryFree>

; trpe - main, hom, xref, affix, abbr, supplemental, foreign

; key - 엔트리의 알파벳 순서를 반영하는 문자 포함. sort key

5.2 계층 레벨

<entry> <entryFree>

<hom> ; <entry> 내에서 하나의 동형이의어와 관련있는 정보를 그룹화

<sense> ; 한 단어의 의미와 관계있는 모든 정보들을 그룹화

5.3 그룹과 구성요소

- 사전 엔트리의 최상위 구성요소

단어의 형태에 관한 정보 (발음, 정자법 등)

문법정보

정의, 외국어로의 번역

어원

예

사용법

상관참조

주기

관련어를 위한 엔트리

- <form> 표제어의 발음 형태와 필기 형태에 대한 모든 정보가 그룹

화

<gramGrp> 형태, 통사 정보를 그룹화. 예를 들어 <pos>, <gen>, <number> 등

<def> 정의문

<trans> 다언어 사전에서 한 엔트리 내 관련 정보의 번역

<eg> example text

<usg> usage information

<xr> text 내에서 다른 위치를 지정하는 구문, 문장, icon

<etym> 어원정보

<re> 표제어와 관련된 어휘정보를 위한 엔트리

<note>

4장. TDMS 시스템 구조

TDMS 시스템은 크게 3 개 요소로 구성된다. 표준 사전 표기 언어(SDML)로 그 사전 형식이 정의되어진 표준 사전(SD), 표준 사전을 생성, 편집, 검색 등을 할 수 있는 사전편집기(SDE)와, 표준 사전이 아닌 기존의 사전들을 표준 포맷으로 바꾸어 주고 또, 표준 포맷을 원하는 특수 포맷으로 바꾸어 주는 변환 프로그램(SD Encoder/SD Decoder)들로 구성된다.

1 절. 표준사전(SD)

SD 는 SDML(Standard Dictionary Markup Language)로 작성된 사전이며, 각 사전마다, SDML 이 허용하는 범위 내에서 다양한 형태를 가질 수 있다. SD 는 SDML 에서 정의된 구조로 만들어 지고, 각 구조에는 SDML 에 정의된 태그가 부착된다. 사전의 각 내용들은 모두 텍스트 형식으로 표시되며, 따라서 특별한 프로그램 없이도 바로 내용을 확인해 볼 수 있다. SD 의 앞 부분에는 사전의 구조를 나타내는 부분이 있어서 그 앞부분의 파싱을 통해 뒤에 오는 사전의 형태를 올바르게 파악할 수 있다.

이런 사전의 구조는 사실상 SGML 의 문서형태 정의(DTD: Document Type

Definition)로 기술된 것이다. 따라서 각각의 사전마다, SDML 규칙에 근거한 사전 구조를 나타내는 SGML DTD 의 일부가 포함되어 있으며, 이러한 각각의 형식을 표준사전 형식(SDF: Standard Dictionary Format)으로 부른다.

2 절. 사전편집기(SDE)

SDE 는 SD 사전의 내용 및 구조를 편집하고 수정할 뿐만 아니라, 내부의 내용을 검색 또는 브라우징(browsing) 할 수 있다. SDML 로 만들어진 SDF 를 해석하여 SDE 의 내부 구조를 정할 수 있는 기능, 기존 SDF 사전을 인식하여 하나의 내부 구조로 변형시키는 import 기능, 반대로 내부구조를 선형형식(linear form)으로 바꾸어 표준 사전오로 만들어 내는 export 기능을 지원한다.

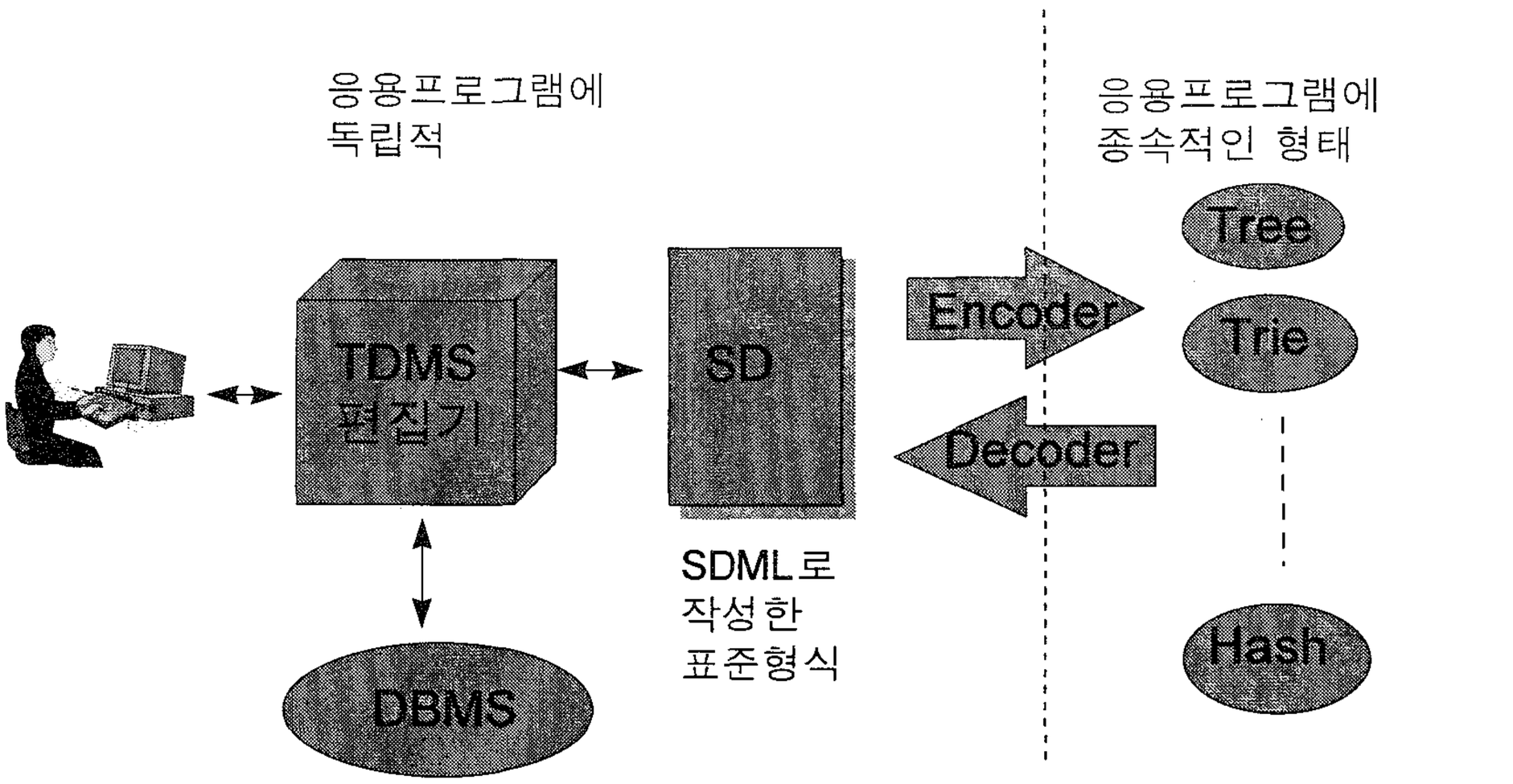
사전의 내용 변경과 구조 변경을 지원하기 위해 SDE 는 2 가지 모드로 된다.

1. 데이터 편집 모드: 사전 보기 및 사전 입력을 하기 위한 모드이다. 새로운 엔트리의 추가/삭제나, 각 필드의 내용 추가/삭제/변경을 할 수 있으며, Browsing, 검색 등을 할 수 있다.
2. 구조 편집 모드: 사전의 구조 변경을 할 수 있는 모드이다. 즉 기본이 되는 SDF를 변경시켜 새로운 구조를 만들 수 있다. 사전의 데이터를 새 구조로 연결하여 필요한 부분을 복사함으로써 새로운 형식으로 변형시킬수 있다. 또한 이미 존재하는 다른 구조의 사전에 필요한 부분만을 복사하여 사전을 확장할 수도 있다.

3 절. 사전 변환 프로그램(SD Encoder/SD Decoder)

기존의 사전 형태는 사용되는 곳의 필요에 따라 여러 가지 구조의 이진화일, 또는 텍스트화일 등으로 저장되어 있다. 이러한 형태의 사전을 SD 형태로 바꾸어 주는 것이 SD Decoder 이며, 반대로 필요한 응용 프로그램에 맞도록 SD data 를 변형시켜 주는 프로그램이 SD Encoder 이다.

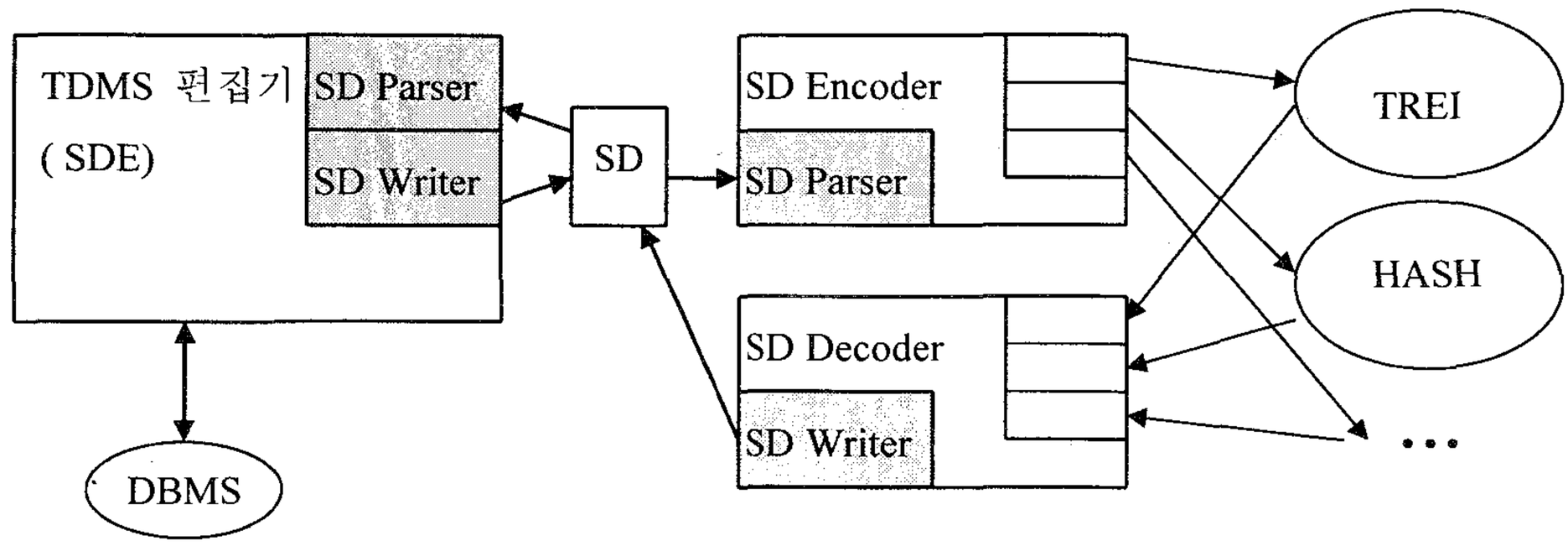
이러한 사전 형태는 너무 다양하기 때문에 이 프로젝트에서는 주로 많이 쓰이는 몇 가지 형태의 사전만을 표준으로 지원해 준다. 지원되지 않는 형태는 각자 SD Decoder 를 개발하여 SDF 형태로 변형시켜주면, SDE 를 사용할 수 있다.



[그림 1] TDMS 시스템 구조

또한 SD decoder 와 SD encoder 개발을 쉽게 할 수 있도록 기본 형태의 모듈을 제공하여 개발자가 그 사전에 특수한 기능만을 추하면 SD decoder/encoder 가 완성될 수 있도록 한다.

1. 기본 구성 요소



[그림 2] TDMS의 기본 구성 요소

Encoder : SD형식의 사전을 특별한 응용 시스템의 사전 구조로 변환하는 도구

Decoder : 다른 응용 사전의 내용을 SD형식으로 변환하는 도구

Encode 의 구성

SD사전을 읽어 SD tree로 구성하는 도구(SD 영역)

SD tree에서 필요한 부분을 추출하는 도구(SD/응용 사전 개발자 공동 영역)

응용 사전을 구축하는 도구(응용 사전 개발자 영역)

Decoder 의 구성

응용 사전 listing 도구(응용 사전 관리자/개발자 영역)

응용 사전 list를 SD tree로 구조 변경하는 도구(공동 영역)

SD tree를 SD file로 write하는 도구(SD 영역)

2. 공통 모듈

Decoder/Encoder공통 모듈

SD를 표현 할 수 있는 general tree structure

Decoder 공통 모듈

SD tree로부터 SD file을 생성하는 SD writer

Encoder 공통 모듈

SD를 읽어 SD tree로 만들어 주는 SD parser

개별 모듈

응용 사전 listing도구

응용 사전 구축 도구

3. Trie 사전 Encoding/Decoding 예

Trie 사전 Encoding 단계

SD 사전을 parsing하여 SD tree를 만든다.

SD tree를 따라 가면서 모든 단어에 대하여

각 단어에 대하여 필요한 정보만을 추출한다.

추출된 단어:정보를 Trie사전에 추가한다.

Trie 사전 Decoding 단계

Trie 사전의 모든 단어에 대하여

단어:정보를 SD tree로 구성한다.

SD tree를 SD file로 저장한다.(필요하다면 SD사전 head정보와 form정보를 생성하여 출력한다.)

Trie사전에 Text입출력 기능이 가능하다면

Encoder는

SD tree에서 Trie사전에 필요한 형태의 Text로 출력

Decoder는

Text를 읽어 SD tree를 구성한다.

5장. 표준사전 형식

1절. SDML 정의

일반적인 사전은 표제어를 인덱스로 하여 엔트리가 반복되어 나타난다. 이 엔트리는 표제어의 성격에 따라서 엔트리 내부의 구조와 내용들이 변한다. 또, 엔트리 내의 필드에서 특정한 부분을 참조하기도 한다. 자연언어 처리용 프로그램에서도 이러한 기본 형식이 유지된다. SDML 은 사전 형태를 표현하는데 필요한 SGML 의 일부 기능만을 제한적으로 사용하도록 한 것이다.

SDML 의 구조

SDML 은 크게 헤더, 정의 부분, 엔트리 그룹으로 구성되어 있다. 헤더는 이 사전의 이름, 버전 등의 내용을 포함하고 있고, 정의 부분(front)에서는 사전 내부에서 쓸 각 속성들이 가질 수 있는 값을 규정하고 있다. 엔트리 그룹은 실제 사전의 내용이 포함되어 있으며, 표제어를 포함하는 각 엔트리들의 반복으로 구성된다.

```
<sd>
        <sdHeader> [헤더 정보] </sdHeader>
        <front> [ SD 내에서 쓰일 attribute 및 value 정의] </front>
        <group>
                <entry>
                        <wname> [표제어] </wname>
                        <body> [표준 사전 요소] </body>
                </entry>
                <entry>
                        <wname> [표제어] </wname>
                        <body> [표준 사전 요소] </body>
                </entry>
                [ entry 의 반복 ]
        </group>
</sd>
```

헤더, 정의 부분의 세부 구조는 다음과 같다.

```
<sdHeader>
<sdName> [SD 이름]        </sdName>
<sdVer> [version 번호]    </sdVer>
<sdType> [DIC 또는 TEXTCORPUS] </sdType>
<sdDate> [작성 날짜]      </sdDate>
<sdAuthor> [작성자]       </sdAuthor>
```

<sdNote> [설명] </sdNote>

</sdHeader>

<front>

<attrdefgroup>

<attrdef> [attribute 의 정의] </attrdef>

<attrdef> [attribute 의 정의] </attrdef>

[attrdef 의 반복]

</attrdefgroup>

<valdefgroup>

<valdef> [attribute value 의 정의] </valdef>

<valdef> [attribute value 의 정의] </valdef>

[valdef 의 반복]

</valdefgroup>

</front>

<attrdef>

<attrname> [attribute 이름] </attrname>

<attrlist>

<attr> [실제 attribute]</attr>

[attr 의 반복]

</attrlist>

</attrdef>

<valdef>

<valname> [valibute 이름] </valname>

<vallist>

<val> [실제 value]</val>

[val 의 반복]

```
            </vallist>
</valdef>
```

SDML DTD 정의

SDML 은 SGML 의 부분 집합으로 아래와 같은 DTD 를 규격을 의미한다. TDMS 에서 처리 할 수 있는 문서는 아래의 SDML DTD 형식을 따라야 하며 표준 사전의 경우에는 여러가지 사전 형식을 표현하기 위해 sd->group->entry->body 의 내용을 재구성하여 사용한다. 표준 텍스트 문서는 sd->group->file 의 내용으로 고정되어 있으나 앞으로 여러가지 문서 형식을 지원하기 위해서는 표준 사전과 같은 재정의 기능들이 많이 개발 되어야 할 것이다.

```
<!-- SDML (Standard Dictionary Markup Language)version 1.0                    -->
<!ENTITY % doctype "SDML" -- document type generic identifier
                -->
<!--            ELEMENTS    MIN         CONTENT                               -->
<!ELEMENT sd                -           (sdHeader, front?, group)             >
<!ELEMENT sdHeader          -           (sdName, sdVer, sdType, sdDate, sdAuthor, sdNote?)
                >
<!ELEMENT sdName            -           (#PCDATA)                             >
<!ELEMENT sdVer             -           (#PCDATA)                             >
<!ELEMENT sdDate            -           (#PCDATA)                             >
<!ELEMENT sdAuthor          -           (#PCDATA)                             >
<!ELEMENT sdNote            -           (#PCDATA)                             >
<!ELEMENT front             -           (attrdefgroup, valdefgroup)           >
<!ELEMENT attrdefgroup      -           (attrdef)*                            >
<!ELEMENT attrdef           -           (attrname, attrlist)                  >
<!ELEMENT attrname          -           (#PCDATA)                             >
<!ELEMENT attrlist          -           (attr)+                               >
<!ELEMENT attr              -           (#PCDATA)                             >
<!ELEMENT valdefgroup       -           (valdef)*                             >
<!ELEMENT valdef            -           (valname, vallist)                    >
<!ELEMENT valname           -           (#PCDATA)                             >
<!ELEMENT vallist           -           (val)+                                >
<!ELEMENT val               -           (#PCDATA)                             >
<!ELEMENT group             -           ((entry)* | (file)*)                  >
<!ELEMENT entry             -           (wname, body)                         >
```

```
<!ELEMENT wname          --          (#PCDATA)                              >
<!-- default definitions                                    ---------------->
<!ELEMENT body           --          (#PCDATA )*                            >
<!ELEMENT ref            --          (#PCDATA)                              >
<!ATTLIST    ref
             id                       ID          #REQUIRED
             target                   IDREFS      #REQUIRED                  >
<!ELEMENT    ptr         - o         EMPTY                                  >
<!ATTLIST    ptr
             id                       ID          #REQUIRED
             target                   IDREFS      #REQUIRED                  >
```

```
<!--         ELEMENTS    MIN         CONTENT                                -->
<!--         텍스트 코퍼스 관련 부분                                        -->
<!ELEMENT file           --          (header, tdmsfiletext)                 >
<!ELEMENT header         --          (project, class, language, textcode,
                         process, version, serialnumber, filename, title, author, date, typist, note )
                                     >
<!ELEMENT project        --          (#PCDATA)                              >
<!ELEMENT class          --          (#PCDATA)                              >
<!ELEMENT language       --          (#PCDATA)                              >
<!ELEMENT textcode       --          (#PCDATA)                              >
<!ELEMENT process        --          (#PCDATA)                              >
<!ELEMENT version        --          (#PCDATA)                              >
<!ELEMENT serialnumber   --          (#PCDATA)                              >
<!ELEMENT filename       --          (#PCDATA)                              >
<!ELEMENT title          --          (#PCDATA)                              >
<!ELEMENT author         --          (#PCDATA)                              >
<!ELEMENT date           --          (#PCDATA)                              >
<!ELEMENT typist         --          (#PCDATA)                              >
<!ELEMENT note           --          (#PCDATA)                              >

<!ELEMENT tdmsfiletext   --          (#PCDATA | tdmstextcode)*              >

<!ELEMENT tdmstextcode   --          (#PCDATA)                              >
```

<!ATTLIST tdmstextcode

type CDATA #IMPLIED >

표준 사전 요소

이 부분은 <body>에 포함되는 요소로서 기본 설정값처럼 단순한 하나의 필드로 구성될 수도 있고, 복잡한 트리구조를 가질 수 도 있다. 이를 위해서 사용자는 필요한 사전의 구조를 정의해서 <body> 의 구조를 변경하여 사용한다. 일반적으로 새로운 구조의 정의는 SGML 형식과 동일하게 정의하지만, SDML 에 제한적으로 사용해야 하는 규칙이 있다.

1. SGML content model 중 사용 가능한 기본 요소는 , | * + ?로 한다. &는 사전 형식을 결정하는데 무의미하므로 제외한다.
2. 이미 정의된 태그가 있으면, 그 태그 이름을 그대로 사용하여 호환성을 가질 수 있도록 한다.
3. 내부에서 사용되는 attribute 는 ATTLIST 명령으로 정의하여 사용하되, TDMS 편집기와 같은 응용 프로그램에서 제대로 처리될 수 있도록 하기 위해서는 SDML 의 <front>에 그 attribute 와 그 attribute 가 가질 수 있는 값을 정의해야 한다.

<body>는 초기에 다음과 같이 정의되어 있다.

<!ELEMENT body -- (#PCDATA)* >

따라서 새로운 구조로 변형하고자 할 경우, 이 element 만을 재정의(overwrite)한다. 다음은 새로운 구조로 변형시키는 예이다.

```
<!DOCTYPE sd SYSTEM "sdml.dtd" [
<!ELEMENT body - o (품사, 하위범주, 대역어)>
<!ELEMENT 품사 - o (#PCDATA) >
<!ELEMENT 하위범주 - o (#PCDATA)>
<!ELEMENT 대역어 - o (#PCDATA)>
]>
```

새로운 사전은 위에서 정의한 내용에 맞게 SD 형식으로 저장된다.

<sd>
<!-- 위에 정의된 구조(SDF)에 맞는 sd 의 내용 -->
:
</sd>

2 절. 표준 텍스트 코퍼스 형식

표준 텍스트 코퍼스 형식도 표준 사전 형식과 마찬가지로 sd 라는 document type 을 사용한다. 텍스트 코퍼스 형식으로 사용될 때의 구조는 다음과 같다.

<!DOCTYPE sd SYSTEM "sdml.dtd">
<sd>
<!-- 표준 텍스트 코퍼스의 내용 -->
<sdHeader> [sd 헤더 정보] </sdHeader>
<group>
<file>
<header> [텍스트 화일의 헤더] </header>
<tdmsfiletext> [텍스트 화일] </tdmsfiletext>
</file>
<file>
<header> [텍스트 화일의 헤더] </header>
<tdmsfiletext> [텍스트 화일] </tdmsfiletext>
</file>
[file 의 반복]
</group>
</sd>

텍스트 코퍼스는 헤더 정보를 SD와 같은 형태로 가지고 있으며, sdType이 TEXTCORPUS로 되어있다.

참고로 TEI에서는 parameter entity를 사용하여 여러 가지 DTD를 선택할 수 있도록 하고 있다. 현재 SDML에서는 간단하게 이를 처리하기 위해 group내에서 두 가지 형식(DIC 또는 TEXTCORPUS)중 하나를 선택하여 사용할 수 있도록 되어있다. 따라서 sdType으로 정의된 형식과 group내에서 전개되어지는 형식간에 차이가 있을 수 있으므로, 주의해야 한다.

헤더 요소

<header> 요소는 다음과 같다.

<project>

이 문서를 만들도록 한 프로젝트의 명칭을 기록한다.

<class>

문서의 종류를 의미한다. 이 속성의 값은 Corpus, Data, Bilingual Corpus, Official Documents, Program, Electronic dictionary, Thesaurus등을 갖을 수 있다.

<language>

문서의 표현한 사용 언어를 의미한다. English, Korean등이 이에 해당한다.

<textcode>

텍스트 코퍼스를 만든 코드 체계를 나타낸다. 여러 가지 표준 KS코드 중 현재는 KSC 5601 1988 완성형 코드를 표준으로 한다. 이 경우는 'KSC-5601-1988'로 표시한다. 따라서 만약 조합형일 경우는 'KSC-5601-1992'로 표시할 것이다.

<process>

text 문서는 문서를 가공하여 새로운 형태의 문서를 생성할 수가 있다. 이러한 새로운 가공은 문서의 종류와는 구분되어 하나의 과정을 거친 것으로 간주하여 이의 표시를 한다. 즉 문서의 문서간의

process라는 link가 생기게 되며 이러한 link의 label로는 다음과 같은 사항이 표시될 수 있다. Tagged, Bracketed, spell-corrected, morphological_analysis등의 과정이 표시되며 이의 pair로 이전의 과정 수행 이전의 문서를 갖게 된다.

<version>

text문서를 processing하여 새로운 문서의 생성시 이의 생성이 여러 개선된 결과를 거치며 표현될 수 있다. 형태소 해석의 결과 text의 경우 형태소 해석기의 version upgrade에 따라 이에 의해 생성된 가공 text도 달라지게 된다. 이러한 정보를 표현하는 것으로 pair로 문서 가공에 쓰인 프로그램이나 날짜 등 또는 지침서를 가리키는 문서를 갖게 된다.

<serialnumber>

이 번호는 문서의 관리상 갖게 되는 일련번호를 의미한다.

<filename>

text문서의 고유의 file name을 저장하게 된다.

데이터의 저장 위치: 자료가 관리 도구 내에 위치되어 있지 않은 외부의 자료의 경우 이의 장소를 저장하게 된다.

<title>

문서의 제목을 의미한다.

<author>

문서의 작성자를 의미한다.

<date>

문서의 생성 날짜를 의미한다.

<typist>

문서의 전자적인 형태의 입력자를 의미한다.

<inputstatus>

입력의 형태를 기술한다. 즉 Raw OCR image, ORC 결과, 수동 입력, 수동 수정 등의 입력 단계에 대한 정보로 이전 단계의 문서를 pair로 갖게 된다.

<note>

이 화일에 관한 특별한 사항 또는 지시 사항 비고등의 부가 정보를 나타낼 때 사용된다.

표준 텍스트 요소

<tdmsfiletext> 요소는 현재 <tdmstextcode> 태그를 제외하고는 특별히 정의된 것이 없다. 따라서, 현재 텍스트 코퍼스에 포함된 태그는 무시하고 처리한다. 이 텍스트 코퍼스 내의 태그는 TDMS 에서 처리하지 않고, 단순한 텍스트로 간주하며, 외부의 응용 프로그램에 의해 필요에 따라 처리된다.

<tdmstextcode> 태그는 KSC-5601-1988 완성형 코드에서 표현할 수 없는 한글 조합형 글자, 한자, 특수 문자, 외국어 등을 표현하기 위해 만들어졌으며, 다음은 완성형 코드에 없는 조합형 글자를 표현하는 예이다.

<tdmstextcode type=1> ㄸㅗㅁ </tdmstextcode>

이를 위해서 SDML 에는 다음과 같은 정의가 포함되어 있다.

```
<!ELEMENT tdmstextcode    - -     (#PCDATA) >
<!ATTLIST tdmstextcode
          type      CDATA   #IMPLIED >
```

여기에서 속성으로 표현된 type 은 encode 된 글자의 종류를 나타내며, 아래와 같이 정의되어 있다.

Type=1: 조합형 한글

Type=2: 한자

Type=3: 특수 기호

Type=4: 외국어

각각의 type 에 대해 해당되는 문자로 표현하는 방식은 표준 엔코딩 정책에 자세히 정의될 예정이다.

6장. SDML 의미 정의

1절. 저장 구조

sd의 저장 구조는 구현의 방법에 따라 다를 수 있다. 여기에서는 현재 가장 많이 쓰이고 있는 데이터 베이스(RDB 또는 RODB)에서 구현될 수 있는 형태를 설명한다. 현재의 TDMS 구현도 이런 데이터 베이스 시스템에서 개발되고 있다. 예를 들면 다음과 같은 사전이 있다고 가정하자.

<wname> 가난
<품사> 명사
<설명> 살림살이가 넉넉하지 못함. 빈곤
<용례> 가난한 집
</설명>
</wname>
<wname> 가다
<품사> 자동사
<설명> 목적한 곳을 향하여 움직이다.
<용례> 학교에 가다.
</설명>
</wname>

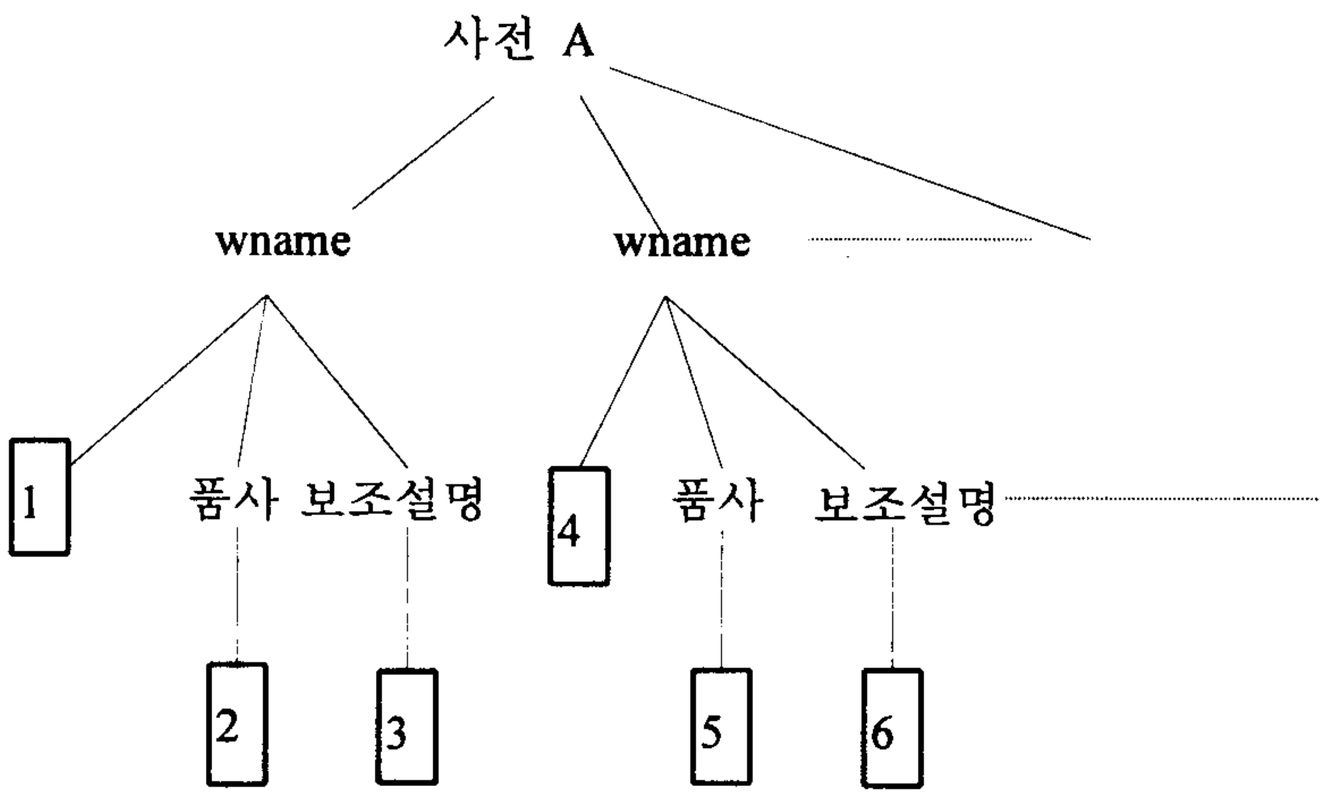
이 사전은 다음과 같은 형태로 내부에 저장된다.

Id	태그 이름	parent	sibling	child	content
0	root	x	x	x	-
1	wname	0	5	2	가난
2	품사	1	3	x	명사
3	설명	1	x	4	살림살이가...
4	용례	3	x	x	가난한 집
5	wname	0	x	6	가다

6	품사	5	x	x	자동사
7	설명	5	x	8	목적한 곳...
8	용례	7	x	x	학교에...

2 절. 구조변환

사전은 기본적으로 트리구조로 표현될 수 있다. 따라서 각각의 사전 필드는 루트로부터 해당 노드까지의 경로로 표현할 수 있다. 예를 들면, 다음과 같이 사전 A가 있다고 하고 각 노드들을 경로로 표시하면 다음과 같다.



1, 4 노드: "사전 A.wname"

2,5 노드: "사전 A.wname.품사"

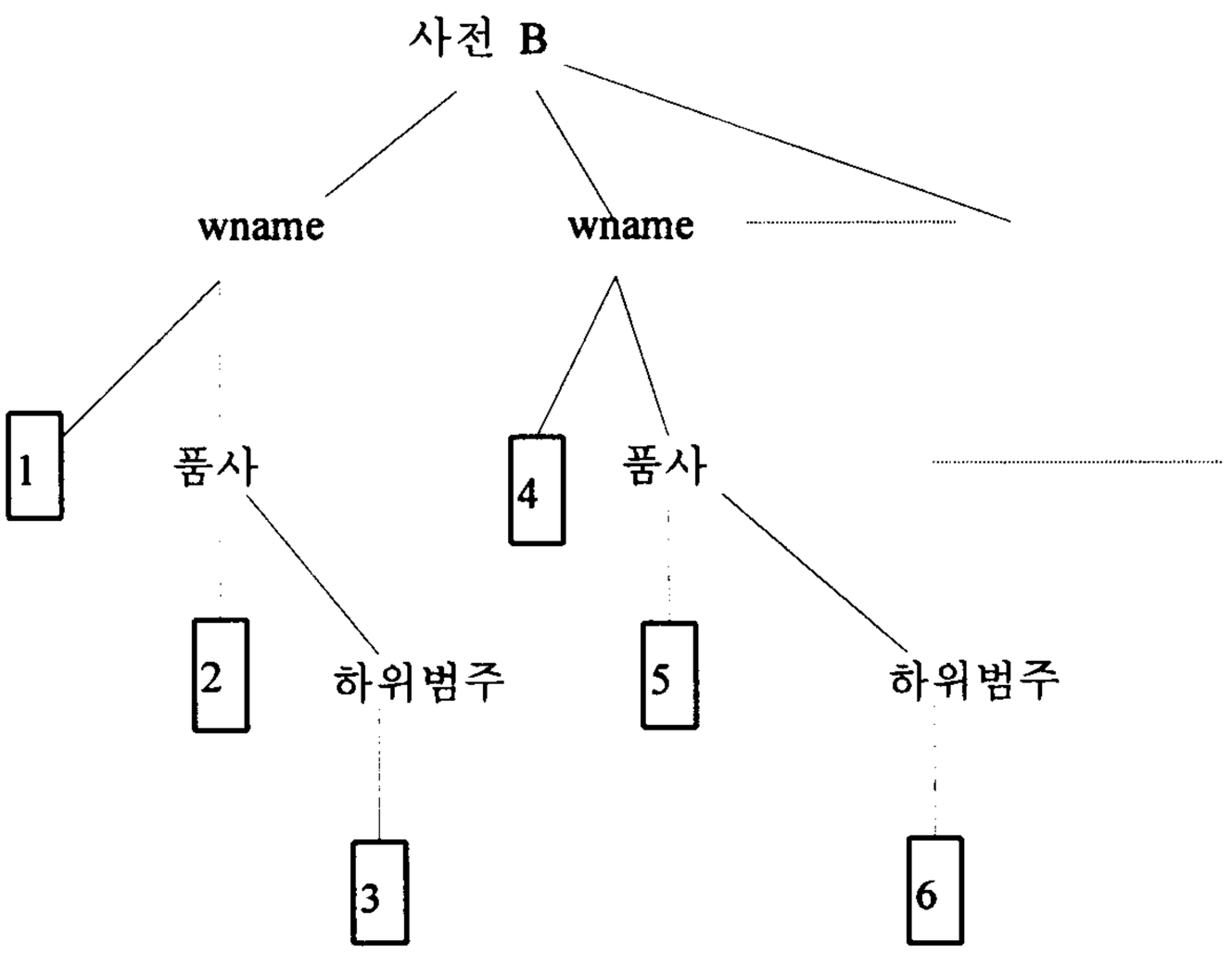
3,6 노드: "사전 A.wname.보조설명"

각 노드가 이와 같이 표시되면, 사전의 구조변환을 위한 operation 을 정의할 수 있으며, 현재의 operation 은 단순한 복사만을 수행한다. 사전 A 를 사전 B 의 구조로 변형시키고자 할 경우, 새로운 사전 구조 B 를 정의한 후, 사전이 A 의 각 필드를 사전 B 의 새 필드에 연결시켜주면 된다. 이러한 연결은 사용자 인터페이스를 통해 구현될 수 있다. 사전 구조 변환의 예를 들면, 아래와 같이 사전 B 의 구조를 사전 A 로 바꿀 경우, 내부적으로는 다음과 같은 수식으로 표현되 처리될 수 있다.

사전 B.wname = 사전 A.wname

사전 B.품사 = 사전 A.품사

사전 B.하위범주 = 사전 A.보조설명



3절. 하이퍼 링크

묵시적 링크(Implicit link)

각 필드 내의 문장 중에 나타난 단어 중에서 다시 사전을 찾아가 확인해 보고 싶은 경우, 그 단어를 선택한 다음 <검색> 키를 누르면 그 단어를 표제어로 하는 곳의 내용을 볼 수 있다. 만약 그 단어가 표제어에 없을 경우에는, 그 단어 또는 어절을 형태소 분석하여 가장 유사하다고 생각되는 표제어의 내용을 보여준다.

명시적 링크(Explicit link)

일반적으로 하이퍼 링크라고 불리는 기능으로서, 문장 중에 밑줄(규격에 따라 다를 수 있음) 등으로 표시된 단어를 클릭하면, 그 단어의 마크업에 표시된 표제어의 내용을 볼 수 있다. 묵시적 링크(implicit link)와는 다르게 사용자가 보고 있는 단어를 표제어로 반드시 찾을 필요 없이 관련된 다른 표제어로 내용을 보여 줄 수 있다. 또 반드시 표제어로 가지 않고, 직접 필요한 필드 내로 뛰어 갈 수 있다.

지원되는 명시적 링크의 마크업으로는 <ref>와 <ptr>이 있다. <ref>는 밑줄로 표시될 단어의 앞뒤에 넣어 주면 되고, <ptr>은 그런 단어 없이 바로 그 위치에서 특수 표시가 되어 링크가 되도록 해 준다. 이 <ref>와 <ptr>은 데이터가 들어 갈수 있는 곳에 임의로 들어갈 수 있다.

<ref target=a27> 예외 규칙이 있다.</ref>

<ptr target=a27>

Hyper text 와 관련된 규칙은 TEI 의 Hyper link 의 부분으로서 HyTime 의 Hyper link 기능과도 호환이 된다.[참조].

이를 위한 SDML 의 정의는 다음과 같다.

```
<!ELEMENT ref        - -     (#PCDATA) >
<!ATTLIST  ref
                     id              ID          #IMPLIED
                     target          IDREFS      #IMPLIED >

<!ELEMENT ptr        - o     EMPTY       >
<!ATTLIST  ptr
                     id              ID          #IMPLIED
                     target          IDREFS      #IMPLIED >
```

4 절. 구조검색

TDMS 에서는 표제어만을 찾아 갈 수 있을 뿐만 아니라, 특정한 필드 내의 특정한 단어가 포함된 문장 등을 찾아 낼 수 있다. 예를 들면, 다음의 사용자 질의에 대해 처리 방법을 살펴보면 다음과 같다.

질의어:

"<용례>에 '학교'가 포함된 문장의 <wname>를 찾아라."

처리방법:

1. content 에서 '학교'가 포함된 record 를 찾는다.
2. 이 record 들 중에 tag name 이 <용례> 인 것만을 찾는다.
3. 각 record 의 parent 를 따라가 tag name 이 <wname> 인 record 를 찾아 표시한다.

경우에 따라, 1 단계와 2 단계를 바꾸어 실행해도 무방하다.

7 장. TDMS 의 구현

1 절. 개발목표

TDMS 는 위에서 제시한 문제점들을 해결하고, SGML 에 관한 지식이 없는 사람이라 할지라도 SGML 형식의 SDF 에 맞춰 만들어진 사전이라면 쉽게 사용할 수 있는 시스템을 개발하는 것이 목표다. 세부적인 목표는 다음과 같다.

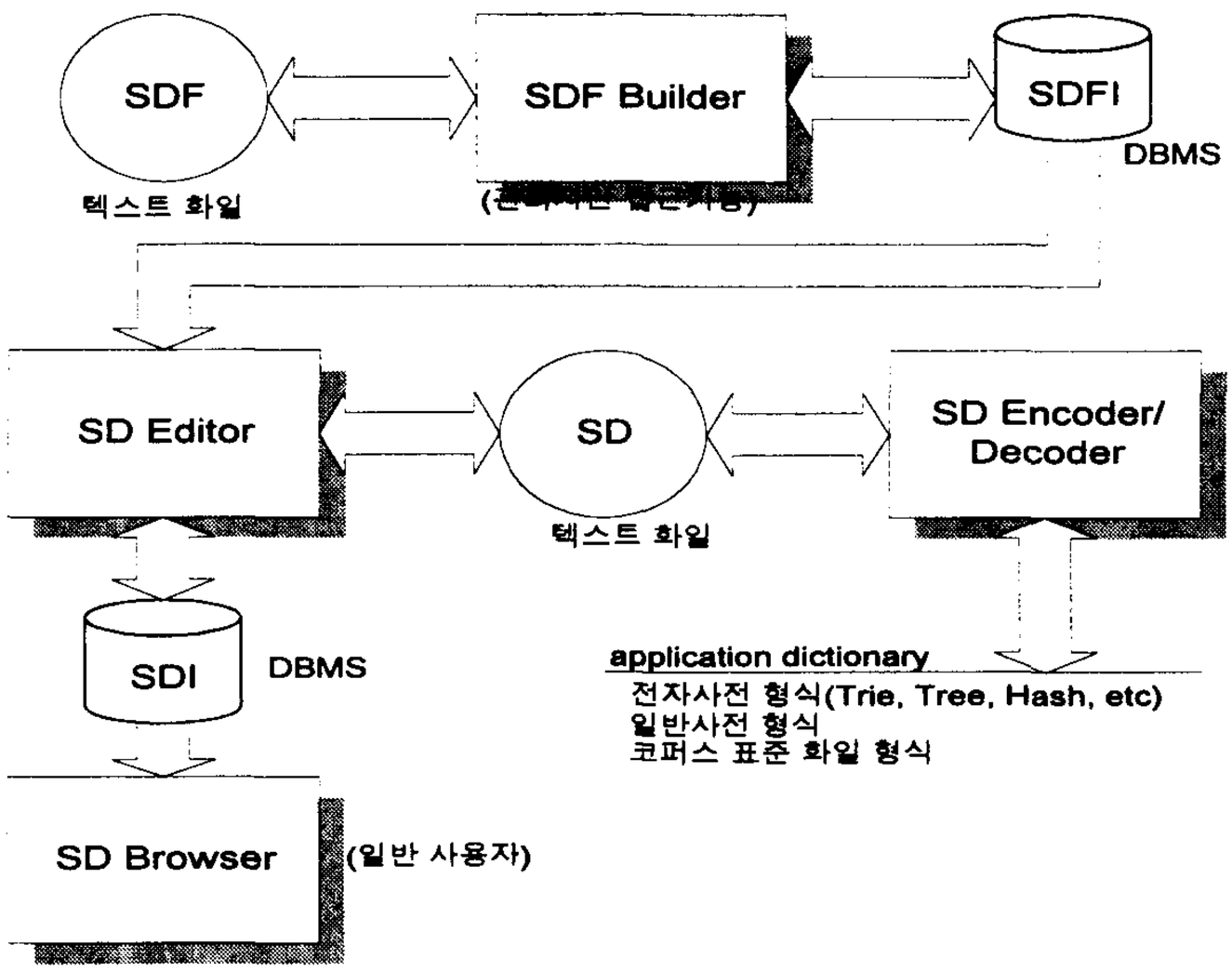
1. SDF 의 작성 및 변경을 자유롭게 할 수 있어야 한다. 특정 분야의 표준사전을 필요로하는 사용자는 SGML 을 몰라도 SDF 를 작성할 수 있으므로 TDMS 의 활용성을 크게 할 수 있다.
2. SD Editor 는 SDF 에 정의되어 있는 정보에 의해서 입력에 필요한 사항을 쉽게 파악을 할 수 있어야 한다. 얼마나 능률적으로 표준사전을 구축하느냐를 좌우한다.

3. 입력된 사전의 내용을 쉽게 찾아 볼 수 있어야 한다. 사전을 효과적으로 검색할 수 있는 다양한 검색 기능과 상관 참조 기능을 부여하여야 한다.
4. 다른 시스템에서 사용하고 있는 전자사전과 호환성을 고려한다. SD 라는 SGML 형식의 문서를 매개로 하여 서로의 데이터를 교환한다. 일반적으로 많이 사용되는 전자사전들은 변환하는 프로그램을 지원한다. SD 를 구축하는데 기존의 전자사전을 이용하여 상당한 시간과 비용을 절약할 수 있다.
5. 다양한 응용 소프트웨어 환경과 하드웨어 플랫폼의 사용을 고려하여 Server/Client 방식으로 설계한다. 서버는 Unix 나 혹은 Windows-NT 로 하며 Client 는 MS-Windows 를 사용하는 환경으로 개발하여 세계에서 가장 널리 보급되어 있는 환경을 사용토록 한다.
6. 데이터베이스는 상용 DBMS 를 사용하여 자료보관의 신뢰성을 높이고 자료접근의 보안을 유지하며 향후 확장이나 유지보수에 만전을 기한다. 또한 DBMS 를 Server/Client 방식과 같이 사용하여 이기종간의 이식성에 높은 효과를 볼 수 있다.
7. 사전관리자, 입력요원과 사용자의 GUI 는 메뉴, 툴바, 도움말 등을 이용하여 쉽게 배우고 사용이 편리하게 하여 생산성을 높인다.
8. 전자 사전은 온라인으로 이용할 수 있어야 하며 각 사전은 네트워크 상에서 운영되어 이용자의 요구 사항에 응답할 수 있어야 한다.
9. 전자 사전을 관리하는 기관의 역할 및 위상을 정립해야 한다. 용어 데이터의 수집, 관리, 갱신, 데이터의 품질 관리, 여러 가지 정보도구의 생성 등에 대한 관리 기관의 역할과 위상을 정하여야 한다.

2 절. TDMS 의 구성

사전 및 텍스트 관리 통합시스템은 다음과 같이 크게 4 개 요소로 구성된다. 표준 사전 형식(SDF)을 유연성 있게 정의할 수 있는 **SDF Builder**, 그 규격의 사전을 만들어 주고 쉽게 유지 보수할 수 있는 **SD Editor**, 입력되어 있는 사전데이터들을 쉽게 검색해 볼수 있는 도구인 **SD Browser**, 그리고 기존의 사전들을 표준 사전 포맷으로 바꾸어 주며 역으로 표준사전을 특수 구조로 바꾸어 주는 변환프로그램인

SD Encoder/Decoder 로 구성된다.(그림 3)



[그림 3] TDMS 구성도

3 절. TDMS 모듈 설계

TDMS 를 구성하는 4 개의 요소를 더 자세히 나누면 모두 13 개의 모듈로 구성된다.

(1) 표준 기술 언어 작성기

SGML 에 관한 전문적인 지식을 가지고 있지 않은 사람도 TDMS 의 표준 사전 포맷(SDF: Standard Dictionary Format)을 쉽게 작성하고 이를 변경할 수 있게 하는 모듈이다. 작성된 SDF 는 DBMS 에 저장된다. SDF 의 작성 및 편집을 GUI 를 통해서 편리하게 할 수 있고, SDF 의 IMPORT 와 EXPORT 기능을 이용하여 SDF 를 파일의 형태로 입력과 출력을 할 수 있다.

(2) 화면 편집기

정의되어 있는 표준 사전 포맷(SDF)에 맞게 데이터를 보여주는 화면을 정의할 수 있도록 하는 모듈이다. 여기서 정의되어 있는 화면의 모양은 사전 입력기에서 사용할 수 있다. 입력하는 위치나 크기, 색깔 등을 변경할 수 있다. 태그 이름도 바꿀 수 있다. 한 SDF 에는 여러 개의 입력 화면을 만들 수 있다. 원하는 입력 화면을 선택하여 입력을 한다.

(3) 사전 입력기

표준 사전 포맷에 따라서 표준 사전 데이터를 입력하는 모듈이다. 사전 데이터를 입력하고 편집을 할 수 있고 SD 의 IMPORT 를 하여 DB 에 저장하거나 EXPORT 할 수도 있다.

(4) 텍스트 입력기

일반 텍스트 코퍼스 데이터를 입력하는 모듈이다. 입력된 데이터는 Rich Text Format 으로 DB 에 저장이 된다. 일반 텍스트 코퍼스를 편집하거나 출력할 수 있다. 입력 데이터의 색깔이나 폰트 등의 속성을 지정할 수 있다.

(5) 데이터 베이스 관리시스템 접속기

TDMS 의 데이터베이스 관리 시스템에 접속을 하거나 사용자를 검사하는데 사용하는 USER OBJECT 이다. 입력된 사전 데이터를 저장하는 데이터베이스는 여러 개가 있을 수 있다. 이를 등록하여 두고 리스트 박스를 이용하여 선택할 수 있게 한다. 사용자 등록도 할 수 있게 한다.

(6) 표준 출력기

데이터베이스에 입력이 되어 있는 표준 사전의 데이터를 선택하여 SDML 형식의 텍스트 파일로 출력을 하는 USER OBJECT 이다. TDMS 에서 지원하는 검색 방법을 이용하여 한다. 검색된 데이터에 대한 필터링을 하여 선택적으로 출력을 한다.

(7) 하부구조 변환기(SD Encoder/Decoder)

TDMS 의 기본 형식이 표준사전(SD)을 각종 응용 사전으로 변환하는 Encoder 와 응용 사전을 표준 사전 형식으로 만들어 주는 Decoder 로 나누어진다. 이 모듈의 목적은 기존의 다양한 형태의 사전으로부터 정보를 추출하여 TDMS 에서 통합 관리하고, 형태소 분석기 등의 응용 프로그램에서 다시 사용할 수 있게 하기 위하여 응용 프로그램에 종속적인 응용 사전을 생성해 내는, TDMS 와 응용사전 간 인터페이스를 담당한다.

(8) 데이터베이스 브라우저

사전 및 텍스트 관리시스템에 의해서 입력되어 있는 사전 데이터들을 일반 검색을 하거나 구조 검색을 하여 브라우징을 할 수 있는 모듈이다. 일반 검색은 표제어를 이용하여 검색하는 것이고 구조 검색은 검색하고자 하는 사전의 구조를 이용하여 검색하는 것을 말한다. 예를 들어 설명하면, "'용례'에 '학교'라는 단어를 포함하고 있는 단어"들을 찾는 것과 같은 식으로 검색을 하는 방법을 말한다.

(9) 서버 관리기

사전 및 텍스트 관리시스템의 사용자, 사전의 종류 및 데이터를 관리하기 위한 모듈이다. 사용자 관리는 데이터에 대한 접근의 제한을 위해서 필요하다. 사전을 새로 만들거나 입력된 데이터에 대한 통계 처리를 한다.

(10) 클라이언트 개인용 데이터베이스 관리시스템

사전 및 텍스트 관리시스템을 STANDALONE 으로 사용할 수 있게 한 모듈이다. 데이터베이스를 로컬에 갖고 독자적으로 표준 사전 포맷을 정의하고 사전 데이터를 입력할 수 있다.

(11) 네트워크 참조부

일반 어플리케이션에서 사전 및 텍스트 관리시스템 서버에 사전 조회를 하는데 사용할 함수들이다. 조회 문장은 SGML 형식을 이용하여 다른 시스템에의 이식성을 높였다. 일반 DYNAMIC LINK LIBRARY 의 형태로 되어 있다. 함수는 서버에 접속을 하는 TdmsConnect(), 실제적으로 조회를 하고 응답을 받는 TdmsQuery(), 그리고 서버와의 접속을 끊는 TdmsDisconnect()의 세가지 가 있다. 통신은 소켓을 이용해서 한다.

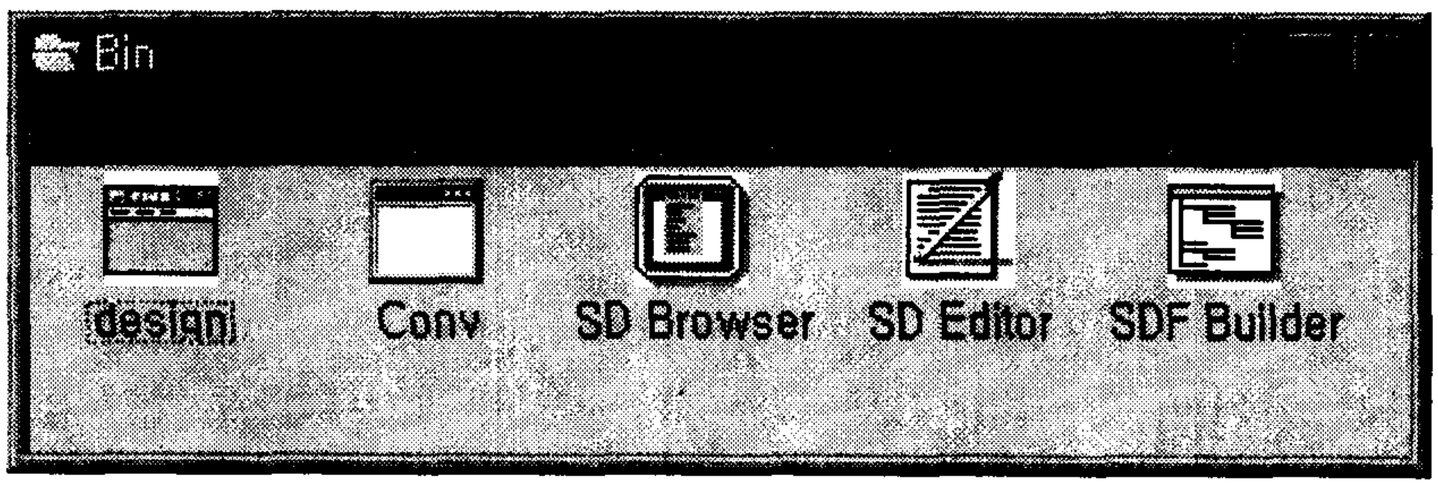
(12) 이형태 데이터베이스 통합기

사전 및 텍스트 관리시스템의 데이터의 양이 많아지거나 여러 기관에서 관리를 하게 되는 경우에는 여러 개의 서버를 사용할 수 있다. 또한 다양한 데이터베이스를 사용할 수도 있다. 이런 경우에는 전체 시스템을 바라보면서 관리를 할 필요가 있다.

(13) 라이브러리 표준 접속장치

사전 및 텍스트 관리시스템의 사전 조회 서버이다. 클라이언트에서 소켓을 통해서 사전 조회를 SGML 형식의 문장으로 하면 이를 분석하여 SQL 문장으로 변환한다. 변환된 SQL 문장을 이용하여 데이터베이스 관리시스템에 조회를 하고 결과를 클라이언트에 돌려준다.

4 절. TDMS 프로그램 모듈



[그림 4] TDMS의 모듈들

TDMS 의 프로그램 모듈은 [그림 4]와 같다. 여기에서 design 아이콘은 사전을 입력하는 화면을 디자인하는 프로그램이고 Conv 아이콘은 DOS 프로그램으로 SD Encoder / Decoder 프로그램이다.

1. SDF Builder

사전은 용도에 따라서 필요한 사항이 다르고 또한 같은 용도를 갖고 있다 하더라도 만든 사람에 따라서 형태가 달라진다. SDF 를 정의하기 위해서는 다양한 형태의 사전들을 용도별로 분류하고 다시 분류된 사전의 형태들에서 표준을 찾아내야 한다. 따라서 SDF 는 해당분야별 전문가들이 상당한 시간을 들여서 신중하게 만들어야 한다. 그리고 SDF 를 만들어 사용하다 보면 변경해야 하는 상황이 발생할 수 있다. SDF 의 작성은 관리자에 의해서 관리되어져야만 한다.

SGML 에 관한 전문적인 지식을 갖고 있지 않은 사람도 쉽게 SDF 를 작성하고 이를 변경할 수 있는 프로그램이다. 작성된 SDF 는 DBMS 에 저장된다. 그림 1 에서 SDFI 라고 이름지어져 있다. 텍스트로 된 SDF 와 구분하여 시스템에 INTERNAL DBMS 에 저장되어 있다는 의미이다.

기존에 정의된 SDF 화일을 읽어 들여 SDFI 로 저장할 수도 있고 SDFI 를 SDF 화일로 출력을 할 수도 있다.

2. SD Editor

정의된 SDFI 를 이용하여 표준사전 SD 를 입력하고 편집하는 프로그램이다. 현재 입력되어 있는 단어를 찾아 편집하거나 삭제할 수도 있고 새로운 단어를 삽입할 수도 있다. 입력된 사전은 DBMS 에 저장되어진다. 저장된 사전을 SDI 로 명명을 한다. 이것 역시 SDFI 와 마찬가지로 INTERNAL 한 형태로 저장되어 있음을 구분하기 위해서 'I'를 붙였다. SD Editor 에서는 SDI 를 SD 로 출력을 할 수도 있고 SD 를 읽어 SDI 로 만들 수도 있다.

단어를 입력할 때는 SDFI 에 정의되어 있는 입력 가능 사항을 보여주고 선택을 할 수 있도록 한다.

여러 가지의 SDI 를 MERGE 하여 새로운 SDI 를 만들 수도 있고, 한 SDI 로 부터 필요한 항목을 골라 새로운 SDI 를 만들 수도 있다.

3 . SD Browser

입력되어 있는 표준사전의 내용을 검색하는 도구이다. 입력되어 있는 데이터를 BROWSING 하여 보여주다가 선택을 하면 선택된 단어만을 더 자세히 보여준다. BROWSING 되는 데이터는 사용자가 임의로 선택을 할 수 있도록 한다. 보여지는 화면에서 단어를 선택하여 해당 단어를 찾아 보여 줄 수도 있다.

4 . SD Encoder/Decoder

SD 를 읽어서 다른 응용시스템에서 사용하는 방법으로 데이터를 입력시켜주거나 다른 시스템에 있는 데이터를 SD 로 변환을 시켜주는 프로그램이다. 다른 응용시스템에서 사용하는 사전의 종류로는 전자사전 형식, 일반사전 형식, 그리고 코퍼스 표준 화일 형식을 지원할 예정이다. 전자사전의 형식으로는 Trie, Tree, 그리고 Hash 등이 있다.

SD 가 일반화되면 Encoder/Decoder 를 거치지 않고, 응용시스템에서 직접 SD 를 읽어 들이게 될 것이다

5 절. TDMS 의 개발 상황

1 . 개발환경

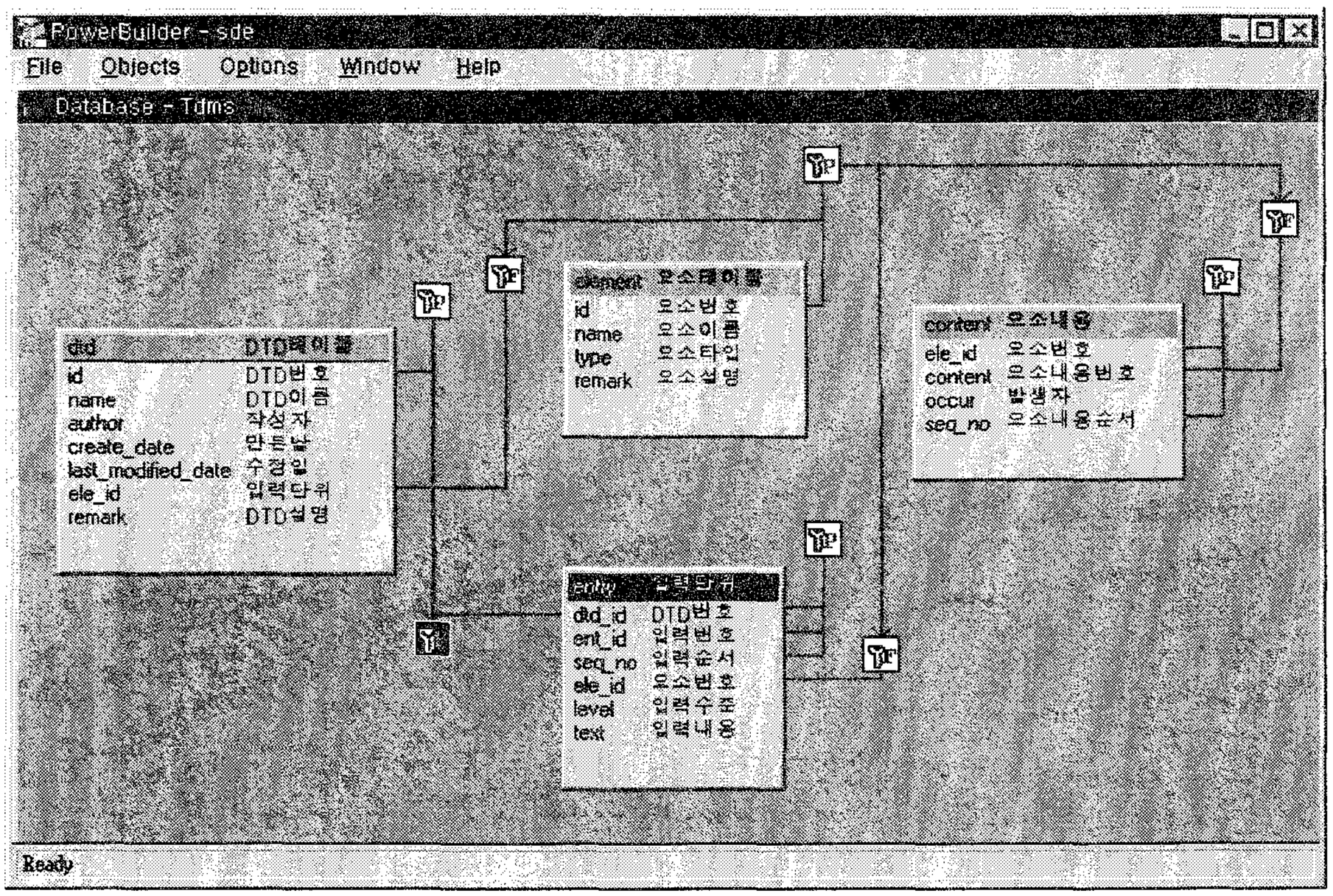
개발환경은 다음과 같다.

- Server OS 는 Windows NT 3.51
- Client OS 는 Windows 95 와 MS Windows 3.1
- 개발도구는 PowerBuilder 4.0 과 WatcomC++ 10.0
- DB 디자인도구는 ER/Win for PowerBuilder

- DataBase 는 SQLServer 6.0 과 WatcomSQL 4.0

2. 데이터베이스의 구조

DB 의 구조를 설계하기 위해서는 충분한 검토를 통해서 이루어져야 한다. DB 의 효율성을 위해서는 다양한 형태를 시험해 보아야 한다. 현재는 [그림 5]에서 설계된 모양을 가지고 모든 프로그램을 개발하였다.



[그림 5] DB구조

DTD 테이블은 SDF 를 의미한다. ELEMENT 테이블은 SDF 에서 사용하거나 사용할 수 있는 요소를 의미한다. 요소가 그룹인 경우에는 CONTENT 테이블에서 그룹요소에 해당하는 내용요소들을 저장한다. 결국 ELEMENT 와 CONTENT 테이블을 이용하여 SDF 의 트리구조를 이룬다.

ENTRY 테이블은 DTD 에 의해서 입력되어지는 데이터를 저장하는 테이블이다. 하나의 ENTRY 는 SDF 에서 정의된 ELEEMENT 하나를 의미한다.

여기에 추가되어야 할 테이블은 요소별 속성을 정의하는 테이블, ENTITY 를 정의하는 테이블 등이 있어야 한다. 그리고 SD 의 관리를 위해서 사용하게 될 사용자 테이블도 있어야 한다.

6 절. 프로그램의 개요

1. stand-alone 버전

TDMS stand-alone 버전은 개인용 컴퓨터 안에 TDMS 의 모든 프로그램과 데이터베이스 및 사전을 가지고 있다. Stand-alone 버전은 데모용으로 사전을 검색하거나 소량의 사전을 편집하기에 알맞다. 또한 네트웍에 연결할 수 없는 경우 부분적인 사전 작업을 할 수 있다.

2. server/client 버전

TDMS server/client 버전은 server 에 데이터베이스 서버와 사전을 가지고 있고 사용자 인터페이스 프로그램은 개인용 컴퓨터에 두고 여러명이 대량의 사전을 공동으로 편집하기에 알맞다. client 에서 작성된 사전은 SD 형태로 저장된 후 한꺼번에 server 에 입력되거나 실시간에 네트웍을 통해 직접 server 의 내용을 고치는 것이 가능하다.

stand-alone 버전과 server/client 버전의 사전 관리기는 동일한 사용자 인터페이스를 가지고 있으므로 server/client 버전을 사용할 때 네트웍 로그인을 하는 점을 제외하면 사용하는 방법은 같다.

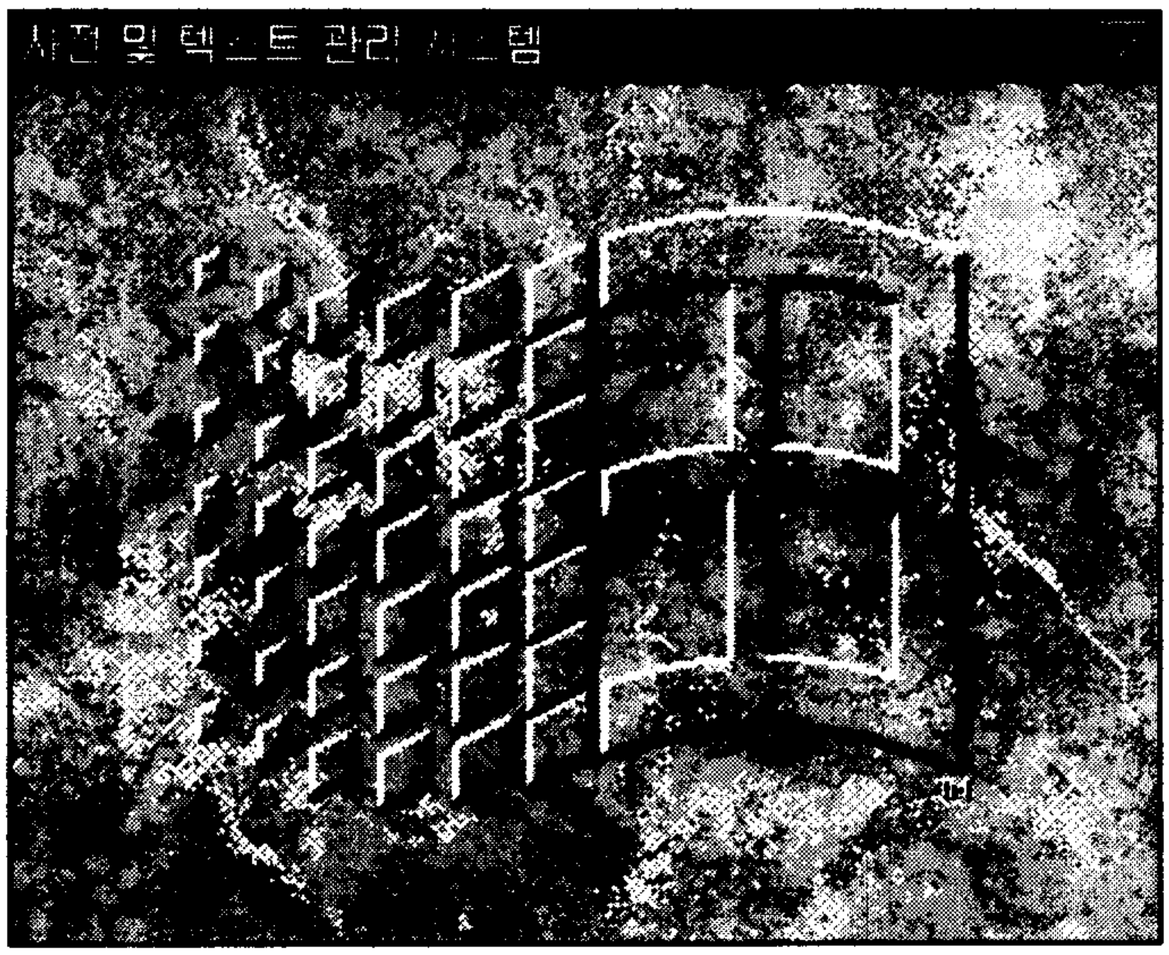
3. SDF Builder

SGML 은 매우 광범위한 내용을 가지고 있다. 따라서 필요한 사전의 SDF 를 만들기 위해서는 SGML 에 관한 많은 연구가 필요하다. 만일 SDF 가 잘 못되어 있는 경우에는 SD Editor 에서 많은 문제점을 발생시킬 것이다. 일반적인 DTD

Builder 에서 SDF 를 작성한 경우에는 이를 읽어들일 수 있어야 한다. SDF 를 읽고 파싱을 하는 것 또한 SGML 을 잘 알아야 한다. 만일 SDF 에 문제가 있을 경우에는 이를 찾아내야 한다.

SDEditor 에서는 SDFBuilder 에서 검사를 해서 합격한 경우에만 사용을 하도록 한다. SDEditor 를 이용해서 데이터가 이미 입력이 된 경우에는 SDF 의 구조를 함부로 바꾸지 못한다. 바꾸기를 원하는 경우에는 전체 데이터를 검색하여 구조를 변경시켜야 하기 때문에 데이터의 양에 따라서는 엄청나게 많은 시간이 필요하게 될 것이다.

프로그램을 처음 실행시키면 다음 [그림 6]의 로고화면이 나타난다.



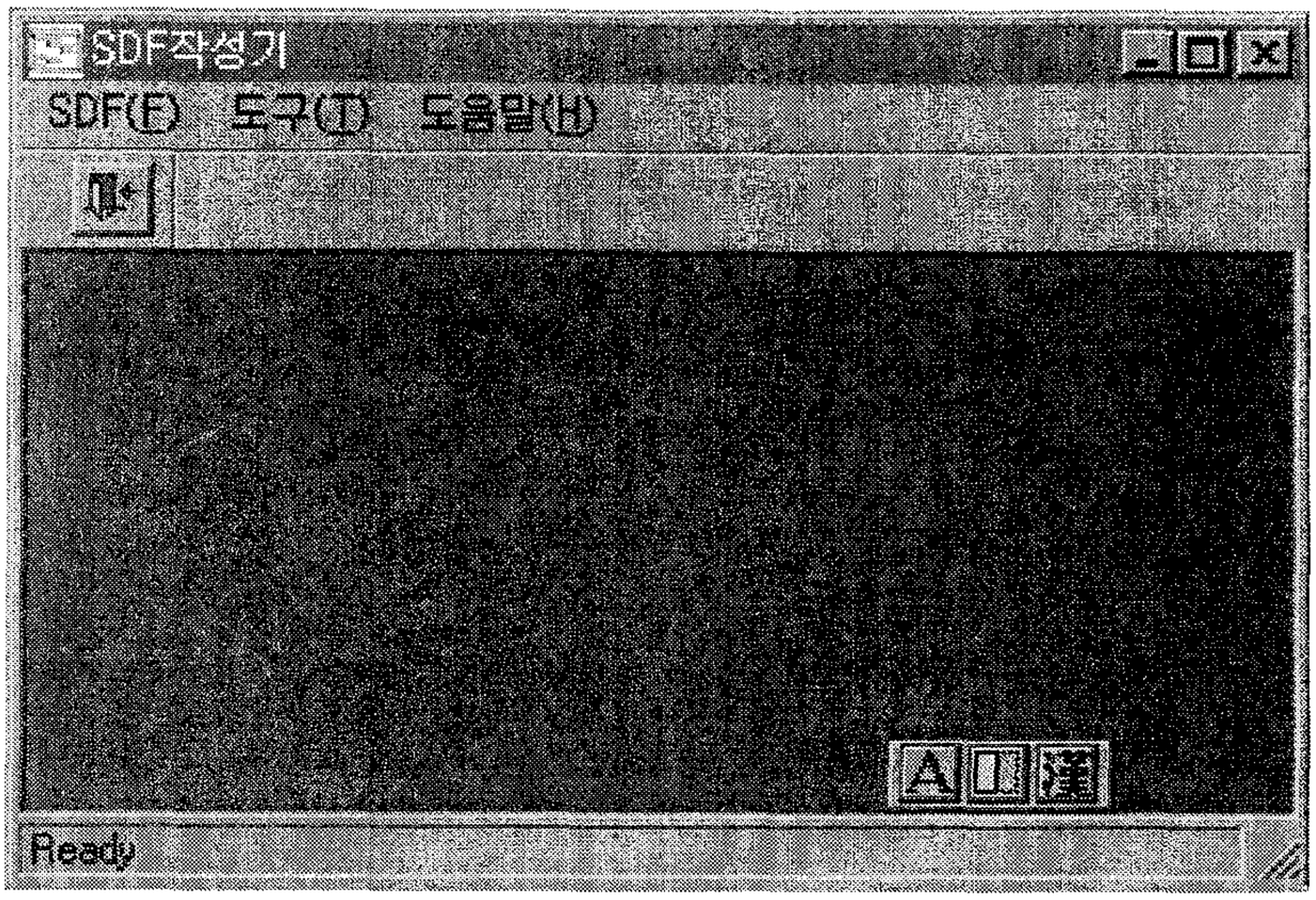
[그림 6] 로고 화면

TDMS 에 있는 프로그램들은 위의 로고화면이 항상 먼저 나타난다. 로고화면이 나타나 있는 동안 프로그램에 필요한 준비를 한다. 준비가 끝나면 로고화면은 사라지고 실제적인 화면이 나타난다.

Server/client 버전의 경우에는 위의 로고화면이 나타나지 않고 대신 네트웍 로

그인 화면이 나타난다. ([부록] "TDMS 설치 및 사용 설명서" 참조)

다음 [그림 7]은 SDF Builder 의 초기화면이다.



[그림 7] SDF Builder의 초기화면

위의 메뉴는 다음의 SDF 작성화면의 메뉴에 다 속하여 있으므로 다음에서 설명을 하겠다.

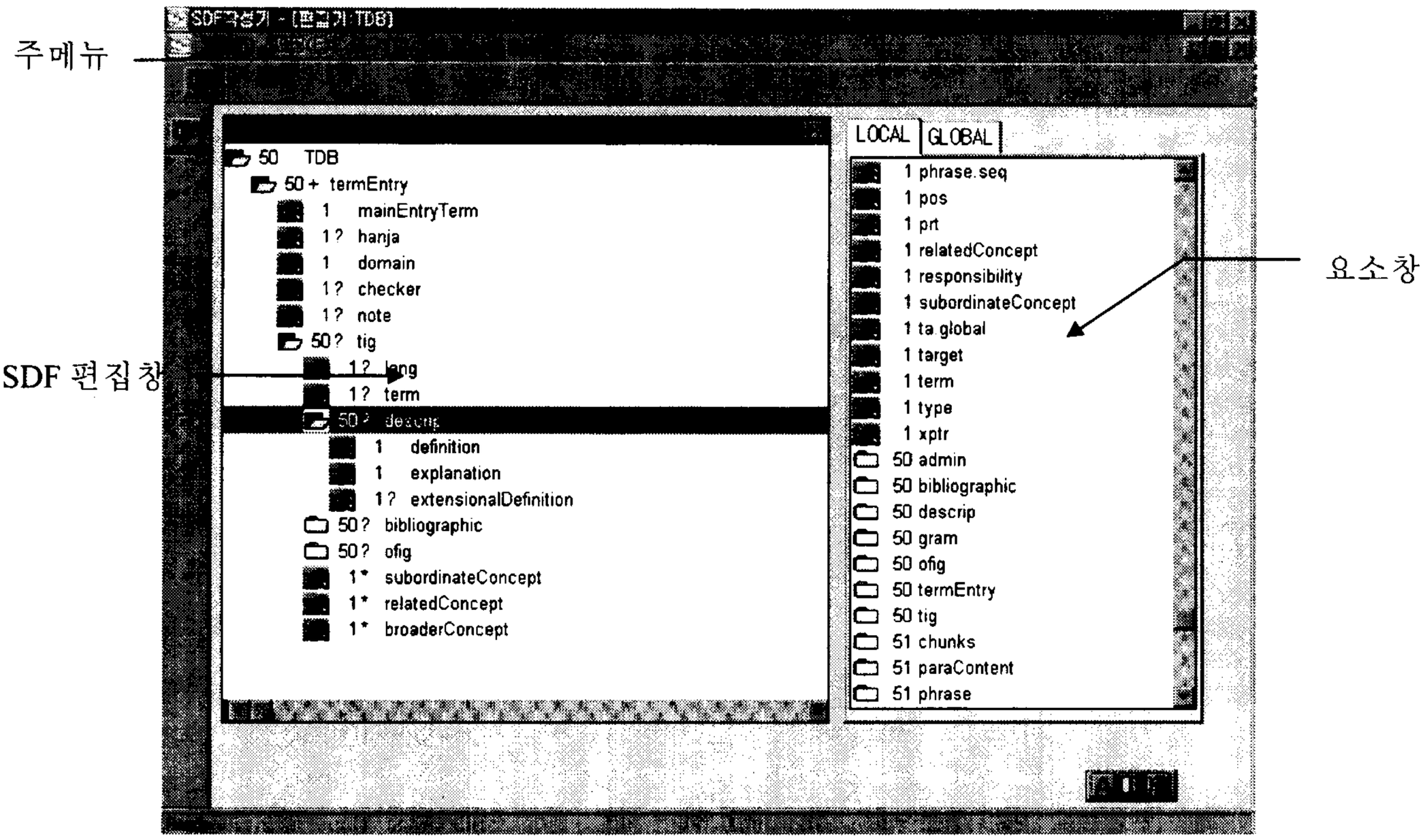
위의 화면의 메뉴에서 SDF 하위 메뉴인 선택메뉴를 선택하면 SDF 를 선택하는 화면이 나타난다. 그 화면에서 SDF 를 하나 선택하면 다음 [그림 8]의 SDF 작성화면이 된다.

(1) 화면설명

SDF 입력화면은 크게 주메뉴와 SDF 편집창 그리고 요소창으로 이루어진다.

왼쪽위에 있는 SDF 창은 현재 만들어진 SDF 의 이름을 보여준다. 여기서 선택을 하여 SDF 를 수정할 수 있다. 새로운 SDF 를 만들기를 원하면 메뉴에서

[그림 8] SDF 작성화면



'새 SDF'를 선택하여 필요한 항목을 입력하면 된다. SDF의 일반적인 사항들을 여기서 결정된다. 왼쪽 밑의 창은 현재 선택되어 있는 SDF에 관한 일반적 정보를 상세하게 보여주는 창이다. 오른쪽 창은 현재 선택되어 있는 SDF의 구성요소를 보여준다. DOUBLE CLICK을 하면 하위의 요소를 보여주고 다시 하면 하위의 요소들을 감춘다.

여기서 각 요소에 대하여 화면에 나타나는 모양을 보자. 맨 윗줄의 단어를 보면 앞에 폴더가 있다. 이것은 단어라는 요소는 그룹이라는 것을 알 수 있다. 그룹이 아닌 경우에는 'P'라는 비트맵이 나오는데 모양은 특별한 의미를 갖지는 않고 RESERVED한 요소임을 의미한다. 비트맵을 이어서 나오는 숫자는 요소타입을 의미한다.

그룹의 경우에는 CONNECTOR의 종류에 따라서 3가지로 분류된다. '50'은는 계속 이어져 나오는 SEQUENTIAL CONNECTOR인 ','에 해당한다. '51'은 OR CONNECTOR인 '|'에 해당하며, '52'는 AND CONNECTOR인 '&'에 해당한다.

AND 는 순서에 상관없이 반드시 그룹에 속한 요소들이 반드시 모두 나와야만 하고 OR 는 그룹에 속한 요소 중에서 한가지만 나와야 함을 의미한다.

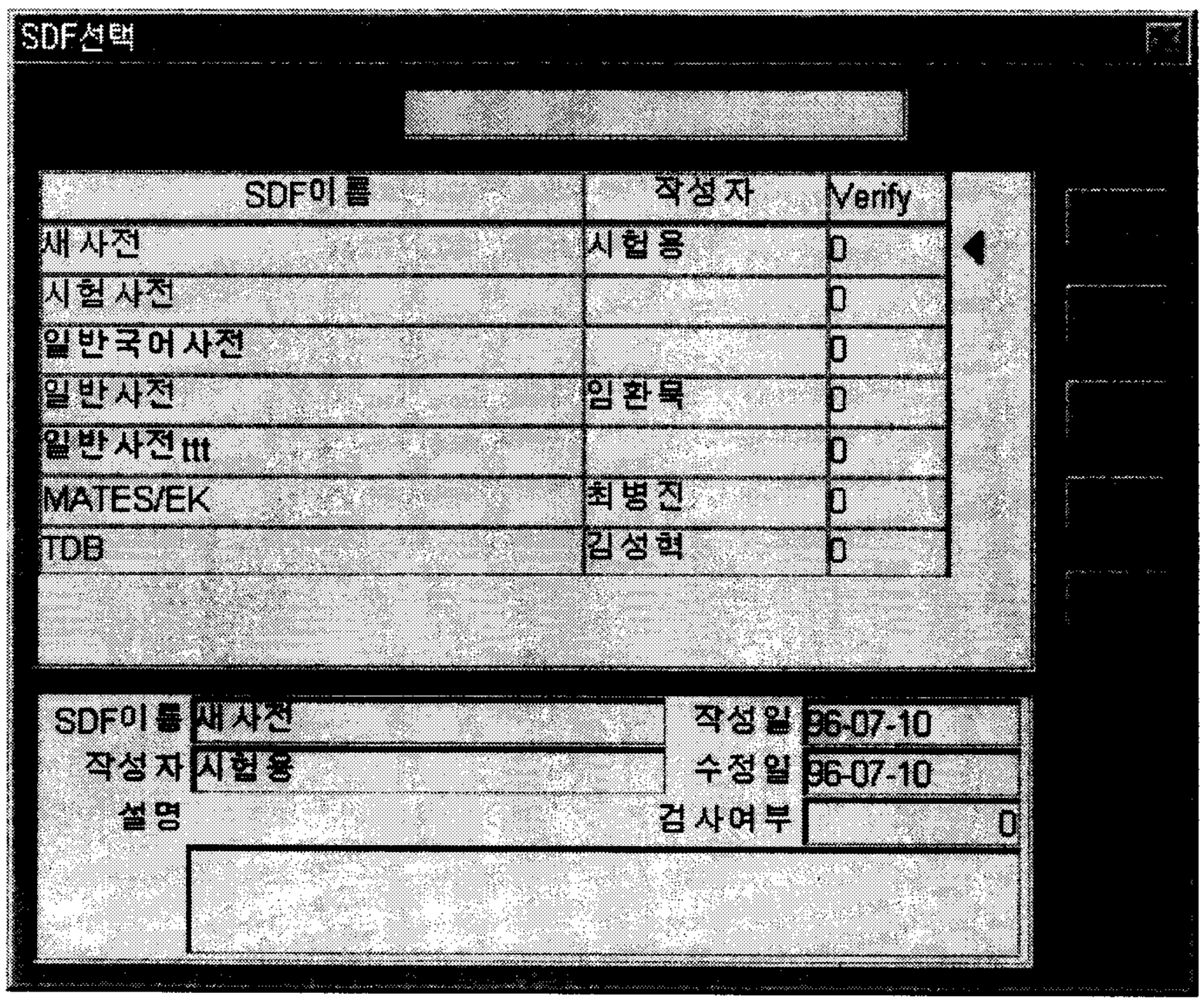
RESERVED 한 요소는 충분한 검토를 통하여 결정해야 할 사항이다. 여기서는 #PCDATA 만을 정했다.

(2) 메뉴

주메뉴는 에서 SDF 는 새로운 SDF 를 만들거나 기존의 SDF 를 선택할 수 있는 화면을 제공한다.

다음 [그림 9]는 SDF 를 선택하는 화면이다.

[그림 9] SDF 선택화면



이 화면에서는 SDF 를 선택하는 것은 물론이고 새로운 SDF 를 만들거나 수정

할 수도 있고 삭제할 수도 있다. 삭제를 하는 경우에는 입력되어 있는 데이터도 모두 삭제되므로 신중을 기해야 한다. 위 그림에서 SDF의 색깔이 다른데 이는 데이터가 들어 있는지를 알린다. 파란색의 경우에는 데이터가 이미 들어 있으므로 선택을 하더라도 SDF의 모양을 변경시킬 수는 없고 단순히 구조를 볼 수 밖에는 없다.

편집 메뉴는 데이터가 이미 들어 있는 SDF일 경우 사용할 수 없다.

오른 쪽의 버튼들은 수정을 하는데 필요한 명령버튼들이다. '수정'은 현재 선택이 된 요소의 OCCURRENCE INDICATOR를 변경할 수 있다. '삽입'과 '추가'는 현재 선택이 되어 있는 레벨과 같은 레벨에 요소를 삽입하거나 추가할 수 있다. '하위'는 그룹요소에 하위의 요소가 없을 경우에 하위요소를 추가하는 명령이다.

'삭제'버튼을 누르면 현재 선택되어 있는 요소를 삭제한다. 그룹인 경우에는 폴더가 닫혀있어야만 한다.

'저장'을 누르면 현재 입력되어 있는 SDF가 실제로 사용가능한 상태인지를 검사한다. 모든 요소가 제대로 입력이 되어 있으면 사용이 가능하다는 표시를 한다. 검사는 맨 하위의 요소가 기본요소로 되어 있는지를 한다. 만일 기본요소로 되어 있지 않으면 그룹이 되기 때문에 그룹에 대한 정의가 다 되어 있지 않다는 의미가 된다. 기본요소로 되어 있지 않으면 경고를 한다. 이러한 SDF는 Editor에서 사용을 할 수가 없다. 따라서 플래그를 만들어 사용이 가능한지를 표시한다. 여기서 OCCURENCE INDICATOR에서 서로 모순되는지의 여부를 검사해야 한다.

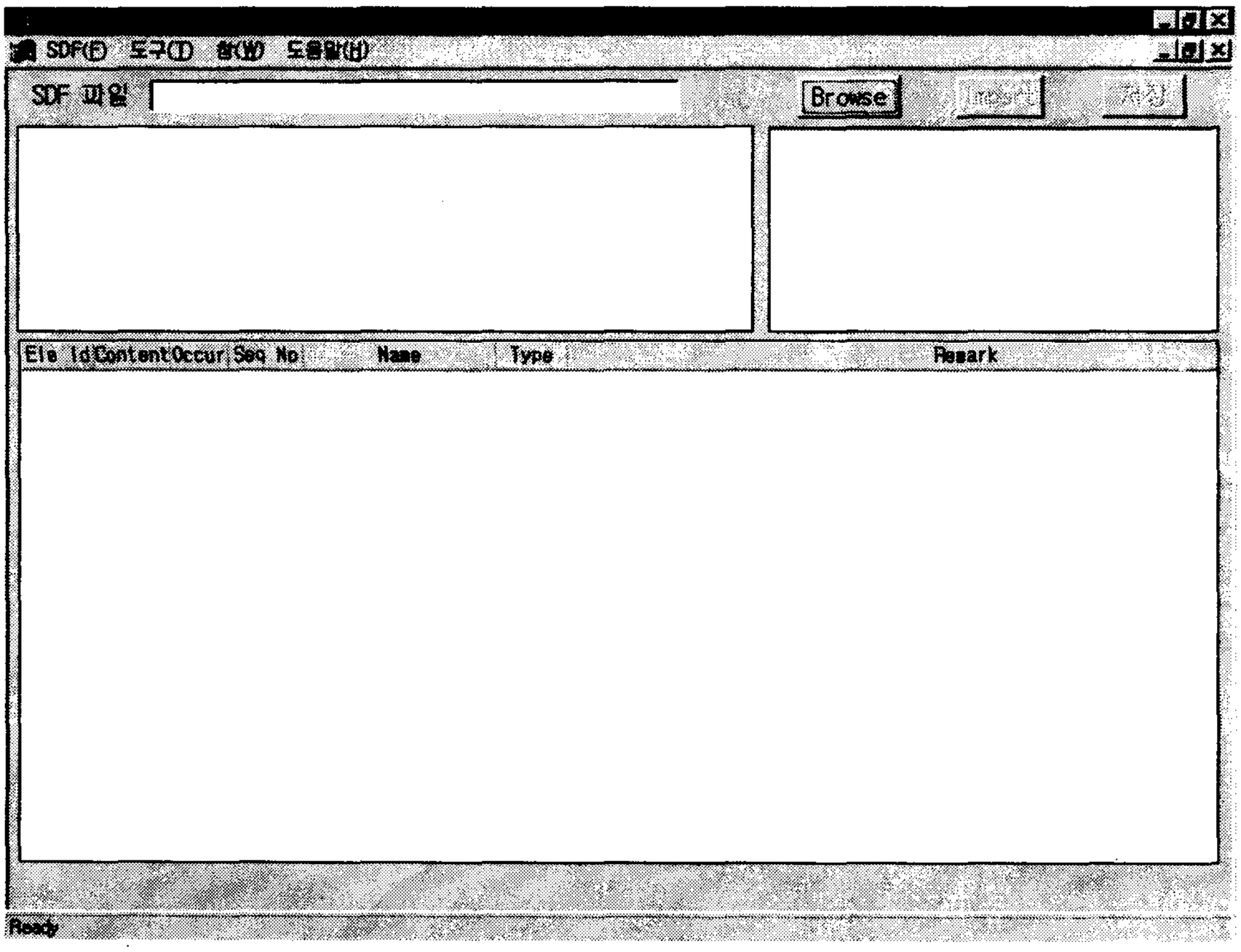
요소를 삽입하거나 추가하고자 할 경우에는 요소창으로 마우스를 옮기고 마우스의 오른쪽 버튼을 누르면 요소의 추가, 삭제 및 변경에 관한 팝업메뉴가 나타난다.

로컬요소창이나 글로벌요소창을 선택한 상태에서 각각에 해당하는 요소들을 편집할 수 있다. 각 요소창의 요소들은 RESERVED한 요소들을 포함하여 해당 SDF를 작성하면서 만든 적이 있는 요소들을 모두 보여준다. 요소이름은 해당 SDF에서 한 번밖에는 사용할 수 없다. 로컬요소를 글로벌요소로 복사를 하는 경우에는 다른 SDF를 만들면서 사용할 수 있다. 그러나 일단 사용되면 로컬요소로 복사를 하여 편집을 할 수도 있다.

(3) SDF Import

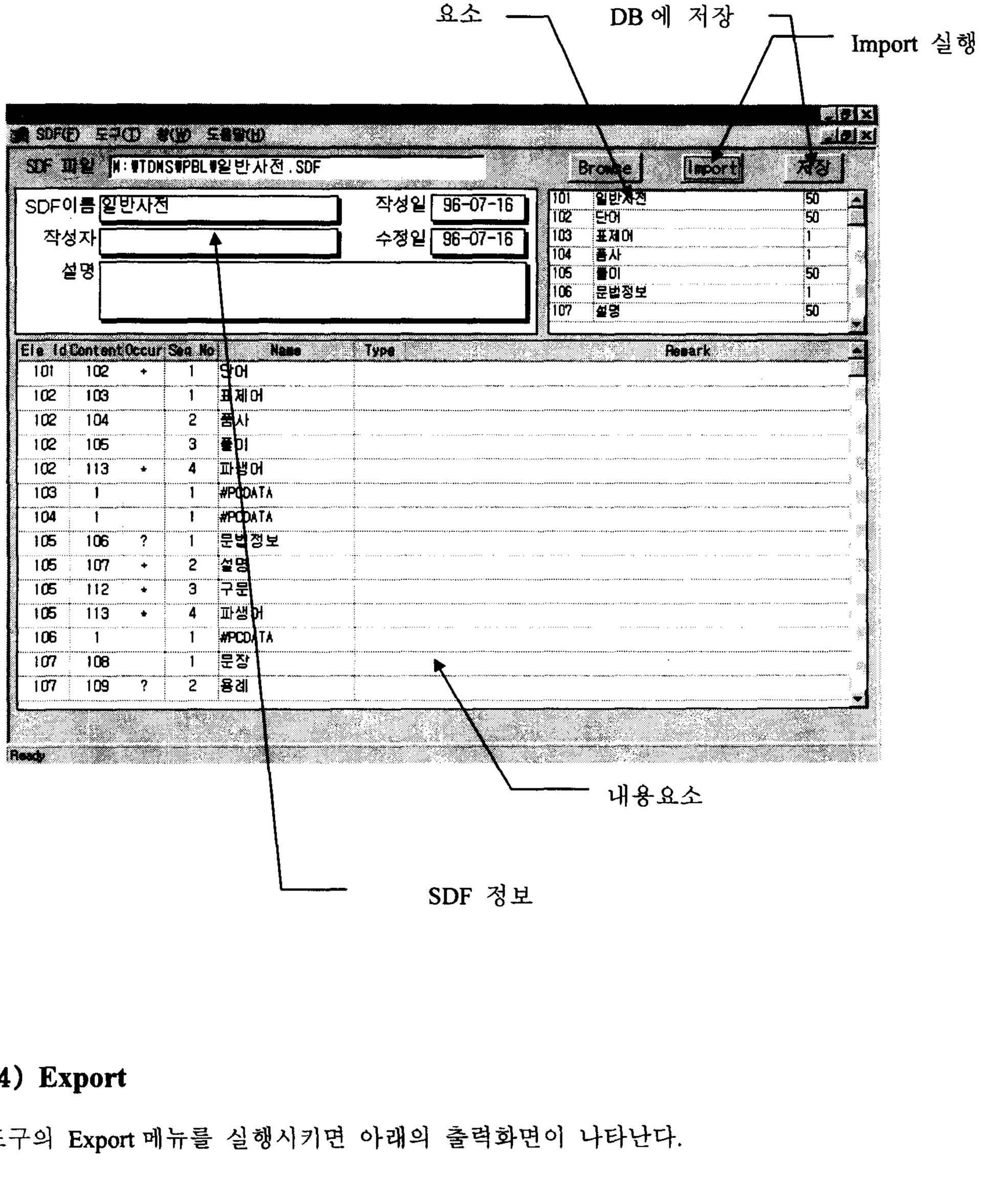
Menu 에서 도구->Import 를 선택하면 다음과 같은 Import 주 화면이 나타난다.

[그림 10] SDF Import 화면



Browse 단추를 누르면 아래와 같은 File 찾기 window 가 나타난다. 해당 화일을 선택하면 'Import' 단추가 Enable 되고 이 단추를 누르면 아래와 같이 SDF 형태로 분석된 자료가 나타나게 되며 '저장'단추를 누르면 자료가 DB 에 저장된다.

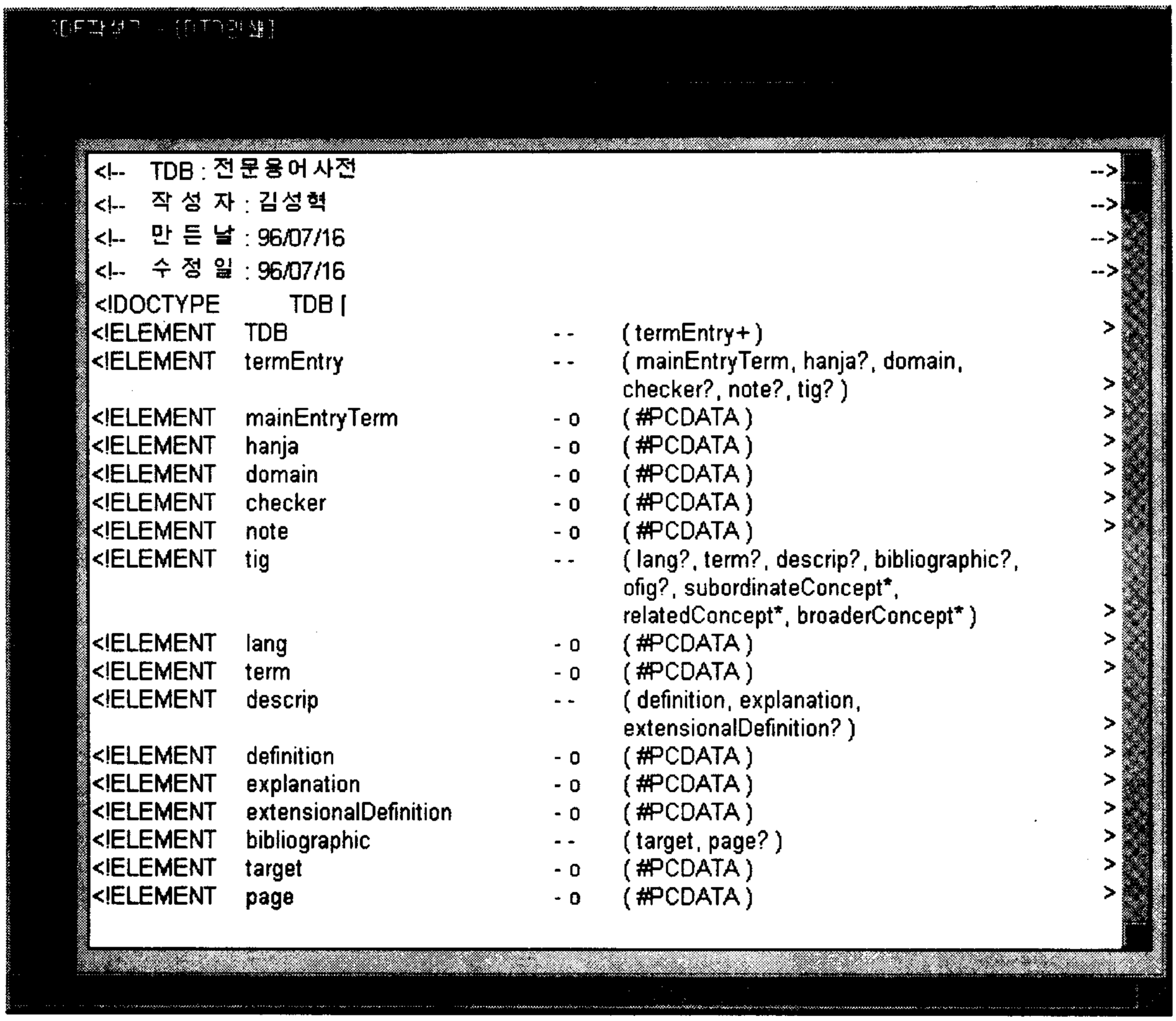
[그림 11] SDF Import 결과 화면



(4) Export

도구의 Export 메뉴를 실행시키면 아래의 출력화면이 나타난다.

[그림 12] SDF 출력화면



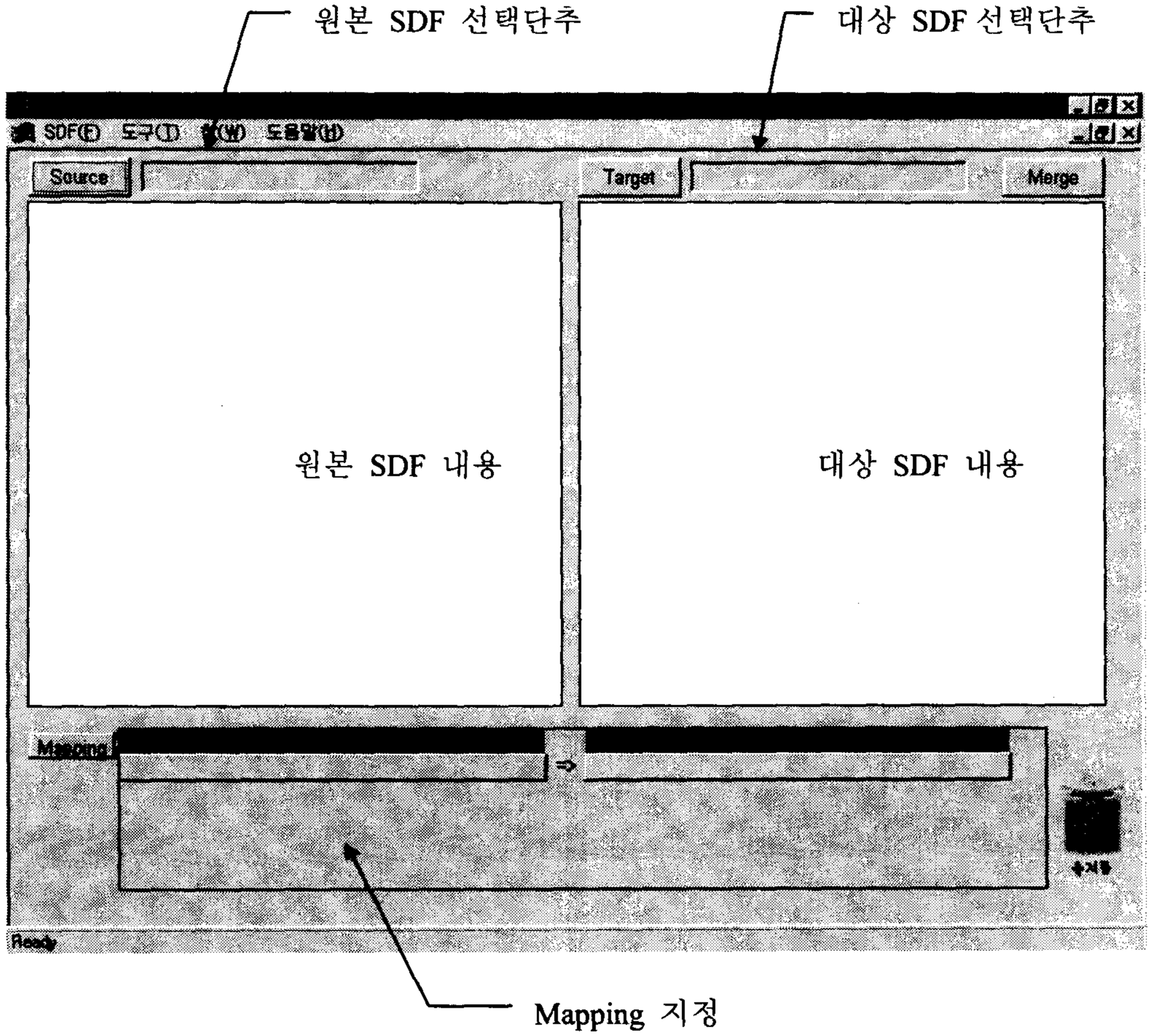
```
<!--  TDB : 전문용어사전                                                    -->
<!--  작 성 자 : 김성혁                                                     -->
<!--  만 든 날 : 96/07/16                                                   -->
<!--  수 정 일 : 96/07/16                                                   -->
<!DOCTYPE     TDB [
<!ELEMENT   TDB                  - -   ( termEntry+ )                                    >
<!ELEMENT   termEntry            - -   ( mainEntryTerm, hanja?, domain,
                                       checker?, note?, tig? )                           >
<!ELEMENT   mainEntryTerm        - o   ( #PCDATA )                                       >
<!ELEMENT   hanja                - o   ( #PCDATA )                                       >
<!ELEMENT   domain               - o   ( #PCDATA )                                       >
<!ELEMENT   checker              - o   ( #PCDATA )                                       >
<!ELEMENT   note                 - o   ( #PCDATA )                                       >
<!ELEMENT   tig                  - -   ( lang?, term?, descrip?, bibliographic?,
                                       ofig?, subordinateConcept*,
                                       relatedConcept*, broaderConcept* )                >
<!ELEMENT   lang                 - o   ( #PCDATA )                                       >
<!ELEMENT   term                 - o   ( #PCDATA )                                       >
<!ELEMENT   descrip              - -   ( definition, explanation,
                                       extensionalDefinition? )                          >
<!ELEMENT   definition           - o   ( #PCDATA )                                       >
<!ELEMENT   explanation          - o   ( #PCDATA )                                       >
<!ELEMENT   extensionalDefinition - o  ( #PCDATA )                                       >
<!ELEMENT   bibliographic        - -   ( target, page? )                                 >
<!ELEMENT   target               - o   ( #PCDATA )                                       >
<!ELEMENT   page                 - o   ( #PCDATA )                                       >
```

SDF 메뉴에서 인쇄를 선택하면 위의 화면과 같은 내용이 프린터로 출력이 된다. 저장을 선택하면 저장할 파일 이름을 묻고 위의 내용이 일반 텍스트 파일의 SDF 가 만들어 진다.

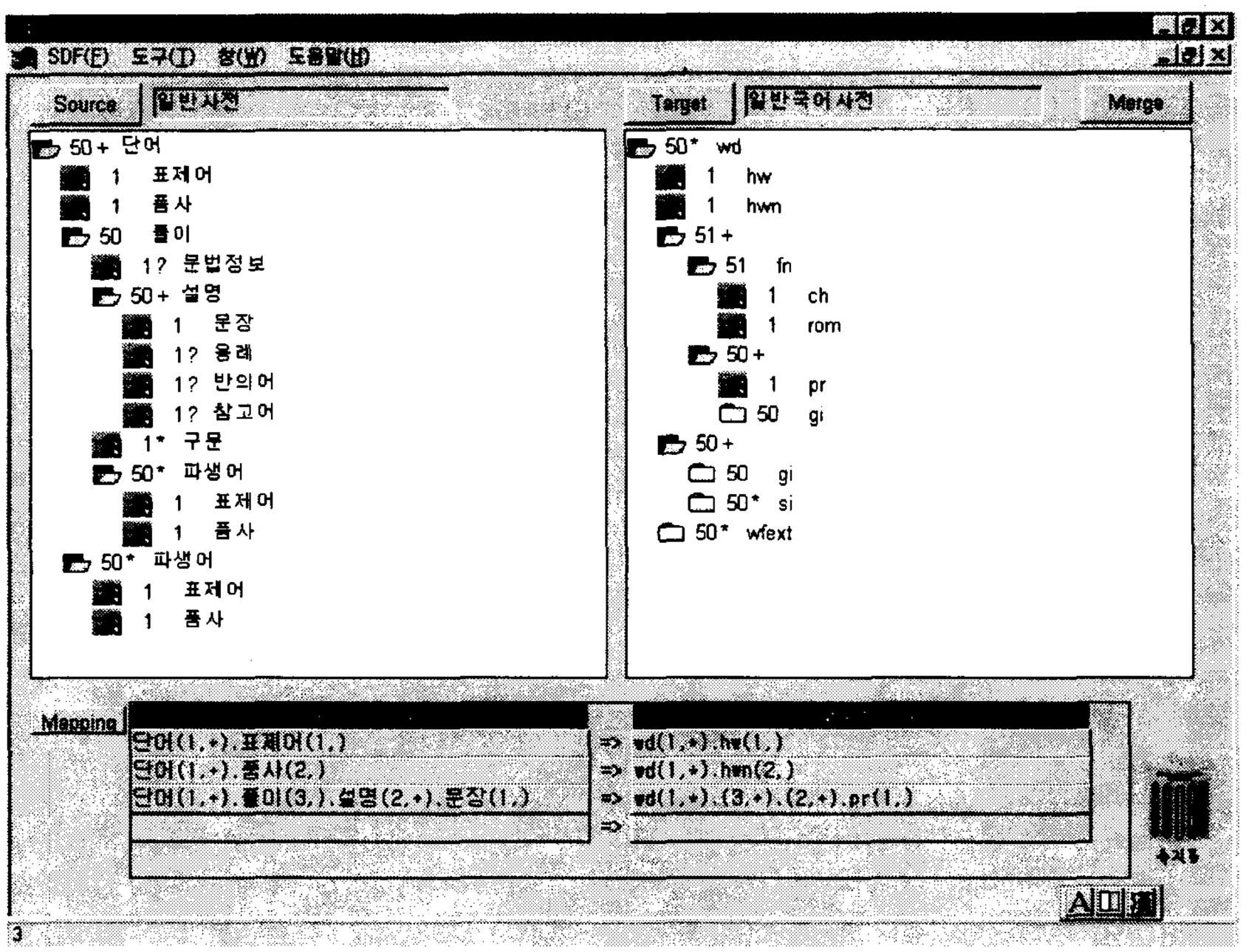
SDF 이름에 의해서 DOCTYPE 의 이름이 결정된다. MINIMIZOR 를 최대한 활용하는 방안을 강구한다. 그러나 시스템의 저장용량이 상당히 커서 데이터의 약간 커지는 것은 크게 상관이 없으므로 일단은 MINIMIZOR 를 사용하지 않는다.

(5) Merge

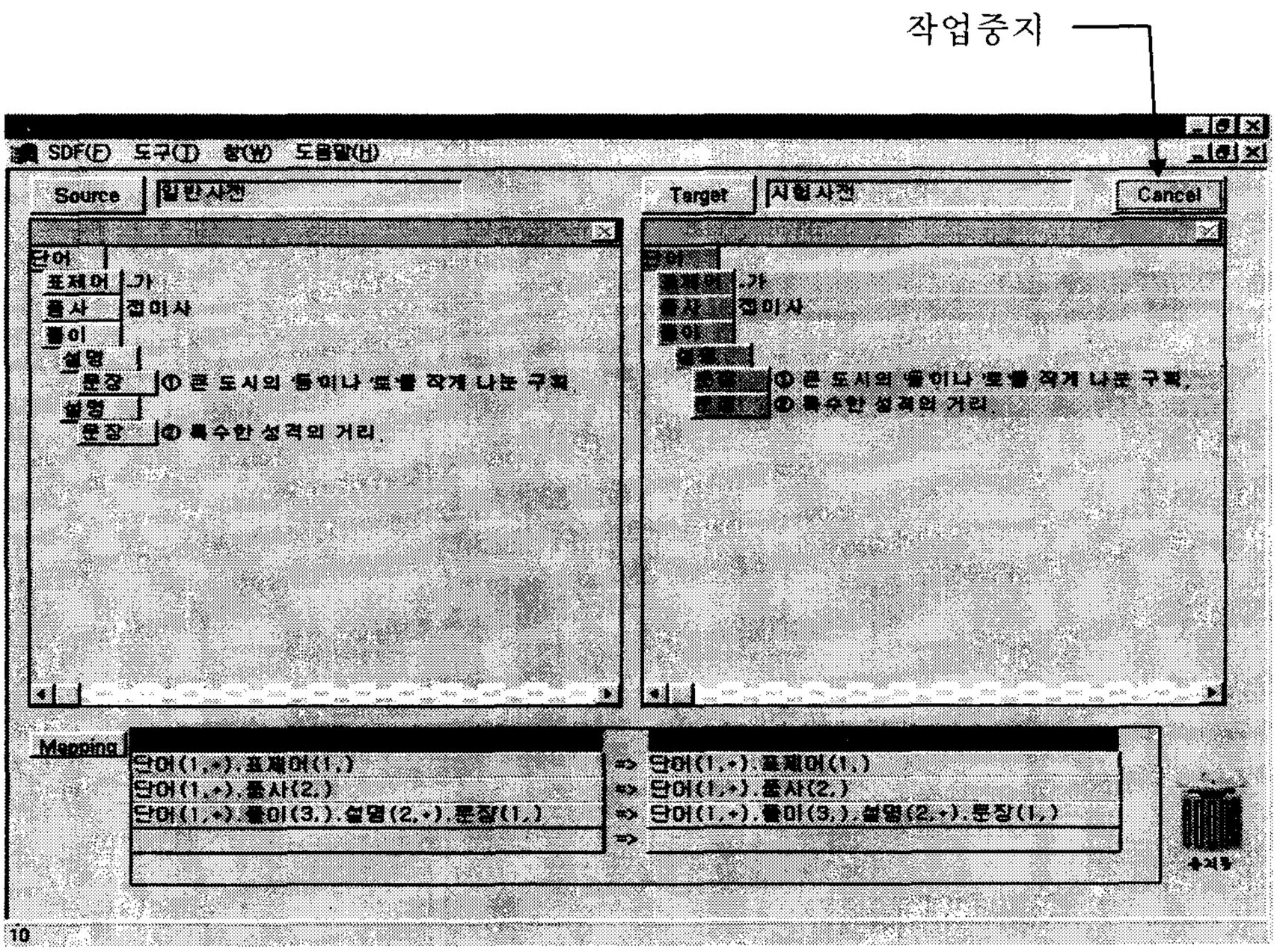
Menu 에서 도구->Merge 를 선택하면 다음과 같은 Merge 주 화면이 나타난다.



먼저 원본 SDF 와 대상 SDF 를 선택한다. 이때 원본과 대상의 SDF 이름은 서로 달라야 한다. 원본과 대상 SDF 에서 요소를 선택하고 Drag & Drop 을 이용하여 Mapping 상자에 mapping 한다. 같은 요소를 mapping 내용을 없애려면 해당 Row 를 선택한 후에 Drag 하여 '휴지통'에 Drop 한다.



Mapping 이 끝나면 'Merge' 단추를 눌러 작업을 실행한다. 작업이 시작되면서 아래의 그림이 나타난다.



실제로 작업이 되고 있는 단어를 화면에 보여준다. 작업을 취소하려면 'Cancel' 단추를 누른다.

(6) 검토사항

현재의 프로그램은 SGML 을 따르기는 하였지만 사전의 특성을 위주로 하였기 때문에 SGML 을 완전히 지원하지는 못한다. 다른 DTD Builder 에서 SDF 를 만들었을 경우에는 IMPORT 에서 PARSING 을 하면서 문제가 발생할 수 있다. 그러나 SDF Builder 를 통해서 만들어진 SDF 를 다른 DTD Builder 에서는 문제가 없게 한다

SDF Builder 에서 각 요소에 대한 속성에 대한 충분한 검토를 못했다. 속성의 형태를 몇가지로 정하여 SD Editor 에서 입력을 쉽게 할 수 있게 한다. 예를 들면 LIST BOX, RATIO BUTTON 을 이용하여 입력할 수 있도록 한다.

ENTITY 는 전혀 지원을 하지 못하고 있다. 그러나 최소한의 기능은 지원해야

한다. 왜냐하면 특수문자의 경우 입력하는 방법이 없다. 이러한 경우에는 ENTITY 로 정의하여 입력을 할 수 있도록 한다.

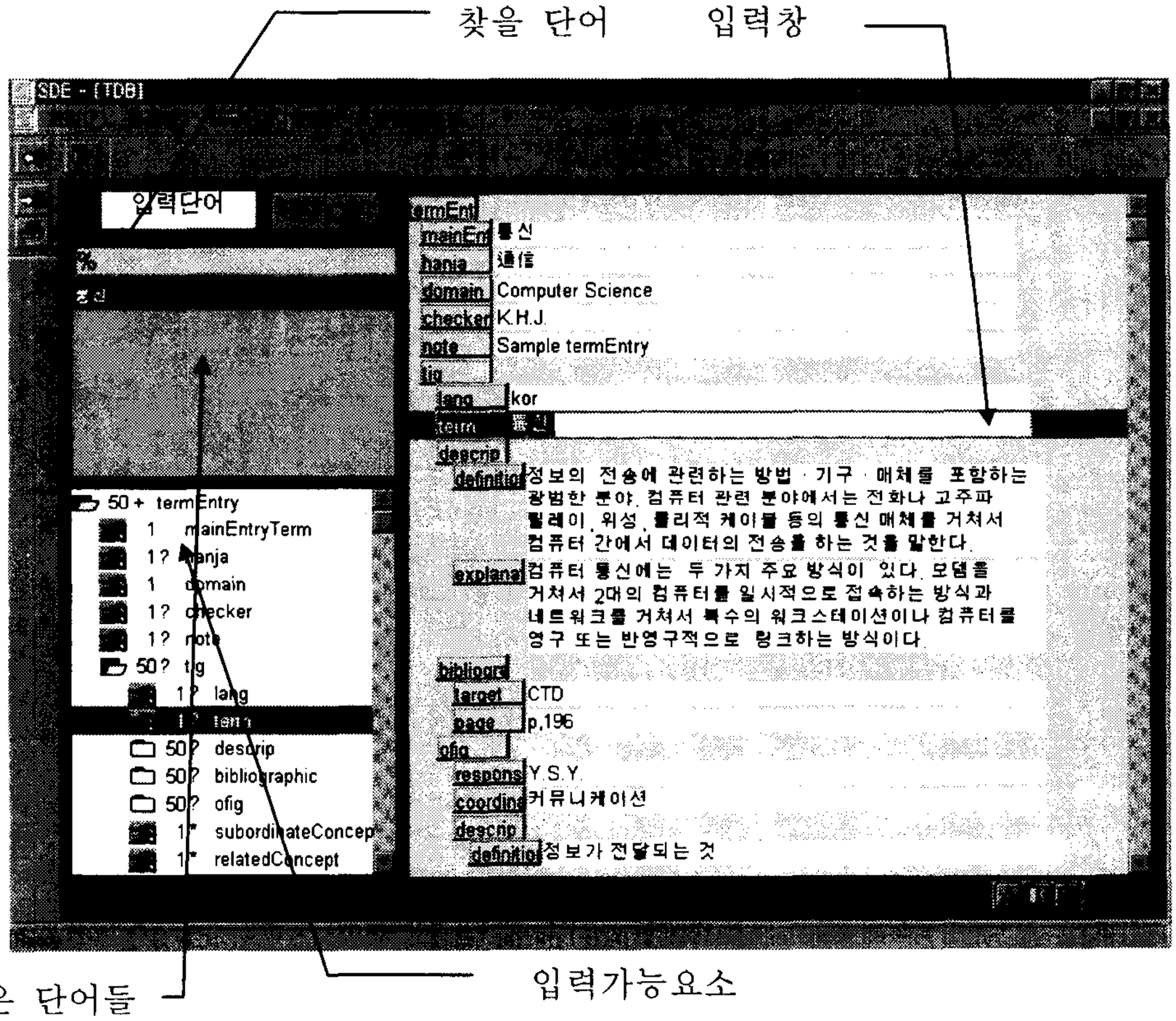
입력화면을 자유롭게 정의할 수 있도록 한다. 입력하는 데이터가 고정되지 않은 상태에서 자유롭게 정의하기에는 많은 어려움이 있을 것이다. 입력화면을 완전히 자유롭게 하는 것은 힘들고 기초적인 형태를 몇 가지 정한다. 각 컬럼의 위치와 크기, 형태와 색깔을 기억하고 있어야 한다. SDF 에는 컬럼의 형태와 크기가 어느 정도 기록이 되어 있다. 같은 SDF 에 대하여 여러 가지의 입력화면을 가질 수 있어야 한다. (편집기에서 구분과 선택을 하기 위한 방법을 생각해야 한다.) 결국은 같은 컬럼을 가지고 위치와 형태를 바꾸는 것이다.

4. SD Editor

(1) 프로그램 설명

입력할 수 있는 내용은 SDF에 이미 규정되어 있다. 따라서 SDF를 선택해야만이 프로그램을 동작시킬 수 있다. 프로그램이 시작되면 SDF를 선택하도록 한다. 만일 선택을 취소하면 프로그램을 끝내게 된다.

[그림 13] SD입력 화면



위의 그림 는 SD를 입력하는 화면으로 현재 입력되어 있는 단어를 찾아 선택하기 위한 창, 현재 입력이 가능한 요소들을 보여주는 창, 그리고 단어의 내용을

입력하는 창으로 구성을 한다.

처음으로 프로그램이 시작되면 찾을 단어가 없다. 찾을 단어를 입력한 후에 ENTER 를 입력하면 찾을 단어에 입력된 단어를 모두 찾아 보여준다. CONTROL 을 누른 상태에서 ENTER 을 입력하면 정확히 일치되는 단어들을 보여준다. 편집을 하고자 하면 찾은 단어들에서 DOUBLECLICK 을 한다. 만일 새로운 단어를 입력하고자 하면 입력 가능 요소창에서 입력단위를 선택한 후에 '삽입' 또는 '추가'버튼을 누른다. 그러면 여러 필드들 중에서 REQUIRED 필드들 만을 삽입해서 보여준다.

필드를 선택하면 해당필드에서 입력이 가능한 요소들이 바뀐다. 입력이 가능한 요소들을 결정하는 방법은 복잡하고 다양하다. 입력창에서 선택하고 있는 필드를 중심으로 입력이 가능한 요소들을 결정해야 한다. 요소창에서는 현재 입력창에서 선택하고 있는 필드와 같은 레벨의 요소들과 직계상위 요소들을 보여준다. 즉, 삽입이나 추가는 현재의 필드가 속한 직계요소들만이 되어질 수 있다는 말이다. 가능요소창에서 required 한 요소인 경우에는 선택을 할 수가 없다.

삽입을 하고자 하는 요소를 가능요소창에서 선택하면 현재 필드가 속한 선택된 요소를 밑으로 밀고 사이에 새로운 요소를 넣는다는 뜻이다. 그러나 단어와 같이 입력단위를 선택하는 경우에는 현재의 입력창에 있는 내용을 저장한 후에 새롭게 시작하게 된다.

추가의 경우에는 선택한 요소를 찾아 맨 마지막에 추가로 넣는다는 뜻이다. 한 번도 입력된 적이 없는 요소인 경우에는 삽입과 추가가 같은 의미를 같게 된다.

요소를 삽입 또는 추가하는 경우에는 하위의 요소들 중에서 required 한 요소를 찾아 요소를 미리 삽입을 시켜야 한다. 삽입을 시키는 방법

삽입할 요소는 optional 과 repeatable 에 의해서 입력이 되어있는 요소일 수도 있고 없는 요소일 수도 있다. 현재와 같은 레벨을 선택한 경우에는 없는 경우도 있을 수 있으나 상위의 레벨인 경우에는 반드시 있어야 한다. 따라서 찾는 방법도 달라질 수밖에 없다. 상위레벨이면 현재의 위치에서 거꾸로 찾으면 된다. 같은 레벨이면 찾는 방법을 여러 가지로 검토해야 한다. 같은 레벨인 경우에는 현재 선택된 필드를 기준으로 한다. 현재 선택된 필드와 같은 요소인 경우에는 삽입은 현재의 위치에 한다. 그러나 추가는 현재 선택된 요소의 맨 끝으로 한다. 선택

된 필드의 필드의 요소가 아닌 경우에는 의 상위요소를 찾고 그 상위요소로 부터 선택된 요소를 찾는다. ocurrence indicator 에 따라서 나타나는 양상이 달라질 수 있다. 경우에 따라서는 아예 없을 수도 있다. 삽입은 맨 처음을 의미하고 추가는 맨 뒤를 의미한다.

삭제를 하는 경우에는 현재 입력창에서 선택된 요소가 속한 요소를 모두 삭제한다. 그러나 삭제하기 전에 미리 삭제여부를 물어보아야 한다. 저장을 하지 않으면 원래의 데이터를 갖게 된다. required 데이터는 삭제할 수가 없다.

입력화면이 나오면 해당하는 필드를 입력한다. 그리고 추가하거나 삭제해야 할 사항은 미리 명령버튼 등을 통해서 할 수 있도록 한다.

입력이 가능한 글자는 윈도우에서 지원되는 글자를 원칙으로 한다. 만일, 특수한 문자를 입력하고자 하는 경우에는 SDF 에서 미리 ENTITY 로서 정의해 놓고 이를 이용해서 입력을 하도록 한다.

한 단어에 대한 설명의 길이는 이론 상으로 무한대까지 해야 하지만 시스템의 한계에 의해서 결정된다. SQL Server 6.0 에서는 240 까지로 제한된다. 이 보다 큰 데이터를 입력해야 하는 경우에는 다른 테이블을 이용해서 DB 에 입력되어야 할 것이다. 앞으로 이를 극복하는 방안에 관한 연구를 해야 할 것이다. 물론 사용자는 이를 인식하지 못하도록 해야 한다.

SDE 의 메뉴는 일반편집기에 준하여 한다.

(2) 검토사항

처음 프로그램이 시작되면 기존에 열려 있던 SDF 를 선택한 상태가 되어야 한다. 그러나 입력하는 사람을 구분하는 방법은 어떻게 할 것인가? 테이블이 하나인 경우에는 그러한 구분을 하기 위한 컬럼을 가지고 있어야 한다. 입력한 날짜와 사람, 그리고 SDF 를 가지고 있는 테이블을 하나 만들고 해당 ID 를 가지고 있도록 한다. 다시 말해서 입력하는 사람을 등록하는 방법이 있어야 한다. 구분이 필요 없을 경우에는 그냥한다. 처음 로고화면에서 입력하는 사람의 ID 와 날짜를 확인하도록 한다. 입력하는 사람에 대한 비밀번호를 확인하도록 해야 할까?

이 전에 입력한 사항을 편집하고자 할 경우에 찾는 방법을 어떻게 할 것인가? 입력한 사람의 ID나 날짜를 이용하는 것은 물론이고 표제어를 이용하여 찾도록 한다. 내용을 이용하는 것은 고려해 본다.

데이터를 입력하면서 다른 데이터를 복사해 올 수는 없을까?

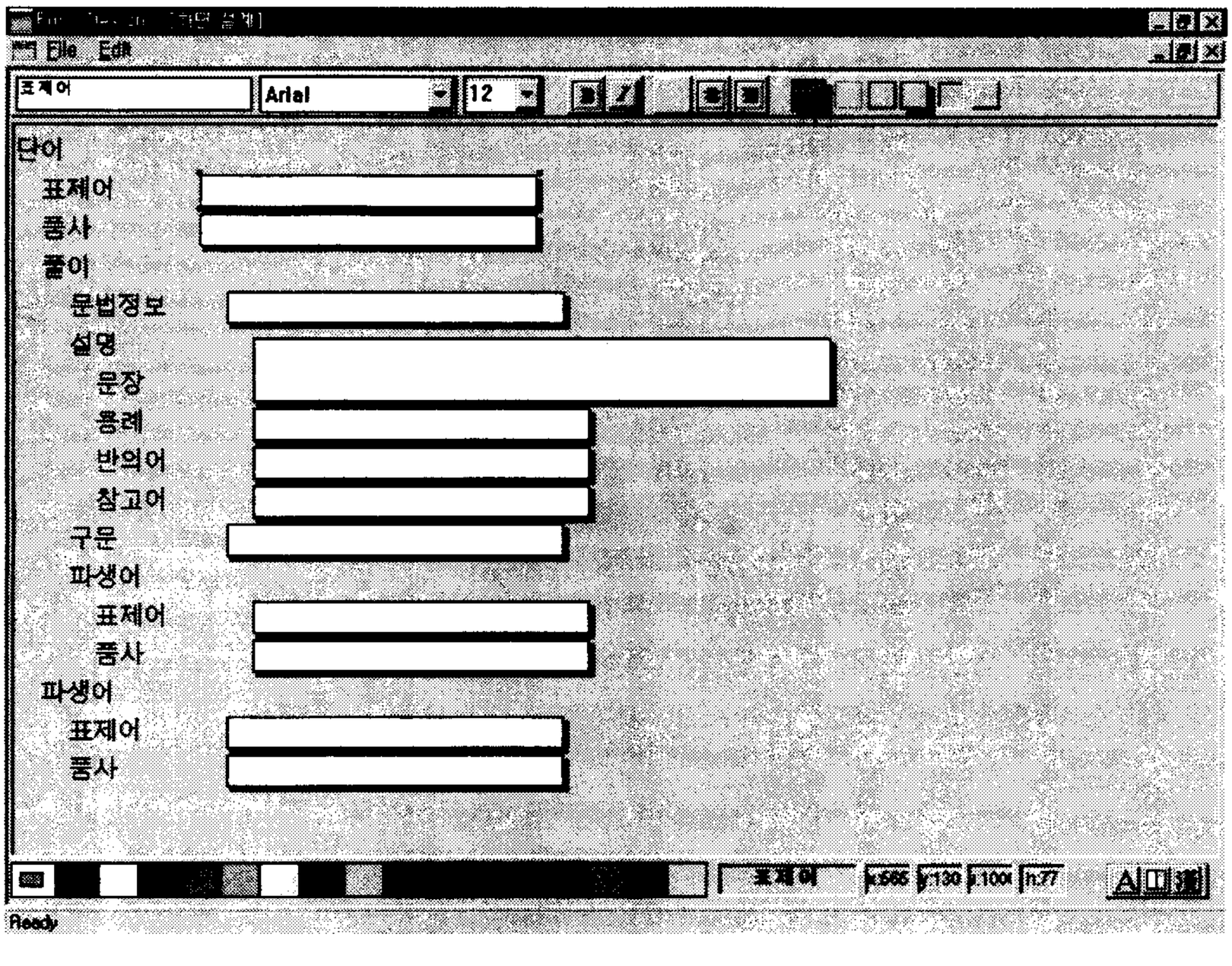
여러 사람이 입력을 하면 데이터가 중복이 될 수도 있다. 이를 어떠한 방법으로 구분할 것인가? 입력하는 사람의 ID를 구분하면 될까? 결국은 사람의 눈으로 찾아 가면서 하나 하나 선택을 해야 한다.

입력화면을 사용자가 마음대로 변경을 하게 하는 방법을 어떻게 설계할 것인가?

5. Form Designer

전자사전에서 SD 입력 FORM을 편집하기 위한 프로그램이다.

[그림 14] SD입력 화면

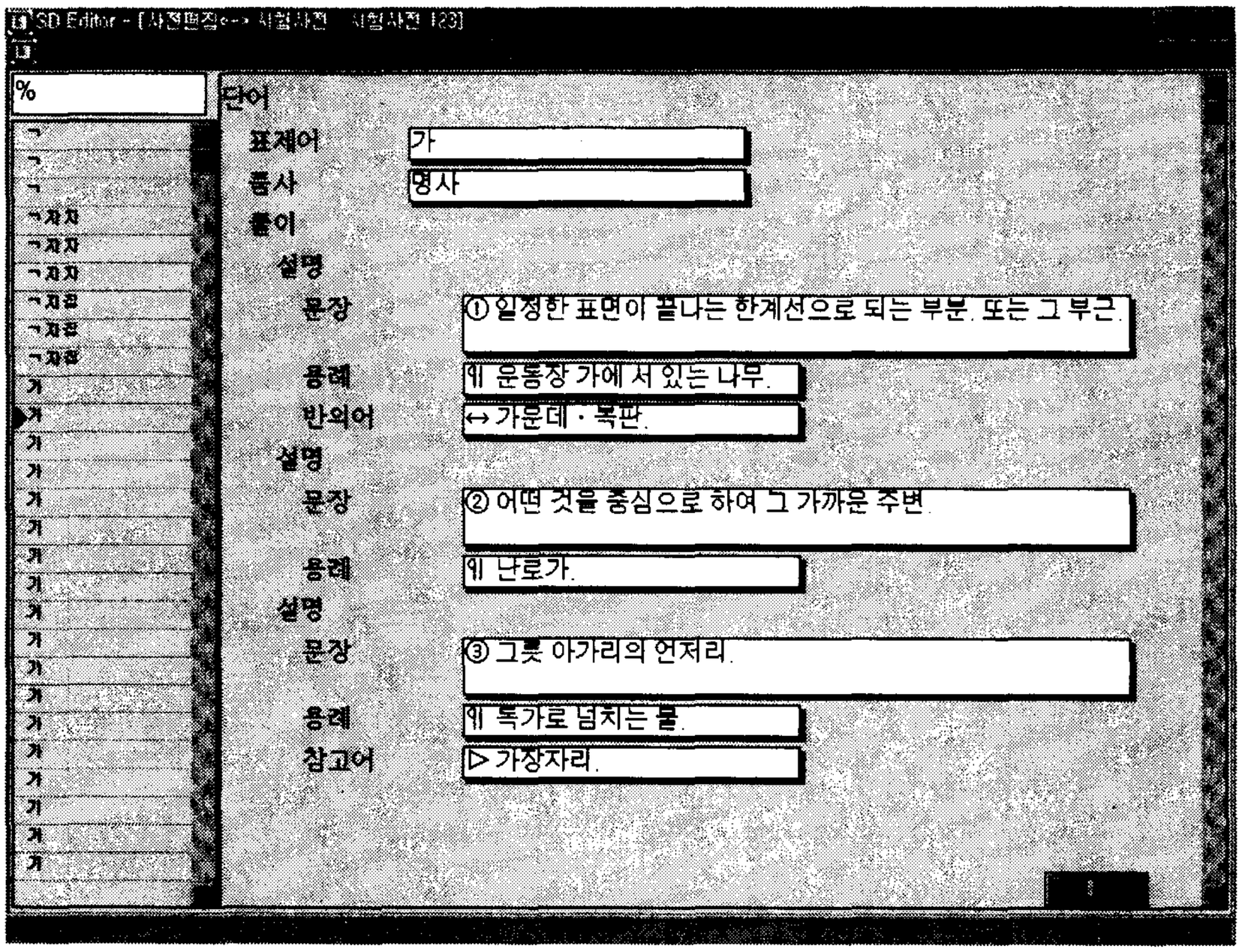


Alignment 의 지정은 상, 하, 좌, 우로의 Position 지정과 높이와 넓이를 정의할 수 있다. 또한 입력 순서를 사용자 임의대로 지정하도록 하였으며 Edit style 를 사용자의 임의대로 정의할 수 있도록 하여 User Interface 의 능률을 꾀하고자 하였다. EditStyle 은 Edit, ListBox, CheckBox, Radio Button 을 설정 할 수 있도록 하였다.

6 . SD Browser

전자사전에서 필요한 내용을 찾아보는 프로그램이다.

[그림 15] SD Browser



SD를 선택해서 필요한 단어를 찾아 볼 수 있다. 필요하면 전체 SD를 찾아 볼 수도 있다. 조건에 맞는 단어를 찾아 볼 수도 있게 한다.

찾아보는 방법은 다양하게 한다. 원하는 단어를 입력해서 찾거나 GRID 화면을 이용해서 찾는다. 찾는 단어가 하나인 경우에는 바로 보여 주고 그렇지 않으면 브라우징화면을 보여 준다. 설명 등에 있는 단어를 찾아보고 싶을 경우에는 블럭을 표시한 후에 특정한 키를 치면 되게 한다. HYPERTEXT 기능을 지원하게 한다. 그러나 입력하면서 이에 대한 정보를 추가해야 한다. 검색한 단어의 HISTORY를 관리할 수 있으면 좋을 것이다.

실제로 눈에 보이는 방법은 입력화면과 같은 형식으로 보여 준다. 사전의 형식을 이용할 수도 있겠지만 SDF와 그에 해당하는 사전의 모양을 미리 정의해 놓아야 하는 번거로움이 있다. 브라우징 화면을 구성하는 방법은 키단어와 품사(내지는 문서의 분류)만을 보여 준다. SDF에서 필요한 항목만을 선택해서 볼 수도 있다.

7. SD Encoder/Decoder

SDI를 SD로 출력하는 것은 큰 어려움이 있다. SDF에서 정의된 대로 출력을 하면 그만이기 때문이다. 그러나 SGML 형식의 문서를 읽어들여 DB에 저장을 하는 것은 쉽지 않다.

먼저 SDF Builder에서 SDF를 IMPORT한다. SDF를 IMPORT를 하는데 있어 가장 큰 문제점은 SGML을 완전히 지원하지 못하는데 있다. 따라서 이 점을 꾸준히 보완을 하여야 할 것이다.

만일, SDF 파일이 없다면 SD를 읽어서 SDF를 생성해 낼 수도 있어야 한다. SD를 통해서 SDF를 생성해 내려면 많은 어려움이 있을 것이다. SDF를 생성해 내려면 MINIMIZOR가 OMMITED로 되어 있는 경우에 주의하여야 한다. 현재 등록이 되어 있지 않은 요소가 있다면 요소의 타입을 찾아내어 등록을 하여야 한다. 속성에 관한 사항은 알 수가 없다. ENTITY이 경우에는 처리하는 방법이 거의 없다. SD를 통해서 SDF를 생성하는 프로그램은 SDF를 IMPORT하는 것보다 많은 문제점을 극복해야 할 것이다.

SD를 IMPORT할 때 이미 SD가 존재하고 같은 의미를 갖는 단어가 있을 경우에는 어떻게 처리할 것인가? 다른 SDF인데 이름이 같게 된 경우에는 어떻게 처리할 것인가? 다른 이름의 SDF를 만들고 나주에 MERGE기능을 이용해서 편집할 수밖에 없을 것이다.

SD가 만들어지면 이를 읽어 들여 다른 응용프로그램에서 필요한 데이터 형태로 만들어야 하고 반대로도 해야 한다. 이를 위해서는 응용프로그램에서 사용하고 있는 데이터 형태에 관한 정보를 정확하게 파악하고 서로 협조를 해야 한다. 전자사전의 경우에는 데이터 형태를 파악하여 SDF를 만들고 이에 맞게 SD를 만드는 작업을 하기가 용이할 것이다.

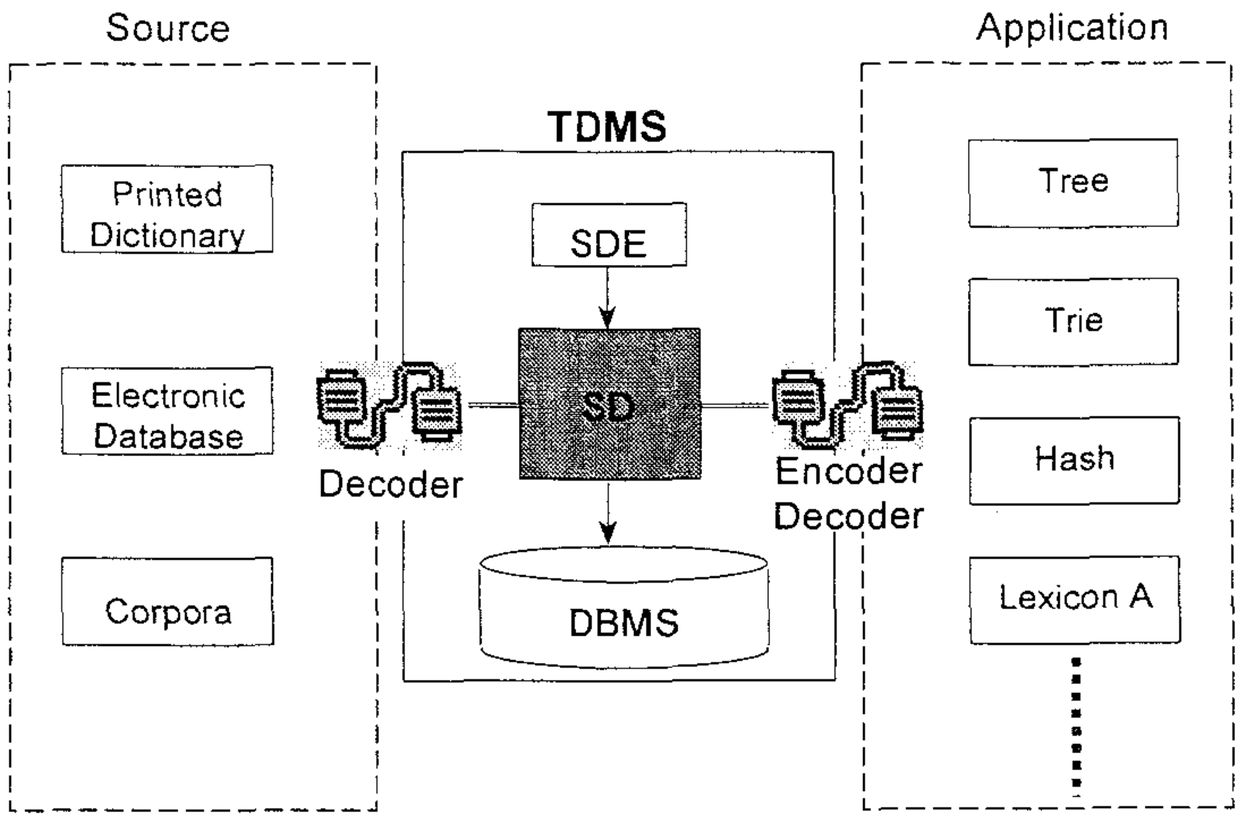
8장. TDMS를 이용한 사전 구축

컴퓨터의 발달과 더불어, 최근 자연언어처리 분야의 일부에서는 문서들을 전자 문서화 하려는 노력이 이루어지고 있다. 이를 위해서는 정형화되어있지 않은 일반 문서들을 정형화해야 하는데, 이에 대한 연구가 최근 활발히 진행되고 있다. 이와

관련된 연구 중 대표적인 것으로 사전을 전자 문서화된 형태로 바꾸는 작업을 들 수 있는데, 외국에서는 이미 10여년 전부터 이에 관한 연구가 활발히 진행되어 왔다. 이에 반해 우리나라에는 아직 이에 견줄만한, 나아가 표준화할 만한 전자 사전이 아직 개발되어 있지 않은 상황이다.

외국에서처럼 기계 가독 형태로 되어 있는 사전을 다시 파싱하여, 이를 정규화된 형태로 바꾸는 단계는 아직 생각도 하지 못하고 있는 실정이며, 또한 기계 가독형 사전이 어떠한 형태의 내부 구조를 지녀야 하는 지에 대한 연구도 아직 준비 단계이다.전산 처리용 사전으로는 개개의 연구 개발자가 연구 목적과 시스템에 따라 각각 전산 처리용 사전을 개발하고 있지만, 이러한 사전 개발 작업에는 많은 시간과 비용이 든다. 또한 하나의 목적으로 개발된 사전이 있다 하더라도, 그 사전을 다른 목적에 이용하는 것도 그리 간단하지는 않다.

이러한 문제점을 극복하기 위해 여기서는 일반 국어사전의 내용을 인코딩할 수 있는 표준 사전 형태의 내부 구조를 제안하고 일반 사전을 이에 맞게 정형화 하고자 한다. 만일 우리가 표준 사전 형태로 저장된 어휘 정보를 가지고 있다면, 우리는 이것을 바탕으로 새로운 종류의 기계 가독형 사전이나 전산 처리용 사전으로 가공할 수 있다. 재사용 (reuse)이 가능하도록 어휘 정보를 표시하는 일이 바로 일반 사전의 구조를 표준화하는 주된 목표의 하나이다.([그림 16] 참조) 어휘 정보를 재사용한다는 의미는 두 가지 면에서 생각할 수 있다. 즉, 이미 존재하는 어휘 정보를 새로운 형태로 변환하는 것 - 예를 들어 인쇄된 사전의 어휘 정보를 기계 가독 형태로 바꾸는 것- 과, 다른 응용 프로그램에 맞게 기존의 어휘 정보를 표준 사전에서 추출하는 일을 생각할 수 있다.



[그림 16] 표준화된 전자 사전의 어휘 정보 구축과 재사용

1 절. 전자 사전을 위한 일반 사전의 DTD

개개의 사전 표제어들은 구조화된 형태를 지니고 있는데, 이러한 형태는 사전마다 매우 다양하게 나타난다. 다른 사전이 다른 구조를 지니는 것은 극히 자연스러운 일이지만, 같은 사전에서조차도 다양한 형태의 구조가 종종 나타난다. 이처럼 각 어휘 항목들의 구조가 사전마다 다르고, 또한 한 사전에서도 매우 다양하게 나타나기 때문에, 일반 사전을 정형화하기 위해서는 모든 사전에 응용될 수 있는 공통적 구조를 찾아내어 인코딩하는데 중점을 두어야 한다. 여기서 주의해야 할 문제는 우리가 중요한 정보들을 효과적으로 다시 추출하기 위해 사전의 어휘 정보들을 컴퓨터에 어떠한 방식으로 코드화 하느냐는 것이다.

사전의 DTD 작성을 위해서는 사전 표제어의 설명에 나타나는 내용을 최소한의 요소로 나누어 이것을 표준 사전의 형식에 맞게 개개의 데이터 자리에 저장한다. 사전의 어휘 정보를 세분화하면 할수록 우리는 세분화된 정보를 더 잘 사용할 수가 있다.

일반 사전에 포함되는 정보들을 나타내기 위해서는 다음과 같은 내용의 정보들

이 주어져야 한다. 우선 최상위 요소로 표제어의 형태에 관한 정보가 있어야 한다. 여기에는 발음, 정서법, 외래어 표기에 관한 정보가 속한다. 이외에 동일한 계층에서 문법, 의미, 조어 (파생어)에 관한 정보가 주어지며, 이들은 각각 하위 계층에 더 세분화된 정보를 지닌다. 이를 트리 형태로 나타내면 [그림 17]에서와 같은 계층 구조적인 특성을 볼 수 있다.

```
단어+     표제어(정서법)
          표제어 번호
          발음 (일본어 발음)+
          외래어 표기+
                - 한자 (일본어) 표기
                - 로마자 표기
          문법  - 품사
                - 활용형
                - 어원
                      한국어 어원
                       외래어 어원
                - 구문정보
          의미+ - 의미번호
                - 전문어 표시+
                - 풀이
                - 용례 정보*
                      용례
                      용례출처
                - 속담 정보*
                      속담
                      속담풀이
                - 숙어 정보*
                      숙어
                      숙어풀이
                - 대응표제어
                  대응 표제어 관계
          조어  - 표제어(파생어)
                - 품사
```

[그림 17] 어휘 정보의 계층 구조

그리고 이러한 트리 형태를 바탕으로 표준 사전 형식을 SDML DTD의 body부분을 재정의하면 [그림 18]과 같다.

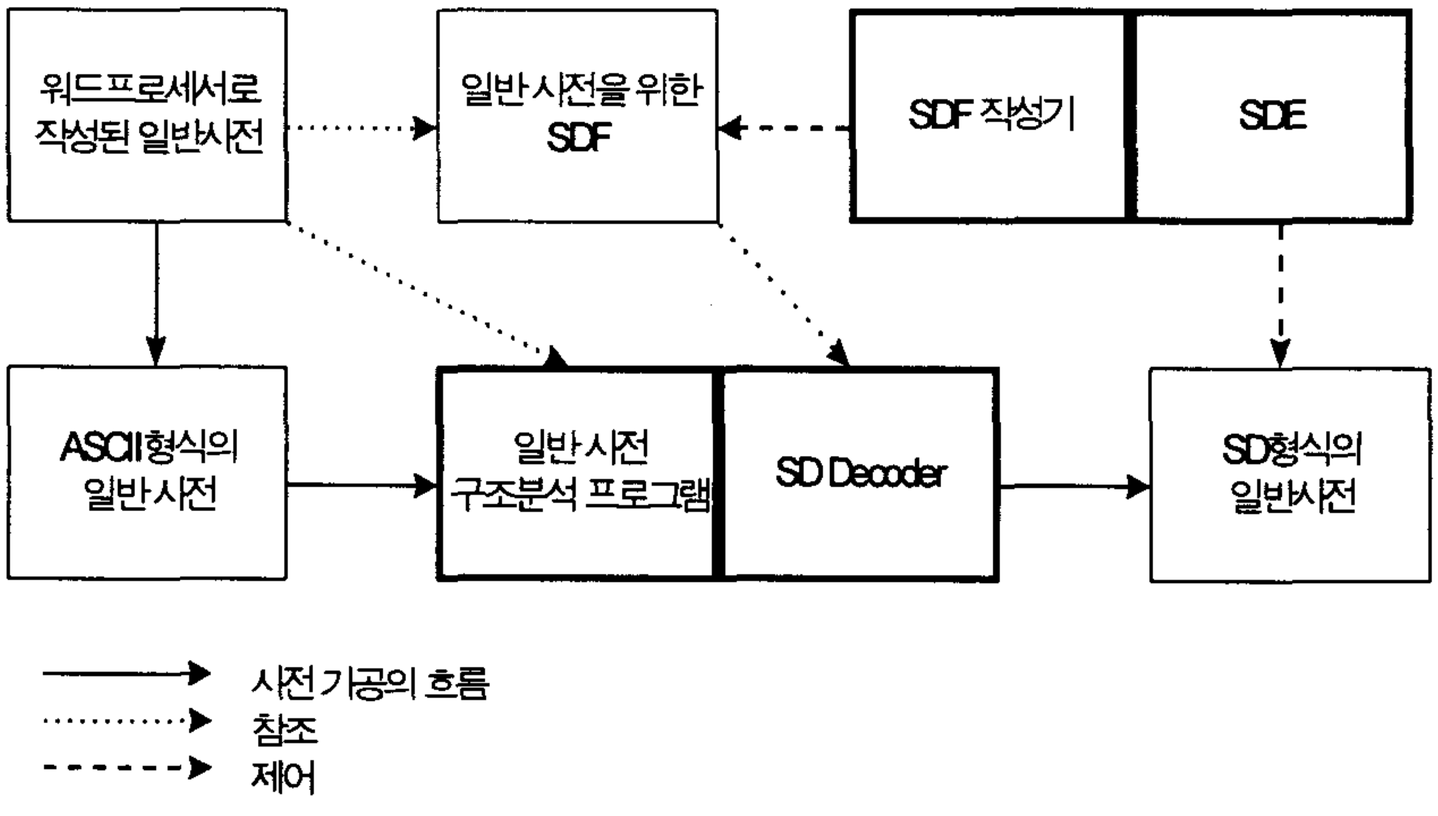
```
<!DOCTYPE sd          SYSTEM          "sdml.dtd" [
<!ELEMENT body        - - (표제어번호, (외래어표기 | 발음)+,
                                  (문법정보, 의미정보+)+, 파생어*)  >
<!ELEMENT 표제어번호   - O (#PCDATA)                                >
<!ELEMENT 외래어표기   - - (한자표기| 로마자 표기)                  >
<!ELEMENT 한자표기     - - (#PCDATA)                               >
<!ELEMENT 로마자표기   - - (#PCDATA)                               >
<!ELEMENT 발음         - - (#PCDATA)                               >
<!ELEMENT 문법정보     - - (품사, 활용형정보? , 어원정보?)         >
<!ELEMENT 품사         - - (#PCDATA)                               >
<!ELEMENT 활용형정보   - O (#PCDATA)                               >
<!ELEMENT 어원정보     - - (고유어어원| 외래어어원)                 >
<!ELEMENT 고유어어원   - O (#PCDATA)                               >
<!ELEMENT 외래어어원   - O (#PCDATA)                               >
<!ELEMENT 의미정보     - - (의미번호, 전문분야*, 풀이, 용례정보* ,
                                  속담정보*, 숙어정보*, 관계어*)   >
<!ELEMENT 의미번호     - O (#PCDATA)                               >
<!ELEMENT 전문분야     - - (#PCDATA)                               >
<!ELEMENT 풀이         - - (#PCDATA)                               >
<!ELEMENT 용례정보     - - (용례, 용례출처?)                       >
<!ELEMENT 용례         - O (#PCDATA)                               >
<!ELEMENT 용례출처     - O (#PCDATA)                               >
<!ELEMENT 속담정보     - - (속담, 속담풀이?)                       >
<!ELEMENT 속담         - O (#PCDATA)                               >
<!ELEMENT 속담풀이     - O (#PCDATA)                               >
<!ELEMENT 숙어정보     - - (숙어, 숙어풀이)                        >
<!ELEMENT 숙어         - O (#PCDATA)                               >
```

```
<!ELEMENT 숙어풀이      - O (#PCDATA)                          >
<!ELEMENT 관계어        - - (표제어, 관계어표지)               >
<!ELEMENT 관계어표지    - O (#PCDATA)                          >
<!ELEMENT 파생어        - - (표제어, (품사, 풀이?))+           >
<!ELEMENT 풀이          - O (#PCDATA)                          >
] >
```

[그림 18] 일반 사전을 위한 표준사전(SD) 예

2 절. 일반 사전의 표준 사전(SD) 가공

국어 사전이나 영한 사전과 같은 일반 사전으로부터 자연언어처리 시스템에 필요한 사전을 가공 하기 위해서는 일반 사전을 좀더 정형화된 형식으로 변환하여야 한다. 다음은 일반 사전을 표준 사전으로 가공하는 과정이다.



[그림 19] 일반사전의 가공 과정

1. 워드프로세서로 작성된 일반 사전 -- 최근 출판된 사전은 거의 워드프로세서나 출판을 위한 특별한 프로그램에 의해 만들어 진다. (예: [그림 20] 참조)
2. 문서의 형식에 대한 정보를 잃어 버리지 않는 범위내에서 ASCII 파일 형식으로 저장한다. 이때 문서의 구조나 글자체에 대한 정보를 잃어버리지 않기 위

해서는 extended style tag 와 같은 형식으로 저장하는 것이 좋다.

3. 사전의 내용을 표현할 수 있는 SDF 를 결정하고 SDE 에 있는 SDF 작성기로 SDF 를 만든다. ([그림 18]에 나타나 있는 DTD 형식으로 정의)
4. 사전의 구조를 분석할 수 있는 일반 사전 구조 분석기를 만든다.
5. 일반 사전을 구조분석하고 표준 사전 형식으로 변환하여 저장한다. ([그림 21] 참조)
6. SDE 로 잃어버린 정보를 추가하거나 필요한 가공을 한다.

다음은 일반 국어 사전에 나타나 있는 표제어에 대한 설명이다.

가난 【명】【하|형】 (←간난) 살림살이가 넉넉하지 못함. 빈곤. ¶ ~한 집 / ~에 쪼들리다. / ~에서 벗어나다. **[가난 구제는 나라도 못한다]**. 가난한 사람을 구제하기는 끝이 없어 개인은 물론 나라의 힘으로도 어렵다. **[가난이 원수]** 가난하기 때문에 고통을 받게되니 가난이 원수같이 느껴진다. **[가난한 집 제사 돌아오듯]** 치르기 힘든 일이 자주 닥침을 비유하는 말. **가난(이) 들다.** 【구】 ① 가난하게 되다. ② 쓸만한 것이 드물어 구하기 어렵다.

가다 【자】【거라불】 ① 목적한 곳을 향하여 움직이다. ¶ 학교에~/ 서울을~. ② (어떠한 목적을 위하여)떠나다. ¶ 사냥을~/ 유학을~. ③ 도달하다. ¶ 내가 한발 먼저 갔다. ④ 몸담아 있던 곳에서 다른 곳으로 옮기다. ¶ 군대에~. ⑤ 앞으로 나아가다. 걷다. ¶ 들길을~. ⑥ 전달되다. ¶ 너에게 소식이 갔느냐? ⑦ 시일이 경과하다. ¶ 어느새 여름이 가고 날씨가 서늘해졌다. ⑧ 지속되다. ¶ 그의 결심은 오래가지 못할 것이다. ⑨ (맛이나 음식이)변하다. ¶ 이 음식은 맛이 갔다. ⑩ 마음이 어떤 상태로 되다. ¶ 이해가~/ 애정이~. ⑪ 어떤 일에 힘이 쓰이다. ¶ 장을 담그는 데는 손이 많이 간다. ⑫ (금이나 흠 따위가)생기다. ¶ 이 항아리에 금이 갔구나. ⑬ 죽다. ¶ 꽃다운 나이로 ~니. ⑭ 전깃불·수돗물 따위가)꺼지거나 중단되다. ¶ 책을 읽는데 감자기 불이 갔다. ⑮ (값이나 무게 또는 차례 따위가)어느 정도에 이르다. ¶ 이 보석은 값이 상당히 간다./ 첫째 가는 기술자. (어떤 상태나 일이)없어지다. ¶ 포도는 이제 한물 갔다./ 아름다운 꿈은 모두 가 버렸다. **[가는 날이 장날]** 우연히 갔다가 뜻하지 아니한 일을 공교롭게 당함을 비유한 말. **[가는 말이 고와야 오는 말이 곱다.]** 남에게 말이나 행동을 좋게 하여야 자기에게도 좋은 반응이 돌아 온다.

[그림20] 일반 국어 사전에 나타나 있는 표제어

위의 표제어에 대한 설명과 그 구조를 앞에 제시한 표준 사전 형식에 의해 다음과 같이 표시되어 진다.

<entry>
<wname> 가난 </wname>
<표제어 번호> 1
<발음> 가난 </발음>
<문법정보>
<품사> 명사 </품사>
</문법정보>
<의미정보>
<의미번호> 1
<풀이> 살림살이가 넉넉하지 못함. 빈곤 </풀이>
<용례정보>
<용례> 가난한 집
<용례> 가난에 쪼들리다
<용례> 가난에서 벗어나다
</용례정보>
<속담정보>
<속담> 가난 구제는 나라도 못한다.
<속담풀이> 가난한 사람을 구제하기는 끝이 없어 개인은 물론 나라의 힘으로도 어렵다.
</속담정보>
<속담정보>
<속담> 가난한 집 제사 돌아오듯
<속담풀이> 치르기 힘든 일이 자주 닥침을 비유하는 말
</속담정보>
<숙어정보>
<숙어> 가난(이) 들다
<숙어풀이> ① 가난하게 되다. ② 쓸만한 것이 드물어 구하기 어렵다.
</숙어정보>
<관계어>
<표제어> 간난
<관계어표지> 유의어
</관계어>
</의미정보>
<파생어>
<표제어> 가난하다 </표제어>
<품사> 형용사 </품사>

</파생어>
</entry>

<entry>
<wname> 가다 </wname>
<표제어 번호> 1
<발음> 가다 </발음>
<문법정보>
<품사> 자동사 </품사>
<활용형 정보> 거라 불규칙
</문법정보>
<의미정보>
<의미번호> 1
<풀이> 목적한 곳을 향하여 움직이다. </풀이>
<용례정보>
<용례> 학교에 가다
<용례> 서울에 가다.
</용례정보>
<의미번호> 2
<풀이> (어떠한 목적을 위하여)떠나다. </풀이>
<용례정보>
<용례> 사냥을가다
<용례> 유학을가다.
</용례정보>
<의미번호> 3
<풀이> 도달하다. </풀이>
<용례정보>
<용례> 내가 한발 먼저 갔다
</용례정보>
<의미번호> 4
<풀이> 몸담아 있던 곳에서 다른 곳으로 옮기다. </풀이>
<용례정보>
<용례> 군대에가다
</용례정보>
<의미번호> 5
<풀이> 앞으로 나아가다. 걷다. </풀이>
<용례정보>
<용례> 들길을 가다
</용례정보>
<의미번호> 6
<풀이> 전달되다. </풀이>

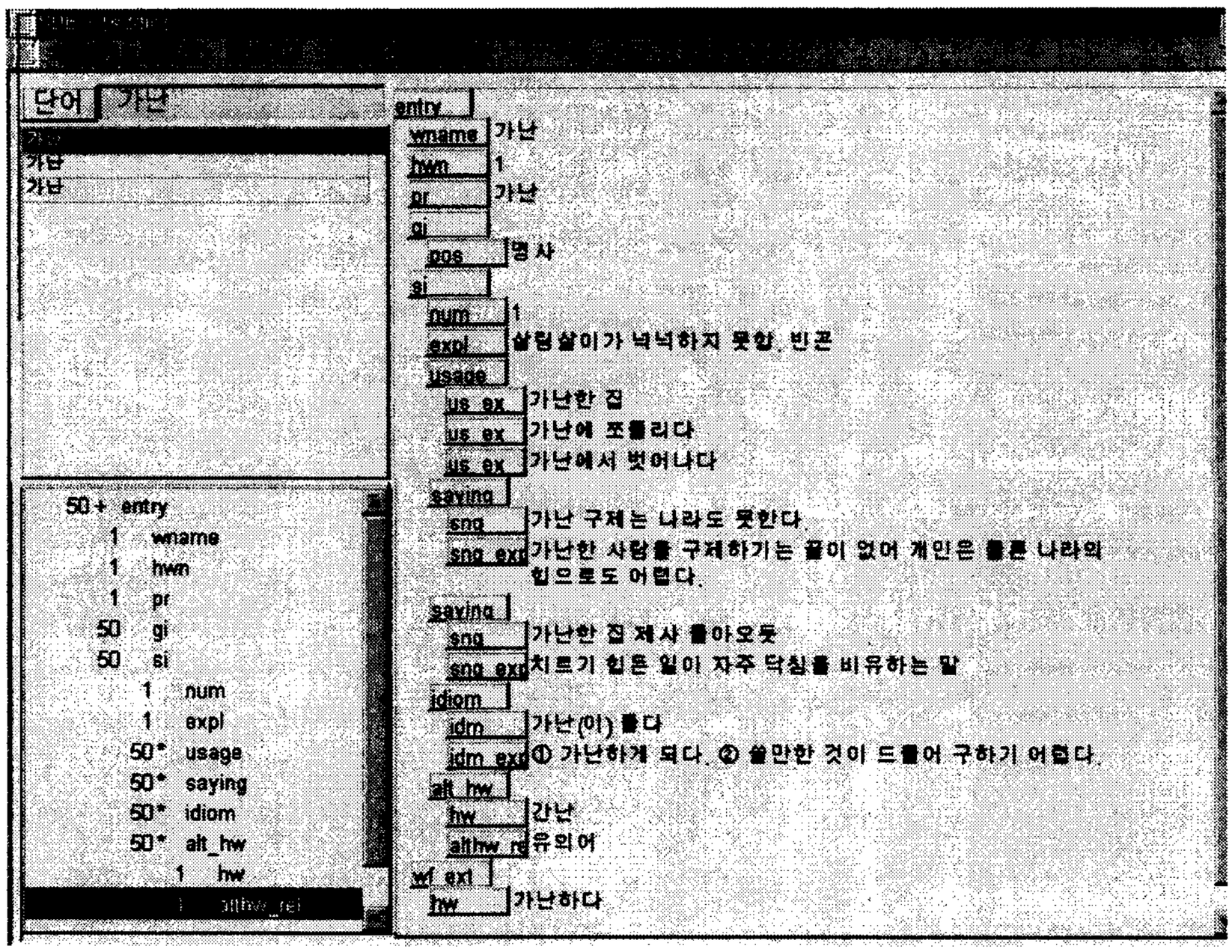
<용례정보>
<용례> 너에게 소식이 갔느냐?
</용례정보>
<의미번호> 7
<풀이> 시일이 경과하다. **</풀이>**
<용례정보>
<용례> 어느새 여름이 가고 날씨가 서늘해졌다.
</용례정보>
<의미번호> 8
<풀이> 지속되다. **</풀이>**
<용례정보>
<용례> 그의 결심은 오래가지 못할 것이다.
</용례정보>
<의미번호> 9
<풀이> (맛이나 음식이)변하다. **</풀이>**
<용례정보>
<용례> 이 음식은 맛이 갔다.
</용례정보>
<의미번호> 10
<풀이> 마음이 어떤 상태로 되다. **</풀이>**
<용례정보>
<용례> 이해가 가다.
<용례> 애정이 가다.
</용례정보>
<의미번호> 11
<풀이> 어떤 일에 힘이 쓰이다. **</풀이>**
<용례정보>
<용례> 장을 담그는 데는 손이 많이 간다.
</용례정보>
<의미번호> 12
<풀이> (금이나 흠 따위가)생기다. **</풀이>**
<용례정보>
<용례> 이 항아리에 금이 갔구나.
</용례정보>
<의미번호> 13
<풀이> 죽다. **</풀이>**
<용례정보>
<용례> 꽃다운 나이로 가다니.
</용례정보>
<의미번호> 14
<풀이> (전깃불 ·수돗물 따위가)꺼지거나 중단되다. **</풀이>**
<용례정보>
<용례> 책을 읽는데 감자기 불이 갔다.

```
</용례정보>
<의미번호> 15
<풀이>(값이나 무게 또는 차례 따위가)어느 정도에 이르다.</풀이>
<용례정보>
  <용례> 이 보석은 값이 상당히 간다.
  <용례> 첫째 가는 기술자.
</용례정보>
<의미번호> 16
<풀이>(어떤 상태나 일이)없어지다.</풀이>
<용례정보>
  <용례> 포도는 이제 한물 갔다.
  <용례> 아름다운 꿈은 모두 가 버렸다.
</용례정보>
<속담정보>
  <속담> 가는 날이 장날
  <속담풀이> 우연히 갔다가 뜻하지 아니한 일을 공교롭게 당함을 비유한 말.
  <속담> 가는 말이 고와야 오는 말이 곱다.
  <속담풀이> 남에게 말이나 행동을 좋게 하여야 자기에게도 좋은 반응이
        돌아 온다.
</속담정보>
</의미정보>
</entry>
```

[그림 21] 일반 국어 사전에 대한 SD가공

이러한 가공 절차를 거치게 되면, 표준 사전이 작성되어 지고, 이것은 TDMS 의 데이터 베이스관리 시스템에 저장이 되며, 사전 편집기를 통해 [그림 22]와 같은 형태로 브라우징하고 편집할 수 있게 된다.

대부분의 자연언어 처리 응용 프로그램들은 특정한 시스템에 맞게 정의되어 구현되어 있어서, 이식성 및 적응성이 부족하다. 전자 사전의 경우에도 예외가 아니며, 다른 프로그램이나, 다른 시스템에서 다시 사용한다는 것이 쉽지가 않다. 하지만 사전이 표준화된 형태로 표현 저장되어 있다면, 필요한 프로그램에 맞게 쉽게 변환하여 사용할 수 있다.



[그림 22] TDMS의 표준 사전 편집기의 브라우징 예

이러한 표준화된 사전의 논리적 내부를 설정하기 위해 기존의 사전들을 분석하였고, 이를 바탕으로 표준화된 DTD 를 작성하였다. 따라서, DTD 를 기준으로 하여 사전이 구성될 경우에, 언어학적으로 유용한 많은 사전(동의어 사전, 반의어 사전, 파생어 사전, 용례 사전, 속담 사전, 형태소 해석 사전, 숙어 사전, 외래어 사전 등)을 자동적으로 가공할 수 있을 것이다. 또한 이를 더 확장하여, 각 사전 표제어의 자질들을 정규화된 형태로 나타낸다면, 전산언어학적인 면에서는 자연언어 처리 시스템에 응용하여 사용할 수 있다. 여기서 제안된 표준 사전 형식은 다양한 전자 사전의 요구를 충족시켜 줄 뿐만 아니라, 표준 사전 형식의 연구와 전문 용어 사전의 표준화에도 활용될 수 있다고 본다.

9 장. 맺음말

TDMS 는 전자 사전의 생성, 변형, 관리를 편리하게 하고, 코퍼스를 편리하게 관리할 수 있도록 설계되었다. TDMS 가 다양한 형태의 사전 형식을 처리할 수 있도록, 유연성있는 사전 정의 언어인 SDML(Standard Dictionary Markup Language)를 정의하였으며, 그 언어를 구현하기 위한 의미에 대한 규격을 기술하였다. 이 규격의 대부분은 이미 "텍스트 및 전자사전 관리시스템 개발" 프로젝트 팀에 의해 구현이 되어 기본적인 성능이 검증이 되었다.

TDMS 에 관한 연구는 기존의 사전관리 시스템에 없는 여러가지 기능들을 제안하고 설계하였으며 앞으로의 이용 방향은 무한하다. 본 연구의 기초적인 부분은 이미 구현이 되었으나 몇가지 복잡한 부분은 설계 단계에 머무르고 있다. 앞으로 해야 할 일은 개발된 기본 규격을 보완 및 확장하며, 아직 구현되지 아니한 구조화 문서 검색, 하이퍼 링크 등의 규격을 더 연구하여 구체화할 계획이다. 또한 기존의 사전을 최대한 많이 이용할 수 있도록 decoder/encoder 부분을 확장하여 여러 사전이 TDMS 를 쉽게 이용할 수 있도록 관련 규격을 정의하며, 각계의 의견을 수렴하여 필요하다면, SDML 을 확장하거나 표준 사전 태그를 정의할 계획이다.

10장. 참고 문헌

[1] Charles F. Goldfarb, " The SGML Handbook," Clarendon Press, Oxford, 1990.

[2] _, "Information technology - Text and office systems - Document Style Semantics and Specification Language(DSSSL) - Draft", ISO/IEC DIS 10179.2.

[3] Sperberg-McQueen and Lou Burnard, "Guidelines for Electronic Text Encoding and Interchange(TEI P3)," Vol I and Vol II, ACH, ACL, ALLC, April 8, 1994.

[4] Lou Burnard and C. M. Sperberg-McQueen, "TEI Lite: An Introduction to Text Encoding for Interchagne," TEI U 5, June 1995.

[5] _, "문서 기술 언어 SGML," KS C 5913-1993, 한국표준협회, 1993.

[6] Ian A. Macleod, "Storage and Retrieval of Structured Documents," Information Processing & Management Vol 26, No 2. pp 197-208, 1990.

[7] Nancy Ide and Jean Veronis, "Text Encoding Initiative Background and Context," Kluwer Academic Publishers, 1995.

[8] Alshawi, H. (1989), "Processing Dictionary Definitions with Phrasal Pattern Hierarchies", in Boguraev, B. and E. Briscoe(eds.), 153-170.

[9] Amsler, Robert A. and W. Tompa(1988), "An SGML-based Standard for English Monolingual Dictionaries, in Proceedings of the 4th Annual Conference of the UW Centre for the New Oxford English Dictionary, Waterloo, Ontario: 61-80.

[10] Bierwisch, M.(1983) Semantische und konzeptuelle Repraesentationen lexikalischer Einheiten, Studia Grammatica XXII, Berlin

[11] Boguraev, B. (1991), Special Issue on Computational Lexicons, International Journal of Computational Lexicography 4

[12] Boguraev, B. (1994) Machine-readable dictionaries and computational linguistics research, in A. Zampolli, N. Calzolari, and M. Palmer(eds.)

[13] Boguraev, B. and E, Briscoe(1989), Computational Lexicography for Natural Language Processing, Longman Limited, Harlow and London

[14] Bryan, M.(1988), SGML: An Author's Guide to the Standard Generalized Markup Language, Addison-Wesley

[15] Copestake, A. (1990), "An Approach to Building the hierarchical Element of a Lexical Knowledge Base from a Machine Readable Dictionary", in Proceedings of the First

International Workshop on Inheritance in Natural Language Processing, Tilburg, The Netherlands:19-29

[16] EDR(1993), EDR Electronic Dictionary Technical Guide, Japan Electronic Dictionary Research Institute.

[17] Zampolli, A., Calzolari, N., and M. Palmer(1994), Linguistica Computationale, Vol. IX.X Current Issues in Computational Linguistics: in Honor of Don Walker, Kluwer Academic Publishers, Dordrecht.

[18] Jae Sung Lee, Key-Sun Choi, "Two Approaches for the Roman to Hangul Transcription Based on Statisitcal Translation Method", ICCPOL97.

[19] Jae Sung Lee, Key-Sun Choi, "Various Transliterations of Foreign Words in Multilingual Information Retrieval", submitted to IRAL97.

[20] Byung Jin Choi, Jae Sung Lee, Woon Jae Lee, Key-Sun Choi, "The Processing of Dictionary Definitions and Maintenance in Text and Dictionary Management System(TDMS)", ICCPOL97.

[21] Byung Jin Choi, Jae Sung Lee, Woon Jae Lee, Key-Sun Choi, "A Logical Structure for the Construction of Machine Readable Dictionaries" The 11th Pacific Asia Conference on Language, Information and Computation (PACLIC11 '96).

[22] 강범모 (1995), “언어 데이터베이스와 언어 연구”, 정광 외 저, 한국어 데이터베이스의 설계 및 응용을 위한 기초 연구, 서울: 민음사

[23] 최기선(1991), “전자사전 연구 개발과 정보 서비스로의 활용”, 우리말 정보화 잔치.317-325

[24] 이재성, 최병진, 이운재, 최기선,"텍스트 및 전자사전 관리시스템의 설계" 제 8 회 한글 및 한국어 정보처리 학술발표, 1996.

[25] 최병진, 이재성, 이운재, 최기선, "표준화를 위한 일반사전의 논리구조" 제 8 회 한글 및 한국어 정보처리 학술발표, 1996.

[26] 노대식, 이혜란, 이재성, 이상기, 주종철, 박동인, "구문 구동형 한글 SGML 문서편집기의 구현" 한국 정보과학회 가을 학술대회, 1996.

[27] 이재성, 최병진, 이운재, 최기선, "텍스트 및 전자사전 관리시스템을 위한 표준 사전 표기언어의 설계", 인지과학 Vol.7, No. 4 , 1996.

[28] 최병진, 이운재, 이재성, 최기선, "기계가독형 사전 구축을 위한 사전항목의 논

리 구조", 인지과학 Vol. 7, No. 2, 1996.

[29] "http://www.iijnet.or.jp/edr/", EDR 사전

[30] "http://madonna.postech.ac.kr/STEP2000/mid-report/thead.html", 한국어 형태소 해석기 및 형태소 해석용 사전 개발

[31] "http://kibs.kaist.ac.kr/", 국어 정보 베이스

여백

[부록]

TDMS 설치 및 사용설명서

1997.8.13

한국과학기술원 & 오롬테크

목 차

I. 설치

1. 설치준비물

- ◆ TDMS Install CD 1 장 혹은 TDMS Install Diskette(3.5" 9 장)
- ◆ 설치 매뉴얼

2. 설치과정

가. Setup 실행

- ◆ Install CD-Rom 혹은 Install Diskette 1 번을 드라이브에 넣는다.
- ◆ Setup.exe 를 실행한다.

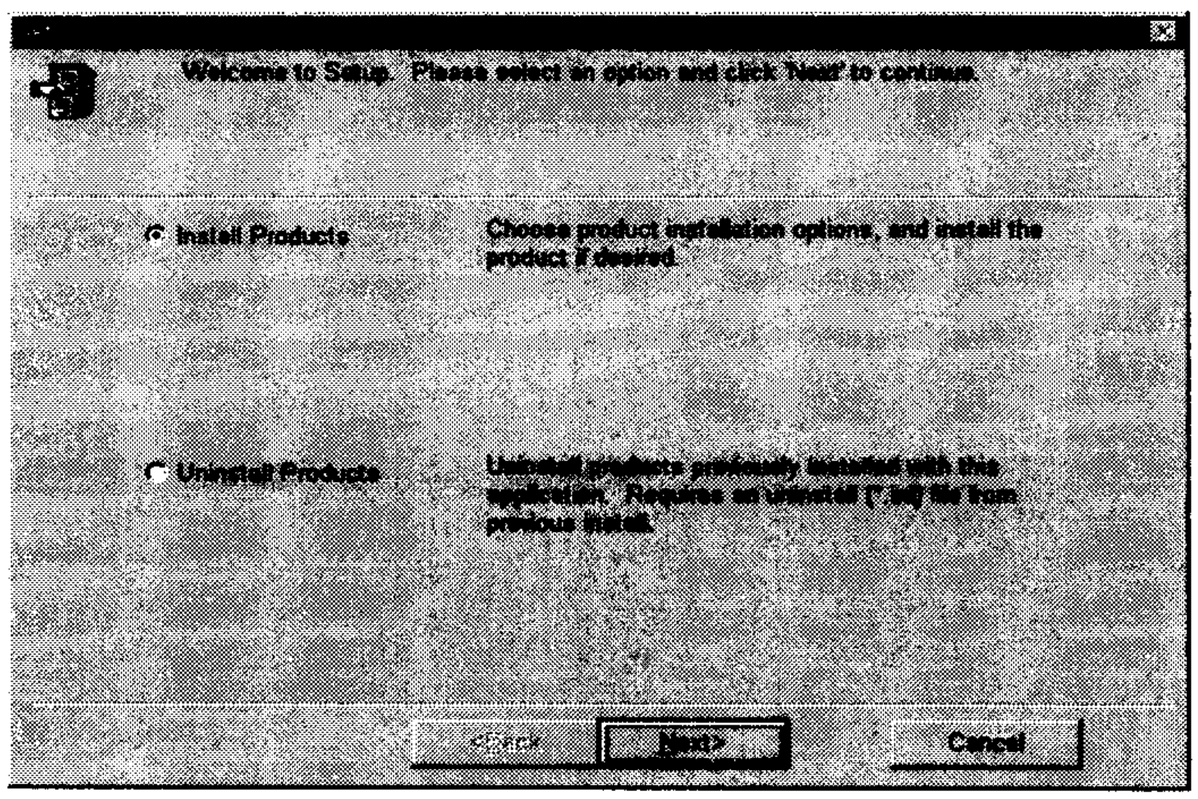


그림 1. Setup 초기화면

◆ Install Product 를 선택한 다음 Next 버튼을 누른다.

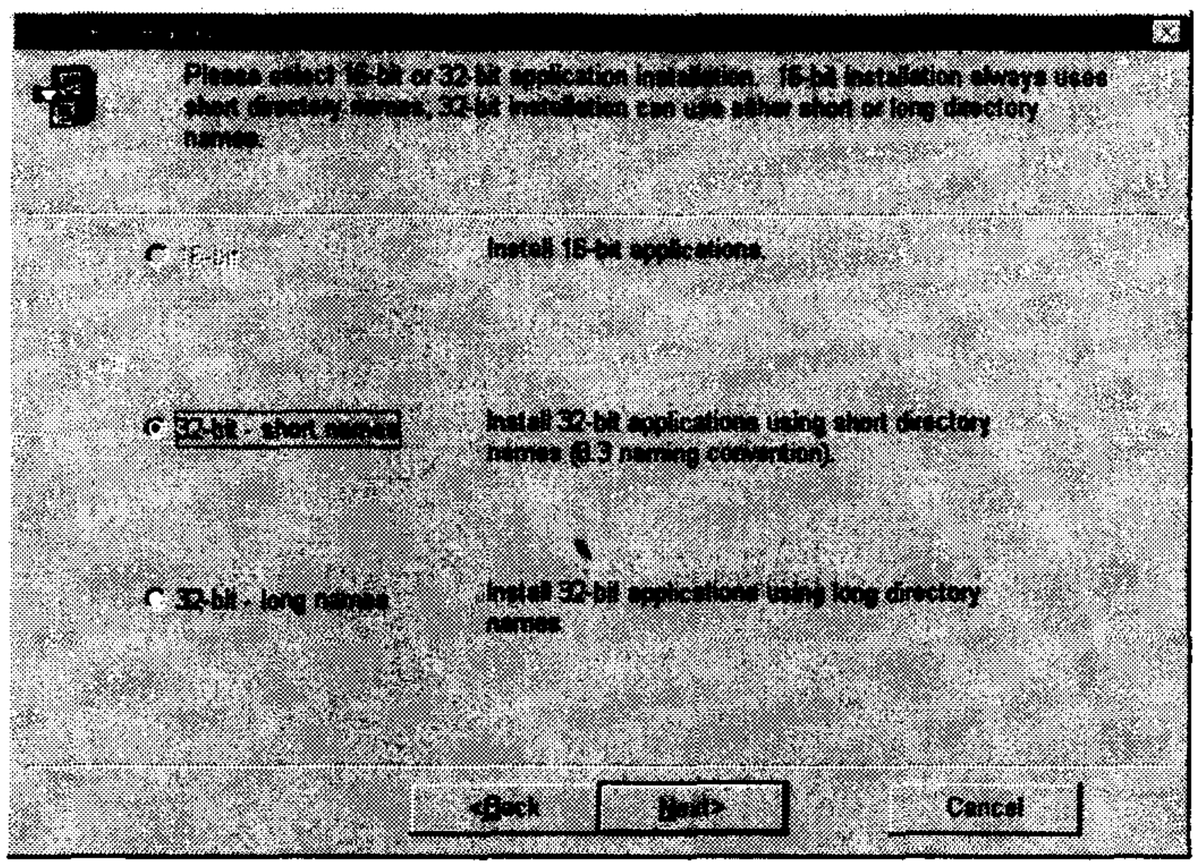


그림 2. OS 종류 선택 화면

◆ Operating System Type 을 32Bit Shortname 로 선택한 후 Next 버튼을 누른다.

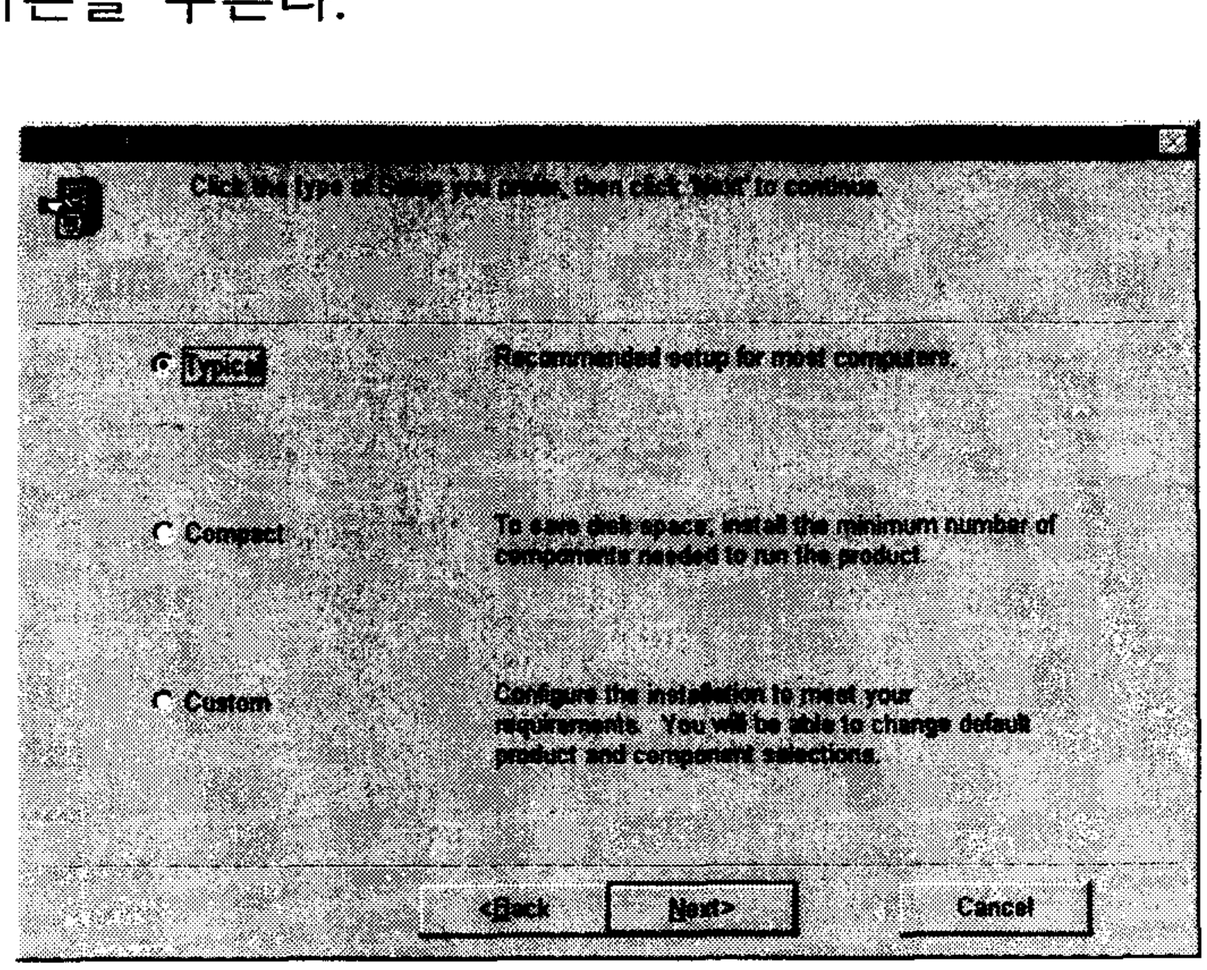


그림 3. Setup Option 선택화면

◆ Setup Option 을 Typical 로 선택한 후 Next 버튼을 누른다.

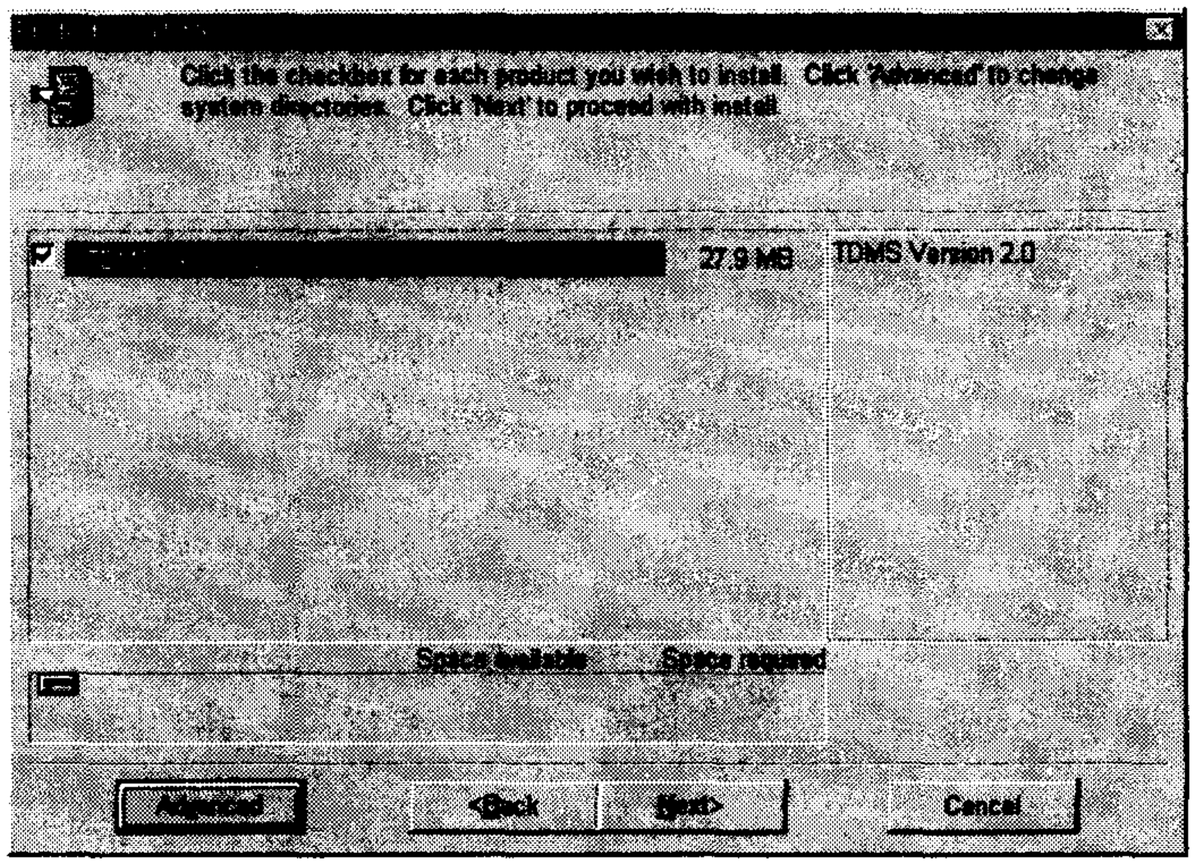


그림 4. 설치 Product 선택화면

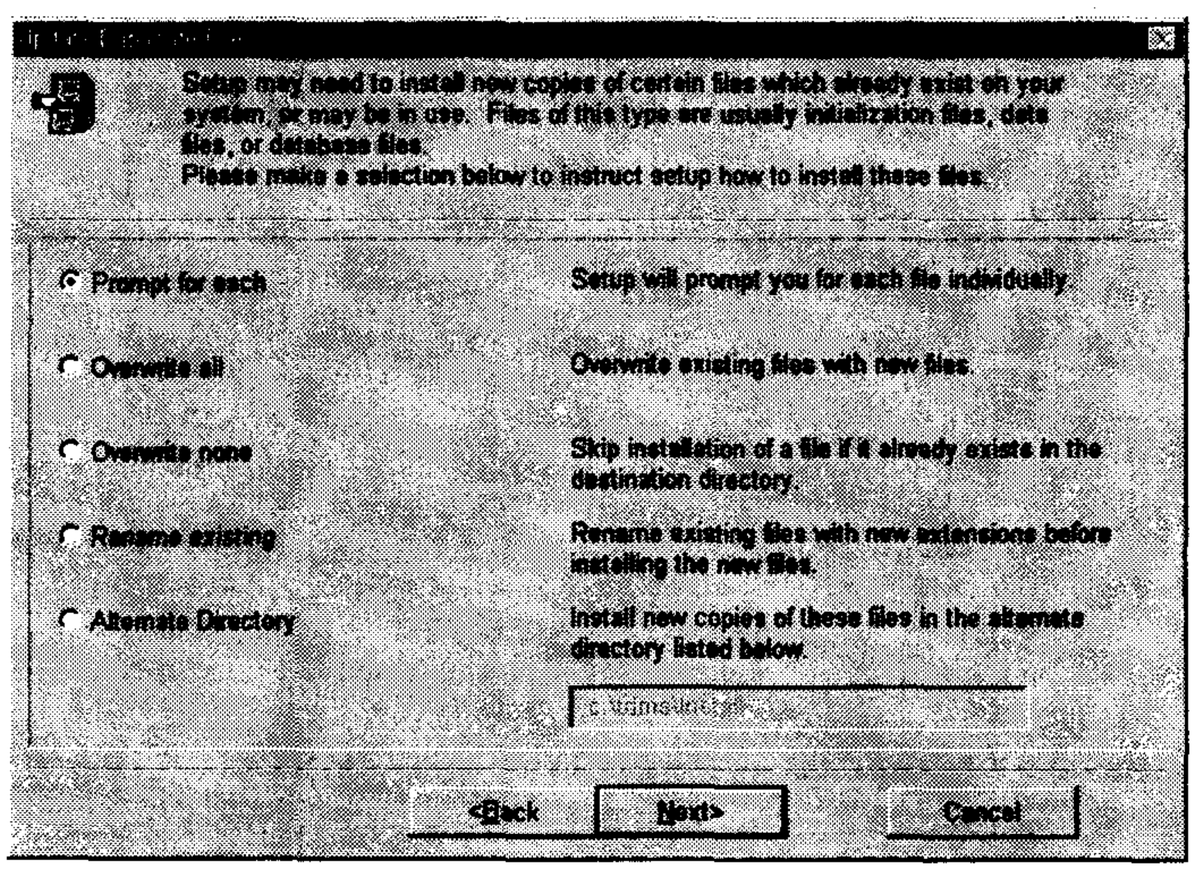


그림 5. 중복파일 설치 옵션 화면

- ◆ TDMS Version2.0 을 선 택한 후 Next 버튼을 누른다.
- ◆ 중복파일 설치 옵션을 Prompt for each 로 선택한 다음 Next 버튼을 누른다.

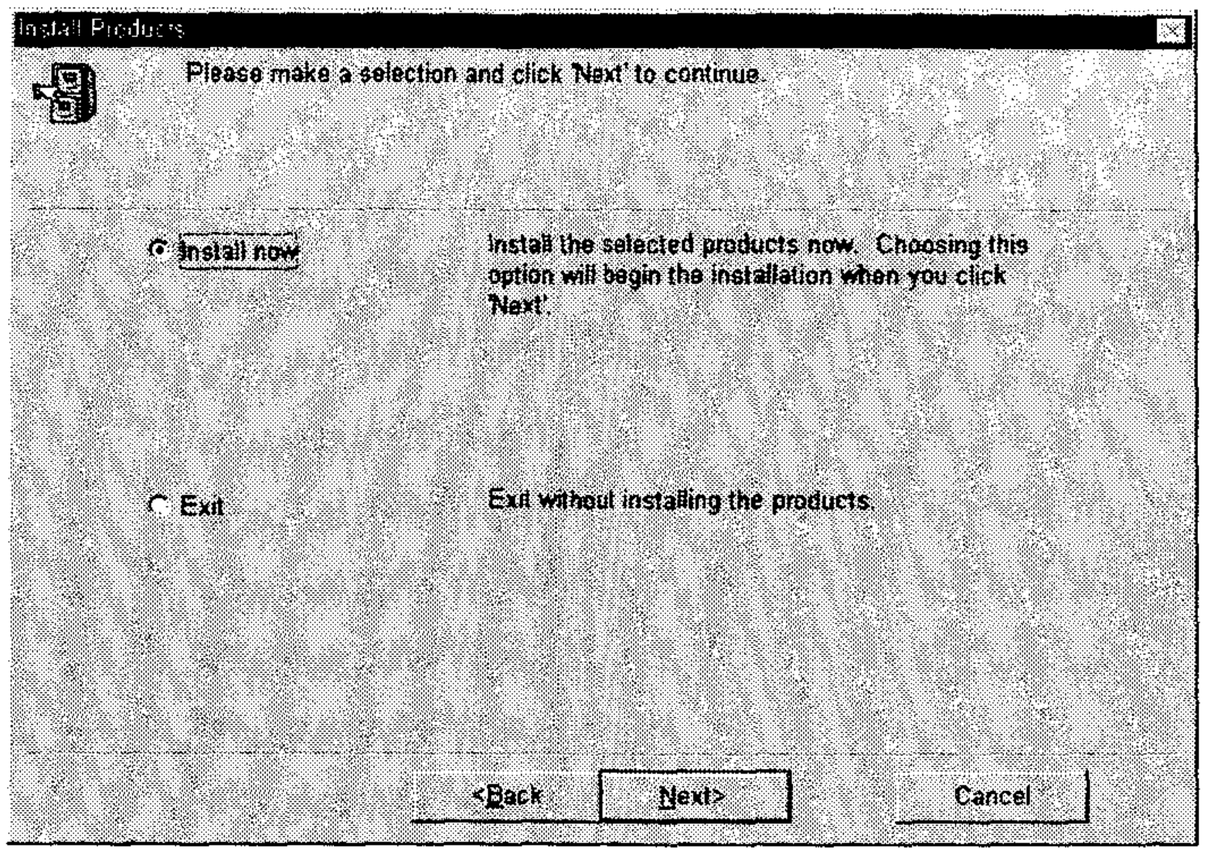


그림 6.설치 확인 화면

◆ Install now 를 선택한 다음 Next 버튼을 누른다.

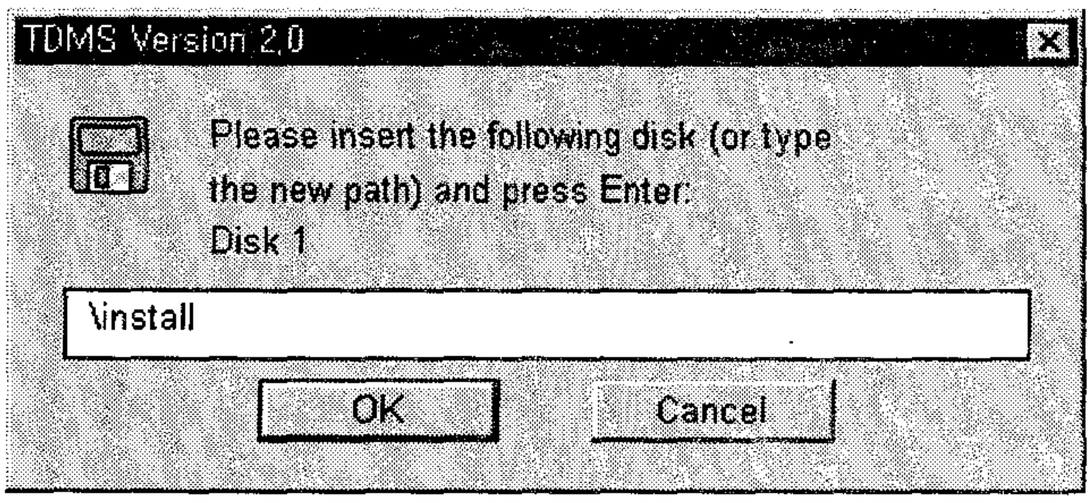


그림 7.setup.exe Path 확인화면

◆ Setup.exe 가 설치되어 있는 Full Path 를 입력한 후 OK 버튼을 누른다. (CD-ROM 으로 설치 할때는 나오지 않음)

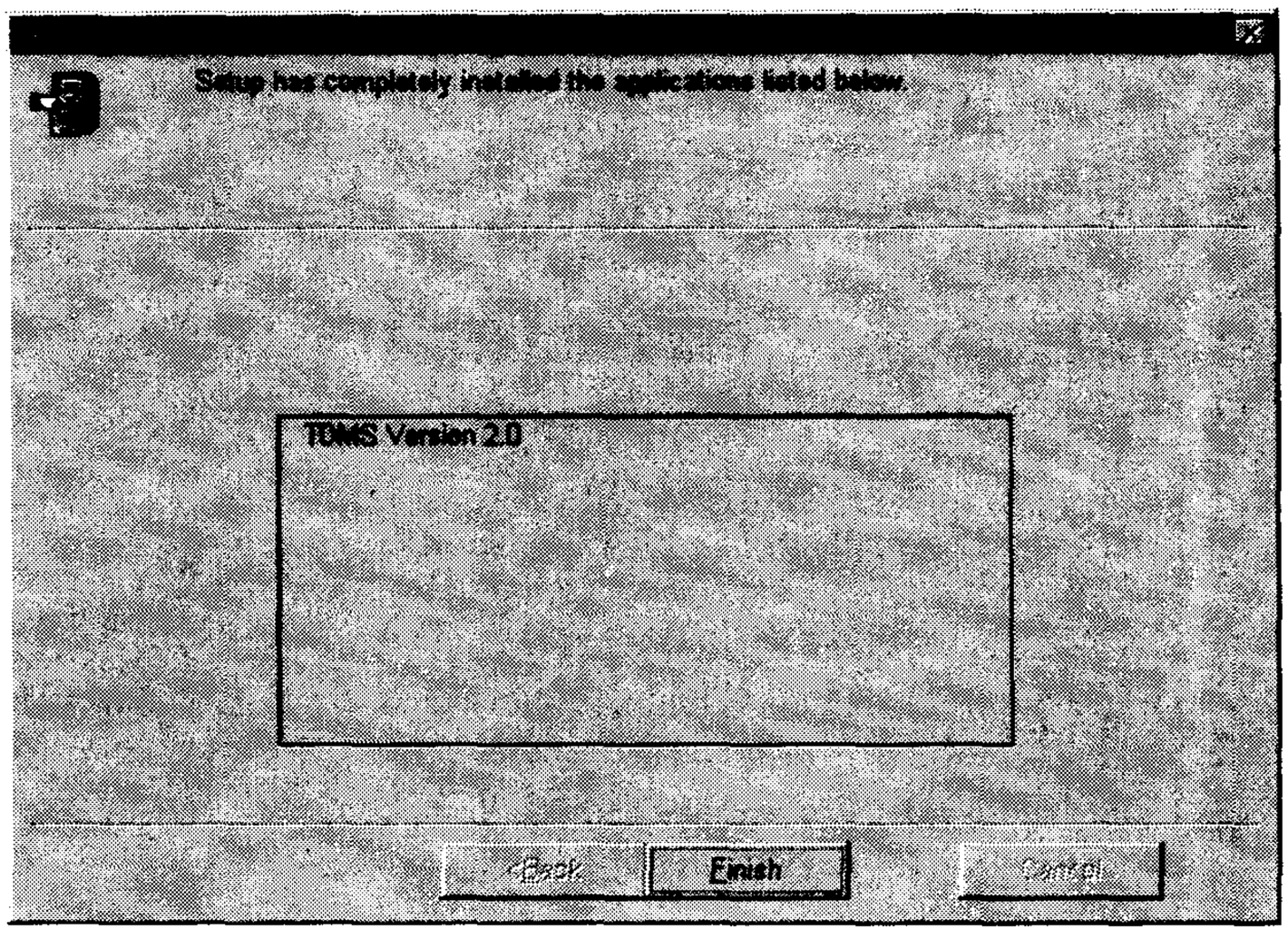


그림 8.설치완료 화면

◆ 위와 같은 설치 완료 화면이 나오면 Finish 버튼을 누른다.

나. Setup 결과

①. TDMS 프로그램 그룹 생성

설치 프로그램이 자동으로 만들어 주는 것으로 아래 그램과 같은 프로그램 그룹이 만들어진다.

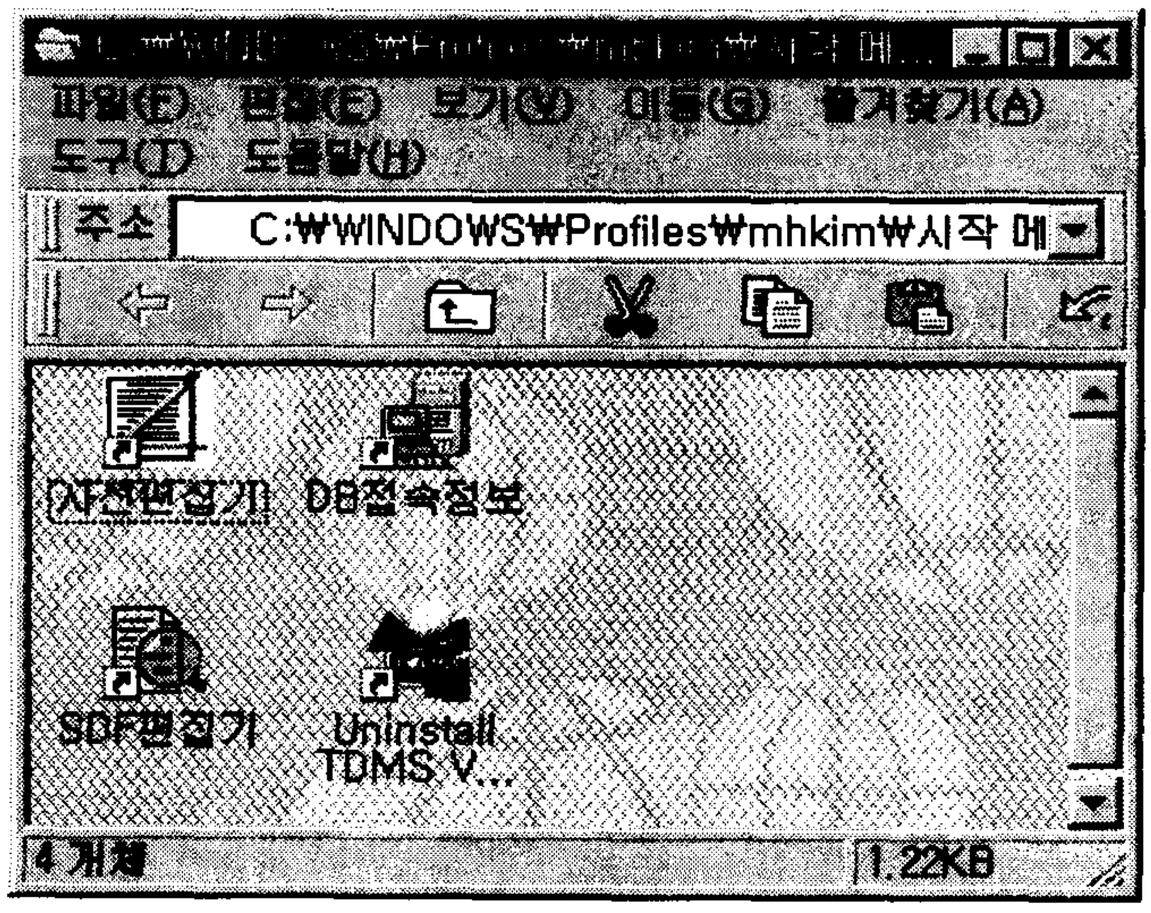


그림 9. TDMS 그룹

②. Registry 정보 추가

설치 프로그램이 자동으로 만들어 주는 것으로 아래 그램과 같은 레지스트리 정보가 추가되어 있는 것을 확인한다.

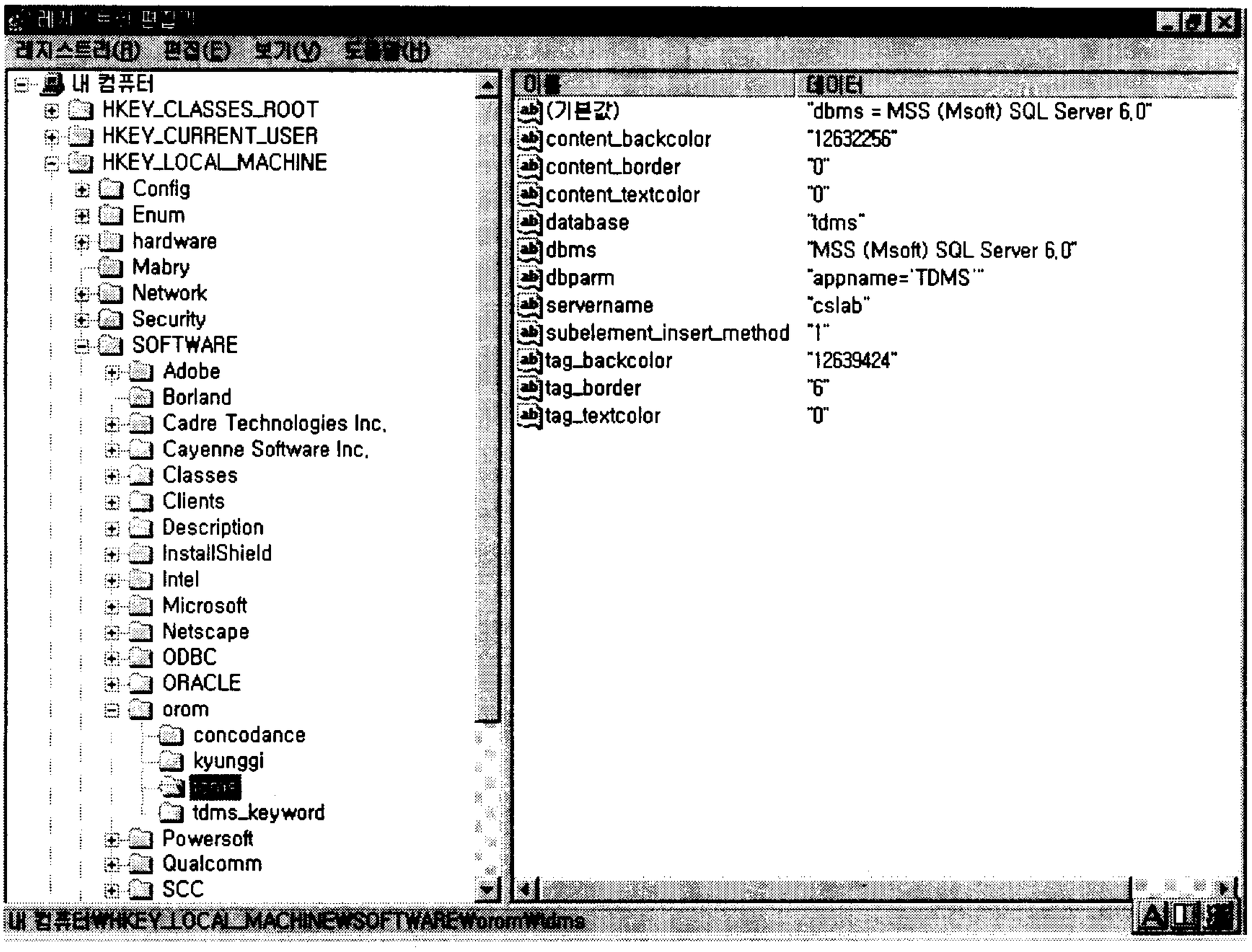


그림 10. Registry 정보 화면

다. 수동 정보 입력

①. 실행 PATH 추가

- \autoexec.bat 파일에 다음과 같은 내용을 추가한다.
- **PATH C:\pwrs;C:\sql60;%PATH%**

②. DB 접속 정보 생성

- TDMS 그룹의 DB 접속정보를 실행한다.(그림 9 참조)

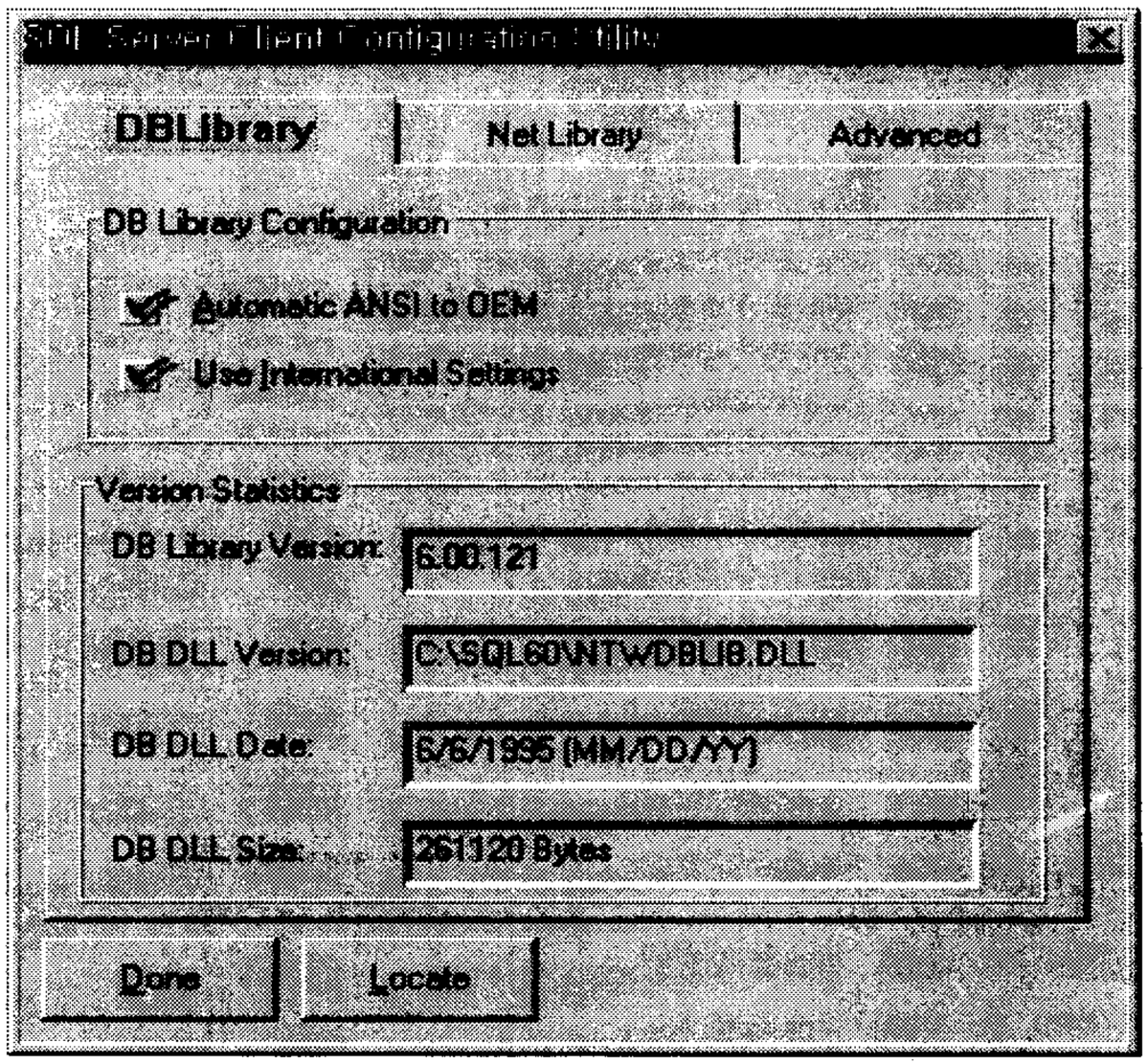


그림 11. DB 접속정보 초기화면

◆ Advanced 를 선택한다.

◆ Server 에 **cslab** (그림 10.의 Registry 정보중 servername 과 일치해야 함)

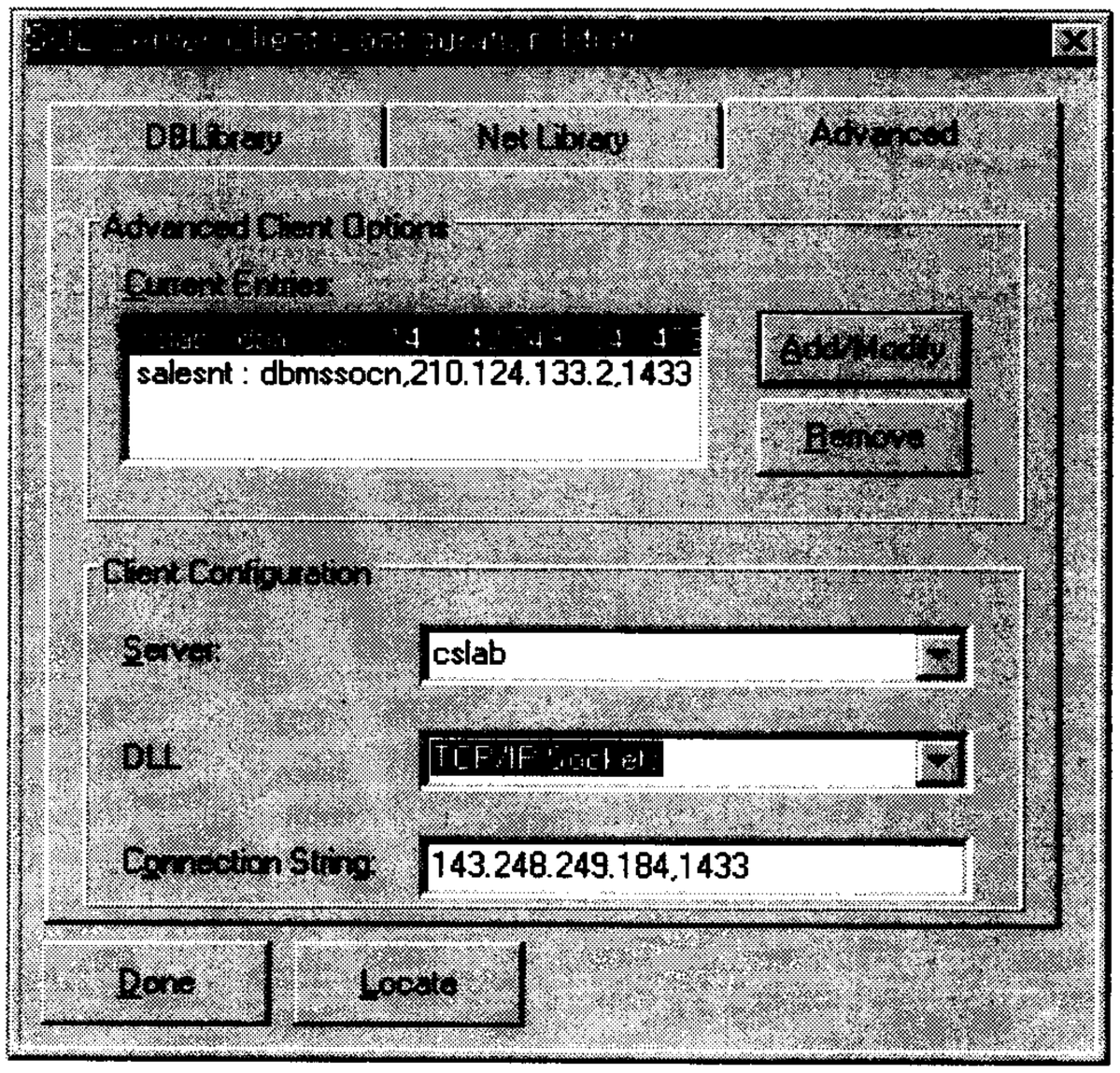


그림 12. DB 접속정보 입력화면

- Server에 **cslab** (그림 10.의 Registry 정보중 servername과 일치해야 함)
- DLL에 **TCP/IP Sockets(dbmssocn)**
- Connect String에 **143.248.249.184,1433**을 입력한 후 Add/Modify 버튼을 누르고 Done 버튼을 누른다.

3. Reboot

4. Install 요약

1. Setup.exe 실행
2. Autoexec.bat에 PATH 추가
3. DB 접속정보 입력
4. Reboot

II. 사용설명서

1. SDF Builder

가. 개요

SDF 는 표준사전포맷(Standard Dictionary Format)으로 SDF Builder 는 사전의 구성요소와 구조를 편집할 수 있다. 작성된 SDF 는 사전편집 및 사전 검색에서 사용되며 구조검색에도 이용된다.

SDF 는 SGML 의 Subset 인 SDML(Standard Dictionary Markup Language:KAIST) 규격을 따르고 있으므로 호환성, 확장성등이 뛰어나다.

나. 기능요약

- ◆ 새 SDF 생성
- ◆ SDF 수정
- ◆ SDF 삭제(사전포함)
- ◆ 요소 생성 및 편집
- ◆ 구조 편집
- ◆ SDF Export
- ◆ SDF Import
- ◆ 사용자관리 등이 있다.

다. 사용설명

①. 프로그램 시작

Window95 왼쪽 하단에 있는 **시작**버튼을 누른 후 **프로그램** ▷ **TDMS** ▷ **SDF 편집기**를 차례로 선택한다.

②. Login 화면

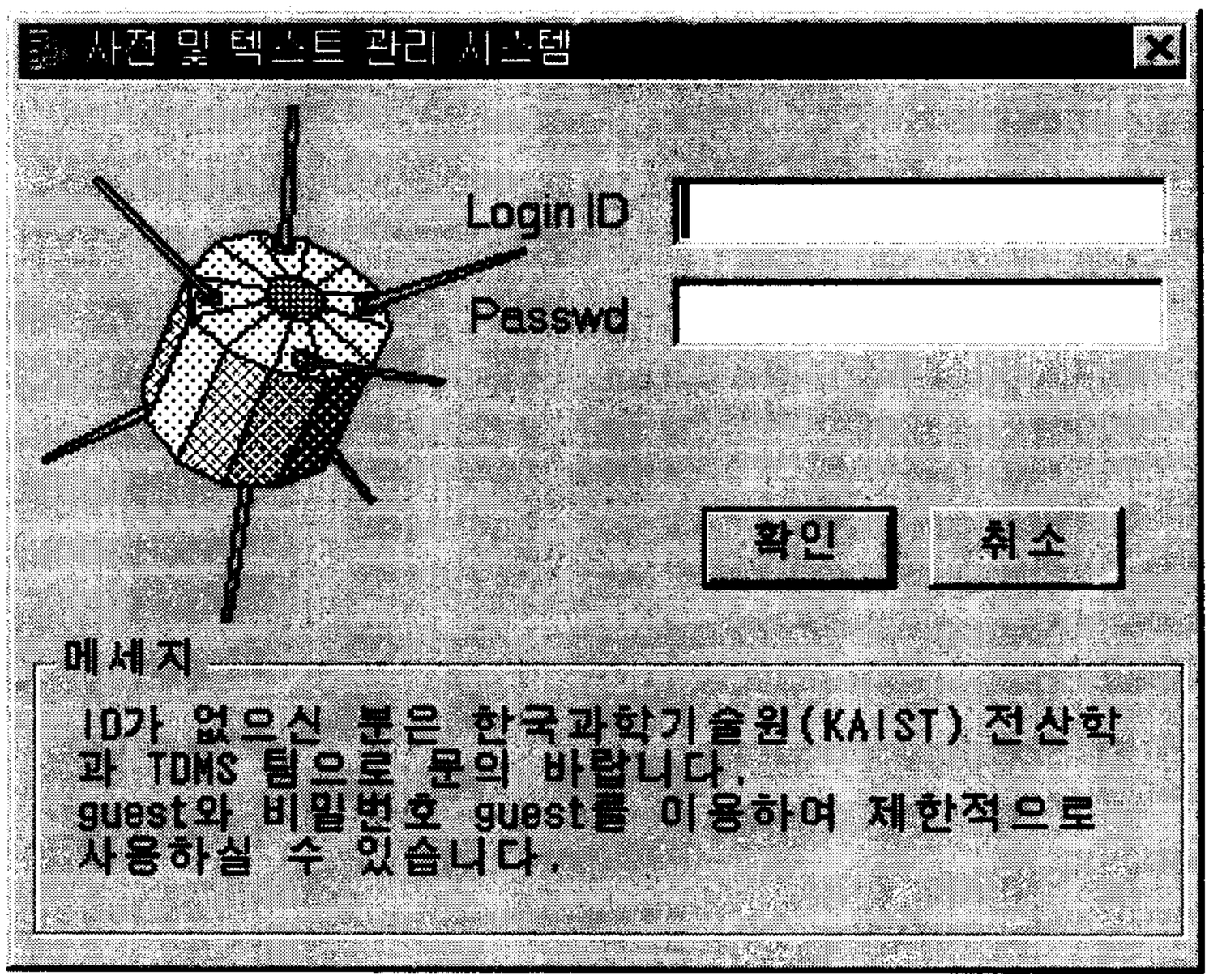


그림 13. Login 화면

- 부여받은 Login ID 와 Passwd 를 입력한 후 확인 버튼을 누른다.
- 이때 입력한 ID 와 Passwd 는 사용 허가권과 연동이 되므로 TDMS Supervisor 이외에는 편집 및 저장등을 할 수 없다.(사용자관리 참조)
- ID 와 Passwd 를 부여받지 않은 사용자는 guest 를 이용하면 된다.

③. 기본화면 설명

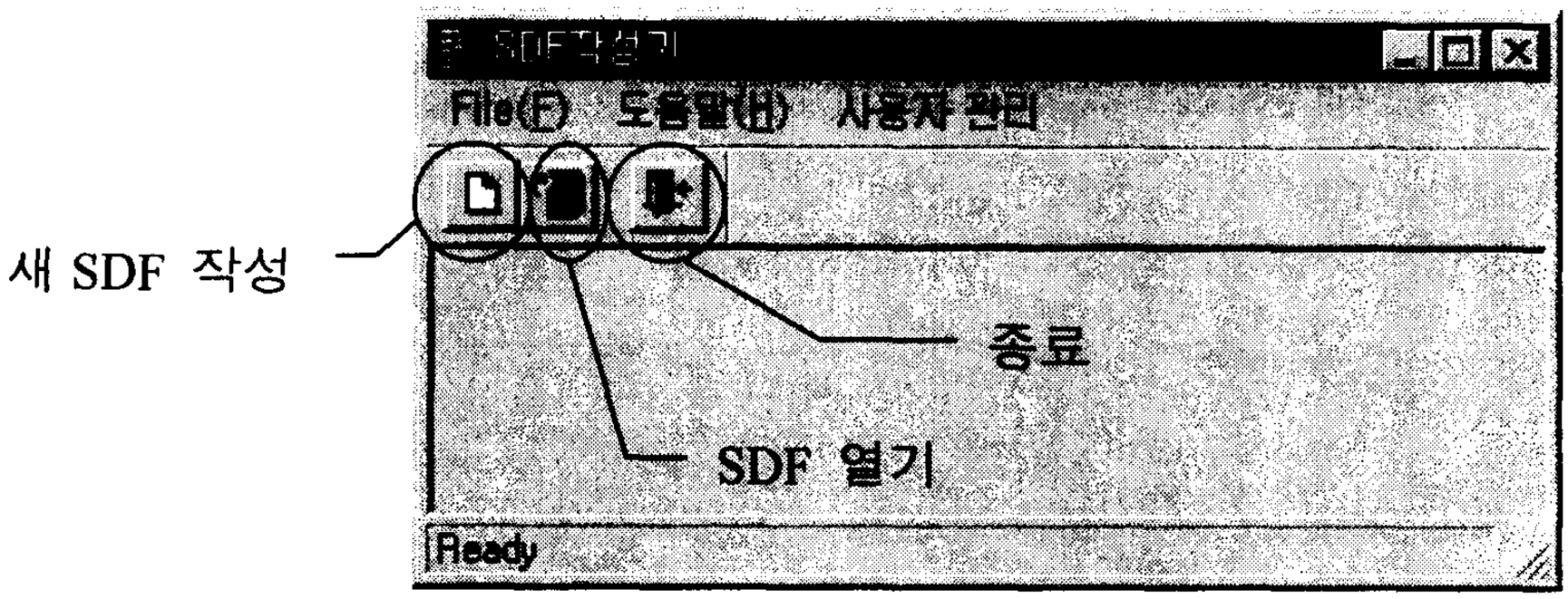


그림 14. SDF 기본화면

SDF 작성기의 기본화면에서는

- ◆ 새 SDF 작성,
- ◆ SDF 열기,
- ◆ 사용자관리를 선택할 수 있다.

(1) 새 SDF 작성

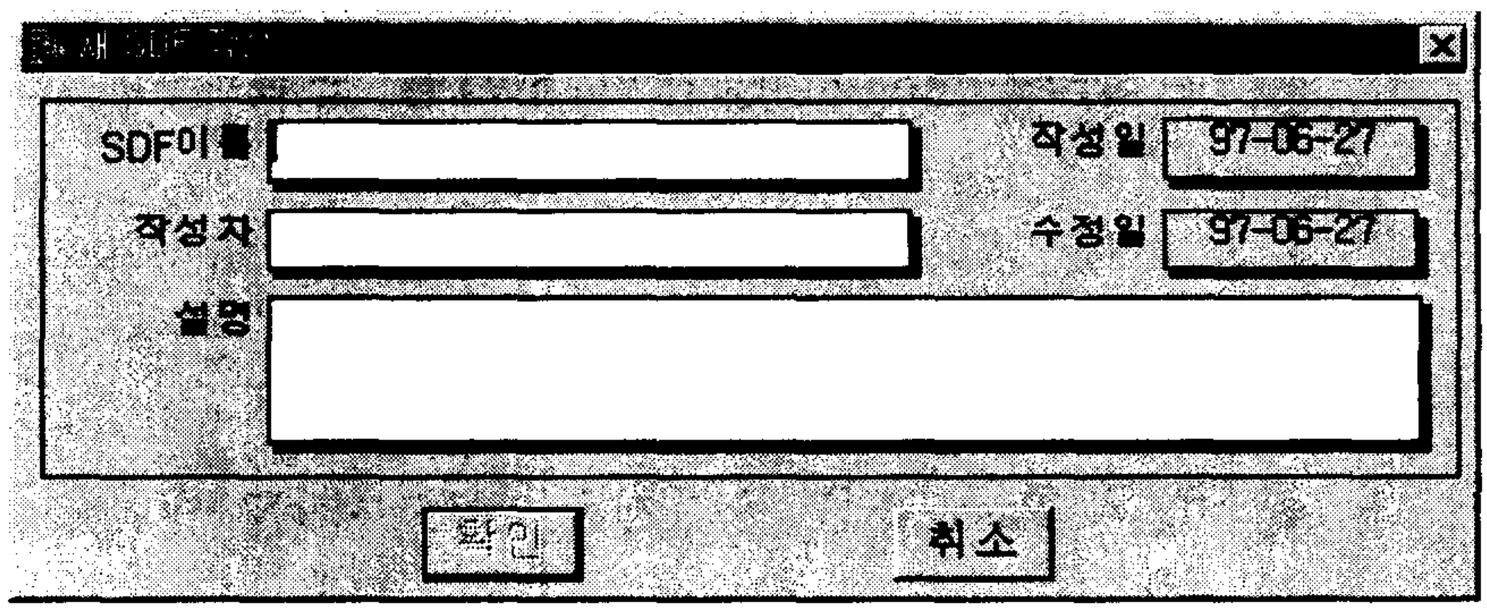


그림 15. SDF 편집 화면

- ◆ SDF 이름을 입력한 후 Tab 을 누른다.
- ◆ 작성자를 입력한 후 Tab 을 누른다.
- ◆ 설명을 입력한 후 Tab 을 누른다.
- ◆ 확인 버튼을 누르면 저장이 된다.
- ◆ 이때 SDF 이름이 기존에 있으면 경고 메세지가 나오며 저

장이 되지 않는다.

(2) SDF 선택 화면

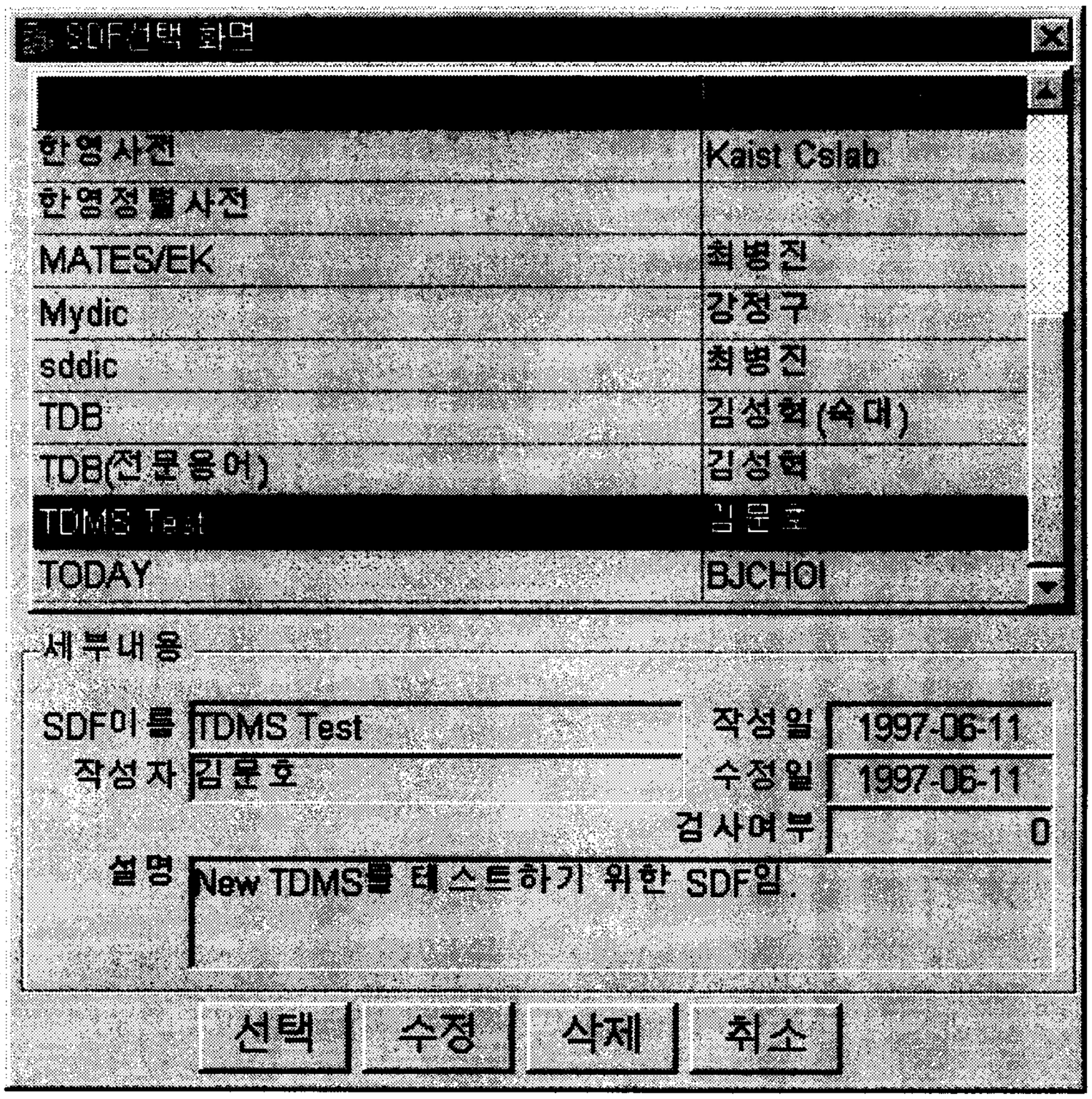


그림 16. SDF 선택화면

- ◆ 작성되어 있는 SDF 정보가 화면에 나타난다.
- ◆ 이때 문자색이 청색인 SDF는 사전 데이터가 입력되어 있다는 표시이고 문자색이 검정색인 SDF는 사전 데이터가 입력되어 있지 않다는 표시이다.
- ◆ 사전 데이터가 입력되어 있는 SDF는 요소 및 구조 편집 시에 제약이 따른다.
- ◆ 원하는 SDF를 마우스로 선택(Click)하면 SDF의 상세한 정보가 나타나며 **선택, 수정, 삭제, 취소** 버튼을 이용하여 해당작업을 할 수 있다.

- ◆ 선택 버튼은 SDF의 요소와 구조 편집을 할 수 있는 화면을 연다.
- ◆ 수정 버튼은 SDF의 이름, 작성자, 수정일자, 비고사항등을 변경할 수 있는 화면을 연다.
- ◆ 삭제 버튼은 SDF를 삭제할 수 있는 기능인데 SDF 뿐만 아니라 사전 데이터까지도 지워지므로 유의해야 한다.

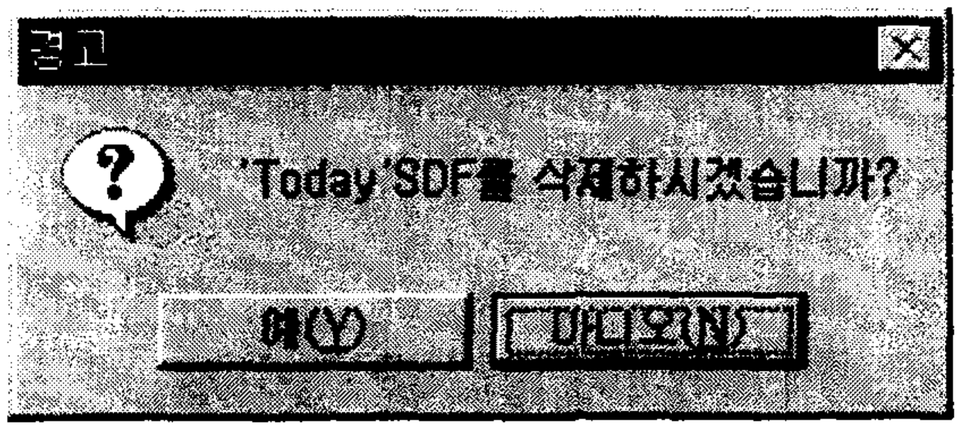


그림 17. SDF 삭제 경고 화면

- ◆ 취소 버튼은 SDF 선택을 취소하고 SDF 기본화면으로 돌아간다.

④. SDF 요소 및 구조 편집

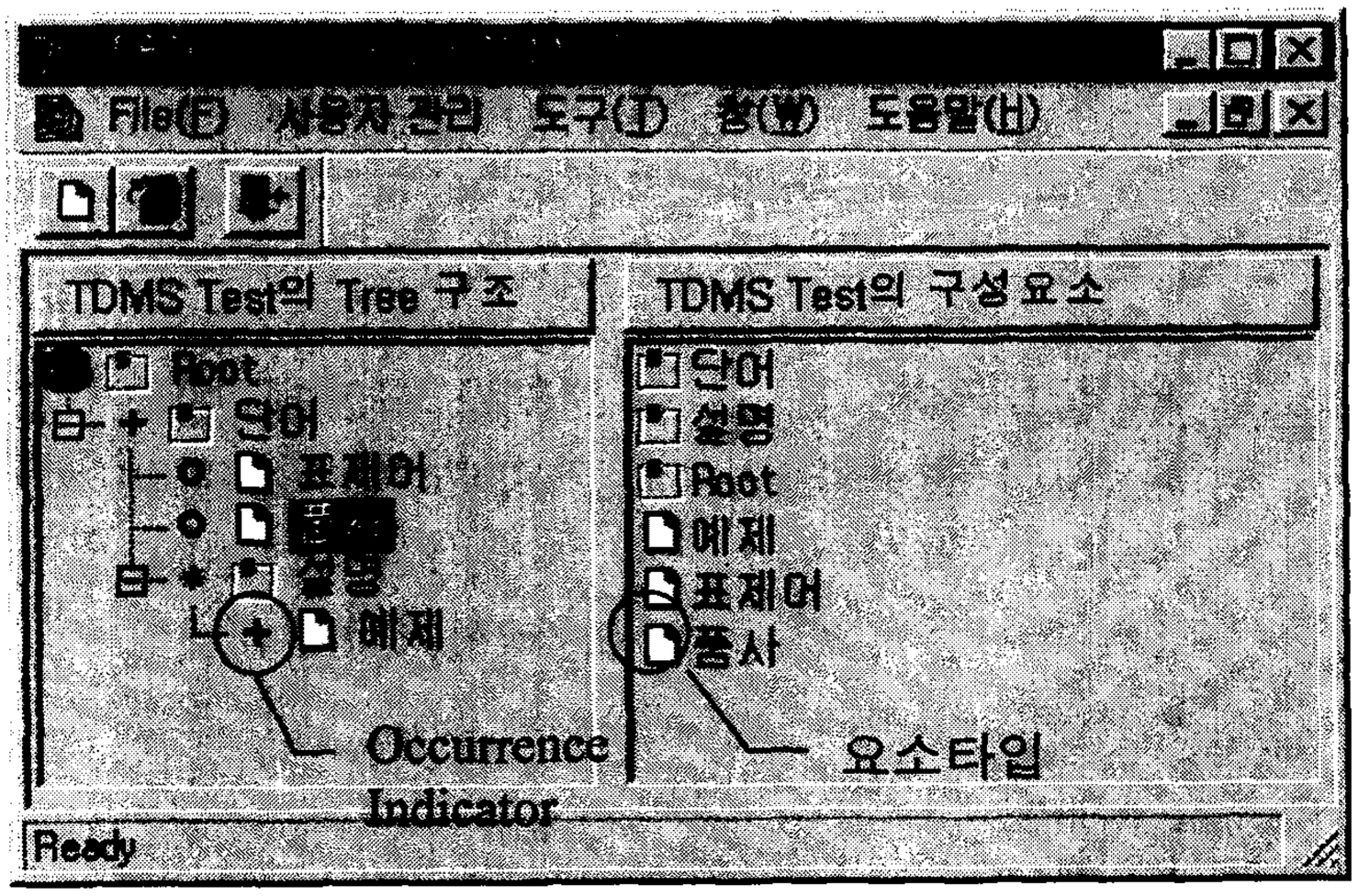


그림 18. SDF 요소 및 구조 편집

(1) 구성요소 편집

◆ 구성요소는 요소의 이름, 타입(#PCDATA, SEQuence Group,AND Group, OR Group)과 설명등으로 정의된다.

◆ 요소의 타입은

#PCDATA

SEQuence Group

AND Group

OR Group 등이 있다.

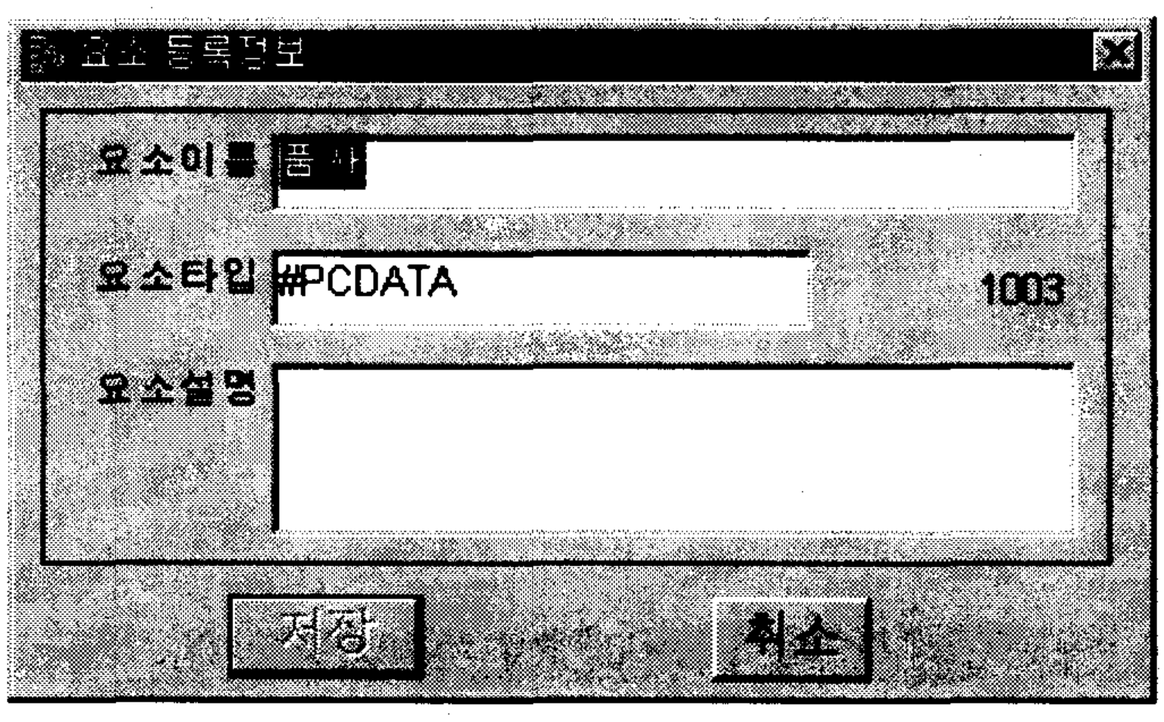


그림 19. 요소 등록정보 화면

◆ 구성요소 화면에서 마우스의 오른쪽 버튼을 누르면 다음과 같은 팝업 메뉴가 나타난다.

그림 20. 구성요소 팝업메뉴

◆ 요소를 새로 등록하려면 **NEW**

◆ 삭제하려면 **삭제/복구**, 이때 삭제된것은 약간 흐리게 나타

나며

◆ 삭제된것을 복구하려면 다시 **삭제/복구**를 선택한다. 이때 복구된것은 다시 진하게 표시된다.

◆ 저장하려면 **저장**을 선택하고

◆ 저장되어 있는 것을 수정하려면 **등록정보**를 선택한다.

(2) Tree 구조 편집

◆ Tree 구조는 상위노드, 상위노드에서의 순서와 Occurrence (Required, Required & Repeatable, Optional, Optional & Repeatable)등으로 정의된다.

◆ Occurrence Indicator 는

o Required,

+ Required & Repeatable,

? Optional,

* Optional & Repeatable 의 4 가지 종류가 있다.

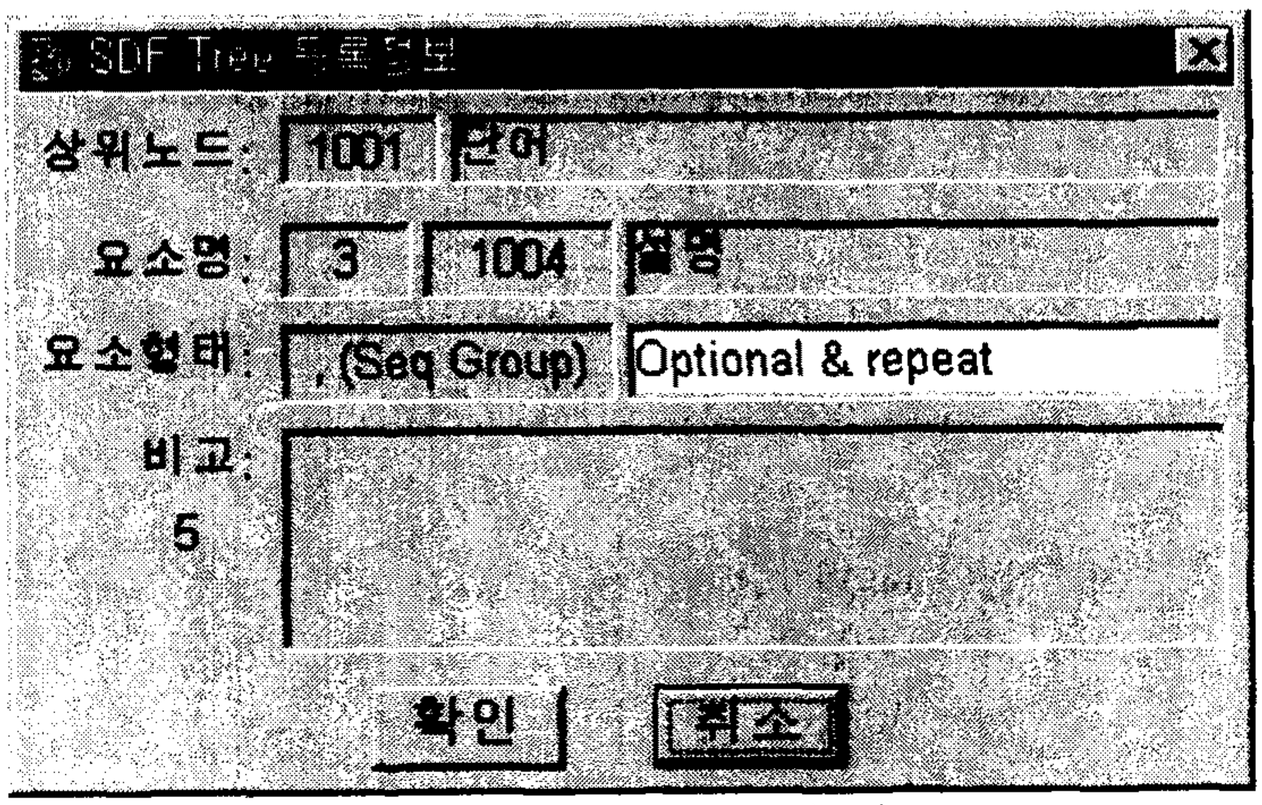


그림 21. Tree 구조 등록정보 화면

◆ Tree 구조화면에서 마우스의 오른쪽 버튼을 누르면 다음과 같은 팝업 메뉴가 나타난다.

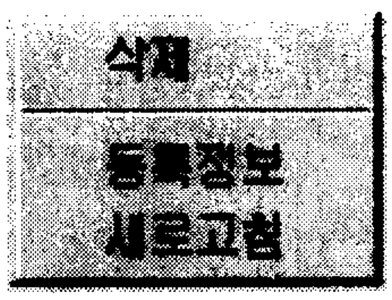


그림 22.Tree 구조의 팝업메뉴

- ◆ Tree 구조의 요소 삭제는 **삭제**를 선택한다. 이때 요소타입이 Group 이면 하위요소도 삭제된다.
- ◆ Occurrence Indicator 를 바꾸려면 **등록정보**를 선택한다.
- ◆ 요소를 추가하려면 구성요소 화면에서 요소를 선택한 상태에서(마우스를 누른 상태로) Tree 구조 화면으로 잡아끌어 원하는 위치에서 마우스를 놓는다.(Drag & Drop)
- ◆ 이때 나타나는 Drag&Drop 팝업메뉴는 아래와 같다.

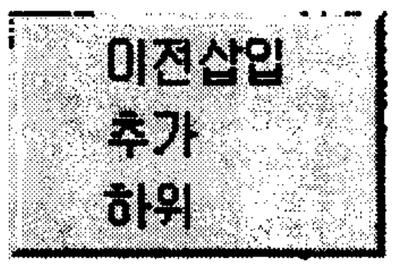


그림 23. Drag&Drop 팝업메뉴

⑤. SDF Import/Export

(1) SDF Export

- ◆ SDF 를 Export 한다는 것은 데이터베이스에 저장되어 있는 SDF 를 Text 파일로 변환한다는 것을 말하며 이렇게 Export 되어진 Text 파일은 다른 DTD Editor 에서 Import 할 목적과 종이에 인쇄할 목적으로 만든다.
- ◆ 그림 18 의 SDF 요소 및 구조 편집 화면에서 **도구 ▷ Import** 메뉴를 선택한다.
- ◆ Export 한 결과가 다음 화면과 같이 나타난다.

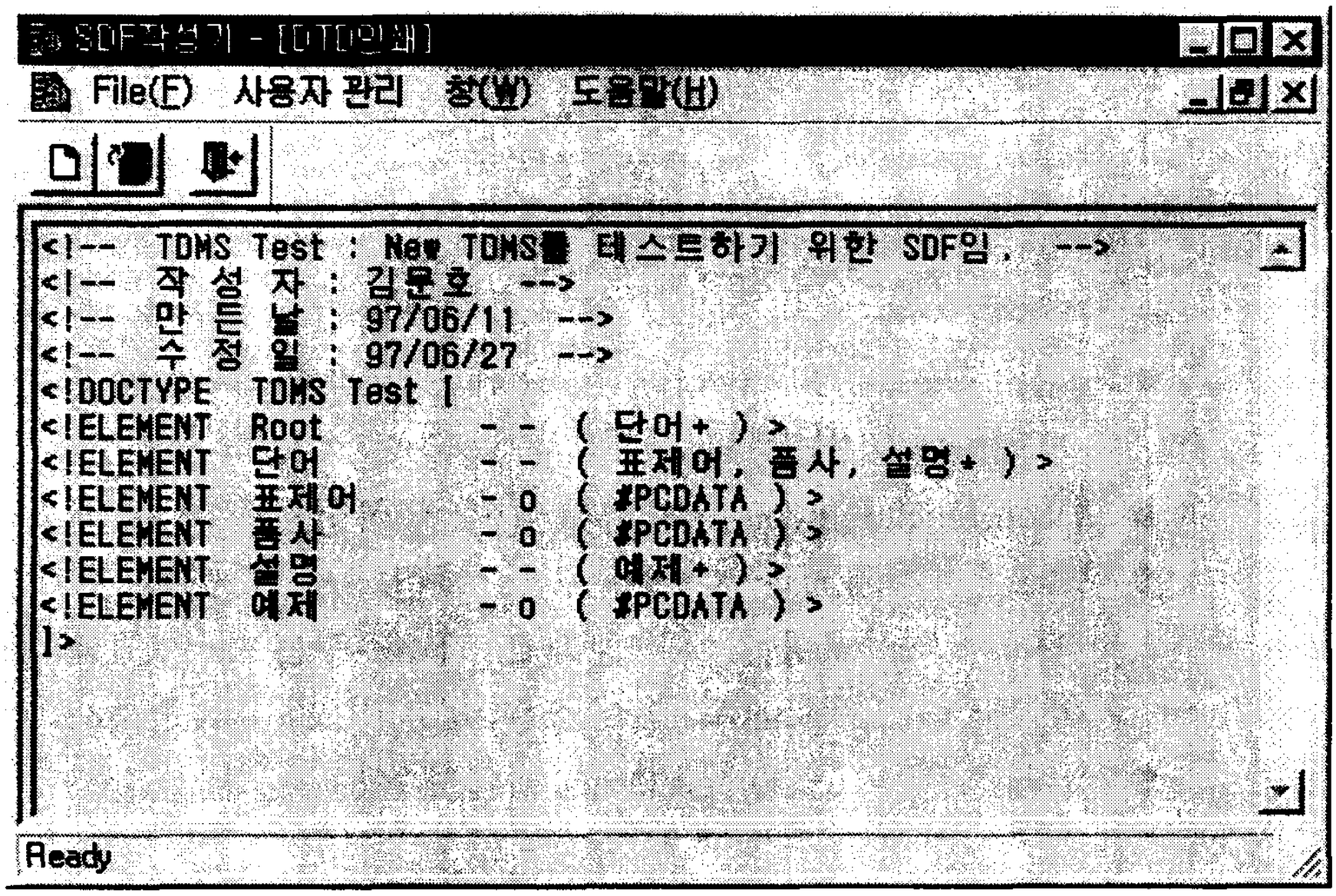


그림 24. SDF Export 화면

◆ 결과를 파일로 저장하려면 File ▷ **저장**을 선택한다.

◆ 결과를 프린터로 출력하려면 **File ▷ 인쇄**를 선택한다.

◆ 이때 인쇄할 프린터를 변경하려면 **File ▷ 프린터설정**을 선택하여 프린터의 등록정보를 변경한다.

⑥. 사용자 관리

◆ TDMS 의 사용자는 3 가지 종류가 있다.

Supervisor : SDF 를 편집하고 사용자를 관리할 수 있는 권한

editier : 사전을 편집할 수 있는 권한. SDF 의 편집은 불가하며 볼수만 있다.

Viewer : 편집은 안되고 보기만 할 수 있는 권한.

◆ Supervisor 의 권한은 데이터의 삭제부터 사용자의 관리까지의 권한이 매우 많으므로 보안에 유의하여야 한다.

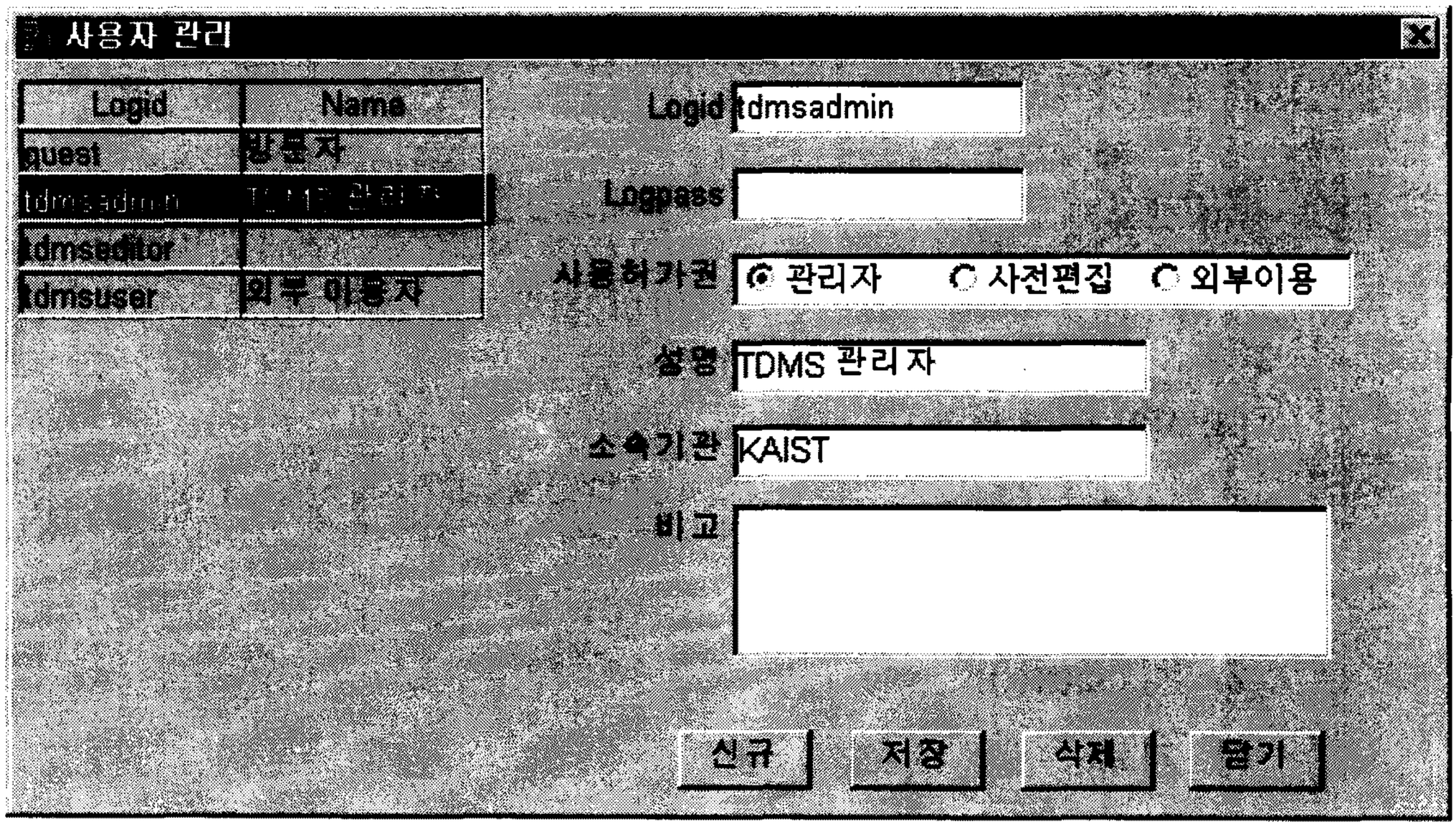


그림 25. 사용자 관리 화면

(1) 신규 사용자 등록

- ◆ 신규 버튼을 누른다.
- ◆ 신규 Logid 를 입력한 다음 Tab 을 누른다.(필수 입력 사항)
- ◆ Logpass 를 신규 Logid 를 입력한 다음 Tab 을 누른다. (필수 입력 사항)
- ◆ 사용허가권을 관리자, 사전편집자, 외부이용자 중에서 선택한 다음 Tab 을 누른다. (필수 입력 사항)
- ◆ 사용자의 이름을 입력한 다음 Tab 을 누른다. (선택 입력 사항)
- ◆ 사용자의 소속기관을 입력한 다음 Tab 을 누른다. (선택 입력 사항)
- ◆ 비고사항을 입력한다. (선택 입력 사항)
- ◆ 저장 버튼을 누른다.

(2) 기존 사용자 편집

- ◆ 편집할 사용자를 화면의 왼쪽에서 선택한다.(Click)

◆ 편집할 내용을 신규사용자 등록과 같은 요령으로 수정항다.

◆ 저장 버튼을 누른다.

(3) 사용자 삭제

◆ 삭제할 사용자를 화면의 왼쪽에서 선택한다.(Click)

◆ 삭제 버튼을 누른다.

(4) 사용자 입력 취소

◆ 신규 버튼을 누르거나 화면 왼쪽에 있는 사용자를 선택한다.

2. SD Editor

가. 개요

SDF Builder에 의해 작성된 SDF를 이용하여 SD(Standard Dictionary:표준전자사전)을 편집하는 프로그램이다.

나. 기능 요약

◆ SDF 선택

◆ 사전 내용 검색

◆ 사전 내용 추가 및 삭제

◆ 사전편집 색상 및 테두리 변경

◆ 하위요소 추가 방법 지정

◆ 사전 Export/Import

다. 사용설명

①. 프로그램 시작

Window95 왼쪽 하단에 있는 **시작**버튼을 누른 후 **프로그램** ▷ **TDMS** ▷ **사전편집기**를 차례로 선택한다.

②. Login 화면

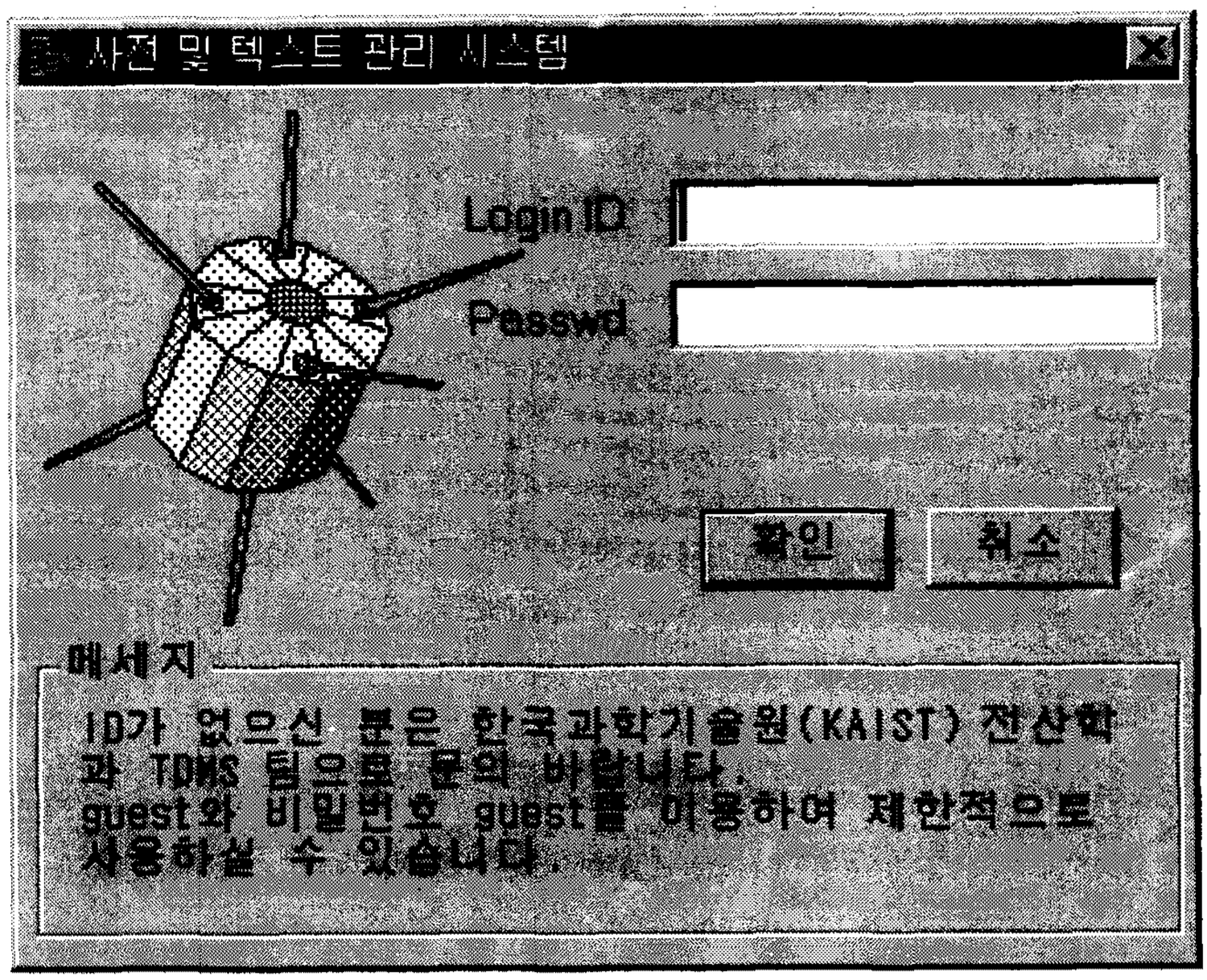
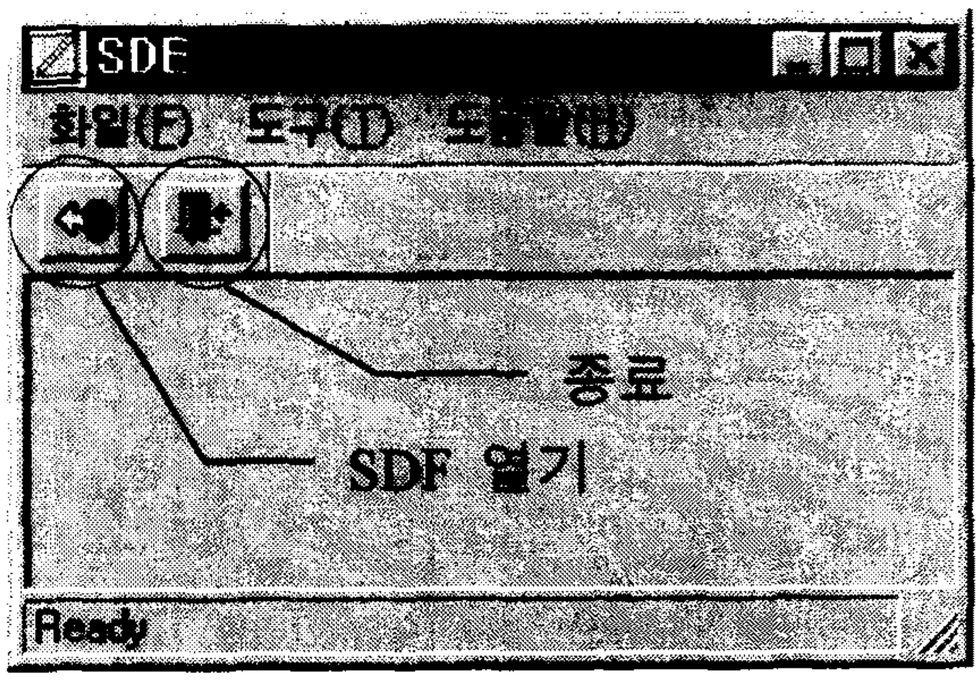


그림 26. Login 화면

- 부여받은 Login ID 와 Passwd 를 입력한 후 확인 버튼을 누른다.
- 이때 입력한 ID 와 Passwd 는 사용 허가권과 연동이 되므로 TDMS Supervisor 이외에는 편집 및 저장등을 할 수 없다.(사용자관리 참조)
- ID 와 Passwd 를 부여받지 않은 사용자는 guest 를 이용하면 된다.

③. 기본화면 설명



SD Editor 기본화면에서는 SDF 열기와 작업 종료를 선택할 수 있다.

(1) SDF 열기

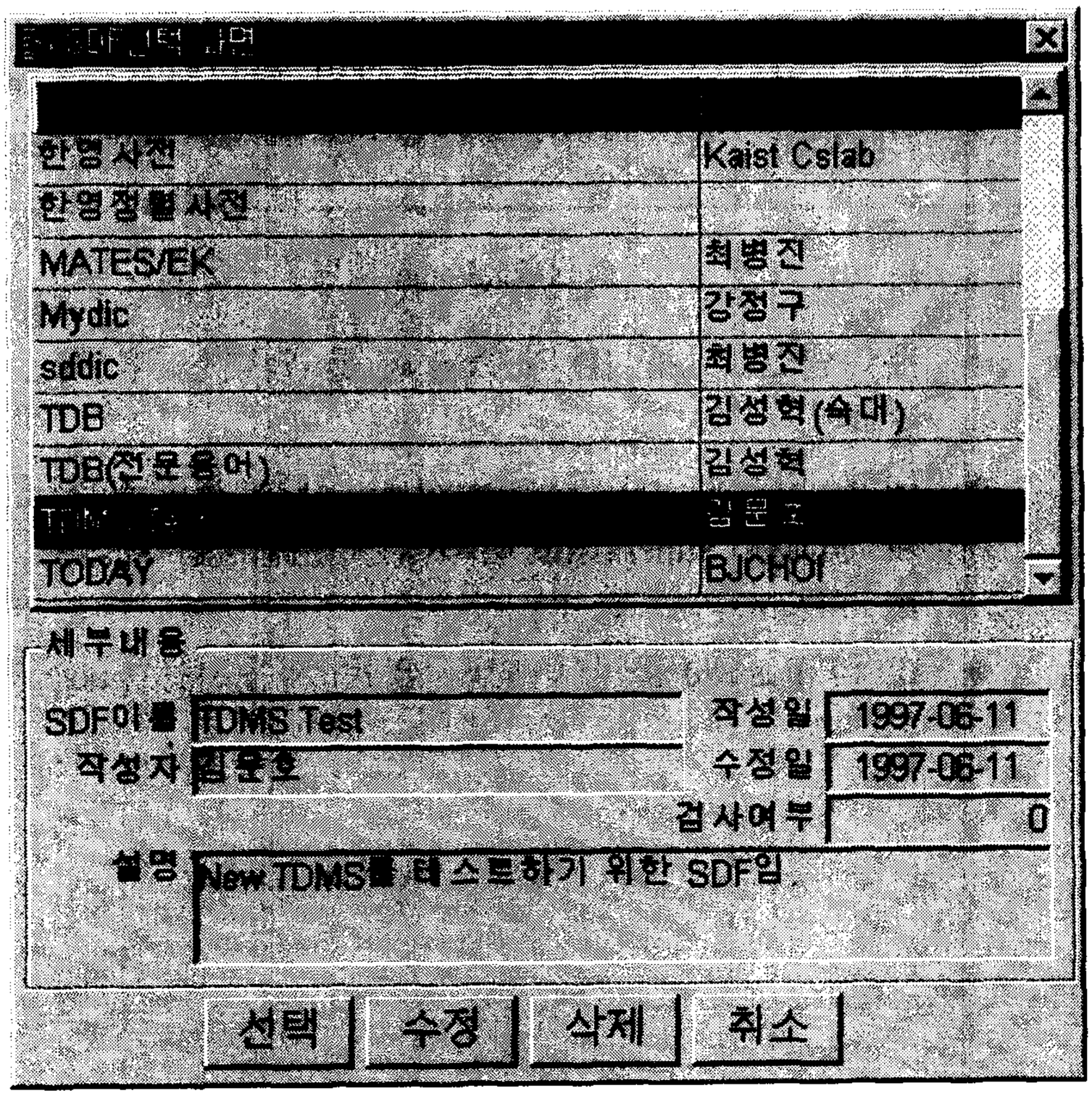


그림 27. SDF 선택화면

- ◆ 작성되어 있는 SDF 정보가 화면에 나타난다.
- ◆ 이때 문자색이 청색인 SDF 는 사전 데이터가 입력되어 있다는 표시이고 문자색이 검정색인 SDF 는 사전 데이터가 입력되어 있지 않다는 표시이다.
- ◆ 사전 데이터가 입력되어 있는 SDF 는 요소 및 구조 편집 시에 제약이 따른다.
- ◆ 원하는 SDF 를 마우스로 선택(Click)하면 SDF 의 상세한 정보가 나타나며 **선택** 버튼을 이용하여 입력할 사전의 SDF 를 선택한다.
- ◆ 취소 버튼은 SDF 선택을 취소하고 SDEditor 기본화면으로 돌아간다.

④. 사전(SD) 편집

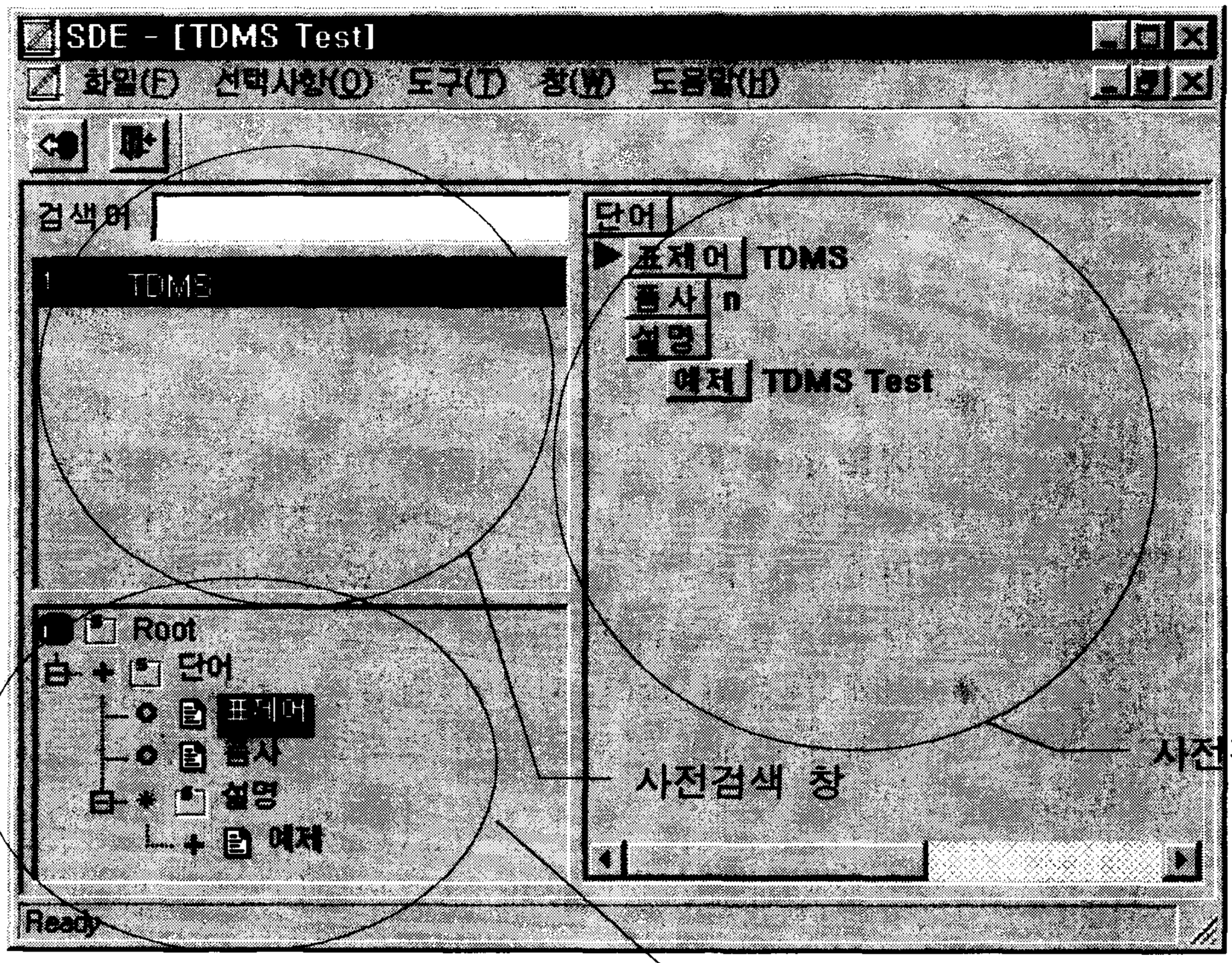


그림 28. 사전 편집 화면

(1) 내용 신규 등록

- ◆ 사전편집 창에서 마우스의 오른쪽 버튼을 눌러 팝업메뉴를 띄운 후 신규를 선택한다.
- ◆ 사전 편집 창에 해당 요소들이 나타난다.
- ◆ 해당 요소들을 마우스나 위아래 화살표를 이용하여 이동하면서 내용을 입력한다.
- ◆ 사전편집 창에서 마우스의 오른쪽 버튼을 눌러 팝업메뉴를 띄운 후 저장을 선택하여 데이터베이스에 저장한다.

(2) 사전내용 검색

- ◆ 사전검색 창에 검색할 단어의 내용을 입력한 다음 Enter를 친다.
- ◆ 입력한 내용과 똑같은 내용을 검색(Exact maching)이면 Ctrl+Enter를 친다.
- ◆ "한국" 부터 "항구"까지 검색하려면 "한국-항구"라고 입력한 뒤 Enter를 친다.
- ◆ 프로그램 기동시에는 구조표시 창의 3번째 요소에서 찾는다. 즉 한국이라고 입력하고 Enter를 치면 표제어가 한국으로 시작하는 내용을 검색한다.
- ◆ 구조표시 창의 요소를 마우스로 2번 누르면(Double click) 이를 검색대상 요소를 지정할 수 있다. 즉 문장을 마우스로 2번 누른 다음 검색할 단어를 검색 단어 창에 한국이라고 입력한 후 Enter를 치면 문장중에 한국으로 시작하는 내용을 검색한다.
- ◆ 검색 내용이 많아 시간이 걸리는 경우에 이를 중단 할 시에는 Cancel 버튼을 눌러 검색을 중단시킬 수 있다.

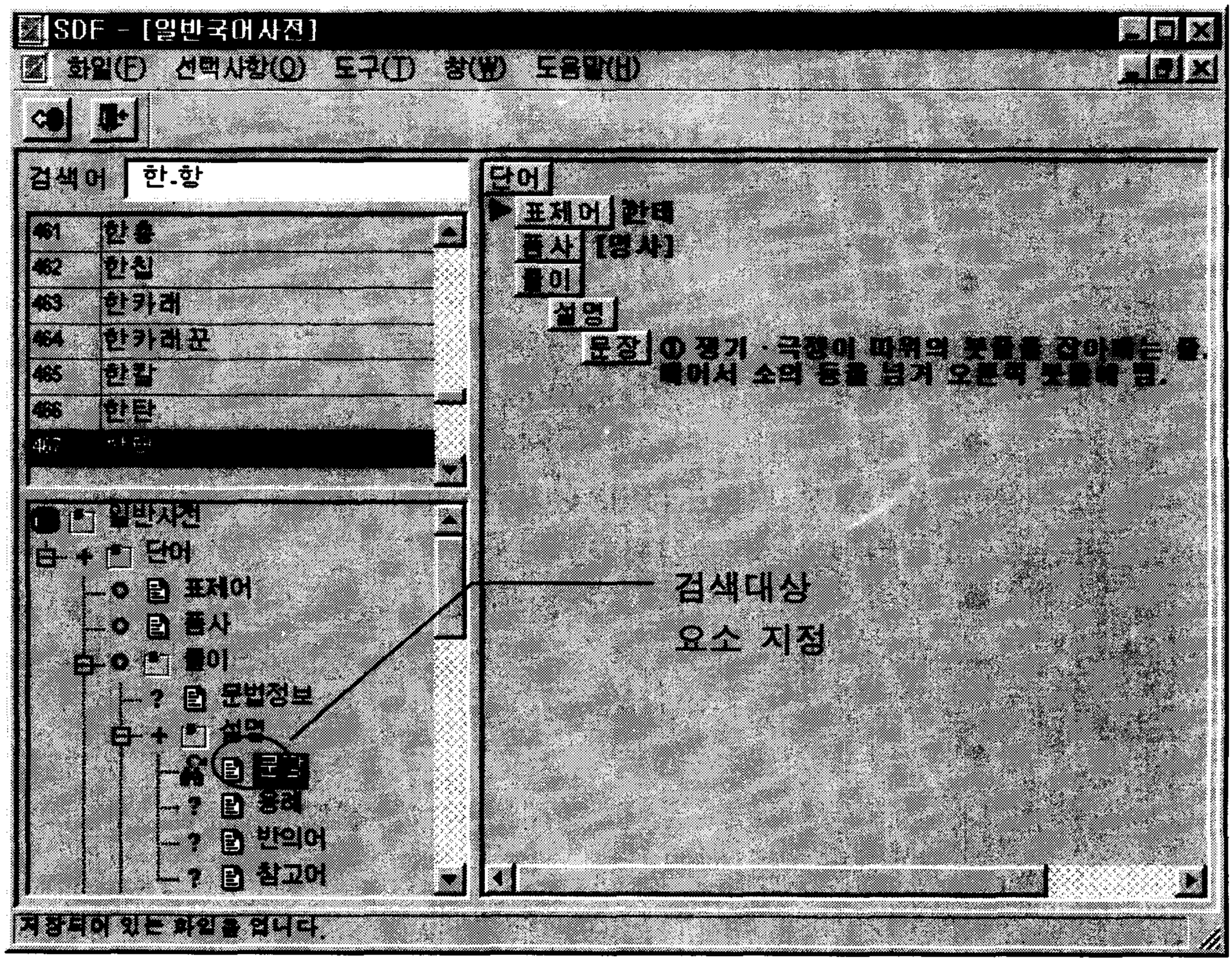


그림 29. 사전 편집 화면

(3) 사전 내용 편집

◆ 위와 같이 원하는 내용을 검색한 뒤 편집할 내용을 마우스로 선택하면 입력되어 있는 정보가 사전편집 창에 표시된다.

◆ 커서를 사전편집 창에 두고 위아래 화살표로 요소들 간의 이동에 따라 구조 표시창의 구조도 같이 이동하는 것을 볼 수 있다. 이것은 사전편집이 현재 구조의 어떤 위치에 있다는 것을 알려주어 편집자가 쉽게 다른 요소들을 추가 및 삭제를 용이하게 한다.

◆ 해당 내용을 마우스의 오른쪽으로 누르면 팝업메뉴가 나타나는데

◆ 내용을 삭제하려면 삭제를 선택하면 된다. 삭제시 해당 내용이 그룹요소이면 하위 내용까지도 삭제를 하며 최상위의 요소를 삭제하면 내용전체(1Record)를 지운다.

◆ 입력 도중 필요한 요소는 구조표시창에서 해당 요소를 선택한 다음 마우스를 누른 상태로 사전편집 창의 적절한 위치로 잡아끈다.(Drag&drop)

◆ 이때 구조표시 창에서 선택한 요소가 하위요소들을 가지고 있는 그룹이라면 아래와 같이 선택한 하위요소 추가 방법에 의해 추가된다.

◆ 입력 도중 필요한 요소는 구조표시창에서 해당 요소를 선택한 다음 마우스를 누른 상태로 사전편집 창의 적절한 위치로 잡아끈다.(Drag&drop)

◆ 이때 구조표시 창에서 선택한 요소가 하위요소들을 가지고 있는 그룹이라면 아래와 같이 선택한 하위요소 추가 방법에 의해 추가된다.

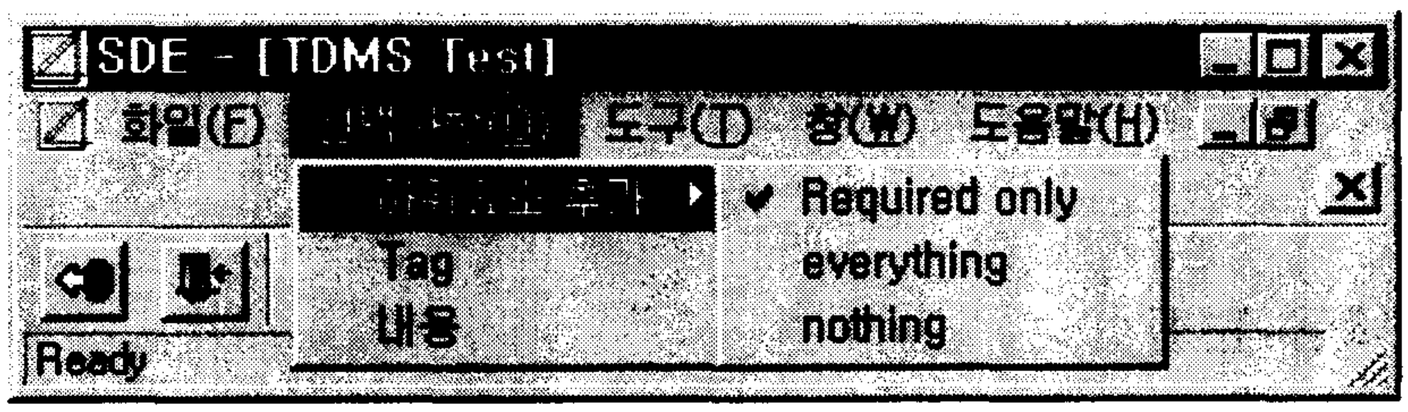


그림 30. 하위요소 추가 메뉴

◆ 즉 Required Only 를 선택하면 Occurence 에서 Required(⊙) 와 Required & Repeatable(⊕)로 표시된 것 만 추가하고 나머지는 추가하지 않는다.

◆ Everything 을 선택하면 Occurrence Indicator 와 관계없이 하위요소를 모두 추가하고, Nothing 을 선택하면 하위요소는 추가하지 않고 자신만 추가하게 된다.

◆ 편집 및 삭제가 완료 되었으면 마우스의 오른쪽 버튼을 눌러 팝업메뉴를 띄운 다음 저장을 선택하여 데이터베이스에 저장한다.

(4) 사전편집 색상 및 테두리 변경

◆ 사전의 내용을 편집하기 좋게 Tag와 내용의 색상과 테두리를 변경할 수 있다.

◆ 메뉴에서 **선택사항 ▷ Tag** (혹은 **내용**)을 선택하면 다음과 같은 화면이 나타난다.

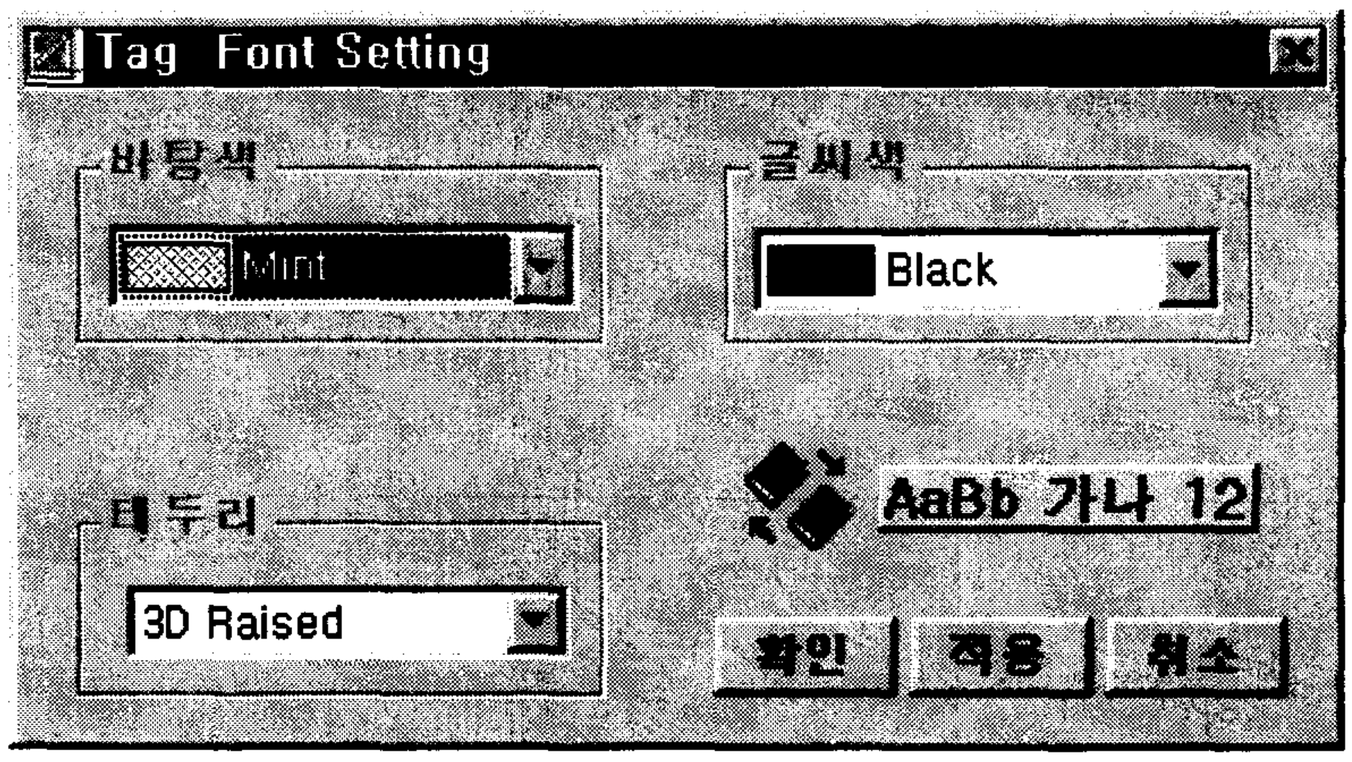


그림 31. 색상 및 테두리 변경 화면

◆ 확인을 누르면 정보가 레지스트리에 입력된 후 색상변경 화면을 닫는다.

◆ 적용버튼을 누르면 사전편집 화면을 바꾸나 레지스트리에 기록하지는 않는다.

◆ 취소 버튼을 누르면 변경을 무시하고 이전 상태로 환원한다.

(5) 사전 Import/Export

◆ 사전검색 창에서 Export 하고자 하는 내용을 검색한다.

◆ 메뉴에서 **도구 ▷Export** 를 선택한다.

- ◆ Filename 을 선택한 다음 저장버튼을 누른다.
- ◆ 내용이 많아 이를 중단할 시에는 Cancel 버튼을 눌러 Export 를 중단할 수 있다.
- ◆ Export 한 파일의 내용은 다음과 같다.

<!-- Export from TDMS 1997/06/30 -->
<!-- 항구 ~ 항구적 (6 Rec.) -->
<!DOCTYPE 일반국어사전 SYSTEM "일반국어.SDF" []>
<일반사전>
<단어>
<표제어> 항구
<품사> [명사]
<풀이>
<설명>
<문장> ① 염전에서 판에 댈 바닷물을 받는 웅덩이.
</설명>
. . . . .
</풀이>
</단어>
</일반사전>

3. 한국어 형태태깅 시스템

한국과학기술원
최기선

여백

3. 한국어 형태태깅 시스템

1장. 서론

1절. 배경

형태소 분석은 최소 의미 단위인 형태소를 추출하는 단계이다. 이는 다시, 한 어절 내에 포함된 가능한 모든 형태소 후보들을 분리한(morpheme segmentation) 후, 형태소 분석용 사전을 검색하여 형태소를 인식하고(morpheme identification), 형태 배열 정보(morphotactic information)을 이용하여 가능한 형태소 열을 구성하는 부분으로 나누어진다.

어떤 형태소가 여러개의 품사를 가질 경우, 그 형태소는 품사 중의성(the ambiguity of part-of-speech)이 있다고 한다. 품사 중의성은 형태소의 주변 문맥 등을 봄으로써 줄일 수 있으며, 품사 중의성을 해소하는 과정을 품사 태깅(part-of-speech tagging)이라고 한다. 자동 품사 태깅 시스템은 약 95%의 정확률을 가지고 있다[8]. 그러나, 많은 자연언어 처리 시스템은 나머지 5%의 오류로 말미암아 문장의 해석에 실패하거나 잘못된 분석을 할 수 있다.

코퍼스를 이용한 자연언어 처리 시스템에는 많은 양의 코퍼스가 절대적으로 필요하다. 그러나, 코퍼스의 양 못지 않게 높은 정밀도도 요구된다. 왜냐하면, 부정확한 자료를 이용하는 시스템의 경우에는 좋은 결과를 기대할 수 없기 때문이다(garbage-in garbage-out)[8].

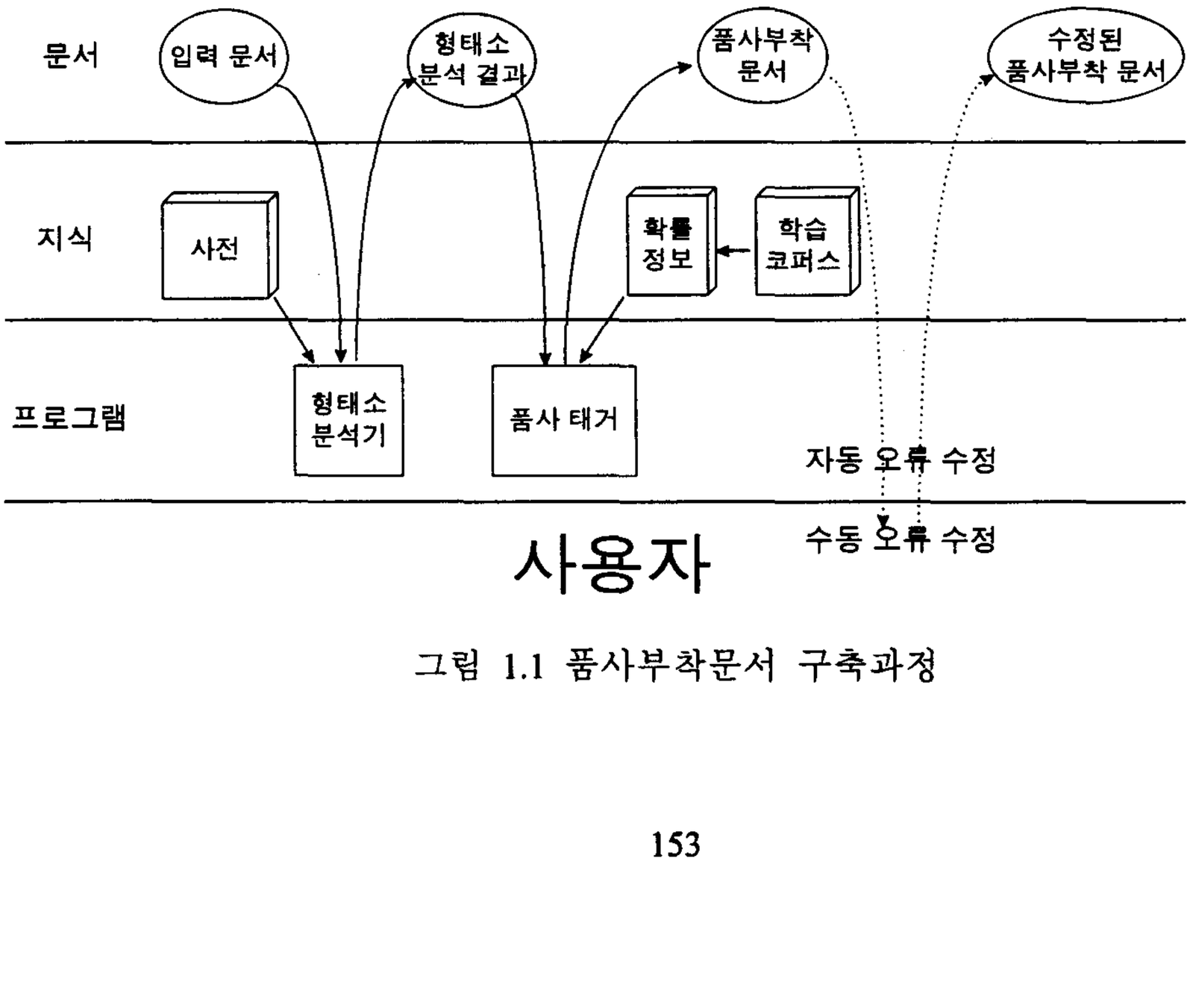


그림 1.1 품사부착문서 구축과정

2절. 문제점 및 동기

일반적인 품사부착 코퍼스 구축 과정을 그림으로 그려보면, 그림 1.1과 같다. 그러나 일반적인 품사부착 코퍼스 구축 과정에는 몇 가지의 어려움이 있다. 첫째, 형태소 분석시 사전 항목의 미흡, 신조어 및 고유명사의 잦은 발생 등으로 인한 미등록어는 형태소 분석 결과의 질을 저하시키고, 이후 분석 단계에 영향을 준다. 둘째, 자동 품사 태깅의 결과에 존재하는 오류를 수정하는 과정의 어려움이다. 수동 오류 교정은 오류를 찾는 일과 찾은 오류를 수정하는 일로 이루어진다. 따라서, 수동 오류 교정에는 막대한 물적, 인적 비용이 소요될 것이고, 수동 오류 교정 후의 코퍼스가 오류를 포함하는 경우도 발생할 것이다. 오류를 자동으로 수정하는 방법이 연구되고 있는데 이 때에는 자동 수정 후에도 오류가 존재한다는 문제점이 있다[18][28].

3절. 목표 및 연구 범위

그림 1.1에서 형태소 분석기는 사전 정보를 이용하여 입력 문서에 대한 형태소 분석 결과를 출력한다. 태거는 학습 코퍼스에서 추출한 확률 정보를 이용해서 형태소 분석 결과로 부터 품사부착 문서를 출력한다. 사용자는 품사부착 문서를 보고 오류를 수동으로 수정한다. 이 때, 형태소 분석과 품사 태거는 각각 오류를 포함하게 되고, 이러한 오류들은 수동 오류 교정의 양을 증가시킨다.

본 연구에서는 형태소 분석과 품사 태깅을 통합하여 사용자에게 편리한 환경을 제공하고, 입력 문서에 존재하는 미등록어를 형태소 분석시 사용자 인터페이스를 통하여 온라인으로 입력함으로써 형태소 분석 결과의 중의성을 줄이는 방법을 사용하였다.

그리고, 품사부착 코퍼스를 구축할 때, 비용을 줄이고 정확도를 높일 수 있는 방법으로, 품사 태깅 오류를 쉽게 검색, 교정할 수 있는 오류 인식, 교정 환경을 구축하였다. 또, 태깅 오류 교정에 사용된 지식의 축적 및 학습 결과를 태깅 시스템에 반영하여 시스템의 정확도를 향상시킬 수 있는데, 이를 통해 학습 코퍼스의 정확도를 향상시켜서 입력 문서에 대한 자동 품사부착 문서의 정확도를 향상시킬 수 있다.

본 보고서의 구성은 다음과 같다. 2.1 절과 2.2 절에서는 형태소 분석과 품사 태깅에 대해서 살펴보고, 2.3 절에서는 품사 태깅후의 자동 오류 수정에 관한 기존 연구들을 알아 본다. 이어서 3 장에서는 새롭게 제시하는 워크벤치에 대해서 설명하고, 4 장, 5 장에서 실험, 결과가 이어진다.

2 장. 관련연구

본 장에서는 자연 언어 처리의 초기 단계인 형태소 분석과 품사 태깅에 대해서 살펴보고, 본 연구와 관련이 있는 자동 품사 태깅 후 자동 오류 수정에 관한 기존 연구들에 대해서 살펴본다.

1 절. 한국어 형태소 분석

형태소 분석이란 주어진 문장의 최소 의미 단위인 형태소를 추출하는 단계이다. 한 어절 내에 포함된 가능한 모든 형태소를 분리하고(morpheme segmentation), 분리된 형태소들이 정당한 배열인지를 검사하여 한 어절의 구조를 파악하고 (morphotactics recognition), 정당한 형태소 열에 대한 가능한 구문 및 의미 정보의 생성(구문 및 의미 정보 등)(information generation) 등을 한다. 한국어 형태소 분석기(morphological analyzer)는 한국어 처리를 하기 위해서 하나의 어절이 가지는 형태소를 추출하고, 각 형태소에 관련된 정보(품사, 원형, 의미 등)가 무엇인가를 찾아내는 시스템을 말한다. 여기서 추출된 정보는 품사 태깅을 통하여 자연언어 처리의 다음 단계인 구문 해석 단계에서 사용된다. 따라서, 형태소에서 얻어진 정보의 양과 질에 따라서 다음 단계의 해석에 큰 영향을 준다[3].

일반적으로 한국어 형태소 분석기는 크게 다음의 네 가지 과정을 거친다. 첫째, 가능한 형태소를 찾아주는 형태소 분리(morpheme segmenting)과정, 둘째, 두 개의 형태소가 결합할 때 발생되는 여러 불규칙 현상 및 음운 현상에 의 해서 변형된 형태소를 복원하는 불규칙 현상 처리(base form recovering) 과정, 셋째, 찾아진 형태소들이 올바른 한국어 어절을 형성하는지의 유무를 검사하는 형태소 배열규칙 검사(morphotactics checking) 과정, 넷째, 사전에 등록되지 않은 단어를 처리하는 미등록어 처리(unknown word processing) 과정이다[8].

한국어 형태소 분석 방법은 최장일치법, head-tail 분리 기억 방법[19], tabular 파싱에 기반한 방법[3][15], 두 단계 모델에 기반한 방법[13], 음절정보에 기반한 방

법[1] 등 여러가지 방법들이 제시되었다. 이들 연구의 대부분은 어떻게 한국어에 대한 형태소를 분석할 것인가에 촛점을 맞추어 왔다[8]. 각각의 처리 방법에 대한 구체적인 설명은 [4]를 참고하기 바란다.

2절. 한국어 품사 태깅

품사 태깅은 여러가지의 품사를 가진 단어가 문장 속에 나타났을 때에 단어의 품사를 올바르게 결정하는 것이라고 할 수 있다. 이를 좀 더 공식적으로 설명하면, 주어진 문장 $W = w_1w_2...w_n$에 가장 적합한 태그열 $T = t_1t_2...t_n$을 대응시키는 것이다. 즉, 품사 태깅은 문장 W를 입력으로 해서 태그열을 출력하는 함수 $\phi : W \rightarrow T = \phi(W)$로 정의할 수 있다. 따라서, 태깅 문제를 푸는 것은 우리가 사용하는 모든 W에 대해서 가장 좋은 성능을 가진 $\phi(W)$를 찾는 것이다[5].

품사 태깅 방법은 크게 규칙에 의한 방법[21], 통계적인 처리에 의한 방법[16][24][25][27][30] 신경망에 의한 방법[20][29], 규칙과 통계의 혼합 방법[18][28] 등이 있다. 대부분의 품사 태깅 시스템은 어떤 한 단어가 주어졌을 때, 그들의 품사는 사전이나 형태소 분석에 의해서 결정되며, 국부적인 문맥(local context)을 이용하여 정확한 품사를 찾으려고 한다는 공통점이 있다[5]. 이 들 각 방법에 대한 구체적인 설명은 [5]를 참조하기 바란다.

3절. 품사 태깅 오류의 자동 발견

태깅 오류를 자동으로 발견하려면 오류가 잘 발생하는 환경을 새로이 모델링하여야 한다. 이와 같은 연구가 [26]에 의해서 부분적으로 이루어졌다. Foster의 기본 가정은 "태깅 오류는 서로 경쟁하는 태그(예를 들면 영어에서 명사 대 형용사, 과거분사 대 과거 등)들이 비슷한 확률을 가지게 될 때 발생된다."는 것이다. 서로 경쟁하는 태그들이 정확한 품사부착 환경에서는 확률값이 크고, 그렇지 않은 환경에서는 아주 비슷하다는 것이다. 따라서 서로 경쟁하는 태그들에 대해서 확률값의 차이가 어느 값 이하가 될 때 오류가 발생할 수 있는 환경으로 간주한다. 여기서 확률값의 차이가 작을 수록 오류 환경을 찾는 횟수는 작지만 오류 환경을 정확하게 찾을 수 있는 가능성은 대단히 높다. Foster는 이와 같은 알고리즘을 적용할 경우에 전체 오류의 1%를 제외하고는 모두 찾을 수 있었고 전체 코퍼스의

약 15%를 오류 환경으로 표시했다[5].

본 절에서는 통계적인 방법으로 자동 품사 태깅한 결과의 오류를 규칙을 이용하여 자동으로 수정하는 방법과 관련된 연구들을 살펴보겠다.

1. 변형 규칙 기반 품사 태깅[18]

[22]의 연구를 한국어에 적용한 연구로서, 태깅에 필요한 언어 지식을 학습 코퍼스로부터 자동 추출하여 소량의 규칙 집합으로 품사 태깅을 수행한다. 변형 규칙 학습 과정의 일반적인 과정은 다음과 같다.

- 초기 태거를 이용하여 학습 코퍼스를 태깅
- 학습 코퍼스의 태깅 결과와 주석 달린 코퍼스의 분석 결과를 비교하여 혼동 행렬 작성
- Scoring 함수와 규칙 틀을 이용하여 혼동 행렬의 오류를 가장 많이 수정할 수 있는 규칙 추출
- 추출된 태깅 규칙을 학습 코퍼스에 적용
- 추출된 규칙 저장
- 찾아진 규칙의 오류 수정 빈도가 임계값보다 작을 때까지 앞의 네 과정 반복

변형 규칙 학습을 위해 사용된 규칙 틀의 형식은 다음과 같다.

[위치 정보, 참조 대상]{[OR, AND][위치 정보, 참조 대상]}

위치 정보 : -n, +m

참조 대상 : 어절 태그, Head, Tail, 어휘

이 연구에서는 101 개의 규칙을 사용하여, 91.6% 정확률의 초기 태거 결과를 94.8%로 높였다.

2. 규칙 기반 오류 수정[28]

이 연구는 [22]의 품사 태깅 모델에 기반한 두 단계 학습 - 통계적인 학습, 규칙 기반 학습 - 모델이다. 규칙 기반 학습은 통계적인 품사 태깅 결과 중 오류들에 대해서만 학습을 한다. 통계적인 품사 태깅의 모델로는 은닉 마르코프 모델을 사용하였다.

오류 수정 규칙의 학습은 다음과 같다. 우선 스키마와 규칙의 형식을 정의한다.

스키마는 규칙을 적용할 수 있는 문맥(context)의 위치를 나타내는 기호이다. 규칙의 형식은 다음과 같다.

[현 형태소][현 태그];([스키마]:[문맥 태그 또는 문맥 형태소 정보])*

→[새 형태소][새 태그]

위의 규칙에서 현 형태소와 현 태그는 규칙을 적용할 수 있는 어절의 형태소와 태그 정보이고, 문맥 태그 또는 문맥 형태소 정보는 스키마가 지정하는 위치의 정보이다. 규칙의 의미는 스키마가 지정하는 위치의 내용이 '문맥 태그 또는 문맥 형태소 정보'의 내용과 일치하면 현 형태소나 태그를 새 형태소나 태그로 변경하는 것이다.

스키마와 규칙의 형식을 정의한 후, 정의한 스키마와 규칙에 태그 집합과 어휘 집합의 내용을 할당해서 규칙 후보 집합을 구성한다. 규칙 후보의 규칙들을 오류가 포함된 품사부착 코퍼스에 적용해서 가장 오류를 많이 수정하는 규칙을 추출한다. 이 추출 과정을 오류가 임계값 이하가 될 때까지 반복한다. 이 방법의 단점은 이 규칙 추출 과정의 비용이 크다는 점을 들 수 있다.

실험은 70,000 어절의 코퍼스를 모아서, 약 70%의 코퍼스를 은닉 마르코프 모델의 학습에 사용하였고, 10,000 어절(약 15%)의 코퍼스로부터 445 개의 오류 수정 규칙을 추출하였다. 그리고, 나머지 10,000 어절에 대해서 실험을 하였는데, 은닉 마르코프 모델만을 사용했을 때는 정확률이 90.1%였고, 오류 수정 과정을 거치면 정확률이 92.4%로 증가하였다.

3장. 연구내용

1절. 품사 태깅 오류 원인

본 절에서는 품사 태깅 오류를 일으키는 주요 원인들을 살펴본다.

1. 사전 미등록어

대부분의 한국어 형태소 분석기는 사전을 기반으로 이루어지고 있다. 사전을 이용할 경우 미등록어란 사전에 등록되지 않은 말로 정의할 수 있다[6]. 대부분의 형태소 분석에서 미등록어는 고유명사 혹은 명사로 가정한다. 그 이유는 개방어

(open class word)에 속하는 대부분의 단어가 이 명사류에 속하기 때문이다. 그러나, 명사류가 아닌 다른 부류의 단어들도 미등록어로서 쉽게 접할 수 있다. 특히 한국어와 같은 형용적 표현이 발달된 언어에서는 더욱 더 그와 같은 현상을 자주 접할 수 있다[6].

기능어(조사, 어미) 이외의 폐쇄어(closed class word)에 대한 가정 없이 - 보다 현실적이다 - 미등록어가 포함된 어절을 분석하려면 모든 가능한 분석 단위들을 형태소로 가정하여야 한다. 미등록어를 포함하는 어절을 분석할 때, 어절의 오른쪽부터 분석을 하면서, 조사나 어미를 찾아내고, 그 앞부분의 어간에 해당하는 부분에 가능한 모든 태그를 부여해야 한다.

예를 들어, '좌우명이나'라는 어절에서 '좌우명'이 미등록어라면, '좌우명이나'는 '좌우명이나', '좌우명+이나', 좌우명+이+나', '좌우명이나+아' 등으로 분리할 수 있고, 각각에 대해 개방어에 속하는 모든 품사들을 할당해야 한다. 만약, 개방어에 속하는 품사가 10 개라면, '좌우명이나'의 형태소 분석 결과는 모두 40 개가 넘게 된다. 또, 미등록어에 대한 형태소 정보가 학습 코퍼스에 없기 때문에, 형태소 확률은 0 에 가깝고, 단지 태그열의 확률만으로 태깅을 하게 된다. 따라서, 미등록어가 존재하는 어절에 대해서는 옳은 태깅 결과를 기대하기가 어렵다.

만약, 위의 예에서 '좌우명'이 사전에 등록이 되어있다면, '좌우명이나'의 형태소 분석 결과는 3-4 개 정도가 되고, 품사 태깅이 옳게 될 확률은 높아지게 된다.

2. 마르코프 가정

태깅 문제는 다음과 같이 정의 할 수 있다.

$$\phi(W) = \arg\max_{T} P(T|W) \qquad (3.1)$$

$$= \arg\max_{T} \frac{P(W|T)P(T)}{P(W)} \qquad (3.2)$$

$$= \arg\max_{T} P(W|T)P(T) \qquad (3.3)$$

단, $W = w_1 w_2 ... w_n$, $T = t_1 t_2 ... t_n$, $\arg\max_{T} P(x)$ 는 확률값 $P(x)$를 최대로 하는 T 를 구하는 것을 의미하고, wi 는 i 번째 위치한 단어이고, ti 는 그 단어에 해당하는 품사이다. 식 3.3 에 사용되는 확률값을 직접 구하는 것은 파라미터가 너무 많고, 방대한 계산량을 요구하기 때문에 $P(W|T)$와 $P(T)$를 각각 다음과 같이 근사하는데,

이때 사용되는 가정이 마르코프 가정이다[23].

$$P(W|T) \cong \prod_{i=1}^{n} P(w_i|t_i) \qquad (3.4)$$

$$P(T) \cong \prod_{i=1}^{n} P(t_i|t_{i-h,i-1}) \qquad (3.5)$$

식 3.4 와 3.5 를 3.3 에 적용하면, 다음과 같이 된다.

$$\phi(W) = \arg\max_{T} \prod_{i=1}^{n} P(w_i|t_i) P(t_i|t_{i-h,i-1}) \qquad (3.6)$$

식 3.6 에서는 현재 어절에 대해서는 형태소 정보를 사용하지만, 주변 어절에 대해서는 품사 정보만 사용한다. 예를 들어, '빨리 커서 어른이 되고 싶다'라는 문장에서 '되'의 품사는 동사이고, '어른이'의 '이'의 품사는 보격조사이다. 그러나, 태깅식에서는 '이'의 품사를 결정할 때 '되'의 품사가 동사라는 것에만 관심을 갖고 그 동사의 형태소가 '되'라는 것에는 관심을 갖지 않기 때문에 일반적으로 동사 앞에서 '이'가 많이 사용된 품사를 '어른이'의 '이'의 품사로 결정하게 된다.

결국 '이'가 동사 앞에서 보격 조사보다는 주격 조사로 많이 사용되므로 주격 조사로 품사를 결정한다.

2 절. 미등록어의 처리

대부분의 형태소 분석기들은 미등록어 처리를 한다. 그러나, 입력어절이 등록어들로 분석이 가능할 때에는 등록어들만의 분석 결과를 출력한다. 만약, 등록어들로 분석이 가능한 어절에 대해서도 미등록어 처리를 하면, 모든 어절의 형태소 분석 결과의 수가 많아지게 된다. 따라서, 등록어들만으로 분석을 하지 못한 경우에 미등록어 처리를 하게 되는 것이다.

본 워크벤치에서는 형태소 분석을 할 때, 등록어들만으로 분석을 하지 못하면, 그 어절에 미등록어가 포함되어 있다고 간주하고, 이러한 어절들을 사용자에게 제시한다. 그리고, 사용자가 사전 관리기로 미등록어라고 인정되는 형태소들을 사전에 삽입한 후, 사용자에게 제시한 어절들을 다시 형태소 분석한다.

기존 형태소 분석기에서 미등록어 처리 방법은 미등록어가 발생했을 때 발생한 미등록어에 대한 가능한 품사를 모두 부여하므로 분석 결과의 중의성이 증가하게 되고, 이 중의성의 증가는 품사 태깅의 정확률을 떨어뜨리게 된다. 그러나, 본 시스템에서 제시하는 방법을 사용하면, 대부분의 미등록어는 사전에 존재하므로, 과다한 후보가 출력되는 일을 방지할 수 있다. 따라서, 태깅 결과도 훨씬 정확해진다. 그리고, 다음에 같은 단어가 발생했을 때 기존 형태소 분석방법에서는 다시 미등록어 처리를 해야 하지만, 본 방법에서는 그 단어들이 사전에 존재하므로 미등록어 처리과정을 거치지 않아도 분석에 성공하게 된다.

본 시스템의 미등록어 추정 과정에서 오류가 발생하는 경우는 두 가지가 있을 수 있다. 첫째는 미등록어가 아닌데도 형태소 분석기가 분석을 제대로 못해서 미등록어로 제시되는 경우이고, 둘째는 미등록어가 있는데도 등록어들로서 분석 성공하여 미등록어 제시를 안하는 경우이다.

둘째 오류의 예로서, '레이져 빔[beam]을'이라는 어절에서 '빔'이 미등록어인 경우에도 '빔을'을 '비[empty](형용사)+ㅁ(명사형 전성어미)+ㄹ(관형형 전성어미)'로 등록어들만으로 분석이 가능하기 때문에 미등록어로서 제시를 안하게 된다.

3절. 오류 수정

어떤 방법으로 태깅을 하더라도 결과에는 항상 오류가 포함된다. 그리고, 이러한 오류를 수정하려면 막대한 수작업이 필요하게 된다. 따라서, 시스템이 자동으로 오류를 찾아주고 고쳐주면, 큰 도움이 될 것이다. 이 기능은 이러한 목적으로, 품사부착 문서를 보고 오류라고 예상되는 분석들에 대해 사용자에게 대안을 제시한다.

자동 오류 지적과 대안 제시에는 규칙과 수동 수정 로그가 사용되는데 이것들에 대해서 살펴보겠다.

1. 규칙에 의한 방법

이 방법은 오류인 어절에 대한 규칙과 대안을 기술하여, 그 규칙과 일치하는 어절이 발견되면, 그 규칙에 맞는 대안을 제시하는 것이다. 이 방법에서 사용되는 규칙의 형식은 그림 3.1 과 같다.

<수정 조건 형태소><수정 조건 품사>)* / 수정할 형태소
또는 품사의 위치 / 수정 후의 형태소 또는 품사

그림 3.1 자동 오류 지적 규칙의 형식

그림 3.1 에서 <수정 조건>은 어떤 형태소와 품사열에 대해서 규칙을 적용할 것인가를 나타내고, <수정할 형태소, 품사의 위치>는 <수정 조건>에서 몇번째 것들을 수정할 것인가를 나타내고, <수정 후의 형태소, 품사>는 <수정할 형태소, 품사의 위치>에서 지정한 위치의 형태소나 품사가 어떻게 수정될 것인가를 나타낸다.

<수정 조건>에는 다음의 네가지 연산자를 사용할 수 있다.

- Don't Care(*) : 이 연산자의 항목은 모든 형태소 또는 태그와의 매칭에 성공한다.
- OR(|) : 이 연산자가 포함된 항목은 이 연산자로 연결된 항목들 중 최소한 하나와 일치하면 매칭에 성공한다.
- Closure(+) : 이 연산자가 끝에 나타난 항목은 '+'앞의 내용까지만 일치하면 매칭에 성공한다.
- NOT(!) : 이 연산자가 맨 앞에 나타난 항목은 '!' 이후의 내용과 일치하지 않으면 매칭에 성공한다.

3.1.2 에서 살펴본 오류의 예 중에서, "'되다'(동사(pvg)) 앞에서 주격조사(jcs)처럼 사용된 조사는 모두 보격조사(jcc)이다"라는 것을 처리하기 위한 규칙은 '* jcs 되 pvg / 2 / jcc'가 된다.

본 시스템에서 실험에 사용한 규칙은 사용자가 직접 기술할 수 있다. 2 장의 관련연구에서 살펴본 자동 규칙 습득 방법을 사용하여서 얻은 규칙을 사용할 수도 있다.

2. 수동 수정 로그를 이용하는 방법

규칙으로 찾을 수 없는 오류들은 사용자가 지적해서 고치게 되는데, 이 오류들과 수정결과들을 모아둬서 다음에 오류 지적시에 이용한다.

수정 로그는 오류 태그열과 대안 태그열로 구성된다. 예를 들어 동작성 명사로 분석된 '다운'을 '답(형용사화 접미사)+ㄴ(형용사형 전성어미)'로 바꾸는 수정 결

과 데이터는 '다운/ncpa 답/xsm+ㄴ/etm'이 된다. 이 수정 결과 데이터를 이용해서 '사람/ncn+다운/ncpa', '학교/ncn+다운/ncpa' 등의 오류를 지적하고 대안을 제시할 수 있다.

수정 로그도 일종의 규칙으로 볼 수 있는데, 앞에서 정의한 규칙은 여러 어절에 걸쳐서 오류를 검색하는데, 수정 로그는 한 어절에 대해서만 적용된다.

워크벤치가 지적하지 못한 오류에 대해서는 사용자가 직접 수정을 하게 된다. 워크벤치는 사용자로부터 옳은 입력을 받으면, 오류와 옳은 분석 결과를 비교해서 수정 로그의 형식으로 변환해서 즉시, 현재 텍스트에 대해서 오류들을 검사한다. 그리고, 사용자가 입력한 분석 결과 중 미등록어가 있으면, 사전에 자동으로 추가하고, 없으면 수정 로그 데이터베이스에 추가한다.

동일 문서내에서 비슷한 오류들이 자주 발생하므로, 이러한 피드백을 반영하여서 많은 양의 오류들을 수정할 수 있다.

시스템이 제시한 오류와 대안에 대해서 사용자는 다음의 세가지 반응을 취할 수 있다.

- 수정 - 시스템이 제시한 오류와 대안이 모두 옳을 경우, 사용자는 시스템이 제시한 오류를 시스템이 제시한 대안으로 바꾸기를 원하게 된다. 이 반응을 취하면, 시스템은 제시한 오류 분석을 제시한 대안 분석으로 바꾼다.
- 수정 안 함 - 시스템이 제시한 오류가 실제로는 오류가 아닌 경우에 사용자는 시스템이 제시한 오류 분석을 바꾸지 않기를 원하게 된다. 이 반응을 취하면, 시스템은 아무 것도 바꾸지 않는다.
- 사용자 입력 - 시스템이 오류는 옳게 지적했는데 제시한 대안이 옳지 않을 경우에 사용자는 지적된 오류를 대안이 아닌 다른 분석으로 바꾸기를 원하게 된다. 이 반응을 취하면, 수동 수정 모드가 되면서 사용자로부터 새로운 분석 결과를 입력받고, 그 결과에 미등록어가 있으면 사전에 추가하고, 없으면 수정 로그에 넣는다.

4장. 워크벤치 구현

1절. 형태소 분석과 태깅 워크벤치의 개념

본 논문에서 워크벤치는 작업환경의 의미로 사용하였다. 즉, 형태소 분석과 품사 태깅과 관련된 작업들을 수행하는 과정에서 사용자에게 편리함을 제공할 수 있는

작업환경을 의미한다. 본 워크벤치의 전체 구성도는 그림 4.1 에 나타나 있다.

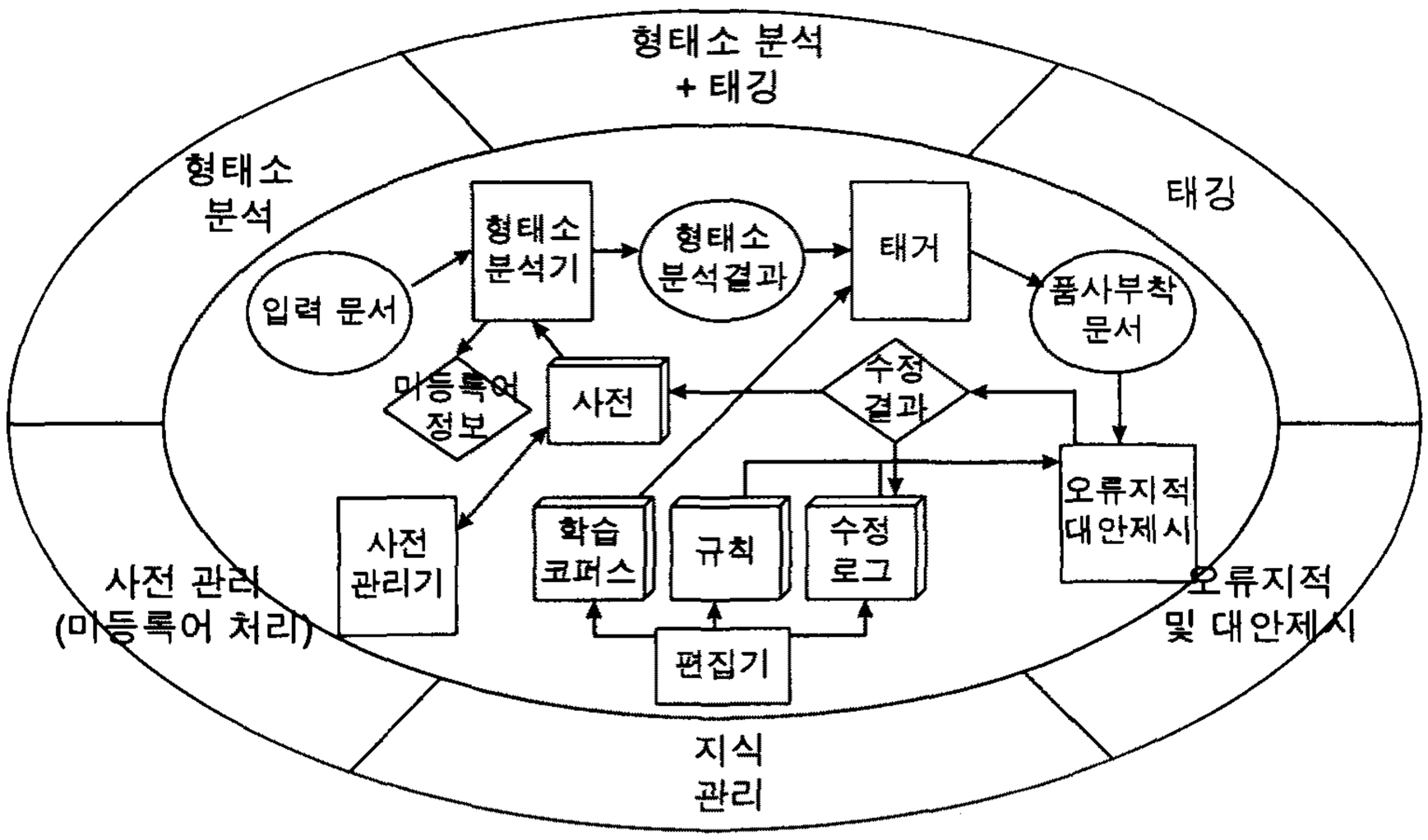


그림 4.1 워크벤치 전체구조

그림 4.1 에서 안쪽 원과 바깥쪽 원 사이의 내용은 사용자가 시스템에 발생시키는 이벤트들이고, 안쪽 원의 사각형은 워크벤치를 구성하는 프로그램, 타원은 문서, 마름모는 형태소 분석과 품사 태깅 중 생기는 정보, 육면체는 시스템에서 사용하는 정보를 각각 의미한다.

워크벤치를 통해서 품사부착 문서를 구축하는 과정은 다음과 같다.

1. 1 차 형태소 분석 & 미등록어 제시 - 문서를 형태소 분석할 때, 미등록어를 포함한다고 짐작되는 어절들을 사용자에게 제시한다.
2. 미등록어 사전 입력 - 사용자가 미등록어들을 사전 관리기를 통해서 사전에 추가한다.
3. 2 차 형태소 분석 - 시스템이 미등록어를 포함한다고 제시한 어절들에 대해서 2 차 형태소 분석을 한다. 이 때, 미등록어들을 사용자가 사전에 추가하였으므로, 형태소 분석시 미등록어 처리를 하지 않아도 된다.
4. 자동 품사 태깅 - 미등록어를 제거한 형태소 분석 결과를 이용해서 자동 품사 태깅을 한다.
5. 오류지적 및 대안제시 - 규칙과 수정 로그를 이용해서 품사 태깅 결과 중 오류를 찾고 대안을 제시한다.

6. 사용자 반응 - 제시한 오류와 대안의 옳고 그름에 따라 사용자의 응답을 받아들여 적절히 동작한다.
7. 5-6 의 과정을 반복하면서, 규칙과 이전 문서까지의 수정로그를 이용해서 자동으로 오류들을 수정한다.
8. 수동 수정 - 시스템이 지적하지 못한 오류에 대해서 사용자로부터 수정 결과를 입력받는다.
9. 피드백 반영 - 수정한 내용을 수정 로그의 형식으로 바꾸어서 전체 품사부착 문서에 대해 오류지적, 대안제시의 과정을 수행하고, 사용자의 응답에 따라 오류를 수정한다.
10. 수동 수정 결과 저장 - 수정한 결과에 미등록어가 있으면 사전에 추가하고, 없으면 수정 로그에 추가한다.
11. 오류가 없어질 때까지 8-10 의 과정을 반복한다.

2 절. 워크벤치의 인터페이스

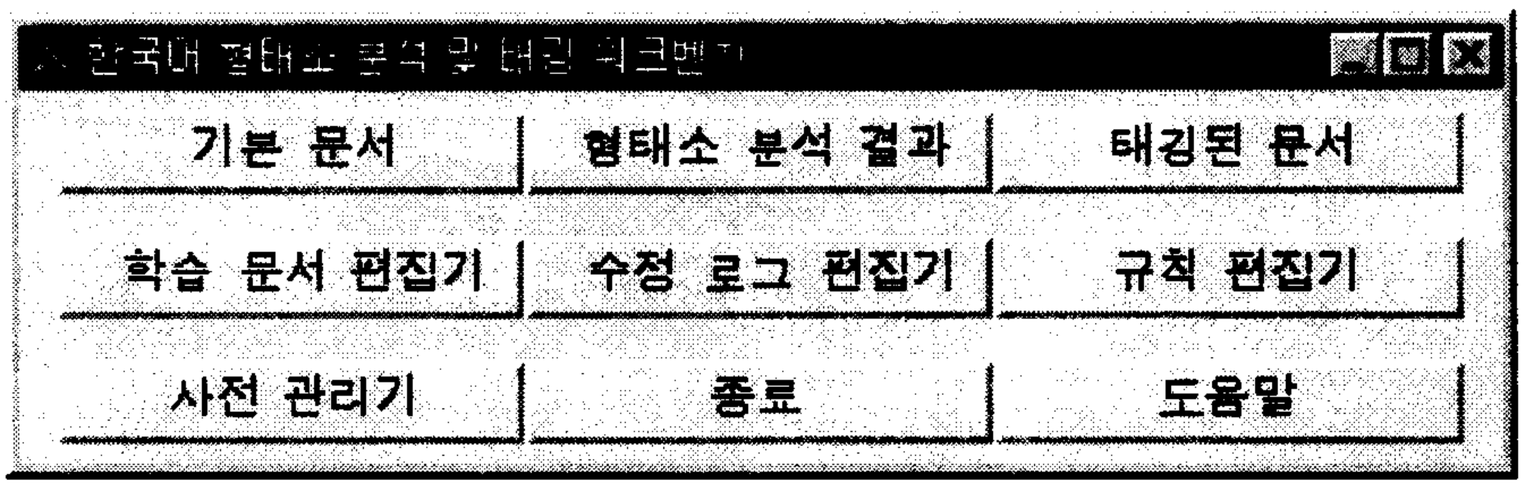


그림 4.2 워크벤치의 주 화면

1. 주 화면

주 화면은 그림 4.2 와 같은 모양으로 다음과 같은 버튼들이 있다.

- 기본 문서 - 형태소 분석의 입력으로 사용되는 기본 문서를 다루는 창을 구동한다.
- 형태소 분석 결과 - 형태소 분석 결과를 다루는 창을 구동한다.
- 품사부착 문서 - 품사부착 문서를 다루는 창을 구동한다.
- 사전 관리기 - 사전 관리기를 구동한다.
- 규칙 편집기 - 오류 지적 및 대안 제시에 사용되는 규칙을 편집한다.
- 수정 로그 편집기 - 오류 지적 및 대안 제시에 사용되는 수정 로그를 편집한다.
- 학습 코퍼스 편집기 - 학습 코퍼스를 편집하고, 자동으로 학습한다.
- 종료 - 시스템을 끝낸다.
- 도움말 워크벤치 인터페이스에 관한 도움말을 제공한다.

2. 기본 문서 화면

기본 문서 화면은 다음의 기능들을 제공한다.

- 화일 관련 기능 - 화일을 읽어서 편집 창에 보여주거나, 편집 창의 내용을 화일로 저장하는 기능을 제공한다.
- 편집 기능 - 편집 창의 내용을 키보드를 통해서 직접 사용자가 수정할 수 있다.
- 형태소 분석과 품사 태깅 - 편집 창의 내용을 형태소 분석하여 형태소 분석 결과 화면에 보여주거나, 품사 태깅 결과를 품사부착 문서 화면에 보여준다. 이 때, 반드시 형태소 분석 단계를 거치게 되는데, 형태소 분석 단계에서 발견되는 미등록어들을 사용자에게 보여준다.

3. 형태소 분석 결과 화면

형태소 분석 결과 화면은 기본 문서 화면의 화일관련 기능들과 편집 기능을 제공하고, 편집 창의 내용을 품사 태깅하여 품사부착 문서 화면에 보여주는 품사 태깅 기능이 있다.

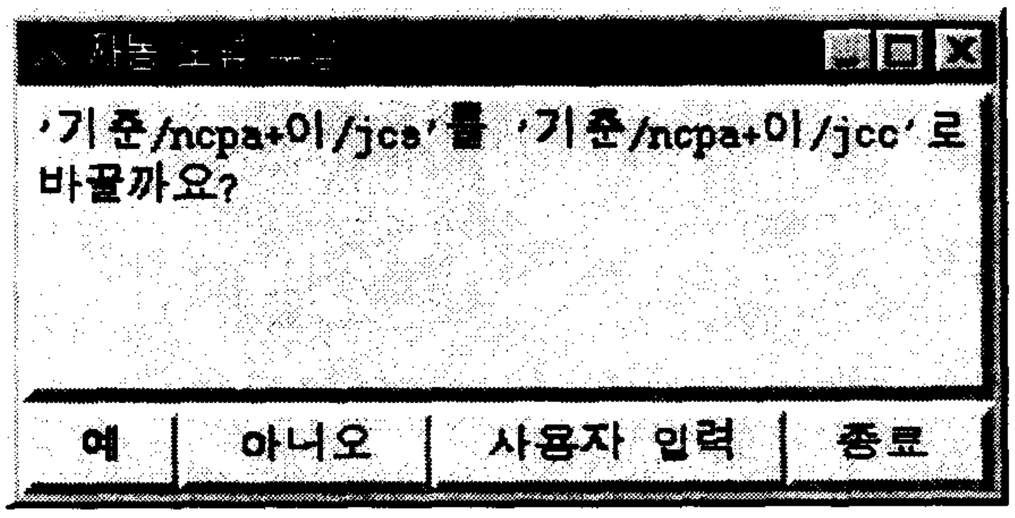


그림 4.3 자동 오류 수정 화면

4. 품사부착 문서 화면

품사부착 문서 화면은 다음의 기능을 제공한다.

- 화일 관련 기능 - 화일을 읽어서 편집 창에 보여주거나, 편집 창의 내용을 화일로 저장하는 기능을 제공한다.
- 편집 기능 - 편집 창의 내용을 키보드를 통해서 직접 사용자가 수정할 수 있다.
- 자동 오류 지적 및 대안 제시 기능 - 이 기능을 선택하면, 오류 규칙과 수정 로그 데이터베이스를 이용해서 편집 창의 내용을 보고 오류를 찾고 찾은 오류에 대한 대안을 그림 4.3 의 화면을 통해서 사용자에게 제시한다. 그림 4.3

에서 사용자가 '예' 버튼을 누르면 자동으로 대안으로 수정하고, '아니오' 버튼을 누르면, 다음 오류를 찾고, '사용자 입력'을 누르면 사용자로부터 직접 옳은 분석결과를 입력받고, 입력받은 내용에 대한 피드백 반영과정으로 들어간다.

- 수동 수정 기능 - 사용자가 편집 창에서 오류를 발견하면 마우스를 그 오류의 위치에 놓고 오른쪽 버튼을 누르면, 옳은 분석을 입력받고, 입력받은 내용에 대한 피드백 반영과정으로 들어간다.

5. 학습 문서 편집기

학습 문서 편집기는 품사 태깅에 사용되는 통계 정보를 추출하는 학습 문서에 대해 편집을 하는 도구이다. 오류지적 및 대안 제시는 품사부착 문서 화면에서 제공하는 자동 오류 수정과 같은 기능이고, 학습 버튼을 누르면 자동으로 편집 창의 내용을 저장하고 통계 정보를 추출하여 다음에 품사 태깅시에는 새로운 통계 정보를 기반으로 품사 태깅을 하게 된다.

6. 규칙 편집기

규칙 편집기는 자동 오류 수정에 사용되는 오류 규칙을 사용자가 편집할 수 있는 도구이다. 규칙을 수정하고 저장을 하면, 이후의 자동 오류 지적 및 대안 제시에서 수정된 규칙이 적용된다.

7. 수정 로그 편집기

수정 로그 편집기는 사용자가 수동으로 수정한 내용을 모아놓은 수정 로그를 편집할 수 있는 도구이다. 수정 로그를 편집한 후에 저장을 하면, 이후 자동 수정시에 편집한 내용이 반영된다.

8. 사전 관리기

사전 관리기는 사용자가 사전 데이터베이스를 편리하게 관리할 수 있게 해 주는 도구이다. 사용자는 사전 편집기를 사용하여 다음의 일들을 할 수 있다.

- 형태소의 내용 검색 - 사전 항목에 형태소를 입력한 후, 'Enter' 키나 '사전 검색' 버튼을 누르면 사전에서 그 형태소를 찾아서 결과를 사전 검색 결과창에 넣는다.
- 새로운 형태소와 태그 삽입 - 사전 항목에 형태소를 입력하고, '삽입, 삭제 또는 수정 전 태그' 항목에 삽입할 태그를 입력한 후, '항목 삽입' 버튼을 누른다. 만약 입력한 형태소가 사전에 있으면 그 항목에 입력한 태그를 추가하고, 사전에 없으면 형태소와 태그를 모두 추가한다.

- 기존의 형태소와 태그 삭제 - '삽입, 삭제 또는 수정 전 태그' 항목에 입력된 태그가 있으면, 사전 항목에 입력된 형태소의 태그만 삭제하고, 입력된 태그가 없으면 형태소 전체를 삭제한다.
- 기존 형태소의 태그 변경 - 사전 항목에 입력된 형태소의 '삽입, 삭제 또는 수정 전 태그' 항목을 수정 후 태그 항목으로 바꾼다.

5장. 실험 및 평가

본 실험에서 형태소 분석기는 [11]의 형태소 분석기를 사용하였고, 품사 태거는 [12]의 품사 태깅 시스템을 사용하였다. 그리고, 품사 태그 집합으로는 [9]의 태그 집합을 사용하였는데 자세한 내용은 부록에 첨가하였다.

실험은 중고교 교과서 4 과목의 약 8,000 어절의 문서와, 일반 문서 2 종류의 약 2,500 어절의 문서를 대상으로 하였다.

먼저, 비교를 위해서 입력 문서들을 형태소 분석기와 품사 태거만을 사용하여 태깅을 하였다. 그 결과가 표 5.1 에 나타나 있다.

문서	어절 수	태깅 오류 수	태깅 오류율
문서 1	1718	171	10.0 %
문서 2	1992	139	7.0 %
문서 3	2145	182	8.5 %
문서 4	2158	137	6.3 %
문서 5	813	77	9.5%
문서 6	891	107	12.0%
문서 7	872	105	12.0%

표 5.1 실험 문서들의 자동 태깅 결과

표 5.1 에서 '어절 수'는 각 문서의 어절 갯수이다. '태깅 오류 수'와 '태깅 오류

율'은 미등록어를 제거하지 않고, 형태소 분석기의 추정에 의한 결과로 자동 태깅했을 때의 오류 갯수와 오류율이다.

1절. 실험 과정

문서 1 에서 문서 7 까지 차례로 실험을 하여서 앞 문서에 대한 수정 결과가 다음 문서에 적용이 되게 하였다. 실험 과정은 입력 문서를 '1 차 형태소 분석'에서 제시된 미등록어를 입력하고 나서 '2 차 형태소 분석'과 '태깅'을 한다. 품사부착 문서에 대해 '자동 오류 지적및 대안 제시'를 통해서 수정을 하고 나서 '수동 수정과 피드백 반영'을 통해서 수정을 한다.

실험 결과는 각 과정에서의 오류 제거율을 계산하였다.

2절. 오류 수정 결과

오류 수정은 미등록어 입력을 통한 수정, 자동 오류 수정과 수동 오류 수정을 통한 피드백 반영으로 나눌 수 있다. 본 실험에서는 오류 수정에 사용된 규칙으로서 '되다(동사)'와 '아니다(형용사)' 앞의 주격조사를 보격조사로 바꾸는 규칙 2 개만을 사용하였다. 표 5.2 에 실험결과가 나타나 있다.

표 5.2 에서 '미등록어 처리'는 미등록어 입력을 통해 수정된 오류의 갯수이고, '자동 수정 수'는 규칙과 수정로그를 사용한 오류지적과 대안제시한 분석 중 옳은 갯수이고, '자동 수정 오류'는 규칙과 수정로그를 사용한 오류지적과 대안제시한 분석 중 틀린 갯수이고 '수동 수정 수'는 사용자가 직접 옳은 분석을 입력한 어절 수이고, 피드백 수정 수는 사용자의 수동 수정을 기반으로 오류지적과 대안제시한 분석 중 옳은 갯수이고, 피드백 오류 수는 사용자의 수동 수정을 기반으로 오류지적과 대안제시한 분석 중 틀린 갯수이다. 그리고 각 %는 전체 오류의 갯수 중 차지하는 비율이고 자동수정 오류와 피드백 오류의 %는 시스템이 제시한 오류와 대안의 갯수 중 차지하는 비율이다.

문서	미등록어 처리	자동 수정 수	자동 수정오류	수동 수정 수	피드백 수정 수	피드백 오류 수
문서 1	16(9.4%)	17(10.0%)	0	86(50.3%)	52(30.4%)	15(22.4%)
문서 2	4(2.9%)	65(46.8%)	10(13.3%)	59(42.4%)	11(7.9%)	5(31.2%)
문서 3	9(5.0%)	72(39.6%)	22(23.4%)	59(32.4%)	42(23.1%)	25(37.3%)
문서 4	14(10.2%)	51(37.2%)	29(36.3%)	54(39.4%)	18(13.1%)	5(21.7%)
문서 5	24(31.2%)	24(31.2%)	6(20.0%)	26(33.8%)	3(3.9%)	31(91.2%)
문서 6	45(42.1%)	33(30.8%)	34(50.7%)	23(21.5%)	6(5.7%)	2(25%)
문서 7	23(21.9%)	27(25.7%)	56(67.5%)	31(29.5%)	24(22.9%)	22(47.8%)

표 5.2 오류 수정 결과

3절. 평가

본 연구에서는 미등록어 제시를 통하여 미등록어로 인한 오류를 제거하였고, 오류규칙과 수정로그를 이용한 자동 오류지적, 대안제시를 통하여 수동 오류 수정의 양을 줄였다.

실험 결과를 그래프로 표현하면 그림 5.1과 같다. 그림 5.1에서 미등록어 처리는 전체 오류 중 미등록어를 통해서 제거한 오류의 양이고, 자동 수정은 미등록어 처리 이후에 존재하는 오류 중 자동으로 수정한 오류의 양이고, 수동 수정은 미등록어 처리 이후에 존재하는 오류 중 수동으로 수정한 오류의 양이다.

그림 5.1을 보면, 워크벤치를 이용해서 품사부착 코퍼스를 구축해 나가면서 자동 수정의 양이 점점 증가하고 수동 수정의 양은 점점 감소함을 볼 수 있다.

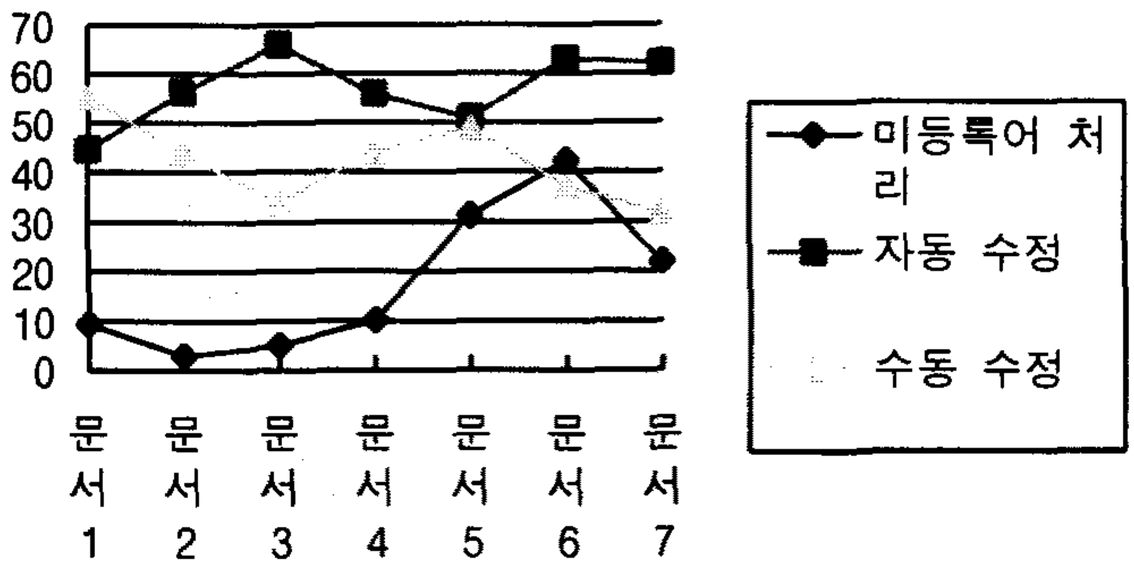


그림 5.1 실험 결과

[18]에서 사용된 규칙은 표현하지 못하는 상황이 존재하는 단점이 있고, [28]에서 사용된 규칙은 사용자가 쉽게 이해하거나 기술하기가 어려운 단점이 있다. 본 연구에서는 오류 규칙의 형식을 보다 간단하게 함으로써 사용자가 쉽게 오류 규칙을 기술할 수 있게 하였다.

본 실험에서는 중의성이 크고 오류로 발생할 확률이 적다고 판단되는 오류들에 대해서는 수정 로그에 넣지 않음으로써 오류지적, 대안제시의 확률을 높였는데, 만약 사용자가 오류지적, 대안제시의 확률이 낮더라도 자동 수정의 양을 늘리려면, 모든 오류 수정결과를 수정 로그에 넣으면 된다.

본 실험에서 사용자가 직접 입력을 하지 않고 시스템이 제안한 대안으로 자동 수정한 오류의 양은 전체 오류의 약 63.2%이다.

6장. 결론

최근에 코퍼스를 이용한 연구가 활발해지면서 코퍼스 구축의 중요성이 점점 커지고 있다. 일반적으로 품사부착 코퍼스를 구축하는 방법은 형태소 분석과 자동 품사 태깅 과정을 거친 후 수작업을 통하여 오류를 수정한다. 그러나, 오류 수정에 막대한 인적, 물적 비용이 들어가게 된다.

본 연구에서는 이러한 비용을 줄여줄 수 있는 작업 환경을 제시하였는데, 형태소 분석과정에서 미등록어를 제시함으로써 미등록어로 인한 오류를 제거하였고, 오류 규칙과 수정 로그를 이용해서 자동으로 오류를 찾고 대안을 제시하는 방법을 제시하였다. 그리고, 간단한 실험에서 약 63.2%의 오류를 자동으로 수정할 수 있었다.

참고문헌

[1] 강승식, 음절 정보와 복수어 단위 정보를 이용한 한국어 형태소 분석, 서울대학교 컴퓨터 공학과 박사학위 연구, 1993.

[2] 김덕봉, 예측 중심의 형태소 분석:한국어 어절 인식을 위한 계산 모델, 한국과학기술원, 전산학과, 박사학위 연구, 1996.

[3] 김성용, Tabular Parsing 방법과 접속정보를 이용한 한국어 형태소 해석기, 한국과학기술원 석사학위 연구, 1987.

[4] 김재훈, 한국어 형태소 처리에 관한 고찰, 한국과학기술원, 전산학과 컴퓨터 시스템 실험실, 내부 메모, 1992.

[5] 김재훈, 서정연, 품사 태깅: 검토 및 다루기 어려운 문제들, 한국과학기술원, 인공지능연구센타, CAIR-TR-94-53, 1994.

[6] 김재훈, 서정연 & 김길창, 실용적인 한국어 형태소 해석, 한국과학기술원, 전산학과, 기술문서(CS-TR-95-98), 1995.

[7] 김재훈, 임철수 & 서정연, 은닉 마르코프 모델을 이용한 효율적인 한국어 품사 태깅, 한국정보과학회 논문지, Vol. 22, No. 1, pp. 136-146, 1995.

[8] 김재훈, 오류-보정 기법을 이용한 어휘 모호성 해소, 한국과학기술원, 전산학과, 박사학위 연구, 1996.

[9] 김재훈, 최기선, 김덕봉, 최병진, 한영균, 남영준, 박석문, 김진규, 김진수, 이춘택, 통합국어정보베이스를 위한 한국어 형태, 통사 태그 설정, 1996.

[10] 남기심, 고영근, 표준 국어 문법론, 탑출판사, 1987.

[11] 문화체육부, 국어 정보 처리 기반 구축을 위한 연구, 1994.

[12] 신중호, 한영석, 박영찬 & 최기선, 어절구조를 반영한 은닉 마르코프 모델을 이용한 한국어 품사 태깅, 제 6 회 한글 및 한국어 정보처리 학술대회 발표논문집, pp. 389-364, 시스템공학연구소, 대전, 1994.

[13] 이성진, Two-level 한국어 형태소 해석, 한국과학기술원 석사학위 연구, 1992.

[14] 이상호, 미등록어를 고려한 한국어 품사 태깅, 한국과학기술원 석사학위 연구, 1995.

[15] 이은철, CYK 법에 기반한 한국어 형태소 분석에서의 개선기법, 포항공과대학 대학원, 전자계산학과, 석사학위 연구, 1992.

[16] 이운재, 한국어 문서 태깅 시스템의 설계 및 구현, 한국과학기술원 석사학위 연구, 1993.

[17] 임철수, HMM 을 이용한 한국어 품사태깅 시스템 구현, 한국과학기술원 석사학위 연구, 1994.

[18] 임희석, 김진동, 임해창, 한국어 특성에 적합한 변형 규칙 기반 한국어 품사 태깅, 춘계 인공지능 연구회 학술발표 논문집, pp. 3-10, 1996.

[19] 최형석 & 이주근, 자연어 처리 앨고리즘, 한국정보과학회 가을 학술대회 발표 논문집, 제 11 권, 제 2 호, pp. 36-43, 1984.

[20] J. Benello, A. W. Mackie, J. A. Anderson, "Syntactic Category Disambiguation with Neural Networks", *Computer Speech and Language*, Vol. 3, pp. 203-217, 1989.

[21] E. Brill, "A Simple Rule-Based Part of Speech Tagger", *Proc. Of the 3rd Conf. on Applied Natural Language Processing*, Trento, Italy, pp. 153-155, April, 1992.

[22] E. Brill, A Corpus-Based Approach to Language Learning, Ph. D. Thesis, Department of Computer and Information Science, University of Pennsylvania, 1993.

[23] E. Charniak, C. Hrickson, N. Jacobson, and M. Perkowita, "for Part-of-Speech Tagging", *Proc. Of Nat'l Conf. On Artificial Intelligence*(AAAI-86) pp. 784-789, 1993.

[24] Kenneth Ward Church, "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text", *Proceedings of Applied Natural Languate Processing*, Austin, Texas, 1988.

[25] S. J. DeRose, "Grammatical Category Disambiguation by Statistical Optimization", *Amer. J. of Computational Linguistics*, Vol. 14, No. 1, pp. 31-39, 1988.

[26] G. F. Foster, "Statistical Lexical Disambiguation", Master's Thesis, McGill Univ. School of Computer Science, Montreal, Canada, 1991.

[27] Julian Kupiec, "Robust Part-of-Speech Tagging Using a Hedden Markov Model", *Computer Speech and Languate*, Vol. 6, pp. 225-242, 1992.

[28] Geunbae Lee & Jong-Hyeok Lee, "Rule-based error correction for statistical part-of-speech tagging", Korea-China Joint Symposium on Oriental Language Computing, 1996.

[29] M. Nakamura, K. Maruyama, T. Kawanata, K. Shikano, "Neural Network Approach of Word Category Predic for English Texts", *Int'l Conf. on Computational Linguistics*(Coling-90), pp. 213-218, 1990.

[30] R. Weischedel, R. Scewartz, J. Ralmucci, M. Meteer, L. Rawshaw, "Coping with ambiguity and unknown words through Probabilistic Model", *Computational Linguistics*, Vol. 19, No. 2, pp. 359-382, 1993.

여백

4. 구문분석기 및 구문트리 태깅 Corpus

한국과학기술원
최기선

여백

4. 구문분석기 및 구문트리 태깅 Corpus

1장. 한국어 구문트리 태깅 코퍼스의 필요성

대량의 코퍼스는 실제 사람들이 사용하는 문장들을 그 기반으로 하고 있다. 최근의 자연언어 처리 분야에서는 이와 같은 대량의 코퍼스로부터 추출한 언어 지식(linguistic knowledge)을 이용하여 자연언어 처리 분야의 문제점들을 해결하려는 노력이 증가하고 있다. 이는 규칙 기반의 문제 해결 방식에서의 한계점을 언어적 정보를 담고 있는 실제의 코퍼스로부터 자동으로 추출한 정보를 이용하여 해결해 보고자 하는 노력이다. 또한, 대량의 코퍼스로부터 자동적으로 추출할 수 있는 통계적 정보는 자연언어 처리 문제 해결에 학습(learning)이라는 과정을 도입할 수 있도록 해준다.

한국어 구문 트리 태깅 코퍼스는 각각의 한국어 입력 문장에 대해, 그 문장에 해당하는 구문 구조를 구문 트리 형태로 표현하고 있는 코퍼스를 의미한다. 문장의 구문구조는 문장을 구성하고 있는 단어들 간에 서로 어떠한 관계로 연결되어 있는가를 명시해 준다. 즉, 입력 문장에서 단어와 단어 간에 어떠한 관계를 갖고 있는가, 어떠한 단어가 서로 더 밀접하게 관련되어 있는가와 같은 구조적인 표현을 나타냄으로써, 문장의 의미 파악을 수행하기 위한 전처리 역할을 수행할 수 있다. 이러한 정보를 담고 있는 대량의 구문 트리 태깅 코퍼스는 기본적인 자연언어 처리 전반에 걸친 주요한 정보원일 뿐 아니라, 한국어 구문 분석기를 위한 구문 규칙을 추출할 수 있고, 한국어 용언의 하위 범주화 정보 및 한국어 구문 유형이나 한국어 단어의 사용례에 대한 연구의 근원이 될 수 있다. 또한, 자연언어 처리뿐 아니라, 한국어 음성인식이나 문자인식, 정보검색, 사전구축 등의 연구 분야에까지 중요한 연구 자료가 될 수 있으리라 기대된다.

본 연구에서는 구문 트리 태깅 코퍼스의 구축을 지원하는 시스템의 개발을 그 목적으로 한다. 본 연구에서 개발하는 구문 트리 태깅 시스템은 자동 구문 트리 태거의 결과를 사람이 검사하여 그 적법성 여부를 판단하고 오류가 있을 시에는 수정 작업을 통해서 정확한 구문 트리를 생성하여 저장해 준다. 이 때, 수정되어

저장되는 구문 트리는 자동 구문 트리 태거의 성능 향상에 영향을 미칠 수 있어야 한다. 또한, 한국어 구문 구조를 표현하는 구문 트리 표현 방식에 대한 표준안을 제안하기 위해 한국어 구문 태그를 설정하고, 한국어 구구조 규칙의 형태를 제안한다. 이를 이용하여 한국어 약 10 만 문장 수준의 구문 트리 태깅된 구문 트리 뱅크의 구축을 수행하였다. 구문트리 태깅 코퍼스는 여러 사람들에 의해 구축되어지므로, 무엇보다도 일관성 유지에 신중을 기하여 구축하였다. 또한, 되도록이면 양질의 코퍼스를 구축하기 위하여 구문 트리 태깅 결과에 대한 검증 과정을 수행하였다.

본 보고서의 구성은 다음과 같다. 2 절에서는 한국어 구문 트리 뱅크 구축에 필수적인 대상 코퍼스와 구문 트리 태깅을 위한 구문 태그 집합, 구문 트리 태깅에서 구절 형성의 원칙, 그리고 구문 트리 코퍼스의 표현 방법에 대해 기술한다. 3 절에서는 한국어 구문 트리 태깅 시스템의 전체적인 시스템 구성과 그 기능에 대해 기술하고 4 절에서는 확률을 기반으로 하는 한국어 구문 분석기에 대해 기술한다. 5 절에서 본 연구를 수행한 결과 및 토의 사항을 기술하고 마지막으로 결론을 맺도록 한다.

2 장. 한국어 구문트리 태깅 코퍼스

1 절. 품사부착 코퍼스

본 연구에서 구문트리 태깅의 대상으로 삼는 코퍼스는 특정 분야에 치우치지 않는 문서 집합이 되도록 구성하였다. 구문 트리 태깅 코퍼스 구축을 위한 입력 형태는 품사가 부착되어 있는 코퍼스이다. 한국어는 실질어에 기능어가 결합함으로써 문법적 기능을 수행하는 첨가어이다. 그렇기 때문에 품사가 부착되어 있는 형태로부터 원문을 추출하는 것이 어렵다. 이러한 이유로 인하여 구문 트리 태깅 코퍼스 구축을 위한 입력의 형태 또한, 원문과 품사가 부착되어 있는 두 가지 형태가 된다. 구문 트리 태딩 코퍼스를 위한 품사부착 코퍼스의 입력 형태는 다음과 같다.

그	그/mmd
이유는	이유/ncn+는/jxt

갑자기	갑자기/mag
뜨거운	뜨겁/paa+ㄴ/etm
물에	물/ncn+에/jca
들어가면	들어가/pvg+면/ecs
혈압이	혈압/ncn+이/jcs
오르기	오르/pvg+기/etn
때문이다.	때문/nbn+이/jp+다/ef+./sf
이런	이런/mmd
사람은	사람/ncn+은/jxt
욕조	욕조/ncn
속에서	속/ncn+에서/jca
몸을	몸/ncn+을/jco
충분히	충분히/mag
덥히지도	덥히/pvg+지/ecx+도/jxc
못하고	못하/px+고/ecs
나오게	나오/pvg+게/ecx
된다.	되/px+ㄴ다/ef+./sf

동일한 이유로 인하여 구문트리 태깅 코퍼스에도 원문을 보존해야 한다. 본 연구에서는 구문트리 태깅 코퍼스에서 원문을 보존하기 위하여 각 문장에 대한 구문 구조와 더불어 그 첫번째 줄에 원문을 기술한다.

2절. 구문트리 태깅 코퍼스 구축에 사용하는 한국어 구문 태그 집합

한국어 구문 태그 집합을 정의하기에 앞서, 우선 한국어 구문 구조를 표현하는 기본 단위에 대하여 언급하고자 한다. '구'란 둘 이상의 단어가 한 덩어리가 되어 마치 한 품사의 단어처럼 쓰이는 것이고 '절'이란 주어와 서술어를 다 갖춘 온전한 문장이 어느 한 품사의 단어처럼 쓰이는 것을 의미한다.

(예제 1) 가. 학교에 간 일이 있다.
나. 철수와 같이 학교에 간 일이 있다.
다. 나는 철수와 같이 학교에 간 일이 있다.

앞의 구와 절의 정의에 따르면, (예제 1.가)와 (예제 1.나)의 경우에는 관형구가 되고, (예제 1.다)의 경우에는 관형절이 된다. 그러나, 반드시 주어를 필요로 하

지 않는 우리말에서는 '구'와 '절'을 나눈다는 것은 많은 문제가 있다[김기혁 1995]. 본 연구에서는 구와 절의 명확한 구분 대신에, '구절'이라는 단어를 사용하여 구와 절을 하나의 개념으로 보고자 한다. 다른 언어 단위와 마찬가지로 한국어 문장에 대한 정의는 매우 다양하다. 학교문법에서는 문장은 의미상 완결된 사상, 감정을 나타내는 것으로 '이야기'다음으로 문법의 가장 큰 단위라고 정의하고 있다. 또한, 문장의 성격으로는 문장은 완결된 사상이나 감정을 나타내기 때문에 완결의 표시로 문장 종결 부호 '.', '?', '!' 등을 두며, 일반적으로 하나의 문장은 주어부와 서술부를 지니고 있다. 본 연구에서는 다음의 (예제 2)의 모든 것을 문장으로 간주한다.

(예제 2) 가. 철수는 착한 학생이다.
나. 장미꽃이 가장 아름답다.
다. 아!
라. 누구니?
마. 불이야!

본 연구에서는 한국어 구문트리 태깅 코퍼스 구축 시 사용하는 구문구조 표현 방법으로 구구조 문법(Phrase-Structure Grammar)을 사용하고자 한다. 구구조 문법에서는 사용하는 구절(Phrase, Constituent)은 어떠한 단어들이 서로 관련을 갖으면서 결합되는지를 알려줄 수 있다. 이와 같은 각각의 구절을 표현하기 위해서는 그 구절에 해당하는 구문 태그가 있어야 한다. 이러한 구문 태그는 그 구문 태그를 사용하고자 하는 언어를 고려하여 구성되어야 한다. 한국어에서 사용되는 문장의 구성 성분은 크게 다음과 같이 나누어 볼 수 있다.

성분	주성분	주어(主語)
		서술어(敍述語)
		목적어(目的語)
		보어(補語)
	부속성분	관형어(冠形語)
		부사어(副詞語)
	독립성분	독립어(獨立語)

또한, 한국어에서 사용하는 구의 갈래와 그에 해당하는 예제는 다음과 같다.

- 명사구 : "저 새 차는 철수네 것이다."
- 동사구 : "봄이라 꽃이 활짝 피었다."

- 형용사구 : "우리 반의 순이는 매우 친절하다."
- 관형사구 : "순이는 아주 새 옷을 입고 왔다."
- 부사구 : "조용히, 그리고 간절히 우리를 부르고 있다."
- 독립구 : "어머나, 네가 아주 새 사람이 되었구나!"

한국어의 이와 같은 문장 구성 성분에 대응하여 본 연구에서는 다음과 같은 구문 태그 집합을 정의하여 사용하였다.

- **문장(S)** :

  마침표로 마무리 되는 단어들의 나열이 하나의 문장을 형성한다. 문장은 문장 종결 부호로 마무리지어지는 단어의 나열이다. 문장 종결 부호로는 (.),(?),(!) 등이 사용된다. 그러므로, 종결 어미로 끝나는 경우가 아니더라도 문장 종결 부호가 있으면 하나의 문장이 될 수 있다.
- **명사구절(NP)** :

  명사구절은 격표시 조사에 따라, 문장 성분 중 주어, 목적어, 보어, 관형어, 부사어가 될 수 있다. 즉, 명사구절에 주격 표시의 조사가 결합되면 주어가 되고, 보격 조사가 겹합하면 보어가 되며, 목적격 표시의 조사가 결합되면 목적어가 되며, 부사격 조사가 결합되면 부사어가 될 수 있다.
- **동사구절(VP)** :

  동사구절은 동사나 서술격 조사, 동작성 명사에 의해 서술어를 형성하며, 동사구절에 관형사형 어미가 결합되면 관형어가 될 수 있다.
- **형용사구절(ADJP)** :

  형용사구절은 형용사나 상태성 명사에 의해 서술어를 형성하며, 형용사구절에 관형사형 어미가 결합되면 관형어가 될 수 있다.
- **부사구절(ADVP)** :

  부사류가 부사구절을 형성하여 부사어가 될 수 있다. 단, 단어와 단어를 연결시키는 단어 접속 부사의 경우에는 부사구절을 형성하지 않는다.
- **관형사구절(MODP)** :

  관형사류가 관형사구절을 형성하여 관형어가 될 수 있다.
- **독립구절(IP)** :

  감탄사나 제시어, 표제어 그리고 문장 접속에 관계되는 부사들이 독립구절을 형성하여 독립어가 될 수 있다.
- **보조용언구절(AUXP)** :

한국어에서 시제(tense), 존칭(honorific), 상(aspect), 법성(modal) 등을 나타내는 보조 용언들이 보조용언구절을 형성하여 동사구절이나 형용사구절과 결합한다.

구문 태그 집합과 이에 해당하는 한국어의 문장 구성 성분 간의 관계는 표 3.1과 같다.

< 표 3.1 : 한국어 문장의 성분과 구문 태그와의 관계 >

문장의 성분	구문 태그 + 문법적 관계
문장	S
주어	NP+jcs
서술어	VP ADJP
목적어	NP+jco
보어	NP+jcc
관형어	MODP NP+jcm VP+etm ADJP+etm
부사어	ADVP NP+jca NP+jct NP+jcr
독립어	IP
기타	AUXP

3절. 한국어 구절 구성의 기본 제약 조건

한국어 구문 트리 태깅 코퍼스 구축에 사용하는 태깅 방법에서 구절을 형성하는 기본 제약 조건은 다음과 같다.

- 한국어에서 어절은 띄어 쓰기의 단위이며, 하나의 어절은 하나 이상의 실질 형태소로만 구성되거나, 하나 이상의 실질 형태소와 하나 이상의 형식 형태소(기능어)로 구성될 수 있다.

➠ 영어에서는 구절의 최소 기술 단위가 띄어 쓰기의 단위인 하나의 단어가 된다. 그러나, 한국어에서 띄어 쓰기의 단위는 어절이며, 한 어절 내에는 문법적 기능을 달리하는 형태소가 여러 개 올 수 있으므로 구절을 기술하는 데

있어서 최소 단위는 어절이 아닌, 형태소를 사용한다.

- 한국어는 의미를 나타내는 실질 형태소(어근)에 조사와 어미 같은 문법적 관계를 나타내는 형식 형태소(기능어)가 붙음으로써 문법 기능을 한다. 형식 형태소에 속하는 것으로는 조사, 어미, 접사 등이 대표적이다.

➠ 실질 형태소는 독립적인 구절을 형성하도록 한다. 문법적 관계를 나타내는 형식 형태소는 독립적인 구절을 형성하지 않고 구절 간의 문법적 관계를 명시하도록 한다. 이는 형식 형태소를 고려하여 구절을 형성하게 되면, 구절의 종류가 많아질 뿐더러, 보조사의 사용이 빈번한 한국어의 경우 정확한 구절의 구분이 어렵기 때문이다. 표 3.2 에는 독립구절을 형성하지 않는 형태소들을 보이고 있다.

$NP \rightarrow ncn$ 실질형태소 (명사)
$VP \rightarrow pvg$ 실질형태소 (동사)
$VP \rightarrow NP + jcs \quad VP$ 형식형태소 (주격조사)

< 표 3.2 : 독립구절을 형성하지 않는 형태소 >

형식 형태소	주격조사(jcs),목적격조사(jco),보격조사(jcc),부사격조사(jca), 호격조사(jcv),관형격조사(jcm), 접속격조사(jcj), 공동격조사(jct), 인용격조사(jcr),통용보조사(jxc), 종결보조사(jxf),서술격조사(jp), 대등적 연결어미(ecc), 종속적 연결어미(ecs), 관형사형 어미(etm), 명사형 어미(etn), 종결어미(ef), 동사파생접미사(xsv), 형용사파생접미사(xsm), 명사파생접미사(xsn), 부사파생접미사(xsa)
실질 형태소	비단위성 의존명사(nbn), 단위성 의존명사(nbu), 단어 접속 부사(maj), 각종 문장 기호(s)

- 한국어의 기능어는 그 기능어가 붙어 있는 단어와 함께 한 어절을 형성하지만, 그 문법적 기능은 한 어절이 아닌 문장, 용언구 또는 체언구 전체에 미친

다. (예제 3.가)에서 '음'은 형용사 '없다'와 한 어절을 이루지만 그 실제 기능은 문장 '그가 죄가 없다' 전체를 명사화하고 있다. (예제 3.나)의 경우에는 '었'이라는 선어말 어미가 문장 '나는 밥을 먹다'의 시제를 과거로 만들어 주고 있다. 이와 같이 기능어의 문법적 기능은 그 기능어가 결합된 단어에 의해 이끌리는 용언구 전체에 영향을 미칠 수 있다.

(예제 3) 가. 그가 죄가 없<u>음</u>이 드러났다.
나. 나는 밥을 먹<u>었</u>다.

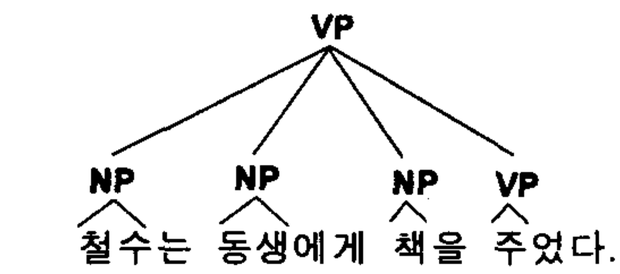
➠ 기능어는 그 기능어가 영향을 미치는 용언구나 체언구를 형성한 후에 결합하도록 한다. 위의 (예제 3.가)에서 '그가 죄가 없다'라는 형용사구(*ADJP*)를 형성한 후에 '음' 이라는 기능어를 결합하여 명사구(*NP*)를 형성하도록 한다.

$NP \rightarrow ADP + etn$ (NP (ADJP 그가 죄가 없)+음/etn)

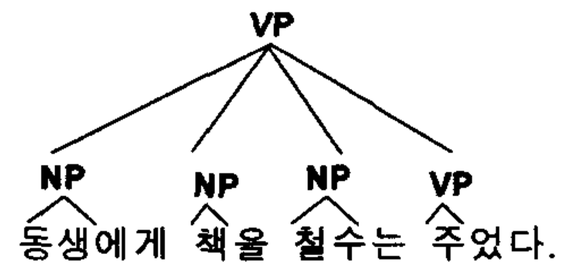
4 절. 한국어 구문트리 태깅 코퍼스를 위한 한국어 구구조 규칙 형태

한국어 구문 트리 태깅 코퍼스 구축에 사용하는 구문 구조는 구구조 규칙에 의해서 표현될 수 있다. 한국어는 영어와 다른 특성을 갖고 있기 때문에 구문 구조를 표현하는 구구조 규칙의 형태 또한 영어와는 달리 표현되어야 한다. 한국어는 부분 자유 어순으로 문미 술어를 제외한 다른 구성 성분들의 자리 옮김이 자유롭다. 또한, 한국어는 문장의 근간 성분인 주어나 목적어가 매우 쉽게 생략되어 질 수 있다[이익섭 1983]. 그림 3.1 에서 보는 바와 같이, 하나의 용언 '주다'에 대해 여러 개의 가능한 한국어 문장이 발생함을 볼 수 있다. 이는 한국어의 부분 자유 어순과 주어나 목적어가 빈번히 생략될 수 있다는 특징에서 기인한 것이다. 부분 자유 어순의 특성을 갖고 있는 한국어를 위한 구구조 규칙을 영어권의 규칙과 동일하게 표현했을 경우, 규칙의 증가량은 다음과 같다. 하나의 용언과 그 용언에 대한 하나의 구성 성분으로 이루어진 규칙을 최소 규칙이라고 하고, 이와 같은 최소 규칙의 갯수를 N_{min}이라 할 때, n개의 구성 성분과 하나의 용언으로 구성된 규칙의 갯수는 $N_{min}{}^{n}$에 비례하여 증가한다. 이와 같이 영어권에서 사용하는 규칙의 형태를 그대로 한국어에 적용할 경우, 기하급수적으로 규칙의 수가 증가하게 되는 단점을 지니게 된다. 규칙의 증가는 구문 분석 결과의 중의성 증가에 많은

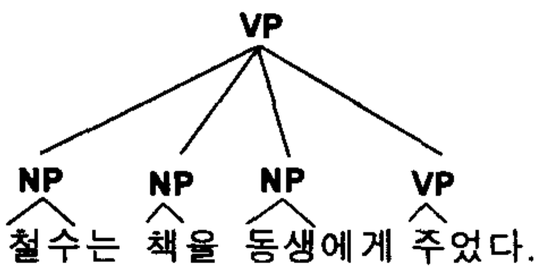
영향을 미칠 뿐만 아니라 구문 분석기의 효율을 저하시키는 원인이 될 수 있다.



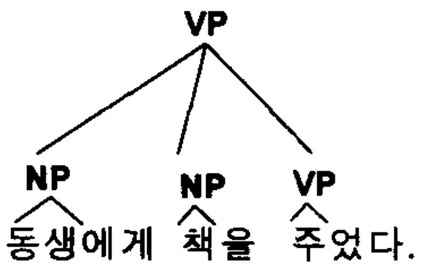
$VP \rightarrow NP_{+는}\ NP_{+에게}\ NP_{+을}\ VP$



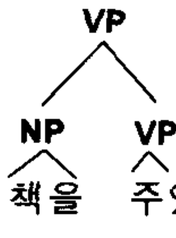
$VP \rightarrow NP_{+에게}\ NP_{+을}\ NP_{+는}\ VP$



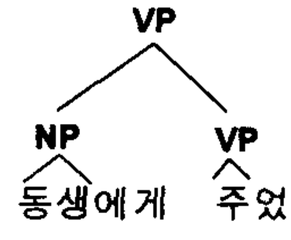
$VP \rightarrow NP_{+는}\ NP_{+을}\ NP_{+에게}\ VP$



$VP \rightarrow NP_{+에게}\ NP_{+을}\ VP$



$VP \rightarrow NP_{+을}\ VP$



$VP \rightarrow NP_{+에게}\ VP$

< 그림 3.1: 용언 '주다'에 대한 가능한 구문 트리 (일부) >

본 연구에서는 이와 같은 단점을 극복하기 위하여 구구조 규칙의 형태를 제한하여 사용하고자 한다. 하나의 규칙에는 문법적 기능이 다른 하나의 형식 형태소만이 포함되도록 규칙의 형태를 제한한다. 여기서 문법적 기능이 다른 하나의 형식 형태소라고 언급하는 것은 한국어에서는 한 어절 내에 여러 개의 형식 형태소들이 중복되어 나타날 수 있기 때문이다. 이러한 경우에, 문법적 기능을 달리하는 형식 형태소들만을 고려하도록 한다. 즉, 의미적 첨가 기능을 수행하는 보조사의 경우에는 하나의 규칙을 형성하는 형식 형태소에서 제외하기로 한다. 이와 같이 규칙의 형태를 제한하게 되면, 앞서 언급했던 한국어의 특성으로 인한 규칙의 갯수가 급증하는 것을 막을 수 있을 뿐 아니라, 하나의 규칙에서 하나의 문법적 기능 -- 하나의 형식 형태소에 의한 -만을 다루기 때문에, 그 처리 상에 있어서도 간단해 질 수 있다.

한국어의 형식 형태소는 그 형태소가 붙어 있는 구절이 다른 성분들에 대하여 어떠한 역할을 수행하는지를 알려 준다. 한국어의 형식 형태소는 크게 조사, 어미, 접사류로 나누어 볼 수 있다. 본 논문에서는 [김재훈 1996]의 품사 분류 체계를

따르며, 이 품사 분류 체계에서는 모두 23 개의 형식 형태소를 사용하고 있다.

(예제 4) "밥을 먹기가 힘들었다."

(예제 4)에서 어절 '먹기가'는 실질어 '먹다'에 명사형 전성어미 '기'가 그리고 주격조사 '가'가 결합되어 형성된다. 명사형 전성어미 '기'는 문장 "밥을 먹다"'를 명사화하는 역할을 하며 주격 조사 '가'는 이러한 '밥을 먹기'를 '힘들었다'의 주어로 만드는 역할을 하고 있다. 위의 예제에서도 알 수 있듯이 기능어는 성분을 변화시키는 것과 다른 구절과의 관계를 명시하는 것으로 나누어 볼 수 있다. 이의 구분은 주로 그 형식 형태소가 어절에서 어느 위치에 주로 나타나느냐에 따라 구분해 볼 수 있다. 주로 성분의 변화를 유발시키는 형식 형태소는 어절의 제일 끝에 나타나지 않는데 비해 다른 구절과의 관계를 명시하는 형식 형태소는 주로 어절의 제일 끝에 나타나는 경향이 많다. 이러한 기능어의 특성을 알아보기 위하여 기능어를 어절 내 위치 분포에 따라 자동 분류해 보았다. 21,715 어절을 대상으로 23 개의 기능어를 자동 분류하여 얻은 2 개의 분류 결과는 표 3.3 과 같다. 본 연구에서는 주로 어절 말미에 나타나는 기능어들을 구절 간 관계 명시를 나타내는 것이라 하고 어절 말미보다는 어절 내부에 발생하는 기능어들을 구절 내 관계 명시라고 하기로 한다. 이와 같이 분류한 기능어에 따라, 구구조 규칙의 형태를 다음과 같이 제한하고자 한다.

< 표 3.3 : 기능어의 분류 >

기준		
구절 간 관계 명시	격조사	주격조사(jcs), 목적격조사(jco), 보격조사(jcc), 부사격조사(jca), 관형격조사(jcm), 공동격조사(jct), 인용격조사(jcr), 접속격조사(jcj), 통용보조사(jxc)
	어미	대등적연결어미(ecc), 종속적연결어미(ecs), 관형사형어미(etm)
구절 내 관계 명시	격조사	서술격조사(jp), 호격조사(jcv), 종결보조사(jxf)
	어미	명사형어미(etn), 선어말어미(ep), 종결어미(ef)
	접사	명사파생접사(xsn), 동사파생접사(xsv), 형용사파생접사(xsm), 부사파생접사(xsa)

- **구절 내 관계 명시**

구절의 문법적 성분의 변화를 유발시키거나 속성을 결정하는 기능어들이다. 이와 같은 기능어가 사용된 구절에 대한 구구조 규칙의 형태는 다음과 같다.

Class I: $A \rightarrow B + \tau$

여기서 τ는 구절 내 관계를 명시해 주는 기능어들과 보조 용언 구절(*AUXP*), 그리고 ε이 가능하다. 이 규칙이 의미하는 바는 구절 B가 τ에 의해서 그 속성이 결정되거나 기능이 변화함을 의미한다. 이에 속하는 규칙으로는 다음과 같은 것들이 가능하다.

$VP \rightarrow NP + jp$	명사구절이 동사구절로 변함
$NP \rightarrow VP + etn$	동사구절이 명사구절로 변함
$NP \rightarrow ncn + xsn$	명사의 속성이 결정됨

- **구절 간 관계 명시**

이에 속하는 기능어들은 문법적 성분의 변화와 더불어 두 구성 성분 간의 문법적 관계를 명시하는 역할을 담당한다. 구절 간 관계를 명시하는 기능어가 사용된 구절에 대한 구구조 규칙의 형태는 다음과 같은 두 가지가 가능하다.

Class II: $A \rightarrow B + \gamma\ C$

여기서 γ는 구절 간 관계를 명시해 주는 기능어들 중에서 병렬을 표현하는 접속격 조사(*jcj*)와 대등적 연결어미(*ecc*)를 제외한 기능어들이 가능하다. 또한, 격조사의 생략이 빈번하므로 ε도 가능하다. 이 규칙이 의미하는 바는 구절 B가 구절 C와 γ의 문법적 관계를 형성함을 의미한다. 이에 속하는 규칙으로는 다음과 같은 것들이 가능하다.

$VP \rightarrow NP + jcs\ \ VP$	명사구절이 동사구절을 주격관계로 한정
$NP \rightarrow VP + etm\ \ NP$	동사구절이 명사구절을 관형어로 한정
$VP \rightarrow VP + ecs\ \ VP$	동사구절이 다른 동사구절을 종속적으로 한정

Class III: $A \rightarrow A_1 + \gamma'\ A_2 + \gamma'\ \ldots A_n$

여기서 γ'는 병렬 구조를 표현하는 접속격 조사와 대등적 연결어미, 그리고 나열을 표현하는 쉼표(*sp*)와 단어 접속 부사(*maj*)가 가능하다. 구절 간 관계를 명시하는 기능어들 중에서 병렬 형태의 규칙을 따로이 나누어 설정한 것은 *Class II*와는 달리 이에 속하는 규칙의 오른편(RHS)에는 여러 개의 구절이 올 수 있기 때문이다. 이 규칙 형태는 주로 병렬 형태의 구조를 표현하며, 이에 속하는 규칙으로는 다음과 같은 것들이 가능하다.

$$VP \rightarrow VP + ecc \;\; VP + ecc \;\; VP$$

$$NP \rightarrow NP + jcj \;\; NP + jcj \;\; NP$$

5절. 구문태그 예제

1. 문장 형성

- *동사구절로 구성되는 문장*

  (예문) 직선제의 장점은 이를 능가하다.

  (S

  (VP 직선제의 장점은 이를 능가하)+다/ef+./sf)

  S ---> VP+ef+sf

- *형용사 구절로 구성되는 문장*

  (예문) 따라서, 우주의 원리는 조화에 있다.

  (S (IP 따라서/maj)+sp

  (VP 우주의 원리는 조화에 있)+다/ef+./sf)

  S ---> IP+sp ADJP+ef+sf

- *명사구가 문장을 이끄는 경우*

  (예문) 한 학자가 화학비료와 농약으로 재배하는 가지.

  (S

  (NP 한 학자가 화학비료와 농약으로 재배하는 가지)+./sf)

  S ---> NP+sf

- *독립구절로 구성되는 문장*

  (예문) 어머나!

  (S (IP 어머나/i)+!/sf)

  S ---> IP+sf

2. 명사구절 형성 예제

- *단순한 명사구절*

  (예문) 농어민의 사기진작에 꽤 도움이 된다.

  (NP 사기/ncn+진작/ncn)

  NP ---> ncn+ncn

- *명사화 전성어미에 의한 명사절*

  (예문) 막대기가 급속히 움직이기 때문에 그 타격력이 줄기 전체로 분산되는 일이 없다.

  (NP

  (VP 막대기가 급속히 움직이)+기/etn)

  NP ---> VP+etn

- *병렬 표현*

  (예문) 고문법은 규칙과 학설 및 학자를 갖는 하나의 법학을 형성하였다.

  (NP 규칙/ncn+과/jcj 학설/ncn 및/maj 학자/ncn)+를
  NP ---> ncn+jcj ncn maj ncn

- *체언 상당어구*

  (예문) 이제, 어떻게 하느냐를 결정하자.

  (NP
  (VP 어떻게 하느)+냐/ef)+를
  NP ---> VP+ef

3. 동사 구절 형성 예제

- *동사에 이끌리는 구절*

  (예문) 농어민 생활향상을 위하는 경제 단체

  (VP (NP 농어민 생활향상)+을/jco 위하/pvg)+는
  VP ---> NP+jco pvg

- *서술격 조사에 이끌리는 구절*

  (예문) 민법의 중심적 기본 개념인 권리는 그 대상이 물건이다.

  (VP (NP 물건)+이/jp)+다.
  VP ---> NP+jp

- *동사구절의 연결*

  (예문) 달면 삼키고, 쓰면 뱉어서는 안되지.

  (VP (VP 달면 삼키)+고/ecc+,/sp
  (VP 쓰면 뱉))+어서는 안되지.
  VP --> VP+ecc+sp VP

- *형용사구절과의 연결*

  (예문) 잔주름살이 없고 살갗이 처지지 않아야 하기 때문에...

  (VP
  (ADJP 잔주름살이 없)+고/ecc
  (VP 살갗이 처지지 않아야 하))+기
  VP ---> ADJP+ecc VP

- *동사구절의 병렬*

  (예문) 종로 보신각의 종소리를 듣고 잠을 자고 잠에서 깨어나다.

  (VP
  (VP 종로 보신각의 종소리를 듣)+고/ecc
  (VP 잠을 자)+고/ecc
  (VP 잠에서 깨어나))+다.
  VP ---> VP+ecc VP+ecc VP

- *보조용언과의 연결*
  (예문) 깊숙이 스며 있는 각질을 피부 표면으로부터 없애 준다.
  (VP (VP 깊숙이 스미)+(AUXP 어 있))+는 각질을
  VP ---> VP+AUXP

4. 부사 구절 형성 예제

- *단순 부사구절*
  (예문) 자연을 잘만 활용하면 우리의 피부도 나날이 새롭게 태어난다.
  (ADVP 잘/mag+만/jxc) 활용하면
  ADVP ---> mag+jxc

- *수식을 받는 부사구절*
  (예문) 이것은 독자로 하여금 스스로 생각하게 하는 매력적인 책이다.
  (ADVP (NP 독자)+로/jca 하여금/mag)
  ADVP ---> NP+jca mag

- *관형어의 수식을 받는 부사구절*
  (예문) 양산은 마치 그런 선물이 싫다는 듯이 행동한다.
  (ADVP
  (ADJP 마치 그런 선물이 싫)+다는/etm 듯이/mag)
  ADVP ---> ADJP+etm mag

5. 관형사 구절 형성 예제

- *단순 관형사*
  (예문) 그 동물들은 스스로 체내에 단백질을 쌓아 두고 있다.
  (MODP 그/mmd)
  MODP --> mmd

- *부사의 수식을 받는 관형사구절*
  (예문) 아주 새 옷을 더럽혔구나.
  (MODP (ADVP 아주) 새/mma)
  MODP --> ADVP mma

- *병렬 형태의 관형어구*
  (예문) 학교에서의, 또는 집에서의 행동을 주목하라.
  (MODP 학교/ncn+에서/jca+의/jcm+,/sp 또는/maj 집/ncn+에서/jca+의/jcm)
  MODP --> ncn+jca+jcm+sp maj ncn+jca+jcm

6. 독립구절 형성 예제

- *감탄사, 호격조사의 결합, 제시어, 표제어*
  (예문) 자, 이것이 무엇을 의미하는가를 한번 생각해 보자.

(IP 자/ii)+,/sp
IP ---> ii

- *호격조사의 결합*
  (예문) 영희야, 더운데 창문 좀 열어라.
  (IP (NP 영희/nq)+야/jcv)+,/sp
  IP ---> NP+jcv

- *문장 접속 부사 구절*
  (예문) 옛 것에 대한 정서, 다시 말하면 전통이나 관습 따위를....
  (IP 다시/mag 말하/pvg+면/ecs)
  IP ---> mag pvg+ecs

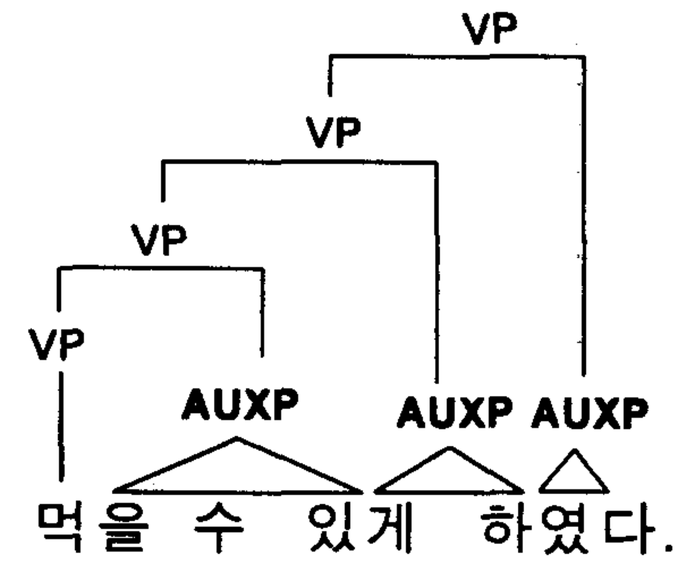
7. 보조 용언 구절 형성 예제

다음과 같은 경우에 보조 용언 구절을 형성한다.

- 시제(tense) : 었,았,겠,더,리,았었,어야겠,었겠
- 존칭(honorific) : 시
- 양상(aspect) : 보조적 연결 어미(ecx) + 보조 용언(px)의 형태

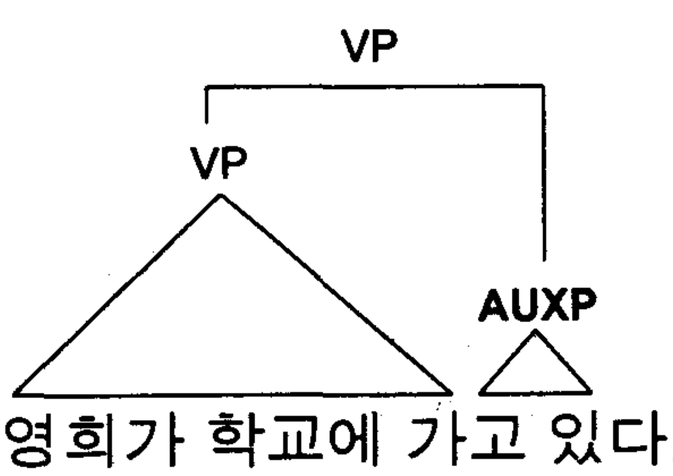
ㄹ/etm 수/nbn 있/paa	ㄹ/etm 수/nbn 없/paa
ㄹ/etm 리/nbn 없/paa	ㄹ/etm 뿐/nbn+만/jxc 아니/paa
ㄹ/etm 뿐/nbn+이/jp	ㄹ/etm 것/nbn+이/jp
ㄹ/etm 것/nbn+뿐/xsn+이/jp	ㄴ/etm 것/nbn 같/paa
ㄴ/etm 셈/nbn+이/jp	기/etn 때문/nbn+이/jp
기/etn 마련/nbn+이/jp	게/ecx 마련/nbn+이/jp
기/etn+도/jxc 하/px	기/etn+만/jxc 하/px

- 보조 용언의 계층적 구조 : 보조 용언이 중복되어 나타날 경우, 다음의 그림 3.2와 같이 용언과 가장 가까이 있는 보조 용언 구절부터 계층적으로 접속하여 용언 구절을 형성한다.



< 그림 3.2 : 보조 용언 구절의 계층적 구조 >

- 보조 용언의 수식 범위 : 보조 용언의 수식 범위는 그 용언에 의해서 하위 범주화 되는 단어들에만 영향을 미치도록 한다. 다음 그림 3.3에서 보조 용언의 수식 범위를 도형적으로 보이고 있다.



< 그림 3.3 : 보조 용언 구절의 수식 범위 >

- *보조 용언 구절 형성 예제*

(예문) 마치 절단되지 않았던 것처럼 말이다.

절단되+(AUXP 지/ecx 않/px)+(AUXP 았더/ep)

AUXP ---> ecx px

AUXP ---> ep

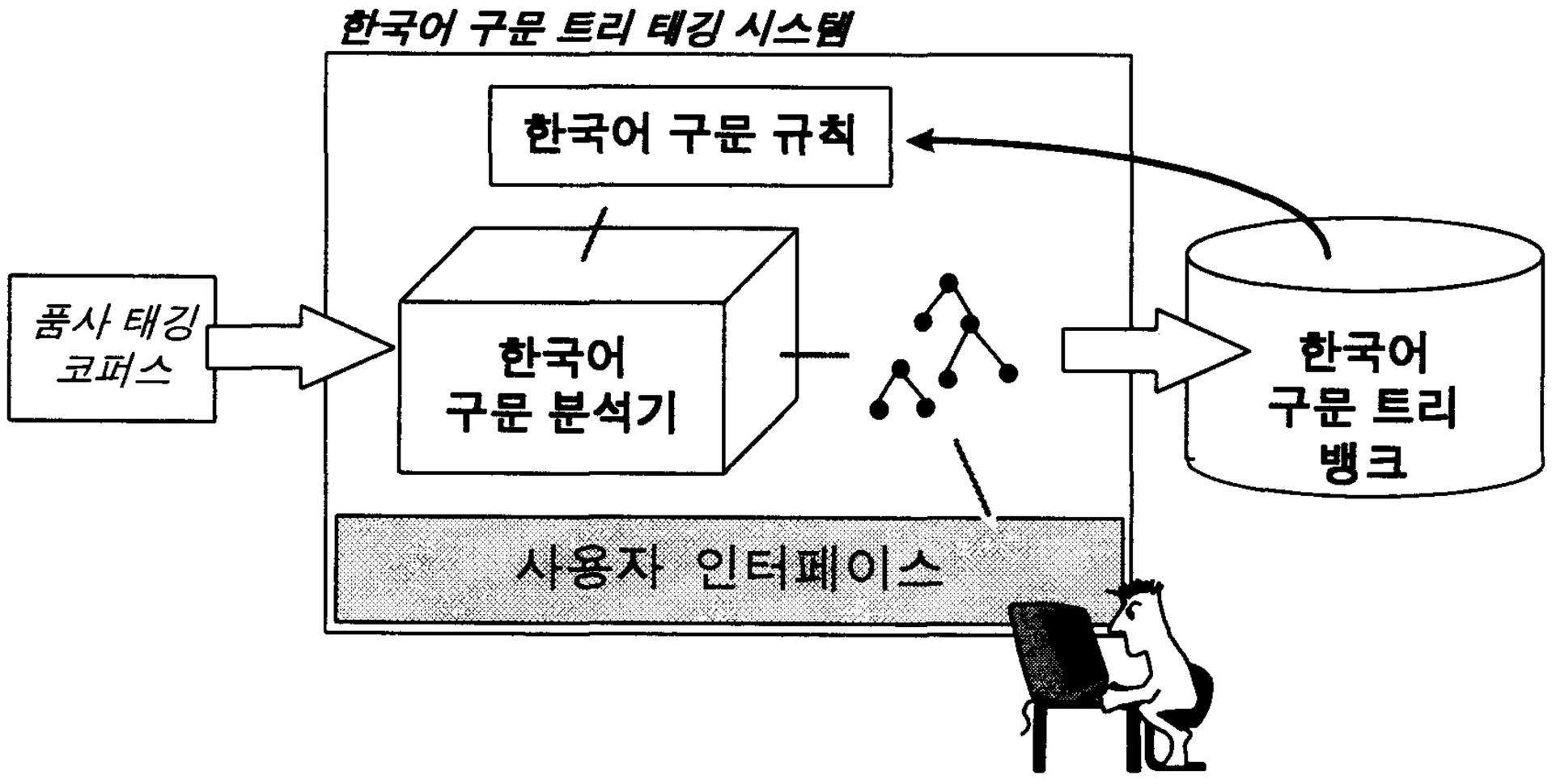
(예문) 초가집으로 바뀔 수밖에 없었습니다.

바뀌+(AUX ㄹ/etm 수/nbn+밖에/jxc 없/paa)+(AUX 었/ep)+습니다.

AUXP --> etm nbn+jxc paa

AUXP --> ep

3장. 한국어 구문트리 태깅 시스템



<그림 3.4 : 구문 트리 태깅 시스템의 구성도>

본 연구에서 개발하는 구문 트리 태깅 시스템의 구성은 그림 3.4 와 같다. 한국어 구문 트리 태깅 시스템은 형태소 단위의 품사가 태깅 되어진 문장을 입력으로 받아, 한국어 구문 분석기의 분석 과정을 거친다. 구문 분석기가 결과로 내주는 결과 구문 트리를 사용자가 수정을 거친 후, 그 결과를 구문 트리 뱅크에 저장한다. 본 연구에서 개발하는 구문 트리 태깅 시스템은 기본적으로 UNIX system 을 platform 으로 하여 구축한다. UNIX system 으로 구축된 구문 트리 태깅 시스템을 Windows 95 버전으로 포팅하여 PC 에서도 쉽게 사용할 수 있도록 한다.

본 연구에서 개발하는 구문 트리 태깅 시스템은 첫째, 구문 트리 태깅 결과의 일관성을 유지할 수 있어야 하며 둘째, 구문 트리 태깅 작업의 속도와 정확도를 향상시킬 수 있어야 한다. 또한, 트리 태깅 작업을 수행하는 사람에 의한 수정 결과가 구문 트리 태깅 시스템에 반영될 수 있어야 한다. 본 연구에서는 이와 같은 조건들을 기반으로 하는 태깅 도구를 개발하였다.

1 절. 한국어 구문트리 태깅을 위한 부분 구문 분석기

구문 트리 태깅 도구는 사람이 구문 트리 태깅 작업을 경제적으로 수행할 수 있도록 도와주어야 한다. 여기서 구문 트리 태깅 작업이 경제적이라 함은 구문 트리 태깅에 소요되는 시간과 구문 트리의 재괄호 치기(re-bracketting) 과정의 빈도

수가 적을 수록 구문 트리 태깅 작업이 경제적이라고 할 수 있다[Marcus et al 1993]. 본 연구 과제에서 사용하는 구문 트리 태깅 도구로서의 구문 분석기는 다음과 같은 특징을 지니고 있다.

- 구문 태깅 도구로서의 구문 분석기는 정확히 하나의 분석 결과만을 제시하여 준다. 그렇게 함으로써, 구문 태깅을 하는 사람은 여러 개의 분석 결과 중에서 정확한 구문 트리를 찾기 위한 탐색 작업이 필요 없게 된다. 일반적으로 여러 개의 분석 결과로부터 한 개의 정확한 결과를 찾는 것보다는 하나의 분석 결과를 정확한 결과가 되도록 수정하는 것이 경제적이다.
- 구문 트리 태깅 도구로서의 구문 분석기는 또한, 견고해야 한다. 입력 문장에 오류가 존재한다든지, 시스템이 갖고 있는 문법의 부족으로 구문 분석의 결과가 나오지 않는 경우라도 부분적인 분석 결과를 제시함으로써, 구문 트리 태깅을 수행하는 사람으로 하여금 적절한 조치를 취할 수 있도록 해야 한다.
- 구문 트리 태깅 도구로서의 구문 분석기는 구문 분석기가 단순한 구문 정보만으로는 분석 결과를 결정하기 어려운 상황에 대해서는 미정인 채로 결과를 제시하고, 그 결정은 구문 태깅을 수행하는 사람에게 일임하도록 한다. 이렇게 하는 것이 재괄호 치기의 과정을 거의 제거하여 주기 때문에 구문 트리 태깅 과정을 더 경제적으로 수행할 수 있도록 해준다.

이러한 여러 가지 고려 사항을 종합해 보았을 때, 구문 트리 태깅 도구로써의 구문 분석기는 구문적 정보만으로도 다소 정확한 분석 결과를 제시해 줄 수 있는 부분적 구문 분석(partial parsing)을 수행한다. 부분 구문 분석은 모호성이 그리 많지 않은 부분에 대한 분석만을 수행하고 모호성을 많이 유발시키는 구조는 생성하지 않는 분석 방법으로써, 영어의 경우, 전치사구 부착(PP-attachment) 문제와 같이 분석 결과의 모호성을 급증시키는 부분은 제외하고 분석을 수행한다. 이렇게 함으로써, 부분 분석의 결과는 입력 문장에 대해, 하나의 구문 트리가 아닌, 여러 개의 부분 구문 트리가 된다. 본 연구에서의 사용하는 부분 구문 분석은 규칙 기반의 분석을 수행한다. 본 연구에서의 부분 구문 분석기가 사용하는 규칙의 형태는 다음과 같다.

Front_condition | *sub_string* | *rear_condition* ---> *sub_structure*

< 그림 3.5 : 부분 구문 분석기가 사용하는 규칙의 형태 >

그림 3.5 에서 세로 막대(vertical bar)에 둘러 쌓여 있는 부분이 부분 분석되어질 대상 스트링이다. 또한, 세로 막대 앞과 뒤에는 각각 규칙이 적용될 수 있는 조건이 기술되어 있다. 그림 3.5 의 규칙은 *front_condition* 과 *rear_condition* 이 만족되어질 경우, *sub_string* 을 *sub_structure* 로 부분 분석하라는 의미이다.

(1)	| Q/etm 수/nbn 있/paa |	→	(AUXP W W W)
(2)	| Q/etm 수/nbn 없/paa |	→	(AUXP W W W)
(3)	| Q/etm 리/nbn 없/paa |	→	(AUXP W W W)
(4)	| Q/etm 수/nbn + Q/jxc 있/paa |	→	(AUXP W W+W W)
(5)	| Q/etm 수/nbn + Q/jxc 없/paa |	→	(AUXP W W+W W)
(6)	| Q/etm 리/nbn + Q/jxc 없/paa |	→	(AUXP W W+W W)
(7)	| Q/etm 수/nbn + Q/jxt 있/paa |	→	(AUXP W W+W W)
(8)	| Q/etm 수/nbn + Q/jxt 없/paa |	→	(AUXP W W+W W)
(9)	| Q/etm 리/nbn + Q/jxt 없/paa |	→	(AUXP W W+W W)
(10)	Q/J | Q/N + 적/xsn + 이/jp |	→	(VP (NP W+W)+W)
(11)	Q/J | Q/N + 이/jp |	→	(VP (NP W)+W)
(12)	Q/J | Q/N + Q/N + 이/jp |	→	(VP (NP W+W)+W)
(13)	Q/eos | Q/N + Q/jco Q/pvg |	→	(VP (NP W)+W W)
(14)	Q/eos | Q/N + Q/jco Q/ncpa + Q/xsv |	→	(VP (NP W)+W W+W)
(15)	Q/eos | Q/N + 들/xsn + Q/jco Q/pvg |	→	(VP (NP W+W)+W W)
(16)	Q/J | Q/N + 적/xsn Q/N | + Q/J	→	(NP (NP W+W) W)

<그림 3.6 : 부분 구문 분석을 위한 규칙 (일부)>

그림 3.6 은 본 연구에서 사용하는 부분 분석을 위한 규칙의 일부이다. '어휘/품사'의 형태로 규칙을 기술하고 있다. 'Q'는 임의의 어휘라는 의미이며, 'N'은 체언류를 'J'는 조사류를 의미한다. ('S'는 기호류, 'P'는 용언류, 'M'은 부사/형용사류, 'E'는 어미류의 일부를 의미) 또한 *sub_structure* 에서 'W'는 부분 스트링(*sub_string*)에서 하나의 형태소를 의미한다. 규칙 (1)에서 (9)번 까지는 보조용언구절을 형성하는 규칙으로서, 앞, 뒤 조건(condition)에 상관없이 규칙이 적용되는 경우이다. (16)번 규칙의 경우, 앞에 조사류(J)가 나타나고 뒤에 다시 조사류(J)가 발생할 경우, 스트링 (명사류 + 적 명사류)를 (NP (NP W+W) W)의 구조로 분석하라는 규칙이다. 현재 위와 같은 부분 분석을 위한 규칙은 언어적 지식을 갖고 있는 사람에 의해 기술되어 사용되고 있다. 이를 이용한 부분 분석 과정은 다음과 같

이 진행되어 진다.

```
for input sentence W = W1 W2 W3...Wn
current_position = 1;

do
        i = current_position ;
        search the rule r that cover the longest substring Wi,Wi+1,Wi+2...Wi+j ;
        apply rule r into the substring Wi ...Wj+j;
        set current_position = i+j+1 ;
while current_position is not end of sentence
```

위의 pseudo-code 에서 알 수 있듯이 최장 일치법을 사용하여 규칙을 적용한다. 만약 같은 길이의 규칙이 동시에 적용이 된다면, 규칙이 나온 순서에 따라서 적용하게 된다. 즉, 규칙은 우선 순위를 가지고 있다.

2 절. 구문트리 태깅 툴의 기능

구문 트리 태깅 툴이 갖는 기본적인 기능은 다음과 같다. 트리를 태깅하고 저장하는 과정 이외에도, 트리의 입력이 되는 품사 태깅된 말뭉치의 원문/태깅에 에러가 있을 경우에 수정하는 기능 또한 필요하다.

Tree tagging	• 부분 트리의 선택 (*marking subtree*) • 부분 트리의 구문 태깅 (*syntactic labelling*) • 부분 트리의 수정 (*breaking subtree*) • 결과 구문 트리를 보여주는 기능 (*drawing tree*)
Tree saving	• 구문 태깅된 문장을 파일에 저장하는 기능 (*saving tree*) • 입력 문장을 제거하는 기능 (*deleting tree*) • 트리 태깅 도구의 구문 문법 규칙의 선호도 조절 기능
Tree Correction	• 입력 원문의 오류 수정 기능 • 입력 원문의 품사 오류 수정 기능

본 연구 과제에서 제공하는 구문 트리 태깅 도구는 UNIX 시스템의 CURSES Library 를 이용하여 구축하였다. 기본적으로 마우스(mouse)를 이용하여 태깅을 수행한다. 이렇게 개발된 UNIX 시스템의 구문 트리 태깅 도구를 비슷한 사용자 인터페이스를 가진 Windows 95 버전으로 포팅하였다. 사용법 또한 비슷하기 때문에,

여기에서는 UNIX 버전을 중심으로 설명한다. Windows 95 버전의 경우에는 기능을 단순화하기 위하여 아래에 설명한 내용 중에서 품사 태깅된 문장의 원문/품사 오류를 수정하는 기능은 제외하였다.

구문 트리 태깅을 위해 부분적으로 분석되어진 부분 트리를 마우스를 click 함으로써 선택할 수 있다. 부분 트리가 선택되어지면, 메뉴에 따라 구문 트리 태깅을 할 수 있다. 또한, 잘못 태깅 되어진 부분에 대해서는 마우스의 오른쪽 버튼을 이용하여 부분 트리의 수정을 수행할 수 있다. 구문 트리가 태깅 되어질 때마다 새로이 형성되는 부분 트리가 화면에 다시 그려진다. 한 문장에 대한 구문 트리 태깅이 완료되면, 저장 기능을 이용하여 결과 구문 트리를 화일에 저장하게 된다. 또한, 입력 문장에 오류가 존재하여 구문 트리의 가치가 없을 경우에는 그러한 입력 문장을 제거하는 기능 또한 갖고 있다.

구문 트리 태깅 툴은 한텀(hanterm)에서 동작하며, 화면의 크기가 적어도 110x32 이상이 되어야 한다. 현재는 트리 태깅을 하는 작업 공간이 스크롤되지 않기 때문에 가능한 큰 화면에서 작업하여야 하며, 이러한 화면을 넘어가는 트리 태깅 문장은 툴 내에서 작업할 수 없고, vi 와 같은 에디터로 작업하여야 한다. 사용법은 다음과 같다.

% teb [-s|m|p] *infile outfile*

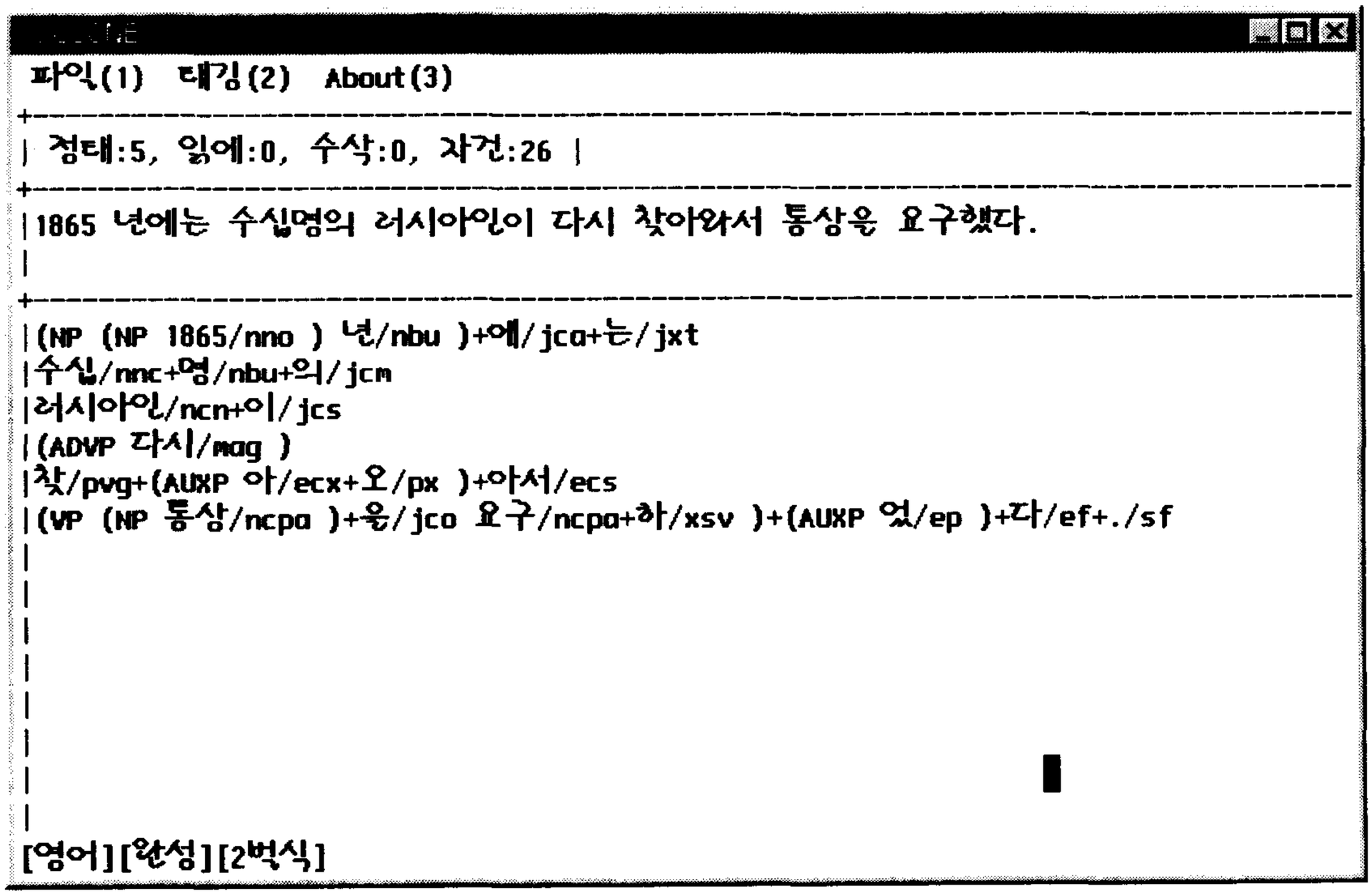
infile 은 트리 태깅을 위한 말뭉치이고, 트리 태깅 툴에서 사람이 트리 태깅한 결과가 새로운 트리 태깅을 위한 말뭉치 화일인 outfile 로 만들어져서 나온다. -s 옵션이 주어지지 않으면, 완전히 태깅된 문장은 툴에서 자동적으로 건너뛰고 태깅이 완전하지 않은 문장만을 보여주고, 태깅을 하도록 한다. 완전히 태깅된 문장들을 검사하는 경우에는 -s 옵션이 필요하다. -m 옵션의 경우에는 마킹(marking)이 되어진 문장에 대해서도 처리할 수 있도록 해준다. -p 옵션의 경우에는 부분 구문 분석만을 수행하도록 해준다.

트리 태깅을 위한 말뭉치는 다음과 같은 점을 준수하여야 한다.

- '#'를 특수하게 사용 : '#'는 구문 트리 태깅에서 주석문으로 처리한다. 구문 트리 태깅 툴에는 전혀 뜻이 없으며, *infile* 에 있는 것이 *outfile* 의 같은 위치에 그대로 나타나게 된다.
- ';'를 특수하게 사용 : ';'는 다음에 나오는 트리 태깅된 문장의 원문을

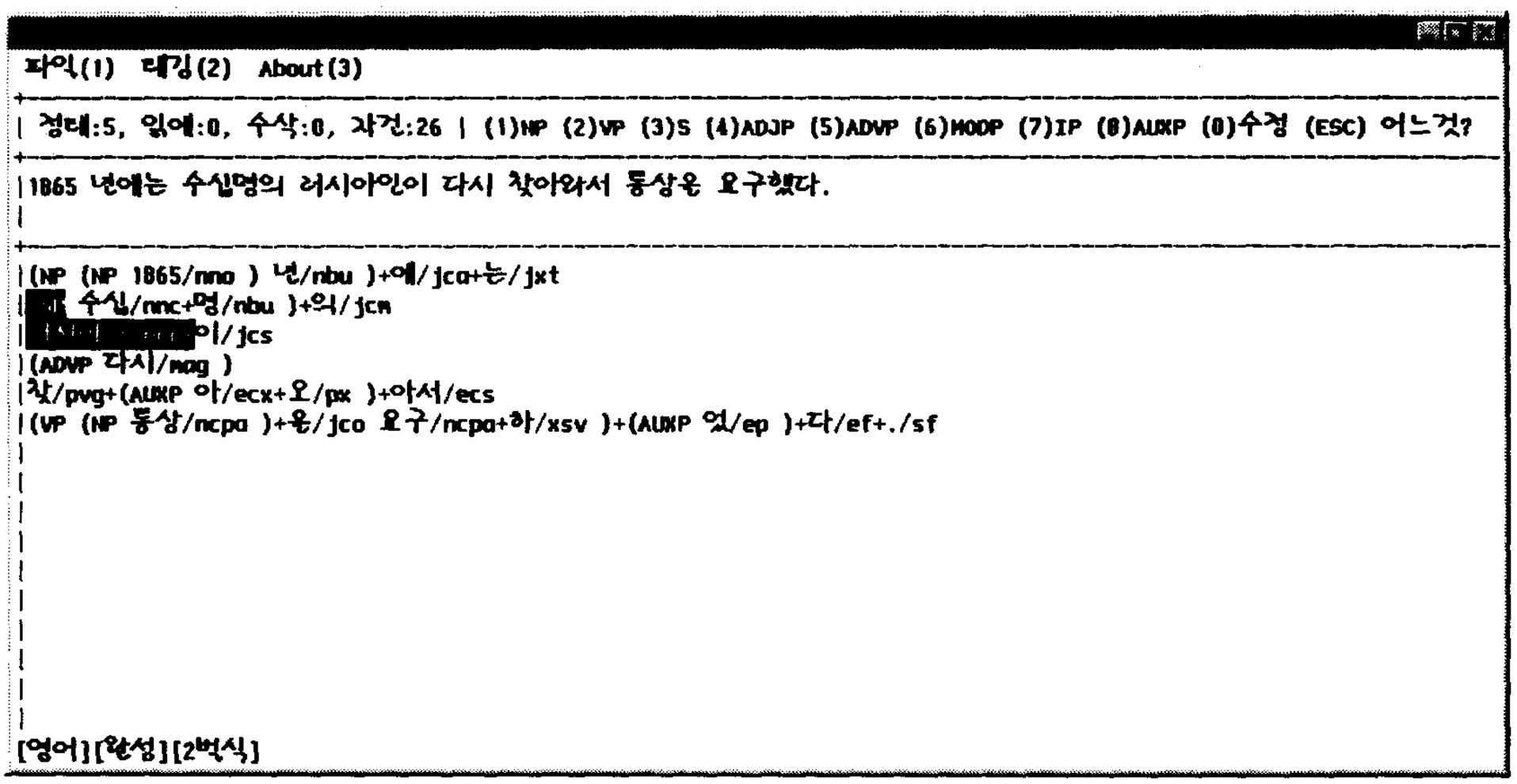
나타낸다. 트리 태깅된 문장에서 원문을 정확히 얻어낼 수 없기 때문에 (형태소 해석 단계에서 일어나는 불규칙과 같은 처리 때문이다. 예를 들어, 어두운 -> 어둡/pvg+운/etm 으로의 복구가 쉽지 않다) 이 항목이 꼭 필요하다. 그리고 이 원문은 하나의 라인에 모두 들어가야 한다.

- 트리 태깅 문장은 공백 라인으로 구분 : 하나의 트리 태깅 문장과 다음 문장은 하나의 공백 라인으로 구분된다. 그리고 ';'로 시작하는 원문은 현재의 트리 태깅 문장의 바로 위의 것을 사용한다. 트리 태깅 문장은 공백 라인이 나올 때까지 몇 개의 라인으로 되어 있어도 상관없다.
- 트리 태깅 문장에서 품사가 틀릴 경우나 괄호의 갯수가 맞지 않아서 트리 생성이 실패할 경우에는 자동적으로 그 문장은 지우게 된다. 화면의 "읽기에러" 항목은 이렇게 자동적으로 지워진 문장의 수를 나타낸다.



< 그림 3.7 : 구문 트리 태깅 툴의 화면 I >

그림 3.7 은 구문 트리 태깅 툴의 시작 화면이다. 구문 트리 태깅 툴은 입력 문장에 대해, 앞서 언급했던 부분 구문 분석기를 수행하여 부분적인 구문 분석 결과로서 그림 3.8 과 같은 결과를 제시한다. 구문 트리 태깅을 수행하는 사람들은 부분 구문 분석 결과를 확인하고, 이를 기반으로 하여 나머지 트리의 구조를 완성해 나갈 수 있다.



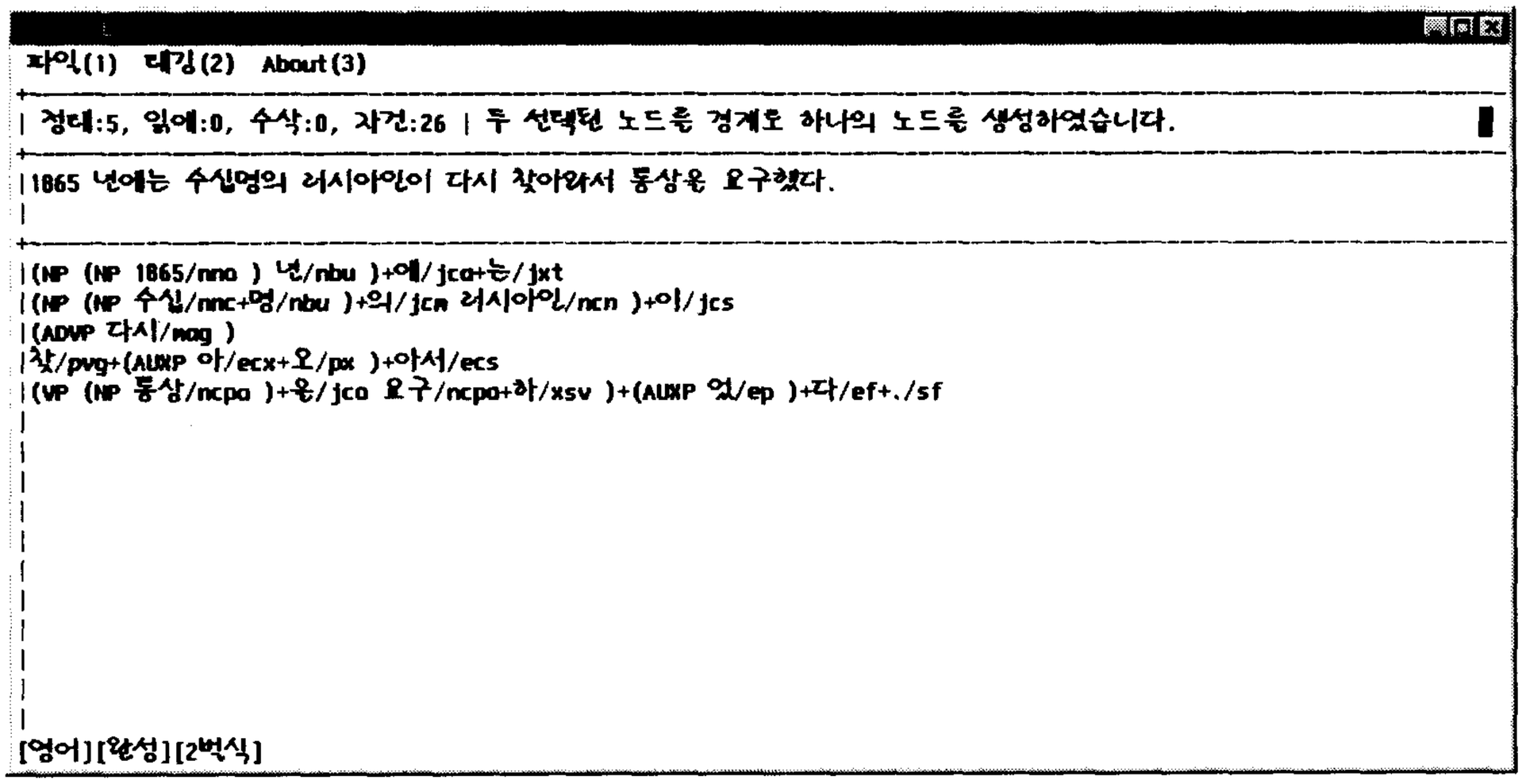
< 그림 3.8 : 구문 트리 태깅 툴의 화면 II >

그림 3.8 은 구문 트리 태깅을 수행하는 과정이다. 하나의 구절을 형성하기 위하여 (NP 수십/nnc+명/nbu)와 '러시아인/ncn'가 선택되어 반전되어 나타나 있다. 선택하고자 하는 문장의 요소 위에서 마우스의 왼쪽 버튼을 누르면 선택이 되고, 선택의 표시로서 글자가 반전되어 나타난다. 하나의 구절을 형성하기 위하여 두 구성 요소가 선택되고 나면, 화면의 위에서 보이는 것과 같이

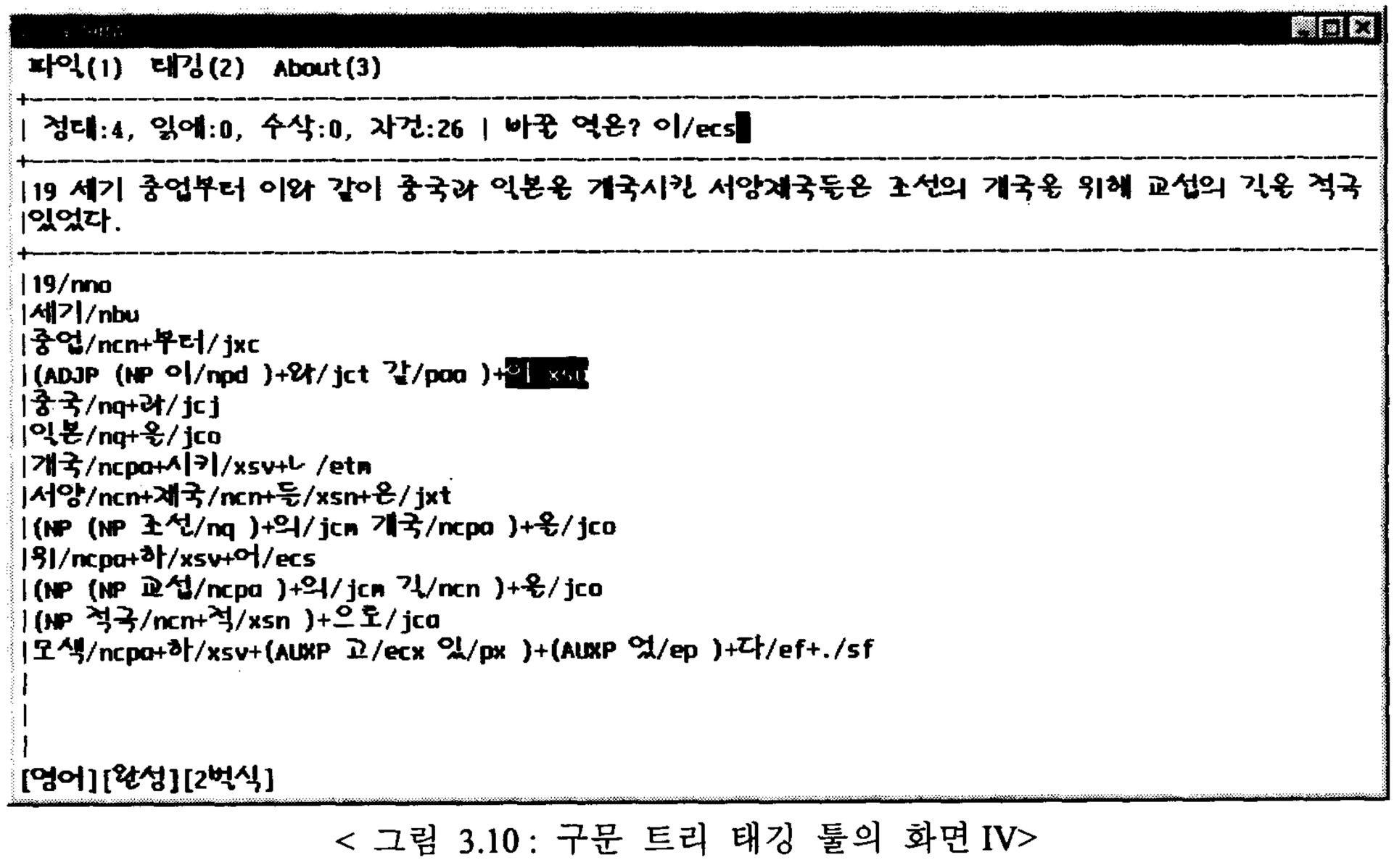
(1)NP (2)VP (3)S (4)ADJP (5)ADVP (6)AUXP (7)IP (8)MODP (ESC) 어느것?

와 같은 물음이 나온다. 이때, 그 구절에 해당하는 번호를 입력함으로써, 하나의 부분 구조(subtree)를 형성할 수 있다. 다음의 그림 3.9 에는 위의 그림 3.8 에서 명사구를 형성한 결과를 보이고 있다. 이미 형성한 부분 구조를 깨뜨릴 경우에는 그 구조의 레이블에서 마우스의 오른쪽 버튼을 누르면, 해당 부분 구조가 제거된다. 하나의 문장에 대한 구문 트리 태깅을 마쳤을 경우에는 's' 버튼을 누름으로

써, 해당 문장과 그 구문 트리 구조가 화일에 저장된다.



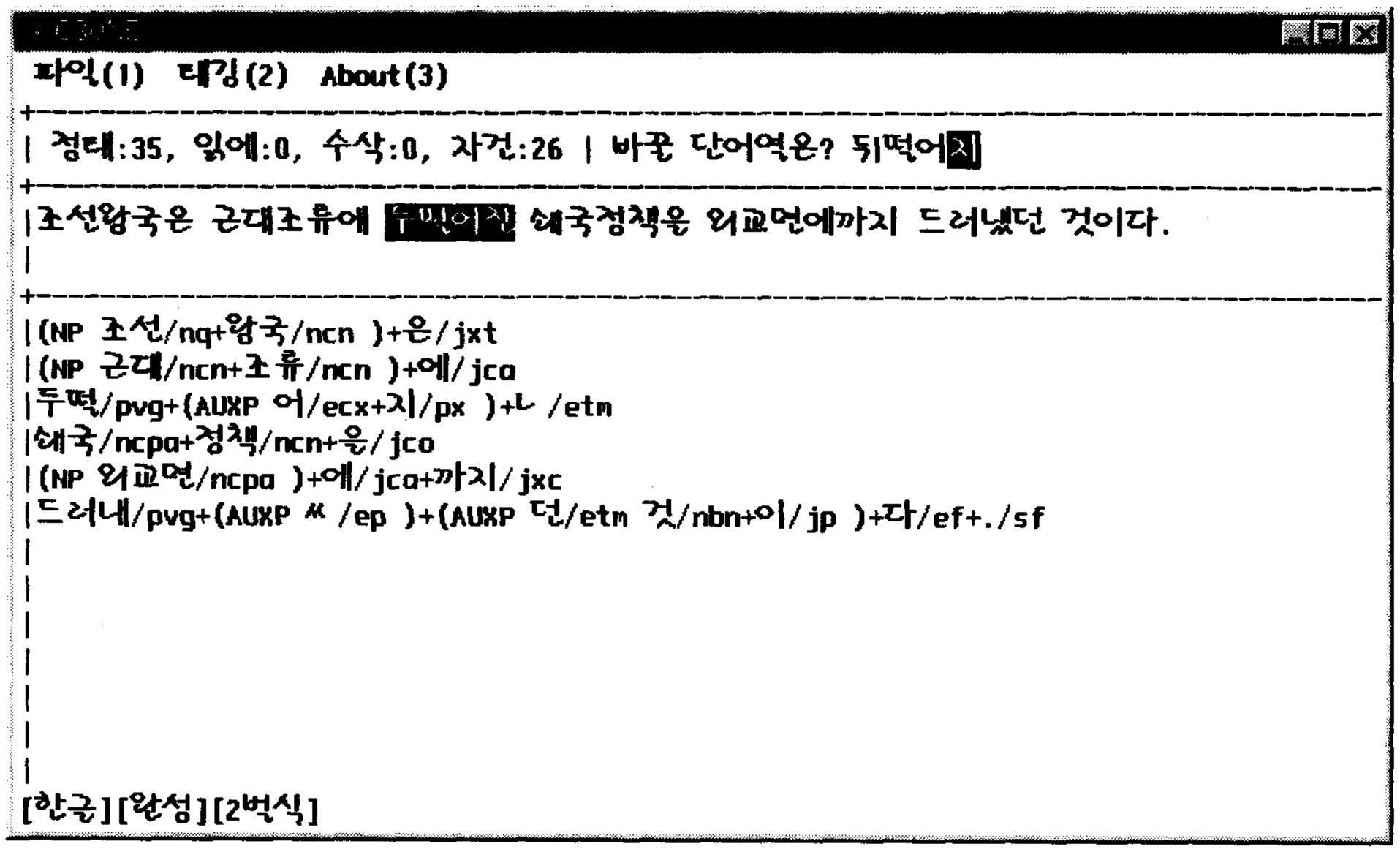
< 그림 3.9 : 구문 트리 태깅 툴의 화면 III >



< 그림 3.10 : 구문 트리 태깅 툴의 화면 IV>

그림 3.10 에는 입력 문장에 있는 오류를 수정하는 과정을 보이고 있다. 그림에

서 반전되어 나타나고 있는 부분의 품사를 수정하기 위해서, 마우스를 이용하여 수정할 부분을 선택하고, 이를 수정하는 과정을 보이고 있다.



< 그림 3.11 : 구문 트리 태깅 툴의 화면 V>

그림 3.11 에서는 원문의 오류를 수정하는 과정을 보이고 있다.

이러한 품사 태깅된 말뭉치의 원문/품사를 수정하는 기능은 구문 트리 태깅 도구에서 빠질 수 없는 부분이다. 품사 태깅을 하는 경우에는 일반적으로 구문 트리 태깅을 전혀 생각하지 않기 때문에 발생하는 에러도 있으며, 품사 태깅 자체에서의 에러, 원문의 에러가 있기 때문이다. 좀더 효율적인 태깅 과정을 위해서는 품사 태깅 워크벤치와의 연계 방안을 고려할 필요가 있다. 하지만, 본 연구의 시스템 구현상에서 연계까지 고려하기 위해서는 일의 범위가 너무 커지기 때문에, 일단 기초적인 에러 수정 기능만을 구현하였따.

3 절. 한국어 구문 트리 뱅크

한국어 구문 트리 뱅크는 다음과 같이 구성되어 있다. 현재 약 10 만 문장에 대하여 구문 트리를 태깅하여 구문 트리 뱅크가 구축되어 있으며, 구문 트리 뱅크는

다음과 같은 형태로 저장되어 있다. 우선, 하나의 출처를 지닌 원문은 하나의 화일에 저장시키도록 하였으며, 화일의 제일 처음에는 원문에 관한 정보를 덧붙여 놓음으로써, 그 출처를 밝히고자 하였다. 또한, 한국어는 첨가어로서 형태소 분석 결과로부터 원문을 추출하는 것이 어렵다. 그렇기 때문에, 각각의 문장에 대해서, 그 구문 트리 구조뿐만 아니라, 원문까지도 함께 저장하도록 하였으며, 문장과 문장 사이에는 반드시 하나 이상의 빈줄을 첨가함으로써, 문장의 분리를 확실히 수행하도록 하였다. 또한, 구문 트리 태깅된 문장의 경우에는 인덴테이션(indentation)을 적절하게 자동으로 만들어줌으로써, 말뭉치만 보면 어렵지 않게 구문 트리 태깅 결과를 바로 알 수 있도록 하였다. 그림 3.12 에는 구축된 구문 트리 태깅 코퍼스의 일부를 보이고 있다.

```
##########################################################
#
#    Tree Tagged Corpus   1996. 11. 2
#    원문 정보
#
#
##########################################################
; 우리가 이웃인 까닭은 가까이 있음이 아니고 따뜻한 정을 나눔입니다.
(S
   (VP
      (NP
         (VP (NP 우리/npp )+가/jcs
            (VP (NP 이웃/ncpa )+이/jp ))+ㄴ/etm 까닭/ncn )+은/jxt
      (VP
         (ADJP
            (NP
               (ADJP (ADVP 가까이/mag ) 있/paa )+ㅁ/etn )+이/jcs
            아니/paa )+고/ecc
         (VP
            (NP
               (VP
                  (NP (ADJP 따뜻하/paa )+ㄴ/etm
                     정/ncn )+을/jco 나누/pvg )+ㅁ/etn )+이/jp )))+ㅂ니다/ef+./sf )

; 기쁨을 나누면 배가 되고 슬픔을 나누면 반이 됩니다.
(S
   (VP
      (VP
         (VP (NP 기쁨/ncn )+을/jco 나누/pvg )+면/ecs
         (VP (NP 배/ncn )+가/jcc 되/pvg ))+고/ecc
      (VP
         (VP (NP 슬픔/ncn )+을/jco 나누/pvg )+면/ecs
         (VP (NP 반/ncn )+이/jcc 되/pvg )))+ㅂ니다/ef+./sf )

; 대도시의 도로는 이제 포화상태입니다.
```

```
(S
    (VP (NP (NP 대도시/ncn )+의/jcm 도로/ncn )+는/jxt
        (VP (ADVP 이제/mag )
            (VP (NP 포화상태/ncn )+이/jp )))+ㅂ니다/ef+./sf )

; 안국화재는 항상 여러분을 든든히 지켜드리고 있습니다.
(S
    (VP
        (VP
            (VP (NP 안국화재/nq )+는/jxt
                (VP (ADVP 항상/mag )
                    (VP (NP 여러분/npp )+을/jco
                        (VP (ADVP 든든히/mag )
                            지키/pvg ))))+(AUXP 어/ecx+드리/px ))+(AUXP 고/ecx 있/px ))+습니다/ef+./sf )

; 미리미리 준비하는 지혜가 필요합니다.
(S
    (ADJP
        (NP
            (VP (ADVP 미리미리/mag ) 준비/ncpa+하/xsv )+는/etm
            지혜/ncn )+가/jcs 필요/ncps+하/xsm )+ㅂ니다/ef+./sf )
```

< 그림 3.12 : 구문 트리 태깅 코퍼스 (일부)>

4장. 한국어 구문 분석기의 개발

1절. 확률 기반의 한국어 구문 분석기

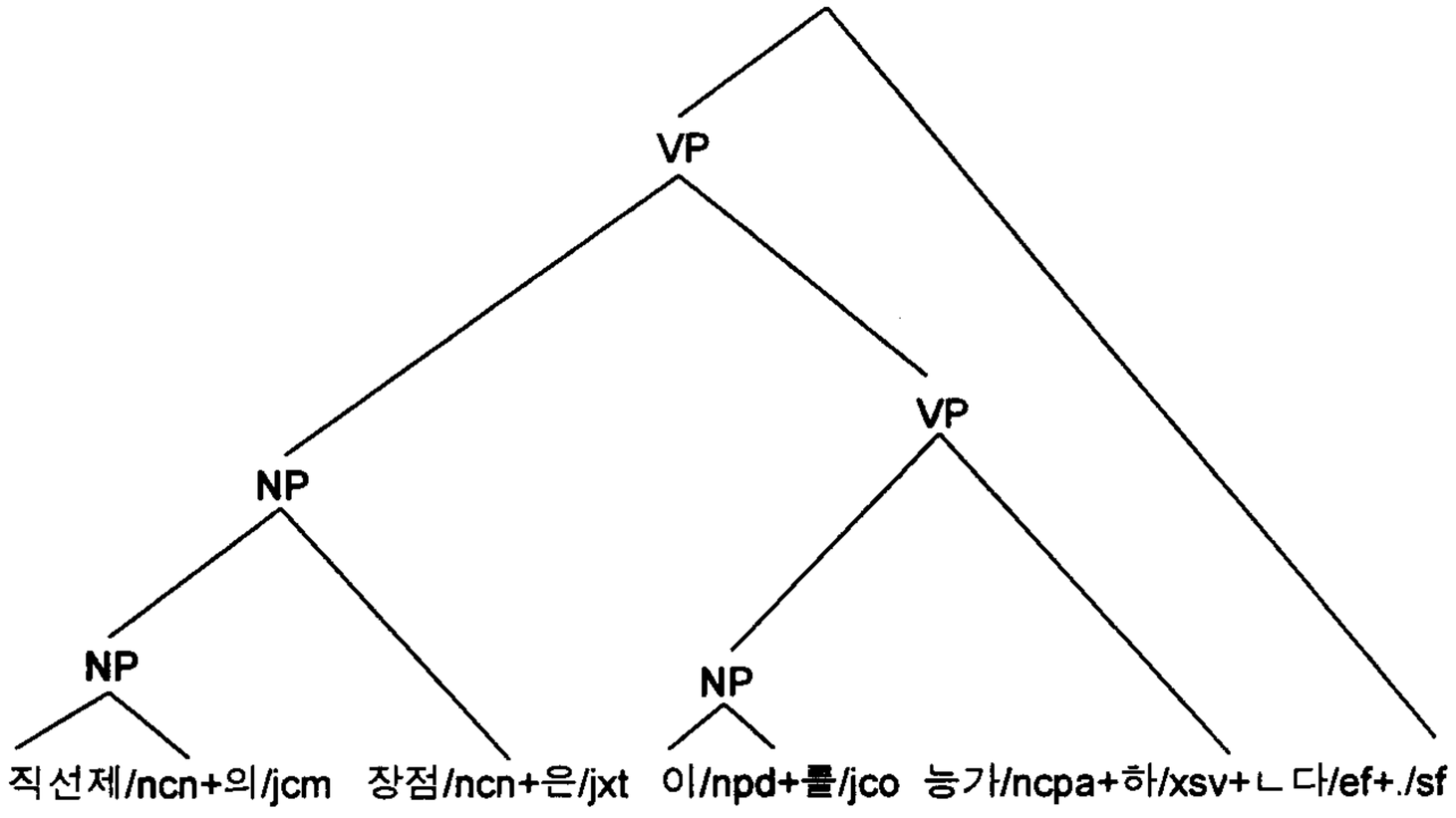
본 연구에서는 크게 두 종류의 구문 분석기가 구현되었다. 첫째로는 앞절에서 언급한 구문 트리 태깅 툴을 위한 부분 구문 분석기이고, 둘째로는 확률 모델을 기반으로 하는 완전 구문 분석기이다. 본 절에서는 확률 모델을 기반으로 하는 구문 분석기의 구현에 대해 기술하고자 한다.

구문 분석기가 사용하는 각 규칙에는 그 선호도에 따른 확률값이 부여되어 있다. 이와 같은 규칙의 확률값은 구문 트리 태깅된 코퍼스로부터 자동으로 학습할 수 있다. 확률적 언어 모델은 입력 문장에서 가능한 각각의 구문 트리에 대해 그 구문 트리에 적절한 값을 부여해 줄 수 있으며, 또한, 이를 통해서 여러 개의 가능한 구문 트리 중에서 가장 적절한 구문 트리를 선택해 줄 수 있다. 본 연구에서 사용한 구문 분석에 사용하는 확률적 언어 모델은 다음과 같다.

$$P(T|S) = \prod_{A \in T} P(A \rightarrow \alpha | a_l, a_r) \qquad \text{(식 1)}$$

식 1 에서 S 는 입력 문장을 의미하며, T 는 이에 해당하는 구문 트리를 의미한

다. al 과 ar 은 각각 문법 규칙 $A \rightarrow \alpha$가 적용되는 오른쪽과 왼쪽의 문맥에 해당하는 품사 정보를 의미한다. 이 언어 모델은 문법 규칙이 적용될 경우, 그 적용되는 규칙의 양쪽 단어(양쪽 문맥)에 대해서만 의존적이라는 가정을 토대로 삼고 있다.



<그림 3.13 : 구문 트리 예제 >

$$P(T|S) = P(S \rightarrow VP + ef + sf | eos, eos) \cdot P(VP \rightarrow NP + jxt \quad VP | eos, ef)$$
$$\cdot P(NP \rightarrow NP + jcm \quad NP | eos, jxt) \cdot P(VP \rightarrow NP + jco \quad ncpa + xsv | jxt, ef) \quad \text{(식 2)}$$
$$\cdot P(NP \rightarrow ncn | eos, jcm) \cdot P(NP \rightarrow npd | jxt, jco)$$

식 1 의 확률적 언어 모델을 이용했을 경우, 그림 3.13 에 나타나 있는 문장의 구문 트리에 대한 확률값은 식 2 와 같이 계산되어 질 수 있다.

2 절. 확률 기반의 규칙 학습 및 매개변수 평활화

확률 규칙의 확률값 추론 방법은 기본적으로 최우추정법(Maximum Likelihood Estimation)의 방법을 이용하였다. 본 연구에서 사용하는 규칙에 대한 확률값은 다음과 같이 구할 수 있다.

$$P(A \rightarrow \alpha | a_l, a_r) = \frac{C(A \rightarrow \alpha, a_l, a_r)}{\sum_i C(A \rightarrow \alpha_i, a_l, a_r)} \qquad \text{(식 3)}$$

식 3 에서 $C(\cdot)$는 학습 코퍼스에서 나타나는 빈도수를 의미한다. 이와 같은 학습방법에 의해서 학습되어진 확률 규칙의 일부를 그림 3.14 에서 보이고 있다. 그

림 13 에서 1 열에 나오는 숫자는 그 규칙의 확률값을 의미하며, 규칙 다음에 괄호 안의 정보는 그 규칙에 대한 앞/뒤 문맥 정보를 의미한다.

```
0.338596   VP → VP + AUXP   (eos, ef)
0.019084   VP → VP + AUXP   (jxt, ef)
0.215311   VP → NP + jxt   VP   (eos, ep)
0.076321   VP → NP + jxt   VP   (eos, etm)
0.056272   VP → VP + ecs   VP   (jxt, ef)
0.112746   NP → NP + jcm  nq   (ecc, jcs)
0.098271   ADJP → pad    (jxt, etm)
```

<그림 3.14 : 문맥 의존 확률 기반의 구구조 규칙 (일부)>

대부분의 확률 모델이 갖고 있는 문제와 마찬가지로 위와 같은 문법 규칙도 자료 희귀 문제(data sparseness)를 안고 있다. 본 연구에서는 이를 해결하기 위하여 back-off 방법[Katz 1987]을 이용하여 매개변수 평활화(smoothing)을 수행함으로써 자료 희귀 문제를 극복하였다.

$$P(A \rightarrow \alpha | a_l, a_r) = \begin{cases} \dfrac{C(A \rightarrow \alpha, a_l, a_r)}{\sum_i C(A \rightarrow \alpha_i, a_l, a_r)} & if\ C(A \rightarrow \alpha, a_l, a_r) > 5 \\ d_c \times \dfrac{C(A \rightarrow \alpha, a_l, a_r)}{\sum_i C(A \rightarrow \alpha_i, a_l, a_r)} & if\ 0 < C(A \rightarrow \alpha, a_l, a_r) \le 5 \\ Q(a_l, a_r) \times (\lambda_1 \cdot P(A \rightarrow \alpha | a_l, \_) + \lambda_2 \cdot P(A \rightarrow \alpha | \_, a_r) + \lambda_3 \cdot P(A \rightarrow \alpha)) & otherwise \end{cases} \quad \text{(식 4)}$$

$$Q(a_l, a_r) = \sum_i (1 - d_c) \cdot \frac{C(A \rightarrow \alpha, a_l, a_r)}{C(A \rightarrow \alpha, a_l, a_r)}$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

식 4 에서 dc 는 MLE 에 대한 감소량을 의미한다. 식(4)의 back-off 방법은 $C(A \rightarrow \alpha, a_l, a_r)$의 빈도수가 5 이상일 경우에는 그 값이 매우 신뢰성 있다고 판단되어 지므로 MLE 와 동일한 방법에 의해 확률값을 추정한다. 그러나, $C(A \rightarrow \alpha, a_l, a_r)$이 학습 코퍼스에 발생하지 않았을 경우에는 $C(A \rightarrow \alpha, a_l, _)$와 $C(A \rightarrow \alpha, \_, a_r)$, 그리고 $C(A \rightarrow \alpha)$의 조합에 의해서 그 값을 추정하도록 한다.

5장. 토의 및 결론

구문 트리 태깅 작업은 여러 사람들에 의해 그 작업이 이루어지기 때문에 구문 트리 뱅크의 일관성 유지가 가장 어려운 문제로 남아 있다. 물론, 구문 트리 태깅 툴의 사용으로 구문 트리 태깅 툴이 제시하는 결과에 대해서는 일관성을 유지할 수 있지만, 여러 사람들에 의해 작업되는 많은 부분에서는 그 일관성을 유지한다는 것이 매우 어려운 과제로 남아 있다. 유용한 구문 트리 뱅크 구축을 위해서는 구문 트리 뱅크의 일관성 유지를 위해, 좀 더 많은 방법의 적용이 요구되어진다. 앞으로의 과제는 대용량의 구문 트리 뱅크의 구축도 있겠으나, 이미 구축되어진 구문 트리 뱅크를 양질의 데이타로 만드는 일도 주요 과제라 할 수 있겠다.

구문 트리 태깅을 수행하다 보면, 문장의 구문 트리 태깅 뿐 아니라, 문장 자체의 이해가 어려운 경우가 빈번하다. 또는, 한국어 자체의 많은 문법적 지식의 부족으로 인하여 구문 분석 시 많은 어려움을 겪어 왔다. 즉, 한국어의 다양한 통사적 현상에 대한 많은 고찰이 필요하며, 구문 트리 태깅 자체가 어려운 문장들에 대해서 이를 잘 표현할 수 있는 방법에 대한 연구도 필요하다고 보여진다.

구문 트리 태깅 된 코퍼스는 한국어 처리 분야의 여러 분야에서 많은 정보를 제공해 줄 수 있는 지식의 근원이 된다. 본 연구에서는 이와 같은 트리 태깅된 코퍼스를 구축하는 데 필수적인 구문 트리 태깅 도구를 제공하고 이를 이용하여 구문 트리 태깅된 코퍼스를 구축하였다. 향후 진행 방향으로는 코퍼스 구축에 좀 더 유용하도록 사람의 사용에 따른 피드백을 보완하여 구문 트리 태깅 도구의 편의성을 증강시키고, 구문 트리 태깅된 코퍼스의 양질을 보장하기 위하여 구문 트리 태깅된 결과의 일관성 유지에 그 초점을 맞출 것이다. 또한, 이렇게 보강된 기능을 지닌 구문 트리 태깅 툴을 이용하여 현재 10만 문장 수준의 구문 트리 뱅크를 점차적으로 확장 시켜 나갈 것이다.

참고 문헌

[김재훈 & 서정연 1994] 김재훈, 서정연, 자연언어 처리를 위한 한국어 품사 태그, *CAIR-TR-94-55*, 1994.

[조규빈 1993] 조규빈,하이라이트 고교문법 자습서, *지학서*, 1993.

[김기혁 1995] 김기혁,국어 문법 연구, *도서출판 박이정*, 1995.

[이주행 1993] 이주행,현대국어문법론, *대한교과서주식회사*, 1993.

[홍사만 1990] 홍사만,국어특수조사론, *학문사*, 1990.

[서정수 1983] 서정수,국어구문론연구, *탑출판사*, 1983.

[남기심 & 고영근 1995] 남기심, 고영근, 표준 국어문법론, *탑출판사*, 1995.

[남기심 1993] 남기심, 국어 조사의 용법, *서광학술자료사*, 1993.

[안동언 1995] 안동언, Corpus를 기반으로 하는 한국어 술어의 양상 생성,한국과학기술원 박사학위논문, 1995.

[장석진 1992] 장 석진, "한국어 문법 -- NLP를 위한 HPSG/K," *한국과학기술원 인공지능연구센터 기술보고서*, 1992.

[조혁규, 장명길 & 권혁철 1989] 조 혁규, 장 명길, 권 혁철, "KPSG에 기반한 한국어 해석기의 구현," *정보과학회 학술발표회 논문집 89 년 봄*, 1989.

[윤덕호, 김영택 1989] 윤 덕호, 김 영택, "미지문법관계 속성을 이용한 LFG에서의 한국어 문장 분석," *정보과학회논문지 89 년 9 월*, 1989.

[홍영국, 이종혁, 이근배 1993] 홍 영국, 이 종혁, 이 근배, "의존 문법에 기반을 둔 한국어 구문 분석기," *정보과학회 학술발표회 논문집 93 년 봄*, 1993.

[윤덕호, 김영택, 1992] 윤 덕호, 김 영택, "다단계 여과 및 탐색을 이용한 의존 문법에 기반을 둔 한국어 분석 알고리즘," *정보과학회 논문지 92 년 11 월*, 1992.

[권혁철, 최준영 1992] 권 혁철, 최 준영, "단일화 기반 의존 문법을 이용한 한국어 분석기," *정보과학회 논문지 92 년 9 월*, 1992.

[나동렬 1994] 나 동렬, "한국어 파싱에 대한 고찰," *정보과학회지 94 년 9 월*, 1994.

[양승원 et al 1993] 양 승원, 황 이규, 이 기오, 이 용석, "조건 단일화 기반 PART II 문법을 이용한 우리말 분석," *제 5 회 한글 및 한국어 정보처리 학술대회 논문집*, pp.3--14, 1993.

[나동렬 1992] 나 동렬, "패턴-액션 규칙을 이용한 한국어 구문 분석," *제 4 회 한글 및 한국어 정보처리 학술발표 논문집*, pp.131--140, 1992.

[Allen 1995] James Allen, Natural Language Understanding, *The Benjamin/Cummings Pub.*, 1995.

[Marcus et al 1993] Mitchell P. Marcus, Beatrice Santorini and Mary Ann Marcinkiewicz, "Building a large annotated corpus of English : the Penn Treebank," *Computational Linguistics*, Vol.~19, No.~2, pp.~313--330, 1993.

[Beatrice 1991] Beatrice Santorini, "Bracketing Guidelines for the Penn Treebank Project," *internal memo*.

[Collins 1996] Michael John Collins, "A New Statistical Parser Based on Bigram Lexical Dependencies," *Proceedings of Association of Computational Linguistics*, 1996.

[Katz 1987] Slava M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* Vol. ASSP-35, No. 3. 1987.

5. 한국어/영어 정렬 시스템

한국과학기술원
최기선

여백

5. 한국어/영어 정렬 시스템

1장. 서론

기계 번역 단계를 크게 분석, 변환, 생성의 세 단계로 구분할 때 분석과 생성은 각각의 언어에 대하여 독립적인 사전이나 문법규칙 등을 이용하여 수행되어 왔다. 그러나 변환의 경우 번역 대상이 되는 언어들에 대하여 대역 사전이나 변환규칙 등의 호환성(compatible) 있는 언어정보가 요구된다. 호환성있는 언어 정보 추출은 대상 언어들의 요소(단어, 구, 문법규칙 등)를 동시에 수용해야 하므로 하나의 언어를 대상으로 한 언어 정보 구축에 비해 더 어려운 작업이다. 최근 들어 호환성 있는 언어 정보 추출의 방법으로 같은 내용에 대해 두 가지 언어로 구축된 병렬 코퍼스(parallel corpus)를 분석하는 방법이 시도되고 있다.

정렬[1] (alignment)은 병렬 코퍼스 분석의 한 방법으로 병렬 코퍼스 내에서 서로 대응하는 요소를 찾는 문제이다. 크게 문서 단위로 구축된 병렬 코퍼스에 대한 대응 문장 정렬과 문장 단위로 구축된 병렬 코퍼스에 대한 대응 단어 정렬로 나눌 수 있다. 정렬된 병렬 코퍼스는 확률적 기계번역을 비롯하여 대역사전 자동구축, 의미 모호성 해소, 대역구 추출 등에 응용되고 있다. 최근에 이용 가능한 병렬 코퍼스가 증가함에 따라 병렬 코퍼스의 이용에 대한 중요성이 부각되고 있다.

문장 단위의 정렬의 경우 문장의 길이나 문서내 단어의 위치와 같은 일반적인 정보만을 이용하여 95% 이상의 높은 정확도로 정렬이 가능했다. 이에 반해 단어 단위의 정렬은 기본 단어 단위나 단어 순서, 문장 구조 등 대상 언어의 특성에 따라 난이도가 크게 달라진다. 인도-유럽어에 속하는 영어와 불어 간의 정렬에 대해 단어 간의 문장 내의 공기 정보(collocation)와 위치 정보만을 이용하여 정렬이 시도되었다. 영어와 불어는 비슷한 문장구조를 가지고 단어의 단위나 순서 등에서 서로 비슷한 언어적 특성을 공유하므로 단어 간의 공기정보나 문장 내 위치 정보가 큰 역할을 할 수 있었다. 그러나 서로 다른 어족에 속하는 한국어와 영어를 정렬할 경우 기본 단어 단위의 상이성과 한국어와 영어 간의 단어 순서가 주는 정보가 미약하다는 특징이 있다. 더우기 기본 단위의 상이성으로 인하여 여러

---

[1] 본 보고서에서 정렬(alignment)은 문서 정렬(text alignment)를 일컫는다.

단어와 여러 단어가 대응되는 경우도 영어와 불어 병렬 코퍼스에 비해 한국어와 영어 병렬 코퍼스에서 더 빈번히 발생한다.

본 보고서는 한국어와 영어 병렬 코퍼스를 정렬함에 있어 문제가 되는 기본 단위의 상이성을 극복하고 한국어와 영어 간의 단어 순서 정보 이외의 정렬에 유용한 정보를 반영하는 모델을 제안한다. 이 모델은 파라미터들(단어 대역확률, 구 대응 확률, 위치/기능어 확률)의 학습을 위하여 EM(expectation-maximization) 방법을 이용하게 된다. 이 방법을 사용한 것은 기본 단위의 상이성을 극복하기 위하서인데, 이렇게 됨으로써 단어 단위의 정렬모델을 구단위 정렬 모델로 확장하였다. 구단위 정렬 모델로의 확장을 통해 기본 단위의 상이성 극복과 구단위의 정보를 추출이 가능하였다. 뿐만 아니라 기존의 단어 단위 정렬 방식에 비해 더 정확한 대역 단어 정보를 추출하였다. 그리고 한국어와 영어 사이의 단어 위치 정보 대신에 한국어의 기능어와 영어의 단어 위치 정보를 이용하므로써 정렬 정확도 향상에 기여하였다.

2장. 본론

1절. 정렬 문제 정의

두 가지 언어로 구축된 코퍼스를 양국어 코퍼스(bilingual corpus)라고 부른다. 양국어 코퍼스 중 같은 내용을 대상으로 하여 두 가지 언어로 구축된 코퍼스를 병렬 코퍼스(parallel corpus)라고 하고 흔히 양국어 코퍼스라는 말로 병렬 코퍼스를 나타내기도 한다.

본 보고서에서는 정렬을 이러한 병렬 코퍼스 내에서 서로 대응하는 요소를 매칭시키는 과정으로 정의한다. 예를 들어 문장 단위 정렬의 경우 문서 단위로 구축된 병렬 코퍼스에서 대응 문장을 찾는 문제이고 단어 단위 정렬의 경우 문장 (문서) 단위로 구축된 병렬 코퍼스에서 대응 단어를 찾는 문제가 된다. 그림 1 에서 문장 단위 정렬과 그림 2 에서 단어 단위 정렬의 예를 보이고 있다.

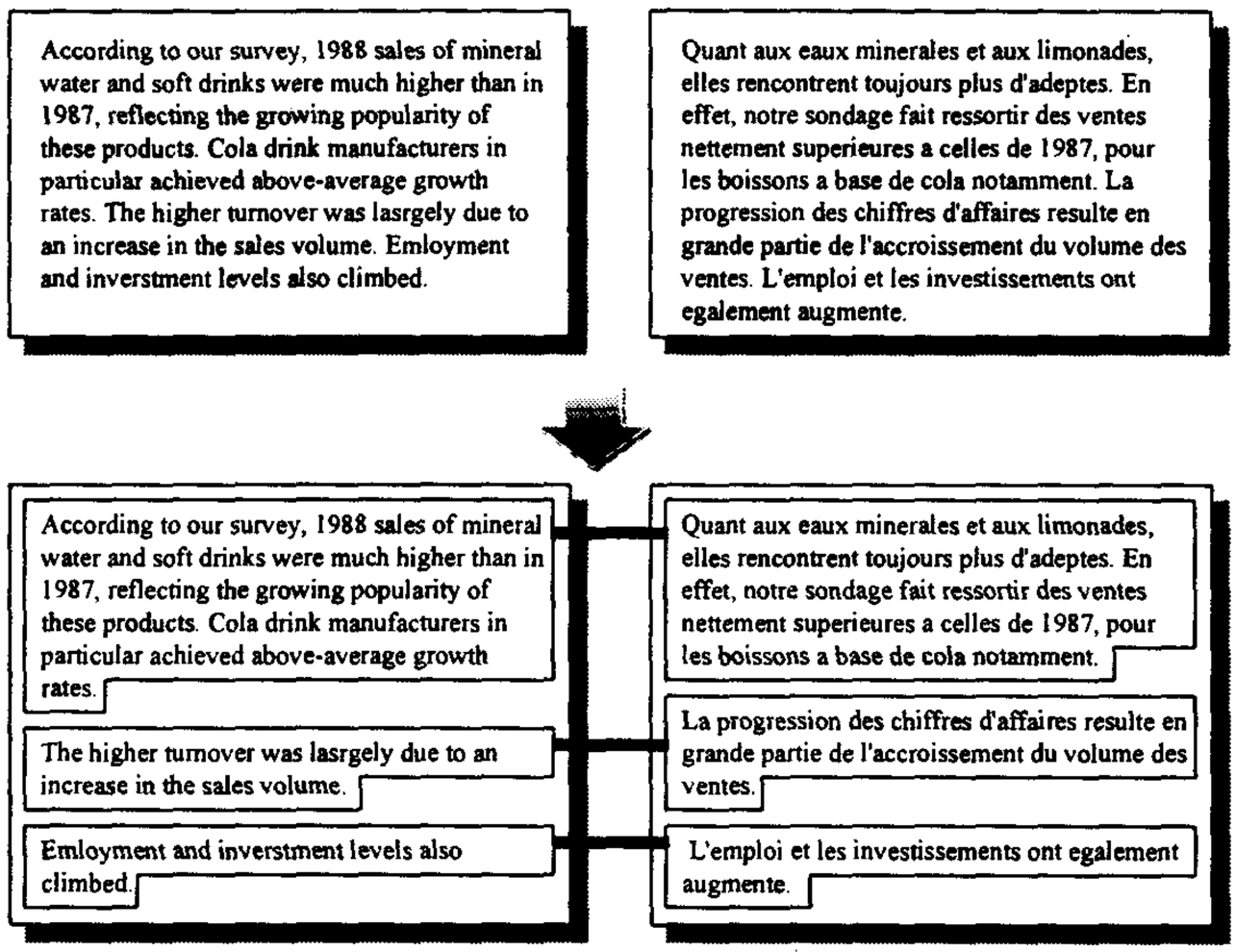


그림 1 : 문장 단위 정렬의 예

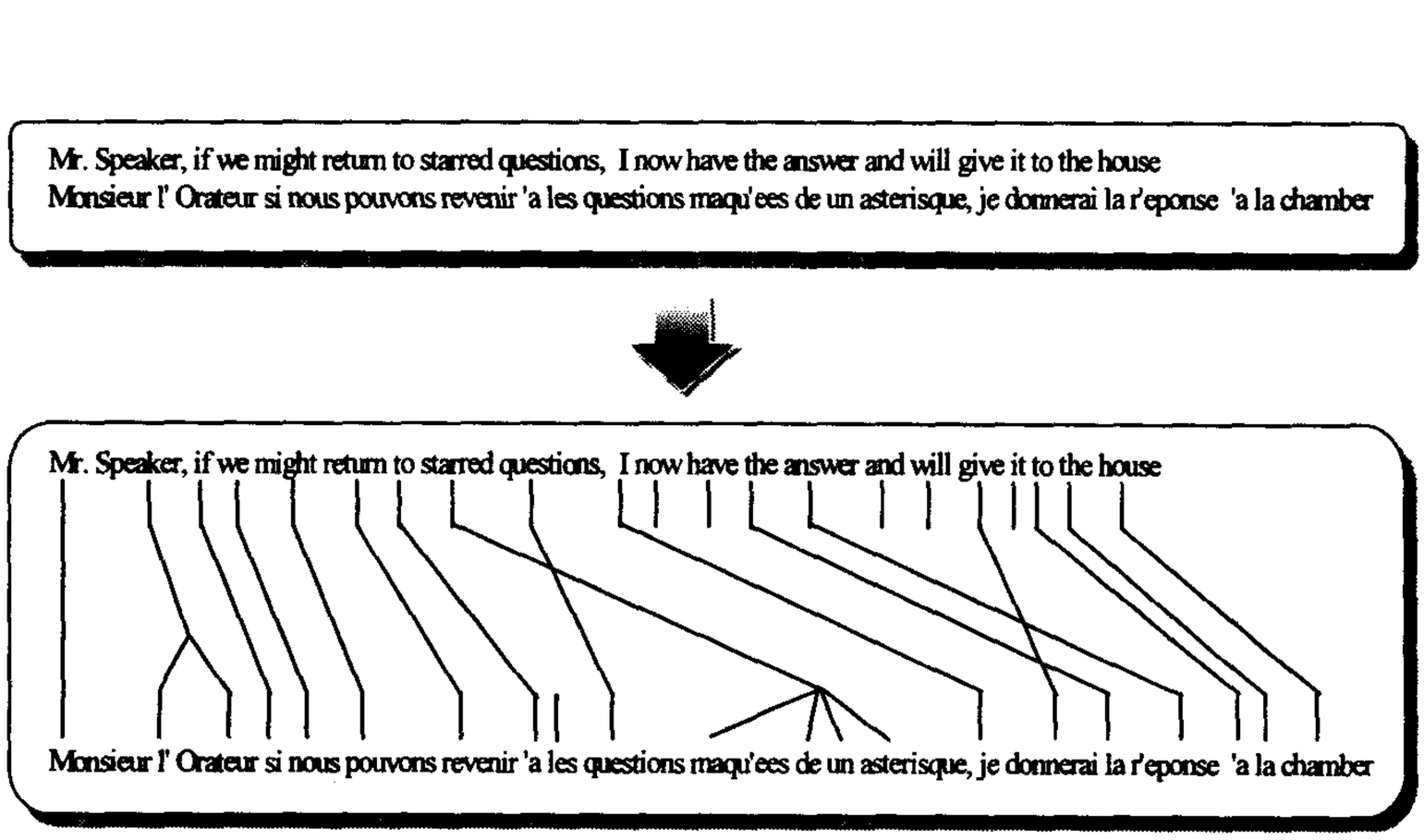


그림 2 : 단어 단위 정렬의 예

1. 정렬 문제의 특징

정렬 문제의 특징에 비추어 정렬 모델을 설계함에 있어 기본적으로 요구되는 사항들을 다음 다섯 가지로 나눌 수 있다.

(1) 견고성 (robust)

정렬은 문제 특성상 정렬에 필요한 정보들을 사전에 완벽하게 구축하기가 힘들다. 단어 단위 정렬에서 기본이 되는 정보인 대역사전의 경우를 예로들면 두 가지 언어에 대한 양국어 사전 구축자체가 큰 작업을 요구할 뿐더러 새로운 용어가 생겨남에 따라 사전 정보를 계속 추가해야 한다. 또한 대상 영역(domain)이 바뀔 때마다 추가 작업이 필요하다. 이러한 어려움에 비춰볼 때 정렬 시스템은 정보가 충분하지 않는 상황에도 견고하게 대처할 수 있어야 한다.

(2) 자동 학습(learning)

앞에서 예를 들었던 대역사전과 같은 방대한 정보를 추출하는데 수작업에 의존한다면 많은 시간과 비용을 들여야만 한다. 그러므로 정렬 모델이 자동으로 이용할 정보를 추출하는 기능을 갖는 것이 바람직하다. 즉 수작업을 최소화할 수 있어야 한다.

(3) 파라미터 학습 (parameter estimation)

대역사전과 같은 정렬에 필요한 정보가 충분히 주어졌다고 가정하더라도 이들 정보간의 중요도를 측정하는 것은 매우 힘든 작업이다. 그러므로 정렬 시스템은 주어진 정보에 대한 중요도(weight) 값을 쉽게 측정할 수 있어야 한다.

(4) 규칙의 명료성

정렬에 이용할 규칙을 정의할 때 예상지 못한 입력이나 매우 긴 문장에 대하여도 적용할 수 있도록 규칙을 정해야 한다. 즉 긴 문장과 복잡한 문장구조에도 적용할 수 있는 국부적인 정보(local information)를 이용하여 정렬을 수행하는 것이 바람직하다.

(5) 모델에 대한 평가

모델에 대한 정확한 평가를 위해서 단순히 정렬 결과의 정확도 측정 외에 모델의 구조적 특성 및 앞으로의 향상 가능성에 대한 평가도 요구된다. 이와 함께 모델의 성능이 이용하는 정보의 질에 의존하므로 모델의 평가를 위해서는 이용하는 정보의 질에 대한 근거제시가 필요하다.

위의 기준에 대하여 정렬 모델링의 방법으로 크게 규칙기반의 방법과 확률기반의 방법에 대하여 비교해 볼 때 규칙기반의 방법은 학습 코퍼스를 필요로 하지 않고 모델의 설계가 쉽다는 장점이 있지만 자동학습이나 파라미터 조정이 불가능

하고 견고성이 약하다. 또한 규칙은 기술하는 사람에 따라 규칙의 정확도가 달라질 수 있으므로 모델 정확도에 대한 정확한 근거 제시가 어렵다. 이에 반해 확률적인 방법은 자동학습이나 파라미터 조정이 가능하고 정보가 빈약한 상황에서도 견고성이 있다. 또한 주어진 학습 코퍼스에 대하여 제안하는 확률모델의 파라미터 학습에 대한 수학적 근거를 제시할 수 있다는 장점이 있다. 따라서 정렬 문제의 경우 규칙 기반의 방법보다 확률기반의 방법이 학습할 코퍼스가 있는 경우 더 적합하다. 실제로 지금까지 연구된 대부분의 방법들이 극단적으로 확률에 기반한 방법을 쓰고 있다.

2 절. 기존의 접근 방법

정렬 시스템을 설계하기에 앞서 정렬의 기본 단위가 먼저 결정되어야 한다. 기본의 단위의 선정은 정렬 대상 언어 간의 특성에 크게 의존하고 정렬 시스템의 성능에도 결정적인 영향을 미친다. 기본 단위를 작게 삼을 수록 대상 언어 간의 기본 단위의 상이성 문제가 심각해 진다. 예를 들어 문자 단위의 정렬의 경우 영어와 불어 같이 같은 알파벳을 공유하고 많은 단어들이 어원이 같은 비슷한 형태를 갖는 언어들 사이에서만 가능하고 한국어와 영어의 정렬과 같은 문자 단위에서의 관련성이 거의 없는 경우는 수행하기 어렵다. 파싱된 트리나 구와 같은 상위 단위를 기본 단위로 할 경우 언어에 관계없이 일반적인 단위를 갖게 되므로 기본 단위의 상이성 문제를 극복하기 쉬어진다. 그러나 상위 단위로 갈수록 정확한 정렬 기본단위의 추출이 힘들어짐으로 정렬 오류에 요인이 될 가능성이 크진다. 예를 들어 파싱된 트리 간의 정렬을 수행할 경우 정확하게 파싱된 트리가 선택되어야 한다는 문제와 파싱된 트리 간의 호환성(compatibility)의 문제를 안고 있다.

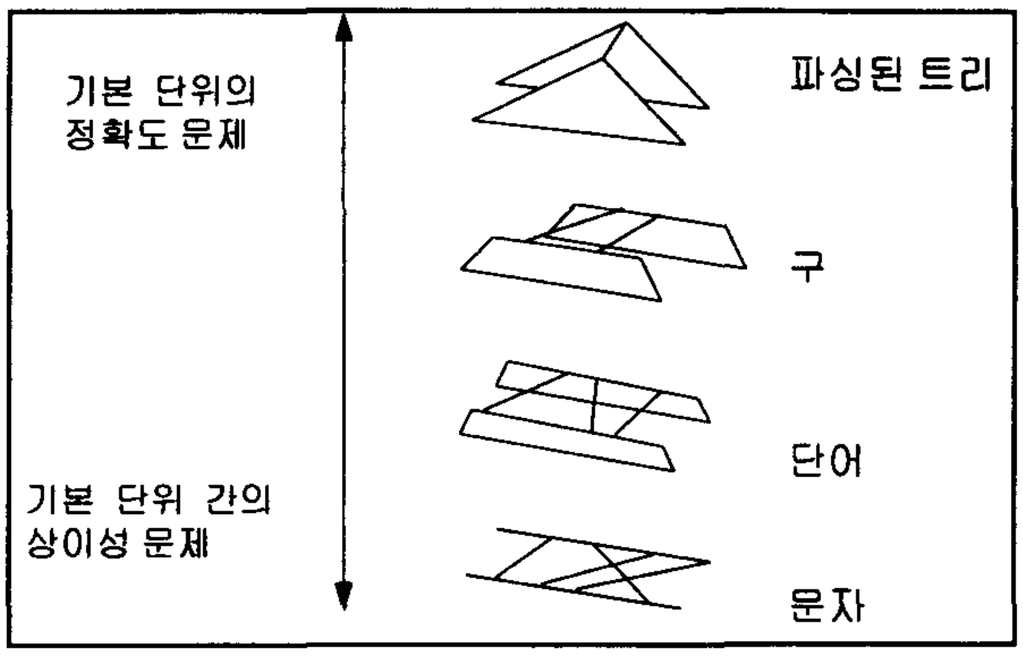


그림 3: 정렬 단위간의 특성 비교

정렬의 단위를 기준와 대상 언어의 특성을 중심으로 지금까지 연구된 정렬 시스템을 정리하면 다음과 같다.

1. 문장 단위 정렬

한 언어의 긴 문장을 번역하면 대응 언어에서도 긴 문장이 될 가능성이 크고 짧은 문장은 짧은 문장이 될 가능성이 크다는 성질을 이용하여 문장단위의 정렬 모델을 제안하였다. 대응 문장의 관련도는 대상 문장의 문서 내 위치와 문장을 구성하는 문자수를 이용하여 측정하였다. 실제 모델에서 문자 수 l_1개로 구성된 문장 s와 문자 수 l_2개로 구성된 문장 t 사이의 거리 차이 함수는 식 2.1과 같이 정의하였다. 식 2.1에서 임의의 언어 L_1이 다른 언어 L_2로 정렬이 될 때 c는 문서 L_1과 L_2 사이의 평균 문자 수 차이을 나타내고 s^2은 이들 문자수 차이의 분산을 나타낸다. 그리고 $Pr(match)$는 문장 당 대응이 각각 1-1, 1-0(0-1) ,2-1(1-2), 2-2 대응으로 될 확률을 나타낸다.

$$d(s,t) = \Pr(match) \times 2\left(1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\delta} e^{-z^2/2} dz\right) \tag{2.1}$$

$$\delta = \frac{|l_2 - l_1 c|}{\sqrt{l_1 s^2}} \tag{2.2}$$

위에서 정의한 거리차이 측정함수를 이용하여 실제 다음과 같은 동적프로그램(dynamic programming)을 이용하여 최대확률값(maximum likelihood)을 갖는 문장간의 정렬을 수행한다. 이때 초기값으로 $D(i, j) = 0$으로 할당되고 정렬 과정은 다음 식 2.2과 같이 재귀적으로 정의하였다.

$$D(i,j) = \min\begin{cases} D(i,j\text{-}1) + d(0,t_j) \\ D(i-1,j) + d(s_i,0) \\ D(i\text{-}1,j\text{-}1) + d(s_i,t_j) \\ D(i\text{-}1,j\text{-}2) + d(s_i,t_{j\text{-}1}t_j) \\ D(i\text{-}2,j\text{-}1) + d(s_{i\text{-}1}s_i,t_j) \\ D(i\text{-}2,j\text{-}2) + d(s_{i\text{-}1}s_i,t_{j\text{-}1}t_j) \end{cases} \qquad (2.3)$$

위의 식 2.3 에서 $d(a,b)$는 식 2.1 에서 정의한 문장 a 와 문장 b 가 대응할 때의 거리차를 나타내고, a 혹은 b 가 0 인 경우는 대응문장이 없은 경우를 나타낸다.

Gale 의 방법은 특정한 언어에 의존하지 않는 단순히 문자 갯수 정보만을 가지고 정렬하였으므로 범용적인 적용이 가능한 방법이다. 실험적으로 인도-유럽어들 같이 서로 비슷한 언어들 사이에서는 96% 이상의 정렬 정확도를 보였다. 그러나 중국어-영어의 경우와 같이 문자 수가 판이하게 다른 경우에도 잘 적용되는지는 검증이 필요하다.

2. 영어/불어 정렬 시스템

Brown 은 영어와 불어에 대하여 대역 단어는 서로 대응되는 문장의 쌍에서 자주 공기한다는 성질과 대역 단어는 서로 대응 문장 내에서의 위치적 연관성이 있음을 이용하여 정렬 모델을 제안하였다. 정렬의 기본 단위로는 크게 영어 한 단어와 불어 한 단어가 대응되는 모델과 영어 한 단어와 불어 여러 단어가 대응되는 경우의 모델로 나누어 구현하였고 모델의 파라미터(대역확률, 위치확률 등)는 EM 알고리즘을 이용하여 학습하였다.

(1) 단어 대 단어 정렬 모델

영어와 불어 단어 간의 쓰임에 영어 문장(단어열) **e** 와 불어 문장(단어열) **f** 사이의 가능한 정렬을 **a** 로 나타낼 때 **e** 가 **f** 에 대응할 확률을 조건부 확률 **P(f,a|e)**를 써서 다음과 같이 나타낼 수 있다.

$$\Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{f},\mathbf{a}|\mathbf{e}) \qquad (2.4)$$

식 (2.4)에 대하여 **e** 가 l 개의 단어 $e_1e_2..e_l$로 구성되어 있고 **f** 가 m 개의 단어 $f_1f_2..f_m$으로 구성되어 있을 때 정렬은 0 과 l 사이의 값을 갖는 m 개의 변수열 $a_1a_2..a_m$으로 정의할 수 있다. 이때 $a_j = i$ 는 j 번째 불어 단어가 i 번째 영어 단어와

정렬됨을 나타낸다. 이것을 수식으로 나타내면 식 (2.5)와 같다.

$$\Pr(\mathbf{f},\mathbf{a}|\mathbf{e}) = \Pr(m|e)\prod_{j=1}^{m}\Pr(a_j|a_1^{j-1},f_1^{j-1},m,\mathbf{e})\Pr(f_j|a_1^{j},f_1^{j-1},m,\mathbf{e}) \qquad (2.5)$$

식 (2.5)에 대하여 정렬이 단어 간의 대역확률(translation probability)에만 의존한다는 가정 하에 식 (2.6)과 같은 정렬 모델을 세웠다. 식 (2.6)에서 c는 문장의 단어 길이에 의하여 결정되는 상수값이다.

$$\Pr(\mathbf{f}|\mathbf{e}) = c\sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}\prod_{j=1}^{m}\Pr(f_j|e_{a_j}) \qquad (2.6)$$

$$= c\prod_{j=1}^{m}\sum_{i=0}^{l}\Pr(f_j|e_i) \qquad (2.7)$$

식 (2.7)는 확률의 곱들에 대한 합으로 표현된 식 (2.6)을 확률의 합들의 곱으로 바꾼 형태이다. 즉 m=3 이고 l=1 인 경우를 $p_{ji} = Pr(f_j|e_i)$로 간략하게 써서 예를 들면 식 (2.6)은 $p_{10}p_{20}p_{30} + p_{10}p_{20}p_{31} + \ldots + p_{11}p_{21}p_{30} + p_{11}p_{21}p_{31}$ 형태로 나타낸 것이고 식 (2.7)의 경우 $(p_{10}+p_{11})(p_{20}+p_{21})(p_{30}+p_{31})$로 나타낸 것이다. 예에서도 쉽게 알 수 있듯이 식 (2.6)는 식(2.7)에 비해 훨씬 적은 계산을 필요로 한다.

대역확률의 학습은 EM 알고리즘을 이용하여 다음 식 (2.8)-(2.9)와 같이 수행하였다. 즉 임의의 영어 단어 e가 불어 단어 f로 대역될 확률은 서로 대응되는 문장에서 공기되는 정도에 의해 결정된다. 식 (2.8)에서 λ_e는 확률의 합을 1로 만들기 위한 상수이고, $\delta(a,b)$는 a와 b가 같을 때 1, 그 외는 0의 값을 가지는 함수이다.

$$\Pr(f|e) = \lambda_e^{-1}\sum_{\mathbf{e},\mathbf{f}\in Corpus} c(f|e;\mathbf{f},\mathbf{e}) \qquad (2.8)$$

$$c(f|e;\mathbf{f},\mathbf{e}) = \frac{\Pr(f|e)}{\Pr(f|e_0)+\cdots+\Pr(f|e_l)}\sum_{j=1}^{m}\delta(f,f_j)\sum_{i=0}^{l}\delta(e,e_i) \qquad (2.9)$$

위의 식 (2.7)에서 정의한 모델 경우 정렬을 단어 간의 대역확률로만 설명하였으므로 대역확률에 대한 정보가 미약한 경우, 즉 학습 코퍼스에서 공기 빈도가 작은 단어들로 구성된 문장의 경우에는 정확한 정렬이 이루어 지지 못한다. 이에 대한 해결책으로 영어와 불어가 문장 구조가 비슷하다는 점을 반영하였다. 즉 서로 대응하는 영어와 불어 단어는 문장 내에서 비슷한 위치에서 쓰여진다는 사실

을 반영하여 식 (2.10)과 같이 모델을 확장하였다.

$$\Pr(\mathbf{f}|\mathbf{e}) = c\prod_{j=1}^{m}\sum_{i=0}^{l}\Pr(f_j|e_i)\Pr(i|j,m,l) \qquad (2.10)$$

(2) 영어 한 단어가 불어 여러 단어가 대응하는 경우

식 (2.7)과 (2.10)은 영어 한 단어가 불어 한 단어로 대응되는 과정을 모델링한 것으로 한 단어가 여러 단어로 대응되는 현상을 반영하지 못 하고 있다. 예를 들어 영어 단어 "*not*"가 불어 에서는 "*ne ... pas*"로 대응되는 것을 식 (2.7)과 (2.10)에서는 *ne, pas* 에 대해 상호 연관성 없이 독립적으로 취급되었다. 이에 대한 해결책으로 정렬과정을 영어 단어 하나에 대해 불어 단어 하나가 대응하는 관점에서 영어 한 단어가 불어 여러 단어와 대응하는 모델로 다음과 같이 확장하였다.

우선 영어 단어에 대해 이 단어가 불어 단어 몇 개와 대응할 지를 결정하고 대응 가능한 불어 단어들의 리스트를 결정한다. 그리고 대응 가능한 불어 단어 후보들을 조합하여 대역할 불어 문장을 생성한다. 영어 단어 e_i와 대응 가능한 불어 단어의 갯수를 ϕ_i로 e_i와 대응 가능한 불어 단어 리스트를 τ_i라고 할 때 k번째 단어를 τ_{ik}로 나타낸다. 또한 τ_{ik}의 불어 문장 내의 위치를 π_{ik}로 나타낸다. 그리고 이 변수들의 열을 각각 $\boldsymbol{\phi} = \phi_1\phi_2\cdots\phi_l$, $\boldsymbol{\tau} = \tau_1\tau_2\cdots\tau_l$, $\boldsymbol{\pi} = \pi_1\pi_2\cdots\pi_l$로 정의할 때 정렬 과정을 다음과 같이 나타내었다.

$$\Pr(\tau,\pi|\mathbf{e}) = \Pr(\phi|\mathbf{e})\Pr(\tau|\phi,\mathbf{e})\Pr(\pi|\tau,\phi,\mathbf{e}) \qquad (2.11)$$

$$\Pr(\mathbf{f},\mathbf{a}|\mathbf{e}) = \sum_{(\tau,\pi)\in<\mathbf{f},\mathbf{a}>}\Pr(\tau,\pi|\mathbf{e}) \qquad (2.12)$$

식 (2.12)에서 (τ,π)가 <**f**,**a**>에 속하기 위해서는 모든 i=0,1,...,l 과 k=1,2,...,ϕ_i에 대하여 $f_{\pi_{ik}} = \tau_{ik}$와 $a_{\pi ik} = i$를 만족하여야 한다.

일반적인 정렬 과정을 나타낸 식 (2.11)에 대하여 식 (2.13)-(2.15)와 같은 제약을 두어 모델을 구축하였다. 식 (2.13)에서 p_ε는 영어 단어 e가 어떠한 불어 단어와도 대응하지 않을 확률을 나타낸다.

$$\Pr(\phi|\mathbf{e}) = \prod_{i=1}^{l} \Pr(\phi_i|e_i) \binom{\phi_1+\cdots+\phi_l}{\phi_0} p_\varepsilon^{\phi_0}(1-p_\varepsilon)^{\phi_1+\cdots+\phi_l} \quad (2.13)$$

$$\Pr(\tau|\phi,\mathbf{e}) = \prod_{i=0}^{l}\prod_{k=1}^{\phi_i} \Pr(\tau_{ik}|e_i) \quad (2.14)$$

$$\Pr(\pi|\tau,\phi,\boldsymbol{\theta}) = \frac{1}{\phi_0!}\prod_{i=0}^{l}\prod_{k=1}^{\phi_i} P_{ik}(\tau_{ik}) \quad (2.15)$$

식 (2.15)에서 P_{ik}는 불어 단어 τ_{ik}에 대한 문장내의 위치 확률을 나타낸다. 불어와 영어는 서로 비슷한 문장 구조를 가지기 때문에 영어에서 인접한 단어에 대응하는 불어의 단어들 사이에도 역시 위치적 연관성이 있다. 이러한 특성을 반영하여 영어 한 단어와 대응하는 여러 불어 단어들을 문장 내에서 맨 처음에 위치하는 단어와 나머지 단어들로 나누어 모델링 하였다. 먼저 e_i와 대응하는 여러 불어 단어 중 그 첫번째 단어에 대하여 식 (2.16)과 같이 위치 확률을 정의하였다.

$$\text{k=1 인 경우 : } P_{ik}(j) \equiv \Pr(j - \Theta_{[i-1]}|t(e_{[i-1]}), t(f_j)) \quad (2.16)$$

불어 단어와 대응하지 않는 영어 단어들을 고려 대상에서 제외시키기 위하여 하나 이상의 불어 단어와 대응하는 i번째 영어 단어에 대하여 $e_{[i]}$로 나타내었다. 식 (2.16)에서 $t(w)$는 단어 w의 품사를 나타내고 $\Theta_{[i-1]}$는 $e_{[i-1]}$와 대응하는 불어 단어들의 평균 위치를 나타낸다.

영어 한 단어에 대하여 불어 여러 단어가 대응할 경우 대응되는 불어 여러 단어 사이에는 위치적 연관성이 있다. 이러한 불어 단어 간의 위치적 연관성을 식 (2.17)과 같이 반영하였다.

$$\text{k > 1인 경우: } P_{ik}(j) \equiv \Pr(j - \pi_{[i]k-1}|t(f_j)) \quad (2.17)$$

지금까지 살펴보았듯이 Brown 의 영어/불어 정렬 모델은 단어 간의 대역확률과 문장 내의 위치정보를 이용하여 정렬을 수행하였다. 문장 내 위치 정보는 특정한 단어에 의존하지 않고 일반적으로 그리고 쉽게 추출할 수 있는 정보로 영어와 불어와 같은 상호 간의 단어순서가 연관성이 큰 언어 사이에서는 효율적인 방법이었다. 그러나 각각의 단어에 대하여 독립적인 요소로 취급함으로써 여러 단어와 여러 단어가 대응하는 관계를 모델링하기 위해서는 지수적 계산 복잡도가 불가피하였다. 이러한 제약은 영어/불어 정렬과 같이 언어 간의 기본 단어 단위가 비슷

한 경우에는 큰 문제가 되지 않지만 기본 단어 단위가 일치하지 않는 경우, 즉 여러 단어와 여러 단어 간의 대응이 자주 일어나는 경우 큰 문제가 된다. 또한 단어 간의 위치 정보 혹은 순서 정보를 이용하는 경우 한국어와 영어 같이 상호 간의 단어의 쓰임에 있어 규칙성이 미약한 경우 직접적인 적용이 어렵다.

3. 영어/중국어 정렬 시스템

중국어의 경우 공백(space)을 이용한 단어 구분없이 모든 단어를 붙여서 문장을 구성한다. 그러므로 영어/중국어 정렬을 위해서는 우선 중국어 문장에서 적절한 단어를 분리하는 과정(Segmentation)이 선행되어야 한다. 한 문장이 여러 가능한 단어열로 분리될 수 있으므로 정확한 단어 분리가 힘들 뿐더러 분리된 중국어 단어가 영어 단어와의 정확한 대응관계를 찾기도 힘들다. 예를 들어 '管理局'이 'Authority'와 대응되기 위하여 하나의 단어로 분리되어야 함에 반해 '財政司'의 경우 각각 'Finance Secretary'로 대응되기 위해 '財政'과 '司'로 나뉘어 구분되어야 한다.

管理局將會責向**財政司**

The Authority is accountable to the Finance Secretary

이러한 기본 단위 상이성 문제를 영어/불어 정렬과 구별되는 가장 큰 특징으로 들 수 있다. 이에 대한 대표적인 연구로 다음 두가지 연구를 들 수 있다.

(1) Wu의 모델

Wu는 영어/중국어 간의 기본 단위 상이성 문제를 해결하기 위하여 영/중 대역 사전을 이용한 중국어 단어 분리(Segmentation) 과정을 정렬 전처리 과정으로 두었다. 대역 사전을 통해 영어 단어와 비슷한 단위로 중국어 문장을 분리한 후 식 (2.7)과 같이 단어 간의 대역확률을 이용하여 정렬을 하였다. 그러나 대역 사전을 정렬 이전에 수동으로 구축해야 한다는 부담이 있고 대역 사전에 없는 단어의 경우를 고려하지 않았다는 문제점이 있다.

(2) Fung의 모델

영어/중국어와 같이 서로 상이한 언어를 정렬함에 있어 모든 문장성분을 정확히 정렬하기는 힘들다. 더우기 구축된 병렬 코퍼스의 양이나 질도 비슷한 언어 간의 병렬 코퍼스에 비해 좋은 상황을 기대하기 힘들다. 이러한 현실적 조건을 고려하

여 Fung 은 정렬대상의 문장성분(category)을 언어에 관계없이 비교적 일치하는 고유명사나 명사로 국한시켰다. 품사태거에 의하여 추출된 고유명사와 명사에 대해 문서 내의 위치 정보를 반영하여 DTW(dynamic time warping) 알고리즘를 이용하여 정렬하였다.

4. 기타 정렬 시스템 및 정렬 결과 응용에 관한 연구

병렬 코퍼스에 대한 정렬이나 응용에 관한 연구는 다음 표 1 과 같이 정리할 수 있다. 앞 절에서 설명한 문장단위의 정렬, 문자 단위의 정렬, 단어 단위의 정렬 외에 정렬의 기본 단위에 따라 명사구 단위의 정렬, 대역 문법 규칙에 의한 정렬, 파싱된 트리끼리의 정렬로 나눌 수 있다. 명사구 단위 정렬의 경우, 정렬대상을 명사구로 한정하고 정렬을 수행하기 이전에 태깅과 명사구 추출 단계를 전처리 단계로 두었다. 이외에 파싱된 트리 간의 정렬에서 생기는 파싱된 트리가 선택 문제와 파싱된 트리 간의 호환성(compatibility)의 문제를 해결하기 위해 호환성 있는 대역 문법 규칙을 정렬 과정에서 학습해 나가는 연구가 시도되었다. 정렬된 코퍼스의 응용에 관한 연구로는 모호성을 갖는 단어에 대하여 정렬된 코퍼스 상의 대응 단어를 이용하여 모호성을 해결하는 방법이 연구되었다. 또한 정렬된 파싱된 트리에서 구단위 번역 예를 추출하는 연구와 코퍼스를 이용한 기계번역이 시도되었다.

정렬 수행 단위	정렬 결과를 이용한 응용
문장(GC91A,BLM91)	
문자(Chu93)	
단어(BPPM93, WX94,Fun95)	의미 모호성 해소(BPPM91)
명사구(Kup93)	
대역 문법 규칙(Wu 95a)	구 단위 번역 예 추출(Wu95b)
파싱된 트리(KYY92,MHT93)	기계 번역(CC95,BPPM93)

표 1: 병렬 코퍼스 정렬과 응용에 관한 관련 연구

3 절. 한국어/영어 정렬 시스템

이 절에서는 본 보고서에서 제시하는 한국어/영어 정렬 모델과 모델을 어떻게 구

현할 것인가에 대해서 설명해 보기로 한다. 한국어와 영어사이의 특징에 대해서 논의하고, 그 특징에 알맞은 모델을 제시한 후 어떤 방법을 쓸 것인가에 대해서 논의하게 된다.

1. 한국어/영어 정렬의 특징

한국어와 영어를 정렬함에 있어 같은 어족에 속하는 영어와 불어의 정렬과 구별되는 대표적인 특징으로 기본단위의 상이성와 한국어와 영어 간의 단어 순서가 주는 정보가 미약함을 들 수 있다.

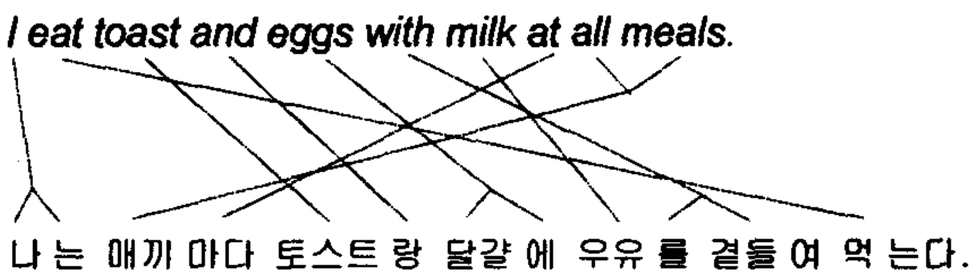


그림 4 : 한국어/영어 정렬 예1(형태소 대 단어)

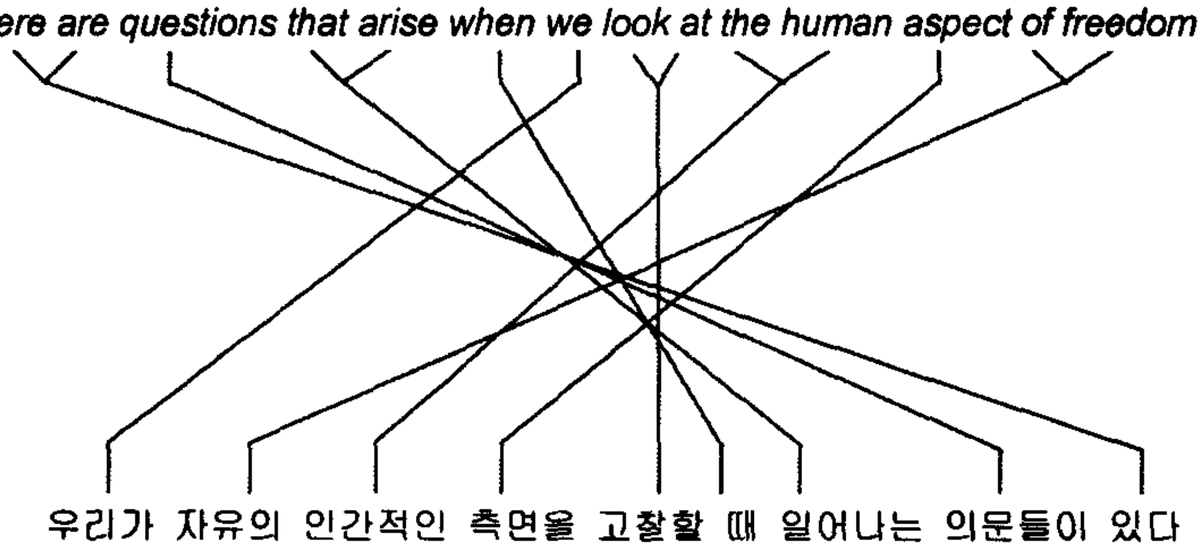


그림 5 : 한국어/영어 정렬 예2(어절 대 단어)

영어와 불어는 서로 비슷한 단어 단위를 가지므로 단어 간의 대응을 한 단어끼리의 대응과 단어와 여러 단어와의 대응으로 제약하고 정렬 모델을 구축하였다. 그러나, 한국어와 영어의 정렬의 경우, 문장을 구성하는 기본 구성 요소의 상이성이 두드러진다. 즉 영어의 경우 단어들이 문장을 구성하는데 반해 한국어의 경우 형태소들의 조합으로 이루어진 어절로 문장이 구성된다. 일반적으로 한국어 형태소는 영어 단어에 비해 작은 단위이고, 어절의 경우 다소 큰 단위가 되어 형태소와 영어 단어 혹은 어절과 영어 단어 간의 정확한 대응이 이루어 지지 않는다. 표 2는 한국어와 영어 간의 기본 구성요소 간의 상이성 정도를 개략적으로

파악하기 위해 대응되는 한국어와 영어 30문장에 대한 수동 분석한 결과이다. 표 2에서 볼 수 있듯이 한 단어가 한 형태소와 대응하는 경우는 전체의 32.1%에 지나지 않았다. 즉, 한국어와 영어 정렬에 있어서 여러 단어와 여러 단어가 대응되는 현상이 두드러지게 나타난다.

한국어와 영어는 서로 다른 문장 구조를 가지고 언어적 문화적 차이로 인하여 표현방식도 같은 어족이나 같은 문화권에 속하는 언어들에 비해 크게 다르다. 그러므로 한국어 문장을 구성하는 단어들의 순서와 영어 문장을 구성하는 단어들의 순서 간의 규칙성을 추출해 내기가 힘들다.

한국어형태소 / 영어단어	0	1	2	3	4이상
0		2.2%			
1	4.2%	32.1%	28.8%	9.7%	1.5%
2	4.3%	7.3%	3.6%	1.9%	
3이상	1.1%	1.3%	0.6%	0.4%	

표 2 한국어 형태소와 영어 단어 간의 대응 수 비교

살펴본 결과와 같이, 한국어와 영어의 정렬을 모델링함에 있어서 여러 단어들끼리의 대응의 수용이 필요하고 또 정렬의 정확도를 높이기 위하여 단어 순서 정보 외의 정렬에 유용한 정보의 반영이 필요하다.

2. 한/영 정렬 모델

한국어와 영어 사이에도 대역확률이 큰 단어 간에 정렬될 가능성이 크다. 이러한 사실을 이용하여 식 (2.7)에서 제안된 대역확률을 이용한 정렬 모델을 한국어와 영어의 기본 정렬 모델로 삼았다. 대역확률의 학습도 식 (2.8)-(2.9)에서 이용한 대응 문장 내 공기 정보에 기반한 방법을 이용하였다

(1) 모델 1(대역확률만 이용한 모델)

$$\Pr(e|k) = c\prod_{j=1}^{n}\sum_{i=0}^{m}\Pr(e_j|k_i) \qquad (3.1)$$

식 (3.1)에서 m은 한국어 문장 **k**를 구성하는 단어 수를 n은 영어 문장 **e**를 구성하는 단어수를 나타낸다. 단어 간의 대역확률 만을 이용하여 정렬을 할 경우, 학습 코퍼스에서 공기 빈도가 작은 단어들로 구성된 문장의 경우 정확한 정렬을 할 수 없었다. 이에 대한 추가정보로 영어/불어 정렬모델에서는 문장 내의 위치 정보를 반영하였다. 그러나 한국어와 영어와 같이 문장구조가 크게 다른 언어 사이에서는 대역 단어 간의 문장 내 위치 정보는 변별력이 미약하다.

(2) 모델 2 (위치/기능어 정보 반영)

한국어의 어절은 크게 실질 형태소와 형식 형태소로 나누어 지고, 실질 형태소는 어절의 의미를, 형식 형태소는 어절의 문법적 역할을 나타낸다. 한국어는 기능어[2]가 발달되어 있으므로 단어의 쓰임에 있어, 부분적이나마 자유로운 어순(partially free order)이 가능하다. 특히 중심어 후위의 특징을 가지므로 여러 어절이 하나의 구를 이룰 경우 마지막 어절의 형식 형태소가 구를 문법적 특성을 대표한다. 예를 들어 '빠른 새가'가 하나의 구를 이룰 경우 마지막 어절의 기능어인 '가'가, '빨리 달린다' 의 경우는 'ㄴ다'가 구의 문법적 성격을 대표하는 기능어가 된다. 이러한 한국어의 기능어는 단순히 어절을 형태소 분석함으로써 얻을 수 있는 정보이다.

이에 반해 영어의 경우 전치사와 같은 기능어 뿐만아니라 문장 내 단어의 위치가 문법적 특성을 반영한다. 즉 '주어+동사+목적어+부사' 와 같이 문장을 구성하는 단어 순서가 엄격히 지켜진다. 본 보고서에서는 문법적 성격을 나타내는 쉽게 얻을 수 있는 정보인 한국어의 기능어 정보와 영어의 문장 내 위치 정보를 식 (3.2)와 같이 반영하였다.

$$\Pr(\mathbf{e}|\mathbf{k}) = c\prod_{j=1}^{n}\sum_{i=0}^{m}\Pr(e_j|k_i)\Pr(f(k_i),i|j,m,n) \qquad (3.2)$$

---

[2] 본 보고서에서는 형식 형태소를 간단히 기능어로 칭한다.

식 (3.2)에서 $f(k_i)$는 형태소 k_i가 속하는 어절의 대표 기능어를 나타낸다.

(3) 모델 3(구단위 모델 확장)

모델 2에서 반영한 위치/기능어 정보는 모델 1에 비해 좀 더 변별력 있는 정렬에 기여할 수 있지만, 한국어/영어 정렬에서 반드시 고려해야 할 여러 단어와 여러 단어가 대응하는 경우에 대한 해결책이 될 수 없다. 예를 들어 'with age'가 '세월이 흘러감에 따라'와 같이 여러 단어들끼리 대응되는 현상을 반영하지 못 하였다. 모델 3에서는 이러한 제약에 대한 해결책으로 정렬을 단어 단위의 정렬에서 식 (3.3)과 같은 구단위 정렬로 확장시켰다. 예를 들어 P(toast and eggs | 토스트 랑 달걀)의 경우 P(명사 등위접속사 명사|명사 접속조사 명사)×P(toast and eggs, 토스트 랑 달걀)사이의 대역확률로 구대응 확률을 정의하였다.

$$\Pr(\mathbf{e}|\mathbf{k}) = \sum_{<p_k,p_e>\in s(\mathbf{k},\mathbf{e})} \Pr(\mathbf{e}^{p_e}|\mathbf{k}^{p_k}) \tag{3.3}$$

$$= \sum_{<p_k,p_e>\in s(\mathbf{k},\mathbf{e})} \sum_{\mathbf{a}(p_k,p_e)} \Pr(\mathbf{e}^{p_e}, \mathbf{a}(p_k,p_e)|\mathbf{k}^{p_k}) \tag{3.4}$$

식 (3.3)에서 **s(k,e)**는 각각 **k**와 **e**를 구성하는 같은 수의 구들로 구성된 모든 가능한 구의 열의 집합이다. 그리고 $<p_k,p_e>$는 **s(k,e)**에 포함되는 임의의 구의 열이고, $\mathbf{a}(p_k,p_e)$는 이들 사이의 가능한 정렬을 나타낸다. 정렬 관점에서 구를 보통 '명사구'나 '전치사구' 처럼 통사적 제약을 가지지 않고, 단순히 가능한 모든 단어열을 구로 정의한다.

구대응을 단어열과 단어열의 대응이라고 볼 수 있는데 각각의 단어열에 대한 정보를 모두 반영할 경우 자료희귀문제가 심각해 진다. 그러므로 식 (3.5)와 같이 구를 이루는 단어들의 품사열을 이용하여 구대응 정보를 반영하였다. 그리고 구를 구성하는 각각의 단어들끼리의 대역확률을 식 (3.7)와 같이 반영하였다. 그리고 식 (2.6)에서 식 (2.7)로 바꾸는 과정과 같은 방식으로 계산이 효율적인 식 (3.6)으로 전개하였다.

$$\Pr(\mathbf{e}^{p_e}|\mathbf{k}^{p_k}) = \sum_{a_0=0}^{\#(p_{ke})} \cdots \sum_{a_{\#(p_e)}}^{\#(p_{ke})} \prod_{i=1}^{\#(p_e)} \Pr(t(\mathbf{e}_i^{p_e})|t(\mathbf{k}_{a_i}^{p_k}))\Pr(\mathbf{e}_i^{p_e}|\mathbf{k}_{a_i}^{p_k})\Pr(i|f(\mathbf{k}_{a_i}^{p_k}),m,l) \tag{3.5}$$

$$= \prod_{i=1}^{\#(p_e)} \sum_{j}^{\#(p_k)} \Pr(t(\mathbf{e}_i^{p_e})|t(\mathbf{k}_j^{p_k}))\Pr(\mathbf{e}_i^{p_e}|\mathbf{k}_j^{p_k})\Pr(i|f(\mathbf{k}_j^{p_k}),m,l) \tag{3.6}$$

식 (3.5)에서 $\mathbf{k}^{p_k}$는 문장 $\mathbf{k}$를 구성하는 임의의 구의 열을 나타내고 $\mathbf{k}_j^{p_k}$는 이 중 j번째 구를 나타낸다. $t(P)$ 구 P를 구성하는 단어들의 품사열을 구하는 함수이고 $\#(P_e)$는 구의 열 P_e를 구성하는 구의 갯수를 나타낸다.

$$\Pr(\mathbf{e}_i^{p_e}|\mathbf{k}_j^{p_k}) = \prod_{k=1}^{\#(\mathbf{e}_i^{p_e})} \sum_{l=1}^{\#(\mathbf{k}_j^{p_k})} \Pr(\mathbf{e}_{ik}^{p_e}|\mathbf{k}_{jl}^{p_k}) \qquad (3.7)$$

식 (3.7)에서 $\mathbf{e}_{ik}^{p_e}$는 구 $\mathbf{e}_i^{p_e}$를 구성하는 k번째 단어를 나타내고 $\#(P)$를 임의의 구 P를 구성하는 단어 수를 나타낸다.

구단위의 확장은 대응 단위의 확장 외에 그림 6과 같이 대응 단어들과 공기빈도가 작은 단어의 경우에도 구대응 관계를 고려함으로 정렬 정확도 향상에 기여한다. 예를 들어, '토스트'란 단어의 대역 정보가 빈약한 경우에도 주위의 가능한 구패턴과 대상 문장의 가능한 구패턴 중 가장 구대응 확률이 높은 경우를 선택함으로써 정확한 정렬이 가능했다. 실제 그림 6의 여러 가능한 패턴 중 P(명사 등위접속사 명사|명사 접속조사 명사)가 가장 높은 확률값을 가져 대응 구로 선택되었다.

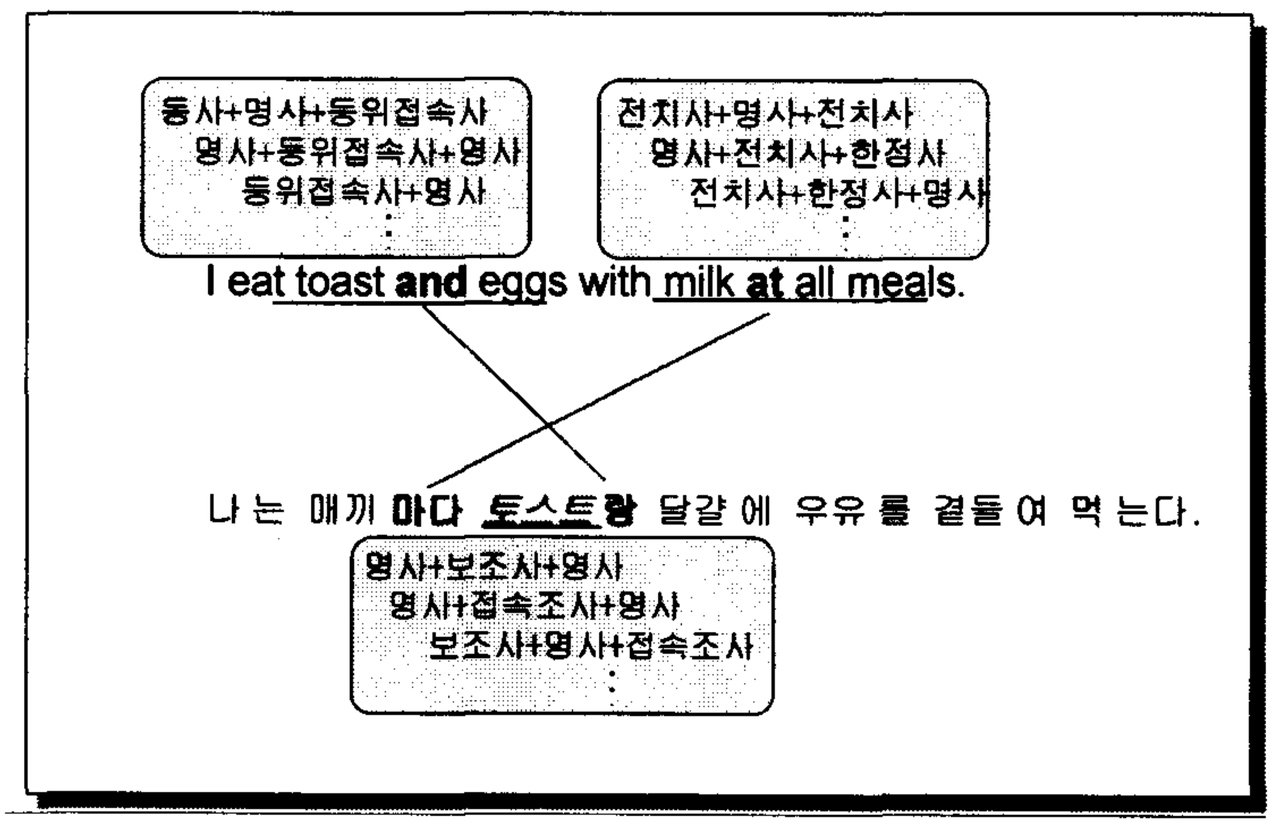


그림 6: 구대응 반영의 예

3. EM 알고리즘을 통한 파라미터 학습

앞에서 제안하는 모델의 파라미터 학습을 위해 EM(expectation-maximization) 알고리즘을 이용하였다. EM 알고리즘은 주어진 학습 코퍼스에 대해 모델의 확률값이

최대가 되도록 반복적으로 파라미터를 조정해 가는 방법으로 HMM 파라미터 학습과 PCFG의 파라미터 학습을 비롯하여 기존의 정렬모델에서 이용한 방법이다. 제안하는 모델에 대하여 적용하여 다음과 같은 과정으로 학습하였다.

우선 주어진 대응 문장 **k,e**에서 임의의 한국어 단어 k가 영어 단어 e에 대응할 확률을 $c(e|k)$로 정의하면 다음 식 (3.8)과 같이 나타낼 수 있다. 마찬가지로 $c(f(\mathbf{K}_j^{p_k}), i \mid j)$는 i번째 어절의 기능어가 j번째 영어 구와 대응할 확률을 $c(t(\mathbf{E}_i^{p_e}) \mid t(\mathbf{K}_j^{p_k}))$는 i번째 한국어 구의 품사열과 j번째 영어 구의 품사열이 대응할 확률로 정의할 때 다음 식 (3.8)~(3.10)와 같이 쓸 수 있다.

$$c(e|k;\mathbf{e},\mathbf{k}) = \frac{\Pr(\mathbf{e}|\mathbf{k})_{<e=\mathbf{e}_{ik}^{p_e},k=\mathbf{k}_{jl}^{p_k}>}}{\Pr(\mathbf{e}|\mathbf{k})} \tag{3.8}$$

$$c(f,i|j;\mathbf{e},\mathbf{k}) = \frac{\Pr(\mathbf{e}|\mathbf{k})_{<f=f(\mathbf{k}_l^{p_k}),i,j>}}{\Pr(\mathbf{e}|\mathbf{k})} \tag{3.9}$$

$$c(t_e|t_k\mathbf{e},\mathbf{k}) = \frac{\Pr(\mathbf{e}|\mathbf{k})_{<t_e=t(\mathbf{e}_i^{p_e}),t_k=t(\mathbf{k}_j^{p_k})>}}{\Pr(\mathbf{e}|\mathbf{k})} \tag{3.10}$$

위의 식에서 $\Pr(\mathbf{e}|\mathbf{k})_{<condition>}$은 문장 **e** 와 **k** 사이의 가능한 정렬 중에서 <*condition*>을 만족하는 정렬들의 확률합을 나타낸다. 즉 식 (3.11)과 같이 정의하였다.

$$\Pr(\mathbf{e}|\mathbf{k})_{<e=\mathbf{e}_{ik}^{p_e},k=\mathbf{k}_{jl}^{p_k}>} \tag{3.11}$$

$$= \sum_{<p_k,p_e>\in s(\mathbf{k},\mathbf{e})} \prod_{i=1}^{\#(p_e)\#(p_k)} \sum_{j} \Pr(t(\mathbf{e}_i^{p_e})|t(\mathbf{k}_j^{p_k})) \prod_{k=1}^{\#(\mathbf{e}_i^{p_e})\#(\mathbf{k}_j^{p_k})} \sum_{l=1} \Pr(\mathbf{e}_{ik}^{p_e}|\mathbf{k}_{jl}^{p_k}) \Pr(i|f(\mathbf{k}_j^{p_k}),m,l)\delta(k=\mathbf{k}_{a_j l}^{p_k})\delta(\mathbf{e}=\mathbf{e}_{ik}^{p_e})$$

위의 식에서 $\delta(a,b)$는 두 인수 a와 b가 같을 때 1의 값을 반환하고 같지 않을 때 0의 값을 가지는 함수이다. 전체 정렬 확률이 최대인 점에서 미분값이 0이 된다. 이러한 성질을 이용하여 전체 정렬 확률의 미분값이 0이 되도록 하기 위하여 식 (3.12)-(3.14)과 같은 재학습식을 유도해 낼 수 있다. 유도과정은 부록 B에서 전개하였다.

$$\Pr(e|k) = \lambda_k^{-1} \sum_{\mathbf{e},\mathbf{k} \in \text{Corpus}} c(e|k;\mathbf{e},\mathbf{k}) \quad (3.12)$$

$$\Pr(f,i|j,m,n) = \mu_{jmn}^{-1} \sum_{\mathbf{e},\mathbf{k} \in \text{Corpus}} c(f,i|j,m,n;\mathbf{e},\mathbf{k}) \quad (3.13)$$

$$\Pr(t_e|t_k) = \upsilon_{t_k}^{-1} \sum_{\mathbf{e},\mathbf{k} \in \text{Corpus}} c(t_e|t_k;\mathbf{e},\mathbf{k}) \quad (3.14)$$

식 (3.12)에서 λ_k는 k에 대한 라그랑지 곱수(Lagrange multiplier)로 다음 식 (3.15)와 같다. 같은 방식으로 식 (3.16)과 (3.17)과 같이 라그랑지 곱수가 주어진다.

$$\lambda_k = \sum_{e} \sum_{\mathbf{e},\mathbf{k} \in \text{corpus}} c(e|k;\mathbf{e},\mathbf{k}) \quad (3.15)$$

$$\mu_{jmn} = \sum_{jmn} \sum_{\mathbf{e},\mathbf{k} \in \text{Corpus}} c(f,i|j,m,n;\mathbf{e},\mathbf{k}) \quad (3.16)$$

$$\upsilon_{t_k} = \sum_{t_k} \sum_{\mathbf{e},\mathbf{k} \in \text{Corpus}} c(t_e|t_k;\mathbf{e},\mathbf{k}) \quad (3.17)$$

위의 수식 (3.12)-(3.14)에 대하여 다음과 같은 과정으로 파라미터를 점진적으로 학습을 하였다. 이와 같이 반복적으로 전체 정렬 확률이 최대가 되도록 파라미터를 조정해 나가는 것을 EM(Estimation-Maximization)알고리즘이라고 부른다.

1. $\Pr(e|k)$, $\Pr(f,i|j,m,n)$, $\Pr(t_e|t_k)$ 에 대한 초기값 설정
2. 주어진 학습 코퍼스에 대해 정렬 수행
3. 식 (3.15)-(3.17)에 의하여 라그랑지 곱수 계산.
4. 식 (3.12)-(3.14)에 의하여 $\Pr(e|k)$, $\Pr(f,i|j,m,n)$, $\Pr(t_e|t_k)$에 대한 확률 재학습
5. 2-4 의 과정을 수렴할 때 까지 반복

EM 알고리즘은 전체 정렬 확률이 최대가 되게 하기 위하여 정렬 확률들의 곱들의 미분값이 0 이 되도록 파라미터를 재학습시켜 가는 방법이다. 그러므로 국부 최소(local minimum)에서 수렴이 될 수도 있다. 학습을 하는 과정에서 많은 파라미터를 동시에 학습하게 되면 국부 최소에 빠질 가능성이 높다.

본 보고서에서는 많은 파라미터를 동시에 학습함으로써 국부 최소에 쉽게 빠지는 것을 방지하기 위하여 모델 1 과 모델 2 를 중간모델로 이용하여 단계적으로

학습시켜 나갔다. 우선 모델 1을 학습시키고 모델 1에서 학습한 단어 대역확률 값을 모델2의 초기값으로 사용하여 학습시킨다. 모델 2에 대한 학습이 끝나면 모델 2에서 학습한 단어 대역확률과 기능어/위치 확률을 모델 3의 초기값으로 사용하였다.

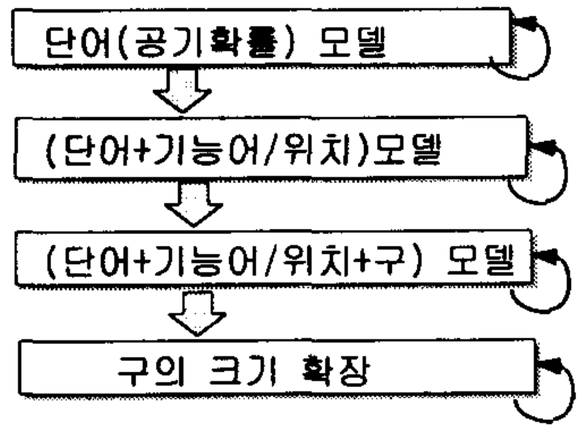


그림 7 : 중간 모델을 이용한 단계적 학습

처음 모델 3을 학습하는 과정에서 구를 최대 단어 3개로 구성된 것으로 제한을 두어 학습하였다. 학습된 구대응 확률을 이용하여 정렬을 수행한 다음 한국어와 영어에서 모두 인접한 구들에 대하여 구를 확장하였다. 이 과정을 수식으로 나타내면 식 (3.18)과 같다. 여기서 $M(p_i)$는 한국어 구 p_i가 대응하는 영어 구의 위치를 나타낸다.

$$c(t(p_j p_{j+1})|t(p_i p_{i+1})) = \frac{\Pr(\mathbf{e}|\mathbf{k})_{<M(p_i)=j,M(p_{i+1})=j+1>}}{\Pr(\mathbf{e}|\mathbf{k})} + \frac{\Pr(\mathbf{e}|\mathbf{k})_{<M(p_i)=j+1,M(p_{i+1})=j>}}{\Pr(\mathbf{e}|\mathbf{k})} \quad (3.18)$$

3장. 실험

본 연구에서 구현한 정렬 시스템은 총 25만 3천 영어 단어와 이에 대응하는 17만 8천 한국어 어절을 이용하여 학습하였다. 학습에 이용된 코퍼스의 구성 내역은 다음 표 4.1과 같다.

문서 종류	영어 단어 수(단위:만)
중학 영어 교과서	4.64
고교 영어 교과서	7.65
대입 참고서및 독해집	7.68
기타 일반 서적	5.34

표 3 : 대상 문서의 구성

본 논문에서 제안하는 정보들의 실제 정렬에서의 유용성을 증명하기 위해서 모델간의 비교실험을 수행하였다. 모델 1은 단어 간의 대역확률만 이용한 경우이고 모델 2는 모델 1에 위치/기능어 정보를 추가한 모델이고 모델 3은 모델 2에 구대역 정보를 추가한 모델이다. 모델 1과 2의 경우 한국어/영어 간의 여러 단어끼리 대응하는 경우를 반영하지 못 하였으므로 모델 3과 정렬의 정확도를 통하여 비교할 수 없었다.

정렬의 정확도를 통한 비교를 대신하여, 학습을 통해 얻어진 각 단어들의 대역확률의 정확도를 통하여 모델들을 비교하였다. 각 단어에 대한 대역단어의 정확도는 다음 두가지 관점에서 측정하였다. 첫번째는 최대의 확률 값을 가지는 대역 단어에 대해서만 정확도를 측정하였다. 두번째 방법은 추출된 대역 단어들의 정확도를 측정하였다. 정확도 측정을 학습 코퍼스 내에서 발생 빈도가 1-3번 6-7번 15-16번인 단어들로 나누어 각각 임의의 100개의 단어에 대해서 조사한 결과 아래 그림 8과 같은 결과가 나왔다. 같은 기준으로 실험2에 대하여 결과를 조사한 결과 그림 9와 같은 결과가 나왔다.

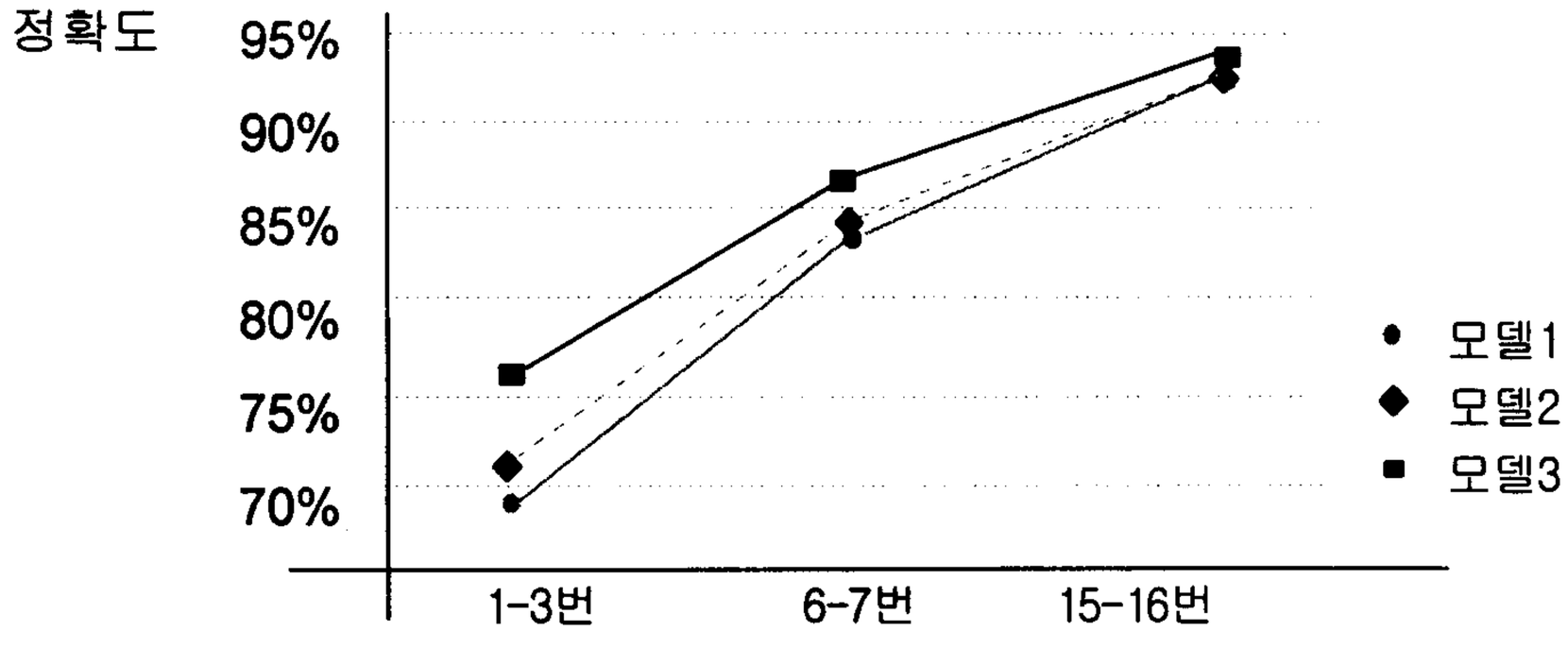


그림 8 : 실험 1(최대 확률 값을 가지는 대역 단어의 정확도)

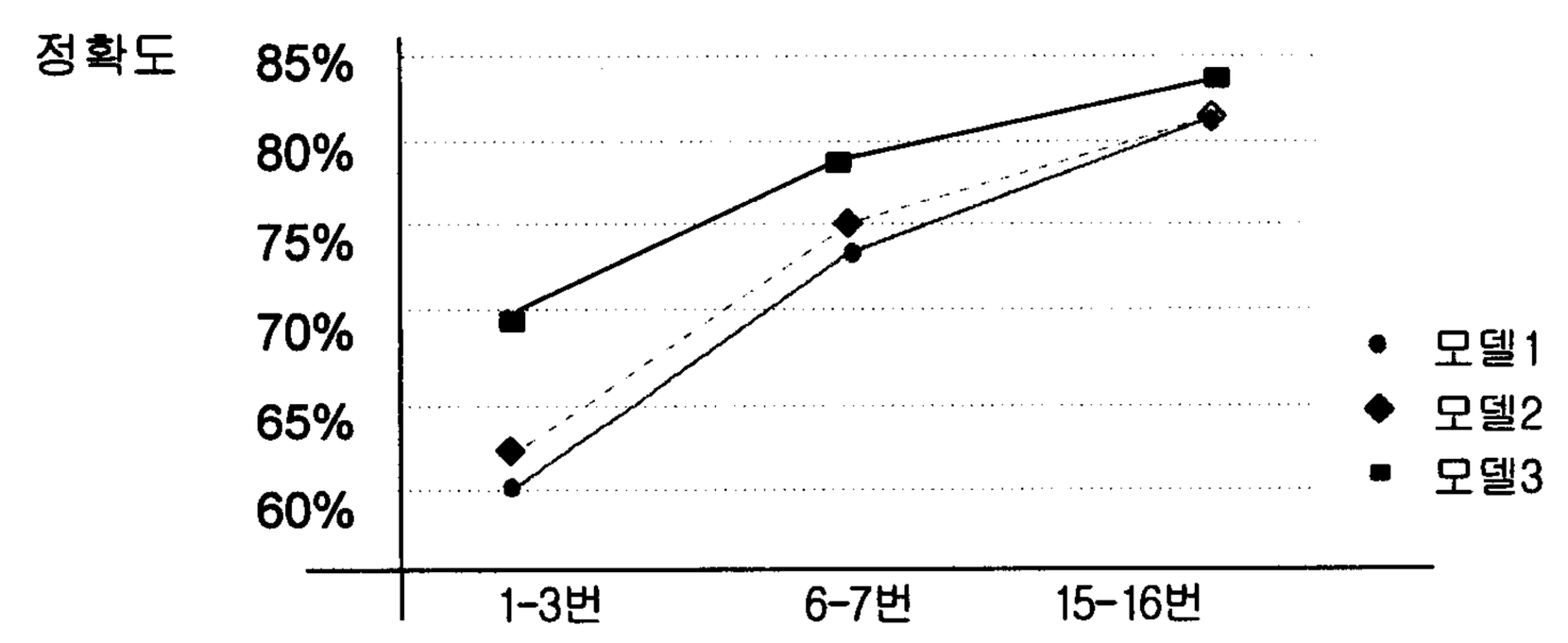


그림 9 : 실험 2(추출된 대역 단어들의 정확도)

정렬의 정확도를 측정하기 위하여 모델 3 에 대하여 정렬을 수행하고 임의의 200 문장을 선택하여 정확도를 측정하였다. 그 결과 68.7%의 구단위 정렬 정확도를 보였다.

그림 10 은 학습 코퍼스의 크기를 영어 만 단어부터 25 만 3 천 단어까지 증가시키면서 실험한 결과로 학습 코퍼스의 증가에 따라 구단위 정렬의 정확도가 59.6%에서 68.7%로 향상 되었다.

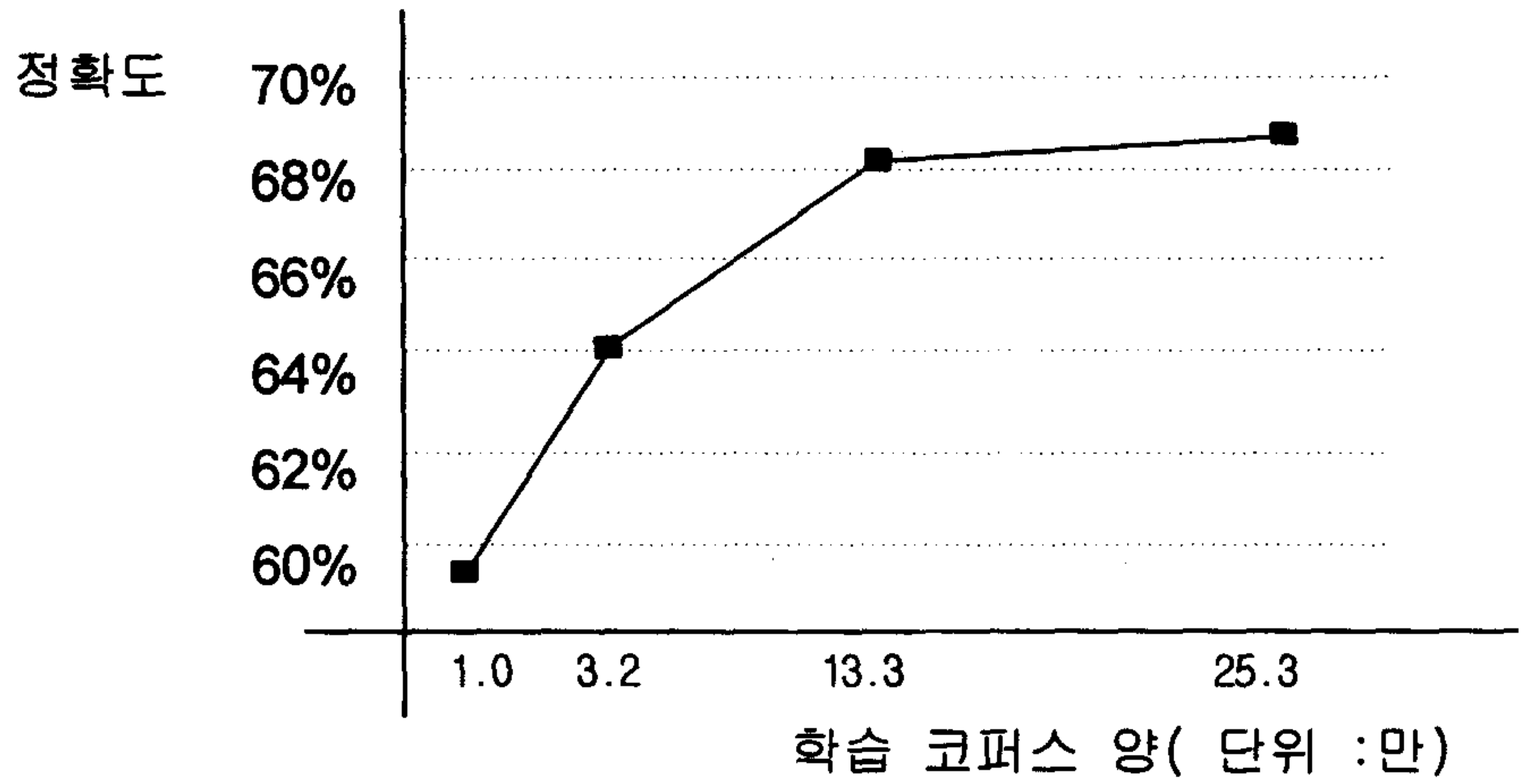


그림 10: 실험 3(학습 코퍼스 양에 따른 정렬 정확도 측정)

● 오류 분석

본 논문에서 제안하는 모델은 계산상의 복잡도를 줄이기 위하여 구를 문장 내에서 인접한 단어열로 제한하였다. 이러한 제약으로 인해 문장 내에서 서로 인접하지 않은 단어들이 대응 단위로 구성되는 경우를 제안하는 모델이 수용하지 못 하였다. 인접하지 않은 단어들이 대응 단위로 구성되는 경우를 다음과 같이 유형별로 정리하였다.

1. 의미가 분리되어 해석되는 경우:

She stays with them for only half an hour.

그녀는 가족들과 단지 반 시간 동안만 같이 있는다.

2. 숙어

You must keep the fact in mind.

너는 그 사실을 명심해야 한다.

3. 단어의 생략:

예) 영어의 대명사가 중복되어 쓰일 때 한국어에서는 일반적으로 생략된다.

They looked for flashlights, but they could not find them.

그들은 전등을 찾았지만, 발견할 수가 없었다.

4. 영어의 완료 구문이 의문문으로 쓰일 때

Has Korea ever won a gold medal in the Olympic marathon?

올림픽 마라톤에서 한국이 금메달을 딴 적이 있니?

이외에 한국어와 영어 간의 문화적 언어적 차이로 인하여 같은 상황에 대하여 판이하게 다른 구문적 형태로 표현하는 경우이다. 의역 문제가 여기에 해당한다. 의역의 경우 규칙적인 형태로 일어나는 것이 아니라 상황에 따라 각기 다른 형식으로 표현되므로 본 논문에서 제안하는 확률 모델로는 수용할 수 없었다.

예) I could not make myself understood in English

나는 영어로 의사 소통을 할 수 없었다.

4장. 결론

기존의 정렬 시스템에서는 비슷한 단어 단어나 단어 순서를 가지는 언어를 대상으로 연구되었다. 그러나 한국어/영어의 정렬과 같이 서로 다른 어족에 속하는 언어를 정렬할 경우 기본 단어 단위가 일치하지 않고 단어 순서 또한 규칙성을 찾기 힘들다는 문제점이 있다.

본 연구에서는 이러한 문제에 대처하기 위해 구대응 관계와 기능어와 위치 관계를 반영하는 한국어/영어 정렬 모델을 제안하였다. 제시하는 모델은 단어 대응에서 구대응으로 확장함으로써 기본 단위간의 상이성 극복은 물론 구단위의 대역 정보도 얻을 수 있었다. 제안하는 모델의 파라미터 학습 EM 알고리즘을 이용하여 수행하였고 학습한 파라미터를 이용한 정렬 해 선택 알고리즘을 제시하였다.

영어 25만 3천 단어와 이에 대한 한국어 17만 8천 어절을 학습 데이타로 이용하여 실험한 결과 68.7%의 구단위 정확도와 89.2%의 대역사전의 정확도를 보였다. 또한 모델 간의 비교 실험을 통하여 구대응 관계 반영과 위치/기능어 정보 반영이 대역사전 정확도 향상에 기여하는 정도를 평가하였다. 앞으로 더 많은 한국어/영어 병렬 코퍼스가 구축됨에 따라 정렬의 정확도가 향상이 예상된다.

본 연구에 대한 향후 연구 과제로 다음 세 가지를 들 수 있다.

첫째, 한국어/영어 간의 완벽한 정렬을 위하여 한국어와 영어 간의 체계적인 비교 연구가 필요하다. 본 논문에서 반영한 정보가 한국어/영어 정렬에 있어 기존 방식에 비해 유용했지만, 한국어/영어 간의 전체적인 언어적 차이점을 반영하지는 못 했다.

둘째, 본 논문에서는 구대응 관계를 반영함으로써 구단위 정보 추출이 가능했다. 그러나 구문 분석된 트리 대역 정보와 같은 대응 문장 간의 전체 구조를 반영하는 정보를 얻기 위해서는 모델의 확장이 필요하다.

셋째, 정렬 시스템은 근본적으로 병렬 코퍼스를 가정하고 있으므로 실제 기계번역에 직접적으로 적용하기 위해서는 정확한 병렬 코퍼스가 주어지지 않은 상황에 대한 고려가 필요하다. 즉 병렬 코퍼스를 이용하여 추출한 정보를 이용하여 확률 기계번역 시스템으로의 확장에 대한 연구가 필요하다.

참고문헌

[1] Baayen H. (1991). " A stochastic process for word frequency distributions." In *Proceedings, 29th Annual Meeting of the Association for Computational Linguistics*, Berkely, CA

[2] Brown, P; Lai, J. C; and Mercer, R.L. (1991). "Alignning sentences in parallel corpora." In *Proceedings, 29th Annual Meeting of the Association for Computational Linguistics*, Berkely, CA

[3] Brown, P et. al. (1990) " A statistical approach to machine translation." *Computational Linguistics*, 16, 79-85

[4] Brown, P et. al (1993) "The mathematics of statistical Machine Translation: Parameter Estimation" *Computaional Linguistics*

[5] Church, K. W and Hanks, P (1990) "Word association norms, mutual information and lexicograph." *Computational Linguistics,* 16 (1), 22-29.

[6] Dekai Wu, Xuanyin Xia. (1994) "Learning An English-Chinese Lexicon from a parallel corpus", in Proceedings of Association for Machine Translation in the America, Columbia, MD, pp.206-213

[7] Dekai Wu. (1995) "An Algorithm for Simultaneously Bracketing Parallel Texts by Aligning Words." in *Proceedings of the* 33nd *Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA

[8] Dekai Wu. (1995) Trainable Coarse Bilingual Grammars for Parallel Text Bracketing. in *Proceedings of the third Annual Workshop on Very Large Corpora*, Cambridge, MA

[9] Drela, M. and Langford, J.S. (1985) "Human-powered flights", *Scientific American*, 253(3).

[10] Drela, M. and Langford, J.S. (1986). "Fliegen mit Muskelkraft." *Spektrum der Wissenschaft*

[11] Gale, W. A. and Church, K.W. (1991) "A program for aligning sentences in bilingual corpora." In *Proceedings, 29th Annual Meeting of the Association for Computational Linguistics*, Berkely, CA

[12] Gale, W. A. and Church, K.W. (1993) "A program for aligning sentences in bilingual corpora." *Computaional Linguistics*

[13] Kay, M and Roscheisen, M. (1988) " Text-Translation alignment." Technical Report, Xerox Palo Alto Research Center

[14] Kay, M and Roscheisen, M. (1993) " Text-Translation alignment." *Computaional Linguistics*

[15] Michael R. Brent, "From Grammar to Lexicon: Unsupervised Learning of Lexicla Syntax" *Computational Linguistics*, 1993.

[16] Peter F. Brown, Stephen A. Deela Pietra, Vincent J. Della Pietra, Robert L. Mercer, "Word Sense disambiguation using statistical methods.", *Proceedings of 29th Annual Meeting of ACL*. Berkeley CA, June 1991.

[17] Kuang H. Chen, Hsin H. Chen, "Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic Evaluation." *Proccedings of 32th Annual Meeting of ACL*. 1994, 242-247.

[18] Kuang H. Chen and Hsin H. Chen. "Machine Translation: An Integrated Approach." in *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, 1995

[19] A.P. Dempster, N.M. Laird, and D.B. Rubin. "Maximum likelihood from incomplete data via the EM alogrithm. *Journal of the Royal Statistical Society*, B39:1-38,1977.

[20] Judith Klavans and Evelyne Tzoukermann, "The bicord system" *Proceedings of COLING-90*, Helsinki, Finland, August 1990, 174-179.

여백

6. 정렬 워크벤치(AWB)의 설계 및 구현

한국과학기술원
최기선

여백

6. 정렬 워크벤치(AWB)의 설계 및 구현

1장. 개요

통계적인 방법으로 병렬언어 코퍼스로부터 사전정보를 추출해 내는 연구가 세계 여러곳에서 진행되고 있다(신중호 1996; Dagan 1996; Fung 1995; Kupiec 1993). 이를 통해 추출해 낸 사전정보는 대개 번역 대응어와 대응 확률을 포함하고 있으며, 불필요하거나 잘못된 요소들도 포함되어 있어 실제 이 사전을 이용하기 위해서는 재조정 작업이 필요하다. 재조정 작업을 하기 위해서는 사전정보에 있는 단어나 구들의 정렬이 제대로 되었는가를 직관적으로 확인할 수도 있지만, 좀더 정확한 조정을 위해서는 이러한 정렬이 코퍼스의 어떤 문장에서 나온 것인가 등을 확인할 필요가 있다. 이를 위해서는 각 단어가 어떤 문장으로부터 추출된 것을 알아낼 수 있도록 정렬 과정에서 필요한 정보를 제공해 주어야 하며, 이 정보를 이용하여 사전의 내용을 확인하고 수정할 수 있는 환경이 필요하게 된다. 정렬 워크벤치는 이와같은 작업을 효율적으로 처리할 수 있도록 하는 프로그램이며, 정렬 시스템에서 워크벤치용 정보를 포함하여 만들 사전에 대해서 작동된다.

2장. 정렬 사전의 문제점

정렬 결과로 만들어진 사전(이하 정렬사전)은 통계적으로 계산에 의해 만들어진 사전이므로 직접 사용하기에는 아직 문제가 많다. 그림 1 은 실제 신중호(1995)의 정렬시스템에 의해 만들어진 사전의 예이다. 이 그림의 (1)과 (2)는 "가격"이 "price"와 "cost"로 번역되며, 각각의 대역어로 정렬된 확률이 0.153801 와 0.12462 임을 나타낸다. 이 경우는 비교적 적절한 정렬의 예가 된다.

원어	대역어	대역확률	
가격	price	0.153801	..... (1)

가격	cost	0.124624	..... (2)
국제어	international	0.273192	..... (3)
국제어	language	0.261181	..... (4)
국제어	an	0.201413	..... (5)
국제어	provided	0.135824	..... (6)
원형	amphitheate	0.156269	..... (7)
경기장	stadium	0.335752	..... (8)
ㄴ가	something	0.201688	..... (9)
ㄴ가	would	0.178573	..... (10)
ㄴ가	you	0.107707	..... (11)

그림 1. 정렬사전의 일부 (설명의 편의상 순서를 재조정함)

하지만 정렬 사전을 살펴보면 잘못된 정렬이 많이 포함되어 있다. 예를 들어 "국제어"의 경우, 원어가 두개의 단어로 분리되어 정렬하기 보다는 하나로 정렬하는 것이 더 올바른 결과를 얻을 수 있을 것이다. 즉 (3)과 (4)의 대역어가 하나로 합쳐져서 "international language" 로 되어야 한다. "국제어"의 또 다른 정렬인 (5)와 (6)은 잘못된 정렬이다. 이와는 반대로 영어 "amphitheater"는 "원형 경기장"으로 번역되어져야 하지만, 원어를 한 단어로 국한시켜 정렬하였기 때문에 제대로 된 결과를 찾지 못하고 일부의 단어로만 정렬되었다.

정렬의 사전을 보고 대략 추정할 수 있는 것도 있지만, 실제 사용된 문장을 보는 것이 더 정확한 정렬을 할 수 있다. 예를 들면 "ㄴ가"와 같은 경우, 실제 문장 예를 보지 않고서는 정확한 정렬이 되었는지를 추측하기 어렵다. "ㄴ가"는 태거에 의해 분석되어 나온 종결어미(ef)로서 그림 1 의 (9), (10), (11)처럼 3 가지로 정렬되었다. (9)의 경우에 해당하는 문장을 하나 찾아 보면 다음과 같다. (이 문장에는 정렬의 전처리 과정으로 태거를 사용하여 태깅된 문장들이다.)

호진/npd+은/jx	Ho-chin/IN
무엇/npd+이/jcp+ㄴ가/ef	held/VBN
들/pv+어/ecs	up/IN
올리/pv+었/efp+다/ef	something/NN
./s.	./S.

이 경우, "ㄴ가"는 "somthing"으로 정렬되어 있지만, 사실상 "무엇+이+ㄴ가"가 모두 "something"으로 정렬되어야 함을 알 수 있다. (10)의 경우의 문장 예는 다음과 같으며 "ㄴ가"가 "would"와 적절하게 정렬된 것으로 보인다.

누가/npd	Who/WP
아테네/nq+까지/jca	would/MD
뛰/pv+어/ecx+가/px+서/ecs	like/VB
그/npp+들/xn+에게/jca	to/TO
소식/nc+을/jc	run/VB
전하/pv+고/ecx	to/TO
싶/px+ㄴ가/ef	Athens/NNP
?"./sy	and/CC
	tell/VB
	them/PRP
	the/DT
	news/NN
	?/S.

(11)의 경우에 해당되는 문장 예는 다음과 같은 데, 이 경우 "ㄴ가"가 "you"로 정렬된다고 보기는 어렵다.

아름답/pa+ㄴ/exm 여인/nc+아/jcv ,/s, 그대/npp+는/jx 누구/npp+이/jcp+ㄴ가/ef ?/sy	Who/WP are/VBP you/PRP ,/, beautiful/JJ lady/NN ?/S.

이 사전의 예에서 알수 있듯이, 잘못된 항목의 유형은 다음과 같은 것이 있다.

- 불필요한 원어 표제어가 포함되어 있다.
- 분리되어야 할 표제어가 합쳐져서 나와 있다.
- 잘못된 역어가 포함되어 있다.
- 이러한 결과로 잘못된 번역확률이 포함되어 있다.

이와같이 정렬의 결과를 확인하고 분석하기 위해서는 정렬 결과와 관련 문장을 쉽게 연결시켜 볼 수 있어야 한다. 또한 필요한 경우, 정렬 사전을 수정하여 원하는 사전으로 만들수 있으면 편리할 것이다.

3장. 관련 연구

이런 잘못된 정렬의 결과는 정렬 시스템을 수정함으로써 향상시킬 수도 있지만, 엄청난 양의 코퍼스가 필요하게 되고 현실적으로 그 한계가 있다. Wu(1994)는 기존의 사전을 정렬의 초기 사전으로 삼아 작은 코퍼스에서 정렬이 가능하도록 했다. 또, Fung(1995)과 Kupiec(1993)은 실용성있는 정렬사전을 만들 경우, 중요한 항목은 주로 명사나 명사구인 것에 착안하여 전처리단계로 태깅을 하여 명사나 명사구에 한하여 정렬을 하였다. 이러한 연구에서는 전처리 또는 후처리 단계에서 정렬 사전을 수작업처리해야 할 필요가 있게된다.

Termight (Ido 1994)는 기술 용어의 번역이 올바르게 되었는가를 확인하기 위해 만들어진 전문 번역가용 워크벤치이다. 이 시스템은 원어의 용어들을 우선 나열한 후, 그에 대응되는 역어들을 찾아낸다. 첫단계를 위해서는 우선 태거를 사용하여 대부분 기술용어의 대상이 되는 복합 명사구(multiword noun phrase)를 찾아낸다. 이 단계에서도 태거의 결과로 나온 대상명사구를 사람이 쉽게 편집할 수 있는 환경을 제공한다. 다음 단계로 정렬 프로그램을 이용하여 복합 명사구에 대응되는 역어들을 찾아 내고, 그 역어들중 올바른 것들만을 골라 번역 용어집을 만들수 있도록 했다.

4장. 정렬 워크벤치의 작동 환경

이미 전년도에 구현된 정렬시스템은 문장단위 정렬 코퍼스를 각각 형태소 해석 및 태깅을 한 다음, 이 결과를 가지고 한/영 정렬을 하였다. 정렬 결과는 여러가지 형태의 사전으로 생성되어 나오며, 그 사전에는 대역구 사전, 단어 대역 사전, 위치 및 기능어 대역 사전이 나온다. 각각의 사전은 모두 확률을 포함하고 있다. 그러나 이러한 사전에는 그 단어 또는 구(phrase) 나 기능어 등이 실제 코퍼스 문장의 어떤 부분에서 나온 것인지를 포함하고 있지 않다. 따라서 이러한 정보를 포함할 수 있도록 정렬 시스템이 수정되었다.

출처정보로는 원문 문장 번호, 원어 단어 위치, 역문 문장 번호, 역어 단어 위치를 사용한다. 정렬과정에서 이정보는 계속적으로 사용될 필요는 없으므로, 초기에 일단 값을 가지고 있다가, 최종 정렬쌍들이 정해지면, 이 정렬쌍에 추가한다. 실제 정렬가능한 모든 쌍에 대해 출처정보를 가질 경우, 많은 저장 공간이 필요하게 되므로 제대로 확인된 정렬쌍에 대한 정보만을 갖거나, 관심있는 정렬쌍에 대해서만 출처정보를 제공하는 것이 좀더 효율적일 것이다. 여기에서 관심있는 정렬쌍이란, 예를 들어, 명사들만의 정렬이거나, 동사, 조사 등 관심이 있는 품사 등을 제한하여 정렬시키는 것을 말한다. 출처정보를 처리하는 과정을 그림으로 나타낸 것이 그림 2 이다.

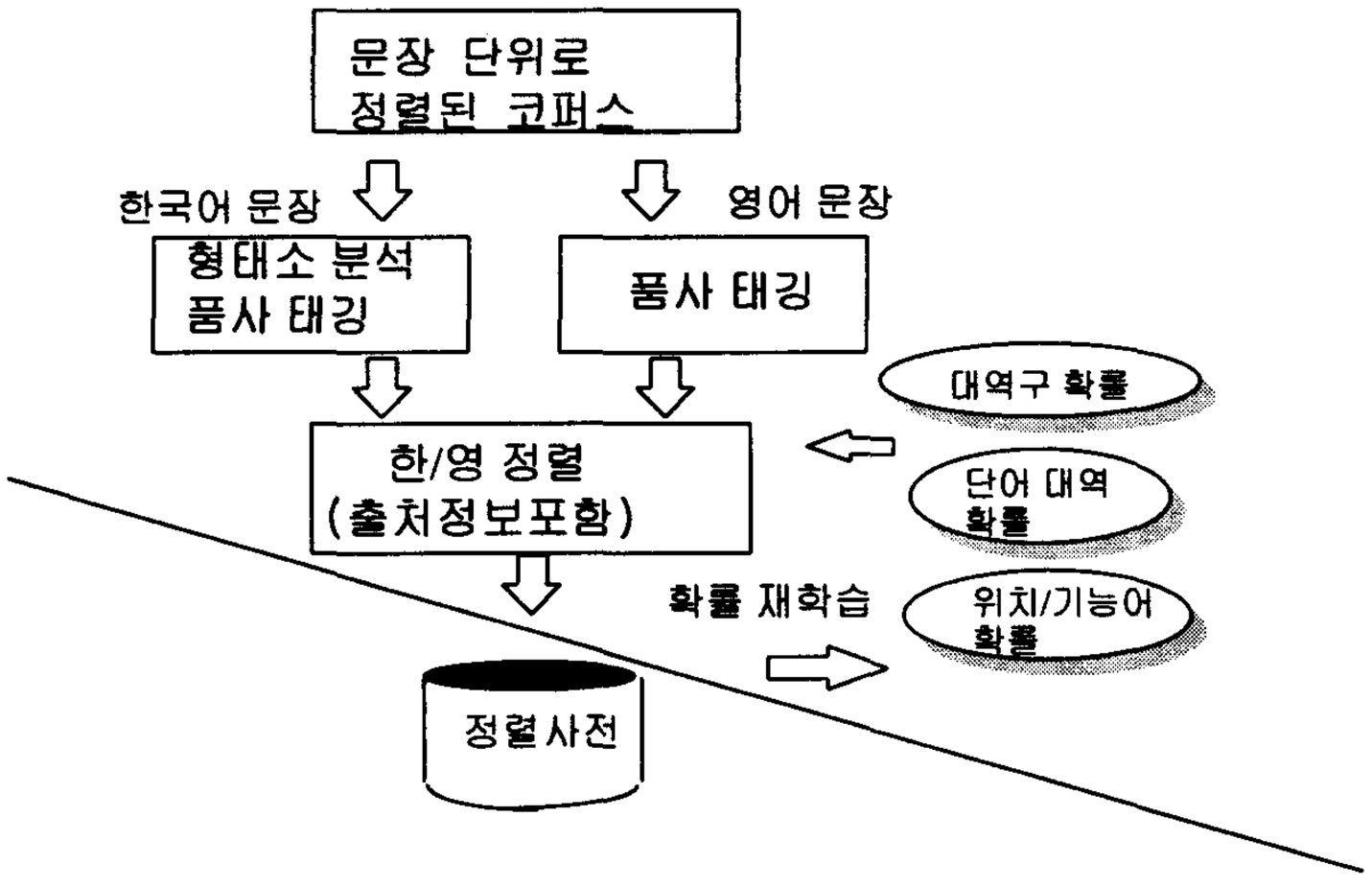


그림 2. 워크벤치 정보를 생성하는 정렬시스템

정렬 워크벤치가 다른 정렬시스템의 결과로 나온 정렬사전도 처리할 수 있도록 정렬사전의 표준형식을 지정하여 사용하고 있다. 이를 정렬워크벤치 사전형식으로 부르고 약어로서 AWD 형식으로 정하고 있다. (이에 대한 자세한 구조는 6장을 참조하기 바람.) 이 AWD 형식으로 작성된 사전이 만들어 질 경우, 정렬 워크벤치는 대역언어의 사전 구축을 위한 도구로 사용될 수 있다. 또한 역으로 번역문이 올바르게 번역되었는지를 검사할 수 있는 시스템으로도 사용될 수 있다. 즉, 어떤 중요단어가 일관되게 대역어로 사용되고 있는지, 혹은 문맥에 따라 적절히 구사되고 있는지를 확인해 볼 수 있다.

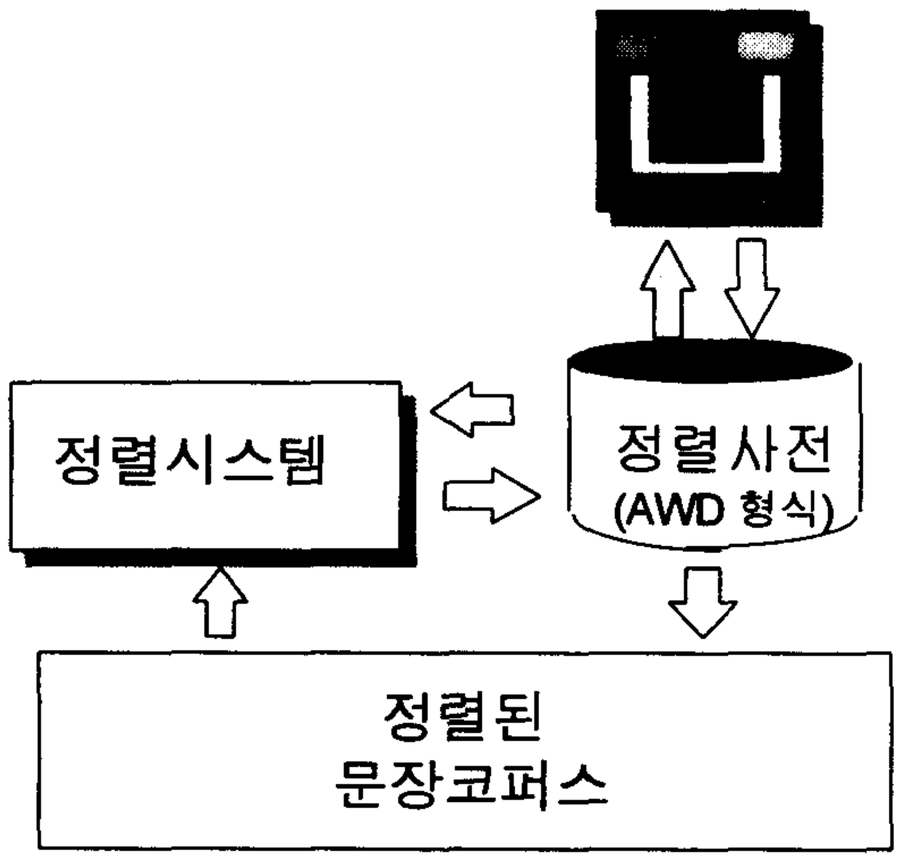


그림 3. 정렬워크벤치의 작동 환경

5장. 기본 동작

사용자가 AWB를 수행시키면, 그림 4와 같이 초기 화면이 나타난다. 이 화면의 메뉴에서 화일-열기를 선택하여 AWD 형식으로된 정렬사전을 연다. 화일이 열리면, 화면의 "원어" combo box에 원어 항목들이 나타난다. 사용자가 원하는 원어를 한 항목 클릭하여 선택하면, 대역어들이 대역확률과 함께 "대역어" combo box와 "확률" combo box에 나타난다. 사용자가 또 다시 대역어를 선택하면 원어와 대역어가 사용된 양국언어 번역예가 "번역문 대조" 상자에 나타난다. 각각의 문장 앞에는 그 문장의 번호가 표시되고, 해당되는 원어와 대역어는 밑줄과 붉은색으로 표시되어 구분을 쉽게 하도록 한다. 원어 "날씨"와 그의 한 대역어인 "weather" 에 대해 정렬된 문장예들을 확인하고 있는 화면이 그림 5이다.

새로운 항목을 추가하고자 할 경우는 바로 combo box의 편집 영역에 새로운 항목을 입력한 후, "원어(역어) 추가" 버튼을 누르면 된다. 원어 항목(또는 대역어 항목)중 삭제나 수정이 필요한 경우, 우선 그 항목을 선택한 후, 바로 "원어(역어)

삭제" 버튼을 눌러 삭제하거나, combo box 의 편집 영역에서 내용을 수정한 후, "원어(역어) 수정" 버튼을 눌려 내용이 수정되도록 한다. 대역어 확률은 새로 추가되거나 수정된 항목에 대해서는 정의되지 않은 값으로 표시된다.

원어를 삭제할 경우, 관련된 역어가 모두 한꺼번에 삭제되며, 반대로 역어를 삭제할 경우에는 그 단어만 삭제된다. 만약 한 원어에 대한 마지막으로 남은 하나의 역어가 삭제될 경우에는 그 원어도 한꺼번에 삭제된다.

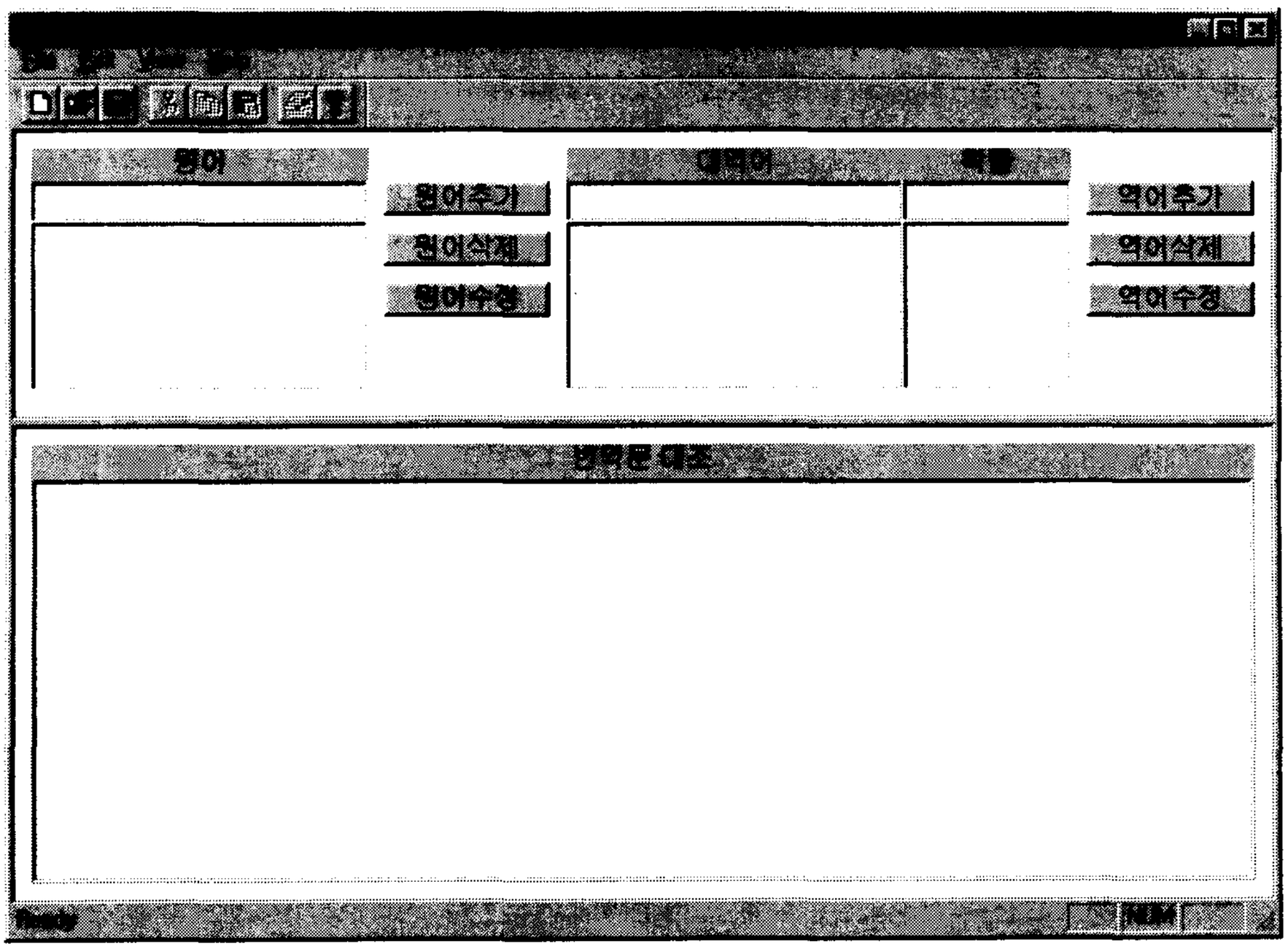


그림 4. 정렬워크벤치의 초기 화면

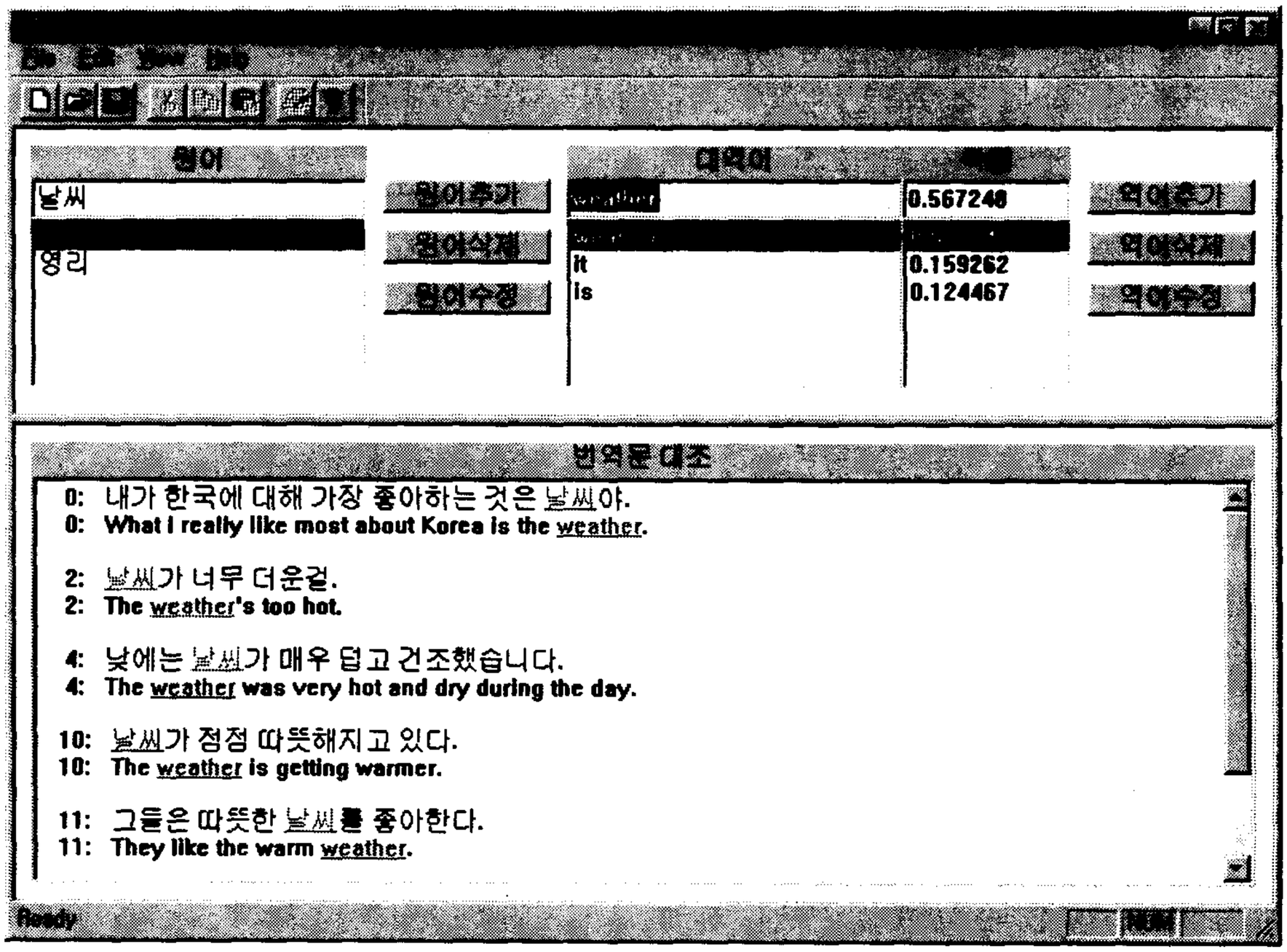


그림 5. "날씨" 와 "weather"로 정렬된 문장예를 확인하는 화면

6 장. AWD 사전 구조

사전은 원어(<SWORD>)의 나열과 그에 대한 대역어(<TWORD>) 나열, 그리고 문장예(<SSENT> 및 <TSENT>)의 나열로 구성된다. 정렬사전의 구조를 도표로 표시하면 그림 6 과 같다.

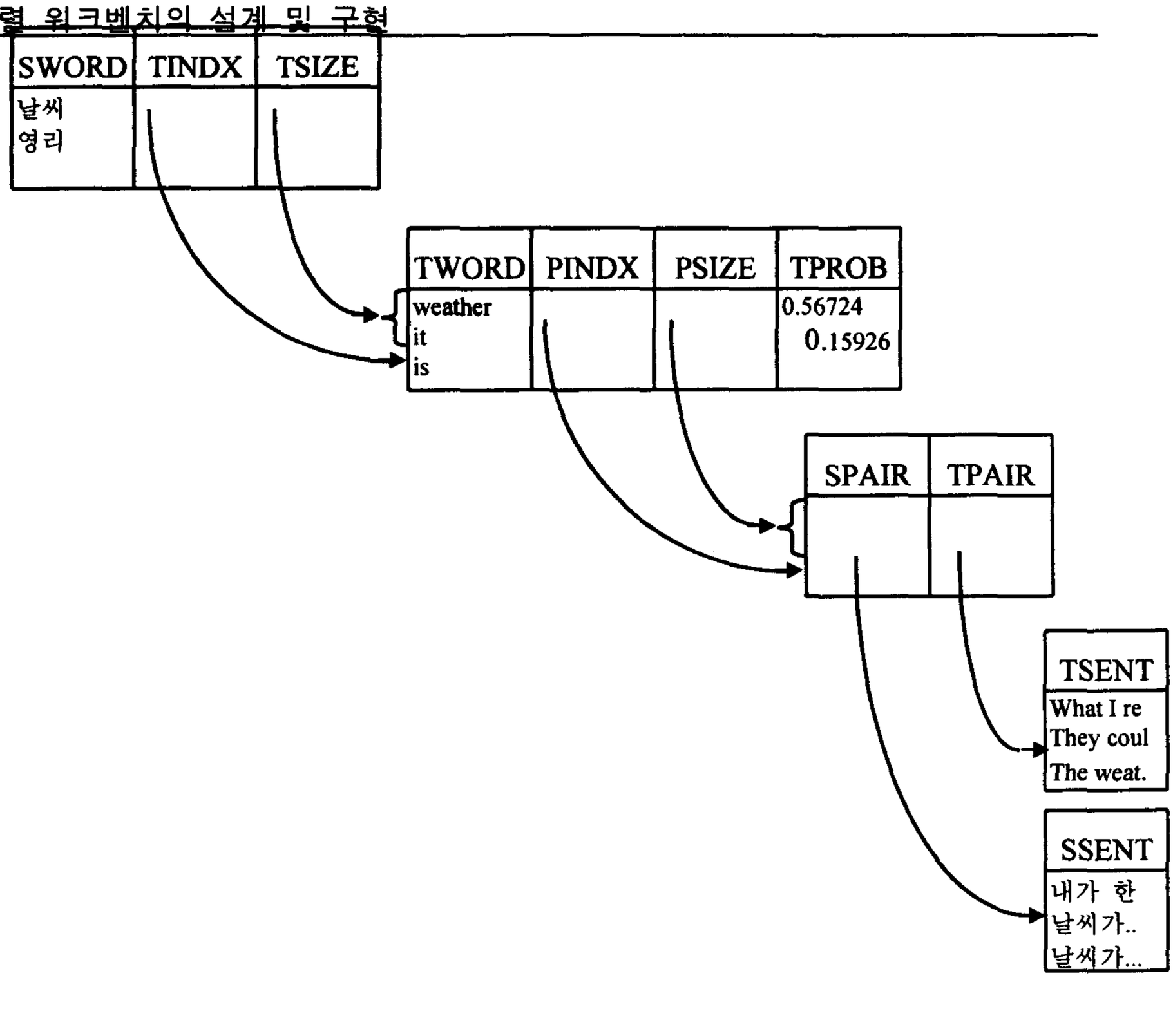


그림 6. 정렬사전의 연결 구조

각각의 항목에 대한 설명은 다음과 같다. 우선 사전의 맨앞에는 사전의 헤더 정보로서 각각의 데이터가 얼마크기로 저장되었는지를 다음과 같은 파라미터에 저장한다.

SWORD

원어의 단어 갯수

TWORD

역어의 단어 갯수

SPAIR

정렬쌍 문장의 갯수

SSENT

원어 문장의 총 갯수

TSENT

역어 문장의 총 갯수

이 헤더 다음에 실제 그림에서 설명한 사전 구조를 표현하기 위한 데이터가 저장된다. 각 데이터의 구분은 태그로 구분되며, 그 태그 뒤에 실제 데이터가 저장된다. 각 태그에 대한 설명은 다음과 같다. 구체적인 예는 부록에 나타나 있다.

<SWORD>

원어 단어의 문자열이다.

<TINDX>

역어 단어의 시작 위치(인덱스)를 나타낸다.

<TSIZE>

대응되는 역어 단어의 갯수를 나타낸다.

<TWORD>

대역어의 문자열을 나타낸다.

<TPROB>

앞의 원어에 대한 대역어의 번역 확률을 나타낸다.

<PINDX>

정렬쌍에 대한 문장예의 시작 위치(인덱스)를 나타낸다.

<PSIZE>

정렬쌍에 대한 문장예의 갯수를 나타낸다.

<SPAIR>

원어 문장예의 인덱스 순서를 나타낸다.

<TPAIR>

역어 문장예의 인덱스 순서를 나타낸다.

<SPOS>

원어 문장예에서 단어의 위치를 나타낸다. 위치는 문장 첫글자로부터의 글자(1 바이트 단위) 갯수이며, 문장의 첫글자는 0 이 된다.

<TPOS>

역어 문장예에서 단어의 위치를 나타낸다. 위치는 문장 첫글자로부터의 글자(1 바이트 단위) 갯수이며, 문장의 첫글자는 0 이 된다.

<SSENT>

원어 문장의 문자열이다.

<TSENT>

역어 문장의 문자열이다.

7 장. 프로그램 구현

AWB 는 Visual C++ 로 구현되었으며, MFC (Microsoft Foundation Class)를 사용하였다. 따라서 기본적으로 객체지향 프로그램으로 되어 있고, 한글 윈도우 95 에서 작동되도록 되어 있다.

AWB 는 SDI (Single Document Interface)로 만들어 졌다. 하지만, 한 윈도우에서 2

개의 화면을 크기를 변경하며 볼 수 있도록 split window를 사용했다. 각 객체의 관계를 간단하게 표시하면 다음과 같다.

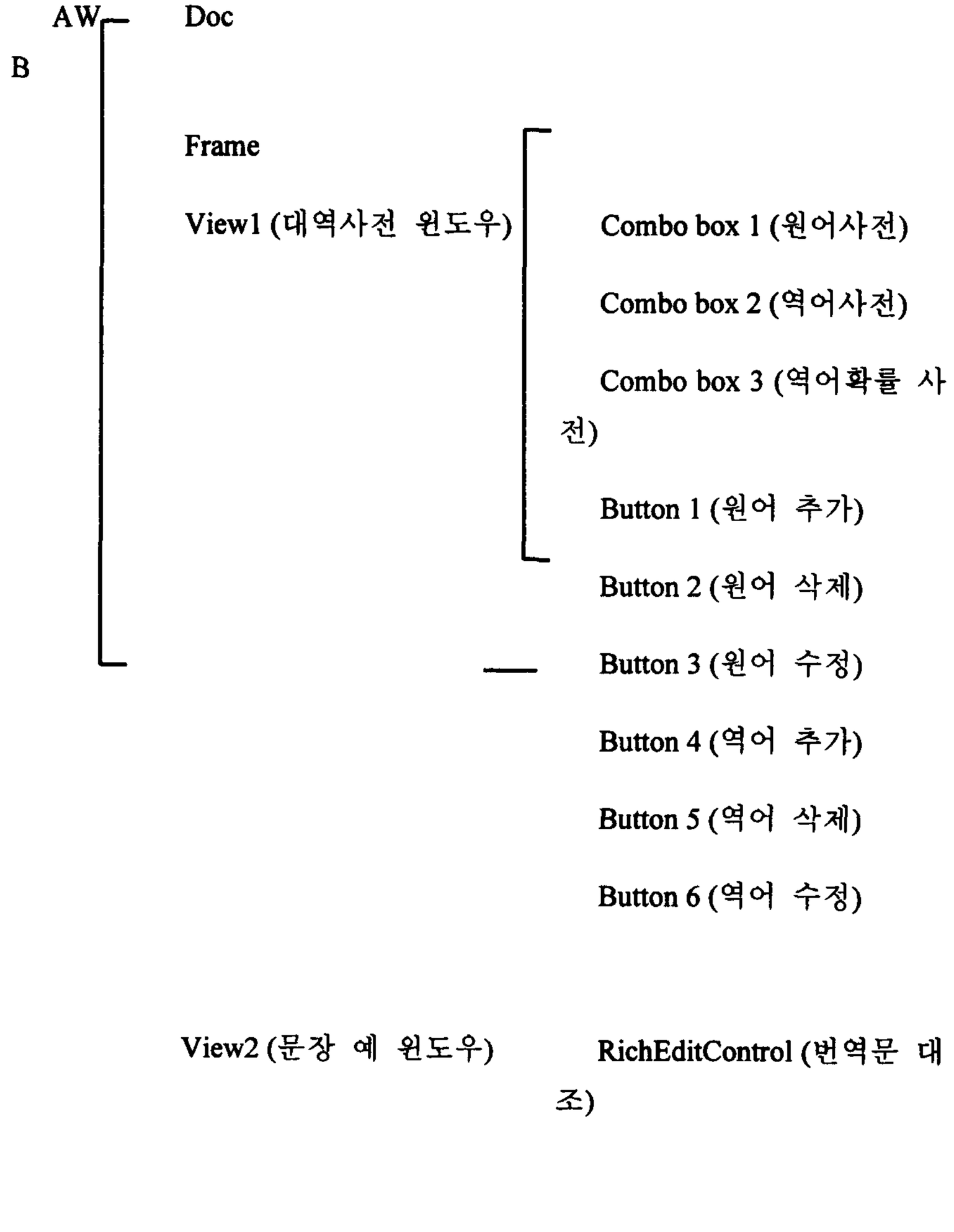


그림 7. 정렬워크벤치 프로그램의 주요 객체 관계도

View1과 View2는 Frame 윈도우에서 split 윈도우로 포함되어 있으며, split bar

를 사용자가 움직일 경우, 화면의 크기가 변화된다. 사용자가 AWB 전체 윈도우의 크기를 조정하거나, 이 split bar 를 조정하여 View1 과 View2 의 윈도우 크기가 변했을 경우, 그 이하의 자식 윈도우들도 모두 새로운 크기로 조정된다. 이는 WM_SIZE 메시지에 따라 수행되는 OnSize 함수에서 처리한다.

원어사전과 역어사전(확률포함)은 DOC 객체에서 만들어준 데이터를 읽어와 combo box 형식으로 입력시켜 화면에 표시한다. 원어의 경우, 초기에 한번 사전이 구성되어 있으면 편집을 하지 않는 한 combo box 내용에는 변화가 없다. 역어의 경우는 원어의 선택에 따라 동적으로 새로운 사전 엔트리가 구성되어 combo box 에 저장된다. Combo box 1 에서 사용자가 새로운 선택을 할 때마다, 이 내용 반영할 수 있도록 DOC 객체의 SetupTword 함수를 호출한다. 이 결과로 새로운 역어사전을 구성해 둔다. 이 내용이 Combo box 2 에게 알려지어 화면에 표시될 수 있도록 Combo box 1 에서 화면 수정을 요구한다.

사용자가 다시 Combo box 2 나 3 를 통해 역어를 선택하면, 그에 해당되는 문장예들이 준비된다. 이 문장예는 DOC 객체의 SetupSent 함수를 호출하여 이루어 지고, 이 결과를 View2 의 RichEditControl 에 알려준다. View1 에서 View2 로의 호출은 위해 윈도우 핸들을 통해 이루어지며, 필요한 경우 View1 에서 직접 View2 의 함수를 호출하기도 한다.

초기의 화일 읽기는 File Open 을 통해 AWD 화일을 읽어 이루어 진다. 이 내용은 DOC 객체에서 처리하며, 외부 화일 형태를 내부에서 처리할 수 있는 형태로 변환한다. 반대로 내부 형태의 정렬사전을 외부형태의 사전으로 변환시켜 저장할 수 있다. 이 경우는 편집을 통해 사용자가 정렬사전을 수정한 경우이며, 문장예는 참고용이므로 편집을 할 수 없다.

8 장. 맺음말

많은 이중언어 코퍼스(병렬 코퍼스)가 많들어지고 있다. 이를 이용하여 유용한 번

역 정보를 언어내는 정렬시스템에 대한 연구가 이미 전년도에 수행되었다. 이 정렬 시스템은 통계적 방법을 이용여 보다 객관적인 정보를 제공해 주고, 또한 대량의 데이터를 손쉽게 처리할 수 있도록하고 있다. 그러나 이 시스템은 정확한 정보를 제공하지 못하고 잘못된 정보를 포함하여 제공하고 있다. 이러한 정보를 올바르게 가려내어 유용한 정보만을 찾아 낼 수 있는 도구로서 정렬 워크벤치 시스템을 설계하고 완성하였다. 이 시스템은 현재 기본 기능을 완성하였고, 많은 데이터와 사전구축에도 이용될 수 있도록 설계되었다. 따라서 실제 사전 구축이나 정렬시스템의 보완, 번역문의 확인 등에 이용될 수 있으며, 그 유용성에 대한 정확한 평가는 여러분야에서 사용해본 후에 내릴 수 있을 것이다.

이 시스템은 현재 기본적인 기능에 대해 완성되어 있으며, 정렬시스템과는 분리되어 작동되고 있다. 경우에 따라서는 정렬시스템과 함께 작동하는 것이 보다 편리할 수도 있을 것이다. 또한 사용자들이 보다 편리하게 번역 용례등을 찾아볼 수 있는 시스템으로도 변형하여 사용할 수도 있을 것이다.

참고문헌

신중호 (1996), "한국어/영어 병렬 코퍼스에 대한 단어단위 및 구단위 정렬 모델," 석사학위논문, 한국과학기술원.

P. F. Brown, and et al. (1990), "A Statistical Approach to Machine Translation," Computational Linguistics, Vol 16, Num. 2, June.

P. F. Brown, and et al. (1993), "The Mathematics of Statistical Machine Translation: Parameter Estimation," Computational Linguistics, Vol 19, Num. 2.

I. Dagan, et al. (1994) "Termight: Identifying and Translation Technical Terminology," ACL ANLP.

I. Dagan. (1996), "Bilingual Word Alignment and Lexicon Construction," ACL96 Tutorial, June.

Pascale Fung (1995), "A Pattern Matching Method for Finding Noun and Proper Noun Translations from Noisy Parallel Corpora," ACL95.

Julian Kupiec (1993), "An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora," ACL.

Dekai Wu and Xuanyin Xia (1994), "Learning An English-Chinese Lexicon from a parallel corpus," in Proceedings of Association for Machine Translation in the America. 206-213.

부록. AWD 사전 예

<AWB dictionary file>

SWORD 2

TWORD 6

SPAIR 22

SSENT 22

TSENT 22

<SWORD>

날씨

영리

<TINDX>

0

3

<TSIZE>

3

3

<TWORD>

weather

it

is

clever

smart

cleverness

<TPROB>

0.567248

0.159262

0.124467

0.441500

0.170629

104207

<PINDX>

0

6

12

13

18

20

<PSIZE>

6

6

1

5

2

2

<SPAIR>

0

2

4

10

11

12

1

3

6

7

8

9

5

14

15

16

17

18

13

21

19

20

<TPAIR>

0

2

4

10

11

12

1

3

6

7

8

9

5

14

15

16

17

18

13

21

19

20

<SPOS>

37

1

1

8

8

28

6

8

22

1

1

15

20

20

8

6

20

6

18

20

24

22

<TPOS>

44

50

5

1

5

23

13

5

7

5

5

20

23

22

8

25

54

8

69

27

32

18

<SSENT>

내가 한국에 대해 가장 좋아하는 것은 날씨야.

날씨가 흐렸기 때문에 달이나 별들을 볼 수 없었다.

날씨가 너무 더운걸.

어제는 날씨가 흐렸어.

낮에는 날씨가 매우 덥고 건조했습니다.

오늘은 진열장 구경하기에는 날씨가 너무 좋다고 생각 하지 않니?

내일 날씨가 좋을까?

그러나 날씨가 춥다.

이것 봐! 참 아름다운 날씨지, 안 그래?

날씨도 따뜻해.

날씨가 점점 따뜻해지고 있다.

그들은 따뜻한 날씨를 좋아한다.

그토록 덥고 비오는 날씨는 마침내 지나갔니?

그것은 작지만 매우 영리해.

참으로 영리한 농부야!

그는 영리한 하인에 관한 이야기를 읽었다.

여행이 끝날 무렵에 영리한 하인이 운반할 것은 아무것도 없었다.

그는 영리하였다.

그는 자신이 너무 영리해서 아무도 자신을 속일 수 없다고 생각했다.

그는 때때로 자신의 영리함을 자랑했다.

벤자민은 다시는 자신이 영리하다는 말을 하지 않았다.

그래, 이 기계는 매우 영리하지.

<TSENT>

What I really like most about Korea is the weather.

They could not see the moon or the stars because it was cloudy.

The weather's too hot.

It was cloudy yesterday.

The weather was very hot and dry during the day.

Don't you think today is too beautiful for window shopping?

I wonder if it will be fine tomorrow.

But it is cold.

Look! It is a beautiful day, is not it?

And it is warm, too.

The weather is getting warmer.

They like the warm weather.

Has all that hot, wet weather finally gone away?

It is small but very smart.

What a clever farmer!

He read a story about a clever servant.

At the end of the journey, there was nothing for the clever servant to carry.

He was clever.

He thought that no one could play a trick on him because he was too clever.

He often talked about his cleverness.

Benjamin never spoke about his cleverness again.

Sure. It is very smart.

<end of AWB dic>

여백

7. 통합 국어정보베이스 인터페이스와 WWW 디자인

우송산업대학교
이창조

여백

7. 통합 국어정보베이스 인터페이스와 WWW 디자인

1장. 연구개발의 필요성

1절. 연구개발의 경제, 사회, 기술적중요성

1. 인터넷을 이용한 정보의 공유

인터넷이란 컴퓨터 통신의 한 종류이다. 흔히 인터넷을 가리켜 정보의 바다라든가 정보의 보고, 또는 정보고속도로, 네트웍의 네트웍 등의 표현을 쓴다. 인터넷에는 방대한 양의 정보가 들어있다는 의미이다. 그러나 이러한 표현은 인터넷을 수식하는 용어일뿐, 인터넷 그 자체가 무엇인가는 설명해 주지 못한다. 따라서 인터넷의 정의를 요약해 보면 다음과 같다.

- 인터넷은 전세계를 연결하는 '컴퓨터 통신망'
- '그 통신망 속에 들어있는 정보'
- '그 통신망을 만들고 이용하는 사람들의 공동체'

2. 월드와이드 웹(World Wide Web)의 등장

인터넷은 이제 30년 가까운 역사를 갖게 되었다. 그 동안에 서비스의 내용과 질에서도 많은 변화를 겪게 되었는데, 그 중 최근에 가장 주목받는 서비스가 월드와이드웹이다. World Wide Web, 즉 세계적으로 광범위한 Web(=거미줄)이라는 뜻이다. 거미줄이라는 표현을 붙인 이유는 인터넷이 네트웍으로 이루어졌기 때문에 거미줄에 비유한 것이다. 월드와이드 웹(World Wide Web)은 줄여서 WWW, W3, 웹(Web) 등으로 부르기도 한다. 웹은 1989년 스위스에 소재한 입자물리연구소(CERN)의 연구원인 Tim Berners-Lee가 제안한 프로젝트로부터 시작되었다. 기존의 Plain Text를 Hypertext 방식으로 바꾸어 정보를 제공하자는 아이디어에서 출발하여 웹의 내용을 볼 수 있는 다양한 프로그램(Web Browser)들이 등장함으로써 인터넷의 대명사가 된 것이다. 웹은 하이퍼 텍스트 방식의 정보제공 외에도 몇가지 특징

을 갖고 있다. 원격접속, 파일전송, 전자우편, 뉴스, 고퍼 등의 인터넷의 대표적인 기능을 포함하고 있는 것이다. 또 한가지는 멀티미디어 기능을 제공한다는 점이다. 1970 년대와 1980 년대 UNIX 환경에서 텍스트 위주로 이용되던 인터넷이 1990 년대부터는 윈도우 환경으로 바뀌고 멀티미디어 위주로 변화되었다. 웹도 처음에는 문자정보로 시작되었다. 초기에는 'www', 'lynx'와 같은 프로그램이 개발되어 문자만을 읽고 찾을 수 있는 방식으로 운영되었다. 그러나, 1990 년대 초에 모자이크(Mosaic), 넷스케이프(Netscape)같은 프로그램이 개발되어 웹은 문자뿐 아니라 그림과 음성, 동화상 등을 지원하는 멀티미디어 정보로 제공됨으로써 많이 사용하고 있다. 그러므로 World Wide Web 을 이용한 우리 국어정보의 체계적인 구축 및 효율적인 디자인을 이룩함으로써 국내외의 기술적 동향 및 정보의 기반을 구성하고 널리 이용하는데 그 목적이 있다.

3. 국내 인터넷 회선 서비스 기관 현황

나우콤 http://www.nowcom.co.kr

넥스텔 http://www.nextel.net

데이콤 http://www.dacom.co.kr/

제이씨현시스템 http://www.elim.net

아이네트기술 http://www.inet.co.kr

아이월드 http://www.iworld.net

SDS 유니텔 http://www.unitel.co.kr

한국통신(KORNET) http://www.kornet.nm.kr/

현대전자 아미넷 http://www.aminet.co.kr/

한국 PC 통신 http://www.kol.co.kr

한국무역정보통신 http://www.ktnet.co.kr

한글과컴퓨터 인터넷 서비스 http://hncnet.co.kr/

4. 국내 웹서버 구축 기관현황

나라비전넷 http://www.nara.co.kr

넥슨 http://www.nexon.co.kr

다음커뮤니케이션 http://www.daum.co.kr/

마하인터넷서비스 http://www.directory.co.kr/

미래정보기술 http://www.fit.co.kr

브라이트시스템즈 http://www.bright.co.kr

블루버드인터넷 http://www.go.co.kr

사이버플래닛코리아 http://www.cpk.co.kr

사이버랜드 http://www.cyberland.co.kr/

사이버텍 홀딩스 http://www.nana.cybertek.co.kr

씨엠아이코리아 http://www.koreaweb.co.kr

아라크네 웹 디자인 http://www.arachne.co.kr

아이소프트 http://aisoft.trigem.co.kr/

아이큐브 http://ain.icube.co.kr

애드버네트 http://advernet.co.kr

와이즈디베이스 http://www.wisedb.co.kr

원인포메이션 http://sol.nuri.net/~oneinfod

이미지 시스템 http://www.image.co.kr/

이지넷 http://easynet.image.co.kr

인터네트 코리아 http://ink.ink.co.kr/

큰틀 http://www.knet.kntl.co.kr

테일자동제어 http://cyberbiz.co.kr

2&5 시스템 http://tfsys.co.kr/

파워넷 http://www.power.co.kr

한국인터넷시큐리티 http://www.kms.co.kr

2절. 문제점 및 전망

1. 국내 웹디자인의 현황

여러 프로젝트를 진행하면서 , 대부분의 국내 Web 디자인이 어떤 작가의식 없이 진행된다는 것이다. 물론 디자인이라는 것이 작가의식만 고집한다고 되는 것은 아니지만 그렇다고 자신의 디자인을 규격화 해놓고 얼마짜리에는 이렇게, 얼마짜리에는 저렇게라는 식으로 제작이 된다면 문제가 있지 않을까 하는 생각이 든다. 최소한 이것은 내가 만들었고 그렇기 때문에 어디에 내놓아도 자랑스럽다라는 느낌이 있어야 할텐데 일을 맡고 끝내는데 급급하는 것 같다. 그리고 이러한 것들은 단순히 디자이너들만의 문제가 아니라고 본다. Web 디자인을 의뢰하는 기업들의 의식에도 많은 문제가 있다고 본다.

대부분의 기업들은 Web HomePage 를 특별한 목적의식 없이 남들이 하니까, 경쟁사도 이미 만들었는데...라는식으로 제작 을 시작한다. 그러다 보니 가지고 있는 브로슈어를 그대로 Web 으로 올리는 정도밖에는 진행을 못하고 있다. 게다가 Web HomePage 란 기업의 CI 이상으로 그 회사를 대표할 수 있는 얼굴 역할을 하게 된다. 하지만 대부분의 기업들은 이를 간단한 브로슈어 제작 정도로만 생각하고 일을 의뢰하는것 같다. 결국 나오는 결과도 그 정도라고 할 수 있다. 가장 이상적인 형태는 작가정신을 가지고 있는 제대로 된 집단과 선진화된 마인드의 기업이 만났을때라 생각한다.

2. 웹디자인 관련 Site

CMDesigns의 HomePage 라든가 Brad Johnson의 Home Page, Organic Online, ReZn8등 재미있는 Design 관련 Site 들이 많이 있다. 하지만 더욱 재미있는 것들은 스스로가

그러한 Site 들을 찾아보는 것이 좋은 방법이다.

3. 인터넷 광고

인터넷이 세계를 묶어주는 정보체계로 등장하면서 인터넷을 상업적인 목적으로 이용하려는 움직임이 중요하게 부각되고 있다. 즉 많은 기업과 단체, 심지어 국가나 정부기관, 대학과 박물관, NASA 같은 조직, 전시화를 하려는 단체들까지도 인터넷을 이용하여 자신을 알리고 발신자와 수신자가 1:1 의 관계를 가지므로, 고객관리를 하거나 상품을 주문받거나 고객의 불만을 처리하는 등 대외적인 커뮤니케이션에 활용하며, 나아가서 접속하는 수신자의 정보를 데이타베이스로 구축하고 이를 활용하여 데이다 베이스 마케팅을 실시하려는 시도도 눈에 띄게 증가하고 있다. 그러나 아직은 미국을 중심으로 한 영어권을 제외한 곳에서 활동영역이 자국에 머무르는 단체나 기업의 경우는 종전이 미디어보다 훨씬 저렴한 가격에도 불구하고 별다른 효과를 보기 어렵다. 그것은 인터넷 사용자가 주로 자국보다는 국외정보에 관심을 기울이기 때문이다. 인터넷 이용자는 대체로 정보지향적이며 고학력 보유자들이며 소득수준도 상대적으로 높을 것은 분명하지만 이들이 과연 기업의 상품판매를 위한 광고의 목표 고객으로서 가치가 있는가는 아직 미지수이다.

인터넷에 광고를 할 때에는 먼저 인터넷상에서 유력한 Web Site 선택하고 게재하는 위치를 결정하며 노출빈도를 결정하는 일이 필요하다. 인터넷상에서 이용빈도가 높은 WebSite 는 방송이나 신문의 시청율, 구독률이 높으면 광고단가가 높게 책정되는 것처럼 광고의 단가가 높게 책정된다. WebSite 에 따라서 게재하는 광고가 세계의 몇개의 지역으로 나누어서 그 지역에만 노출될 수 있도록 하는 경우도 있으므로 지역을 고려한 집행도 부분적으로 가능하다.

인테넷에서 많이 알려진 2 개의 웹사이트에서 광고를 집행하는 방법을 알아보면 가장 유명한 인터넷 웹브라우저의 하나의 넷스케이프의 경우 접속자의 수를 기준으로 3 개의 프로그램으로 구분하여 광고비를 달리 책정하고 있다. 넷스케이프 자체에도 정보검색을 위해 20 여개의 개별 카테고리로 구성되어 있는데 몰(mall)이용자가 100 만명 이상인 카테고리를 Platium Program 으로 묶어 월 3 만 4 천 달러 이상으로 책정하고 있고 그보다는 낮은 이용자수를 가진 카테고리를 묶어 Gold Program 이라하여 월 2 만 3 천 달러정도 이용자수가 50 만명 수준인 카테고리

를 묶어 Silver Program 이라하여 광고단가는 만 7 천 달러 정도로 하고 았다. 그리고 각각의 카테고리는 이용자가 늘어나면 상위프로그램으로 올라가고 이용자가 줄면 하위 프로그램으로 조정하여 미디어 집행의 합리적 운영을 기하고 있다.

인터넷에서 많이 이용되는 검색엔진인 야후의 경우 11 개의 카테고리가 있는데 각 카테고리는 각기 다른 광고단가를 책정해 놓고 있는데 가장 비싼 광고비용을 요구하는 카테고리는 Entertainment로 월 5만7천 달러 정도이다. 광고이 운영에 있어서 단독으로 광고를 하는 경우와 몇개의 광고가 번갈아 가면서 등장하도록 하는 방법을 모두 채택하고 있어 야후측과 협의하기에 달려있다. 이 광고 단가는 야후측과 전자메일로 연락하면 바로 자세히 알 수 있다.

인터넷광고는 통상 Banner 광고라고도 하는데 이 광고의 크기는 사용하는 모니터의 화면에서 해상도를 어떻게 설정하는가에 따라 달라지나 640×480 의 경우에 약 3cm×18cm 가 되는데 이 Banner 광고에는 많은 내용을 실을 수 없고 단지 시선을 끌고 주목하게 하는 기능을 하며 이 Banner 광고를 클릭하면 홈페이로 연결할 수 있도록 하고 있다. 그러나 인터넷광고를 한다는 것은 단순히 Banner 광고를 만드는 것이 아니라홈페이지가준비되어야 하는 것은 필수적이다. 다시말해 기업이나 단체는 자신의 홈페이지를 개발하여 알리고 싶은 메세지를 만들고 이 홈페이지로 유도하는 기능을 하는 것이 Banner 광고의 역할이라고 할 수 있다. 인터넷상의 무잔장한 정보들중에서 자신의 홈페이지를 고객들이 찾아 간다는 것은 세계적인 명성을 얻고 있는 기업이나 단체가 아니면 불가능하므로 충분한 지명도를 가지지 못한 기업/단체는 자신의 홈페이지로 연걸되는 고리로써 인터넷 광고를 활용할 수 있을 것이다.

2 장. 연구개발의 목표 및 내용

1 절. 통합국어정보베이스 WWW 구축을 위한 인터페이스 디자인(Interface Design)

디자인 개념이 생겨난 이래로 다양한 분야에서 여러가지 디자인 개념이 도입되어 왔으나 오늘에 이르러 그 기능적 측면이 보다 강조되고, 컴퓨터와 각종 첨단 시스템들이 생활과 밀접한 관계를 갖게 되자 디자인은 단순한 심미적 적용에서 벗어나 보다 현실적인 공학적 측면과 깊은 상관관계를 맺게 되었다. 즉 디자인을 행해

야 할 범위가 보다 넓어졌을 뿐 아니라 그 깊이 또한 디자이너의 끊임없는 노력과 도전을 필요로 하는 전문적 지식의 수반을 요구하고 있는 것이다. 그러므로 인터페이스 디자인이 필수로 하고 있는 공학적 측면을 유심히 관찰한다면 시스템과 사용자 간의 인터페이스가 곧 디자이너와 사용자 간의 대화임을 느낄 수 있을 것이다.

인터페이스란 일반적으로 두 종류의 서로 다른 세계가 상호 교섭하는 장을 의미한다. 이를 바탕으로 '사용자 인터페이스(User-Interface)'란 용어가 파생하였고 이것은 1차적으로 사람과 시스템 등의 접점 혹은 하나의 대상과 또 다른 대상과의 접점을 의미하며 2차적으로는 사용자와 각각의 시스템 사이의 '정보채널'로 받아들여지고 있다. 즉 이 정보채널의 과정에서 사용자와 시스템 간의 대화가 보다 효율적으로 이루어질 수 있도록 심미적, 공학적인 프로그래밍을 개입시키는 것이 바로 '인터페이스 디자인'인 것이다.

'정보채널'이 존재하고 있는 모든 것이 인터페이스의 영역이며 그 개념 또한 점차로 확대되어가고 있는 추세이다. 무선 리모콘으로 조정되는 TV 수상기에서부터 최첨단 가상현실 시스템에 이르기까지 사용자의 사고와 시스템간의 교류가 이루어지는 모든 곳에서 인터페이스 디자인은 영향을 끼친다. 특히 cd-rom title, web, software programing 등의 첨단 정보 시스템은 사용자 인터페이스가 다양하고 지속적으로 요구되는 분야로써 디자인의 역할이 중요한 몫을 차지하고 있다.

인터페이스 디자인은 단순히 칼라,아이콘 등의 그래픽적 요소 뿐 아니라 메뉴의 진행방식등과 같이 프로그램의 전반적인 진행에도 영향을 미친다. 그러므로 인터페이스 디자이너는 그래픽 디자인 뿐 아니라 프로그래밍, 인간공학, 심리학 등 다방면에 걸친 지식이 필요하다.

인터페이스 디자인은 디자인의 기본 요소인 기능적 아름다움에 더하여 '직관성'이라는 가장 중요한 요소를 포함한다. 그것은 단순화의 개념으로서 메뉴의 진행은 단순명료하며 최소한의 선택으로 소기의 목적을 달성할 수 있도록 설계해야 한다. 또한 사용자에게 즉각적인 반응을 보이도록 함으로써 사용자의 부담을 덜어주도록 한다. 이것은 대화형 시스템이라고 불리우는 것으로 현재 사용되는 대부분의 소프트웨어들이 이 방식을 채용하고 있다. 예를 들어 컴퓨터가 계산하는 동안 포인터의 모양을 시계의 형태로 바꾸어 보여주는 것이 그것이다.

시스템의 기능적 변화들은 인터페이스의 그래픽 요소들에 대하여 많은 새로운

것들을 요구한다. 네트워크를 통한 시스템의 광역화는 인터페이스가 만족시켜야 할 사용자의 범위를 예측할 수 없게 만들었으며 이로 인해 그래픽 요소들은 그 자체로써 존재하는 것이 아니라 텍스트 이상의 정보 전달 기능을 가지게 되었다.

컴퓨터는 무척이나 다양하고 복잡한 시스템이 그 특징으로 대변되며 사용자가 이를 구체적으로 이해한다는 것은 거의 불가능한 일이라고 할 수 있다. 그러나 인간은 이보다 훨씬 더 복잡한 사고체계를 갖추었으며 그 유형과 반응도 가지각색이다. 그러므로 컴퓨터와 인간의 대화를 유용하게 하는 효과적인 인터페이스를 이끌어내기 위해서는 컴퓨터의 테크놀로지를 이해하는 것 뿐 아니라 '인간'에 대한 이해가 무엇보다 중요하다고 할 수 있다.

2 절. 유용한 인터페이스디자인-Graphical User Interface Design

1. 사용자의 의도를 잘 반영해야 한다.

좋은 프로그램은 사용자가 의도하는 방식으로 작동된다. 이는 사용자가 데스크탑을 원하는 방식대로 배열하도록 하는 것을 의미한다. 또한 이미 하고 있는 작업에 근거해서 다음 작업을 예상할 수 있는 것을 의미하기도 한다.

2. 예측 가능하여야 한다.

예측 가능성에 대해서는 두 부분으로 나눌 수 있는데, 첫째는 세상에서 익힌 경험에 바탕을 둔 예측력이며 둘째는 어플리케이션의 내용에 기초한 예측력이다. 그래픽 환경에 있어 사용자는 마치 실제 사물을 보고 있는 것과 같기를 기대한다. 즉 예를 들어 살펴보자면 그리기 도구는 연필과 같은 아이콘으로, Zoom Tool 은 확대경으로 표시될 수 있을 것이다. 만약 Zoom Tool 이 Z 라는 글자모양의 아이콘으로, 또는 형편 없이 그려진 현미경의 아이콘으로 표시되어 각각의 아이콘에 대해 일일이 설명해야 할 정도라면 그래픽 환경에서의 작업의미를 잃은 것이라고 봐야 할 것이다.

좋은 GUI 디자이너는 현실에서 사물을 인식하는 방법도 반영하지만, 동시에 다른 어플리케이션 또한 그 세계의 일부라는 것을 인정해야 한다. 예를 들면 대부분

의 사용자들이 영역 채우기에 해당하는 Fill Tool 을 롤러브러쉬 아이콘으로 인식하는데 익숙해져 가고 있다. 그렇기 때문에 모든 어플리케이션에서 Fill Tool 을 표시할 때 롤러브러쉬를 이용하는 것이 적합하다. 비록 다른 아이콘이 롤러 브러쉬보다 이해하기 쉽다하더라도, 새로운 아이콘 표시는 사용자로 하여금 그 어플리케이션의 사용을 더욱 힘들게 만들 것이다. 좋은 GUI 어플리케이션은 이미 성공한 다른 것에 기초를 두어 구성되는 경우가 많다. 그러므로 사용자는 새로운 어플리케이션을 배울 때마다 그 사용방법을 다시 배울 필요가 없다. 즉 이미 하나의 GUI 어플리케이션을 배웠었다면, 다른 것도 매우 쉽게 이용할 수 있는 기초가 형성된 것이다.

GUI 는 '직관적'이라고 표현된다. 그러나 불행하게도 사용자 인터페이스의 모든 요소들이 직관적인 것이 아니다. 예를 들어 더블 클릭은 전혀 직관적이지 않다. 그럼에도 불구하고 많은 사람들이 'GUI 는 직관적이다'라고 말한다. 그 이유는 만약 GUI 에 대한 설명을 들으면 전에 이용해 본 적이 없는 어플리케이션을 쉽게 수용할 수 있게 되고 또한 새로운 상황 하에서도 무엇이 발생하리라는 것을 직관적으로 알 수 있기 때문이라는 데에 있다. 그리고 만약 사용하는 GUI 어플리케이션이 정말 직관적이라면 사용자의 직관적 추측은 정확히 적중하게 될 것이다.

3. 매력적으로 보이게 해야 한다.

만약 어플리케이션을 보기 좋게 만들 수 있다면 사람들은 그것을 쉽게 이용할 수 있을 것이다. 여백이나 텍스트 크기, 아이콘의 선택이나 배치의 문제들에 있어서 미적으로 보이게 하는데 관심을 기울일 필요가 있다.

4. 읽기 쉬워야 한다.

어플리케이션 대화상자와 도움말 파일이 읽기 쉽게 되어 있다면 더 많은 주의를 끌 것이다. 생소한 활자체와 특수용어를 피해야 한다. 생소한 폰트는 개발자의 창의적인 욕구를 만족시켜주는 반면 텍스트를 읽고 소화하는 것을 더 어렵게 만들 것이다. 예를 들어 대화상자에서 가장 많이 등장하는 OK, Cancel 버튼을 기존의 방식과는 다르게 나열하는 시도는 하지 않아야 한다. 비록 디자이너의 의도와는 거리가 멀고 '직관적'이지 못하더라도 사람들이 사용해 온 가장 대중적인 방식을 고

수하는 것이 좋다.

5. 모니터의 유형이나 해상도와는 독립적으로 작동되어야 한다.

좋은 GUI 어플리케이션은 640x480 과 1024x768 의 해상도에서 동일하게 작동된다.

6. 개인의 요구에 맞게 개별화할 수 있어야 한다.

각각의 사람들은 각기 다른 방식으로 일하는 습관을 가지고 있으므로 모든 사람의 방식이 수용된 어플리케이션을 디자인할 수는 없다. 그러나 어플리케이션은 미적으로나 구조적으로 융통성을 가져서 보다 많은 사용자의 편의를 도모해야 한다. 그렇다고 이용자에게 모든 작업을 맡기는 것도 바람직하지 못하다. 적어도 대부분이 편하게 사용할 수 있는 Default 는 제공하도록 해야 한다.

7. 좋은 GUI 는 사용자 인터페이스의 일관성을 요구한다.

좋은 GUI 는 일관된 작동방식이 요구된다. 또한 이 원칙은 GUI 뿐만 아니라 모든 사용자 인터페이스에 해당하는 가장 중요한 사항이다.

8. 산만해서는 안된다.

GUI 어플리케이션은 미관상 보기에 좋아야 할 뿐 아니라, 또한 적절히 감춰질 수도 있어야 한다. 즉 어플리케이션의 핵심적 부분은 대화상자를 열고, 컨트롤 버튼을 누르거나 메뉴를 읽는 것이 아니라는 것이다. 중요한 것은 약속한 작업을 제대로 해내는 것이다. GUI 디자이너의 목적은 보다 생산적으로 만드는데 있는 것이지 단순히 흥미를 유발하는 어플리케이션을 만들어내는 것은 아닌 것이다.

예를 들어 만화처럼 그려진 아이콘이 줄을 지어있는 File Open 대화상자가 있다고 하자. 이 대화상자는 처음에는 즐거움의 근원이 될 수 있으나, 조금만 지나면 말할 수 없이 성가신 것이 될 수도 있다.

9. 다양한 어플리케이션을 통합할 수 있어야 한다.

좋은 GUI 어플리케이션은 여러 종류의 다른 파일들을 지원할 수 있어야 한다. 예를 들면 OLE 기능을 사용하여 문서 파일 속에 그림이나 사운드를 삽입시키는 것이 가능해야 한다는 것이다.

10. 타인이나 단체의 저작권을 침해해서는 안된다.


뒤로
앞으로
중지
새로고침
시작
검색
즐겨찾기
인쇄
주소 http://wwwkle.seri.re.kr/
국어공학센터는 우리나라의 '국어 정보처리기술' 발전을 위하여 과학기술처와 문화체육부의 공동지원으로 1994년 11월 19일 시스템공학연구소내에 설립되었습니다.
귀하는 국어공학센터의 1220 번째 방문자이십니다.
제8회 한글 및 한국어 정보처리 학술대회에 보내주신 성원에 감사드립니다.
언어이해 연구실
Language Understanding Lab.
기계번역 연구실
Machine Translation Lab.
정보검색 연구실
Information Retrieval Lab.
안내 | 자료참고 | 연구과제 | 게시판
언어이해 | 기계번역 | 정보검색
방명록 작성 | 방명록 보기(부원만)
접속통계
w3master@wwwkle.seri.re.kr
안 내
자료창고
연구과제
게시판
국어공학

3 절. 타이포그라피 -Typography 적용

신 타이포그라피의 본질은 명쾌함을 나타내는 것이다. 신 타이포그라피는 명쾌함을 미의 기초로하며, 표현에서 최고의 최고의 경제성을 요구하는 현대인에게 명쾌함은 절대적 요소이다.또한 정보의 논리적 강조 가치관계에서 타이포그라피는 활자 크기나 굵기와 관련, 순서, 색채, 사진 등을 활용하여 국어공학정보의 정보흐름을 적용 명쾌한 표현을 주는 것이 가장 중요하다.

신 타이포그라피는 정보사용자에게 시선을 하나의 낱말, 그 낱말무리에서 다음 무리로 계속 유도할 수 있도록 텍스트를 디자인하는 것이다. 따라서 중요한 것은 크기의 차이, 강도, 공간에서의 위치, 색체등에 의하여 텍스트를 합리적으로 디자인하는 것이다

3 장. 추진전략 및 방법

구 분	연구 개발 목표	연구개발 내용 및 범위
1 차년도 (1996)	WWW DesignCD-ROM title 을 위한 Graphic DesignPoster Design	Audio, Video, Text, Image, 3DICON, Graphic, Cover Design통합국어 정보베이스 S/W 홍보포스터 Design
2 차년도		
3 차년도		

- 통합국어정보베이스 시스템은 국어공학센터의 연동성을 고려하여 기본 프레임은 국어공학센터의 프레임을 채택한다.

- 통합국어정보베이스 정보 분석

- 현재 구축된 통합정보베이스 WWW 의 시각 및 정보 커뮤니케이션의 효율성을 극대화하기위한 그래픽 디자인 및 인터페이스 디자인(아이콘 및 타이포그라피 등)

- 통합정보베이스 소프트웨어 CD-ROM 과 관련 CD Cover design

- 통합정보베이스 WWW design 기본방침

  - 한화면 (800X600)을 기준

  - 사용자의 정보흐름 및 흠미를 유발하는 디자인

  - www browser Explorer 및 Netscape 양쪽을 고려하여 디자인한다.

- 계층은 4 단계로 나눈다.

  - 계층 0 : 통합 국어정보 베이스 홈페이지 화면

  - 계층 1 : 대분류에 의한 화면

  - 계층 2 : 중분류에 의한 화면

  - 계층 3 : 세분류에 의한 화면

- 센터에서는 계층 0 부터 계층 2 까지 작성하고, 계층 3 에 대하여는 기본 방침만 설정한다.

- 계층 2 까지는 한화면 (800*600 기준)에 표시되도록 한다.

- 색상은 256 칼라를 기본으로 한다.

- 한화면의 아이콘 메뉴부분을 제외하고 그림/이미지 전체 영역이 텍스트 전체 영역보다 작아야 한다(예 : 이미지 최대 크기 640*270 이내).

- 하단의 메뉴는 다음과 같이 4 가지로 분류한다.

  - 안내(계층번호 1-1)

  - 자료창고(계층번호 1-2)

  - 연구과제(계층번호 1-3)

  - 게시판(계층번호 1-4)

  - mail(계층번호 1-5)

  - FAQ(계층번호 1-6)

- 통합 국어 정보베이스(계층번호 1-7)

• 좋은 그림은 한국적이고 한글공학과 관련된 그림을 삽입한다.

• 분류는 중과제 이상으로 과제예산 규모가 일정 수준 이상의 연구과제를 중심으로 분류한다.

- STEP2000

  • 한글 언어처리 기반 기술 (1-3-1)

  • 통합 국어 정보베이스(1-3-2)

  • 지능형 처리기 개발(1-3-3)

- 기계번역(구축중 표시)

• 통합 정보 베이스 분야중 기본적인 사항만 국어공학센터와 통일 시킨다.

- 연구책임자 : 성명 (전자우편 주소)

- 개요 (통합국어정보베이스에 대한 개요 - 1 쪽 정도)

- 발표자료 (중간 발표, 최종발표 자료 및 시연 자료)

- 논문

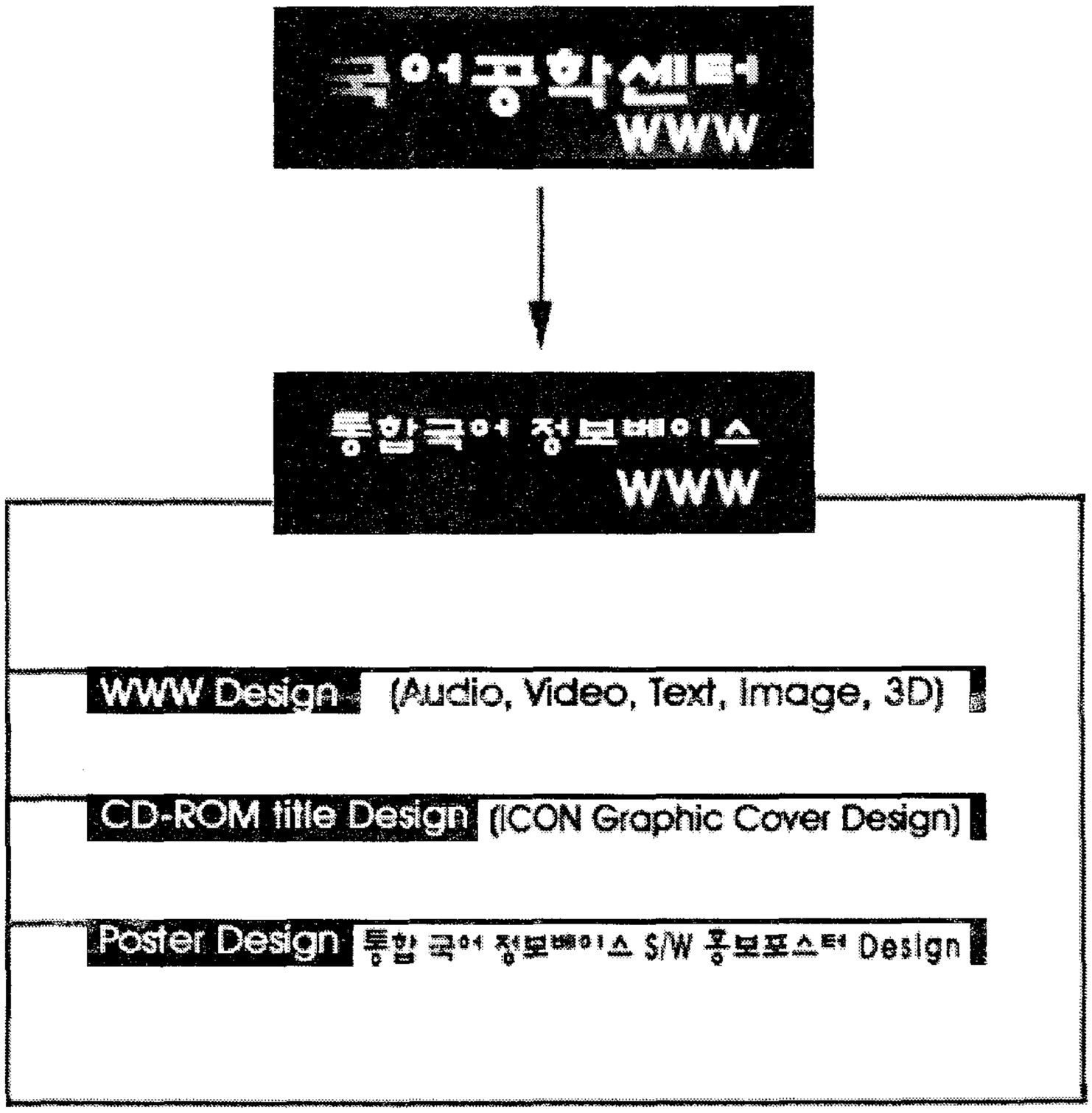
- 협력기관 연구

  • 위탁과제명 1(해당 연구실의 본사업과 관련된 디렉토리로 연결)

  • 위탁과제명 2

- 과제에 대한 의견, 평가 등을 듣는 곳입니다.

4장. 연구개발 추진체계



5장. 기대성과

- 관련 연구단체의 표준제시
- 국어공학센터의 연동으로 인한 명실상부한 국어정보 WWW 의 정보 수출입 창구역할
- 우리나라 문화환경에 효율적인 WWW Design 제시

6 장. 활용방안

- Internet 을 통한 통합국어정보베이스 정보를 검색 및 활용

- 관련 연구단체의 정보제공

- 통합국어정보베이스 소프트웨어의 확장

7 장. 연구평가의 척도

- 사용자 인터페이스 고려하여 정보의 제공 및 활용률 측정

- 효율적인 편집디자인

- 효율적인 시각디자인

- CD-ROM Cover Design 및 그래픽 디자인의 우수성

통합국어정보베이스 WWW Design 구조도

Home Page

초보자

기초국어정보베이스

한국어말뭉치 : 기초말뭉치, 태깅된말뭉치, 구문트리, 범주화된 말뭉치, 문형자료 모음
음성자료모음
글자자료모음 : KLE DB Homepage, 글자단위 검색

국어정보처리도구

사전개발 및 관리시스템
태거
구문트리태거
한영정렬시스템
문서구조표현을 위한 표준화
한국어 입출력 표준환경
균형화 코퍼스 구축표준방법론
품사사전 규칙과 시범패키지

용어사전

전문용어사전
분류체계에 기반한 대역어사전
형태소 분석사전 및 사전편집기

숙련자

기초국어정보베이스

한국어말뭉치 : 기초말뭉치, 태깅된말뭉치, 구문트리, 범주화된 말뭉치, 문형자료 모음
음성자료모음
글자자료모음 : KLE DB Homepage, 글자단위 검색

국어정보처리도구

사전개발 및 관리시스템
태거
구문트리태거
한영정렬시스템
문서구조표현을 위한 표준화
한국어 입출력 표준환경
균형화 코퍼스 구축표준방법론
품사사전 규칙과 시범패키지

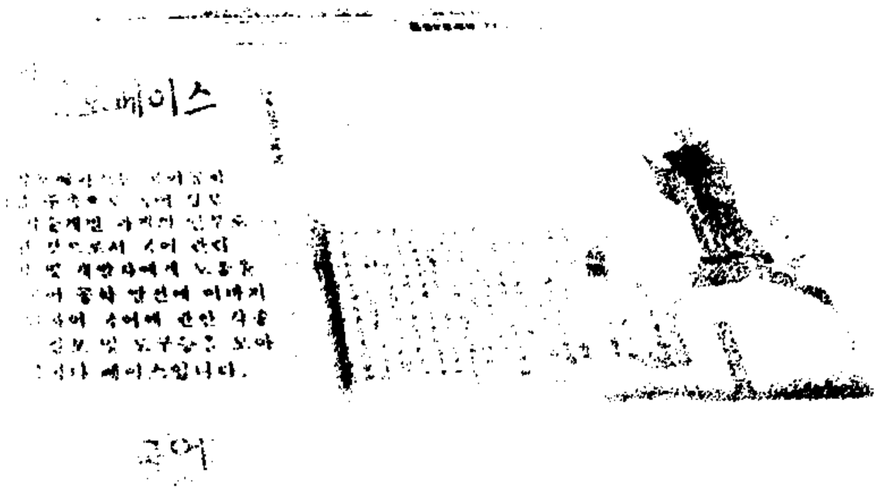
용어사전

전문용어사전
분류체계에 기반한 대역어사전
형태소 분석사전 및 사전편집기

구조도

도움말

등 록



통합국어정보베이스

홈 페 이 지

파일(F) 편집(E) 보기(V) 이동(G) 즐겨찾기(A) 도움말(H)

뒤로 앞으로 중지 새로고침 시작 검색 즐겨찾기 인쇄 글꼴 메일 편집

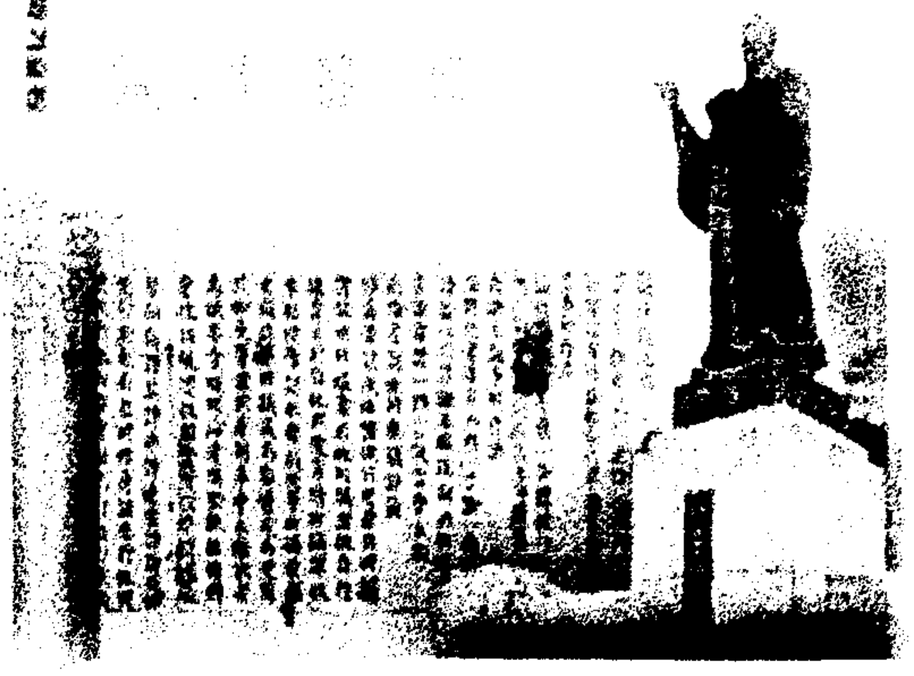
주소 C:₩work₩test3.htm 빠른 연결

KAIST 구조도 도움말

Korean Information Base System

국어 정보베이스

국어정보베이스는 국어공학 센터를 주축으로 국어 정보 처리 기술개발 과제의 일부로 추진된 것으로서 국어 관련 연구자 및 개발자에게 도움을 주고 국어 공학 발전에 이바지 하기 위하여 국어에 관한 각종 자료와 정보 및 도구등을 모아 놓은 데이타 베이스입니다.



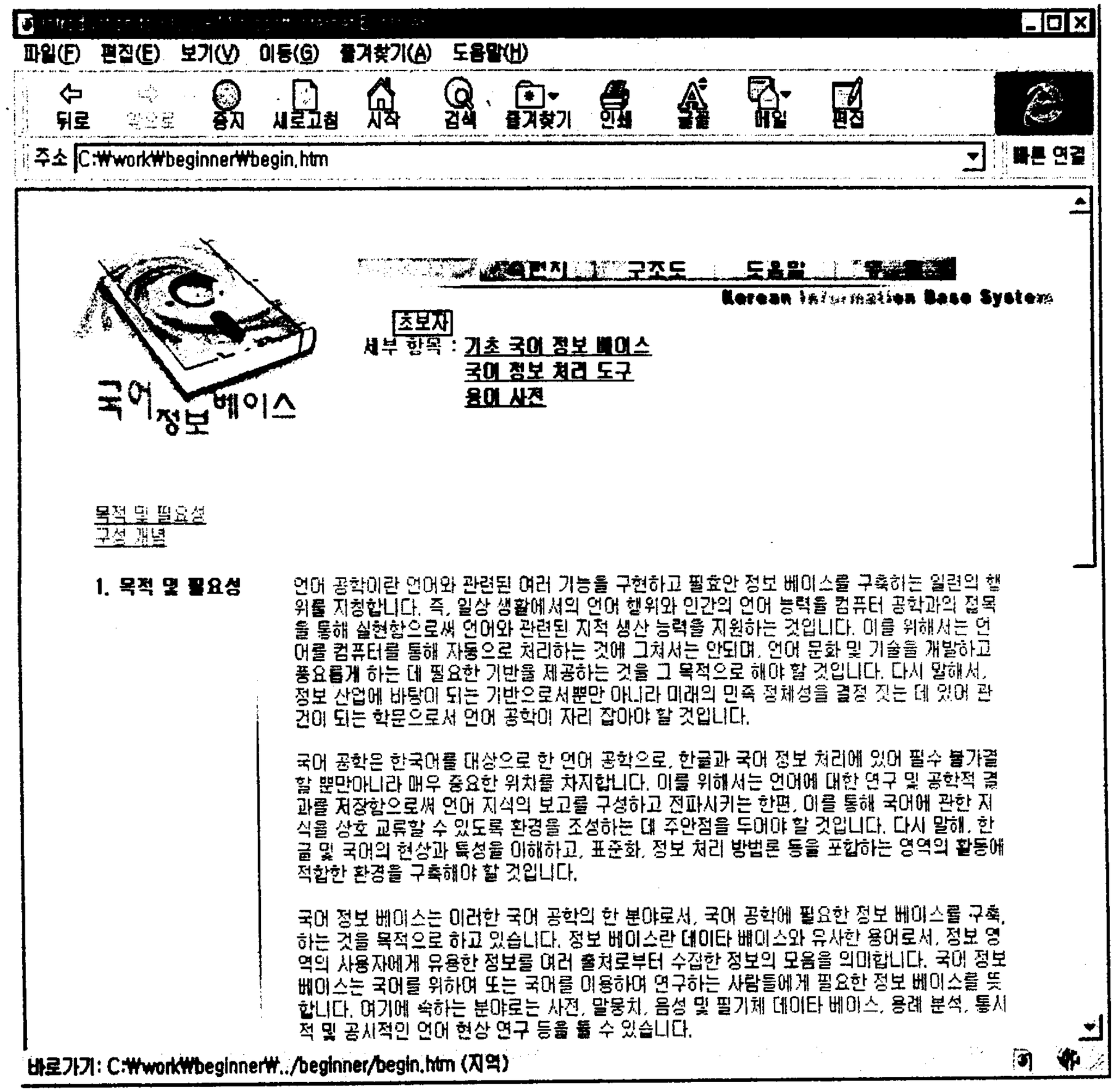
국어 공학 센터

연구책임자 : 최기선
e-mail : kschoi@cs.kaist.ac.kr

완료

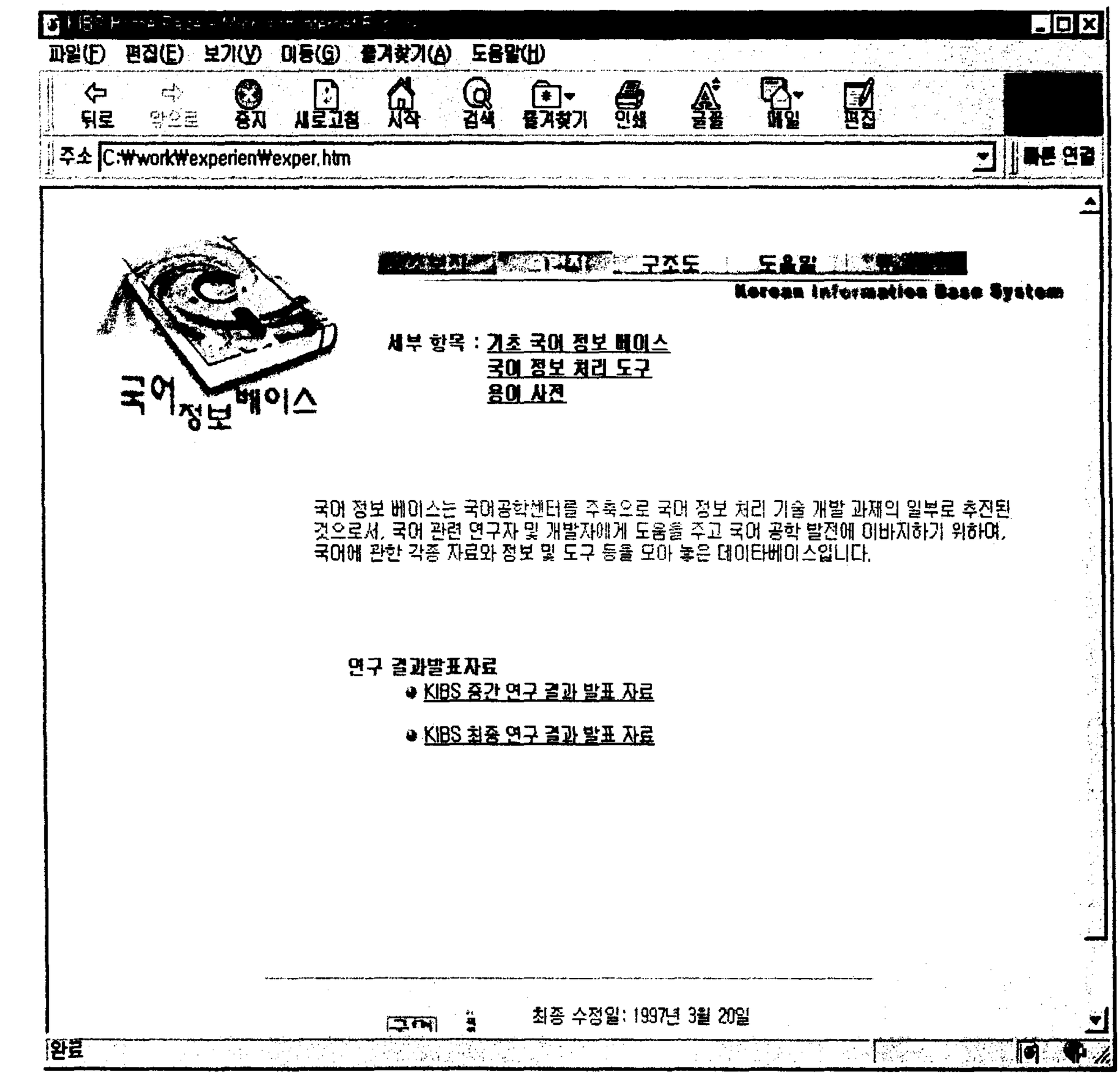
통합국어정보베이스

초 보 자



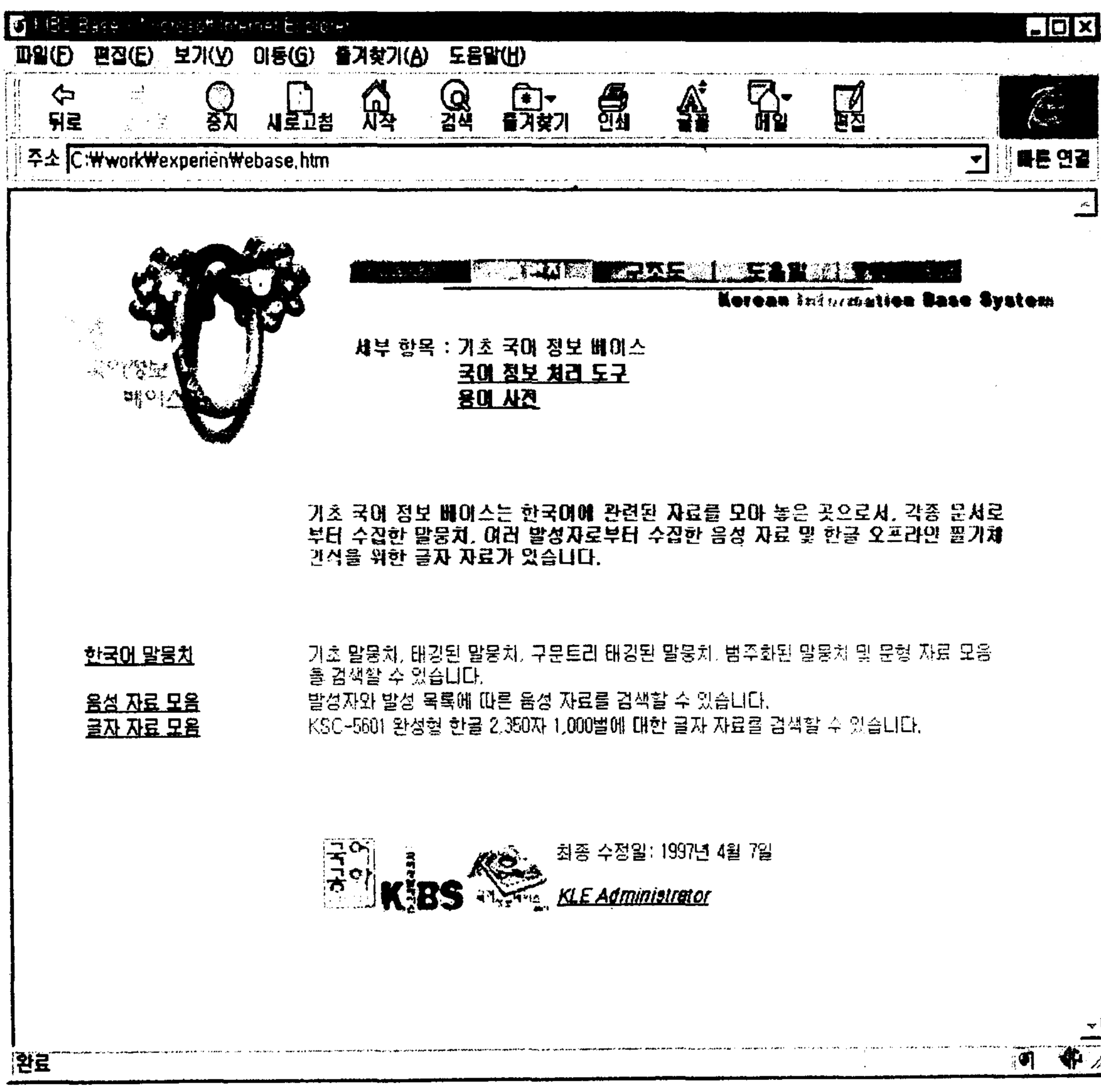
통합국어정보베이스

숙 련 자



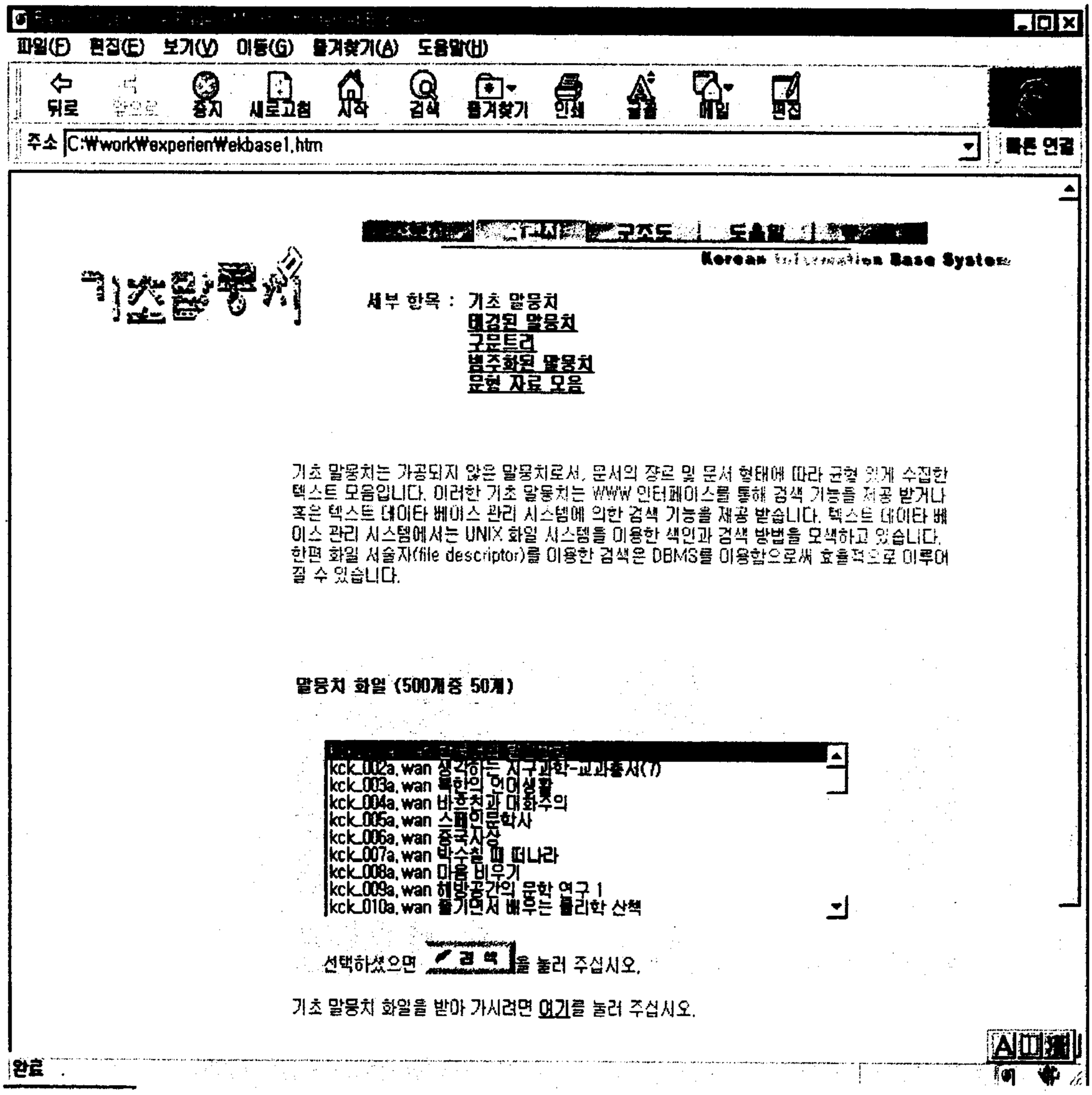
통합국어정보베이스

숫련자 기초국어 정보베이스



통합국어정보베이스

숙견자 기초말뭉치



통합국어정보베이스

구 조 도

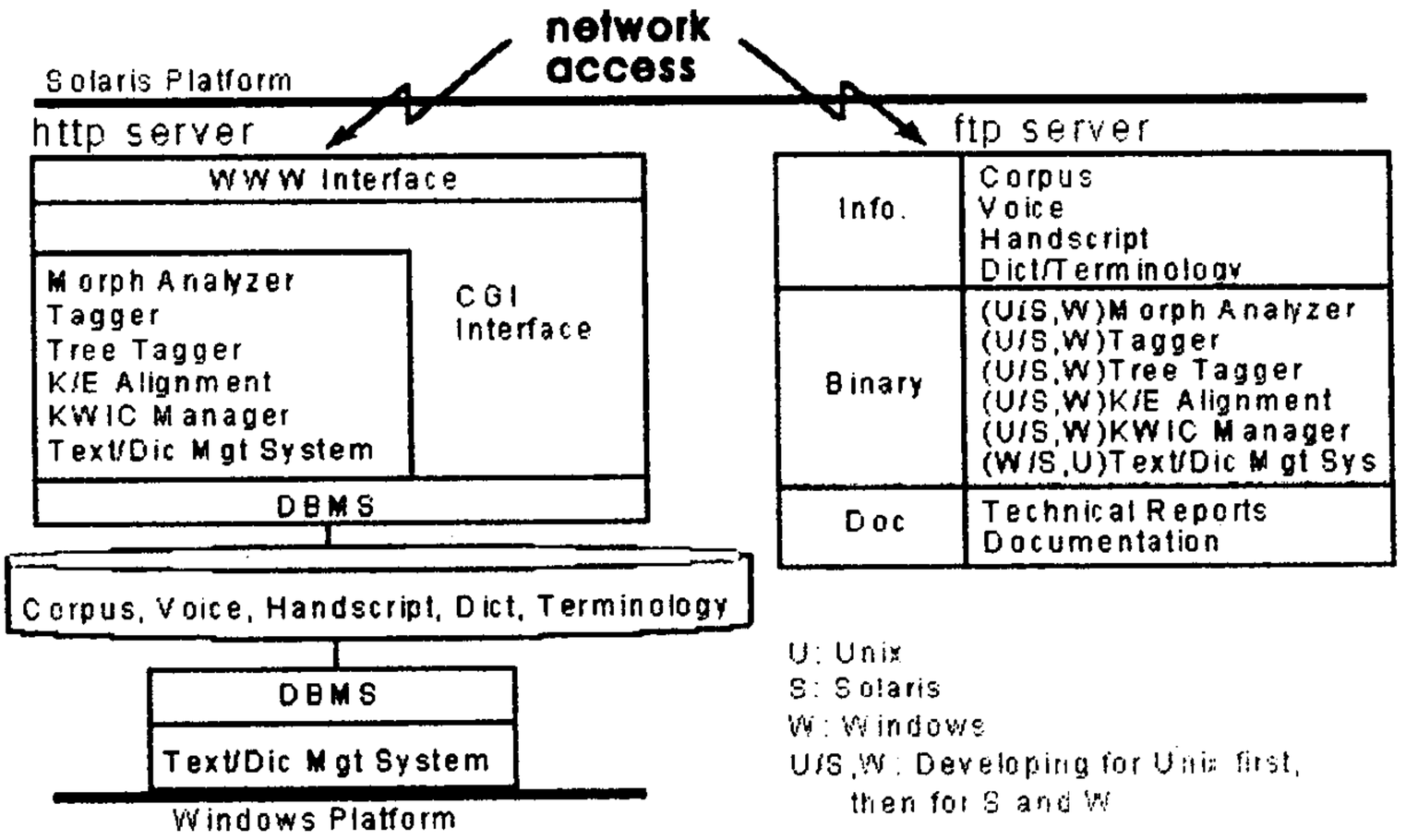
파일(F) 편집(E) 보기(V) 이동(G) 즐겨찾기(A) 도움말(H)

뒤로 중지 새로고침 시작 검색 즐겨찾기 인쇄 글꼴 메일 편집

주소 C:₩work₩structure.htm 빠른 연결

Korean Information Base System

국어 정보 베이스 구조 이 페이지에서는 클릭을 통해서 KIBS의 각 구조로 이동을 할 수 있습니다.



최종 수정일: 1997년 4월 7일

KLE Administrator

완료

통합국어정보베이스

도 움 말

파일(F) 편집(E) 보기(V) 이동(G) 즐겨찾기(A) 도움말(H)

뒤로 앞으로 중지 새로고침 시작 검색 즐겨찾기 인쇄 글꼴 메일 편집

주소 C:₩work₩help.htm

Korean Information Base System

세부 항목 : 국어정보베이스 웹 스페이스 구조
기능 페이지
설명 페이지
당부 사항

1. 국어정보베이스 웹 스페이스 구조

이 페이지에서는 국어정보베이스(KIBS) 시스템의 페이지 구성에 관하여 설명합니다. 이 설명은 처음 사용하는 사람들이 효과적으로 사용할 수 있도록 하기 위하여 쓰여졌습니다.

이 페이지에서 모두 설명되는 내용인, KIBS의 웹 스페이스를 간단하고 효과적으로 구성하기 위해 처음으로 씌여진 **화이트 페이퍼**를 보실 수 있습니다.

1.1 기본적인 페이지 구성

기본적으로 페이지들은 보통 두 가지로 나뉩니다. 하나는 아이콘을 선택해서 각 세부기능을 선택할 수 있는 페이지이고, 다른 하나는 초보자들이나 설명을 원하는 경험자들을 위한 기능 설명 페이지들입니다. 편의상 처음의 페이지들을 **기능 페이지**, 설명을 위한 페이지를 **설명 페이지**라고 부릅니다.

기능/설명 페이지의 구분 외에 각 레벨을 구분하기 위하여 다른 색깔을 사용합니다.

0 레벨 (KIBS 홈페이지) : 엷은 파랑
1 레벨 : 엷은 초록
2 레벨 : 엷은 노랑
3 레벨 : 엷은 빨강

2. 기능 페이지

기능 페이지는 왼쪽에 위치한 진한 색깔의 바아(bar)로 구분을 할 수 있습니다. 이 바아에는 각각의 기능이나 설명 페이지로 넘어가기 위한 아이콘이 있습니다. 바아의 가장 위쪽에는 그 페이지를 나타내는 아이콘이 있습니다. 그 다음으로 각각의 기능 페이지를 설명하고 있는 페이지로 이동할 수 있는 아이콘이 나옵니다.

몸체에 해당하는 부분에는 다른 세부 기능으로 이동할 수 있는 아이콘이 존재하고, 마지막에는 KIBS 홈페이지나 국어정보베이스의 홈페이지 또는 바로 위 레벨의 기능 페이지로 이동하기 위

완료

통합국어정보베이스

등 록

파일(F) 편집(E) 보기(V) 이동(G) 즐겨찾기(A) 도움말(H)

뒤로 앞으로 중지 새로고침 시작 검색 즐겨찾기 인쇄 글꼴 메일 편집

주소 C:₩work₩register.htm

빠른 연결

Korean Information Base System

1. 등록을 하면

KIBS에서는 여러가지 소스나 데이타들을 다운로드 받으실 수 있습니다. 이러한 소스나 데이타들의 다운로드는 관리자에 의해서 인정된 분들에게만 허용되고 있습니다.

등록을 하시면 관리자에게 판단에 의하여, 사용자의 레벨이 주어지고, 사용할 수 있는 계정과 암호가 주어지게 됩니다. 이러한 결정 사항은 전자우편으로 알려드립니다. 또한 등록 정보는 KIBS의 사용 통계를 내는데에도 사용됩니다.

2. 등록

[주의] 주민등록번호, 이름, 전자우편은 반드시 입력해야 합니다.

한글이름 (영어이름) :
예 : 홍길동 (Hong Gil Dong)
전자 우편 :
전화번호 :
예 : 042-821-7777
주민등록번호 :
예 : 700519-1400811
소속 :

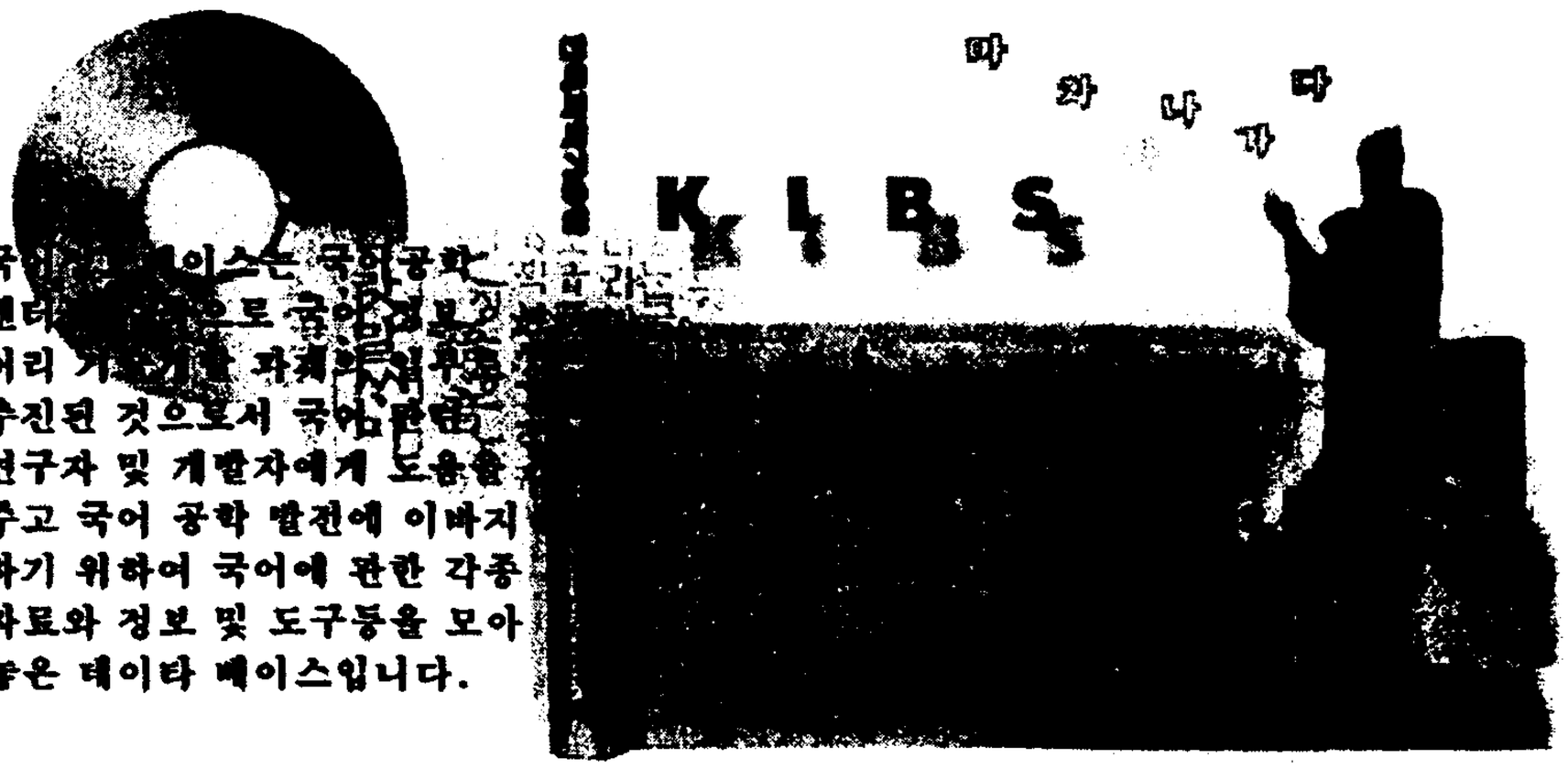
확인 취소

최종 수정일: 1997년 4월 7일

KLE Administrator

KAIST

Korean Information Base System



국어정보베이스는 국어공학센터에서 국어정보처리 기술 과제의 일환으로 추진된 것으로서 국어 관련 연구자 및 개발자에게 도움을 주고 국어 공학 발전에 이바지하기 위하여 국어에 관한 각종 자료와 정보 및 도구등을 모아 놓은 데이타 베이스입니다.

의견이 있으시면 전자우편으로 연락하여 주시면 감사하겠습니다.

본 과제는 STEP 2000 국어정보처리기술 개발과제의 일부로 국어공학센터와 협력으로 수행하고 있습니다.

연구책임자 : 최기선
e-mail : kschoi@cs.kaist.ac.kr

세부 항목 : 기초 국어 정보 베이스
국어 정보 처리 도구
용어 사전

목적 및 필요성
구성 개념

1. 목적 및 필요성

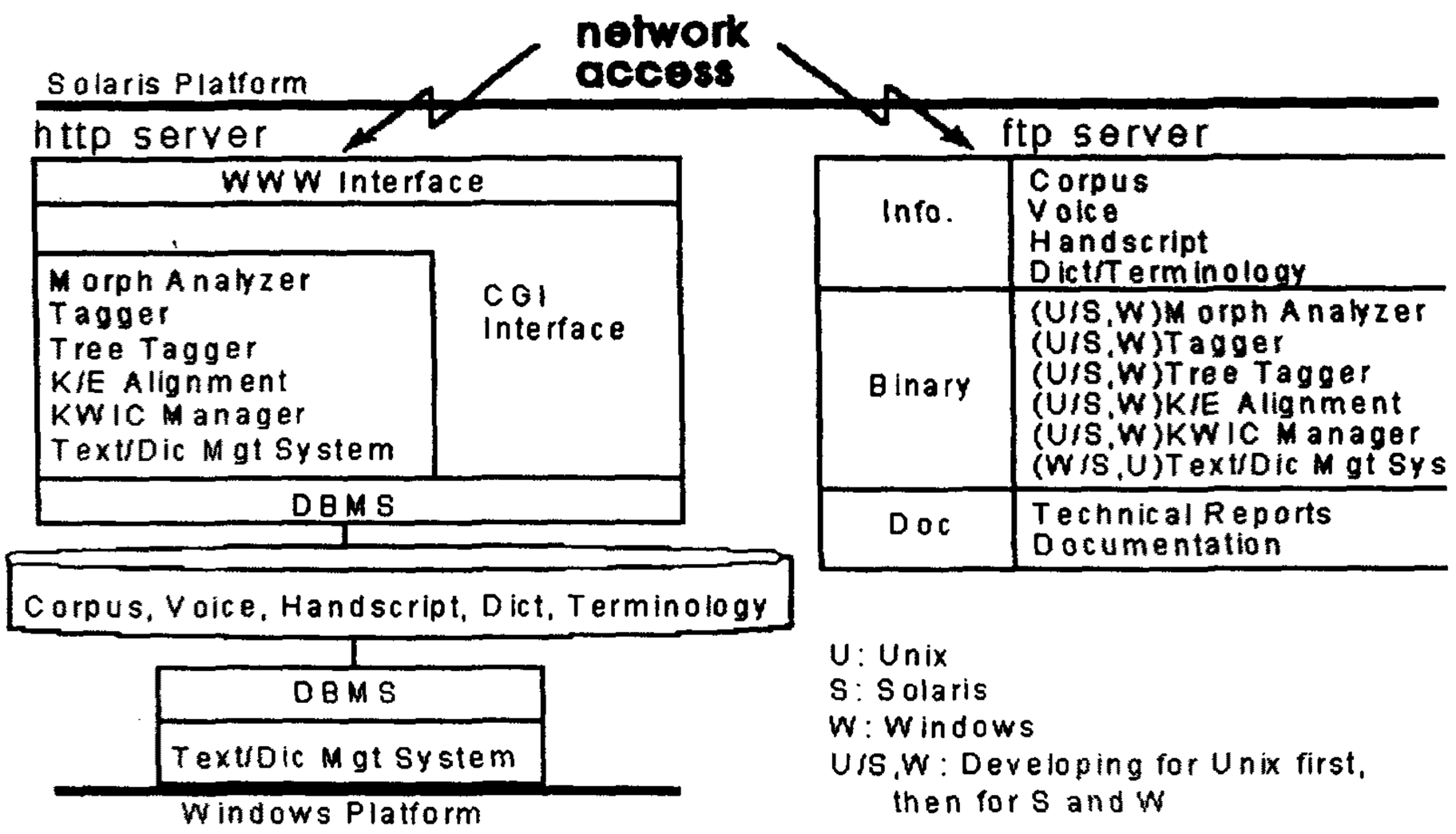
언어 공학이란 언어와 관련된 여러 기능을 구현하고 필요한 정보 베이스를 구축하는 일련의 위를 지칭합니다. 즉, 일상 생활에서의 언어 행위와 인간의 언어 능력을 컴퓨터 공학과의 접 을 통해 실현함으로써 언어와 관련된 지적 생산 능력을 지원하는 것입니다. 이를 위해서는 언어를 컴퓨터를 통해 자동으로 처리하는 것에 그쳐서는 안되며, 언어 문화 및 기술을 개발하고 풍요롭게 하는 데 필요한 기반을 제공하는 것을 그 목적으로 해야 할 것입니다. 다시 말해서, 정보 산업에 바탕이 되는 기반으로서뿐만 아니라 미래의 민족 정체성을 결정 짓는 데 있어 관건이 되는 학문으로서 언어 공학이 자리 잡아야 할 것입니다.

국어 공학은 한국어를 대상으로 한 언어 공학으로, 한글과 국어 정보 처리에 있어 필수 불가결 할 뿐만아니라 매우 중요한 위치를 차지합니다. 이를 위해서는 언어에 대한 연구 및 공학적 결과를 저장함으로써 언어 지식의 보고를 구성하고 전파시키는 한편, 이를 통해 국어에 관한 지식을 상호 교류할 수 있도록 환경을 조성하는 데 주안점을 두어야 할 것입니다. 다시 말해, 한글 및 국어의 현상과 특성을 이해하고, 표준화, 정보 처리 방법론 등을 포함하는 영역의 활동 적합한 환경을 구축해야 할 것입니다.

국어 정보 베이스는 이러한 국어 공학의 한 분야로서, 국어 공학에 필요한 정보 베이스를 구축 하는 것을 목적으로 하고 있습니다. 정보 베이스란 데이타 베이스와 유사한 용어로서, 정보 영역의 사용자에게 유용한 정보를 여러 출처로부터 수집한 정보의 모음을 의미합니다. 국어 정보 베이스는 국어를 위하여 또는 국어를 이용하여 연구하는 사람들에게 필요한 정보 베이스를 의미합니다. 여기에 속하는 분야로는 사전, 말뭉치, 음성 및 필기체 데이타 베이스, 용례 분석, 통계적 및 공시적인 언어 현상 연구 등을 들 수 있습니다.

국어 정보 베이스를 효과적으로 구축하기 위해서는 각 분야의 정보가 통합적으로 관리 및 유지될 수 있는 환경을 우선적으로 구축하는 것이 필요합니다. 따라서 여기에서는 이러한 구성 작업을 수행함으로써 일원화된 정보의 제공과 관리 및 유지의 편의성을 확보하는 데 목적을 두고 있습니다.

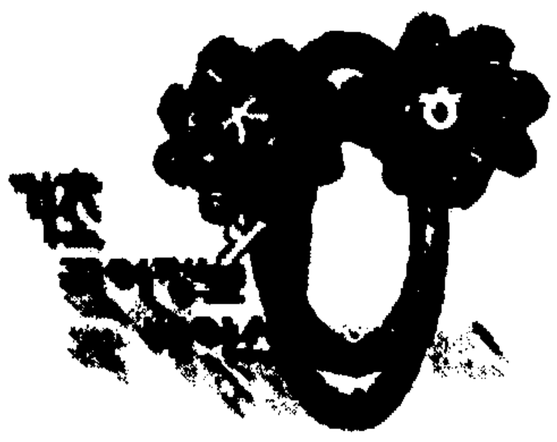
2. 구성 개념



이 그림은 국어 정보 베이스의 통합 구성도를 개념적으로 나타낸 것입니다. 국어 정보 베이스를 구현한 국어 정보 기반 (Informaiton Platform) 체계는 http server와 ftp server로 나누어지고, http server는 다시 기초 국어 정보 베이스와 상위 계층의 유틸리티 및 도구들로 나누어지는데, 이 두 가지는 데이타 베이스 관리 체계와 화일 시스템을 통해 서로 연결됩니다. Http server는 internet을 통한 사용자의 접근을 허용하며, 텍스트 외에도 음성 및 이미지 등을 제공하는 기능을 가지고 있고, ftp server는 유용한 프로그램 코드나 정보를 통신망을 통해 쉽게 전송해 줄 수 있습니다.

최종 수정일: 1997년 4월 7일

KLE Administrator



조보지

Korean Information Base I

세부 항목 : 한국어 말뭉치
음성 자료 모음
글자 자료 모음

개요

한국어 말뭉치는 현대 한국어 문서를 대상으로 문서 처리 시스템에서 사용하려는 목적에 맞기 여러가지의 말뭉치를 모아 놓은 것입니다. 이러한 말뭉치들은 언어 공학자에게 중요한 연구 료가 될 뿐만아니라 통계적 모델의 수립이나 문법과 관련된 제 이론의 개발, 음성에서의 운율 적인 현상 탐구 또는 분석 모델의 적합성 평가 및 비교 등에 있어 매우 유용합니다.

음성 자료 모음은 단어 및 문장에 대한 음성 자료를 파형 형태로 데이타 베이스화한 것입니다 현재 음성적으로 균형 있는 단어 집합 (Phonetically Balanced Words : PBW)과 함께 단일 절, 4연 숫자, 그리고 몇 가지 이야기문에 대해 4명의 표준 발음을 파형 형태로 모아 놓은 상 이며, 앞으로 각 음성 데이타를 DBMS의 관리하에 두도록 하는 한편 CD-ROM화하는 방안 추진하고 있습니다.

글자 자료 모음은 KSC-5601 (1987)에 있는 2,350자를 대상으로 1차년도에는 520자, 2차 도에는 990자, 그리고 3차년도에는 2,350자 전체에 대해 각 1,000벌의 정자체 및 자유필체 료를 수집, 300 dpi로 저장할 계획입니다. 이러한 자료는 필기 인식 연구를 위한 분야에 주로 쓰이기 마련이며 또한 자료의 검색도 특정 필체의 특정 벌 수보다는 대용량으로 이루어지기 련이므로, 검색 인터페이스를 구현할 때 그 방법상의 문제보다는 무엇을 검색해야 하는가에 한 결정이 더 중요합니다. 현재 기본적인 예제 자료의 검색 기능이 제공되고 있으며 검색 방법 론에 대한 세부 사항이 해당 목적 및 필요에 맞게 결정된 후 세부 검색 기능이 제공될 예정입 다.

최종 수정일: 1997년 3월 20일

KLE Administrator

초보자

Korean Information Ba

세부 항목 : 기초 말뭉치
태깅된 말뭉치
구문트리
범주화된 말뭉치
문형 자료 모음

기초 말뭉치의 구성
텍스트 프로필
인코딩 원형

기초 말뭉치에 대한 설명은 다음에 제공될 예정입니다.

텍스트 프로필에 대한 설명은 다음에 제공될 예정입니다.

1.기초 말뭉치의 구성
2.텍스트 프로필
3.엔코딩 원형

인코딩 원칙에 대한 설명은 다음에 제공될 예정입니다.

최종 수정일: 1997년 4월 7일

KLE Administrator

1.서론

말이나 글로서 발화된 언어 자료, 즉 코퍼스(corpus)는 언어연구의 중요한 기반이 되고 있다 특히, 실제 발화된 각 단어에 대하여 품사 정보가 부착된 코퍼스는 단어인식, 단어생성, 음성 식, 음성합성, 문자인식, 정보검색, 사전구축 등과 같은 언어정보처리 연구에 아주 중요한 기 자료로 쓰일 수 있다.

2.태그 셋 설정의 필요성및 원칙

국어정보베이스에서는 한국어 처리는 물론 한국어에 관심있는 모든 연구자들이 두루 사용할 있도록 형태, 통사 태그에 대한 분류 기준이 명확히 제시될 필요가 있다. 품사태그 집합의 표 화 작업은 다음과 같은 원칙에 바탕을 두어 진행되었다.

1. 본 품사태그 집합의 설정은 기본적으로 한국어를 대상으로 하나, 한국어 운장에서 두 사용되는 외국어나 특수기호들도 대상으로 삼는다.
2. 또 품사태그 집합의 설정은 학교문법을 최대한 반영하며, 통사론적 분석 위주보다는 형태론적 분석 위주로 한다.
3. 품사태그는 여러 분야(형채소 레벨과 통사적 레벨)에서 다양한 용도로 사용할 수 있도 록 계층적으로 분류한다.

이와 같은 원칙에 입각하여 다음과 같은 한국어 품사태그 집합을 설정하였다.

3.태그 셋 설정의 실제

가. 기호(s)

1) sp(쉼표, pause)
예) ,(쉼표),:(쌍점),/(빗금) 등
2) sf(마침표, full stop)
예) .(마침표), !(느낌표), ?(물음표) 등
3) sl(여는 따옴표 및 묶음표, left quotation and parenthesis mark)
예) "(여는 큰 따옴표), '(여는 작은 따옴표), ((여는 소괄호), { (여는 중괄호), [(여는 대괄호) 등
4) sr(닫는 따옴표 및 묶음표, right quotation and parenthesis mark)
예) "(닫는 큰 따옴표), '(닫는 작은 따옴표),) (닫는 소괄호), } (닫는 중괄호),] (닫는 대괄호) 등
5) sd(이음표, dash)
예) -(붙임표), ~(물결표) 등
6) se(줄임표, ellipsis)
예) XX, OO(숨김표), ……(줄임표) 등
7) su(단위기호, unit)
예) m, cm, mm, ft, yd, g, kg(단위를 표현하는 기호들) 등
8) sy(기타 기호, other symbols)
예) +, -, x 등

나. 외국어(f)

1) f(외국어, foreign word)
예) Esperanto, 아임 쏘리 등

다. 체언(n)

1) nc(보통명사)

ㄱ) ncp(서술성 명사)

- ncpa(동작성 명사, active-predicative common noun)
예) 가공, 가담, 가정, 계약, 문의, 박탈, 방송, 병행 등
- ncps(상태성 명사, stative-predicative common noun)
예) 가난, 가능, 고상, 동일, 마땅, 만족 등

ㄴ) ncn(비서술성 명사)

- ncn(비서술성 명사, non-predicative common noun)
예) 의자, 책상, 나무, 가랑비, 가마솥, 미움, 정신 등

2) nq(고유명사)

ㄱ) nq(고유 명사, proper noun)
예) 경기, 고성, 동남, 서귀포, 삼국유사 등

3) nb(의존명사)

ㄱ) nbu(단위성 의존명사, unit bound noun)
예) ~ 분, ~ 개, ~ 그루, ~ 원, ~ 마리 등
ㄴ) nbn(비단위성 의존명사, non-unit bound noun)
예) 것, 나위, 데, 듯, 둥, 뿐, 수, 양, 척, 체, 터 등

4) np(대명사)

ㄱ) npp(인칭 대명사, personal pronoun)
예) 나, 우리, 저, 너, 누구, 당신 등
ㄴ) npd(지시 대명사, demonstrative pronoun)
예) 이것, 그것, 여기, 저기, 이때, 이쪽, 무엇, 어디 등

5) nn(수사)

ㄱ) nnc(양수사, cardinal numerals)
예) 하나, 둘, 셋, 한둘, 일, 이, 삼 등
ㄴ) nno(서수사, ordinal numerals)
예) 첫째, 둘째, 셋째, 제일, 제이 등

라. 용언(p)

1) pv(동사)

ㄱ) pvd(지시 동사, demonstrative verb)
예) 저러다, 고러다, 요러다, 조러다 등
ㄴ) pvg(일반 동사, general verb)
예) 가다, 건너다, 눕다, 늙다, 닮다, 뜨다, 마시다 등

2) pa(형용사)

ㄱ) pad(지시 형용사, demonstrative adjective)
예) 그렇다, 아무렇다, 어떠하다, 이렇다, 저렇다, 조렇다 등
ㄴ) paa(성상 형용사, attributive adjective)
예) 꼼꼼하다, 기쁘다, 깨끗하다, 맑다, 좋다, 크다 등

3) px(보조용언)

ㄱ) px(보조용언, auxiliary verb)
예) ~ 되다, ~ 만들다, ~ 않다 등

마. 수식언(m)

1) mm(관형사)

ㄱ) mmd(지시관형사, demonstrative adnoun)
예) 그, 그런, 무슨, 본, 이, 이런, 저, 저런 등
ㄴ) mma(성상관형사, attributive adnoun)
예) 새, 옛, 오른, 왼, 헌 등

2) ma(부사)

ㄱ) mad(지시부사, demonstrative adverb)
예) 이리, 그리, 요리, 고리, 여기, 저기, 어찌, 아무리 등
ㄴ) maj(접속부사, conjunctive adverb)
예) 또, 또는, 곧, 및, 혹은, 따라서, 한편 등
ㄷ) mag(일반부사, general adverb)
예) 가령, 갑자기, 거의, 결코, 글쎄, 기어이, 흡사 등
바. 독립언(i)

1) ii(감탄사)

ㄱ) ii(감탄사, interjection)
예) 그래, 아이구, 아, 암, 여보, 응, 저 등
사. 관계언(j)

1) jc(격조사)

ㄱ) jcs(주격조사, subjective case particle)
예) -이, -가, -에서, -서 등
ㄴ) jco(목적격조사, objective case particle)
예) -을, -를 등
ㄷ) jcc(보격조사, complemental case particle)
예) -이, -가 등
ㄹ) jcm(관형격조사, Adnominal case particle)
예) -의 등
ㅁ) jcv(호격조사, vocative case particle)
예) -야, -아, -여, -시여, -이여 등
ㅂ) jca(부사격조사, adverbial case particle)
예) -에게, -같이, -한테, -만큼, -라고, -이라 등
ㅅ) jcj(접속격조사, conjunctive case particle)
예) -과, -다, -랑, -와, -하고 등
ㅇ) jct(공동격조사,comitative case particle)
예) -과, -와, -하고 등
ㅈ) jcq(인용격조사,quotative case particle)
예) -라고, -고 등

2) jx(보조사)

ㄱ) jxc(통용보조사, common auxiliary particle)
예) -곧, -까지, -나, -은, -조차 등
ㄴ) jxf(종결보조사, final auxiliary particle)
예) -마는, -그려, -요 등

3) jp(서술격조사)

ㄱ) jp(서술격조사, predicative particle)
예) -이- 등
아. 어미(e)

1) ep(선어말어미)

ㄱ) ep(선어말어미, prefinal ending)
예) -겠-(미래), -더-(회상), -시-(높임), -었-(과거) 등

2) ec(연결어미)

ㄱ) ecc(대등적 연결어미, coordinate conjunctive ending)
예) -거나, -거니, -고, -느니, -면서 등
ㄴ) ecs(종속적 연결어미, subordinate conjunctive ending)
예) -거든, -거늘, -ㄴ데, -지만은 등
ㄷ) ecx(보조적 연결어미, auxiliary conjunctive ending)
예) -게, -고, -아, -지 등

3) et(전성어미)

ㄱ) etn(명사형어미, nominalizing ending)
예) -기, -ㅁ, -음 등
ㄴ) etm(관형사형어미, adnominalizing ending)
예) 는, (으)ㄴ(과거), ㄹ(미래) 등

4) ef(종결어미)

ㄱ) ef(종결어미, final ending)
예) -게(명령형), -구나(감탄형), -으십시오(청유형) 등

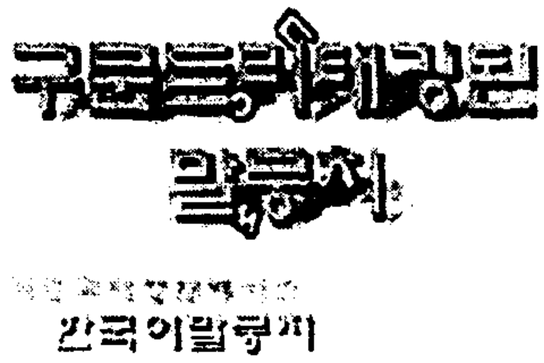
자. 접사(x)

1) xp(접두사)

ㄱ) xp(접두사, prefix)
예) 갖, 개, 외 등

2) xs(접미사)

ㄱ) xsn(명사파생접미사, noun-derivational suffix)
예) -군, -님, -들, -기 등
ㄴ) xsv(동사파생접미사, verb-derivational suffix)
예) -하-, -되-, -시키- 등
ㄷ) xsm(형용사파생접미사, adjective-derivational suffix)
예) -하-, -답-, -롭- 등
ㄹ) xsa(부사파생접미사, adverb-derivational suffix)
예) - 히 등



세부 항목 : 기초 말뭉치
태깅된 말뭉치
구문트리
범주화된 말뭉치
운형 자료 모음

구문트리 태깅(bracketting)의 원칙
기초 통계 자료

구문트리 태깅(bracketting)의 원칙은 다음에 제공될 예정입니다.

1.구문트리 태깅 (bracketting)의 원칙
2.기초 통계 자료

기초 통계 자료에 대한 설명은 다음에 제공될 예정입니다.

최종 수정일: 1997년 4월 7일

KLE Administrator


조보자
Korean Information Base
세부 항목 : 기초 말뭉치
태깅된 말뭉치
구문트리
범주화된 말뭉치
운형 자료 모음
하위 범주 집합
하위 범주화 규칙
기초 통계 자료
하위 범주 집합에 대한 설명은 다음에 제공될 예정입니다.
.하위 범주화 규칙에 대한 설명은 다음에 제공될 예정입니다.
1.하위 범주 집합
2.하위 범주화 규칙
3.기초 통계 자료
기초 통계 자료에 대한 설명은 다음에 제공될 예정입니다.
최종 수정일: 1997년 4월 7일
KLE Administrator

세부 항목 : 한국어 말뭉치
음성 자료 모음
글자 자료 모음

음성 자료 모음의 개요
음성 자료 모음의 필요성
음성 자료 모음의 요구사항
음성 입출력의 특징
시험판 음성 자료 모음의 화일 구조

1.음성 자료 모음의 개요 음성 연구에 있어서 음성 자료는 필수적이다. 이 음성 자료는 다종 다양한 것이 필요하다. 지까지는 각 연구자가 필요에 따라 음성 자료를 만들어 보관하고 이용해 왔다. 음성 연구가 진보 되어감에 따라 처리해야 할 자료 수는 많아지며, 준비해야 할 자료량도 대폭적으로 증가되어 왔다. 최근에는 음성 인식의 경우, HMM이나 bigram,trigram 등 언어 모델로 대표되는 통계 수법의 발달에 따라 대량의 음성 자료가 시스템의 학습에 필요하게 되었다. 한편 음성 정보 처리 시스템의 연구 개발을 위해서는 분석, 합성, 인식의 각종 알고리즘을 적절하게 비교 평가할 필요가 있지만, 이를 위한 방법으로는 현재까지는 공통 음성 자료를 이용하여 알고리즘을 수행하고 그 결과를 비교 하는 방법 이외에는 알려져 있지 않다. 따라서 공통 이용 가능한 각종 다음성 자료를 수록, 보관, 공개하는 것은 연구 개발 과정에서의 이용 및 인식 장치의 성능 평가 양면에서 필요하다. 이러한 목적으로 이용하는 음성 자료를 일반적으로 음성 자료 모음이라고 부른다.

2.음성 자료 모음의 필요성 외국의 경우 LDC(Linguistic Data Consortium) 등에서 이미 TIMIT과 같은 음성 자료 모음을 제작하여 CD-ROM에 담아서 보급하고 있고, COCOSDA(the Coordinating Committee for Speech Database and Assessment)에서는 POLYPHONE과 같은 프로젝트를 통해서 각국 언어별로 음성 자료 모음을 제작하고 있다. 따라서 앞에서 언급한 음성 연구의 필요성 이외에 우리가 서두르지 않으면 우리 언어의 음성 자료는 우리의 손이 아닌 외국에서 제작되어 우리 역수입해야 하며, 이렇게 되는 경우 음성 연구 분야에 있어서의 대외 종속을 면할 수 없다는 에서 한국어 음성 자료 모음의 구축이 시급하다.

3. 음성 자료 모음의 요구사항

가. 발성 내용

- **단음절**: 모음, 자음 + 모음, 모음 + 자음, 자음 + 모음 + 자음 등
- **단어**: 단독 숫자, 지명, 최소 음소쌍, PBW (Phoneticall Balanced Word), 고빈도 단어, 기능어 등
- **연속 단어**: 단어와 문장의 중간적 형태로, 연속 발성의 일종이다.
- **문장**: 일기 예보, 음운 발란스 문장

나. 발성자
- **연령별**
- **직업별 단어**
- **출신지별**: 12세 이전의 거주 지역을 기준으로 함
- **학력별**
- **표준말 및 방언 (지역별)**

다. 자료량
많으면 많을수록 좋으나 현실적으로 여러가지 제약이 있으므로, 갖추어야 할 음성 자료의 최소량은 다음과 같다.
- **특정 화자용**: 소수 화자, 최소한 2회 이상 발성
- **불특정 화자용**: 다수 화자, 최소한 1회 이상 발성

라. 녹음 조건
- **녹음 장소**: 무향실, 방음실, 사무실, 잡음이 있는 방 등
- **입력 장치**: 마이크, 전화기

마. 기록 매체
아날로그 녹음 테이프, 디지틀 MT, DAT cassette, CD-ROM, Optical Disk 등이 있으나 보존성 기억 용량 및 대량 복사 등을 고려하여 볼 때 DAT 및 CD-ROM이 좋다

바. 음성 자료 보관 형태
음성 파형 혹은 분석 처리를 한 특징 파라메터의 형으로 저장하는 방법이 있으며, 후자의 경우에 저장하기 위한 정보량의 압축 및 이용시에 계산량이 절감되는 등의 장점이 있으나 이용할 분석법에 대한 선택의 어려움과 특정 분석법에 의한 이용상의 제약이 있어, 배포를 목적으로 하는 경우에는 음성 파형의 형태로 저장하는 것이 바람직하다.

사. 편집 방법
녹음된 음성 자료를 편집할 때 크게 다음의 3가지 방법으로 생각할 수 있다.
- **편집하지 않는 경우**: 작업량은 적지만 기록 매체의 양이 많아진다.
- **무음 구간의 편집**: 음성 자료 시작점 및 끝점 전후에 300ms 정도의 무음 구간을 두고 편집한다.
- **헤더를 붙이는 편집**: 녹음된 자료에 단어명, 발성자, 날짜, 녹음 환경 등 각종의 정보를 부가한다.

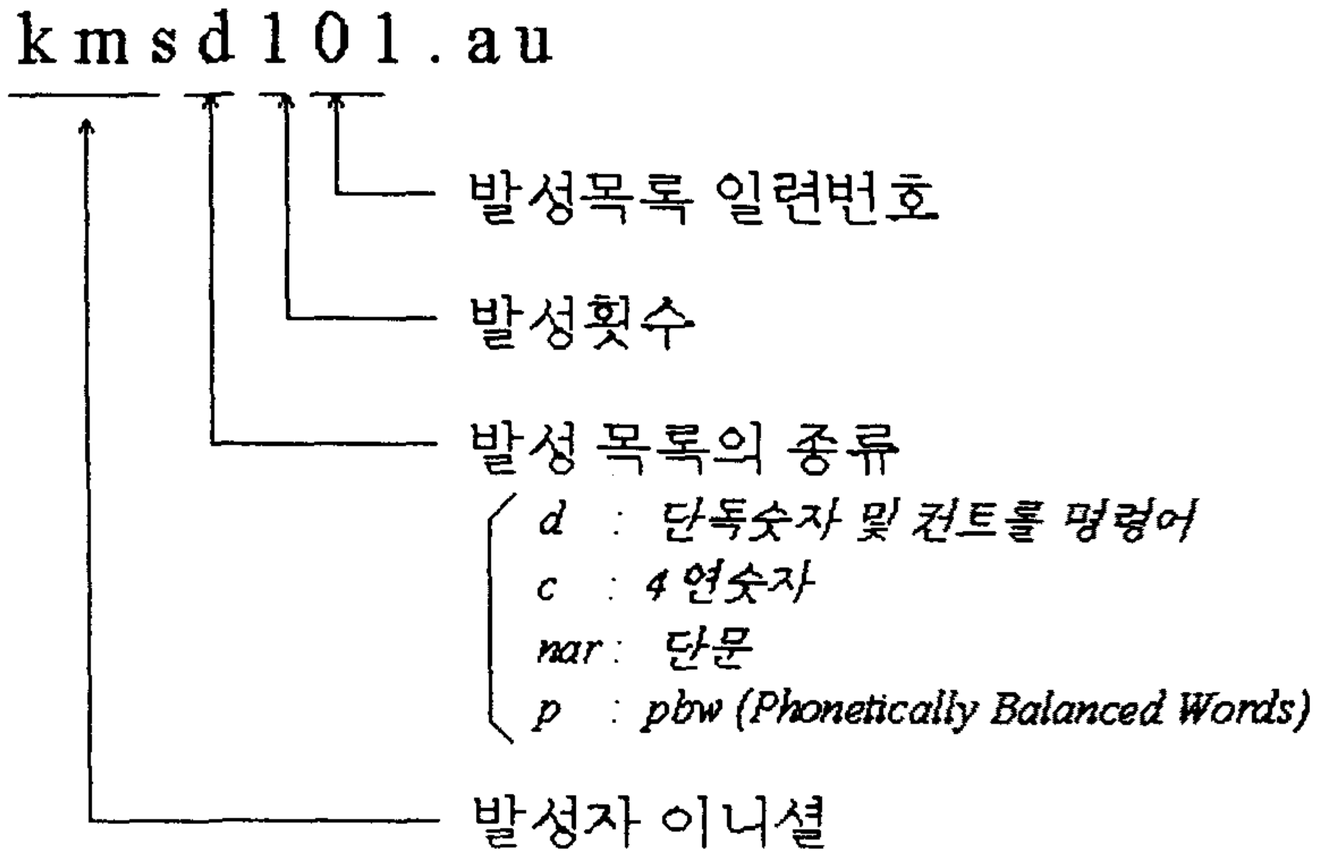
4. 음성 입출력의 특징

가. 편리성: 음성을 생성하는 동작은 지극히 자연스럽게 이루어지므로, 특별한 훈련이 불필요
나. 병렬성: 다른 동작과 병행이 가능하다. 즉, 손이나 눈으로 다른 작업을 하면서 말하고 들을수 있다.
다. 운동의 자유: 비교적 좁은 범위이기는 하나, 무선 마이크 등을 이용하여, 이동시 정보 전달 가능.

라. 고속성: 글로 쓰는 것보다 약 8배, 타이프라이터보다 약 3배 정도 빠르다.
마. 신뢰성의 향상: 온라인 입력이고 인식 결과를 곧바로 입력 단말에 표시해 주므로, 자료 입력의 신뢰성을 향상시킬 수 있다
바. 응답 특성의 향상: 원격 단말로부터 직접 입력이 가능하므로 신속한 응답 특성을 갖는 시템의 구성이 가능.
사 위기 입력이 가능: 예기치 않은 사태에서도 임기응변적인 입력을 신속히 수행 가능.
아. 화자 대조도 동시에 가능: 언어 정보와 함께 정보의 발생자를 확인할 수 있기 때문에 비유지, 위험 방지가 가능하다.
자. 장기 연속 입력이 가능: 자판을 통한 입력보다 장시간 연속해서 입력하는 경우에 신뢰성 향상된다.
차. 간략화 및 경제화: 직접 입력에 의해 중간 단계의 자료 수집 및 글쇠 입력 작업 등을 생략 수 있기 때문에 결과적으로 시스템 전체의 운용을 경제적으로 할 수 있다.
타. 민감성: 주변 잡음의 영향을 받기 쉽다. 또한 발성마다의 변동이 있다.
카. 순간성: 하드카피를 할 수 없다. 따라서 증거가 되지 않는다. 반면 불필요한 종이 등의 소비가 없다.

5. 시험판 음성 자료 모음의 화일 구조

가. 화일 명명 규칙



나. 발성자의 연령 및 성별: kms와 nsm는 20대의 여성이며 lyj는 40대의 남성 그리고 yoy는 20대의 남성입니다.

다. file type: 자료 화일은 16 kHz로 sampling하고 signed 16 Bits로 양자화 하였으며, 화일의 구조는 40 Byte의 .au화일 헤더와 raw 자료로 이루어져 있습니다.

조보자

Korean Information Base

세부 항목 : KLE DB Home Page
글자 단위 검색

개요

글자 자료 모음은 오프라인 글자 인식을 위한 자료를 제공하기 위해 KSC-5601 완성형 코드 중 520자, 1,000자, 그리고 2,350자 각각에 대하여 실세계에서 수집한 자료를 모아놓은 것입니다. 현재는 글자 단위 검색 서비스를 제공하고 있습니다.

최종 수정일: 1997년 3월 20일

KLE Administrator

KLE DB Home Page

조보지 [illegible]

Korean Information Base

세부 항목 : KLE DE의 소개
KLE DE의 구조
KLE DE의 통계자료
KLE DE의 사용법

[illegible]

KLE DB 서버에 오신 것을 환영합니다.
본 서버에서는 아래와 같이 메뉴 방식으로 한글 글씨 데이타베이스를 운영하고 있습니다. 아래의 메뉴를 선택하면 자세한 정보를 얻을 수 있습니다.
메뉴를 선택하세요.

개요

KLE DB에 관한 모든 소유권은 시스템공학연구소 부설 국어공학센터에 있으므로, 이를 사~ 하기 위해서는 반드시 국어공학센터의 허가를 얻어야 합니다.

KLE DB는 기본적으로 화일 단위로 구성되어 있으므로, Home Page의 구조 화면을 통하여 요한 화일을 선택한 다음 자신의 컴퓨터로 옮겨서 작업할 수 있습니다..

현재 모자익상에서는 화일에 대한 정보및 간단한 샘플 데이타베이스를 볼 수 있습니다. 샘플 화일에 관한 사항은 데이타베이스의 구조 화면을 참고하시고, 필요한 DB 화일은 국어공학센로 요청하시기 바랍니다.

기타 의문사항은 국어공학센터로 ...

[illegible] KBS [illegible]

최종 수정일: 1997년 4월 7일

KLE Administrator

KLE DB의소개

조보자 [redacted]

Korean Information Base

세부 항목 : KLE DE Home Page
KLE DE의 소개
KLE DE의 구조
KLE DE의 통계자료
KLE DE의 사용법

[redacted]

개요

KLE DB는 시스템공학연구소 국어공학센터에서 주관하는 국책 사업인 STEP 2000 프로젝트 국어 정보 처리 기술 개발과 관련하여 국어 정보 베이스 구축의 일환으로 고려대학교 컴퓨터과 Human Interface연구실에서 구축중입니다.

1994년 12월부터 3년간에 걸쳐 KS C 5601 완성형 한글 2,350자 1,000벌을 수집할 예정입니다.

KLE DB의 특성은 아래와 같습니다.

- 대상 문자 : KS C 5601 완성형 한글 사용 빈도순 2,350자 1,000벌
- 수집 인원 : 1,000명
- 수집 용지 : 양면 아트지, 건식 복사지, 갱지
- 필기 도구 : 사인펜, 볼펜, 수성 마킹펜 0.5mm
- 필기 유형 : 정서체, 자유필체
- 저장 형태 : 8bit gray level 영상(화소당 1byte 표현)
- 품질 평가 : 사람이 판단하여 3단계로 분류
- 저장 매체 : 고용량 하드 디스크
- 화일당 글자 수 : 수집 년도의 수집 문자 1셋트를 한 화일로

국어공학 KBS KLE DB Home Page

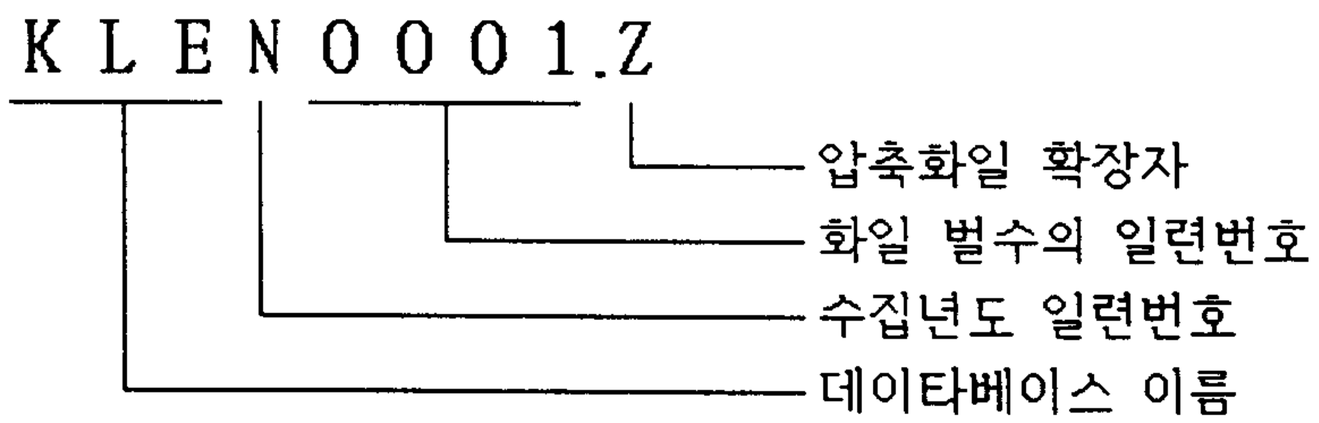
최종 수정일: 1997년 4월 7일

KLE Administrator

세부 항목 : KLE DE Home Page
KLE DE의 소개
KLE DE의 구조
KLE DE의 통계자료
KLE DE의 사용법

개요

- **형태** : ordinary file들의 집합 (당해 년도의 수집 대상문자 1세트를 1개의 화일로)
- **화일 이름의 작명 규약** :



- **데이타베이스 화일의 구조** :

byte 1 – 수집 년도(1: 1995년, 2: 1996년, 3: 1997년)
byte 2,3 – 벌수의 일련번호
byte 4 – 문자의 필기 형태 (1: 정서체, 2: 자유필체)
byte 5 – 문자의 품질 표시 (1: 양호, 2: 보통, 3: 불량)
byte 6,7 – 수집 년도의 전체 문자수(화일에 저장된 글자의 수)
byte 8 – 문자의 가로 픽셀 수
byte 9 – 문자의 세로 픽셀 수
byte 10 – 필기자의 성별 구분 (1: 남자, 2: 여자)
byte 11 – 필기한 손 (1: 오른손, 2: 왼손)
byte 12 – 수집 용지(1: 양면 아트지, 2: 건식 복사지, 3: 갱지)
byte 13 – 필기 도구(1: 사인펜, 2: 볼펜, 3: 수성 마킹펜 0.5mm)
byte 14 – 스캐너 종류 코드
byte 15 – 스캐너 Brightness
byte 16 – 스캐너 Contrast
byte 17,18 – 스캐너 해상도(dpi)
byte 19–25 – Reserved
byte 26....– 필기 문자 영상 데이타(1 화소당 1 byte)

- 문자의 필체는 **정서체**와 **자유필체**로 나뉜다. 다음은 1차년도의 경우에 해당한다.

정서체 화일 : 10001.Z에서 10500.Z까지
자유필체 화일 : 10501.Z에서 끝까지

- **데이타베이스의 사용 빈도순 별 분류 :**

520자 , 1,000자 , 2,350자

최종 수정일: 1997년 4월 7일

KLE Administrator

KLE DB 이용안내

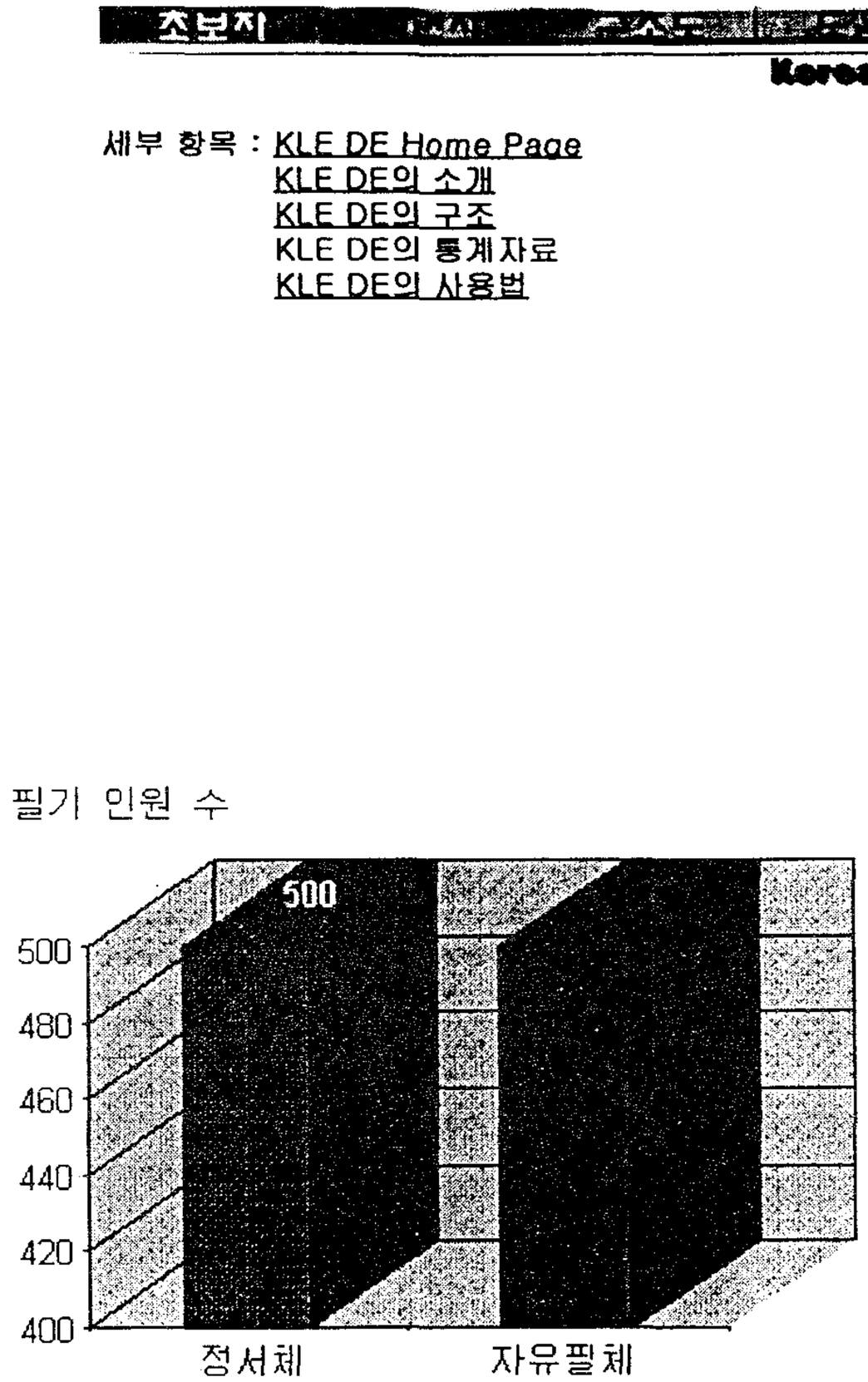
조보지 [illegible]

Korean Information Base

세부 항목 : KLE DE Home Page
KLE DE의 소개
KLE DE의 구조
KLE DE의 통계자료
KLE DE의 사용법

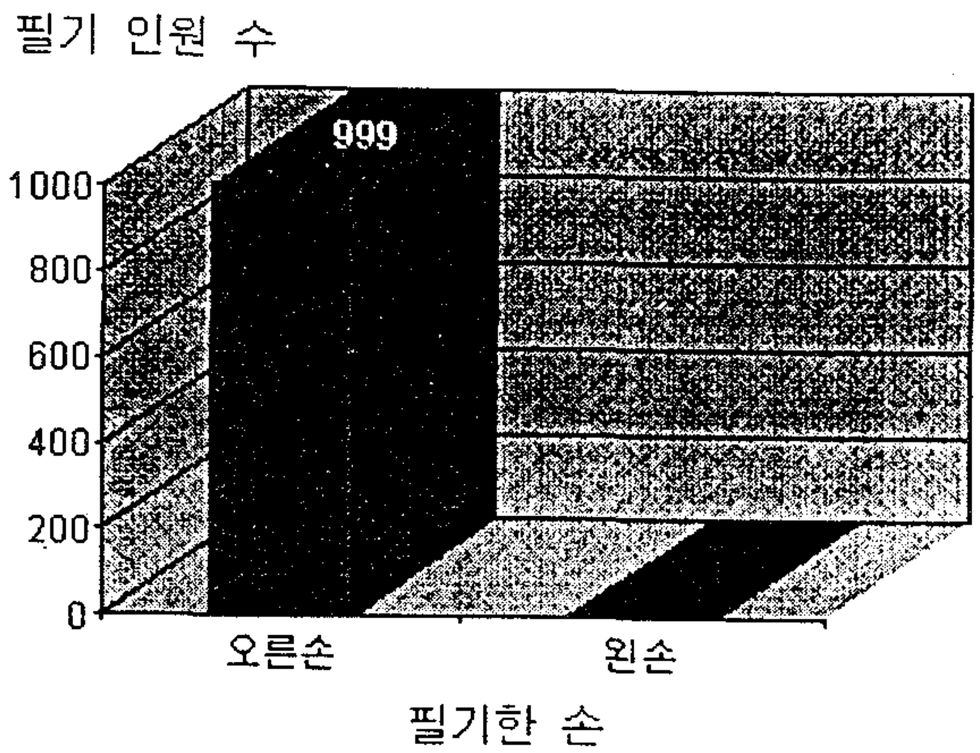
필기형태
데이타의 품질 상태
성별
필기한 손
용지 종류
필기 도구
지역

1.필기 형태

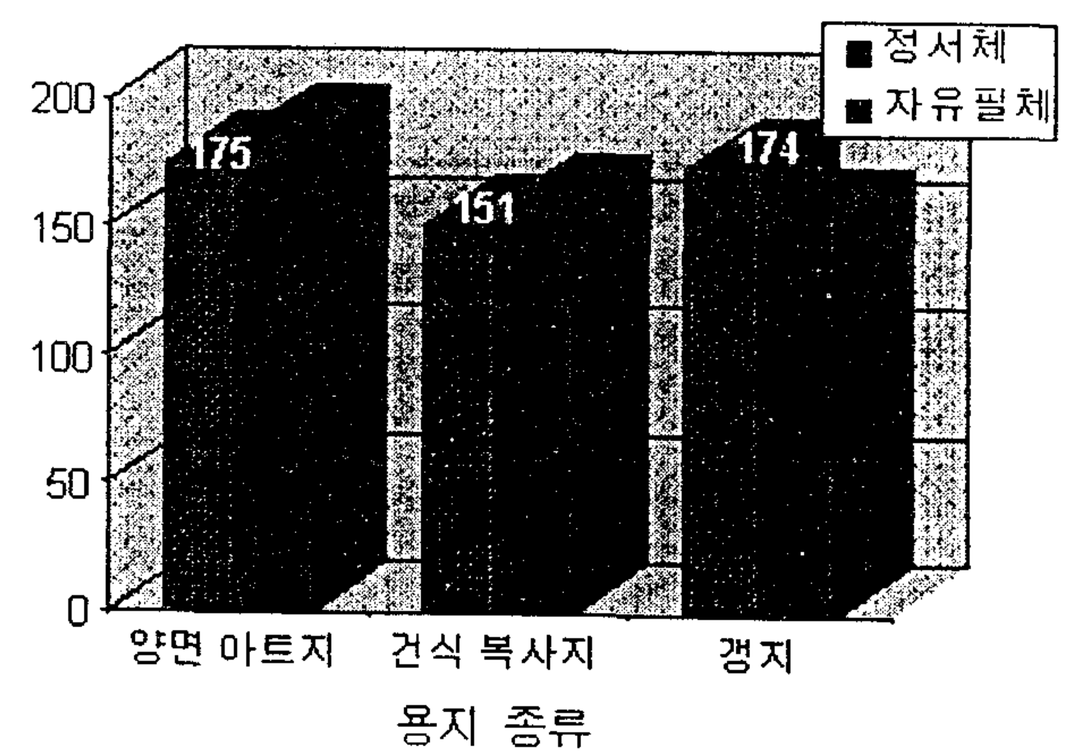


필기 형태

2.데이타의 품질 상태

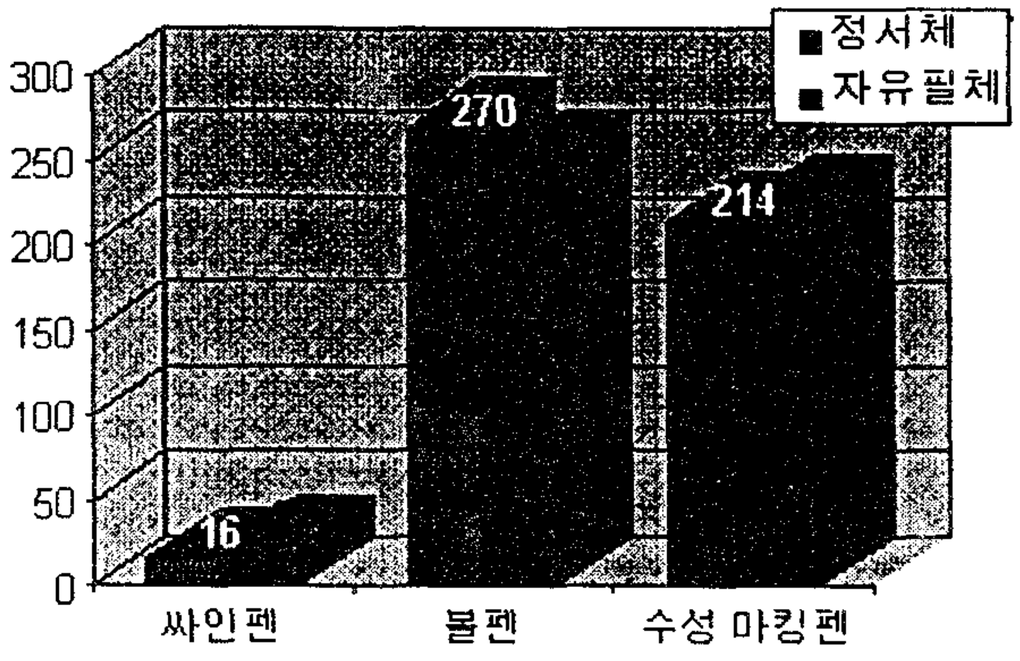

필기 인원 수
999
1000
800
600
400
200
0
오른손
왼손
필기한 손

5.용지 종류


필기 인원 수
정서체
자유필체
200
150
100
50
0
175
151
174
양면 아트지
건식 복사지
갱지
용지 종류

6.필기 도구

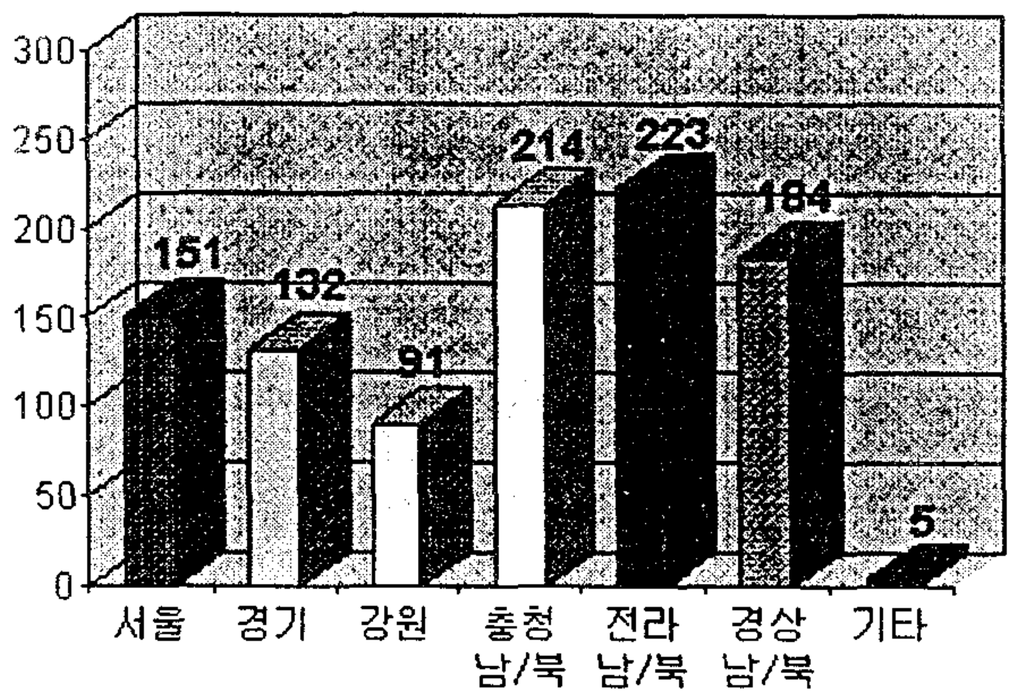
필기 인원 수



필기 도구

7.지역

필기 인원 수



지역

KLE DB 의사용법

조보지 (주소) 국어정보베이스 구축

Korean Information Base

세부 항목 : KLE DE Home Page
KLE DE의 소개
KLE DE의 구조
KLE DE의 통계자료
KLE DE의 사용법

개요

KLE DB에 관한 모든 소유권은 시스템공학연구소 부설 국어공학센터에 있으므로, 이를 사~ 하기 위해서는 반드시 국어공학센터의 허가를 얻어야 합니다.

KLE DB는 기본적으로 화일 단위로 구성되어 있으므로, Home Page의 구조 화면을 통하여 요한 화일을 선택한 다음 자신의 컴퓨터로 옮겨서 작업할 수 있습니다..

현재 모자익상에서는 화일에 대한 정보및 간단한 샘플 데이타베이스를 볼 수 있습니다. 샘플 화일에 관한 사항은 데이타베이스의 구조 화면을 참고하시고, 필요한 DB 화일은 국어공학센로 요청하시기 바랍니다.

기타 의문사항은 국어공학센터로 ...

국어학 KBS KLE DB Home Page

최종 수정일: 1997년 4월 7일

KLE Administrator

세부 항목 : KLE DB Home Page
글자 단위 검색

현재는 간단한 샘플 데이타베이스만을 제공하므로 이 외의 데이타베이스가 필요한 경우에는
국어공학센터로 문의하시기 바랍니다.

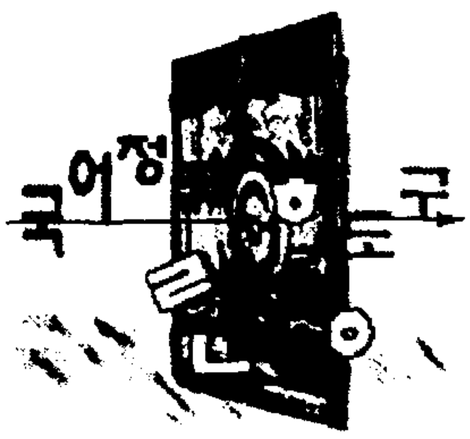
원하는 글자를 입력하세요.

다음의 항목을 선택하면 샘플 데이타베이스를 볼 수 있습니다.

- 정서체 문자 화일
- 자유필체 문자 화일

최종 수정일: 1997년 4월 7일

KLE Administrator



조보지

Korean Information Base I

세부 항목 : 기초 국어 정보 베이스
국어 정보 처리 도구
용어 사전

사전 개발 및 관리 시스템

태거
구문 트리 태거
한/영 정렬 시스템
문서 구조 표현을 위한 표준화
한국어 입출력 표준 환경
균형화 코퍼스 구축 표준 방법론
품사 사전 규칙과 시범 패키지

사전 개발 및 관리 시스템은 사전의 개발 및 관리를 위한 환경을 제공할 뿐만아니라 기존 시
으로부터 사용자의 요구에 맞는 특성을 가진 사전을 작성할 수 있는 일반적이고 유용한 기능
을 제공하려는 목적을 가지고 있습니다. 이 시스템의 사용자가 일반 사용자보다는 사전 작
업에 관련된 개발자나 관리자일 경우가 많고, 또한 시스템 개발상의 신속한 프로토타이핑 필
요성과 사용자 편의성을 제고시키는 그래픽 인터페이스의 필요성 등에 따라, WWW 인터페
스를 통해서는 사전 내용의 검색 기능만을 제공하게 하였으며, 나머지 부분에 대해서는 Tcl
터페이스를 근간으로 설계 및 구축 중입니다. 또한, 표준적인 사전 편집기 및 형식 변환기를
구현하기 위한 기간 작업으로 표준 사전 기술 언어 (Standard Dictionary Markup Langua
SDML)을 설계 중에 있습니다.
대화형 또는 일괄 방식으로 수행하도록 인터페이스가 구성되어 있습니다.
대화형 또는 일괄 방식으로 수행하도록 인터페이스가 구성되어 있습니다.
현재 한/영 양국어 문서 자료를 수집하는 동시에 기술 현황에 대한 조사가 이루어지고 있습
다.
SGML 및 TEI 기반 전문용어 관리 시스템 개발, TEI-K 에디터 및 파서를 제공하고 100 개
용어 시범 패키지를 구축합니다.
국어정보처리용 기본 공통루틴 및 도구를 개발합니다.
코퍼스 구축 방법론을 개발하고, 300만 어절 시범 패키지 구축을 합니다.

품사 분류의 표준화및 다른 품사 분류간의 변환기를 개발합니다.

최종 수정일: 1997년 3월 20일

KLE Administrator

세부 항목 :
사전 개발 및 관리 시스템
태거
구문 트리 태거
한/영 정렬 시스템
문서 구조 표현을 위한 표준화
한국어 입출력 표준 환경
균형화 코퍼스 구축
품사 사전 규칙과 시범 패키지

1. 개요

사전은 자연언어처리의 기초적인 정보이면서 동시에 가장 중요한 정보이기도 하다. 그러나 자연언어처리를 하는 여러 시스템이 같은 사전을 공유할 수 있는 것은 아니다. 사전을 이용하는 여러 시스템에서는 응용 분야의 특성에 맞는 독자적인 사전을 가지고 있으며 그러한 사전은 용과 구조가 다른 시스템에서 사용하기 어려운 독특한 형태이다. 따라서 새로운 시스템을 개하는 경우에 새로운 사전을 만들기 위해 또는 기존의 사전으로부터 새로운 시스템에 맞는 사으로 변형하기 위해서는 상당히 많은 양의 시간과 노력을 투자해야 한다.

사전은 응용 시스템에 따라 다른 구성과 다른 방식으로 저장되어 사용되어 왔기 때문에 사전 관리 즉 구축,수정, 검색과 같은 작업은 주로 윈도우식의 인터페이스나 지원 환경 없이 이루(져 왔다. 그러한 시스템이 있다고 해도, 특정한 응용을 위한 것이어서 다양하고 수시로 변하는 많은 자연 언어 처리 시스템에 이용하는 데 문제가 있다.

사전의 관리의 효율성과 응용 시스템으로의 이식 문제를 해결하기 위해서는 하나의 표준 사전과 표준 사전을 개발하고 관리하는 도구가 필요하며 응용 시스템으로의 이식을 위한 지원이 어야 한다. 사전 관리 도구는 윈도우 등을 통한 편리한 사용자 인터페이스와 각종 사전 관리 능뿐만 아니라 사전의 표준화를 위한 표준 사전의 정의와 표준 사전을 기술하기 위한 언어 등 필요하다.

본 연구에서는 여러 응용 시스템에 필요한 사전의 개발과 관리를 쉽게 하기 위하여 표준 사전 기술 언어(SDML)와 표준 사전 형식(SDF), 표준 사전 편집기(SDE)를 포함하는 사전 관리 도구(DDMS)를 제안한다. 사전 관리 도구는 하나의 통합된 표준 사전을 구축하고, 표준 사전으로부터 각 시스템 사전으로의 구조 변환 도구를 지원함으로써 시스템마다 사전을 개발하는 데 드는 노력을 줄일 수 있다.

2. 사전 구조와 사전 정보에 대한 연구

표준 사전을 정의하기 전에 여러 가지 사전의 구조나 내용들을 살펴볼 필요가 있다. 여기에서 먼저 일반 사전에 들어가는 내용들과 몇 가지 응용 시스템의 사전 구성을 보고 사전 작성에 관련된 문제들을 설명하려고 한다.

가. 출판된 국어 사전의 구조

일반적인 국어사전의 단어는 다음과 같은 구조로 표현되어 있다. ([KoreanDict : 설명도] 참

```
표제어 [[한자|로마자|일본어 가나]]* [읽기]* [품사]*
       [어원|외래어 어원]*
       {(1) 설명1 . {P 용례1/용례2/ ...} *
       (2) 설명2 . {세부 설명1. 세부 설명2 ...}* ...}
       ([속담] 설명)*
       그림*
```

대부분의 국어사전의 형식은 비슷하며 이러한 사전 구조가 일정한 형식을 따르기는 하지만 정확한 형식을 기술하기는 어렵다.

나. 컴퓨터 시스템에 사용되는 전자사전

컴퓨터 시트템에 사용되는 사전의 종류는 여러 가지가 있다. 연구를 목적으로 하는 경우는 형태소 해석용 사전, 구문 해석용 사전 의미 해석용 사전 등으로 분류되며 응용 시스템으로는 기계 번역, 대화체 기계 번역, 색인과 정보 검색 시스템, 철자 검색 및 교정 시스템 등과 자연 언어 처리에 사용되는 정보 축적을 위한 태깅 시스템, Concordance 등이 있다.

다음에 설명하는 분야는 기계번역 시스템을 위한 MATES/EK사전과 일본의 통합 전자 사전 , 스템인 EDR 사전이며, EDR 사전은 규모가 크고 잘 정리되어 있다.

1) MATES/EK

MATES/EK는 한국과학기술원에서 개발한 영한 기계 번역 시스템이며, MATES/EK가 사용하는 사전은 4가지이고 일반어와 전문어, 해석 사전과 대역 사전으로 구분된다.

```
일반어/전문어 해석 사전
일반어/전문어 대역 사전
```

해석 사전의 예(영어 명사 사전)

```
#ROOT          <generosity>
#UID           <1>
#LDOCE_UID     <1>
#SUBCAT        <COMM>
#INF_CODE      <3>
#PROPRTS       <ADJL>
#SEM_MK        <APH>
#NUMS          <U>
```

대역 사전의 예 (영한-명사)

```
#ROOT          <generosity>
#EK_UID        <1>
#E_UID         <1>
#E_POS         <N>
#K_POS         <NOUN>
```

```
#PRIORITY       <500>
#LEXICON1       <관용>
#POS1           <NOUN>
```

여기에 사용된 품사와 의미 특성의 종류는 다음과 같다.

```
영어 품사 - 명사,동사,형용사,부사,접속사,한정사,전치사,조동사,
            대명사,BE-동사, BE-조동사 등
영사 feature - root,uid,mlex,ldoce_uid,subcat,inf_code,inf_data,
    proprts,nums,mpos,sem_mk,article,position,gender,prep_lex1 등
```

이 사전의 경우에는 특정 시스템에 맞추어진 것이기 때문에 일정한 형식을 가지고 있기는 하만 다른 시스템에서 사용하기 어려운 부분이 있다. 즉 다른 시스템에서 사용하지 않는 정보들 있으며 다른 시스템에 필요한 정보가 없는 경우이다.

2) EDR 사전

다음은 일본의 전자사전연구소(EDR)에서 만들어 낸 Electronic Data Technical Guide에 L온 내용 중 사전의 구조와 내용에 대한 것을 정리한 것이다. EDR의 사전 만들기는 9년간의 ㅋ가 지원 프로젝트로 컴퓨터 시스템에서 사용하기 위한 대규모의 사전을 개발하기 위하여 시작되었다.

가) EDR사전의 구조

```
단어 사전 -    일반 어휘 -    일본어 20만 단어
                              영어   20만 단어
               기술 용어 -    일본어 10만 단어
                              영어   10만 단어
개념 사전 -    개념 분류 사전 40만 개념
               개념 설명 사전 40만 개념
공기 사전 -    일본어 공기    30만 단어
               영어 공기      30만 단어
대역 사전 -    일-영 사전     30만 단어
               영-일 사전     30만 단어
말뭉치 -       영어 말뭉치    25만 문장
               일어 말뭉치    25만 문장
```

나) 각 사전의 역할

```
- 단어 사전의 역할
      형태소해석, 구문 해석, 의미 해석을 위한 대량의 사전
- 개념 사전의 역할
      적절한 의미 표현을 생성
      semantic contents의 유사성 판단
      비슷한 semantic contents로 변환
- 공기 사전의 역할
      적절한 단어 선택
- 대역 사전의 역할
      대역 단어의 선택
      대역어의 특성 부여
- 말뭉치의 역할
      문서 Database
      kwic 생성
      문장 선택(문장의 종류에 따른 구분)
      문장 분석
```

다) 일본어 단어 사전의 구조

Headword information	- Headword	- 표제어
	Constituent(s)	- 구성 성분
	Notation	- 표기
	Adjacency attributes	- 관련 특성
	Kana notation	- 가타 표기
	Pronunciation	- 발음
Grammatical Information	- Part of speech	- 품사
	Syntactic tree	- 구문 트리
	Conjugation	- 활용
	Surface cases	- 표층격
	Aspect information	- 양태
	Function word information	- 기능어정보
Semantic Information	- Concept identifier	- 개념
	Concept illustration	- 개념 설명
Supplementary Information	- Usage	- 용례
	Frequency	- 빈도수

EDR사전의 경우 대부분의 단어와 정보를 포함하는 대량의 사전임에는 틀림 없으나 다음과 i
은 문제는 남아 있다. 첫째, 사전의 구조가 정해져 있어서 응용에 따라서는 새로운 정보를 추
하기 위하여 구조를 바꾸는 일이 쉽지 않다. 둘째, 사전 관리 도구를 만드는 일은 상당히 복잡
문제다. 셋째, 사전을 만드는 데 있어서 기존의 사전을 이용하는 방법이 명확하지 않다.

다. 국내 사전 개발/관리 시스템

국내에서는 포항공대에서 92년에 제안한 사전 관리 시스템을 비롯하여 여러 가지의 사전 개
/관리 시스템이 있으나, 특정 분야나 특정 구조에 한정되어 있기 때문데 일반적이고 총괄적인
시스템의 개발은 미흡하다. 다음에 설명하는 국어연구원의 사전 개발 시스템의 진행 중에 있
며 본 연구에 참고가 될 수 있다.

[국어연구원의 사전 개발 시스템]
국립국어연구원에서 최근에 전자 사전 개발을 위한 시스템을 설계 및 구현하고 있다. 기존에
나와 있는 대표적인 국어사전을 전산화하고 또한 통합 사전을 개발하며 관리(수정,검색)하는
것을 목표로 하고 있다. 기존의 사전은 다음과 같은 것이다.

금성판, 민중판, 삼성판, 한글학회판, 조선말 대판, 조선말판

각 표제어에 대해서 위의 사전의 정의를 모두 카드에 모아 기술한 후 전산 입력한다. 사전 정
는 다음과 같이 광범위하다.

품사,어원,발음,용례,뜻풀이(정의,설명),관용 표현,빈도 정보,
참고 어휘(thesaurus)

각각의 정보를 입력하기 위한 윈도우와 편집 방법들이 만들어졌다. 국어연구원의 결과는 다
과 같은 관점에서 유용하다.

1. 기존 사전의 전산화를 위한 실질적인 작업
2. 일반 사전의 편집을 위한 시스템
3. 많은 사람이 공동으로 사전을 개발하는 시스템

그러나 계산언어학 혹은 자연 언어 처리 관점에서 다음과 같은 한계점이 있다.
1. 만들어진 사전은 너무 비대해서 일부의 정보만 필요로 하는 응용 시스템이 쓰기에는 비효적이다.
2. 사전 정보는 언어 처리 시스템이 쓰기에는 덜 가공된 것이어서 추가적인 가공을 해야 한다
3. 사전 정보의 내용이 덜 공식화(formalize)되어 있다.
4. 사전 개발 환경은 원래 목적에 적합하지만, 다양한 자연 언어 처리 시스템을 위한 사전을 들기에는 불필요한 것이 많다.

라. 자연 언어 처리 시스템에 사용되는 사전의 하부 구조

자연 언어 처리에 사용되는 하부 구조는 시스템의 성격과 개발자의 선호에 따라 다르지만 대 Trie, Hashing, Btree 등이 많이 사용된다.

1) TRIE구조

TRIE구조는 형태소 해석기 등에 많이 사용되며 크기와 검색 속도 면에서 우수하지만 약간 복잡하다.

- TRIE의 기본 구조
- TRIE의 한 노드(sibling들이 묶인 형태)

하나의 노드는 여러 개의 sibling과 각각의 sibling에 대하여 child와 information (품사 정도을 가질 수 있다.

2) automata

사전을 automata로 표현 할 수 있다. Trie의 각 node를 state로 보고 input-char에 따라 각 다른 state(child node)로 갈 수 있다. state에 information이 있으면 사전에 단어가 있는 것다.
이 경우는 속도는 상당히 빠르지만 저장 공간을 많이 차지하는 단점이 있다.

3) dbm 화일 형식

key:contents 쌍을 dbm 화일에 넣고 key로 찾는다. dbm 화일은 UNIX에서 제공하는 hashing 방식의 저장.구조이므로 프로그램을 작성하기는 쉬우나 문제는 형태소 해석기에서 어를 찾기 전에 먼저 분리 해야 한다는 점이다. 또한 단어를 순서대로 출력할 수 없다.

마. 사전 정보의 종류

사전에 있어야 하는 정보의 종류는 응용 분야에 따라 다르며 정보의 내용도 다양하다. 통한 사전 관리를 위해서는 정보의 종류와 내용에 대한 표준을 정하는 일도 중요하다.
형태소 해석용 사전의 경우에는 다음과 같은 간단한 구조를 가진다.

단어(표제어) 품사

품사는 POS(Part of speech), Tag 등으로 불리기도 한다. 품사를 지정하는 방식은 초기의 우 접속정보에서 현재는 많이 달라졌다. 좌우 접속정보 방식의 문제는 음운 현상,형태소 결합 구문 정보를 모두 접속 정보로 표현하였으며, 숫자로 표시돼 읽기가 어렵다는 것이다. 현재는 Symbol에 의한 표시가 많아지고 있으며 구조적인 표현이 가능하다. 대표적인 것으로는 [김 훈]의 tag가 있으며 주로 구문 해석이나 tagging을 위한 tagset이다.

일반 사전에 사용되는 품사는 기본 8품사에 몇 가지 특성을 고려해서 약 20여 개의 품사가 사
되고 있으나 분석의 정확성을 고려하지 않아 세분화되지는 않았다. 시스템에서 사용하는 전기
사전의 경우는 정확성을 고려해야 하기 때문에 상당히 자세하게 분류하고 있으며 경우에 따라
100-200 개 정도의 품사가 사용된다.

형태소 해석용 사전이라도 사람이 보기 위한 또는 시스템이 사용하는 여러 가지 정보를 포함
수 있다.

단어(표제어) 품사 기타(빈도수,설명문,용례,comment 등)

구문 해석을 하는 경우에는 품사가 더 세분화되어야 할 것이다. 형태소 해석에 필요한 품사 ?
보는 구문 해석에서 사용하는 품사 정보와는 다르다.

형태소 해석을 위한 정보
  품사 - 명사, 동사, 형용사 등
  음운 정보 - 각종 불규칙 표시, 한글의 경우 종성 유무

구문 해석을 위한 정보
  품사 - 고유명사, 의존명사, 추상명사 등으로 세분화
  격 - 명사는 주격 목적격 보격 등으로 나눈다.
    동사의 필수격을 표시하기 위한 격틀 정보

의미 해석을 위하여서는 동사와 동사의 앞에 나오는 단어간에 의미를 검사하고 확정하기 위
여 다음과 같은 정보가 필요하다.

AKO - A Kind Of
  명사의 Type hierarchy 를 표시하는 방법
  명사의 의미 종류를 표시하는 정보를 가진다.
VKO - Verv Kind Of
  동사의 의미 종류를 표시하는 정보를 가진다.

정보 검색 등의 분야에서는 유사어 등의 관계를 표시하는 시소러스 정보가 필요하다.

바. 코드

한글의 경우 단어를 사전에 입력할 때는 컴퓨터 시스템이 사용하는 코드를 그대로 사용(주로
완성형과 조합형)하지만 실제 시스템 내부에서는 별도의 코드([이성진 코드], 3바이트 코드,
n바이트 코드 등)로 변환하여 사용하기도 한다. 자연 언어 처리에서 특히 형태소 해석 단계에
는 자소를 모두 분리하여 인식해야 하므로 완성형을 그대로 쓰는 것은 어렵다. 또한 조합형의
경우에는 5 bit 단위이므로 다루기가 불편하다.

특수 문자를 사전에 넣어야 하는 것은 또 다른 문제다. 그리스어나 다른 문자 기호, 도표 등을
사전에 넣는 것은 어떠한 코드를 사용하는 가에 따라 다르고 일반적이 코드 체계에는 없는 문
자나 기호가 있다. 꼭 필요한 code는 다음과 같이 정의 해서 사용하는 방법을 제공해야 한다

WSigma WAlpha

물론 형태소 해석 등에서도 특수 문자는 위와 같이 바꾸어 준 상태로 사전을 찾는다.

"ㄴ다" 같은 단어는 여러 가지 내부 표현이 있는데 'ㄴ'은 한글 자소 중 초성과 종성에 모두 ㄴ
온다. 따라서 초성인지 종성인지 아니면 낱자인지를 구분해서 넣을 필요가 있다. 또한 접두어

등은 '가-' 와 같은 형태로 들어 가는데 이것은 사람이 보기 위한 부분이고 실제로는 접두어ㄷ
는 품사를 알면 '가' 라고만 입력해도 상관 없다.

실제로는 표시되지 않는 음소의 표현 -- 형태소 해석기 중 일부는 실제로 표시되지 않는 음
를 다루는 경우가 있다. 예를 들면

(아/어)를 표시하는 음소
(애/개 모음 으)를 표시하는 음소
불규칙 동사의 표시(돕$)

이런 것들을 표기하는 방법 또한 제공되어야 한다.

WA
WU
W$

장/단음의 표시 -- 일반 사전에는 필요하지만 전자 사전에는 필요한 지 아직 알 수 없으며 ㅈ
/단음이 들어가는 경우 sorting의 문제가 심각해진다.

3.사전 개발/관리 시스템 앞에서 설명한 몇가지 사전과 사전 관리 시스템 등은 다음과 같은 문제를 가지고 있다.
1. 대부분의 경우 각 응용 시스템마다 다른 구조의 사전을 사용하고 있어서 사전을 관리하는
반적인 도구를 만들기 어렵다.
2. 새로운 응용 시스템을 만들게 되면 사전을 새로 만들어야 하는데 기존의 사전에서 정보를
출하는 방법이 불명확하여 사전을 만드는데 많은 시간과 노력이 소요된다.
3. 사전 관리 시스템은 대부분 특정 구조만을 지원하기 때문에 새로운 사전이 만들어 질 때미
새로운 사전 관리 시스템이 만들어져야 하다.
4. 사전을 구성하는 요소에 대한 연구가 충분하지 않고 표준화 되지 않아서 이에 대한 연구기
매번 반복된다.

이러한 문제를 해결하기 위하여 본 연구에서는 여러 응용 시스템에서 사용할 수 있는 통합 ㅅ
관리 도구를 설계하였다. 여기에 제안하는 사전 관리 도구는 일반적인 사전 관리 기능(검색,
수정 등)에 여러 가지 사전의 통합 관리, 유연한 구조 변경, 응용에 맞는 사전 생성 등을 지원
다.

DDMS(Dictionary Development and Management System)는 컴퓨터 시스템에서 사용되
여러 가지 형태의 전자 사전 개발과 관리를 보다 쉽게 하기 위하여 표준 사전 형식에 의한 사
의 개발과 관리, 검색 등을 지원하며 여러 형태의 전자 사전으로 Encoding/Decoding 방법
제공하는 총괄적인 사전 개발/관리 시스템이다.

DDMS는 사전의 기술 형식을 표준화하여 다음과 같은 장점을 가진다.
1. 여러 가지 사전의 통합과 개발이 쉽다.
2. 사전의 구조 변경이나 여러 가지 응용에 대처하기 쉽다.
3. 사전 개발/관리를 위한 표준 도구를 지원한다.
4. 각 응용 시스템에 맞는 형태로의 변환 도구를 제공한다.

DDMS는 다음과 같은 도구를 지원한다.
1. SDML(Standard Dictionary Markup Language)
사전을 기술하는 언어
3. SDML 컴파일러
사전형식에 대한 정의와 사전 내용의 구조를 분석하는
파서와 사전형식과 사전 내용에 대한 의미를 파악하여
적당한 테이블과 내부 구조인 SDIF(Standard Internal

Format:Tree구조)를 만들어 주는 부분으로 나뉜다.

4. SDF(Standard Dictionary Format)
   사전의 표준 형식. SDML언어로 기술된 사전 구조 정의부에 해당한다. SDF는 사전 형식과 사전 편집기 등의 관리 도구를 표준화 하기 위하여 정한 표준 형식이다.
5. SDE(Standard Dictionary Editor) — SDF사전 editor/browser
   SDF 형식에 맞는 SD사전의 구조적 편집을 돕는 도구이다. 사전의 표시 형식과 편집 방식은 SDF에 정의된다.
6. SD Encoder/Decoder
   표준 사전형식을 각 시스템에 적합한 형태로 바꾸어 주거나 반대로 각 시스템의 사전을 표준 사전형식으로 바꾸어 주는 도구

가. DDMS의 구성

DDMS의 전체 구성은 [구성도:DDMS Overview]에 나타나 있다.

다음은 주 모듈에 대한 설명이다.

- SDE(Standard Dictionary Editor)는 SDF로 된 사전(SD: SD는 SDF형식에 맞는 사전을 미한다.)을 읽어 들여 SDE의 내부 형태인 SDIF로 바꾼다. 이 SDIF는 SD의 구조를 나타내는 부분과 데이타 부분으로 나누어진다. SDE의 내부 모듈로 있는 SDML compiler는 SD를 변환시켜 SDIF로 바꾸어 주며, 이 SDIF에 대해 Edit, Browsing 등이 수행된다. SDIF가 변경되면 그 내용을 SDF 형태로 사전에 저장한다.

- SD Encoder는 SD를 읽어서 응용 프로그램에 맞는 형태로 변환시켜 준다. 대부분의 응용 로그램들은 수행상의 편의와 공간을 절약하기 위해 이진 화일 형태로 사전을 가지고 있고, 이 화일은 내부 구조가 Trie형태, Tree형태, Btree형태, Hashing table형태 등을 띠고 있다. SD Encoder는 이러한 형태로 SD를 변환시켜 화일에 저장한다. SD Decoder는 SD Encoder와 반대로 각 응용 프로그램에서 사용되는 내부 형식을 파악하여 SDF로 된 사전에 맞도록 변환 준다.

나. 표준 사전 표기 언어(SDML)

SDML(Standard Dictionary Markup Language)은 여러 가지 사전 형태를 기술하기 위한 언어이다. 사전의 기본 형태는 아래와 같은 Tree구조이며 사전의 내용 구성에 대한 정의는 SDF(Standard Dictionary Format)이라고 정의한다. SDML은 SGML(Standard Generaliz Markup Language)의 문법을 그대로 따르는 Subset이라 할 수 있으며, 사전 내용의 구성에 대한 정의는 SDF라는 이름의 DTD(Document Type Definition) 화일에 있다.

```
<ClassName (atrr_name=attr_value)*>
        (String |SubClass)*
</ClassName>
```

- 위의 구조를 Class(또는 Element)라고 정의한다.
- ClassName은 Class의 Start와 End에 사용되는 이름이며 ClassName으로 정의된 Class는 사전의 내용을 구성하는 요소가 된다.
- ClassName으로 시작하는 사전의 한 요소는 Instance라고 정의한다.
- attr_name은 같은 Class에 속하는 Instance의 속성(Attribute)을 표시하기 위하여 사용된다.
- attr_value는 attr_name의 값중 하나이며 SDF.dtd에 정의되어 있다.

- String은 문자열을 의미한다.
- SubClass는 Class이며 Sub Tree를 내포하기 위한 구조이다.
- '('.')'는 없어도 되는 부분이다.
- *가 붙으면 여러 개가 나올 수 있음을 의미한다.

다음은 SDML Compiler를 만들기 위한 자료 흐름도이다.

- SDML은 SDML.l(lexical rule)과 SDML.y(syntactic rule)으로 구성되며 lex와 yacc을 거쳐 SDML parser로 만들어진다. ([SDML.y SDML.l: 설명도] 참조)
- SDML은 SGML의 subset이다.
- SDML parser는 SDML compiler와 SD Encoder의 구성 성분이 된다.

– SDML semantics 사전 화일의 내용 구조는 SDF라 하며 SDF.dtd에 정의한다([SDF.dtd: 명도] 참조). 사전의 SDF형식을 Tree 구조로 보면 다음 그림과 같다.

실제로는 그림에 있는 요소가 모두 나타나는 것이 아니고 일부는 생략될 수 있고 일부는 반복 수 있으며 순서를 지켜야 하는 것과 순서가 없는 것이 있다. 그 자세한 내용은 SDF.dtd에서 설명한다. SDF.dtd의 사전 구조는 완벽한 것이 아니고 실험 단계에 불과하며 완벽한 구조를 들기 위해서는 많은 수정이 필요하다.

– SD(Standard Dictionary: SDF 형식으로 기술된 사전)의 예

사전은 SDF에서 정한 Class의 Instance로 시작한다.

```
<SDF>
        <DNAME> 기초 사전 </DNAME>
        <Date> 1995.7.17 </Date>
        <Author> 이 운재 </Author>
        <WLIST>
                <WORD>
                        <WNAME> 학교 </WNAME>
                        <Term>
                                <HanJa> 학교(한자로) </HanJa>
                                <Pronun> 학꾜 </Pronun>
                                <QuasiSy> 서당 </QuasiSy>
                                <QuasiSy> 학원 </QuasiSy>
                                <Broad> 교육기관 </Broad>
                                <Narrow> 고등학교 </Narrow>
                                <Narrow> 중학교 </Narrow>
                        </Term>
                        <Gram>
                        <POS> 명사 </POS>
                        <SUBC> 보통명사 </SUBC>
</Gram>
<Desc>
                        <Define>
                        일정한 목적.설비.제도 및 규칙에
                        의거하여 교사가 계속적으로 피교육자
                        에게 교육을 실시하는 기관
                                <Usage> 학교 보건 </Usage>
                        </Define>
                        <Phrase> 학교 종이 땡땡땡 </Phrase>
                        <Picture> school.bmp </Picture>
```

```
                    </Desc>
                </WORD>
        </WLIST>
</SDF>
```

다. SDE -- SDF editor/browser

1) SDE의 구성

SDML editor의 사용자 인터페이스는 윈도우 환경이며 그림은 SDE의 개략적인 구성과 자료 흐름을 보여 준다.

그림에 보이는 각 요소들은 다음과 같다.

- M-SDE : Manager Mode SDE.사전의 구조를 바꿀 수 있다.
- U-SDE : User Mode SDE, 사전의 내용을 검색 수정할 수 있다.
- Tcl Translator : SDIF의 내용을 근거로 Tcl프로그램인 U-SDE 모듈을 자동 생성한다.
- SDML Compiler : SD를 읽어 SDIF를 생성한다.
- SDE Converter : SDIF를 SD로 저장한다.

(자세한 내용은 다음 각 항목에 대하여 자료 흐름도 참조)

SDE가 초기화되고 사용자가 SD를 열면, 처음으로 SDML compiler가 호출된다. 이 compiler는 SD를 읽고 그 형태가 SDF에 맞는지 확인하면서 SDIF로 변환시켜 준다.

U-SDE는 SDIF에서 Tcl Translator를 통해 생성되어진 프로그램이며 User Interface를 담당한다.

U-SDE(User mode SDE)는 SD를 edit, browsing을 할 수 있지만, SD의 사전 구조를 바꿀 수는 없다. 사전 구조를 바꾸기 위해서는 M-SDE(Manager mode SDE)를 수행해야 한다.

SDIF가 만들어지고 나면, Tcl Translator가 호출되고 Tcl Translator는 SDIF에 있는 내부 형의 사전 구조에 근거하여 U-SDE의 원시 프로그램인 Tcl 프로그램을 자동 생성한다. 이 자동 생성된 Tcl 프로그램은 다시 Tcl interpreter에 의해 수행되어 U-SDE가 수행된다.

U-SDE는 SDIF의 데이타 부분을 처리할 수 있는 모듈을 포함하고 있어서, 사용자가 U-SDE 프로그램을 통해 SD을 edit, browsing 등을 할 수 있다.

U-SDE을 통해 수정된 SDIF를 SD에 저장할 경우, 사용자는 Save 명령을 수행한다. 이 명령은 내부의 SDF converter를 수행시켜 SDIF 데이타를 SD 사전으로 변환시켜 준다.

M-SDE(Manager mode SDE)는 SD의 구조를 바꿀 수 있는 프로그램이다.

M-SDE가 초기화되면, 일단 SDML compiler를 호출하여 SD를 SDIF로 변환한다. M-SDE 바로 SDIF 중에 있는 structure(사전 구조) 자료를 읽어 화면에 표시한다. 즉, SD의 구조를 표시한다.

SD 구조는 바로 M-SDE에서 수정할 수 있으며, 수정이 된 결과는 SDIF에 저장된다. 그러나 이 때, 그 구조 변경에 따른 사전 데이터도 변경되어야 하기 때문에 Reflector가 호출되어 그 조에 맞도록 데이타 형태도 변형시켜 준다.

구조까지 변환된 SDIF는 SDF converter에 이해 SD로 변환되어 저장된다. 이때 U-SDE에서 의 Save와는 다르게 SD 구조까지 저장한다.

SD의 구조를 수정한 다음 그 것을 verify하기 위해서 임시로 중간에 U-SDE를 수행시켜 볼 있다. 이 경우, U-SDE는 Tcl Translator를 호출하고, Tcl Translator는 SDIF에서부터 U-SDE를 자동 생성해 낸다. 이 프로그램은 Tcl interpreter에 의해 수행되며, 수행된 U-SD 를 이용하여 내부 구조를 browsing해 볼 수 있도록 한다. 이 때, 편의상 U-SDE에서 수정이 Save는 하지 못하도록 한다.

M-SDE에서는 사전을 merge할 수 있다. 이 merge 명령은 2개의 SD을 open하고 SDIF로 모리에 load하도록 한다. 이 때, 필요에 따라 SD의 구조만을 우선 load할 수도 있다. 일단 load가 되면 이 두 구조가 같은지 비교하고, 구조가 같으면 한 사전의 entry을 다른 사전의 entry에 추가한다. 이 때, 같은 entry등을 check할 수도 있다.

구조가 다를 경우는 merge가 불가능하다는 메시지를 내 보내고 끝낸다. 이 때는 M-SDE의 조 변경 기능을 이용하여 한 사전의 구조를 다른 사전 구조와 같게 변형한 다음 merge를 하 된다.

2) SDE의 기능

- 새로운 사전 항목의 추가 또는 기존 항목의 수정/삭제
- 항목의 복사와 위치 이동(sorting 기능에 의한 자동 이동 포함)
- 정보의 수정
- browsing - 검색과 위/아래로의 순차적 이동을 지원한다.
    - keyword 탐색
    - linear 탐색: browsing 기능
        - 위로(prev) : 전 단어 찾기
        - 아래로(next): 다음 단어 찾기
        - history: 지금까지 본 단어들 리스트
        - foreward: history point에서 다음으로 가기
        - backward: history point에서 이전 단어로 가지
- 기타 주변 기능
    - 로그 정보 화일 작성
    - history기능
    - undo 기능
    - off-line 작업들
        encoding decoding을 포함하는 시간이 걸리는 일

3) SDE 윈도우 사용자 환경

- 메뉴
    - 화일
        - 읽기(parser를 통하여 읽어 들인다.)
        - 쓰기(SDF 형태로 출력한다.)
        - 다른 이름으로(다른 형식으로) 저장
        - 프린트
    - 편집
        - 찾기, 삽입, 수정, 삭제

- 옵션
  - 사용자 환경 설정
    - 사용자 이름 변경
    - 작업 디렉토리
    - 화면 표시 형식
      기능 버튼의 배치
    - 출력 형식
    - 기타
      save/load config
      log filename
      memo
  - 도움말
    - DDMS에 관하여
    - SDML editor 사용법
- 버튼
  - prev
  - next
  - backward
  - forward
  - edit
  - search
- 기본 작업 윈도우
  - 주 작업 윈도우는 SDF에 기술된 형식으로 각 항목을 배치한다.

라. SD Encoder/Decoder

사전 검색/편집기가 SDF 형식의 사전만을 처리하므로 이것을 다른 형태로 바꾸거나 다른 형에서 SDF로의 변환이 필요하다. 변환의 target 구조는 이미 정의된 몇 가지로 제한한다.

마. 사전의 개발과 관리 과정

DDMS를 통한 사전의 개발과 관리 과정은 다음과 같다.

1) 사전 구조 정의(Dictionary Definition)

현재는 SDF라는 형식으로 고정되어 있다. 그러나 앞으로 SDF가 사전의 모는 내용을 완벽하 표현할 수 있기 위해서는 계속 사전 구조의 정의를 바꾸어 보강해야 한다. 또한 가능하다면 / 전 구조의 정의는 유연하게 바뀌어질 수 있어야 하며 사전 구조를 바꾸는 도구를 개발해야 흔

-

2) 사전 입력

사전 관리 도구를 통하여 사전을 입력하거나 다른 형태의 사전을 변형하여 새로운 사전을 만다.

3) 관리

사전 관리는 다음과 같은 내용을 포함한다.

- 사전 구조의 변경
  - 새로운 항목의 추가 또는 기존 항목의 삭제
  - 항목간의 위치 변경
  - data range의 변화 -- 품사 set의 변화 등
- 사전 내용의 변경

- 새로운 단어의 추가 또는 기존 단어의 삭제
- 두 단어의 병합
- 한 단어의 분리
- 두 가지 사전의 통합
  - 사전 형식이 다른 경우
    - SDF인 경우
      - 한쪽이 다른 쪽의 부분 집합
      - 전혀 다른 형태인 경우
    - SDF 가 아닌 경우
      - 한쪽이 다른 쪽의 부분 집합
      - 전혀 다른 형태인 경우
  - 사전 형식이 같은 경우

바. DDMS prototype

SDF Dictionary및 SDE의 구현에 대한 가능성과 문제점들을 알아보기 위해서 아래와 같은 Prototype으로 실험을 해 보았다. 시험 결과로 SDML parser(실제로는 SDML parser가 아닌 test용 parser이다.)를 포함하는 간단한 SD editor와 내부 저장 구조(Tree 형태의 SDIF prototype)을 만들어 냈다.

1) 사전 입력 형식

```
<ENTRY>
<EDIT name=FORM>                    </EDIT>
<LIST name=POS>                     </LIST>
<LIST name=SUBC>                    </LIST>
<MEDIT name=SAMPLE>                 </MEDIT>
</ENTRY>
```

EDIT는 editing이 가능한 field를 의미한다.
LIST는 List box에 들어 있는 Item들 중 하나를 고르는 field를 의미한다.
MEDIT (Multi-line EDIT)는 여러 라인을 editing하는 것을 의미한다.

2) User Interface 윈도우

윈도우1
윈도우2
윈도우3
윈도우4

FILE
- Open : 입력 사전을 컴파일하고 그 내용을 윈도우에 Display 해 준다.
- Close : 현재 윈도우에 Display된 사전 화일을 닫는다.
- Save : 현재 윈도우에 Display된 사전을 화일에 저장한다.
- Save As : 현재 윈도우에 Display된 사전을 화일에 다른 이름으로 저장한다.
- Exit : DDMS를 끝낸다.

Next, Prev : Mosaic에서처럼 Navigate하는 기능.
Insert, Delete : 한 Entry를 삽입, 삭제하는 기능.
FORM : 단어의 형태. 예) '학교'
POS : Part of Speech. 단어의 품사. 예) '명사'
SUBC : Subcategory. 하위 품사. 예) '보통 명사'
SAMPLE : 용례들.

3) Parser 생성

1)의 입력을 이용해서 2)의 윈도우를 만들기 위해서는 1)의 규칙들을 parsing하기 위한 Parser가 필요하다. 이 때, Lex와 Yacc을 이용해서 Parser를 만들고, 이 Parser는 입력을 Tree로 만들어 준다. 생성된 Tree를 이용해서 Tcl로 2)의 윈도우를 그린다.

Lex와 Yacc Specification은 아래와 같다.

```
Lex Specification
        <ENTRY> ; 한 ENTRY의 시작
        <EDIT           ; EDIT Line의 시작
        <LIST           ; LIST Line의 시작
        <MEDIT          ; MEDIT(Multi-line EDIT)의 시작
        </ENTRY>        ; 한 ENTRY의 끝
        </EDIT>         ; EDIT의 끝
        </LIST>         ; LIST의 끝
        </MEDIT>        ; MEDIT의 끝
        >
        =
        [a-zA-Z]*       ; 알파벳으로 이루어진 문자열
        .               ; 기타 입력
Yacc Specification
        ENTRY_LIST      -> (ENTRY)*;
                                /* ENTRY_LIST는 ENTRY의 반복 */
        ENTRY           -> '<ENTRY>'(EDIT|LIST|MEDIT)+'</ENTRY>';

        EDIT            -> EDIT_HEAD text_list EDIT_TAIL;
        EDIT_HEAD       -> '<EDIT' attr '=' attr_name '>';
        EDIT_TAIL       -> '</EDIT>';

        LIST            -> LIST_HEAD text_list LIST_TAIL;
        LIST_HEAD       -> '<LIST' attr '=' attr_name '>';
        LIST_TAIL       -> '</LIST>';

        MEDIT           -> MEDIT_HEAD (text_list)* MEDIT_TAIL;
        MEDIT_HEAD      -> '';
        MEDIT_TAIL      -> '';

        attr            -> NAME;
                /* NAME은 입력 문자열 */
                /* NAME은 미리 정의된 문자열들 중 하나 */
        attr_name       -> NAME;
        text_list               -> LITERAL;
                /* LITERAL은 임의의 문자열 */
```

4) 사전 예

```
<ENTRY>
        <EDIT name=FORM> 학교 </EDIT>
        <LIST name=POS> 명사 </LIST>
        <LIST name=SUBC> 보통명사 </LIST>
        <MEDIT name=SAMPLE>
                나는 학교에 가는 중이다.
                학교에서는 공부를 한다.
        </MEDIT>
```

```
</ENTRY>
<ENTRY>
        <EDIT name=FORM> 먹다 </EDIT>
        <LIST name=POS> 동사 </LIST>
        <LIST name=SUBC> 일반동사 </LIST>
        <MEDIT name=SAMPLE>
                나는 지금 밥을 먹는다.
                모든 생물은 먹어야 산다.
        </MEDIT>
</ENTRY>
```

5) Parsing 결과 tree

4. 다음 단계에서 처리해야 할 문제점들

가. search algorithm

search algorithm은 내부 구조에 따라 달라진다. 그러나 일반적으로는 Key에 대하여 Index를 가지는 것이 보통이다.

- 불리언 질의어에 의한 탐색은 또 다른 문제를 가지고 있다. 여러 개의 field에 대한 Index를 가지고 있어야 하는데 공간적/시간적 낭비를 최소화하거나 불리언 질의어 탐색을 포기해야 한다.

```
명사이면서 동사인 단어
어원이 일본어인 모든 명사
```

- similarity match의 경우는 Index를 특별히 설계해야 한다.

```
가*     : 가로 시작하는 모든 단어
*ㅐ     : ㅐ로 끝나는 모든 단어
```

나. hyper link

사전의 기본 구조는 Tree이다. '학생'이라는 단어를 보다가 용례에서 '학교'하는 단어를 발견하고 그 단어의 의미를 알고 싶다면 '학교'라는 단어를 click함으로써 단어를 찾을 수 있을 것이다. 여기에는 두 가지 방법이 가능한데, 한 가지는 click한 위치의 단어를 Index에서 Search하는 방식이고 다른 한 가지는 '학교'라는 단어 자체에 자신을 설명하는 위치에 대한 정보가 있어서 직접 찾아가는 방식이다. 현재는 전자의 방법으로 충분히 기능을 발휘할 수 있어서 구현을 고려하지는 않았으나 Index를 만들기 어려운 경우에 고려할 만하다.

다. dictionary operation

- 사전 병합, 사전 형식 변환, 부분 정보 추출 등
- 사전 연산 기능: dictionary DB(join,select...)
  불리언 질의에 의해 나온 결과의중간 저장 구조와 Browsing방법 등
- sorting
- 일관성 검사 -- 정보간의 모순이 없는가에 대한 검사
- redundancy check — 정보의 중복에 대한 검사
- concordance, spelling check, statistics computation

5. 결론

본 연구에서는 자연 언어 처리를 위한 사전 개발/관리 시스템인 DDMS를 설계하고 일부에 대
한 프로토타입을 만들어 가능성을 시험하였다. 지금까지 자연 언어 처리를 위한 사전 시스템
개별적으로 상당히 많이 만들어졌기 때문에 사전을 만드는 시간과 비용이 상당한 양이었다.
한 개별적인 사전 관리 시스템은 문제의 복잡성만을 증가하고 추가적인 부담일 뿐 전체적인
전 개발과 관리라는 측면에서 별 도움이 되지 못했다.

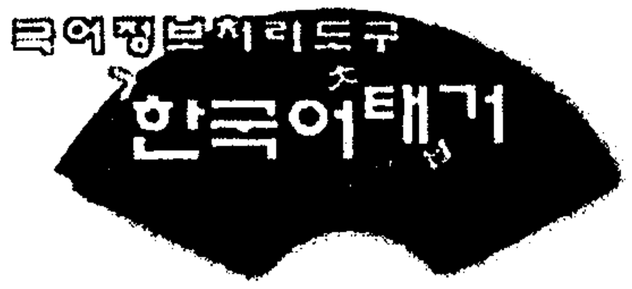
본 보고서는 통합 사전의 형식과 문법을 기술하는 SDML을 제안하였으며 표준 사전 형식으로
SDF를 보였다. SDML은 SGML의 부분집합으로서 표준화에 바탕을 두고 있으며 표준화에 의
한 여러 가지 이점을 가지고 있다. SDF editor(SDE)는 표준 사전 형식을 따르는 사전의 개발
과 관리를 도와주는 도구이며 각종 사전 관리 모듈을 포함하고 있다. 또한 각 응용 시스템에
용하는 사전을 만들기 위한 SD Encoder/Decoder를 제안하여 관리는 한 곳에서 응용은 여러
곳에서 하도록 하였다.

본 연구에서 DDMS를 설계하면서 통합 사전 개발/관리의 가능성을 보였으며 이 분야에 상당
발전을 예측할 수 있었다. 사전 개발/관리 시스템은 앞으로 여러 가지 종류의 사전과 여러 가
종류의 다른 정보들-- 즉 시소러스, 개념 정보(AKO,VKO), Corpus등--과의 결합이 가능
것이며 사전 개발자는 새로운 사전을 개발하기 위하여 사전 구조에 한 항목을 추가하거나 기
의 정보를 변형하거나 기존의 정보를 참조하여 사전을 만드는 일이 더욱 쉬워질 것이다.

최종 수정일: 1995년 11월 01일

KLE Administrator

초보자 Korean Information Ba



세부 항목 :
사전 개발 및 관리 시스템
태거
구문 트리 태거
한/영 정렬 시스템
문서 구조 표현을 위한 표준화
한국어 입출력 표준 환경
균형화 코퍼스 구축
품사 사전 규칙과 시범 패키지

품사 태깅이란?,
연구 개발의 필요성,
태깅 모델,
연구 개발 현황,

1. 품사 태깅이란?

가. 태그(tag)

태그(tag)는 본래 가방이나 옷에 달려 있는 가격표 같은 것을 말합니다. 언어 처리 시스템에서의 태그는 여러가지 종류가 있을 수 있지만 여기에서는 단어마다 붙어 있는 품사를 의미합다. 예를 들어

나는 학교에 간다.

라는 문장에 대하여 태그를 붙이면

나/npp+는/jx 학교/nc+에/jca 가/pv+ㄴ다/ef ./s.

와 같이 됩니다. npp,jx,nc,jca,pv,ef,s. 등은 모두 품사를 나타내는 기호(tag)이며, 이러한 호들의 집합을 TAG SET이라고 합니다.

나. 태깅(tagging)

단어에 태그를 붙이는 작업을 태깅(tagging)이라고 합니다. 한국어를 태깅하려면 일단 어절 분석하여 단어로 나누어야 합니다. 이 과정을 형태소 해석이라고 하는데 하나의 어절에 대하 여러가지 해석이 가능한 경우가 있습니다.

나는 ==>
나/동사+는/어미
나/대명사+는/조사
날/동사+는/어미

한 단어에 대한 태그는 하나만 붙이는 것을 원칙으로 합니다. 따라서 형태소 해석의 결과 중 나를 선택해야만 합니다.

태깅에는 수동 태깅과 자동 태깅이 있는데 수동 태깅은 사람이 직접 태그를 붙이는 작업을 하 것이고 자동 태깅은 프로그램을 이용하여 자동으로 태그를 붙이는 것을 말합니다.

다. 태거(tagger)

태거(tagger)는 자동으로 단어에 태그를 붙이는 프로그램입니다. 태거의 역할은 크게 형태소 해석과 중의성 해결 두 가지로 볼 수 있습니다. 영어의 경우에는 형태소 해석이 거의 필요치 않으나 한국어에서는 형태소 해석이 중요합니다. 한편, 중의성 해결 방법에는 규칙을 이용하는 방법과 확률을 이용하는 방법이 있으며, 확률을 이용하는 방법이 최근에 많이 사용되고 있습니다.
태거는 미등록어 처리를 포함하기도 합니다. 즉, 미등록어는 사전에 없는 단어를 가르키는데 미등록어가 있으면 주위의 단어를 보고 미등록어의 품사를 추정해서 태그를 붙여 줍니다.

라. 품사 태깅

태그를 붙이는 일 중에서 특히 품사 태그(part-of-speech)를 붙이는 일을 품사 태깅 (part-of-speech tagging)이라고 하며, 이러한 일을 하는 시스템을 품사 태깅 시스템 혹은 품사 태거라고 합니다.

2. 연구 개발의 필요성

태깅을 하게 된 배경은 언어 자료의 수집과 이를 이용하기 위한 기초 분석에 있습니다. 언어를 분석하는 기초 도구로는 형태소 분석기, 용례 분석기, 태거, 구분트리 태거 등이 있으며 이 중에서 태거의 역할은 상당히 중요합니다.
대량의 말뭉치를 태깅하여 얻은 태깅된 말뭉치는 단어의 분포, 인접 단어 정보, 품사 n-gram 등을 비롯한 유용한 정보를 제공하며, 구문 해석기 등의 시스템에서 test 문서로 이용됩니다.

태깅 시스템의 다른 용도로는 구문 해석기의 전처리기나 문자 인식의 후처리기로 사용되는 경우를 들 수 있습니다. 구문 해석기의 입력, 즉 형태소 해석의 결과가 중의성을 가진다면 구문 해석기는 필요 없는 품사들에 대한 구문트리를 만들기 위해 상당히 많은 시간이 걸리게 되는데 태거를 전처리기로 사용한다면 시간을 절약할 수 있고 좋은 결과를 낼 수 있습니다.

3. 태깅 모델

본 연구에서는 통계적인 처리의 일환으로 은닉 마르코프 모델(Hidden Markov Model)을 기본 모델로 하였고 어절 관계와 형태소 관계를 동시에 은닉 마르코프 모델에 반영하여 태깅의 정확도를 높인 모델을 이용하여 한 국어 태깅 시스템을 구축하였습니다.

4. 연구 개발 현황

지금까지 구현된 태깅 시스템은 규칙을 이용하는 방법과 확률을 이용하는 방법으로 나눌 수 고 이외에도 신경망을 이용하는 방법과 퍼지망을 이용하는 방법도 있지만 크게 확률을 이용ㅎ 는 방법으로 볼 수 있습니다.

규칙을 이용한 방법은 태깅 연구의 초기 단계에 주로 개발되었던 방식으로서, 이 방법은 규ㅊ 만들기 어렵고, 견고성이 없고 규칙의 일관성을 유지하기 어렵다는 문제를 가지고 있습니다. 그러나, 태깅을 위해 필요한 규칙의 수가 적고, 시스템의 성능 향상을 쉽게 꾀할 수 있으며, l 른 태그 집합, 다른 유형의 말뭉치, 다른 언어에 쉽게 적응할 수 있다는 장점도 있습니다. 하 만 규칙 기술과 관리에 드는 수작업의 단점으로 인해 지금은 활발히 연구되고 있지는 않은 편 입니다. 대표적인 시스템으로 TAGGIT 시스템 , Klein의 태깅 시스템, Hindle의 태깅 시스템 을 들 수 있습니다.

확률을 이용하는 방법은 말뭉치에서 필요한 정보를 얻게 되므로 사람이 규 칙을 기술할 필요 없고 대상 말뭉치에도 같은 엔진을 이용할 수 있다는 장점이 있지만 학습을 위한 대량의 말뭉 치가 필요하다는 단점이 있습니다. 그러나 구축된 말뭉치는 다른 여러 분야에서도 유용한 기 데이타가 되므로 말뭉치 구축 자체로도 큰 의미를 가질 수 있습니다. 대표적인 시스템으로 Charniak의 태깅 시스템, Kupiec의 태깅 시스템, Benello의 신경망을 이용한 태깅 시스템, 윤재의 HMM/3 등을 들 수 있습니다.

조보자

Korean Information Ba

사전 개발 및 관리 시스템
태거
구윤 트리 태거
세부 항목 : 한/영 정렬 시스템
문서 구조 표현을 위한 표준화
한국어 입출력 표준 환경
균형화 코퍼스 구축
품사 사전 규칙과 시범 패키지

구문트리 태거란?,
목적 및 목표,
중요성

1. 구문트리 태거란?

구문 태깅이란 구문적인 구조를 찾아내는 프로그램으로, 구문 분석에서 구조적인 분석만 수
한 결과를 냅니다. 구문 구조를 찾아내기 위해서 본 연구에서는 확률적 의존 문법을 사용하였
는데, 어순의 자유로움과 생략 현상이 빈번한 한국어에 잘 적용된다고 알려진 의존 문법에 통
계적인 선호도를 부여함으로써 더 견고하고 정확한 분석을 시도합니다. 형태소 태거가 출력
준 (모호성이 제거된) 형태소 분석 결과를 이용하여 구조적인 분석을 하고 나면, 의미적 격 분
석을 하기 바로 전 상태의 구문트리가 생성됩니다.

2. 목적 및 목표

이 연구의 목적은 확률 및 통계에 기반한 구문 분석을 위한 언어 자료의 구축에 있습니다. 이
따른 연구의 목표로서는 먼저 구축해야 하는 자료의 내용과 형식을 정의하고 타당성을 밝히
이 자료를 구축하는 방법론을 제안하는 것입니다.

3. 중요성

규칙 기반 언어 처리의 한계를 극복하기 위하여 통계 및 확률 기반의 언어 처리 방법이 등장
게 되었는데, 이에 대해 [Foster91]에서는 다음과 같은 3 가지의 중요한 장점을 제시하고 있
니다.

첫째, 통계적인 모델들은 단순하면서도 계산적으로도 부담이 적다.
둘째, 모델의 중요한 변수들이 자동으로 얻어질 수 있으므로 여러 대상에 적용이 쉽다.
셋째, 자연 언어같이 규칙이 변화 가능한 개방성에 잘 대처할 수 있는 견고한 성질이 있다.

이러한 장점들 때문에 근래에 한국어 처리에 통계 및 확률 모델을 적용하려는 시도가 있었고
이 중 대부분은 [이운재93],[임철수94],[이상호94a],[이상호94b]과 같이 품사 태깅에 맞추(
져 있는데, 이는 품사 태깅의 문제가 비교적 단순하며 통계 정보를 제공하는 자료의 구축이 쉽
이하기 때문입니다.

그러나, 한국어 구문 분석의 경우 통계 및 확률에 근거한 처리가 상당한 장점이 있음에도 불

하고 뒷받침할 만한 언어적 자료가 절대 부족하고 구축하기 어렵다는 점 때문에 그동안 연구 활발하게 이루어지지 못하였습니다. 하지만, 구문 분석용 통계 정보를 획득할 수 있는 언어 자료를 구축하는 것은 매우 필요한 일이라고 판단됩니다.

최종 수정일: 1995년 11월 01일

KLE Administrator

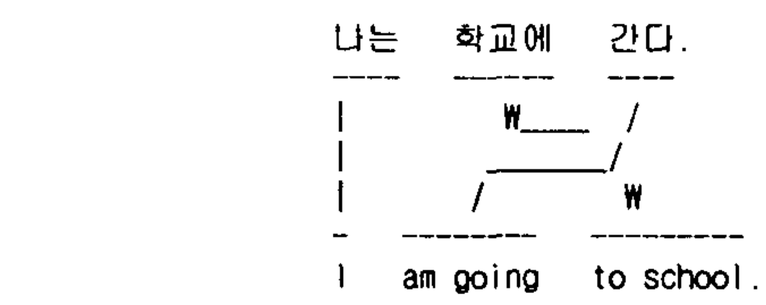
세부 항목 :
- 사전 개발 및 관리 시스템
- 태거
- 구문 트리 태거
- 한/영 정렬 시스템
- 문서 구조 표현을 위한 표준화
- 한국어 입출력 표준 환경
- 균형화 코퍼스 구축
- 품사 사전 규칙과 시범 패키지

한/영 정렬 시스템이란?,
연구 개발의 개요/필요성,
응용 분야,
연구 개발 현황.

1. **한/영 정렬 시스템이란정렬**(alignment)이란 다른 언어로 표기된 단어, 구, 문장에 대하여 서로 관계가 있는(의미가 비슷한) 것끼리 연결하여 주는 행위를 뜻합니다. 예를 들어

나는 학교에 간다.
I am going to school.

라는 한/영 두 문장을 구 단위로 정렬하면



와 같이 됩니다.

한/영 정렬 시스템은 말뭉치에서 자동으로 학습된 **확률** 모델을 이용하여 한국어와 영어의 잉국어(bilingual) 문서에서 서로 대응하는 문장, 구, 단어 등을 찾아 내는 시스템입니다.

2. **연구 개발의 개요/필요성** 이러한 한/영 대응 쌍(Pair)들은 기계 번역기의 기초가 되는 대역 사전 작성의 기초 자료로 사용되며, 언어간의 상이성 조사와 번역의 지침(Guideline) 등을 제공하게 됩니다.

정렬 시스템에 대한 연구는 외국에서도 이제 연구 초기 단계에 들어가 있으며 국내에서는 아
그 예를 찾아 볼 수 없고, 또한 상용화되어 있거나 이에 상응하는 수준의 연구는 이루어지지
은 상태입니다.

양국어 말뭉치(bilingual corpus) 구축 및 이를 바탕으로 하는 정렬 연구는 어느 특정한 언어
서 뛰어난 기술을 발전을 이룩하였다고 해서 바로 다른 언어에 바로 적용할 수 있는 것이 아
니다. 즉 두 가지 언어 모두에 해당하는 말뭉치를 구축하고 두 언어 사이의 특징에 부합되는
렬 시스템을 개발하여야 합니다.

정렬 시스템은 양국어 말뭉치의 제약으로 인하여 외국에서도 기초적인 연구가 수행되고 있는
단계이므로 이 시점에서 한국어와 영어에 대한 양국어 말뭉치를 구축하고 정렬 시스템을 개
함으로써 앞으로의 응용 및 기술 발전에서 주도적인 위치에 서야 할 것입니다. 국내에서는 정
시스템에 관련된 연구가 아직 수행된 바가 없으므로 그 의의가 더욱 크다고 할 수 있습니다.

3. 응용 분야

지금까지 연구된 정렬 결과를 이용한 응용으로는 우선 기계 번역으로의 직접적인 적용을 들
있습니다. 이외에 단어의 의미 모호성 해소, 대역 사전 구축, 단어 단위로 정렬된 코퍼스를 이
용하여 명사를 구분하는 연구 등이 시도되고 있습니다.

정렬 단위는 주로 문장 단위에서 이루어져 왔으나 단어 단위나 구 단위로 갈수록 더욱 제공하
정보가 많아지므로 응용의 가능성도 그만큼 커지게 됩니다. 이와 관련해서는 영어와 프랑스(
사이의 명사구 매칭에 관한 연구가 수행된 바가 있습니다.

현재 구문 정보가 포함된 양국어 상에서 두 언어 사이에 구조적 매칭을 시도하는 연구도 수행
되고 있으며, 앞으로 양국어 말뭉치(biligual corpus)가 증가함에 따라 이와 관련한 연구가 증
가할 전망입니다.

4. 연구 개발 현황

최근 들어 미국과 유럽을 중심으로 말뭉치에 기반한 기계 번역의 연구 결과 등이 나오고 있으
그 응용으로 문서 정렬이 연구되고 있습니다. 그러나 외국에서도 많은 연구 결과를 찾아 볼
수는 없고 이제 발아기에 해당하는 시기입니다.

문장 단위 정렬은 AT&T Bell Lab 이나 Xerox Palo Alto Research Center에서 연구되어 왔
며 95% 이상의 높은 성능을 보이고 있습니다. 그 반면 단어 단위 및 구 단위 정렬은 IBM
Watson Research Center에서 연구되고 있으나 정확한 성능은 발표되지 않았습니다. 특히
재까지의 연구는 영/프, 영/독과 같이 같은 어족에 속하는 언어들에 대해 주로 수행되어 왔고
한/영의 경우와 같이 다른 어족에 속하는 언어 쌍에 대한 연구는 활발히 진행되지 못한 상태
니다.

세부 항목 :
- 사전 개발 및 관리 시스템
- 태거
- 구문 트리 태거
- 한/영 정렬 시스템
- 문서 구조 표현을 위한 표준화
- 한국어 입출력 표준 환경
- 균형화 코퍼스 구축
- 품사 사전 규칙과 시범 패키지

문서구조 표현을 위한 표준화란?,
연구 개발의 개요/필요성,
응용 분야,
연구 개발 현황.

1. 문서구조 표현을 위한 표준화란?

문서구조 표현을 위한 표준화에 관한 연구는 전문 정보의 디지털화에 필수적인 인코딩 기법을 중심으로, 각 유형의 전문정보의 인코딩에 관한 표준을 제안하고자 합니다. 또한 전문정보의 한 예로 정보학 및 전산학 용어 100 term을 선정하여 용어 데이터를 수집하고 이들을 본 연구에서 개발한 DTD를 사용하여 sample 용어 DB를 구축할 것입니다

2. 연구 개발의 개요/필요성/목표

- 전문 (full-text)정보의 디지털화에 대한 필요성
- 전문정보의 구조를 표현할 수 있는 인코딩 방법론 연구
- 각 유형에 따른 전문정보의 인코딩에 관한 표준 제안
- 전산학 및 정보학 분야의 용어 100 term 선정
- 용어데이터 수집 및 개발된 용어데이터베이스 DTD 시험
- Sample 용어데이터베이스 구축
- 용어데이터베이스 관리 도구 개발

- 한글 문헌의 전산처리를 위한 기반 구조를 제공
- TEI의 분석과 한글문헌의 구조분석을 통한 TEI-K의 개발
- TEI-K에 대응한 기본 DTD 개발
- TEI-K를 위한 관리도구 개발
- 정보학과 전산학 전문용어사전 구축을 위한 용어사전 포맷 개발
- 용어데이터베이스를 구축하기 위한 가이드라인 작성
- 개발된 용어데이터베이스 DTD에 근거한 Sample 정보학, 전산학 전문용어 사전 구
- 제안된 SGML-K의 보완

3. 응용 분야

4. 연구 개발 현황

- TEI-K 개발
  1. 전문정보의 표준화 동향 분석
  2. SGML 분석 및 DTD 관련 자료 수집

     ISO 8879, KSC 5913, ISO 12083, CALS , TEI etc.

  3. SGML 관련 도구 및 어플리케이션 분석

     각 기능별 요구사항 도출

  4. 문헌의 유형별 문헌구조 분석이 진행중

     단행본, 정기간행물, 기술보고서, 사 전, 용어데이터베이스 등

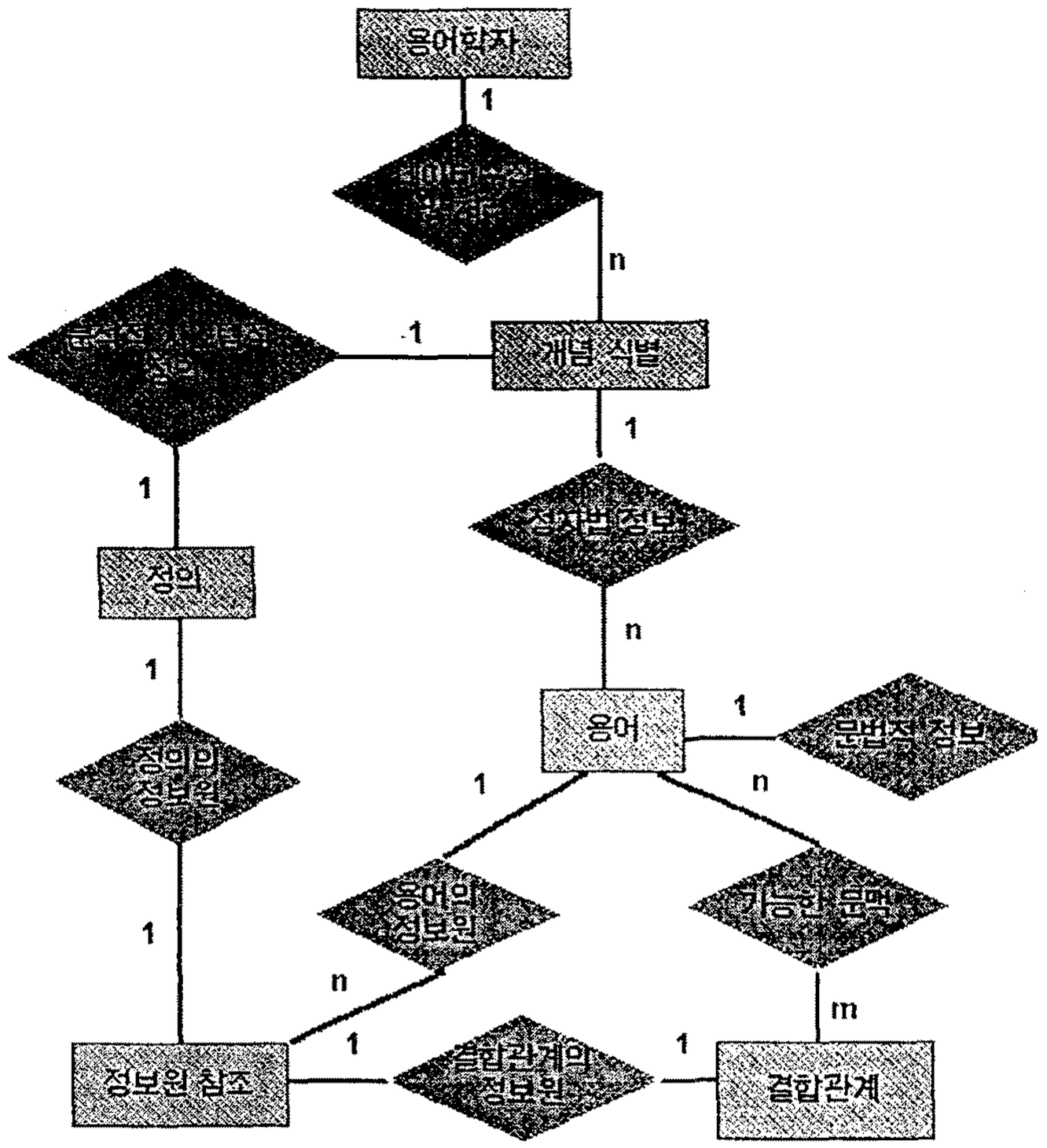
  5. 수식, 테이블, 그래픽 인코딩 기법 분석중
  6. TEI 분석중
- Sample 용어데이터베이스 구축
  1. 용어작업 및 용어학적 편집법에 대한 이해
  2. 100 term 선정

     ISO 2382, INSPEC Thesaurus, LC Subj ect Heading 등 참조

  3. 용어데이터베이스 구축 과정

     1. 정보원 리뷰 및 선정
     2. 용어, 용어에 대한 정의, 설명, 예제 등 용어데이터 수집
     3. 개념 구조 (system of concept) 개발
     4. 용어간 대등 관계 설정
     5. 용어정보 및 개념구조 기록
     6. 용어데이터 수정

  4. Example of an E-R Diagram for a Terminological Data base



5. TEI의 TDB (Terminological DataBase) DTD와 TIF (Term inology Interchange Format) 비교
6. 용어데이터베이스 DTD 구조의 특징

    1. flat 구조와 nested 구조를 지원
    2. 내포규칙과 인접규칙 적용
    3. 데이터 카테고리 적용 (ISO/DIS 12620 Terminology – Computer Application – Data Categories)

7. Sample 용어데이터베이스 DTD 개발
8. 용어데이터 수집중
9. 수집된 용어데이터의 의한 DTD 검증 및 보완
10. 용어데이터베이스 관리 도구 개발중
11. 용어데이터베이스 구축을 위한 가이드라인 작성시 필요 한 데이터 수집

    ISO TR 12618

12. 새로 선정된 type 값

1. <term type = 'hanja'>
2. <term type = 'romanize'>
3. <term type = 'polyseme'>

최종 수정일: 1997년 4월 7일

KLE Administrator

세부 항목 :
사전 개발 및 관리 시스템
태거
구문 트리 태거
한/영 정렬 시스템
문서 구조 표현을 위한 표준화
한국어 입출력 표준 환경
균형화 코퍼스 구축
품사 사전 규칙과 시범 패키지

한국어 표준 입출력 환경이란?.
연구 개발의 개요/필요성.
응용 분야.
연구 개발 현황.

1. 한국어 표준 입출력 환경이란?

가. '정음형' 한글 코드 지원

국어 정보 처리는 문장을 표현하는 문자열에서 언어 정보를 추출해서 이를 처리하는하는 것이다. 언어 정보를 가장 충분하게 표현하는 정음형 코드로 된 자료를 입력하거나 출력하기 위한 기반 환경을 제공한다.

나. 지원 시스템 환경

유닉스 시스템에선 엑스 윈도우 환경을 기반으로 해서 엑스텀(xterm)에서 정음형이 지원되도록 개발하고, 편집기는 비아이(vi)에서 정음형이 지원되도록개발하였다.
피시(PC)에서는 '96년 3차년도에서 개발할 예정이다. 윈95 환경에서 사용 가능한 편집기를 지원다. 이외에도 JAVA등에서 지원이 되도록 한다.

다. '정음형'은 국어정보를 표현하는 그릇

정음형은 낱자소 표현을 기본으로 한다. 그래서 국어정보처리의 모든 응용을 만족시킬 수 있다. 따라서 정음형 코드로 국어정보 자료를 표현하고 각 종 국어정보처리 소프웨어는 이로 이 '정음형' 코드를 지원해야 한다. 현대 한글은 물론 옛한글 전부 한자 특수 문자들을 모두 표현할 수 있는 편집기가 필요하다. 또한 이제 까지 나온 모든 코드를 번역해 주는 기능도 필요하다.

2. 연구 개발의 개요/필요성

가. 언어처리 중심 응용 지원

국어 정보처리는 언어 처리 중심의 응용이다. 이러한 응용들은 문자처리 중심 응용과는 ! 리 형태소 해석을 통하여 날자소에 숨어 있는 언어 정보들을 찾아 내는 응용이다.

나. 국어정보처리에 최적한 한글 코드'정음형' 개발

따라서 언어 정보가 담길 수 있도록 한글 코드를 개발하는 것이 필요하다. 또한 여기에는 기존의 한글 코드로는 국어정보를 처리하기가 어렵다는 것도 포함되어 있다. 본 연구는 (러한 요구를 가지고 국어정보처리에 최적한 한글 코드 "정음형"을 개발하였다.

다. 훈민정음 창제 원리의 규명

여기서 정음형이라 함은 훈민정음 창제 원리를 따르고 있음을 뜻하며, 창제 원리란 훈민; 음 해례에서 정의한 28자의 기본 문자집합과 나머지 규칙을 말한다. 이들에 의거해서 훈 정음은 약 399억 음절자를 생성할 수 있다.

라. 정음형 소프트웨어 부품공장 개념 구현

정음형은 일반화된 한글 코드가 아니다. 그래서 이로써 새로운 소프웨어를 개발하려거나 기존 소프웨어를 수정하려고 한다면 계층 구조로 만들어진 소프웨어를 하나의 소프트웨(부품 공장 개념으로 제공하며 이를 이용하여 응용 계층에서는 쉽게 개발할 수 있다. 다시 말해서 기본 라이브러리를 개발하고 최종적으로 종합 라이브러리를 구축한다.

마. 정음형 한글 플랫폼과 편집기

유닉스 환경에서 xterm을 hunterm으로 개발하고 vi를 hunvi로 개발하였다. 피시에서 윈 95에서 플랫폼을 만들고 edit를 정음형이 지원되도록 할 예정이다.

3. 응용 분야

가. 언어처리 중심 응용

정음형 코드는 낱자에 숨어 있는 형태 정보를 가장 잘 지원해 주는 코드이기 때문에 언어처리와 같은 응용에서 최적한 코드이다. hunterm은 정음형 한글 터미널을 지원 하는 툴킷 품이며 hunvi는 정음형 코드를 지원하는 소프트웨어 개발이나 자료 입력에 사용한다.

나. 워드 프로세서등 문서처리

정음형 코드 체계는 연필로 글을 쓰듯이 쓸 수 있다는 요구는 충족한다. 그래서 초등학교 학생에서 만화가에 이르는 다양한 계층의 글자 표현 요구를 지원하는 워드 프로세서 개발이 가능하다. 정음형 체계에서 현대 한글과 옛한글의 경계가 없다. 왜냐하면 훈민정음 창제 원리를 따르기 때문에 그러하다. 이러한 응용 분야 지원에 높은 적합성을 가지고 있다.

다. 한글의 과학성 이해에 활용

한글의 과학성을 보여 주기 위한 교육 도구로도 사용할 수 있다. 우리는 이제 까지 완성형과 조합형이라는 단순한 논쟁에 관심을 가져왔다. 그러한 논쟁은 훈민정음 창제 원리를 이해하지 못하였을 때 가능하다. 훈민정음 창제 원리를 이해하고 나면 그러한 논쟁은 더 이상 할 가치가 없다는 사실을 알게 된다.

훈민정음 해례는 399억 음절자를 생성할 수 있도록 정의하였다. 현대 한글이 11172 자를 쓰는 것은 인쇄술의 영향도 없지 않았을 것이다. 활자 시대가 아닌 컴퓨터 시대 즉 워드 프로제서를 사용하고 있는 시대에 그러한 활자 인쇄를 위한 글자 수 제한은 필요 없다. 정음형 코드 체계는 바로 이러한 내용을 깨우쳐 줄 수 있다. 훈민정음의 과학성은 다음 자료에서 우선 찾아 볼 수 있다.

- 훈민정음 창제 원리의 공학화에 기반한 한글 코드의 발전 방향

4. 연구 개발 현황

가. 소스 코드와 실행 프로그램

- 소스 프로그램 디렉토리
  * 각 디렉토리에 README가 있다.
    - hunterm.src : hunterm 소스 코드
    - hunvi.src : hunvi 소스 코드
    - conv : 한글 코드 번역기 소스 코드
    - lib : 정음형 기본 라이브러리 함수의 소스 코드
- demo 디렉토리: solaris 2.4용으로 실행 프로그램 화일
  - hunterm : 정음형 코드를 지원하는 xterm

- hunvi : 정음형 코드를 지원하는 vi
- ks2j : 완성형 코드를 정음형 코드로 변환하는 프로그램
- libj.o : 정음형 코드를 위한 기본 함수 및 복합 함수

나. hunterm용 자소형 글자꼴

- smj.bdf
- smj.sdf
- smj.pcf

글자꼴을 설치하는 방법은 다음과 같다.

1. mkfontdir
2. xset +fp
3. xset rehash

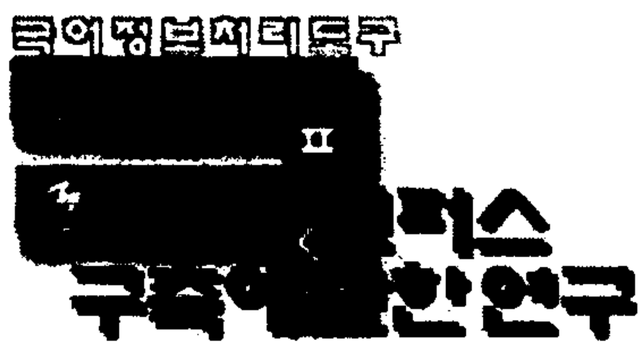
다. 확장자 설명

- *.hun : 정음형 코드로 작성된 화일
- sample1.hun : 세종어제 훈민정음 서문
- sample2.hun : 석보상절

라. 정음형 코드 표

- 정음형-1995는 훈민정음 해례에서 정의한 초성 17자, 중성 11자에 현대에 단모음을 한 넉 자 (ㅐ ㅒ ㅔ ㅖ)를 포함한 15자, 종성 11자이다.

  ISO 646에 근거한 배치표는 다음과 같다.
  - 초성자 : 0xb0 - 0xc1
  - 중성자 : 0xd1 - 0xdf
  - 종성자 : 0xe1 - 0xf1
- 한자는 현재 지원하지는 않지만 접두 코드 을 가지며 충분히 표현할 수 있도록 3 바이트 코드로 한다. 나머지 문장 기호등 특수 문자는 0xf4에서 0xfe까지 접두 코드를 가다. 이들은 현재 계속 추가 중이다.



균형화 코퍼스란?, 연구 개발의 개요/필요성,
응용 분야, 연구 개발 현황,

1. 균형화 코퍼스란?
2. 연구 개발의 개요 /필요성
3. 응용 분야
4. 연구 개발 현황

최종 수정일: 1996년 9월 5일

KLE Administrator

세부 항목 :
사전 개발 및 관리 시스템
태거
구문 트리 태거
한/영 정렬 시스템
문서 구조 표현을 위한 표준화
한국어 입출력 표준 환경
균형화 코퍼스 구축
품사 사전 규칙과 시범 패키지

연구의 목표 및 범위,
연구 개발의 개요/필요성,
국어 형태 품사 태그 규격,

1. 연구의 목표및 범위 본 연구의 목표는 국어정보베이스의 세부 과제의 시스템들이 정보를 공유하고 한 과제의 결과를 다른 과제에서 사용할 수 있도록 품사 분류 변환기를 개발하는데 있다. 또한, 각 시스템의 품사 분류를 비교 연구하여 체계적인 품사 분류 규격을 마련하여 국어 정보 처리의 효율적인 연구협력체계를 확립하는데 있다. 연구의 내용 및 범위는 다음과 같다.

- 국어정보베이스 세부 과제 시스템의 품사 분류 수집
  - 한국어 형태소 해석기(포항공대)
  - 한국어 태깅 시스템(한국과학기술원)
  - 한국어 정보 획득 도구(고려대)
- 각 시스템의 품사 분류 비교 분석
  - 품사 비교 테이블 작성
- 품사 분류 규격 제시
  - 형태론적 품사 분류 체계 정립
  - 통사론적 품사 분류 체계 정립
  - 계층적 분류 체계 정립
- 품사 분류 변환을 위한 사전 작성
  - 표제어
  - 한국어 형태소 해석기의 품사
  - 한국어 태깅 시스템의 품사
  - 한국어 언어 정보 획득 도구의 품사
  - 국어 형태.구문 태그 규격
- 각 시스템간의 품사 분류 변환기 개발
  - 한국어 형태소 분석기와 한국어 태깅 시스템간의 변환기
  - 한국어 태깅 시스템과 한국어 정보 획득 도구간의 변환기

2. 연구 개발의 개요/필요성 국어정보베이스는 한국어의 체계적인 연구를 위한 언어 정보 및 기반 기술을 연구하고 개발함으로써 컴퓨터를 이용한 한국어의 처리와 한국어의 기초 연구를 촉진하고자 하는 목적을 가지고 기초 자료의 축적과 기반 연구를 일관된 환경에서 집중적으로 수행하기 위한 연구 모델이

따라서, 세부 과제에서 개발되는 시스템들은 각각의 결과물을 서로 이용하고 공유할 수 있어야 하며 구축된 지식베이스를 같이 사용하여야 한다.

그러나, 국어정보베이스의 세부 과제들인 한국어 형태소 해석기(포항공대), 한국어 태깅 시스템(한국과학기술원), 한국어 정보 획득 도구(고려대) 등은 각 시스템에서 사용하고 있는 품사 분류가 서로 다르기 때문에 한 시스템의 결과를 다른 시스템에서 그대로 사용할 수 없다. 시스템의 용도와 분야에 따라서 나름대로 품사를 분류하고 있기 때문이다 . 국어정보베이스의 궁극적인 목적을 이루기 위해서는 일단 최소한 시스템들간의 정보교환을 위한 품사 변환 방안이 연구되어야 하며 궁극적으로 시스템들이 공통적으로 사용 할 수 있는 품사 분류 규격이 마련되어야 한다.

품사론은 문장론과 더불어 문법의 근간을 이루고 있다. 품사 정보는 형태소 해석, 태깅, 구문 해석을 잘 하기 위한 가장 기초가 되고 중요한 정보이므로 품사에 대한 연구는 한 국어 정보처리 시스템 구축의 출발점이 된다.

신뢰성 있는 품사의 분류는 한국어 형태소 해석기, 한국어 태깅 시스템, 한국어 정보 획 득 도구, 한국어 구문 해석 등의 한국어 정보처리 시스템의 성능을 향상 시킨다. 즉 품 사 분류는 한국어 정보처리의 기반 기술이자 요소 기술이므로 이의 연구를 통해서 한국어 언어 처리 기술 향상을 이루고 고도화를 이룩하고자 한다. 또한 한국어 전자사전 구축에 기여하고자 한다.

세부 과제들간에 한 과제의 결과를 다른 과제에서 이용할 수 있도록 품사 변환기를 작성 하고 국어정보베이스에서 표준으로 사용할 수 있는 품사 분류 규격을 제시함으로써 국어 정보처리 효율적인 연구협력체계를 확립한다.

3. 국어 형태 품사 태그 규격

상위 분류	태그
기호(s)	sp 쉼표, sl 여는 따옴표 및 묶음표, sd 이음표, su 단위 기호, sf 마침표 sr 닫는 따옴표 및 묶음표, se 줄임표, sy 기타 기호
외국어(f)	f 외국어
서술성 명사 (ncp)	ncpa 동작성 명사, ncps 상태성 명사
비서술성 명사 (ncn)	ncn 비서술성 명사
고유명사(nq)	nq 고유명사
의존명사(nb)	nbu 단위성 의존명사, nbn 비단위성 의존명사
대명사(np)	npp 인칭대명사, npd 지시대명사
수사(nn)	nnc 양수사, nno 서수사
동사(pv)	pvd 지시 동사, pvg 일반 동사
형용사(pa)	pad 지시형용사, paa 성상형용사
보조용언(px)	px 보조용언
수식언(m)	관형사(mm), mmd 지시관형사, mma 성상관형사
부사(ma)	mad 지시부사, mag 일반부사, maj 접속부사
감탄사(ii)	ii 감탄사

조보지

Korean Information Base I

세부 항목 : 기초 국어 정보 베이스
국어 정보 처리 도구
용어 사전

전문 용어 사전

분류 체계에 기반한 대역어 사전

형태소 분석 사전 및 사전 편집기

전문 용어는 다시 대역 전문 용어와 한국어 문예 분야별 전문 용어로 나누어집니다. 검색은 WWW 인터페이스를 통해 이루어지게 되어 있으며, 이를 관리 및 유지하기 위한 기능은 사전 개발 및 관리 시스템에 의해 구현됩니다.

분류 체계에 기반한 대역어 사전은 어휘 분류 체계(ontology)에 기반하여 한국어와 영어간의 대역어를 체계화한 사전으로서, 현재 자료 수집 및 분류 체계 구성 중에 있으므로 검색 인터페이스는 향후 사전의 구축 진척에 따라 구현될 예정입니다.

검색할 어휘 및 형태소를 입력으로 받아 형태소 분석 사전에 있는 정보를 검색하는 기능입니다. 사전 편집기는 형태소 분석용 사전의 내용을 오프 라인으로 편집하는 기능을 제공하고 있습니다.

최종 수정일: 1997년 3월 20일

KLE Administrator

조보지

Korean Information I

세부 항목 :전문 용어 사전
분류 체계에 기반한 대역어 사전
형태소 분석 사전 및 사전 편집기

전문 용어 사전 구조 예(전기/전자)	anode paralleling reactor 대역:양극 병렬 리액터 동의: 陽極竝列ー
전문 용어 사전 예(문예)	anode pulsing 대역:양극 펄싱 동의: 陽極ー 속성:N 대역:家計調査 동의:계열분석, 횡단분석 출전:한국민족문화대백과사전 권1 p22 속성:I 대역:可考 동의:판례집 출전:한국민족문화대백과사전 권1 p23
전문 용어 사전 속성 코드(문예)	A : 인명, 신의 이름 B : 지명 (산, 강, 마을이름 ...), 행정구역명, 유적지, 고분군, 온천, 광산

C : 회사명

D : 기관.단체명 (학교, 병원, 방송사...), 정치기구, 향교, 서원, 종교종파 및 교회

E : 사건명 (역사적인 사건, 대회명 ...), 기념일, 절기, 행사명, 종교기념일

F : 상품명: 토산물, 특정악기명, 특정음식명, 천연기념물, 화폐, 문양이름

J : 사회제도명, 연호

H : 건축명 :건물, 도로, 고분, 조각품, 사찰

I : 제목 :노래, 영화, 문집, 연극 등의 제목, 악곡 이름, 굿이름, 특정놀이이름

G : 전문용어 : 법률명, 종명, 학명, 이론, 처방법

K : 부족, 종족명

N : 일반명사류 : 개념어...

X : 동물명

Y : 식물명

Z : 직함, 관직명, 직업명, 친족어 등의 인명을 제외한 사람 지칭어

최종 수정일: 1997년 3월 20일

KLE Administrator

조보자

Korean Information I

세부 항목 : 전문 용어 사전
분류 체계에 기반한 대역어 사전
형태소 분석 사전 및 사전 편집기

분류 체계에 기반한 대역어 사전

분류 체계에 기반한 대역어 사전의 목적/목표/역할/설명 등은 향후 만들어질 예정입니다.

●책임자:

국어공학센터 국어정보베이스 분류체계 기반 대역어 사전(세부과제) 책임자

●실무자:

국어정보베이스 분류체계 기반 대역어 사전(세부과제) WWW 담당자

최종 수정일: 1997년 3월 20일

KLE Administrator

조보지

Korean Information I

세부 항목 : 전문 용어 사전
분류 체계에 기반한 대역어 사전
형태소 분석 사전 및 사전 편집기

형태소 분석 사전 및 사전 편집기란?

형태소 분석을 하기 위해서는 형태소 분석용 전자사전 속에 분석을 하고자 하는 어절에 대한 어휘들과 그 어휘에 대한 형태소 수준의 정보가 입력이 되어 있어야 한다. 현재 국내에 있는 형태소 분석 알고리즘의 발달이 한계에 와 있는 만큼 형태소 분석기의 성능은 사전에 달려 있다고 하겠다. 사전 편집기는 이러한 전자 사전의 구축과 수정, 검색의 편의를 위한 도구로서 오프라인 편집기와 온라인 편집기가 있다.

형태소 분석 사전 사양

가. 전자 사전의 기능

- 응용 프로그램과의 독립성: 물리적 저장 구조와 인덱스 구조의 분리
- 다계층성: 형태소 분석 정보뿐 아니라 구문분석, 의미분석에도 사용될 수 있는 다계층 구조

나. 전자사전의 구조

- 선형 해쉬 기법을 사용
- 사전 헤드와 사전 화일의 분리 구조

다. 전자사전 표제어

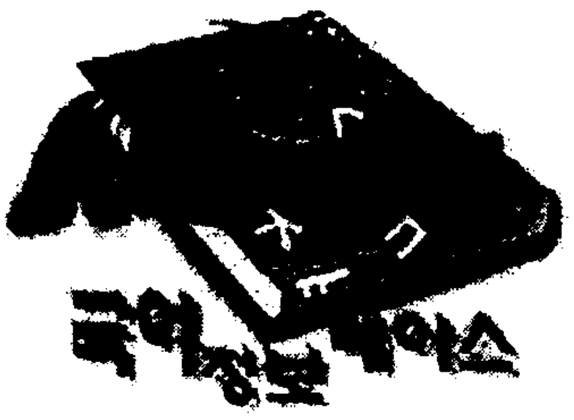
형태소 분석 사전에서 중요한 것은 표제어의 개수와 적절한 표제어의 선정이다. 본 형태소 분석사전은

- 표제어 선정: Tree bank팀과 전문용어 DB팀에서 제공한 말뭉치에서 빈도수를 기반으로 선정
- 표제어 개수: 기능어 1900여개
- 표제어 개수: 실질어는 10만여개를 목표로 현재 35000여개

사전 편집기 사양	가. 사전 편집기란
응용 프로그램 인터페이스 사양	사전 편집기란 전자사전에 표제어를 삽입/삭제하거나 기존 어휘의 검색, 수정 기능을 제공하는 사전 관리 도구이다. 현재 사전 편집기는 off line으로서 새로운 표제어를 삽입하는 기능과 기존 어휘의 검색 기능을 제공한다. 가. 사전 편집기 off line 사전 편집기는 insert와 search기능을 제공한다. ●표제어 입력: 사전을 edit한 후 사전 헤드를 매개변수로 insert루틴을 수행 ●사전 검색: 사전 헤드를 매개변수로 실행시킨 뒤 interactive 방식으로 검색을 수행 나. 전자사전 응용 프로그램에서의 전자사전 이용은 다음과 같은 방법으로 수행된다. ●사전 초기화 루틴을 수행한다. 사전 헤드를 매개변수로 사용 ●사전 헤드와 표제어를 매개변수로 검색 루틴을 수행 ●검색 결과를 이용

최종 수정일: 1997년 4월 7일

KLE Administrator



숙련자

Korean Information Base

세부 항목 : 기초 국어 정보 베이스
국어 정보 처리 도구
용어 사전

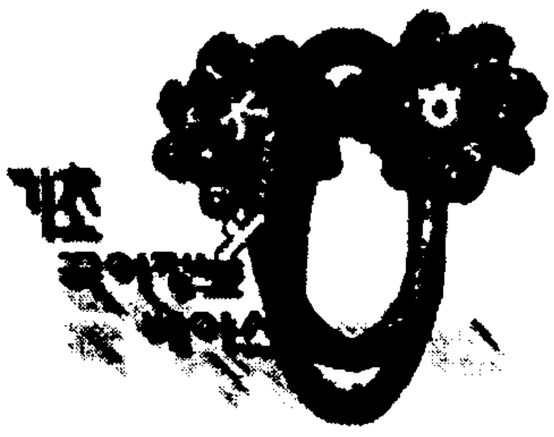
국어 정보 베이스는 국어공학센터를 주축으로 국어 정보 처리 기술 개발 과제의 일부로 추진
것으로서, 국어 관련 연구자 및 개발자에게 도움을 주고 국어 공학 발전에 이바지하기 위하여
국어에 관한 각종 자료와 정보 및 도구 등을 모아 놓은 데이타베이스입니다.

연구 결과발표자료

- KIBS 중간 연구 결과 발표 자료
- KIBS 최종 연구 결과 발표 자료

최종 수정일: 1997년 3월 20일

KLE Administrator



숙린지

Korean Information Base I

세부 항목 :기초 국어 정보 베이스
국어 정보 처리 도구
용어 사전

기초 국어 정보 베이스는 한국어에 관련된 자료를 모아 놓은 곳으로서, 각종 문서로부터 수
집한 말뭉치, 여러 발성자로부터 수집한 음성 자료 및 한글 오프라인 필기체 인식을 위한 글
자 자료가 있습니다.

한국어 말뭉치 — 기초 말뭉치, 태깅된 말뭉치, 구문트리 태깅된 말뭉치, 범주화된 말뭉치 및 운형 자료 모음을 검색할 수 있습니다.

음성 자료 모음 — 발성자와 발성 목록에 따른 음성 자료를 검색할 수 있습니다.

글자 자료 모음 — KSC-5601 완성형 한글 2,350자 1,000벌에 대한 글자 자료를 검색할 수 있습니다.

최종 수정일: 1997년 4월 7일

KLE Administrator

속련지

Korean Information Base

세부 항목 :한국어 말뭉치
음성 자료 모음
글자 자료 모음

한국어 말뭉치는 현대 한국어 문서를 대상으로 문서 처리 시스템에서 사용하려는 목적에 !
게 여러가지의 말뭉치를 모아 놓은 것입니다. 이러한 말뭉치들은 언어 공학자에게 중요한
구 자료가 될 뿐만아니라, 통계적 모델의 수립이나 문법과 관련된 제 이론의 개발, 음성에.
의 운율적인 현상 탐구 또는 분석 모델의 적합성 평가 및 비교 등에 있어 매우 유용합니다.

기초 말뭉치 — 기초 말뭉치를 검색할 수 있습니다.
태깅된 말뭉치 — 형태소 분석 후에 생기는 품사 중의성을 해소하여 품사 태그를 붙여 놓은 대량의 태깅된 말뭉치를 검색할 수 있습니다.
구문트리 태깅된 말뭉치 — 대량의 텍스트에 구문 정보를 부여한 구문트리 태깅된 말뭉치를 검색할 수 있습니다.
범주화된 말뭉치 — 하위 범주화 규칙에 따라 범주화된 말뭉치를 검색할 수 있습니다.
운형 자료 모음 — 문장의 형태에 따라 분류된 운형 자료를 검색할 수 있습니다.

최종 수정일: 1997년 4월 7일

KLE Administrator

숙련자

Korean Information Ba

세부 항목 : 기초 말뭉치
태깅된 말뭉치
구문트리
범주화된 말뭉치
문형 자료 모음

기초 말뭉치는 가공되지 않은 말뭉치로서, 문서의 장르 및 문서 형태에 따라 균형 있게 수집된 텍스트 모음입니다. 이러한 기초 말뭉치는 WWW 인터페이스를 통해 검색 기능을 제공 받거나 혹은 텍스트 데이타 베이스 관리 시스템에 의한 검색 기능을 제공 받습니다. 텍스트 데이타 베이스 관리 시스템에서는 UNIX 화일 시스템을 이용한 색인과 검색 방법을 모색하고 있습니다. 한편 화일 서술자(file descriptor)를 이용한 검색은 DBMS를 이용함으로써 효율적으로 이루어질 수 있습니다.

말뭉치 화일 (500개중 50개)

kck_001a.wan 한국 [illegible]
kck_002a.wan 생각하는 지구과학-교과총서(7)
kck_003a.wan 북한의 언어생활
kck_004a.wan 바흐친과 대화주의
kck_005a.wan 스페인문학사
kck_006a.wan 중국사상
kck_007a.wan 박수칠 때 떠나라
kck_008a.wan 마음 비우기
kck_009a.wan 해방공간의 문학 연구 1
kck_010a.wan 즐기면서 배우는 물리학 산책

선택하셨으면 [검색] 을 눌러 주십시오.

기초 말뭉치 화일을 받아 가시려면 여기를 눌러 주십시오.

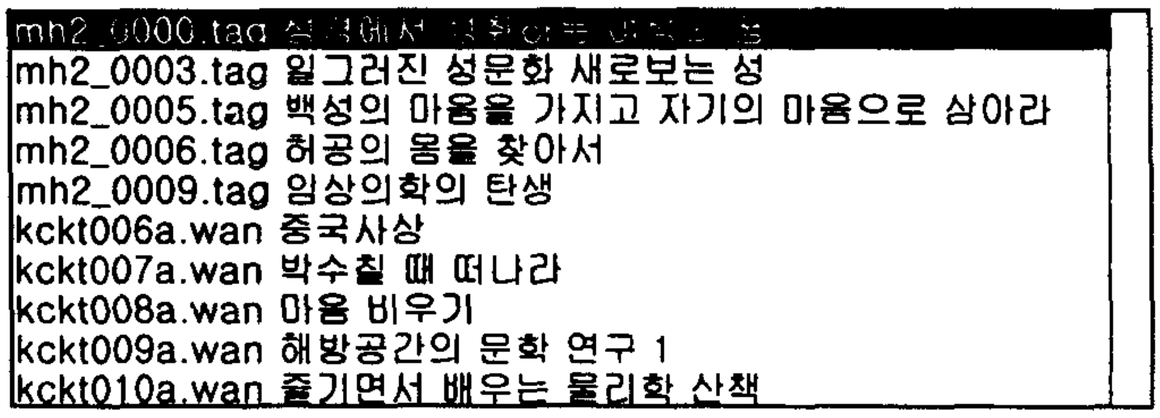
최종 수정일: 1997년 4월 7일

KLE Administrator

세부 항목 : 기초 말뭉치
태깅된 말뭉치
구문트리
범주화된 말뭉치
문형 자료 모음

태깅된 말뭉치는 기초 말뭉치에 대하여 형태소 분석을 수행하고, 이 때 생기는 품사의 중의성 해소하여 품사 태그를 붙여 놓은 말뭉치입니다.

태깅된 말뭉치 화일 (200개중 20개, 앞의 5개는 새로운 태그셋 적용)



선택하셨으면 

을 눌러 주십시오.

태깅된 말뭉치 화일을 받아 가시려면 여기를 눌러 주십시오.

최종 수정일: 1997년 4월 7일

KLE Administrator


숙련자
Korean Information Base
세부 항목 : 기초 말뭉치
태깅된 말뭉치
구문트리
범주화된 말뭉치
문형 자료 모음
구문트리 태깅된 말뭉치는 의존 문법에 기반한 구문 트리 구조로 표현된 말뭉치 이며 구문트
태깅된 말뭉치를 만들기 위해서는 기초 말뭉치를 형태소 분석과 태깅을 한 후 구문트리 태깅
정을 거쳐야 합니다.
이 분야는 향후 확실히 구축될 예정입니다.
말뭉치 화일 (500개중 50개)
T064.new
T065.new
T066.new
T067.new
T068.new
T069.new
T070.new
T072.new
T073.new
T074.new
선택하셨으면
버튼을 눌러주십시오.
구문트리 태깅된 말뭉치 화일을 받아가시려면 여기를 눌러 주십시오.
최종 수정일: 1997년 4월 7일
KLE Administrator

세부 항목 : 기초 말뭉치
태깅된 말뭉치
구문트리
범주화된 말뭉치
문헌 자료 모음

범주화된 말뭉치란 문서를 각 범주별로 분류하여 모아 놓은 것입니다.

말뭉치 화일 (500개중 50개)

kckt005a.wan
kckt006a.wan
kckt007a.wan
kckt008a.wan
kckt009a.wan
kckt010a.wan
kckt011a.wan
kckt012a.wan
kckt013a.wan
kckt014a.wan

선택하셨으면 을 눌러 주십시오.

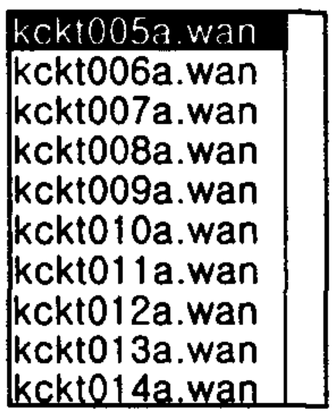
범주화된 말뭉치를 받아 가시려면 여기를 눌러 주십시오.

최종 수정일: 1997년 4월 7일

KLE Administrator

문형 자료 모음이란 각 문형들을 정의하고, 기본 동사 목록을 수집한 다음, 이 분류에 따라 둔 들을 모아 놓은 것입니다.

말뭉치 화일 (500개중 50개)

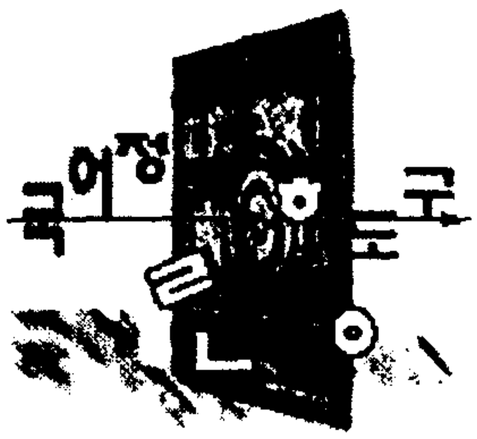


선택하셨으면 검색 을 눌러 주십시오.

문형 자료 모음을 받아 가시려면 여기를 눌러 주십시오.

최종 수정일: 1997년 4월 7일

KLE Administrator



숙린지

Korean Information Base

세부 항목 : 기초 국어 정보 베이스
국어 정보 처리 도구
용어 사전

이 페이지에는 국어 정보를 처리할 수 있는 여러가지 도구들을 모아두었습니다.

사전 개발 및 관리 시스템	: 사전의 개발 및 관리를 위한 환경을 제공합니다.
태거	: 대화 방식 및 화일 방식으로 태깅을 수행합니다.
구문 트리 태거	: 태거의 결과를 입력으로 구문트리 태깅을 수행합니다.
한/영 정렬 시스템	: 한/영 양국어 문서에 대해 문장 단위 정렬 및 구 단위 정렬을 수행합니다.
문서 구조 표현을 위한 표준화 시스템	: 문서 구조 표현을 위한 표준화 시스템 및 전문용어 시범 패키지가 제공됩니다.
한국어 입출력 표준환경	: 한국어 입출력 표준환경에 대한 연구와 그 결과물들을 제공할 예정입니다.
균형화 코퍼스 구축 표준 방법론	: 균형화 코퍼스 구축 표준 방법론 및 시범 패키지가 제공됩니다.
품사 사전 규칙에 대한 시범 패키지	: 품사 사전 규칙에 대한 시범 패키지가 제공됩니다.

최종 수정일: 1997년 3월 20일

KLE Administrator

세부 항목 :
사전 개발 및 관리 시스템
태거
구문 트리 태거
한/영 정렬 시스템
문서 구조 표현을 위한 표준화
한국어 입출력 표준 환경
균형화 코퍼스 구축
품사 사전 규칙과 시범 패키지

텍스트 및 사전 관리 시스템
Text and Dictionary Management System (TDMS)

사전 개발 및 관리 시스템은 사전의 개발과 관리를 용이하게 하기 위한 통합 환경을 제공하고 기존 사전으로부터 사용자의 요구에 맞는 특성을 가진 사전을 작성할 수 있는 기능 등 일반적이고 유용한 기능의 제공을 목적으로 하고 있습니다. 또한, 표준적인 사전 편집기 및 형식 변환을 구현하기 위한 기간 작업으로 표준 사전 기술 언어 (Standard Dictionary Markup Language: SDML)을 설계 하였습니다.

텍스트 데이타 베이스 관리 시스템은 기존의 사전 개발 관리 시스템과 성격이 유사하기 때문에 [텍스트 및 사전 관리 시스템]으로 통합되었읍니다. SDML에 의하여 텍스트 데이타를 정의하고 TDMS에 의하여 관리됩니다.

TDMS는 Windows/NT 서버와 Window95 클라이언트 환경에서 동작합니다. 이곳에서는 데 화면을 볼 수 없읍니다.11

참고자료

- TDMS Overview
- TDMS 구성도
- SDML DTD 정의
- 일반 사전을 위한 SDF 정의
- 일반 사전의 SD 가공
- MATES/EK 사전 변환 작업 예
- 텍스트 코퍼스 및 전자 사전 관리 시스템 설계(보고서)
- 작년 자료

최종 수정일: 1997년 4월 7일

KLE Administrator

숙련자

Korean Information Base

세부 항목 :
- 사전 개발 및 관리 시스템
- 태거
- 구문 트리 태거
- 한/영 정렬 시스템
- 문서 구조 표현을 위한 표준화
- 한국어 입출력 표준 환경
- 균형화 코퍼스 구축
- 품사 사전 규칙과 시범 패키지

직접 문장을 입력하여 태깅을 할 수 있을 뿐만 아니라, 이미 준비되어 있는 데이타를 선택하C 선택된 데이타들이 태깅되는 결과를 볼 수도 있습니다. 그리고 태거를 다운로드받을 수도 있· 니다.

1. 사용자 입력 태깅

입력문장:

입력을 다 하셨습니까 ?

2. 화일단위 태깅

말뭉치 화일

실내원예
집꾸미기
여름음식
전통음식
팥도음식

말뭉치 화일 중의 하나를 선택하셨으면 다음 버튼을 눌러주십시오.

3. 소스 프로그램의 다운로드

소스프로그램을 받아가시려면 여기를 눌러 주세요.

최종 수정일: 1997년 4월 7일

KLE Administrator

세부 항목 :
- 사전 개발 및 관리 시스템
- 태거
- 구문 트리 태거
- 한/영 정렬 시스템
- 문서 구조 표현을 위한 표준화
- 한국어 입출력 표준 환경
- 균형화 코퍼스 구축
- 품사 사전 규칙과 시범 패키지

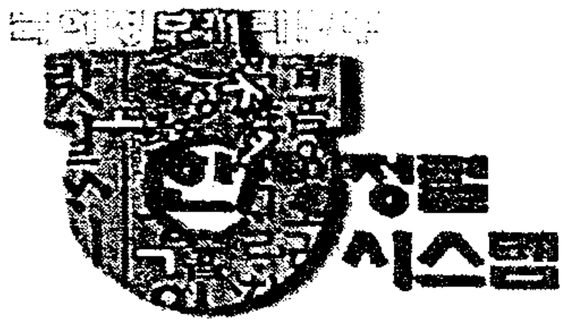
수동으로 태깅한 화일들 어절을 입력하여 구문트리 태깅을 할 수 있을 뿐만 아니라, 이미 준비되어 있는 데이타를 선택하여, 선택된 데이타들이 분석되는 결과를 볼 수도 있습니다. 그리고 구문트리 태거를 다운로드받을 수도 있습니다.

현재는 프로젝트가 진행 중이므로, 데모는 보실 수 없고, 이후에 파싱의 결과가 되는 구문태 결과 화일들을 몇 개 보실 수 있습니다.

- T064.new
- T065.new
- T066.new
- T067.new
- T068.new
- T069.new
- T070.new
- T072.new
- T073.new
- T074.new
- T075.new
- T076.new
- T077.new

최종 수정일: 1997년 4월 7일

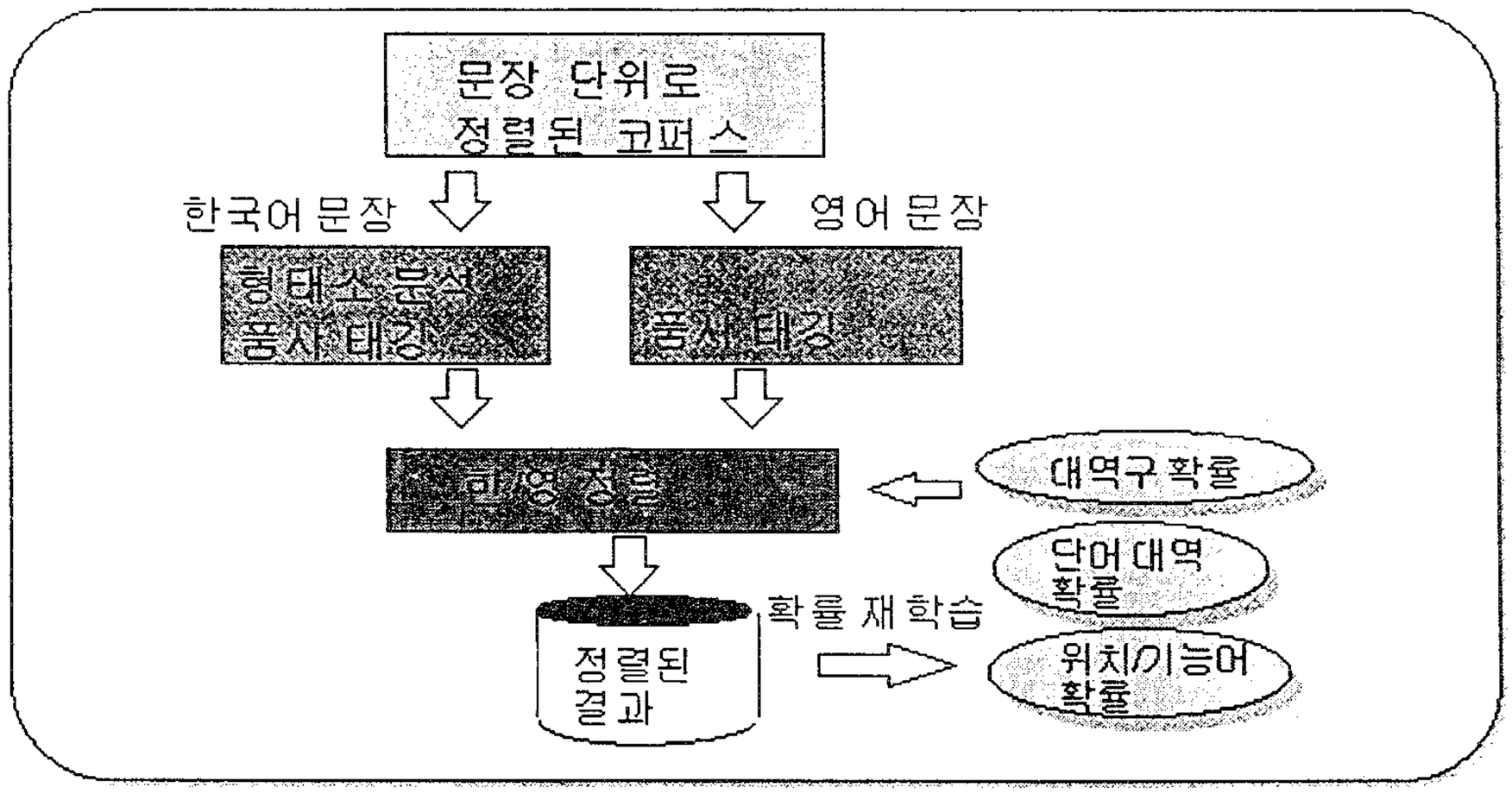
KLE Administrator



세부 항목 :
- 사전 개발 및 관리 시스템
- 태거
- 구문 트리 태거
- 한/영 정렬 시스템
- 문서 구조 표현을 위한 표준화
- 한국어 입출력 표준 환경
- 균형화 코퍼스 구축
- 품사 사전 규칙과 시범 패키지

1. System 구조도

한/영 정렬이란 코퍼스에서 자동으로 학습된 확률 모델을 이용하여 한국어와 영어의 양국어(bilingual) 문서에서 서로 대응하는 문장, 구, 단어등을 찾아내는 시스템입니다.



2. 구축된 영/한 양국어 코퍼스

학습 참고서	영어학습 관련 참고서 5종
시사 잡지	World-News
소설	5권
중-고교 교과서	중-고교 영어 교과서 4종
총 size	9 MB

3. 실험 및 평가

3.1 학습 코퍼스

문서 종류	영어	한국어
중학 영어 교과서	4.64 만 어절	3.21 만 어절
고교 영어 교과서	7.65 만 어절	5.37 만 어절
대입 참고서 및 독해집	7.68 만 어절	5.46 만 어절
기타 일반 서적	5.34 만 어절	3.84 만 어절
총 합계	25.31 만 어절	17.78 만 어절

3.2 모델 평가 : 모델간 비교 실험

모델 1 : 단어간 공기 정보 이용

모델 2 : 모델 1 + 위치/기능어 정보

모델 3 : 모델 2 + 구대역 정보

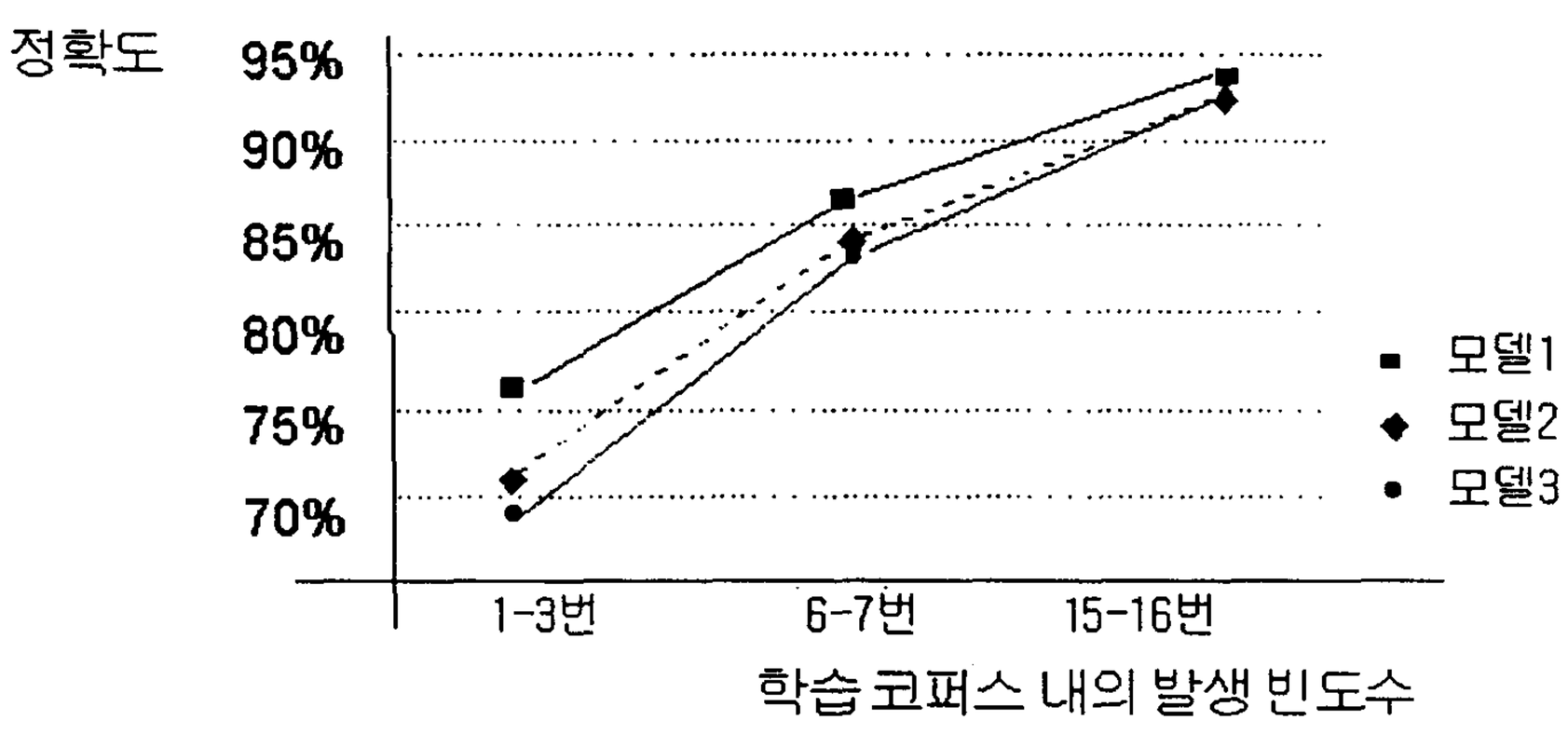
3.3 실험

실험 1 : 최대의 확률 값을 가지는 대역 단어의 정확도

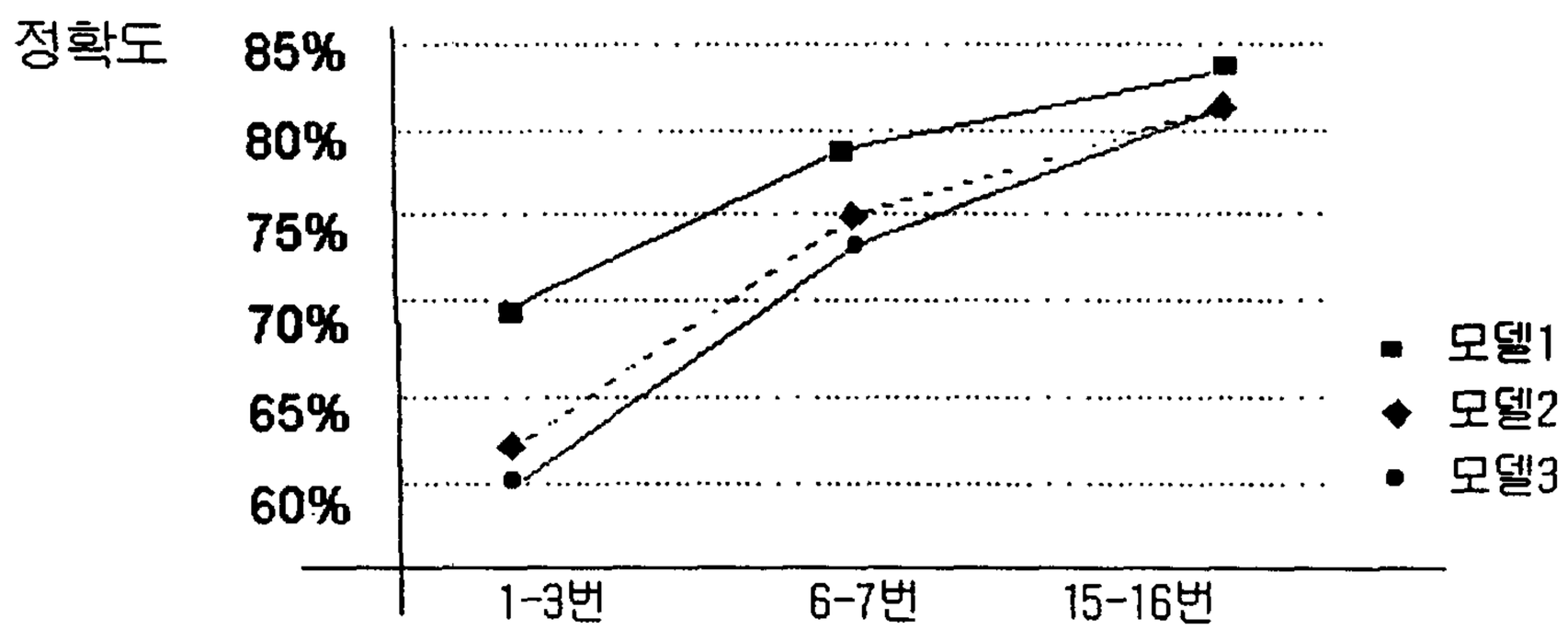
실험 2 : 추출된 대역 단어들의 정확도

3.4 실험 결과

실험 1



실험 2



4. 소스 프로그램의 다운로드

프로그램의 공개가 결정되면 가져가실 수 있도록 올려 놓겠습니다.

최종 수정일: 1997년 4월 7일

KLE Administrator

숙련자

Korean Information Base

세부 항목 :
사전 개발 및 관리 시스템
태거
구문 트리 태거
한/영 정렬 시스템
문서 구조 표현을 위한 표준화
한국어 입출력 표준 환경
균형화 코퍼스 구축
품사 사전 규칙과 시범 패키지

문서구조 표현을 위한 표준화에 관한 연구는 전문 정보의 디지털화에 필수적인 인코딩 기법을 중심으로, 각 유형의 전문정보의 인코딩에 관한 표준을 제안하고자 합니다. 또한 전문정보의 한 예로 정보학 및 전산학 용어 100 term을 선정하여 용어 데이터를 수집하고 이들을 본 연구에서 개발한 DTD를 사용하여 sample 용어 DB를 구축할 것입니다.

최종 수정일: 1997년 4월 7일

KLE Administrator

숙린지

Korean Information B

세부 항목 : 사전 개발 및 관리 시스템
태거
구문 트리 태거
한/영 정렬 시스템
문서 구조 표현을 위한 표준화
한국어 입출력 표준 환경
균형화 코퍼스 구축
품사 사전 규칙과 시범 패키지

본 연구의 내용은 크게 2가지로 분류되는데, 한 가지는 일차년도에 개발한 정음형 입출력 환
을 국어정보처리의 실제 응용에서 활용할 수 있도록 기능을 추가하거나 개선하는 것이고 국(
정보처리에서 공통으로 필요로 하는 기본 루틴들을 라이브러리 형태로 지원하는 것이다.

최종 수정일: 1997년 4월 7일

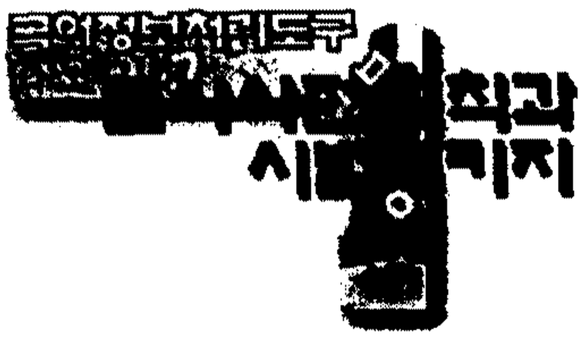
KLE Administrator

세부 항목 : 사전 개발 및 관리 시스템
태거
구문 트리 태거
한/영 정렬 시스템
문서 구조 표현을 위한 표준화
한국어 입출력 표준 환경
균형화 코퍼스 구축
품사 사전 규칙과 시범 패키지

본 연구는 1990년대의 한국어 문어 텍스트류의 성격을 반영하는 균형 잡힌 코퍼스의 구축 방
론의 표준화를 위한 연구를 행한다. 또한 정립된 표준 균형 코퍼스 방법론을 바탕으로 그를 반
영한 기초 데이터베이스와 품사 주석 코퍼스를 구축하는 것으로 목적으로 한다.

최종 수정일: 1997년 4월 7일

KLE Administrator

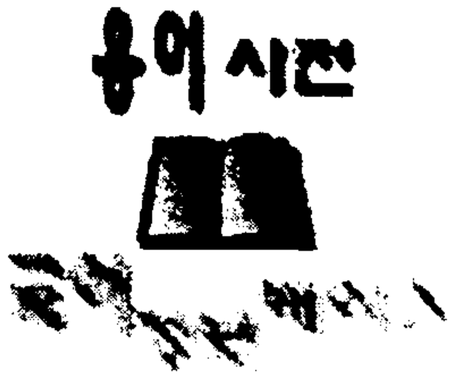


Korean Information Base

세부 항목 :
- 사전 개발 및 관리 시스템
- 태거
- 구문 트리 태거
- 한/영 정렬 시스템
- 문서 구조 표현을 위한 표준화
- 한국어 입출력 표준 환경
- 균형화 코퍼스 구축
- 품사 사전 규칙과 시범 패키지

최종 수정일: 1997년 4월 7일

KLE Administrator



숙련자

Korean Information Base I

세부 항목 : 기초 국어 정보 베이스
국어 정보 처리 도구
용어 사전

전문 용어 사전

분류 체계에 기반한 대역어 사전

형태소 분석 사전 및 사전 편집기

전문 용어는 다시 대역 전문 용어와 한국어 문예 분야별 전문 용어로 나누어집니다. 검색은 WWW 인터페이스를 통해 이루어지게 되어 있으며, 이를 관리 및 유지하기 위한 기능은 사전 개발 및 관리 시스템에 의해 구현됩니다.

분류 체계에 기반한 대역어 사전은 어휘 분류 체계(ontology)에 기반하여 한국어와 영어간의 대역어를 체계화한 사전으로서, 현재 자료 수집 및 분류 체계 구성 중에 있으므로 검색 인터페이스는 향후 사전의 구축 진척에 따라 구현될 예정입니다.

검색할 어휘 및 형태소를 입력으로 받아 형태소 분석 사전에 있는 정보를 검색하는 기능입니다. 사전 편집기는 형태소 분석용 사전의 내용을 오프 라인으로 편집하는 기능을 제공하고 있습니다.

최종 수정일: 1997년 3월 20일

KLE Administrator

세부 항목 :전문 용어 사전
분류 체계에 기반한 대역어 사전
형태소 분석 사전 및 사전 편집기

대역 전문 용어 사전

문예 분야별 전문 용어 사전

대역 전문 용어 사전과 문예 분야별 한국어 전문 용어를 검색할 수 있습니다.

●컴퓨터 분야
●전기 및 전자 분야

●국화 분야

소스파일을 받아가시려면 여기를 Click해 주세요.

최종 수정일: 1997년 4월 7일
KLE Administrator

숙련자

Korean Information I

세부 항목 : 전문 용어 사전
분류 체계에 기반한 대역어 사전
형태소 분석 사전 및 사전 편집기

대역어 사전 검색

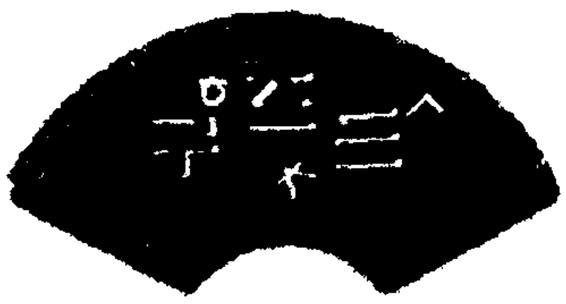
분류 체계에 기반한 대역어 사전은 향후에 만들어 질 계획입니다. 현재는 간단한 텍스트 화면이 제공됩니다.

●대역어 사전 검색

최종 수정일: 1997년 4월 7일

KLE Administrator

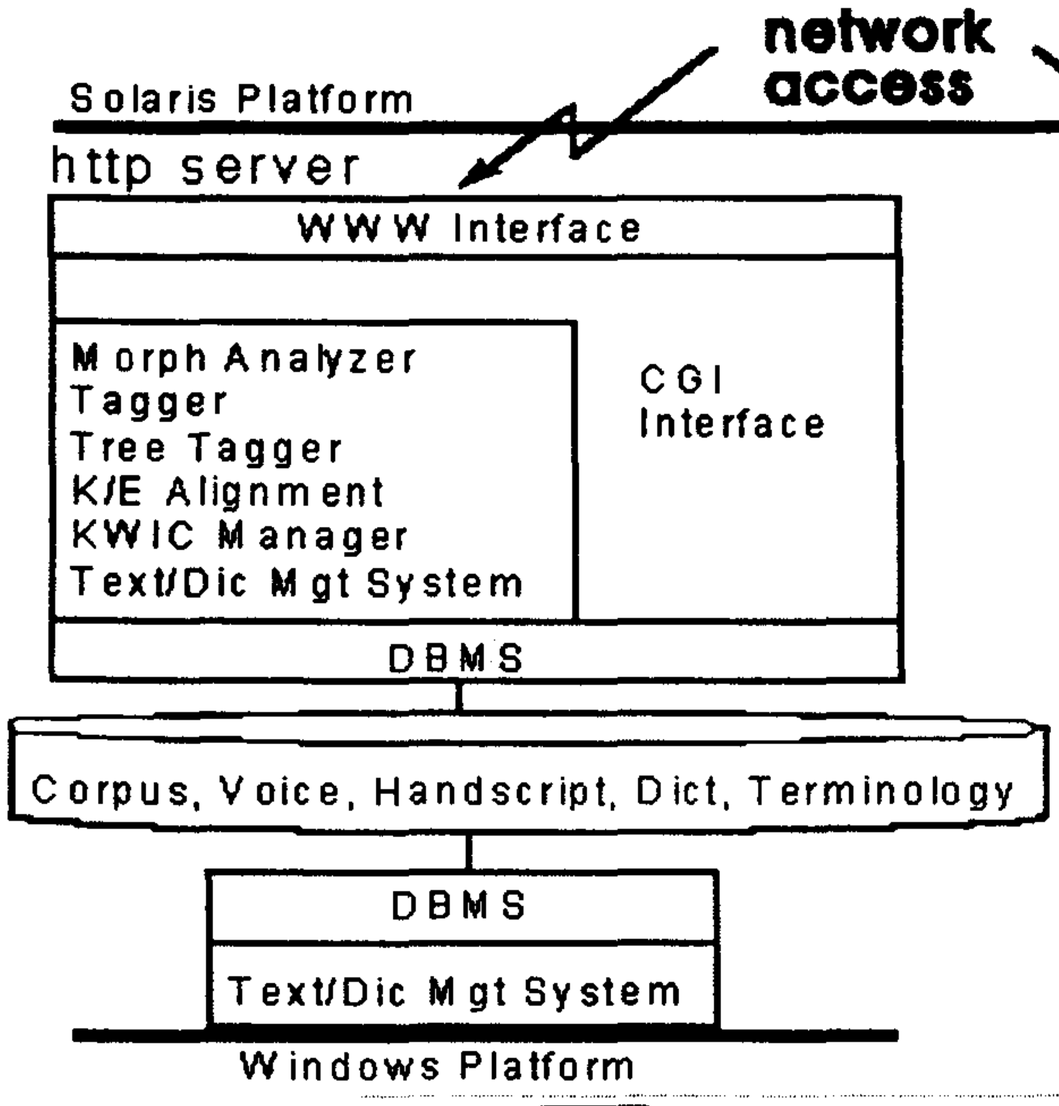

숙란지
Korean Information I
세부 항목 :전문 용어 사전
분류 체계에 기반한 대역어 사전
형태소 분석 사전 및 사전 편집기
형태소의 검색
전자사전에 들어있는 형태소를 검색할 수 있습니다. 그리고 소스 프로그램을 다운로드받을 도 있습니다.
검색할 형태소를 입력하세요.
입력 형태소
입력을 다 하셨습니까?
소스 프로그램의 다운로드
전자사전 프로그램을 받고 싶으면 여기를 Click해 주세요.
최종 수정일: 1997년 4월 7일
KLE Administrator



구조도

Korean Information I

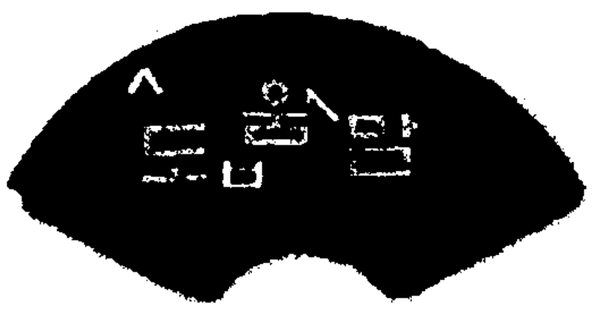
국어 정보 베이스 구조 이 페이지에서는 클릭을 통해서 KIBS의 각 구조로 이동을 할 수 있습니다.



U: Unix
S: Solaris
W: Windows
U/S,W: Developing for Unix first, then for S and W

최종 수정일: 1997년 4월 7일

KLE Administrator



세부 항목 :국어정보베이스 웹 스페이스 구조
기능 페이지
설명 페이지
당부 사항

1. 국어정보베이스 웹 스페이스 구조

이 페이지에서는 국어정보베이스(KIBS) 시스템의 페이지 구성에 관하여 설명합니다. 이 설명은 처음 사용하는 사람들이 효과적으로 사용할 수 있도록 하기 위하여 쓰여졌습니다.

이 페이지에서 모두 설명되는 내용인, KIBS의 웹 스페이스를 간단하고 효과적으로 구성하기 위해 처음으로 씌여진 화이트 페이퍼를 보실 수 있습니다.

1.1 기본적인 페이지 구성

기본적으로 페이지들은 보통 두 가지로 나뉩니다. 하나는 아이콘을 선택해서 각 세부기능을 택할 수 있는 페이지이고, 다른 하나는 초보자들이나 설명을 원하는 경험자들을 위한 기능 설명 페이지들입니다. 편의상 처음의 페이지들을 **기능 페이지**, 설명을 위한 페이지를 **설명 페이지**라고 부릅니다.

기능/설명 페이지의 구분 외에 각 레벨을 구분하기 위하여 다른 색깔을 사용합니다.

0 레벨 (KIBS 홈페이지):엷은 파랑
1 레벨 :엷은 초록
2 레벨 :엷은 노랑
3 레벨 :엷은 빨강

2. 기능 페이지

기능 페이지는 왼쪽에 위치한 진한 색깔의 바아(bar)로 구분을 할 수 있습니다. 이 바아에는 각의 기능이나 설명 페이지로 넘어가기 위한 아이콘이 있습니다. 바아의 가장 위쪽에는 그 페이지를 나타내는 아이콘이 있습니다. 그 다음으로 각각의 기능 페이지를 설명하고 있는 페이로 이동할 수 있는 아이콘이 나옵니다.

몸체에 해당하는 부분에는 다른 세부 기능으로 이동할 수 있는 아이콘이 존재하고, 마지막에 KLE 홈페이지나 국어정보베이스의 홈페이지, 혹은 바로 위 레벨의 기능 페이지로 이동하기 한 아이콘이 위치합니다.

잠깐 실제 기능 페이지 중의 하나를 방문해보십시오. (기능 페이지의 예를 보시고 난 다음에는 브라우저의 백을 이용하셔서 이 도움말로 돌아오십시오.)

실제의 기능 페이지에 나오는 아이콘들은 다음과 같이 사용됩니다.
- 가장 위, 현재의 기능 페이지를 나타내는 아이콘
- 위에서 두번째, 현재의 기능 페이지에 해당하는 설명 페이지
- 마지막 즈음, KLE로 이동하는 아이콘
- 마지막 즈음, KIBS로 이동하는 아이콘
- 마지막 즈음, 바로 위 레벨로 이동하는 아이콘

그 외에 세부 기능 페이지로 이동하는 여타의 다른 아이콘들이 몸체 부분에 나옵니다.

3. 설명 페이지

기본적으로 한 페이지에 관한 설명은 모두 한 페이지에 오도록 페이지를 구성하였습니다. 단 설명이 너무 길어지거나 꼭 필요한 경우에 한해서 하이퍼링크를 두었습니다. 이는 프린팅 시 편의를 위한 것으로 초보자들이 프린터로 찍어서 보는 경우를 생각한 것입니다.

잠깐 실제 설명 페이지 중의 하나를 방문해보십시오. 이 페이지는 앞에서 방문한 실제 기능 페이지를 설명하고 있는 페이지입니다. (기능 페이지의 예를 보시고 난 다음에는 브라우저의 백을 이용하셔서 이 도움말로 돌아오십시오.)

설명 페이지는 기능 페이지와는 달리 왼쪽에 위치한 진한 색깔의 바아(bar)가 존재하지 않습니다. 가장 왼쪽위에는 설명하고 있는 기능 페이지의 아이콘이 있습니다. 이 아이콘을 누르면 해당하는 기능 페이지로 이동할 수 있습니다.

마지막에는 기능 페이지와 동일하게 KLE, KIBS와 바로 위 레벨로 이동할 수 있는 아이콘을 두었습니다. 단, 이 경우에는 홈페이지로 이동하는 것이 아니고, 해당하는 곳의 설명 페이지로 이동하게 되어 있습니다. 다시 말해서, 설명 페이지에서는 해당하는 기능 페이지로의 이동과 설명 페이지로의 이동만이 가능합니다.

각 설명 페이지에서는 세부 설명 페이지로의 이동이 가능합니다. 이러한 설명 페이지들에 대한 이동을 위해 제목에 해당하는 부분의 바로 밑에 이동을 위한 하이퍼링크를 둡니다.

4. 당부사항

각 페이지의 마지막에는 고칠 점을 바로 관리자에게 메일을 보낼 수 있는 링크가 마련되어 있습니다. 각 페이지를 보시다가 의문나시는 점이 있으시면 바로 연락을 주십시오.

최종 수정일: 1995년 11월 01일

KLE Administrator

등 록
Korean Information I

1. 등록을 하면

KIBS에서는 여러가지 소스나 데이타들을 다운로드 받으실 수 있습니다. 이러한 소스나 데이들의 다운로드는 관리자에 의해서 인정된 분들에게만 허용되고 있습니다.

등록을 하시면 관리자에게 판단에 의하여, 사용자의 레벨이 주어지고, 사용할 수 있는 계정고 암호가 주어지게 됩니다. 이러한 결정 사항은 전자우편으로 알려드립니다. 또한 등록 정보는 KIBS의 사용 통계를 내는데에도 사용됩니다.

2. 등록

[주의]주민등록번호, 이름, 전자우편은 반드시 입력해야 합니다.

한글이름 (영어이름):
예:홍길동 (Hong Gil Dong)
전자 우편 :
전화번호 :
예:042-821-7777
주민등록번호 :
예:700519-1400811
소속 :

최종 수정일: 1997년 4월 7일

KLE Administrator

여백

8. 문서구조 표현을 위한 표준화에 관한 연구

숙명여대
김성혁

여백

8. 문서구조 표현을 위한 표준화에 관한 연구 (A Study on the Standardization for Document Structuring)

1장. 서론

컴퓨터와 정보기술의 발달은 대규모의 텍스트 데이터를 필요로 하는 언어처리분야에서의 컴퓨터 이용을 촉진하는 계기가 되었다. 언어처리분야는 응용영역이 다양하고, 언어가 가지고 있는 사회적 및 관습적인 현상, 언어의 의미론적 및 구문론적 특성, 다언어성 등으로 인해 아직까지도 해결되어야 할 많은 문제점을 갖고 있다. 특히, 세계 각국이 고유한 자국의 언어를 갖고 있기 때문에 언어간의 상호운용성을 확보하는 문제는 세계적으로 정보를 공유하고 유통시키는데 커다란 장애 요인이 되고 있다.

이러한 이유중의 하나는 컴퓨터를 이용한 언어처리는 다양하고 방대한 텍스트 데이터베이스를 필요로 하는데 지금까지 이러한 텍스트데이터베이스 구축에 관한 논의가 미미하였다. 외국의 경우, 이러한 텍스트데이터베이스를 약 30년전부터 구축하여 왔지만 구축된 데이터베이스의 공유가 이루워지지 않아 큰 진전이 없었다. 그러나 1987년부터 텍스트데이터베이스의 공유를 목적으로 표준화을 이루기 위한 움직임이 싹트기 시작하였다.

이러한 움직임과 더불어 또 다른 분야, 즉 인쇄된 문헌을 디지털화하기 위한 연구들이 시작되었다. 디지털 기술의 발달로 인해 인쇄된 문헌을 디지털화함으로써 기존의 인쇄문헌이 가지고 있는 문제점들을 해결하고 이용자에게 보다 효율적으로 정보를 제공하려는 시도들이 디지털도서관 분야에서 연구되고 있다. 기존의 인쇄매체의 문헌을 디지털문헌으로 변환하는 과정이라고 할 수 있는 이 연구의 핵심은 인쇄문헌이 표현하고 있는 정보들을 어떻게 디지털로 표현할 것인가 하는 것이다. 텍스트인코딩은 이러한 연구에서 등장한 새로운 분야이다. 인쇄된 문헌이

표현하고 있는 정보들을 특정한 언어를 사용하여 디지털로 표현할 때, 디지털문헌의 이용자는 인쇄매체의 문헌과 동일한 정보를 얻을 수 있다. 만약 인쇄매체에서 얻을 수 있는 정보들을 디지털문헌으로 변환하였을 때 얻을 수 없다면 이는 살아있는 문헌(live document)을 죽은 문헌(dead document)으로 만든 결과가 되는 것이다.

언어처리에 필요한 텍스트데이터베이스 구축도 같은 의미를 갖고 있어야 한다. 즉, 인간이 인쇄된 문헌에서 얻을 수 있는 정보들을 컴퓨터도 동일하게 텍스트데이터베이스에서 얻을 수 있을 때 컴퓨터를 이용한 언어처리가 가능한 것이다. 만약 텍스트데이터베이스 구축 과정에서 그러한 정보들이 표현되지 못하거나 손실된다면, 컴퓨터가 처리한 결과는 부정확하게 되고 말 것이다. 따라서 텍스트데이터베이스 구축과 디지털문헌의 제작 과정에서 가장 중요한 것은 인쇄된 문헌의 텍스트를 어떻게 구조화하고 표현할 것이냐 하는 것이다.

이러한 움직임을 배경으로 텍스트인코딩에 관한 두가지의 연구결과가 발표되었다. 하나는 텍스트인코딩을 위한 언어인 SGML(Standard Generalized Markup Language)의 개발과 이 SGML 이 국제표준으로 제정되었다는 것이다. 또 하나는 유럽과 미국을 중심으로 아카이브와 코퍼스 구축을 위한 공통의 인코딩스키마에 대한 필요성을 인식하여, 다양한 문헌유형에 맞는 인코딩스키마를 개발하기 위한 국제적인 프로젝트인 TEI(Text Encoding Initiative)를 발족시키고, 인코딩 언어로 SGML 을 채택하였다는 것이다. 이는 SGML 이 텍스트인코딩 언어로 인증을 받은 결과로써, 오늘날 선진 각국은 텍스트데이터베이스, 코퍼스, 디지털문헌 등의 구축에 SGML 을 표준으로 사용하고 있다.

한편, 국내에서는 우리말 정보처리를 위한 텍스트데이터베이스와 코퍼스 구축이 1990 년대 초부터 시작되었지만 국제표준을 적용하지 않았기 때문에 정보의 공유가 이루어지지 않았다. 이러한 문제점을 인식하고 우리말 정보처리를 위한 표준의 연구 및 제정에 대한 논의가 1990 년대 중반부터 본격화되기 시작하였다.

이러한 노력의 일환으로 1 차년도에는 TEI 에서 개발한 인코딩스키마(TEI P3)를 기반으로 우리말 문헌 구조를 표현할 수 있는 최소한의 스키마인 TEI-K 를 개발하였다. 특히 지금까지 알려져 있는 용어데이터베이스와는 다른 개념의 용어학적인 용어데이터베이스 구축의 필요성을 제시하였고 TEI 용어데이터베이스를 수정 및 보완하여 우리말 용어를 수용할 수 있는 인코딩스키마를 제안하였다. 그리고

실제로 이 스키마를 적용하여 컴퓨터 및 정보학 분야 용어 100 개를 구축하였다. 나아가 용어데이터베이스 구축을 위한 편집기를 개발하였다.

본 연구는 1 차년도의 연구 결과를 보완 및 확장하고, 특히 용어학적 개념의 용어데이터베이스를 정착시키기 위한 최종 결과이다. 따라서 1 차년도에서 최소한으로 제시하였던 문헌유형별 인코딩스키마를 확장하였고, 이를 사용하기 위한 가이드라인을 작성하였다. 또한 용어학적 개념의 용어데이터베이스 스키마의 경우 1 차년도 결과중 엔티티를 그룹별로 표시하였다. 이를 바탕으로 컴퓨터 및 정보학 분야 용어 100 개를 개발한 스키마에 따라 태깅하여 구축하였다. 용어데이터베이스 구축을 위한 편집기는 1 차년도 결과를 바탕으로 SGML 의 기능들을 보완하였다. 본 보고서에는 다음과 같은 연구결과들을 포함하고 있으며, 이들은 다음 장에서 자세히 설명하였다.

-TEI-K 확장 및 텍스트 유형별 DTD 특성

-TEI-K Guideline 작성을 위한 TEI Guideline 번역

-용어데이터베이스 DTD 확장 및 보완

-전문용어사전 확장

-용어데이터베이스 편집기 기능 확장

2 장. TEI-K 설계원칙

TEI 는 인쇄된 텍스트 유형 및 특성들을 인코딩을 하기 위한 공통스키마(SGML DTD)를 개발하고, 이 스키마에 따라 텍스트를 마크업하는 지침을 작성하였다. 현재 TEI 에서 다루고 있는 텍스트 유형과 특성들은 다음과 같다.

-문자 셋(character sets)

-언어 코포라(language corpora)

-일반적 언어학(general linguistics)

-사전(dictionaries)

-용어학적 데이터(terminological data)

-스포큰 텍스트(spoken text)

-하이퍼미디아(hypermedia)

-문학 산문(literary prose)

-운문(verse)

-드라마(drama)

-역사적인 소스 자료(historical source materials)

-텍스트 비평장치(text critical apparatus)

따라서 현재 나와있는 TEI P3 는 이러한 유형 및 특성들의 마크업에 관한 연구 결과이다. 현재 TEI 에 참여하고 있는 학회 및 기관들은 다음과 같다.

-Modern Language Association

-Association for History and Computing

-American Historical Association

-Association for Documentary Editing

-American Philological Association

-American Philosophical Association

-Association Internationale de Linguistique Appliquee

-Linguistic Society of America

-American Society for Information Science

-Association Internationale Bible et Informatique

TEI 는 텍스트인코딩에 관한 지침서를 계속적으로 수정 및 보완작업을 수행하고 있다. 특별히 SGML 이 가지고 있는 텍스트 표현의 한계점을 보완하는 연구와 작업들이 계속되고 있다.

TEI에 대한 전반적인 개요, 구조, DTD 메카니즘에 관한 사항은 본 연구의 부록에 나와있는 지침서나 본 연구의 1차년도 보고서에 자세히 나와있다.

1절. TEI-K 확장 및 보완

TEI DTD는 텍스트가 표현하고 있는 다양한 구조, 포맷 및 문자 셋 등을 이용 목적에 따라 DTD에 수용할 수 있도록 설계되어 있다. 이는 TEI DTD가 모든 유형의 텍스트를 인코딩할 수 있는 수퍼셋(super set)개념의 인코딩스키마라 할 수 있다. 따라서 이용자는 연구목적에 따라 원하는 스키마를 선택하여 사용할 수 있다. 그렇지만 DTD 구조의 다양성으로 인해 이용자는 DTD를 이해하고, 이를 이용하여 텍스트를 인코딩하는데 어려움이 있다. 그 결과 TEI에서도 이러한 불편을 해소하고 실제 인코딩에서의 편리함을 위하여 TEI Lite: An Introduction to Text Encoding for Interchange을 간행하였다. 그러나 이들 DTD들은 서구중심의 문헌구조를 반영하고 있기 때문에 우리말의 인코딩을 위해서는 우리말이나 구조를 반영된 포맷이 필요하다.

본 연구의 1차년도에는 이와 같은 TEI DTD를 수정하여 고정된 기본적인 스키마를 개발하였는데, 2차년도에는 이를 확장 및 보완하여 1차년도의 스키마를 포함하면서 이용자에게 풍부한 선택권을 갖는 DTD를 개발하였다. 용어데이터베이스 구축을 위한 DTD는 1차년도에 개발한 용어사전 편집기와의 호환성으로 인해 확장을 최소화하였다. 따라서 용어 DTD의 경우는 1차년도의 결과를 보완하고 여기에 맞도록 편집기의 기능을 보완하였다.

모든 TEI 인코딩스키마는 산문 DTD를 기본으로 하여 확장된 포맷이다. 따라서 2차년도에는 산문 DTD를 중심으로 한 포맷을 제시하였다. 주요 보완내용은 TEI 헤더를 추가하였고 1차년도에 개발한 DTD중에서 전반적인 문헌구조의 균형을 위하여 불필요한 엘리먼트와 속성들을 삭제하였다. 특히 지나치게 세분된 속성들은 인코딩의 편리를 위하여 속성의 수를 줄였다.

<부록 1>에 5개 문헌유형에 대한 DTD를 수록하였다.

2 절. TEI-K DTD 의 구조

TEI 문헌은 TEI 헤더 부분과 DTD 에 따라 인코드된 텍스트 본문으로 구성된다. TEI 의 기본구조는 다음과 같은 구조를 갖는다.

TEI header

Text

전방내용 (front matter, optional)

본문 (body)

부속물 (back matter, optional)

TEI 헤더(header)는 인코딩된 텍스트, 텍스트의 자료원(source), 인코딩 배경(encoding history) 등 TEI 문헌의 다양한 서지사항이 기록되는 부분이며, TEI 문헌의 제일 처음에 위치하는 필수요소로 도서관이나 기록보존소에서 목록을 작성하고 자동처리를 하는데 유용하다. 이 TEI 헤더는 컴퓨터 파일 자체에 대한 서지적 기술사항이 수록되어 있는 필수요소인 <fileDesc>과 코딩된 전자문헌과 그 원문과의 관계에 대한 정보가 수록되어 있는 <encodingDesc>, 문헌의 분류정보나 문맥정보를 기술하는 <profileDesc>, 그리고 문헌 구축 기간 동안의 변화에 관한 기록을 담고 있는 <revisionDesc>의 네 부분으로 구성된다.

전방내용 <front matter>은 표제지에 나타나는 표제사항, 저자 사항, 출판사항 등에 대한 문헌요소로 구성되어 있는 부분이며, 본문 (body)은 문헌의 본문에 해당하는 부분으로 장, 절, 문단, 구 등의 문헌요소로 이루어진다. 본문의 그룹은 한 문헌에서 나타난 여러 텍스트들을 그룹으로 모아 주기 위하여 사용하는데, 예를 들어, 한 작가가 쓴 여러편의 수필이 한 문헌에 나타날 때 사용한다. 부속물(back matter)은 문헌의 가장 뒤에 위치하며 참고문헌이나 부록 등으로 구성된다.

이러한 구조를 갖는 TEI 문헌에서 이용되는 태그의 수는 약 400 여 개로, 헤더에서 이용되는 태그는 약 60 여 개 정도이고 핵심이 되는 기본태그는 약 100 개 정보이며, 필요한 것을 선택하여 추가할 수 있다.

TEI DTD 는 TEI 헤더와 유사한 구조를 가지고 있으며 여러 개의 DTD fragment (Tag Set)들로 구성된다.

- 문헌유형 주정의부 (main TEI DTD)는 세가지의 DTD 로 구성된다.
  - 핵심문헌유형정의부 : TEI DTD 의 표준형태 (core DTD fragment)
  - 기본문헌유형정의부 : 텍스트 유형에 의존 (base DTD fragment)
  - 부가문헌유형정의부 : 특수한 목적을 위하여 핵심문헌유형정의부 (core DTD fragment), 기본문헌유형정의부 (base DTD fragment) 와 함께 사용 (additional DTD fragment)

핵심문헌유형정의부는 TEI 문헌이 사용하고 있는 문헌요소를 선언한 파일로서 teicore2.dtd 를 참조하여 core tag set 을 정의하고 있는 TEI.core.dtd 와 TEI 헤더에서 정의된 TEI 헤더의 태그를 선언하고 있는 파일인 teihdr2.dtd 를 참조하여 TEI header set 을 정의하는 TEI.header.dtd 를 구성한다. 기본문헌유형정의부 (base tag set)는 텍스트 유형에 따라 산문, 운문, 드라마, 연설, 사전, 용어데이터베이스, 그리고 한 문헌내에 나타나는 여러 유형의 텍스트를 처리하기 위한 일반태그셋과 혼합태그셋의 8 가지로 구성된다. 부가문헌유형정의부 (additional tag set)는 문헌 분석이나 처리시 서로 다른 인코더 때문에 발생하는 특수 요구사항에 적합한 선택적 태그를 정의한다. 그외에 이용자정의문헌정의부 (user-defined tag set)는 이용자에 의한 요소의 재명명, 삭제, 수정에 필요한 태그들을 정의한다.

문헌유형 보조정의부 (auxiliary DTD)는 문헌을 자동으로 처리할 때 유용한 부수적인 기술적 정보를 인코딩 하기 위하여 사용된다. 다음과 같이 4 부분으로 구성된다.

- independent header : 특정영역에서 문헌으로 간주되는 header tag set 으로 도서관이나 문헌보관소가 서지정보를 교환할 때 사용된다.
- writing system declaration : 문자 세트 또는 번역 스킴의 정의 및 기록
- feature system declaration : 분석적 요소 세트의 정의 및 기록
- tag set declaration : TEI 에 적합한 태그 세트를 위한 기술적 문헌을 정의하고 기록

또한 TEI는 모든 문헌요소에서 공통적으로 나타나는 속성을 정의하기 위하여 global 속성을 사용하였다. 이 속성이 갖는 값은 다음과 같다.

id : 문헌내 요소들의 유일한 식별값

n : 하나의 요소내에서 유일한 값으로 문헌 내에서 유일할 필요 없음

lang : ISO 639 에 정의된 언어코드

rend : 해당 요소가 정보원에서 표현되고 있는 형태

3장. TEI-K Guideline 작성을 위한 TEI P3 번역

TEI P3는 텍스트 유형별로 개발된 DTD를 사용하여 텍스트를 인코딩하는데 필요한 규칙, 관습 및 사례들을 자세하게 설명한 인코딩 지침서이다. TEI는 여러분야의 전문가들이 모여 수년간 연구한 결과이기 때문에 내용의 깊이 및 정도면에서 매우 방대하다. 따라서 제한된 자원으로 우리 실정에 맞는 인코딩스키마를 위한 지침서를 개발한다는 것은 무리라는 판단 아래 본 연구에서는 TEI P3의 번역 작업에 치중하였다. <부록 3>에 나와있는 지침서를 참조하여라.

앞서 지적하였듯이 번역은 양적으로 방대한 작업이고, 국내에 아직 소개되지 않은 용어나 개념들이 많이 나와있어 매우 어려운 작업이었다. 가급적 우리말로 표현하였으나 우리말 용어가 없는 경우에는 영어 표현을 그대로 사용하였다. 또한 우리말로 번역하는 과정에서 번역상의 오류 및 의미상의 오류가 있음을 부인할 수 없다. 그러나 이 지침서가 앞으로 우리말 텍스트인코딩을 위한 기초자료로 활용될 수 있도록 계속적으로 수정 및 보완되어야 한다.

현 시점에서 텍스트인코딩을 위한 메타언어로 SGML만이 유일하게 국제표준으로 제정되어 있지만 국내에서의 SGML 이용은 아직 초보적인 단계이기 때문에 SGML이 우리말 텍스트의 인코딩을 위한 언어로 적합한지의 여부는 미정이다. 아직까지는 우리말을 처리할 수 있는 SGML 관련 도구들의 개발이 미미하기 때문에 국내에서의 SGML 활성화는 다소 지연될 것으로 판단된다. 따라서 SGML의 도구개발에 대한 국가적인 지원 내지는 연구가 필요하다.

외국의 경우에는 다양한 도구들의 개발로 인해 SGML 및 TEI를 이용하여 텍

스트데이터베이스를 활발하게 구축하고 있다. 정보의 공유와 유통이라는 측면에서 텍스트인코딩에 관한 공통스키마를 따른다는 것은 매우 중요하다. 특히, 국내의 경우 텍스트데이터베이스에 대한 본격적인 구축이 진행되고 있는 시점에서 이에 대한 대응책이 필요하다.

4장. 용어데이터베이스 DTD 확장 및 보완

우리말 정보처리에 대한 관심이 고조되면서 전문용어사전을 구축하여야 한다는 움직임 및 연구가 관심을 끌고 있다. 나아가 용어에 관한 이론적인 연구를 수행하는 용어학에 대한 분야도 등장하고 있다. 본 연구에서는 전문용어들을 축적하는 용어데이터베이스 DTD를 개발하는 것이 목적이다. 그러나 국내에 잘못 인식되어있는 전문용어사전의 개념, 즉 단순히 인쇄매체의 전문용어사전을 기계가독형 형태로 변환시킨 사전이라는 개념이 변하지 않으면 본 연구에서 제시하는 용어데이터베이스 DTD의 연구 의의가 없기 때문에 일반적으로 말하는 전자적인 전문용어사전과 본 연구에서 언급하는 전문용어사전 또는 용어데이터베이스와의 구분을 설명할 필요가 있다고 본다.

1절. 용어에 대한 정의

영어인 'terminology'는 어원적으로 3가지 개념이 부여되어 있는데, 첫째 학문으로서의 terminology로 어떤 개념과 그 개념들의 표현(용어, 상징 등)을 다루는 지식의 학제적 분야라는 개념과 둘째, 각 주제분야의 개념체계를 표현하는 용어들의 집합체, 셋째, 용어로 표현된 각 주제분야의 개념체계를 기록한 출판물이라는 개념 등이다. Meyer는 terminology를 '특정 영역의 개념들에 대한 기술과 명칭에 관계되는 분야'라고 정의 하였으며, ISO 1087 Terminology-Vocabulary에는 terminology를 '특정 주제분야의 개념체계를 표현하는 용어들의 집합'이라 정의하고 있다. 따라서 terminology는 단순히 전문용어들을 모으고, 용어들의 정의를 표현한 것이 아니라 용어들간의 개념체계를 포함시켜야 한다.

이러한 개념체계를 확립하기 위해서는 전문용어들에 관한 용어작업이 필요한데, 이들은 지식의 정리 활동, 지식의 공유 및 유통을 가능하게 하는 기반구조를 제공한다. 오늘날 디지털 형태로 모든 정보를 수록하고, 검색, 공유, 유통시키고자

하는 디지털도서관 환경하에서 지식의 표현과 정보의 전달에 대한 기반을 제공하는 용어작업의 중요성이 증가하고 있다.

용어작업은 다음과 같은 활동 및 작업이 반드시 포함되어야 한다.

1) 주제영역에서 개념에 부여된 용어를 수집하고 기록

2) 주제영역의 개념체계를 정리하고 표준화 및 규격화

3) 개념과 용어와의 관계를 표준화 및 규격화

4) 설명과 정의를 사용하여 개념을 기술하고, 그 개념에 부여된 정의를 표준화 및 규격화

5) 개념과 용어에 관련된 데이터를 수집하고 기록하는 작업

결국, 용어작업은 개념 및 용어의 표준화와 그 개념으로 구성되는 주제분야의 지식 전체를 다루는 활동 모두를 포함한다고 볼 수 있다. 이러한 용어작업의 결과는 통상 인쇄매체로 된 여러 종류의 전문용어사전, 주제명표목표, 어휘집, 시소러스 등으로 나타나는데, 이들을 종합적으로 통합하여 전문용어에 관련된 모든 데이터를 기계가독형태로 만든 것이 용어데이터베이스라 할 수 있다.

용어학은 주로 유럽을 중심으로 발달되어 왔다. 오스트리아의 비엔나학파, 체코슬로바키아의 프라하학파, 구소련의 소비에트학파, 그리고 1980 년대 중반에 기계번역에 대한 연구를 진행하면서 독자적인 용어학을 성립한 캐나다 등이 대표적인 용어학 연구 집단이다. 이중 세계 용어학을 이끌고 있는 곳이 비엔나학파로 용어학의 일반이론과 특수이론을 제공하여 용어학의 학문적 기반을 제공하였다. 이들 이론에 기반하여 용어데이터베이스를 정의하면 '용어데이터베이스란 용어작업을 통해 얻은 용어관련 데이터를 체계적으로 축적한 것'이라 할 수 있다. 용어관련 데이터란 ISO 10241 International Terminology Standards: Preparation and Layout 정의에 의하면 용어에 관련된 데이터, 개념에 관련된 데이터 그리고 관리 데이터 등을 지칭한다. 따라서 용어데이터베이스에는 전문용어에 관한 이 3 가지의 데이터가 반드시 포함되어야 한다.

2 절. 용어학의 표준화

용어학의 표준화는 용어작업 전반에 걸쳐 나타나고 있는데, 그 이유는 다양한 언어체계내에서 전문용어들이 형성될 때 사용하는 단어요소보다 개념의 수가 엄청나게 많기 때문이다. 따라서 개념을 표현할 때 단어의 조합이 발생하고, 이는 결과적으로 의미의 변화를 가져와 각 개념체계의 혼란을 일으켜 상호 의사소통시 잡음을 가중시키게 된다.

인간과 인간의 커뮤니케이션은 용어가 표준화되어 있지 않아도 큰 문제가 발생하지 않는다. 그러나 모든 정보가 디지털형태로 축적되어 컴퓨터에 의한 검색이 일반화된 오늘날, 하나의 개념에 대해 다양한 용어들을 사용한다는 것은 정보검색 및 언어처리 등에 심각한 문제점을 일으킨다. 따라서, 개념과 용어에 대한 공통적인 인식 아래 통일된 원칙과 방법을 사용하여 개념과 용어를 다루고 그 표준을 제정하는 것은 정보검색, 기계번역, 언어처리, 인공지능 등 컴퓨터에 의한 자연언어처리 전반에 걸친 기반을 제공한다는 점에서 그 의의가 매우 크다고 할 수 있다.

그러나 일부의 전문번역가들은 표준화 과정을 거친 용어뿐만 아니라 그 언어가 사용되는 언어적 환경 및 문화적 환경과 문맥에 대한 다양한 정보를 원하고 있다. 이러한 상황은 표준화가 비효율적이라는 견해를 갖게 할 수도 있으나 지식과 정보의 공유 및 유통이라는 측면에서 용어학에 대한 표준화는 최대한 지켜져야 한다.

특히 용어작업은 각각의 학문이나 주제분야 모두와 관계를 맺고 있으며, 시간과 비용을 많이 소모하는 작업이므로 국내 또는 국제간의 협력이 필수 불가결한 작업이다.

용어학의 표준에 관계하는 세계적인 기관은 1971 년 UNESCO 가 설립한 Infoterm(International Information Centre for Terminology)과 ISO 내의 TC37 Terminology-Principles and Coordination 이 있다. Infoterm 은 용어학 전반에 걸친 자료수집과 2 차 자료 편집, 교육, 연구활동 및 국제학술회의 개최 등이고, 국제적인 협력 증진을 위해 TermNet(Network for Terminology)을 운영하고 있다. ISO TC37 은 용어의 작성, 편찬 및 조성방법에 관한 국제표준을 제정하는데, Infoterm 의 연구결과를 반영한다. TC37 산하에는 3 개의 SC(Sub-Committee)가 있는데, SC1 Principles of terminology 는 용어의 원칙, 방법 및 개념체계에 관한 국제표준 제정, SC2 Layout of vocabulary

는 용어의 기술원칙과 방법, 로마자 사용, 언어의 알파벳순 배열원칙과 방법 등에 관한 국제표준 제정, SC3 Computational aids in terminology는 데이터요소, 어휘, SGML 응용에 관한 국제표준 제정 등을 담당하고 있다.

3절. 용어데이터베이스 연구 동향

용어데이터베이스는 앞에서 기술한 용어작업을 거쳐 생성된 각종 용어관련 데이터를 수록하고 있는 기계가독형 용어사전이다. 한편, 최근에 등장한 전자사전(electronic dictionary)에 대해 Yokoi는 전자 사전을 언어의 이용에 관한 공통 지식을 모아놓은 대규모 지식베이스로 보았으며, 모든 자연언어처리에 이용되는 다양한 컴퓨터사전들중 마스터사전의 역할을 수행하는 것이라고 하였다. 따라서 전자사전에는 기존의 용어데이터베이스에서 제공하는 정보 이외에 자연언어처리의 기초가 되는 코퍼스(corpus)와 개념사전, 개념기술 사전 등이 통합되어 있으며 다양한 검색접근점을 허용하고 있다.

Nkwenti-Azeh는 용어데이터베이스의 발전단계를 다음과 같은 세단계로 구분하였다. 첫 번째 단계는 1960년대 중반에서 1970년대 초반사이로 번역작업과 관련된 용어데이터베이스의 등장이다. 이 단계의 데이터베이스는 용어학적 이론이나 용어작업단계를 적용하지 않았다. 이러한 용어데이터베이스를 용어지향적(term-oriented)데이터베이스라고 부른다. 오늘날 국내에서 구축하고 있는 전문용어사전들이 여기에 해당한다. 두 번째 단계는 용어간의 계층구조 개념 및 사고를 데이터베이스에 적용한 단계로서 첫 번째 단계보다 진보적이나 용어간의 다양한 관계를 표현하는 데에는 역시 부족하다. 이 단계의 용어데이터베이스를 개념지향적(concept-oriented) 데이터베이스라 한다. 세 번째 단계의 용어데이터베이스는 아직 연구 개발중인 단계로서 용어를 특수한 지식의 표현으로 보고, 이를 수록한 데이터베이스를 전문가시스템으로 간주한다. 이 단계의 용어데이터베이스는 지식지향적(knowledge-oriented) 데이터베이스라고 부른다. 이 데이터베이스는 다양한 검색접근점, 복잡한 개념체계에 직접 접근, 용어뿐만 아니라 다양한 문맥정보와 용어의 변형 및 개념과 개념간의 관계 추적도 가능하게 된다.

오늘날 용어데이터베이스 구축은 세 번째 단계의 방향으로 진행되고 있다. 이러한 용어데이터베이스를 이용하는 이용자는 일반이용자, 주제전문가, 검색전문가, 사서, 전문번역가, 자연언어처리시스템 등이 있다. 그러나 자연언어처리시스템 쪽

으로 갈수록 더 자세한 용어정보와 언어정보를 요구하게 되기 때문에 데이터베이스 구축을 어렵게 만든다. 지식지향적 용어데이터베이스의 특징은 다음과 같다.

1) 개념과 용어사이에 나타나는 다양한 관계를 표현하는데 중점을 두어 용어 정보의 표현에 적합한 데이터모델 개발

2) 용어관련 데이터에 대한 다양한 접근점과 검색기능 제공

3) 다양한 계층의 이용자 요구사항에 적합한 용어정보 제공

4) 이용자 우호적인 인터페이스를 제공하여 용어데이터베이스 내에서 자유로운 탐색 가능

5) 네트워크상에서 다른 용어데이터베이스나 정보시스템과의 호환성 유지

선진 각국은 이러한 용어데이터베이스 구축의 필요성을 인식하고 국가적인 차원에서 지원하고 있다. 국내의 경우, 용어데이터베이스에 대한 이해 부족으로 인해 아직도 본격적인 구축작업이 시작되고 있지 않다.

본 연구는 이러한 필요성을 인식시키고, 우리말 정보처리에 필요한 전문용어데이터베이스 구축을 위한 SGML DTD를 개발하고, 이 DTD에 따라 샘플용어사전을 구축하였다. 따라서 본 연구에서 제안한 용어데이터베이스는 앞서 지적한 두 번째 단계에 해당된다고 볼 수 있다. 궁극적으로는 세 번째 단계인 지식지향적 용어데이터베이스를 구축 하여야 하겠지만, 용어학에 대한 국내의 연구가 아직은 미미하고, 더욱이 두 번째 단계의 데이터베이스를 구축하는 것도 막대한 노력, 시간 및 경비가 소요되기 때문에 이 분야 연구자들의 관심이 요구된다.

4 절. 용어데이터베이스 DTD 확장

용어데이터베이스는 하나의 전문용어가 어떠한 개념을 표시하고 있는지를 나타내기 위하여 다양한 용어관련 데이터들이 수집되어 축적되어야 한다. 본 연구에서는 용어관련 데이터를 ISO 10241 표준에 따라 다음과 같이 3 가지 카테고리로 분류하였으며, 각각의 카테고리에 속하는 데이터를 데이터베이스에 수록할 수 있는 DTD를 개발하였다.

1. 전문용어와 용어의 변형을 위한 데이터 카테고리

전문용어와 전문용어의 변형을 표현하기 위한 카테고리로 우리말 전문용어의 로마자 부분과 한자어 부분을 표현하기 위한 카테고리와 외래어 표기법에 의한 용어의 변형들에 대한 정보까지도 수록한다. 메인 엔트리와 동일하게 취급되는 표현들이 '메인 엔트리의 다른 형태' 부분에 속하며 '용어의 승인 정도'에 관한 데이터 카테고리는 메인 엔트리의 표준화 현황과 관계있는 정보를 수록하기 위한 것이다.

용어의 사용에 대한 언어적 정보의 중요성을 기록하기 위하여 용어의 결합관계나 상투적인 용어, 메인 엔트리와 대등한 구문, 엔트리의 언어적 표현에 대한 정보, 제한사항 등을 제공함으로써 전문용어에 대한 이해를 돕는다.

[표 1] 전문용어와 용어의 변형을 위한 카테고리

	데이터 카테고리	비고
	메인 엔트리	용어
	메인 엔트리의 한자 부분 메인 엔트리의 로마자 부분 외래어 표기법에 따른 용어변형	 예, 디소러스. 시소러스
메인 엔트리의 다른 형태	국제적인 표현 변이형 대체기능한 철자 법률 용어 기호 및 상징 축약형 완전형	국제과학용어 메인 엔트리를 표시 short, form, 두문자어 등 메인 엔트리의 완전형
	표준화된 용어 우선권이 있는 용어 승인된 용어 대체된 용어 사용하지 않는 용어	메인 엔트리의 표준화와 관계있는 용어들을 기록한 다.

	권고된 용어 제안된 용어 비 표준화된 용어 새로운 용어	
어법상 표현	낱말의 결합관계, 연어, 성구, 상투용어 대등한 구문	메인 엔트리가 나타나는 어법상 표현
엔트리의 복잡성	용어의 구성요소 엔트리로부터 가능한 단어의 그룹 및 구	형태소
문법정보	품사, 어형변화, 굴절 파생어	엔트리의 품사 용어의 기본형태로부터 파생된용어
	발음	국제음성기호
	분절	
	어원	
동일성	동의어 유사동의어 동형이의어 유사 동형이의어 다의성 동음성	 발음은 동일하나 의미가 다른 용어
	엔트리의 언어적 표현과 기능에 관한 정보	은어, 방언, 구어체표현, 공 식적 표현 등
제한사항	지역적 제한사항 시간적 제한사항 기타 제한사항	

2. 전문용어의 개념에 관계된 데이터 카테고리

전문용어가 표현하고 있는 개념을 기술하기 위하여 개념에 관계된 데이터 카테고리를 사용한다. 정의, 설명, 문맥, 주기, 예제는 전문용어에 대한 의미적 이해를 도와준다. [표 2]에서 '개념과의 관계'와 '개념체계에서의 위치'에서 제공하는 데이터 카테고리는 주제분야의 개념체계에서 해당 전문용어가 차지하고 있는 위치를 보여주는 정보들이다. 특히, 이들 정보들을 이용하여 시소러스를 구축할 수 있다.

[표 2] 개념에 관계된 데이터 카테고리

	데이터 카테고리	비고
주제	주제분야 하부 주제분야 사용한 주제분류표	 분류기호 포함
정의	정의 내포된 정의 부연 정의 부분 표시 정의	집합적 카테고리
	설명	
문맥	문맥 정의적 문맥 해설적 문맥 연상적 문맥	집합적 카테고리
	특성	엔트리 용어가 내포하고 있는 특성
주기	주기 예	

개념간의 관계	속 부분 연속 계절, 시대 공간적 관계 서로 간섭하는 관계 반의어 보충어 대조, 대비어	엔트리용어에 대하여 각 해당사항에 적절한 용어를 기술한다.
개념체계에서의 위치	최상위어 발생학적 최상위어 부분 전체의 최상위어 상위어 발생학적 상위어 부분 전체의 상위어 하위어 발생학적 하위어 부분 전체의 하위어 관련어	

3. 전문용어의 관리에 필요한 데이터 카테고리

관리정보를 나타내는 데이터 카테고리는 용어데이터베이스의 운영 및 유지활동과 관계있기 때문에 용어데이터베이스의 목적과 이용자층, 용어데이터베이스에서 제공하는 다양한 검색기능에 따라 필요한 데이터 카테고리의 종류가 달라 질 수 있다. [표 3]에서 제안한 데이터 카테고리는 관리정보를 표현하기 위한 최소한의 정보로써, 구축되는 용어데이터베이스의 환경에 따라 데이터 카테고리를 추가할 수 있다. 메인 엔트리의 용어에 대한 색인어, 탐색어, 용어작업에 관련된 사항 및 기능을 표시하기 위한 카테고리를 부여하였다.

[표 3] 관리정보에 관한 데이터 카테고리

	데이터 카테고리	비고
	색인어	메인 엔트리의 색인형
	탐색어	메인 엔트리를 조회하는 표현이나 단어
용어작업에 관련된 사항	트랜잭션 각 트랜잭션에 대한 날짜 책임사항 표준화 기관 원생산자 처리상태	신규, 갱신, 철회, 승인 등을 나타내는 기호로 표시 개인, 위원회 등 기관이나 저자
	정보원에서 엔트리의 위치 정보원의 형태	어휘집, 표준안, 시소러스 등
	레코드에 관한 책임사항	책임기관이나 책임자

상기 데이터를 수록할 수 있는 DTD 개발에 따른 설계원칙에 대한 설명은 본 연구의 1 차년도 보고서에 자세히 나왔다.

2 차년도는 1 차년도의 개발을 근거로 하여 일부의 수정을 가하였다. 그러나 DTD 에 사용하는 엘리먼트들과 속성들은 변하지 않고 1 차년도에 사용되었던 엔티티들을 그룹화하여 전반적인 DTD 의 가독성을 향상시켰다.

2 차년도에 새로 그룹회된 엘리먼트들과 이들 엘리먼트들의 속성 및 역할은 다음과 같다.

1) a-class: 동일한 속성을 공유하고 있는 엘리먼트들을 파라미터 엔티티를 사용하여 그룹화하였다.

-a.analysis: 상세분석이나 또는 해석을 텍스트의 관련 부분과 연결시키기 위한 글로벌 속성을 정의한다.

-a.declaring: 헤더내에서 특정한 declarable element 와 관련되는 엘리먼트들을

그룹화한다.

-a.linking: 하이퍼텍스트나 그 외의 텍스트에 대한 연결을 위해 사용되는 부가적인 속성들을 정의하며, 사용되는 속성들은 다음과 같다:

. corresp(corresponding)-현재의 엘리먼트에 대응하는 엘리먼트를 포인트한다.

. synch(synchronous)-현재의 엘리먼트와 동시에 발생하는 엘리먼트를 포인트한다.

. sameAs-현재의 엘리먼트와 동일한 엘리먼트를 포인트한다.

. copyOf-현재의 엘리먼트가 복사본일 경우, 오리지널 엘리먼트를 포인트한다.

. next-현재 엘리먼트의 다음 엘리먼트를 포인트한다.

. prev-현재 엘리먼트의 이전 엘리먼트를 포인트한다.

. exclude-현재의 엘리먼트를 제외한 나머지 엘리먼트를 포인트한다.

-a.terminology: 용어데이터를 위한 기본 태그셋이 이용되는 문헌의 모든 엘리먼트를 위한 속성들을 정의하며, 여기에 속하는 속성들은 다음과 같다.

. group-적절한 엘리먼트의 n 속성값에 매칭되는 스트링을 명시하여 관련되는 엘리먼트에 대한 그룹(용어나 관련 엘리먼트)을 나타낸다.

. grpPtr-유일식별기호(unique identifier)를 명시하여 관련되는 엘리먼트에 대한 그룹을 나타낸다.

. depend-적절한 엘리먼트의 n 속성값에 매칭되는 스트링을 명시하여 관련되는 엘리먼트의 부모 엘리먼트(parent element)를 나타낸다.

. depPtr-유일식별기호를 명시하여 관련되는 엘리먼트에 대한 부모 엘리먼트를 나타낸다.

2) m-class: 문헌에서 구조적으로 동일한 위치에 나타날 수 있는 구조적으로 유사한 구성원들을 그룹화 하였다.

-m.edit: 편집상의 수정이나 교정을 위한 phrase-level 엘리먼트 클래스로 다음과 같은 엘리먼트들이 포함된다.

. sic: 명백한 에러로 나타난 부분을 표시한다.

. reg(regularization): 표준 또는 조정된 텍스트를 표시한다.

. orig(original form): 조정되지 않은 텍스트의 원 형태(original form)를 나타낸다.

. del(deletion): 삭제되는 단어나 구, 절을 표시한다.

. corr(correction): 복사본에서 에러로 나타나는 passage 의 교정형태를 표시한다.

. add(addition): 저자나 교정자가 텍스트에 삽입하는 단어나 절을 표시한다.

-m.hqinter: 하이라이팅(highlighting)에 관련되는 중간레벨 엘리먼트 클래스로 여기에는 다음과 같은 엘리먼트들이 포함된다.

. quote-텍스트 외부에 있는 화자나 저자에 의해 속성화된 절이나 문장을 나타낸다.

. q-인용된 부분이 언어인지 사고인지를 나타낸다.

. cit-문헌의 source 에 대한 서지참조와 함께, 그 외의 다른 문헌에서 인용한 것을 나타낸다.

-sgmlKeywords: 컨텐드가 태그(엘리먼트 유형의 일반식별기호, 속성명 등)나 SGML 식별기호인 엘리먼트 클래스를 나타내며 여기에 속하는 엘리먼트는 다음과 같다.

. val(value)-single attribute value 가 포함된다.

. tag-opening 또는 closing markup delimiter characters 를 제외한 attribute specification 을 포함하는 SGML 시작태그나 종료 태그 등의 텍스트를 나타낸다.

. gi(generic identifier)-SGML 엘리먼트의 이름을 나타낸다.

. att-현재 텍스트에서 나타내는 속성의 이름을 나타낸다.

상기의 원칙을 적용하여 개발한 용어데이터베이스 DTD 는 <부록 2>에 나와있다.

한편, 개발된 DTD 에 따라 용어작업을 하기 위한 구축과정은 다음과 같다.

. 범위의 정의
. 정보원 검토 및 선정
. 전문용어, 용어에 대한 정의, 설명, 예제 등 전문용어 데이터 수집
. 개념 구조 개발
. 용어간 대등관계 설정
. 용어정보 및 개념구조 기록
. 편집기를 이용하여 DTD 에 따라 마크업
. 파싱
. 용어데이터 수정
. 전문용어사전 데이터 포팅

앞서 지적하였듯이 전문용어데이터베이스 또는 전문용어 사전의 구축은 언어정보처리, 기계번역, 및 정보검색 등의 분야에서 매우 중요하다. 특히, 국제간의 정보 유통 및 전달이 원활하기 위해서 전문용어 사전의 구축은 서둘러야 할 것이다. 그러나 국내의 경우, 아직까지 전문용어 사전의 구축작업이 본격화 되지 않고 단지 인쇄매체로 된 전문용어사전을 단순히 기계가독형 형태로 변환하는 수준에 머무르고 있다. 용어지향적이 아닌, 적어도 개념지향적이나 지식지향적 전문용어 사전의 구축에 대한 시도내지는 필요성을 인식할 수 있는 계기가 되어야 한다.

5 장. 전문용어 사전 확장

본 연구에서 제안하는 전문용어사전은 용어지향적이 아니라 적어도 개념지향적 또는 지식지향적이어야 한다는 것이다. 따라서 본 연구에서 개발한 인코딩 포맷에 따라 전산학 및 정보학분야의 전문용어 100 개를 선정하여, 전문용어사전을 구축하였다. 그러나 전문용어와 관련된 다양한 정보들을 수집하는데는 시간과 노력이 많이 소요되고, 정보원이 부족하다는 문제가 있다.

최근들어 디지털도서관 구축 및 정보검색에서 사용하기 위한 시소러스의 개발이 활발하게 논의되고 있고, 국가적인 차원에서 이를 개발하여야 한다는 주장이 제기되고 있다. 그러나 시소러스보다는 본 연구에서 제시하는 개념지향적 또는 지식지향적 용어데이터베이스를 개발하는 것이 이용이나 개발비용 측면에서 효율적이라고 생각된다.

선진 각국은 오래전부터 전문용어에 대한 이론적인 연구와 더불어 용어지향적이 아닌 개념지향적 또는 지식지향적 전문용어사전을 SGML 이나 TEI 의 인코딩 스키마에 따라 구축하여 이용하고 있으며, 전문용어를 위한 국제적인 활동에 적극 참여하고 있다.

2 차년도에 전문용어 사전에 추가된 전산학 및 정보학 용어 100 개, 그리고 '가상현실'과 '객체지향언어'라는 전문용어에 대한 태깅 예는 다음과 같다.

1. 구문분석

<termEntry>

<descrip type='domain'>Computer Science</descrip>

<tig lang=kor>

<term id=6 type='mainEntryTerm'>구문분석</term>

<descrip type='explanation'>자연어 문장을 문장안의 단어들 사이의 문법적 관계를 나타내는 구조에 대응시키는 것을 말한다. 문장에 대한 이러한 구조는 보통 파스 트리의 형태로 표현된다. 결국 구문 분석이란 문장에 대한 파스 트리를 구성하는 것(파싱)을 말한다. 이는 문장의 의미를 이해 하는데 있어서 지나치게 세부적인 사항을 감추기 위한 수단으로 사용된다. 구문분석의 과정은 주어진 문장이 문법적으로 타당한 것인가를 조사하고, 타당한 문장에 대하여 파스트리를 구성하는 두 단계로 구분 된다.</descrip>

<ref type='bibliographic' target=데이터베이스용어사전>p.31</ref>

<descrip type='definition'>인공어 혹은 자연어의 단위들 간의 관계를 좀 더 기본적인 작은 단위들간의 관계를 설정. 예를 들면, 블록, 문장, 식 그리고 연산자와 피연산자들로 나눈다.</descrip>

<ref type='bibliographic' target=정보통신용어사전>p.127</ref>

<ofig><otherForm type='relatedTerm'>구문해석</otherForm>

<descrip type='difinition'>자연언어 문의 문법적 구조를 규명하는 것</descrip>

<descrip type='explanation'> 프로그램구조를 그 언어의 구문규칙을 기초로 해석하는

것을 말한다. 형식문법으로 말하면, 형식문법 <term lang=en>G=(N, T, P, S)</term>에 대해 입력문 <term lang=en>w ⊆ T*</term> 가 <term lang=en>G</term>의 시작기호 <term lang=en>S</term>에서 도출가능여부를 판정하고 그 도출예를 구하는 것을 말한다.</descrip>

<ref type='bibliographic' target=문헌정보학용어사전>p.40</ref>

<descrip type='explanation'>어느 언어(자연 언어나 프로그램 언어)의 문이나 문장의 구문상의 구조를 해석하는 것. 특히 프로그램 언어에 있어서 구 <term type=hanja>(句)</term> 구조 문법 등의 구문계에 따라서 원시 프로그램의 구문상 구조를 해석하는 것. 프로그램 언어에서는 구문을 문맥 자유 문법으로서 보는 경우가 많고, 이 경우 구문상의 구조는 문맥 자유 문법에서의 도출 트리에 따르는 구조로 파악되는 일이 많다.</desctip>

<ref type='bibliographic' target=컴퓨터용어대사전>p.1058</ref>

</ofig>

</tig>

<tig type='hanja'>構文解析</tig>

<tig type='hanja'> 構文分析</tig>

<tig lang=en>

<term type='equivalentTerm'>syntactic analysis</term>

<descrip type='explanation'>The problem of associating a given string of symbols through a grammar to a programming language, so that the question of whether the string belongs to the language may be answered.</descrip>

<ref type='bibliographic' target=McGraw-Hill DICTIONARY OF SCIENTIFIC AND TECHNICAL TERMS>p.1675</ref>

</tig>

</termEntry>

2. 구조적 프로그래밍

<termEntry>

<descrip type='domain'>Computer Science</descrip>

<tig lang=kor>
<term id=7 type='mainEntryTerm'>구조적 프로그래밍</term>
<descrip type='explanation'>하향식 설계 및 구형과 구조적 프로그램 제어 구조의 엄격한 사용을 적절히 혼합하여 프로그래밍하는 기법이다. 정확하게 프로그래밍할 수 있게 고안된 여러 가지 기법의 총칭이다. 이 기법은 다음과 같은 제한 조건을 갖는다. 첫째, 한 프로그램의 크기는 하나의 코드 용지에 제한된다. 둘째, 프로그램 구조의 종류도 제한된다. 둘째, 프로그램 구조의 종류도 제한된다. 셋째, 오직 하나의 입구와 출구를 갖도록 만들어져야 한다. 넷째, <term lang=en>GOTO</term>문의 사용을 금지한다. 현재 이 기법은 소프트웨어 개발과 관리의 모든 단계에 널리 적용된다.</descrip>
<ref type='bibliographic' target=데이터베이스용어사전>p.31</ref>
<descrip type='explanation'>프로그램자나 다른 이용자들이 보다 쉽게 이해하고 읽을 수 있도록 명확한 구조의 프로그램을 서계 작성하는 기법. 신뢰성이 높은 시스템을 신속하고 저렴하게 제작하는 기법의 하나이다. <name type=person lang=en>E.W.Dijkstra</name>에 의해 개발된 것으로 프로그램 작성을 신속 정확하게 할 수 있고, 프로그램 수정시 간결하게 할 수 있다는 장점이 있다.</descrip>
<ref type='bibliographic' target=문헌정보학 용어사전>p.41</ref>
<descrip type='explanation'> <date>1960 년대</date> 중반에 제창되기 시작한 프로그래밍 방법론으로서 쉽게 이해할 수 있고 검증할 수 있는 프로그램 부호를 생성하는 것을 주목적으로 한다. 즉 구조적 프로그램 부호를 생성하는 것을 주목적으로 한다. 구조적 프로그래밍의 특징은 큰 프로그램을 단계적으로 분할하여 작성하는 하향식 프로그래밍, 한 모듈 안에서는 순차, 선택, 반복의 세 가지 제어 구조만을 사용하고 되도록 <term lang=en>GOTO</term>문을 사용하지 않는 것, 그리고 프로그램의 가독성<term lang=en>(readability)</term>을 높이기 위해 들여쓰기, 주석 등과 문서화를 철저히 하는 것 등이다.</descrip>
<ref type='bibliographic' target=정보통신용어사전> p.131</ref>
<descrip type='definition'>신뢰할 수 있는 프로그램 작성법의 하나. 구체적 수단은 단계적 세련과 제어 구조를 규제하는 것으로 한다. </descrip>
<ref type='bibliographic' target=컴퓨터용어대사전>p.1044</ref>

</tig>

<tig type='hanja'>構造化프로그래밍</tig>

<tig lang=en>

<term type='equivalentTerm'>structured programming</term>

<descrip type='definition'>The use of program design and documentation techniques that impose a uniform structure on all computer programs.</descrip>

<ref type='bibliographic' target=McGraw-Hill DICTIONARY OF SCIENTIFIC
AND TECHNICAL TERMS>p.1940</ref>

<admin type='broaderTerm'>programming</admin>

<admin type='narrowTerm'>Jackson structured programming</admin>

<ref type='bibliographic' target=INSPEC>p.317</ref>

</tig>

</termEntry>

<용어 100 개>

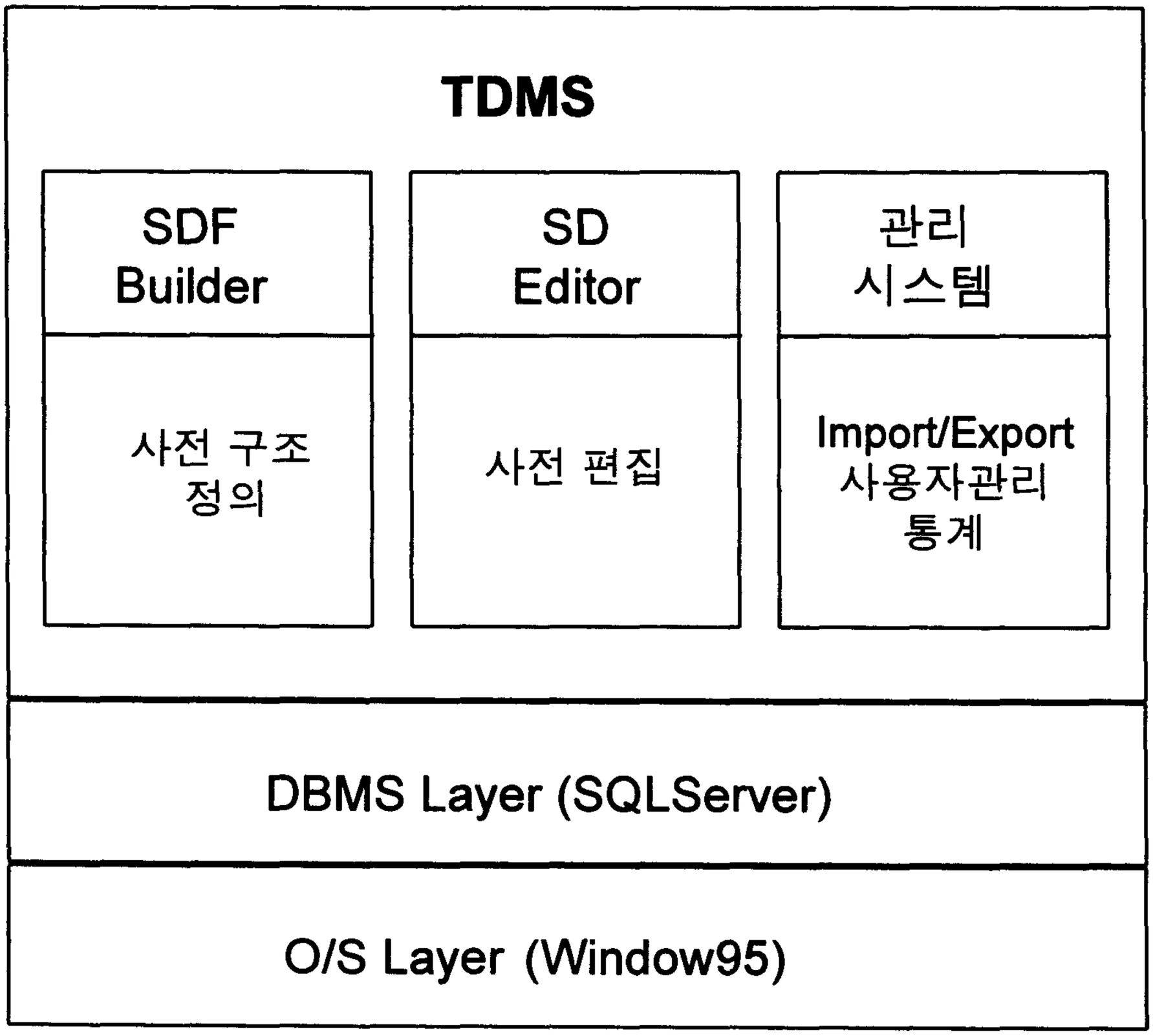
1. 가상현실 artificial reality	2. 객체지향언어 object-oriented language
3. 공공도서관 public library	4. 관계데이터베이스 relational system
5. 광역통신망 wide area networks	6. 구문분석 syntatic analysis
7. 구조적 프로그래밍 structured programming	8. 근거리 통신망 local area network
9. 기억장소 memory	10. 노드 node
11. 다중처리 multiprocessing	12. 다중프로그래밍 multiprogramming
13. 다중매체 multimedia	14. 데이터모형 datamodels
15. 도큐멘테이션 documentation	16. 동일화 unification
17. 동적재배치 dynamic relocation	18. 듀이십진분류법 Dewey Decimal Classification
19. 디스크운영체제 Disk Operating Systems	20. 디지털통신 Digital Communication
21. 라이브러리 프로그램 Library Program	22. 레코드 record
23. 루프 Loop	24. 마우스 Mouse
25. 마이크로 프로그래밍 Microprogramming	26. 마이크로 프로세서 Microprocessor
27. 멀티미디어 Multimedia	28. 멀티태스킹 Multitasking
29. 명령언어 Command Language	30. 모듈 Module
31. 모의실험 Simulation	32. 모형 Model

33. 문장 sentence
34. 미니컴퓨터 Minicomputer
35. 바인딩 시간 Binding Time
36. 버스 Bus
37. 번역기 Translator
38. 변수 Variable
37. 보완 Security
40. 부트스트랩 Bootstrap
41. 부호화 Encoding
42. 분류체계 Classification Scheme
43. 분산 Distribution
44. 분산처리 Distributed Processing
45. 비트 Bit
46. 사무자동화 Office Automation
47. 사전 Dictionary
48. 삽입 Insert
49. 서지레코드 Bibliographic Record
50. 설계 Design
51. 전송속도 transmission speed
52. 전자데이터처리시스템 EDPS
53. 절단 truncation,segmentation
54. 접근경로 access path
55. 정규문법 regular grammar
56. 정규분포 normal distribution
57. 정규화 normalization
58. 정보네트워크 information network
59. 정보전략 information carrier
60. 정보축적 information storage
61.정보흐름분석 information flow analysis
62.정보흐름제어 information flow control
63.정수론 number theory
64.제어문자 control character
65.제어프로그램 control program
66.제한탐색 restricted search
67.조판 live matter
68.종합색인 general index
69.주변장치 peripherals
70.주파수변조 FM
71.중앙집중처리 centralized processing
72.중앙처리장치 central processing unit
73.지식베이스 knowledge base
74.지식시스템 knowledge system
75.지적소유권 intellectural property
76.지프의 법칙 Zip's Law
77.직무명세서 job specification
78.직접액세스 direct access
79.진보적이론 liberal theory
80.질의응답시스템 question answering system
81.질의처리 query processing
82.집적회로 integrated circuit
83.차트 chart
84.참조 reference
85.채널 channel
86.체크포인트 check point
87.초과 overflow
88.추론 inference
89.추론규칙 inference rule
90.추론엔진 inference engine
91.추리통계 inductive statistics
92.축소율 reduction ratio
93.출력장치 Output Equipment
94.출판 publish
95.칩 chip
96.카드목록 Card Catalog
97.캐드 CAD
98.캐시메모리 Cache Memory
99.캠 CAM(Computer Aided Manufacturing)
100.컴 COM(Computer Output Microform)

6 장. 용어데이터베이스 편집기 기능 확장

1 절. 시스템의 구성

TDMS 는 SDFBuilder, SDEditor 와 관리시스템으로 구성되어 있다.



[그림 1] 시스템 구성도

1. SDFBuilder

SDFBuilder 는 사전의 구조(DTD)를 편집할 수 있으며 다음과 같은 주요 모듈로 구성되어 있다.

- 요소(Element)편집
- 구조 편집
- SDF Import/Export

2. SDEditor

SDEditor는 정의된 SDF를 사용하여 사전을 편집할 수 있으며 다음과 같은 주요 모듈로 구성되어 있다.

- 사전편집
- 사전구조 Browser
- 내용 검색
- 사전편집 속성 편집
- SD Import/Export

3. 관리시스템

관리시스템은 SDF와 SD를 관리하기 위한 기능을 가지고 있으며 다음과 같은 주요 모듈로 구성되어 있다.

- Database 관리
- 사용자 관리
- 각종 통계 및 출력

2절. TDMS

1. TDMS의 특징

- TDMS는 SGML의 Subset인 SDML 규약(KAIST)안과 호환된다.
- Server/Client 방식이므로 다양한 하드웨어와 소프트웨어 환경을 채택할 수 있다.

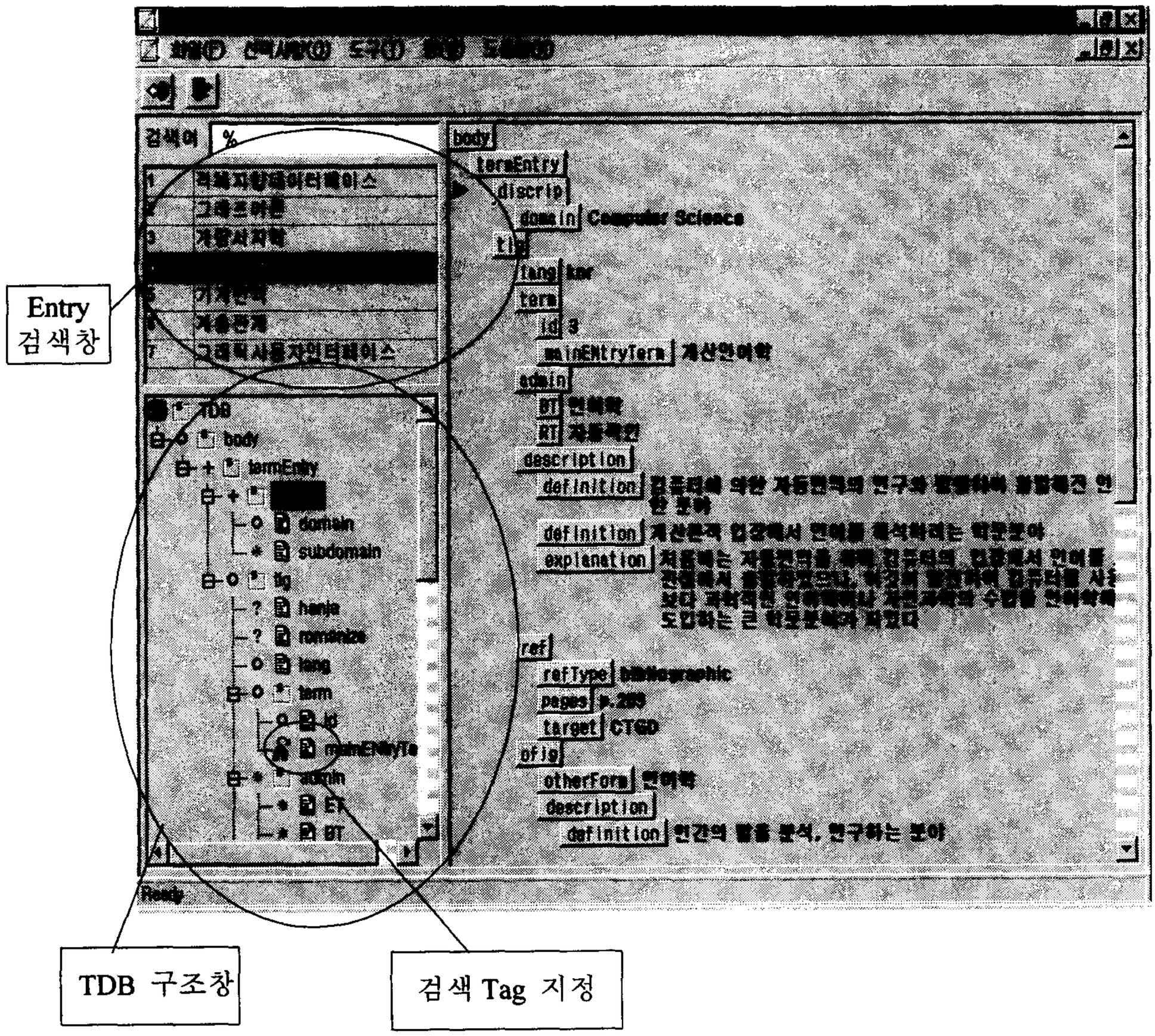
- 일반적인 사전 관리 기능과 사전의 통합관리, 유연한 구조변경, 다양한 응용 형태에 맞는 사전 생성 기능등을 제공한다.
- 데이터베이스는 상용 RDBMS 를 사용하여 자료보관의 신뢰성을 높이고 자료 접근의 보안을 유지하며 향후 확장이나 변경에 용이하다.
- Window95 탐색기와 유사한 UI 와 Drag&Drop 방식을 채택하여 사용이 간편하다.
- 팝업메뉴 및 Drag&Drop 시 적절한 항목만 선택을 하게 하여 사용자의 오류와 선택 회수를 최소화하여 빠른 작업을 할 수 있다. (예: 데이터 수정 및 삭제가 없으면 저장 버튼이 Disable 됨)
- 사용자관리를 통하여 사용자 권한 등급을 조정할 수 있다.
- 사전편집시 화면의 색상이나 테두리를 조정할 수 있다.
- 사용자환경이 변할 때 마다 이를 Registry 에 저장하여 불필요한 환경 조작이 없다.

2. 제공형태

- Install CD-ROM(Window95 용)
- 설치매뉴얼
- 사용자 매뉴얼

3. 용어사전 편집기 화면설명

용어사전 편집기의 화면은 크게 메뉴/툴바, Entry 검색창, 용어사전 구조창, 편집창의 4 가지로 구성되어 있다.



(1) 메뉴와 툴바

파일
- 열기(Ctrl+O)
- 저장(Ctrl+S)
- 프린트설정
- 닫기
- 종료

선택사항
- 하위요소추가
  - Required Only
  - everything
  - nothing
- Tag

내용
도구
Import
Export

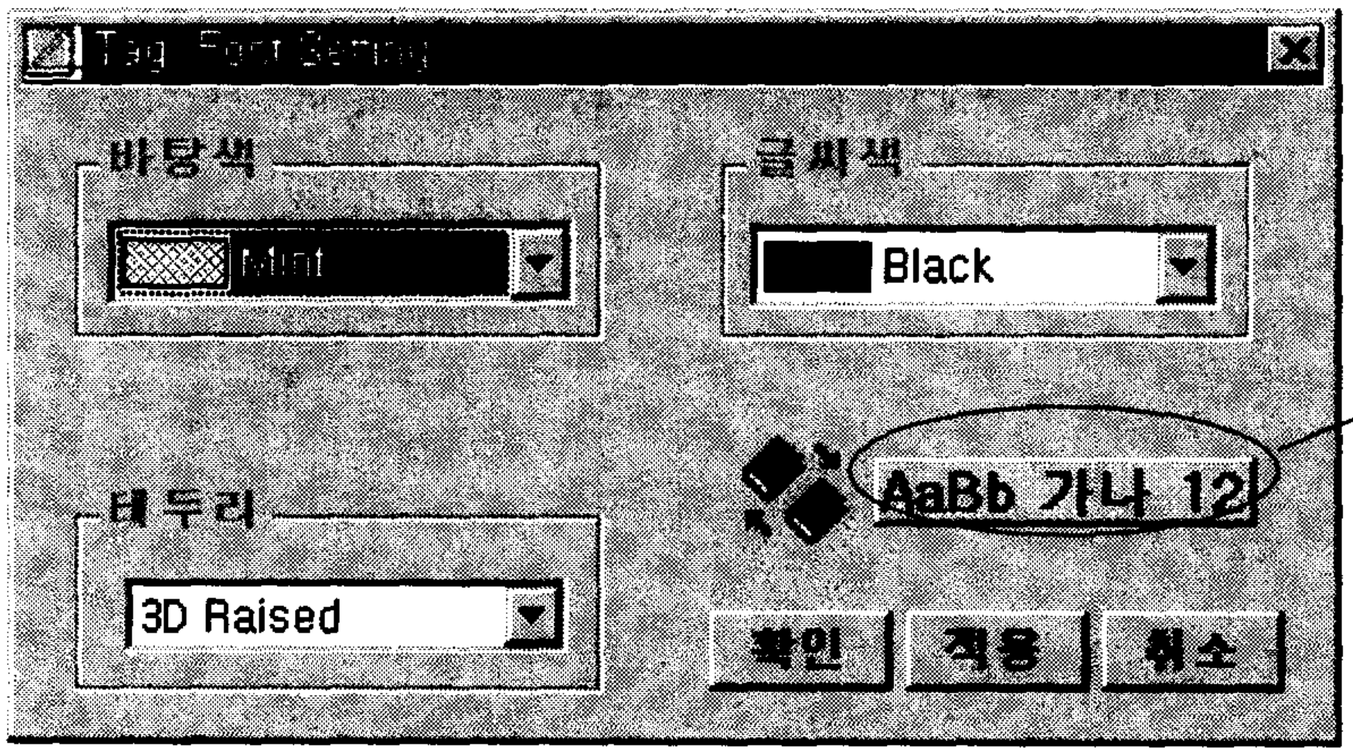
1) 하위요소 추가

Group Element 를 삽입이나 추가를 할 때 하위요소들의 생성방법으로

- Required Only 는 TDB DTD 에서 Required 와 Required & Repeatable 로 선언된 것만 생성하는 것이며
- Everything 는 모든 하위요소를 다 생성시키며
- Nothing 는 하위요소는 생성하지 않고 자신만을 생성한다.

2) Tag / 내용

- 바탕색
- 문자색
- 테두리의 설정 변경



<Export 의 예제>

```
<!-- Export from TDMS          1997/06/17                    -->
<!-- 계산언어학 ~ 계산언어학     (1 Rec.)     -->
<!DOCTYPE TDB SYSTEM "TDB.SDF" []>
<TDB>
<body>
<termEntry>
<discrip>
<domain> Computer Science
</discrip>
<tig>
<lang> kor
<term>
<id> 3
<mainENtryTerm> 계산언어학
</term>
<admin>
<BT> 언어학
<RT> 자동색인
</admin>
<description>
<definition> 컴퓨터에 의한 자동번역의 연구와 병행하여 활발해진 언
    어학의 한 분야
<definition> 계산론적 입장에서 언어를 해석하려는 학문분야
<explanation> 처음에는 자동변역을 위해 컴퓨터의  입장에서 언어를
    보는 관점에서 출발하였으나, 이것이 발전하여 컴퓨터를 사용한
    보다 과학적인 언어학이나 자연과학의 수법을 언어학에 도입하는
    큰 학문분야가 되었다
</description>
<ref>
<refType> bibliographic
<pages> p.209
<target> CTGD
</ref>
<ofig>
<otherForm> 언어학
<description>
<definition> 인간의 말을 분석, 연구하는 분야
<explanation> 언어학과 컴퓨터와는 문법, 구문, 의미론, 형식언어이론,
    자연언어처리 등에서 밀접한 관계가 있다
</description>
<ref>
<refType> bibliographic
<pages> p.625
```

<target> CTGD
</ref>
</ofig>
</tig>
<tig>
<hanja> 計算言語學
<lang> en
<admin>
<ET> linguistics
<BT> humanities
<NT> computational linguistics
<NT> etymology
<NT> grammars
<NT> phonetics
<NT> semiotics
<RT> information science
<RT> linguistic analysis
</admin>
<description>
<definition> The study of human speech in its various aspects, especially units of language, phonetics, syntax, semantics and grammar
</description>
<ref>
<refType> bibliographic
<pages> p.1148
<target> MDIT
</ref>
</tig>
</termEntry>
</body>
</TDB>

(2) Entry 검색창

entry 검색창은 입력되어 있는 내용을 찾을 때 이용한다. 초기에는 구조창의 3 번째 Element 가 대상으로 지정되어 있으나 구조창에서 해당 Element 를 DoubleClick 하여 검색 대상을 변경할 수 있다. 검색을 취소할때는 Cancel 버튼을 눌러 종료한다.

(3) TDB 구조창

해당 DTD 의 구조를 보여주는 창이다.

 : Required

 : Required and repeatable

 : Optional

 : Optional and repeatable

 : Sequence group

 : And group

 : Or group

 : #PCDATA 를 의미한다.

#PCDATA Element 에서 DoubleClick 하면 검색대상 Element 로 지정된다.()
편집창의 Element 선택에 따라 구조창의 선택 Element 도 변화한다.
Element 를 선택한 후 편집 창으로 잡아끌기(Drag&drop)를 하면 해당 요소를 편집 할 수 있다. (단 첫번째와 2 번째 Element 는 제외)

(4) 편집창

데이터를 편집하는 창이다.

- 메뉴의 선택사항->Tag(내용)로 색상 및 모양을 변경할 수 있다.
- 오른쪽버튼 PopUp 메뉴

  신규 : 신규 단어를 입력할 수 있다. 이때 기본적으로 만들어지는 Element 는 선택사항->하위요소추가 방법에 따른다.

  저장 : 편집한 내용을 DB 에 저장한다.

  삭제 : Element 를 삭제(하위 Element 포함)
- 구조창에서 Drag&drop PopUp 메뉴

  이전삽입

  이후추가

  하위

7장. 결 론

인터넷의 확산, 디지털도서관 구현 및 학문의 세분화 현상은 국제간 정보의 공유와 유통을 장벽없이 해결하기 위한 방법론을 필요로 하고 있다. 본 연구는 그러한 필요성중 하나인 텍스트인코딩의 표준화에 관한 연구 결과이다. 외국의 경우 오래전부터 이의 필요성을 인식하여 다양한 연구결과를 산출하여 이용하고 있지만, 국내의 경우 필요성 정도를 인식하고 있는 단계이다.

그러나 본 연구를 통해 텍스트인코딩의 de facto standard 인 TEI 인코딩 스키마를 분석하고 이를 우리말 문헌에 적용할 수 있도록 TEI-K 와 가이드라인을 제시하였다. 또한 전문용어 사전에 대한 개념을 용어지향적이 아닌 개념지향적 또는 지식지향적으로 변화하기 위한 샘플 용어사전을 구축하여 국내 연구자들의 관심을 촉구하였다. 전문용어사전 구축을 위한 편집기를 개발하여 다양한 응용환경에서 이 편집기를 사용할 수 있도록 하였다.

전문용어사전의 구축은 막대한 시간, 노력, 및 경비가 요구되는 작업이지만 하루빨리 시작되어야 하는 연구과제이다. 이를 통해 국제간, 학문간 정보의 공유와 유통을 촉진시킬 수 있다. 아직까지도 용어지향적인 용어사전 구축에 초점에 맞추어져 있다는 것은 국내에서의 전문용어 연구가 초보적인 단계라는 것을 전적으로 나타내는 것이다. 그 결과 전문용어사전을 필요로 하는 기계번역, 정보검색 및 언어처리 분야의 발전이 늦어졌다고 볼 수 있다.

텍스트인코딩의 경우도 기본적으로 SGML 을 기반으로 한 스키마를 채택하여야 한다. 특히 코퍼스 구축은 정보의 공유를 통해 그 효과를 최대화할 수 있다. 지금까지의 특정 대학 또는 특정 연구소 중심에서 국가적 나아가 국제적으로 공유가 가능한 표준이나 포맷을 따라야 한다.

본 연구에서 제안한 인코딩 스키마, 전문용어사전 구축방안, 인코딩 가이드라인 등은 앞으로 계속적으로 수정 및 보완되어야 한다. 동시에 이 분야의 연구자 및 잠재적인 연구자들이 공동연구를 할 수 있는 장을 마련하는 것도 중요하고 시급한 일이다. 컴퓨터를 이용한 언어처리의 기반구조에 해당하는 연구들의 활성화가 가능할 때, 관련 기술 및 이의 응용이 활성화될 수 있다.

참고문헌

김성혁. (1996) SGML 의 기본과 이해. 서울 : 성안당.

김성혁. (1996) "문서구조 표현을 위한 표준화에 관한 연구" 1 차년도 최종보고서. 숙명여자대학교.

김현주. (1995) "우리말 용어데이터베이스 구축을 위한 포맷 설계에 관한 연구." 석사학위논문. 숙명여자대학교.

Alschuler, L. (1995) ABCD SGML: A User's Guide to Structured Information. Boston : Thomson.

Bryan, M. (1995) SGML An Author's Guide. Wokingham : Addison-Wesley.

DeRose, S. J. & Durand, D. G. (1996) Making Hypermedia Work: A User's Guide to HyTime. Boston : Kluwer Academic Publishers.

Dewire, D. T. (1994) Text Management. New York : McGraw-Hill, Inc.

Felber, H. (1994) Terminology Manual. Paris : Inforterm.

Goldfarb, C. F. (1994) SGML Handbook. New York : Oxford Univ. Press.

ISO 8879. (1986) Informtion Processing - Text and Office System - Standard Generalized Markup Language(SGML)

ISO 6156. (1987) Magnetic Tape Exchange Format for Terminological/Lexicographical Records (MATER)

ISO/IEC TR 9573. (1988) Information Processing -SGML Support Facilities - Techniques for Using SGML.

ISO 1087. (1990) Terminology - Vocabulary.

ISO 12083. (1994) Information and Documentation - Electronic Manuscript Preparation and Markup

ISO 12200. (1994) Computational Aids in Terminology - Terminology Interchange Format (TIF) - An SGML Application

ISO RT 12618. (1994) Computational Aids in Terminology - Creation and Use of Terminological Database and Text Corpora

Maler, E. & Andaloussi, E. (1996) Developing SGML DTDs from Text to Model to Markup. New Jersey : Prentice Hall.

Melby, A. (1995) "E-Tif : An Electronic Terminology Interchange Format." Computers and the Humanities 29: 159-165

Nkwenti-Azeh, B. (1994) "New Trends in Terminology Processing and Implication for Practical Translation." Aslib Proceedings 46 (3): 67-74

Smith, J. M. (1992) SGML and Related Standards. London : Ellis Horwood.

Superberg-McQueen, C. M. and Burnard, Lou. ed. (1994) "Guidelines for Electronic Text Encoding and Interchange: TEI P3." Chicago : TEI

Travis, B. E. & Waldt, D. (1995) The SGML Implementation Guide. Berlin : Springer-Verlag.

Turner, R. C., Douglass, T. A. & Turner, A. J. (1995) Readme. 1st: SGML for Writers and Editors. New York : Prentice Hall.

여백

9. 한글 정보처리를 위한 표준 입력 라이브러리 연구

동국대학교
변정용

여백

9. 한글 정보처리를 위한 표준 입력 라이브러리 연구

1장. 서론

1절. 연구의 배경

21세기 첨단 정보화 사회를 내다 보면서 우리 사회의 정보화는 심화되어 가고 있다. 이즈음 한글 정보처리 기술은 단순 문자 처리 중심에서 언어 처리 중심으로 환경이 변하고 있다. 여기서 국어 정보처리 기술은 이분야 기술의 중심이며, 정보처리의 기본 대상인 한글 코드를 어떻게 만들 것인가 하는 문제는 아주 중요하다. 그런데 현행의 국가 표준인 완성형 한글 코드 KS C 5601-1987은 단순히 문자 처리 관점에서 제정되었기 때문에 언어 처리에서 사용할 수 없는 심각한 문제를 지니고 있다. 그런데 이제까지 그다지 크게 문제로 드러나지 않은 이유는 아직 대부분의 실용화된 국어 관련 소프트웨어가 워드프로세서처럼 문자 처리 중심이기 때문이다. 하지만 언어 처리 관련 응용에 관한 연구 규모가 날로 확대되고 있고, 비록 적은 수이긴 하지만 일부 응용들이 실용화 단계에 있다. 또한 현행의 2350 음절자로는 다양한 국어 관련 응용들의 요구를 수용할 수가 없는 분야가 점증하고 있다. 여기에 현재의 분위기로 보아 남북한간에 교류가 활성화되고 정보교환이 이루어질 가능성이 점차 높아 지고 있다. 이러한 사실에서 보아 남북이 비록 같은 완성형 체계를 택하고는 있지만 문자 집단이 다르고 배열 순서가 다르다. 뿐만 아니라 전세계의 모든 한민족이 함께 정보를 교환해야 한다는 관점에서 최적한 한글 코드 체계가 마련되어야 한다.

가장 이상적인 한글 코드는 한글 및 국어 정보 처리 응용 전반에서 요구하는 모든 정보를 제공할 수 있어야 한다. 그런 점에서 표음문자로서 음소 및 음절 문자 특성을 가진 한글 문자의 무엇을 코딩의 대상으로 삼을 것인가는 매우 중요하다. 그것은 국제 표준 기구(ISO)에서 정한 코드 제정 규격에 적합한가라는 문제가 있으며, 여기서 코드화 대상 문자의 수가 중요한 요소로 작용하기 때문에 다양성을 가진 한글에서 코드화 대상을 무엇으로 할 것인가는 신중히 고려해야 할 사항

이다.

이제 까지 세 번에 걸쳐서 한글 코드 표준을 개정해 왔지만 한글 코드에 대한 논쟁은 계속되고 있다. 특히 언어 처리가 중심인 국어 정보처리 분야에서 그 부적합성은 심각해서 각 응용에서 완성형을 사용하지 못하고 자모형이나 조합형 한글 코드를 주로 사용하고 있다는 사실을 주목해야 한다. 여기서 저마다 별도의 코드를 사용한다거나 새로운 코드를 만든다고 할 때 따르는 문제는 먼저 기본 한글 환경을 새로이 개발해야 하며, 기존의 코드 체계와 정보 교환을 위한 여러 가지 작업을 해야 하는 문제가 따를 뿐만이 아니라 새로운 코드의 난립으로 인하여 정보교환에서 또 다른 혼란을 야기하게 된다.

이제 더 이상 시행착오를 하지 말고 이 문제를 근본적으로 해결하는 노력이 필요하며, 그것은 곧 국어 정보처리 및 정보교환에 가장 이상적인 한글 코드를 제정하는 것이다. 본 연구는 1차, 2차년도 연구 결과로써 국어 정보처리의 요구를 조사 분석해서 최적한 한글 코드를 개발하는데 주력하여 '정음형-1995'를 개발하고 여기에 몇 가지 문제점을 보완하여 '정음형-1996'를 제정하고, 부가적인 응용 소프트웨어를 개발하였다. 정음형이라 함은 훈민정음의 창제 원리를 따르는 코드라는 의미에서 붙여진 이름이며, 실제로 훈민정음 해례에 나타난 과학성을 그대로 코드 체계에 표현하고 있다. 여기서 훈민정음의 과학성이란 한글 및 국어 정보처리에서 요구하는 모든 요구를 만족시킬 수 있는 이상적인 코드 체계의 핵심 요소이다. 정음형 코드는 훈민정음에서 제정한 초성, 중성, 종성의 합인 45자로써 적은 수이지만 표현할 수 있는 음절자의 수는 무려 약 399억 음절에 이른다. 하지만 훈민정음의 과학성과 컴퓨터의 과학성이 일맥상통하기 때문에 글자꼴만 지원된다면 컴퓨터에서 그 모두를 표현하는 것은 쉬운 일이다. 또한 정음형은 ISO 646이나 ISO 10646과 같은 국제 표준 규격을 비롯하여 PC와 같은 소형 시스템에서 대형 또는 초대형 컴퓨터에 이르기 까지 일관성 있게 적용될 수 있는 코드 체계이다. 그것은 바로 훈민정음의 과학성에 기인한다. 하지만 이제까지 우리는 이러한 과학성을 이해하지 못하고 있으며, 단지 다른 나라의 문자에 견주어 한글을 비교해서 판단하고 있어서 기존의 음성학적 관점에서 파악되어 온 과학성이 아닌 수학적이고 공학적인 관점에서 그 과학성이 무엇인지를 실질적으로 이해할 필요가 있다.

이러한 사실들에 근거하여 1차, 2차 연도 사업에선 일반적인 국어 정보처리 기술 연구 개발에서 주로 사용하고 있는 UNIX 시스템에서 필요한 한글 환경을

구축하였다. 기본적으로 xterm 이 영문만 지원하고 있기 때문에 정음형 한글 코드가 처리되도록 한 Hunterm 을 개발하였다. 이것은 완성형과 조합형을 지원하는 Hanterm 과 유사하다. 또한 국어정보처리 기술 개발에서 완벽한 자소 자료 입력 및 개발 도구로써 사용할 편집기를 개발하였는데 이것은 유닉스 시스템에서 보편적인 편집기인 vi 를 정음형 한글이 처리 될 수 있도록 개발한 것으로 이를 Hunvi 라 한다. 이와 함께 앞으로 정음형 한글 소프트웨어 개발에 있어서 필요한 기본 소프트웨어 개발이 중복되는 문제를 해결하는 방안의 일환으로 계층적 소프트웨어 방법론을 제안하고, 여기에 딸린 70 여개의 요소 루틴을 개발하였다. 여기서 일층에는 정음형 한글 코드를 상정하고 그 위층에는 기본 함수, 그리고 그 위에는 복합 함수를 두고 그 다음에 응용 계층을 두어서 실제 국어 정보처리 기술에서 각종의 소프트웨어를 개발할 때는 무슨 코드에 가능한 독립적인 위치에서 소프트웨어를 개발할 수 있도록 하였다.

다른 한편 현재 PC 의 보급이 보편화되고 앞으로 그 수는 빠른 속도로 확대될 것이며, 연구 개발에 있어서도 PC 의 활용이 높아졌다. 선행 연구 개발에서 이루어진 환경이 주로 유닉스이기 때문에 PC 에도 이러한 환경을 지속적으로 개발할 필요가 있다. 따라서 앞으로 국어 정보처리 기술 개발 결과의 실용화는 현재 우리나라의 PC 보급 대수가 500 만에 이른다는 사실을 감안할 때 PC 를 도외시 할 수 없는 입장이다. 더구나 본 연구의 결과의 파급 효과를 고려할 때 보급이 보편화 되어 있는 IBM PC 호환 기종에서 실행 가능한 결과를 내는 것이 필요하다. 현재 대부분의 PC 는 완성형 한글만을 지원하는 마이크로 소프트웨어의 윈도우 95 를 탑재하고 있다. 이러한 시스템에서도 국어 정보처리를 하려면 무엇 보다 먼저 정음형 한글 환경을 구축하고 완벽한 자소 정보를 지원하는 한글 자료를 입력하거나 정음형 관련 소프트웨어를 개발하는 데 편집기가 필요하다. 윈도우 95 상에서는 정음형 코드를 사용하는 편집기의 개발이 이루어 진 바가 없다. 윈도우 95 는 현행 공업 표준인 KS C 5601-1987 를 따르고 있으며, 윈도우즈 엔티(Windows NT)에서는 유니코드(Unicode)를 기본적으로 지원하고 있다. 초기에 윈도우 95 는 우리 나라 공업 표준에도 없는 통합형이라는 한글 코드를 자체로 만들어서 보급하려고 했으나 지금은 선택적으로 KS C 5601-1987 에서 정의된 2350 자 완성형 코드를 지원하고 있다. 마이크로 소프트(Micro Soft) 사의 통합형 코드의 문제점은 공업 표준과 맞지 않음은 물론이고, 지원하는 11172 자가 2350 음절이 고정 배치되어 있는 상황에서 나머지 공간을 채웠기 때문에 가나다순 정렬이 되지 않는다는 점이 크게 문제가 되었다. 그래서 95 년도 초반에 이러한 문제에 봉

착하여 결국 현재와 같이 2350 자를 기본적으로 지원하고 있다. 그 때 정부에선 유니코드를 표준안으로 수용한다는 결정을 하였고, 윈도우즈 엔티에서 유니코드를 지원하고 있다. 하지만 유니코드에서 지원하는 11172 자는 완성형 체계를 따르고 있기 때문에 국어 정보처리에선 사용할 수가 없다. 물론 유니코드에는 자소형과 자모형 코드가 포함되어 있다. 이들은 복자모에 대하여 부호화를 하였기 때문에 완벽한 자소 정보를 제공하지 못한다. 그리고 구현에 있어서 완성형을 중심으로 하도록 되어 있다. 또한 이들은 완성형과 혼합해서 사용할 수 있기 때문에 혼합 방법을 다양하게 할 수 있으며 이것은 크게 혼란을 일으킬 소지를 안고 있다. 이러한 현실에서 국어 정보처리 기술 개발은 매우 제한적이고 여러 가지 불편한 요소를 안고 있기 때문에 이에 대한 개선이 요구된다.

현재 대부분의 피시가 윈도우 95 를 탑재하고 있기 때문에 개발 대상 환경을 윈도우 95 환경에서 정음형 환경을 개발할 필요가 있다. 그래서 PC 에서 국어 정보 처리 기술을 개발하거나 통신을 통하여 국어 정보 처리 기술을 활용하려면 PC 에서도 정음형 코드를 지원하는 각종의 소프트웨어를 개발해야 한다. 원칙으로 따진다면 정음형 코드가 표준으로 채택된다면 운영체제 수준에서 지원될 것이기 때문에 이러한 문제들은 저절로 해결이 될 것이다. 하지만 현재로써 당장 표준이 정음형으로 변경되기를 기대하기는 매우 어렵다. 왜냐하면 아직 대부분의 응용들이 단순한 문자 표현 수준이고, 한글의 과학성에 대한 이해 수준이 낮으며, 표준이 아닌 소프트웨어를 따로 개발하는 것에 관심을 거의 갖지 않고 있는 현실을 보면 그렇다. 하지만 앞으로 언어 처리 기술 개발이 확대되어 가고 있다는 사실을 감안하면 현재의 완성형 코드의 문제점은 저절로 드러나게 될 것이고, 그 때에 대비하여 정음형 코드의 이론을 정립하고 관련된 기본 소프트웨어를 개발하여야 필요가 있다.

이러한 사실을 감안하여 단계적인 절차를 밟아 가는 것이 필요하다. 그 기본 단계가 이미 규명된 한글 문자의 과학성을 보편화시키는 일로써 정음형을 활용한 소프트웨어를 많이 개발 보급하는 일이다. 그 가운데 편집기는 정음형 데이터를 생성시키는 그 첫걸음이 된다는 점에서 중요하다. 그리고 편집기를 비롯한 각종의 한글 관련 소프트웨어들이 실행하는 바탕이 되는 한글 환경과 관련 기본 소프트웨어를 개발할 필요성이 있다. UNIX 와 PC 모두 윈도우 시스템이 일반화 되면서 윈도우는 바로 한글 환경을 제공하는 기반 도구가 되고 있다. 따라서 유닉스 시스템에서는 xterm 이 되고, PC 의 윈도우에서는 윈도우 95 가 된다. 앞에서도 말했지만 근본적으로 운영체제에서 지원이 되고 다양한 글자꼴과 함께 각종의 한

글 코드를 자동 인식하고 이들을 상호간에 번역하는 프로그램을 지원할 필요가 있다. 덧붙여서 한자의 처리 요구를 포함하는 것이 필요하다. 뿐만 아니라 각 코드 마다 가진 특수 문자들을 상호간에 번역할 방안과 코드 상호간에 표현할 수 없는 한글 글자를 표현하는 방법과 규정을 마련할 필요가 있다. 결국 전체적으로 기본 공통 루틴을 추출하고 좀 더 복합적인 한글 처리 루틴을 정리하여 정음형 종합 라이브러리를 구축할 필요가 있다.

뿐만 아니라 시스템 간 정보나 자료를 교환하려면 통신에서 정음형의 지원이 불가피하다. 따라서 유닉스 시스템에서 email을 교환하는데 사용하는 sendmail을 정음형이 지원되도록 할 필요가 있다. 이를 위해선 ISO 2022에 따르는 ESC 순차 문자열을 종단 문자인 F의 값을 받아야 한다. 또한 문서를 작성하기 위하여 텍스트 처리기도 필요하다. 여기서 옛 한글 표현이 쉽다면 이 쪽에 응용을 가진 분야에서는 매우 유용하게 사용할 수 있을 것이다. 또한 문서를 작성할 수 있도록 LaTex에서 정음형이 처리 되도록 할 필요성이 있다. 이러한 유용한 도구들을 제공함으로서 정음형의 실용성을 검증 받고 이의 보급을 확대해 나간다면 정음형을 표준화는 길을 넓힐 수 있을 것으로 본다.

2절. 연구 목표

1. 최종 목표

이제까지 문제가 되어 온 한글 코드의 문제를 근본적으로 해결하고, 국어 정보처리 기술 개발에서 필요한 문자 처리에 대한 기반 이론과 생산성을 비롯한 표준화를 이룩할 수 있도록 기본 공통 루틴들과 도구(tool)들을 개발하여 이를 라이브러리(Libraries)로 구축하여 이를 보급한다.

1. 대상의 공학을 위한 과학성 규명이라는 명제에 따라서 훈민정음 원리의 과학성을 규명하여 한글 문자 처리 및 국어 정보 처리에 필요한 기반 이론을 확립한다.

2. 훈민정음 원리와 국어 정보 처리의 모든 응용 분야의 요구를 만족하는 최적한 국어 정보처리용 한글 코드를 개발한다.

3. 따라서 최적한 코드의 개발로 모든 관련 분야의 기술 발전을 촉진시킴은

물론 Unicode 와 ISO 10646 등에서 혼란한 한글코드 체계를 단일화하여 표준화 안을 제안한다.

4. 한글 및 국어 정보 처리에 필요한 기본 공통 루틴들과 유용한 한글 처리 도구로서 편집기를 개발하여 지원한다.

5. 이상을 고급언어용 계층형 라이브러리로 구축하여 보급함으로써 한글 문자 및 국어 정보 처리 기술 개발에서 코드에 독립적인 소프트웨어 생산이 가능하게 하여 표준화를 통하여 생산성 향상을 도모한다.

2. 연차별 목표

년도	계획서 상의 목표	개발 결과	달성도
1차 년도	훈민정음 원리의 과학성 규명 및 공학화 방안 연구, 국어 정보처리용 한글코드의 개발 및 보급	1. 훈민정음 해례에 내재된 과학성의 실체를 규명하고, 컴퓨터의 원리와 부합되는 적용 방안을 개발하였다. 2. 이상의 결과를 ISO 646 에 근거하여 단수 바이트 정음형 한글 코드 체계를 개발하였다.	100% 100%
2차 년도	한글 및 국어 정보처리용 기본 공통 루틴 및 도구 개발	1. 미래의 표준 한글 코드로써 정음형의 성능을 입증하고 이 분야 기술 개발의 기본 도구로서 UNIX 환경에서 한글 환경인 Hunterm 과 편집기인 Hunvi 를 개발하였다. 2. 정음형 소프트웨어 개발에 필요한 기본 공통 루틴들을 개발하였다.	100% 100%

3 차 년도	PC 용 한글 및 국어 정보처리용 환경 및 편집기 개발 및 고급 도구와 표준 라이브러리 구축 및 보급	1. PC WIN95 환경에서 실행 가능한 편집기를 개발하였다. 이것은 MFC Visual C++의 기본 콘트롤을 기반으로 하였다. 2. 정음형을 축으로 하는 통합 코드 변환기를 개발하였다. 3. PC 용 정음형 기본 입력 루틴을 개발하였다.	100% 100%

3 절. 내용 및 범위

최종결과물	한글의 과학성 규명, 정음형 코드, 한글입력기-1,2 차년도:Hunterm & Hunvi, 3 차년도: "바른글", 통합 코드 변환기
연구내용 (기능설명)	KS C 5601-1987 완성형 한글을 지원하는 PC Win95 에서 Win32 API 를 이용하여 정음형 한글 코드를 지원하는 윈도우 콘트롤을 개발하고 함께 정음형 한글 코드 텍스트를 대상으로 하는 편집기를 개발하였다. KS C 5715 자판으로 부터 입력된 자소 문자열에 대한 정음형 오토마타를 통하여 정음형 코드 생성, 음절 접수를 한다. 또한 편집에 있어서 일 차원으로 된 정음형 문자열을 음절 단위와 자소(음소) 편집이 가능하고, 파일 입력부에 통합 코드 변환기가 부착되어 북한의 국규 9566-93 을 포함한 기존의 대부분의 한글 코드를 정음형으로 변환하여 모두 편집할 수 있다.
특 징 (기존 제품과의 차별성, 연구결과의 객관	1. "정음형" 한글 코드를 사용한다. 2. 정음형 편집기 "바른글 (HunEdit)"은 현재 잘 쓰이지 않는 옛 한글 자모 넉자를 포함하여 완벽한 자소 정보를 제공한다. 현재 대부분의 편집기는 완성형 한글을 지원하고 있기 때문에 자소(초성, 중성, 종성) 정보를 얻을 수 없어서 국어 정보

성, 우수성 등에 대해 서술)	처리에서 활용할 수 없다. 일부 조합형을 지원하는 편집기가 있으나 이것은 표준화 규격을 위반하고 있을 뿐만 아니라 완벽한 자소 정보를 지원하지 못하고 있다. 3. "바른글"은 글자꼴을 지원하는 한 훈민정음 해례가 생성할 수 있는 최대 음절수인 약 399 억 음절자를 표현할 수 있다. 여기서 표현 음절자의 범위는 글자꼴에 의하여 제한된다. 4. 편집 과정에서 음절자 모드와 자소 모드로 편집 연산이 이원화되어 있다. 다시 말해서 자소 모드에선 커서가 자소 단위로 이동하고, 음절 모드에선 음절자 단위로 이동한다. 5. 북한의 국규 9566-93 을 포함한 기존의 한글 코드 대부분을 편집기에 올릴 수 있다. 정음형 한글 코드는 한글이 지원할 수 있는 전체 음절자 집합을 지원하기 때문에 이의 부분 집합을 지원하는 기존의 모든 한글 코드의 축이 될 수 있다. 편집기 입력부에 통합 코드 변환기를 부착하여 변환을 할 수 있도록 하였다.
비 고	Win 95 의 환경을 잘 활용하기 위하여 MFC Visual C++ 4.2 를 이용하여 개발하였다. 그러나 Editview 와 같은 제공 클래스를 이용할 수 없었던 것은 IME 라는 완성형 코드 생성기가 기본적으로 활용되고 있었기 때문이다. "바른글"에서는 자모형 자판에서 입력하기 위하여 정음형 문자 접수기를 별도로 개발하였다.

4 절. 기대 효과

1. 기술적 측면

첫째, 한글의 과학성이 규명되었다. 바로 훈민정음 해례에서 훈민정음의 과학적 창제 원리를 찾아내고 이를 현대 수학의 관점에서 과학적이라는 사실을 검증하였다. 이 결과는 완성형이 왜 국어 처리에 부적합한가를 규명하는 자료가 되며 또한 한글 코드는 초성, 중성, 종성의 기본자만으로 구성하면 된다는 사실을 증명하는 근거가 된다. 훈민정음은 기본 45 만을 연서법(連書法), 합용 병서법(合用竝書法), 부서법(附書法) 그리고 성음법(成音法) 만으로 무려 천문학적인 숫자인 399 억 음절자를 생성할 수 있다. 다시 말하면 기본 집합과 규칙만으로 399 억 음절자를

정의한 것으로 이것은 컴퓨터의 하드웨어와 소프트웨어에 대비된다.

둘째, 낱 자모 문자열을 처리하는 기술의 발전을 들 수 있다. 여기서 낱 자모란 훈민정음에서 정의한 그 낱 자모를 이른다. 한글의 특성이 음소 및 음절 문자 특성을 가졌음에도 낱 자모 처리 기술이 발전하지 못한 것은 프로그래밍에서 완성형의 단순성 때문이다. 그 단순성이란 완성형에서는 한 음절을 한 뭉치로 다루니까 우선 프로그램하기가 쉽다. 이러한 결과는 초기의 대부분의 응용이 단순히 한글 문자를 컴퓨터에 저장하고 이것을 출력하는 것에 불과하였기에 가능하였다. 설사 정보를 처리한다고 하더라도 음절 단위의 비교가 고작이었으며, 여기에는 언어 현상과 같은 음소 또는 자소 처리 요구가 없다.

사실 초기에는 자모형 코드를 사용하였다. 그런데 자모형은 한 음절의 길이가 가변적이기에 컴퓨터 내부에서 일 차원으로 존재하고 출력될 때는 몇 바이트가 한 음절이 되는 현상은 한글의 입장에선 당연한 것이지만 영문자 입장에선 번거롭게 여겨진다. 자모형은 한 음절의 길이가 가변적이라 하여 이를 n-바이트 코드라 하였다(사실 이것은 잘 못된 용어이지만 통용되는 용어였다). 그 가변성은 영문 소프트웨어로 한글화하는 과정에서 보면 매우 귀찮게 여겨질 만한 이유가 될 수 있다. 또한 출력할 때 한글 한 음절의 복잡도가 영문자 보다도 높아서 한글 한 음절자에 영문 두 글자의 폭과 같도록 했는데 한글 음절의 구성 문자열의 길이가 가변적이라는 것은 매우 귀찮은 일이다. 그래서 두 바이트에다 초성, 중성, 종성을 5 비트에다 조합해서 넣은 코드가 나왔고, 종국에는 완성형이 출현하여 프로그램을 하기에 좋은 환경을 만들긴 하였지만 자소에 관한 모든 정보를 상실하고 말았다.

완성형의 문제는 표현할 수 있는 문자 수를 제한하여 표현의 자유를 제약한 것이며, 특히 언어 처리에서는 자소 정보가 없다는 점에서 심각한 문제였다. 그래서 96 년도 후반기에 유니코드를 국내 표준으로 수용하면서 현대 한글의 전체 집합인 11172 자를 모두 반영하여 문자 표현의 한계를 풀어 보려는 의지를 보였지만 옛 한글의 표현은 자소형 코드를 혼합하여 사용해야 가능하다. 여기서 코드계를 혼합해서 사용한다고 할 때 매우 커다란 혼란이 발생할 수 있다. 예를 들어서 자모형까지 혼합이 가능한데 이 세가지 코드계를 혼합하면 '국'이라는 음절은 무려 6 가지 이상의 코드 혼합이 가능하기 때문이다.

정음형 코드로 된 문자열 처리 기술은 컴퓨터 내부에선 일 차원의 자소 형태

로 존재하고, 출력될 때는 이 차원의 음절 형태로 보이게 한다는 원칙에 따라 개발하였다. 이것은 단순히 어떻게 처리하든 결과만 맞으면 된다는 논리가 아니라 훈민정음의 과학적 원리를 그대로 컴퓨터에 적용하는 기술을 개발한 것이다. 훈민정음 원리를 그대로 컴퓨터에 적용한 것은 물론 그것이 과학적이기 때문이기도 하지만 이제 까지 한글을 로마자나 가나의 형태로 고쳐서 쉽게 프로그래밍하여 왔던 관례를 깨뜨린 점에서 뜻이 있다. 다시 말하면 시험 문제가 어렵다고 해서 자신이 아는 문제로 고쳐서 문제를 푸는 방식을 취하여 온 것이 한글 처리의 전력이다. 그것도 낱 자소에 대한 정보를 모두 잃어 버려서 언어 처리를 할 수 없게 된 상황에서 그러하다. 앞으로 정음형 한글 처리 다시 말해서 완벽한 낱 자소 풀어 쓰기를 처리하는 기술의 기본 기술들이 개발되었기 때문에 이것을 지속적으로 확대 발전시켜야 할 것이다. 일단 본 연구에서 개발한 편집기와 기본 루틴들은 앞으로 이 분야 연구에 초석이 될 것이다. 이러한 노력의 종국에 가서는 본격적으로 국어 정보 처리를 위하여 정음형 한글 코드를 표준으로 채택할 수 있도록 하여야 할 것이다.

2. 경제적 측면

국어 정보 처리와 같은 언어 처리 분야에선 현재 또는 미래에 표준으로 삼고 있는 완성형 코드를 사용할 수 없기 때문에 어차피 기존의 조합형을 쓰거나 아니면 필요에 따라서 자신이 적합한 코드를 만들어서 써야 한다. 이것은 첫째, 저마다 코드를 만들어 쓸 경우 여러 가지 요인에 따라서 비경제적이다. 앞서도 언급하였지만 조합형을 사용한다 해도 복자모에 대한 자소 정보를 가지고 있지 않기 때문에 이들에 대한 자소 정보를 지원하도록 하는 데 노력이 많이 든다. 둘째, 각자 필요에 따라서 한글 코드를 만든다면 관련 기본 공통 소프트웨어를 새로이 각자가 개발해야 하기에 비경제적이다. 다시 말해서 중복 투자를 없애고, 연구 개발의 시간과 노력을 절약하자는 것이다. 셋째, 이렇게 만들어 진 자료를 상호 직접 교환할 수가 없기 때문에 비경제적이다. 그러나 현재 까지 나온 모든 코드는 현상적으로 또는 경험적으로 해당 코드의 출현이나 존재에 대한 정당성을 말할 뿐 옛 한글을 포함하는 과학적 원리를 제시하지 못하였다. 여기서 개발한 이론과 편집기와 같은 도구들은 앞으로 정음형 프로그램을 작성하거나 자료를 입력할 때 기본적으로 필요한 장비가 된다. 이 분야에 대한 연구를 지속적으로 하려는 경우에는 이러한 도구를 사용하기 때문에 경제성을 얻을 수 있다.

또한 크게 보면 국어 정보처리 기술은 우리의 고유 기술이 될 수 있다. 하지만 현행의 완성형 코드에선 그렇지 못하다. 왜냐면 완성형 코드를 사용함으로써 한글의 구현이 쉬워져서 우리 고유의 기술이 되지 못하고 있다. 완성형이 결국 버려야 할 표준이라면 미리 정음형을 보급하고 이에 대한 소프트웨어 공학 기술과 정음형을 이용한 응용을 개발하여 사전에 충분한 경험을 쌓아야 한다.

3. 관련 산업기술 및 타 연구 개발에 미치는 파급효과

무엇보다도 국어 정보처리 분야의 연구에 절대적인 영향을 미칠 것이다. 이제까지 기존의 관념은 완벽한 풀어쓰기는 취급하기가 곤란하다는 선입견을 가지고 한글을 본다. 그러나 편집기 개발 결과를 보면 아마도 달라질 것이다. 편집기는 자료 입력이나 프로그램 개발에서 사용할 수가 있기 때문에 정음형 코드로 개발하는 각 소프트웨어에서는 이를 이용하면 기초 작업부터 새로 개발할 필요 없이 본론에 들어 갈 수 있다고 본다.

현재 완성형 한글 소프트웨어 개발에 있어서 우리의 기득권은 거의 없다. 완성형은 자소 정보를 가지고 있지 않기 때문에 국어 정보처리와 같은 언어 정보처리에서 활용할 수 없다. 반면에 자소 문자를 사용하는 영문자권이나 음절문자를 사용하는 한자권 등에서 2 :1 또는 1 : 1로 쉽게 대응할 수 있도록 해 준다. 결국 언어 처리를 할 수 없으면서 한글 소프트웨어 개발을 하는데 조차 그 용이성을 제공한다는 것은 매우 어리석은 일이다.

앞으로 국제 표준 코드인 유니코드에는 현대 한글 음절자인 11172 자가 들어가 완성형 형태로 들어 있다. 또한 복자모와 복자소를 코드화 대상으로 삼고 있는 240 자 자소형 코드와 51 자 자모형 코드가 함께 포함되어 있다. 이것은 이미 세 가지 코드 체계를 포함하고 있기 때문에 표준 속에 다시 비표준의 소지를 만들어 둔 격이다. 왜냐면 세 가지 코드 체계는 적어도 두 가지를 혼합해야 현대 한글을 표기함에 조차도 자유로울 수 있기 때 "ㄱ은 가의 첫소리 이며, ..."라고 표현할 때 자소형 또는 자모형과 완성형을 혼합해서 사용해야 하기 때문이다. 이것은 앞으로 표준 한글 코드로 부적합하다. 그리고 누차 언급했듯이 완성형은 언어 처리 분야에서 사용할 수 없는 코드이다.

그렇다면 현행의 한글 코드를 비롯하여 미래의 코드 조차 한글 및 국어 정보처리에 부적합하다면 우리는 이를 개선해야 할 것이다. 그 대안으로써 훈민정음

창제를 원리에 부합하고 컴퓨터의 특성과 국제 문자 코드 표준 규격을 기반으로 그 구현 방안을 마련하고 있는 정음형 한글 코드는 한글 및 국어 정보처리의 요구를 모두 만족하는 이상적인 코드라 할 수 있다. 그런 점에서 미리 이 분야 기술을 개발하고 관련 소프트웨어를 준비해 두는 것은 한글의 특성을 가장 잘 반영하고 있는 정음형 코드가 표준 코드로 되었을 때 국내의 정보 산업이 이 분야의 기술을 선도할 수 있을 뿐만 아니라 한글 처리 관련 기술의 기득권을 누릴 수 있을 것이다. 또한 완벽한 자소 정보의 표현이 가능한 편집기와 관련 기본 공통 루틴을 활용하여 특히 언어 처리와 관련된 응용에서 활용한다면 상당한 경제적 이익을 얻을 수 있다.

2장. 한글의 음소 및 음절 문자 특성

1절. 한글의 과학성

1. 과학적 접근에 의한 문자 창제

과학이란 어떤 대상에 대하여 존재하는 모든 경우에 대하여 적용할 수 있는 통일적 원리를 발견하는 것이다. 여기에 근거할 때 훈민정음은 해례본(解例本)의 정인지서에서 "천지 자연의 소리가 있으면 반드시 이를 적을 수 있는 천지 자연의 글자 (有天地自然之聲 則必有天地自然之文)"가 있다라고 하면서도 배우기 쉽기로는 "지혜로운 이는 아침을 마치기도 전에 깨우치고, 어리석은 이라도 열흘이면 배울 수 있다 (故智者不終朝而會. 遇者可浹旬而學)"고 하여 정말 이율 배반적인 양극단의 특성을 가지고 있다. 다시 말하면 훈민정음은 무한대에 가까운 천지 자연의 소리를 적을 수 있는 한 가지 문자의 원리를 만든 것이라고 할 때 여기에 과학성이 있다고 할 수 있다. 그러면 좀 더 자세히 한글의 과학성을 구체적으로 파악하기 위하여 훈민정음 해례본의 내용을 분석해 본다.

2. 훈민정음 해례의 분석

훈민정음 해례에서 먼저 제자해를 통하여 발성 기관을 기초로 하여 우리말과 한자말의 기본 음을 도출해 내고 거기에 소리의 거셈과 장단에 따라서 다양한 소리가 파생할 수 있다고 하였다. 그래서 초성해에서 닿소리 17 자를 정의하고, 중성

해에서 홀소리 11 자를, 그리고 종성해에서는 초성해와 음가는 다르지만 글자꼴은 17 자로 같다고 정의하였다. 그리고 다시 합자해에서 입순 가벼운 소리는 입술소리 'ㅁ ㅂ ㅸ ㅍ' 아래에 'ㅇ'을 붙인다고 하고, 합용병서를 하는 방법은 초성이나 중성을 각각 두 자 또는 석 자까지 함께 나란히 쓸 수 있다고 하였다. 여기서 크게 확장된 초성자, 중성자, 종성자 집합을 가지고 음절을 구성하는 방법으로 가로꼴 모음은 초성 아래에, 세로꼴 모음은 초성 오른 녘에 쓴다고 하였다. 그리고 이렇게 모아 쓰는 것은 소리를 내는 단위로 한다는 '무릇 글자는 모아야 소리를 이룬다'하는 성음법을 마지막으로 정의하고 있다. 여기서 훈민정음 해례본의 내용을 통해서 천지 자연의 소리가 어떻게 표현할 수 있는지를 알아 보기 위하여 표현 가능한 음절수를 산출하여 보면 다음과 같다.

- 초성해 : 자음 17 자
- 중성해 : 모음 11 자
- 종성해 : 종성부용초성(終聲附用初聲)
- 합자해

  ㅇ 연서법 : ㅁ, ㅂ, ㅸ, ㅍ 아래 'ㅇ'을 붙인다.

  ㅇ 합용병서법 : 초성과 중성은 각각 2 또는 3 자 조합한다.

  - 초성자 : 17x17x17 + 17x17 + 17 = 5219 자
  - 중성자 : 11x11x11 + 11x11 + 11 = 1413 자
  - 종성자 : 5219 + 1 (받침없음) = 5220 자

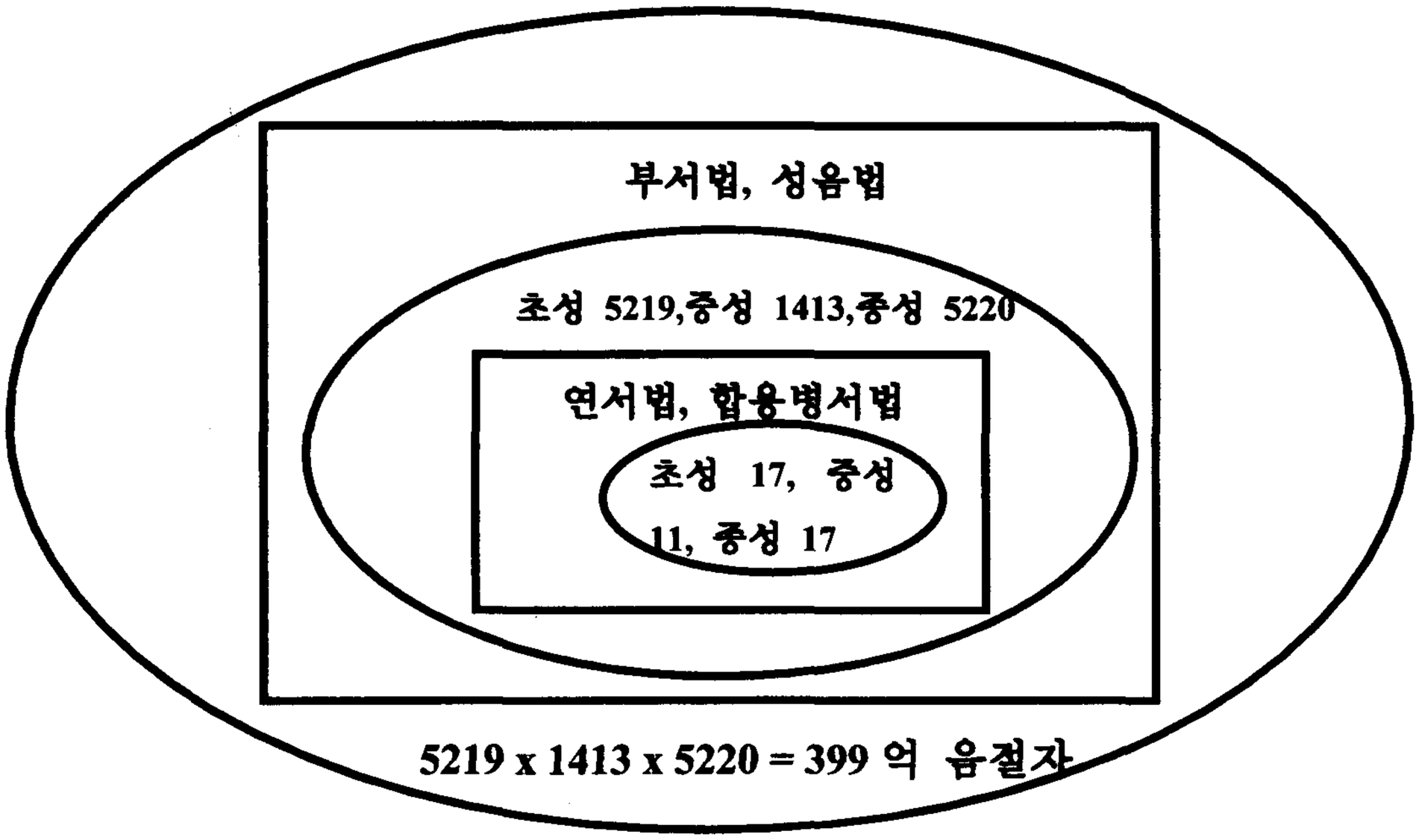
  ㅇ 부서법 : 5219 x 1413 x 5220 = 399 억 음절자

  ㅇ 성음법 : 무릇 글자는 모아야 소리를 이루나니
- 용자례 : 초,중,종성자의 예를 들어 보임

이상에서 보면 훈민정음은 약 399 억 음절자를 생성할 수 있도록 설계되었다. 훈민정음에서 만든 기본 글자는 실제로 45 자이지만 종성과 총성이 음가는 서로 다르지만 글자꼴이 같기에 이를 글자꼴에 있어서 동치관계로 정의하여 이를 종성부용초성이라하고 28 자로 정의하였다.

과학성은 바로 이것을 말하는 것이다. 한자나 가나처럼 모든 소리를 다 정의하는 것이 아니라 생성이 가능한 문자는 기본 자모와 규칙을 통하여 생성하도록 만들었다. 과학은 단순 명료함을 제공하며 통일된 원리는 많은 경우를 포괄하여

표현할 수 있다. 훈민정음 원리의 생성 원리를 도식하면 다음과 같다. 그리고 표현 가능한 모든 음절자를 표기하지 않고 그들을 만들어 낼 수 있는 생성 원리를 제정함으로써 다시 말하면 기본 자모와 규칙을 정의함으로써 음절자 집합의 크기를 최소화 하였다. 이것은 천지 자연의 소리를 표현할 수 있는 하나의 통일된 원리이다. 그래서 배우기 쉽고 쓰기 쉬운 문자를 만들 수 있다.



2.1 훈민정음 원리도

3. 훈민정음 원리의 해석

훈민정음 원리를 제대로 이해하기 위하여 현대적 관점을 가지고 해석해 본다. 먼저 수학적으로 보면 "천지 자연의 소리"라는 천문학적 문자를 가정하였고, 이 무한 수에 가까운 천지 자연의 소리를 표현할 수는 없다. 하지만 소리는 하여간 유한하다고 가정할 때 그 수를 얼마로 할 것인가의 문제가 있으며 그렇지만 거의 천문학적인 수에 가까울 것이다. 해례에서 보면 기본 28 자는 원소 나열법으로 정

의하고 나머지는 규칙으로 정의하였다. 이것은 현대 수학의 집합론에서 원소의 수가 많아서 나열할 수 없을 때 조건 제시법으로 표현한 것과 같다. 그림에서 보듯이 약 399 억 음절자를 28 자와 몇 개의 규칙으로 표현한 것이다.

문자론에서 훈민정음은 표음문자로서 음소 및 음절 문자의 특성을 가졌다. 낱 음소에 대한 글자를 가졌으면서도 소리 마디 단위로 모아 쓰기를 하도록 규정하였다. 이것을 현재 문화와 정치적으로 매우 밀접한 관계에 있는 로마자와 가나 문자에 비교해 보면 가나는 표음 문자로서 음소가 없는 음절문자이고, 로마자는 음절 구조가 없는 음소문자이다. 그런데 한글은 이 양자의 특성을 모두 가지고 있다. 바꾸어 말하면 가나나 로마자는 반쪽 한글에 해당한다.

컴퓨터 공학적으로 보면 28 자는 고정된 것이므로 하드웨어적 성격을 가졌고, 규칙은 소프트웨어적 성격을 가졌다. 컴퓨터 시스템은 하드웨어와 소프트웨어로 결합된 시스템이므로 훈민정음은 컴퓨터 시스템과 매우 닮은 꼴이며 구현에 매우 잘 맞는 문자이다. 우리가 현재 겪고 있는 불편은 컴퓨터가 바로 음소문자 특성만을 가진 로마자 문자권에서 개발되었고, 현재 그들의 기술이 우리 보다 우수하기 때문이다.

국어 정보처리에서 훈민정음은 음소 문자의 특성을 가졌기 때문에 낱소리에 대한 음가를 가장 충실하게 담고 있기에 언어 정보처리를 하는 응용에 매우 적합하다. 이러한 여러 장점을 잘 살려서 코드화 대상을 한글의 특성에 맞도록 낱자소에 두어야 언어 정보처리에 우수한 문자가 됨으로써 우리 민족의 우수성을 검증할 수 있고, 우리의 국어 정보처리 기술의 발전 속도를 더욱 높일 수 있을 것이다.

2 절. ISO 규격에 기반한 정음형 한글 코드 표

문자 코드는 BCD 에서 출발하여 EBCDIC 에서 다시 ASCII 로 발전하였다. ASCII 는 처음에 7 비트의 일 바이트 코드계이었으나 최근에는 8 비트의 일 바이트 코드계로 바뀌었다. 그러다가 컴퓨터의 보급이 확산되어 나라마다 자국 언어 처리가 일반화 도어 가면서 코드의 확장이 필요하여 코드 확장 규격인 ISO 2022 를 제정하였다.

이 규격은 록킹 쉬프트 코드를 많이 사용하여 문자의 수용의 한계와 정보교환

과 내부 처리에 많은 문제가 발생하여 이를 단순화 하려는 노력의 결과 ISO 10646 과 유니코드(Unicode)로 발전하여 지금은 256 x 256 을 문자 표현 범위로 하는 형태로 발전하였다. 이제 마이크로 소프트의 윈도 NT 에서 유니코드를 기본 코드 체계로 채택하였다. 그리고 선 마이크로 시스템의 운영체제인 솔라리스 2.6(Solaris 2.6)에서도 기본 코드 체계로 채택하여 97 년 후반기에 발표할 예정이라고 한다. 그러나 아직 대부분의 응용 프로그램들은 ISO 646 과 ISO 2022 규격의 기반에 있다. 따라서 문자 집단의 크기가 94 보다 작은 정음형의 경우 ISO 646 규격에 기반할 때 간단하며 그 코드표는 다음과 같다.

	8	9	A	B	C	D	E	F
0					ㅎ			ㅎ
1				ㄱ	ㅇ	ㅏ	ㄱ	ㅇ
2				ㄴ		ㅐ	ㄴ	
3				ㄷ		ㅑ	ㄷ	
4				ㄹ		ㅒ	ㄹ	
5				ㅁ		ㅓ	ㅁ	
6				ㅂ		ㅔ	ㅂ	
7				ㅅ		ㅕ	ㅅ	
8				ㅿ		ㅖ	ㅿ	
9				ㅇ		ㅗ	ㅇ	
A				ㅇ		ㅘ	ㅇ	
B				ㅈ		ㅜ	ㅈ	
C				ㅊ		ㅝ	ㅊ	
D				ㅋ		ㅡ	ㅋ	
E				ㅌ		ㅣ	ㅌ	
F				ㅍ		ㆍ	ㅍ	

표 2.1 ISO 646 기반 정음형 코드 표

3장. 통합 한글 코드 변환기

1절. 필요성

국어 정보처리의 요구에 따라서 다양한 한글 코드가 존재할 수 있다. 이렇게 해서 여러 가지 코드로 만들어 지면 이들간의 정보교환이 필요할 때 한글 코드 변환기가 필요하다. 이러한 요구에 대하여 기존의 한글 코드 변환은 주로 1 : 1로 이루어져 왔다. 그래서 새로운 코드가 만들어 지면 양방향 변환이 이루어져야 하기 때문에 기존하는 코드의 배수 만큼의 변환기가 필요하다. 예를 들어서 4개의 코드가 존재할 때 새로운 코드와 변환하려면 8개의 변환기가 필요하다. 이것은 코드가 많아졌을 때 보다 많은 변환기를 개발해야 하며, 만약 코드에 수정이 발생하면 그에 비례하는 양의 수정이 이루어져야 한다. 이처럼 시간 소모적인 일을 줄이려면 어떠한 코드를 축으로 하는 번역을 한다면 새로운 코드가 출현하면 축이 되는 코드와의 양방향 변환기 하나씩을 작성하면 된다. 이것은 매우 경제적인 방법이다.

기존하는 코드 가운데 어떠한 코드가 축이 될 수 있는가? 그 조건은 무엇이어야 하는가? 그것은 표현할 수 있는 음절자의 집단이 가장 큰 코드계가 되어야 한다. 앞에서 제안서 정음형 코드는 399억 음절자를 표현하는 코드계로서 기존하는 모든 코드를 부분 집합으로 포함하는 코드계이다. 예를 들어서 KS C 5601-1974 51자 낱자형(자모형), KS C 5601-1982 조합형, KS C 5601-1987 완성형(2350자), 북한 국규 9566-93 조선글 음절자(2420자), 최근 유니코드의 한글 코드들은 각각 표현할 수 있는 최대 글자수가 30만자를 넘지 않는다. 따라서 정음형 코드계를 축으로 하고 모든 코드는 정음형 코드와 대응을 하는 변환기를 양방향에 대하여 개발하면 기존하는 코드와 쉽게 가능하다.

2절. 대상 한글 코드의 특성 분석

1. KS C 5601-1974 자모형

KS C 5601-1974는 한글 자모를 기본으로 복자음과 복모음을 포함하여 도합 51자에 대하여 부호를 부가한 최초의 표준안으로 이를 정보교환용 부호라고 하였다. 특성은 그림 2.1에서 보는 것과 같이 한글의 자음은 ASCII의 2, 3 열에, 모음은 4, 5 열에 배치되었고, 한글과 영문의 혼용에서는 SI, SO를 호출하여 구분하

였다.

2. KS C 5601-1982 (조합형)

KS C 5601-1974 의 문제점으로 일반이 느끼는 것은 한글의 음절이 개음절과 폐음절로 되어 있어서 한 글자의 길이가 일정하지 않고, 또한 영문과 같은 선상에서 한글 정보처리를 하기가 어렵다고 하여 조합형을 표준안으로 개정하였다. 조합형은 그림 2.3 과 같이 초성, 중성, 종성의 구성이 19, 21, 28 (공간포함)로 되어 있다는 점에 착안하여 2byte 를 5bit 씩 나누어 초성, 중성, 종성으로 나타내고, 최 상위 비트는 한글인 경우 1 로 표시하여 영문과 구분할 수 있게 한 것이다. 그러나 조합형은 결정적으로 ISO 2022 코드 확정법과 KS C 5601-1977 에 위배된다. 결과적으로 코드 체계로 보아 단수 바이트가 코드체계에서 복수 바이트 코드 체계로 바뀌었고, 부가적으로 한글 낱자형 코드에 대한 수정이 이루어 졌다. 즉 2, 3 열의 자음을 영어의 대문자 지역인 4, 5 열의 한글 모음을 영문 소문자 지역인 6, 7 열에 재배치 하였다.

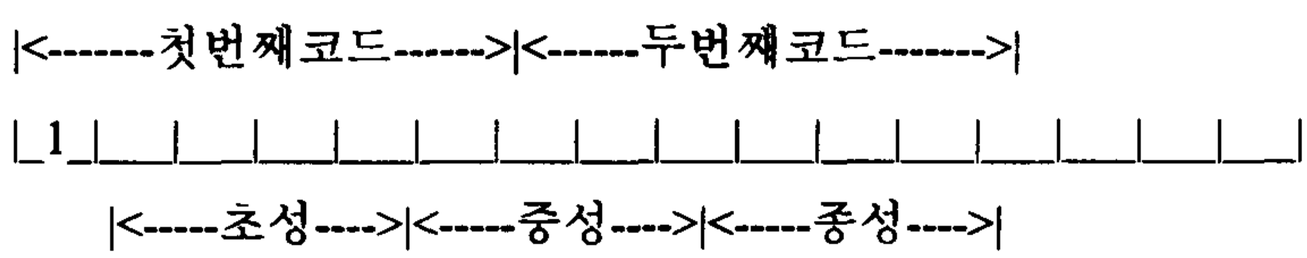


그림 1 조합형 2-Byte 의 구성

3. KS C 5601-1987 완성형 (2350 자)

조합형은 ISO 2022 규정에 위배되어 국제 표준이 될 수 없다는 점, 업체마다 코드가 달라서 정보교환에 불편한 점, 일부 운영체제의 한글화 과정에서 완성형을 취한점, 국가 기관 전산망의 구축에 있어서 코드 통일의 필요성이 증대되었다는 점등이 새로운 코드의 통일안의 마련의 동기가 되어 완성형 코드를 1987 년에 정보 교환용코드로 개정하였다.

KS C 5601-1987 은 한글 음절의 빈도수 0.001% 이상인 2350 자와 일정한 응용분야가 없는 4888 자의 한자 및 380 자의 특수문자, 외국문자, 한글 낱자등을 포

함하고 있는 데 이는 음절에 대하여 코드를 부여했으므로 일명 완성형 코드라 한다. 본 코드는 ISO 규정에 부합되며, 2 Byte 용 소프트웨어에 호완성을 가지는 일본, 대만에 가까운 코드계가 되었다.

4. 국규 9566-93 조선글자의 부호 조선글 음절자 1 수준 (2420 자)

우리나라의 완성형과 같은 유형을 가지고 있으며 북한의 가나다순인 음가 순서로 되어 있다. 크게 차이점은 북한은 초성의 ㅇ은 음가가 없기 때문에 ㅎ 다음에 배열하고 있다. 또한 ㄲ,ㄸ,ㅃ,ㅆ,ㅉ가 뒤로 가 있는 것이 다른다. 북한의 조선글 완성형은 배열순서가 다르고 음절자 2420 자를 선정에서 일부 다른 것도 있다. 숫자로 보아 우리 보다 70 자가 더 많으며 음절의 배열 순서는 완전히 다르다.

5. KS C 5700-1995 (UCS)

UCS 코드는 2 Byte (256*256=65537 자) 로 전세계 언어 (스트립트)와 기호를 표현했다[5]. UCS 코드중 0xac00 에서 0xd7a3 까지 한글 완성형 11172 자를 배치했다. ISO 2022 코드 확정법은 상당히 복잡하며, 이 확정법에 따라 등록된 각 나라문자 모두를 지원하는 소프트웨어는 아직 없다. 다만 각 나라마다 제 각기 스스로의 환경만을 구축하여 사용하고 있을 뿐이다. 이처럼 코드화해야 할 각국의 문자 세트의 수는 늘어감에 따라, ISO 2022 코드 확장법의 복잡한 규격을 쓰기 쉽게 하기 위한 새로운 노력의 결과가 ISO/DIS 10646 과 Unicode 이며 이 두 코드계는 통합되어 2 Byte 체계, 즉 Basic Multilingual Plane (BMP) 가 정해져 있고, 94 문자틀에 제한 받지 않고 내부 코드로 사용할 수 있다. 이와 같이, 제어 문자에 대한 배치를 고려하지 않았으므로 ISO 10646 자소형은 ISO 646 에는 적합하지 않다.

KS C 5601-1987 완성형과 KS C 5601-1982 조합형, KS C 5601-1974 51 자 낱자형 (자모형) 코드 그리고 자소형 가운데 정음형-1995 를 비롯하여 최근에 유니코드를 수용하여 그림 2.3 에서 처럼 국가 표준으로 제정된 KS C 5700-1995 에는 기존의 표준이 되었던 코드 체계를 포함한 자소형 등 세가지 코드가 포함되어 있다.

6. 정음형-1995

정음형 코드는 국어 정보 처리 응용과 같은 구현상의 요구 뿐만 아니라 훈민정음 해례에서 정의한 한글 문자 구조 원리를 규명하고 이를 공학화 하는 방안에 따른 요구를 수용하고 있다. 자소형으로서 초성, 중성, 종성을 각각 1Byte 로 나타내고, 복자모는 해당하는 것을 차례대로 쓴다. 예를 들어 초성의 'ㄲ'은 'ㄱ'(0xb1)을 2 번(0xb10xb1)써서 나타낸다.

3 절. 통합 코드 변환기 구현

1. 정음형 피봇 변환

정음형의 음절의 크기는 훈민정음 창제 당시의 정의를 수용하고 있기 때문에 한글 음절의 전체 집합이다. 이러한 정의에 따라 기존의 한글 코드로 표현이 되어 있는 모든 코드는 정보의 손실 없이 정음형으로 변환할 수 있다.

그러나 정음형을 기존의 코드로 변환하는 것은 정보의 손실을 유발할 수 있다. 표 1 을 토하여 각 코드 별로 음절자 표현의 크기를 비교하여 정음형이 축이 될 수 있는 타당성을 검토해 보면 표현 방법, 자소 정보 제공 등이 완벽한 정음형이 축이 될 수 있다.

코드명	표준 번호	코드화 대상	byte	국제표준규격	표현음절수
완성형	KS C 5601-1987	음절	2	ISO 646/ISO 2022	2350
조선글 완성형	국규 9566-1993	음절	2	ISO 646/ISO 2022	2420
조합형	KS C 5601-1982	자소.음절	2	위반	11172
자모형	KS C 5601-1974	자음.모음	1	ISO 646/ISO 2022	11172
정음형		초성.중성.종성	1	ISO 646/ISO 2022	399 억
유니코드	KS C 5700-1995	자소.자모.음절	2	ISO 10646	11172 +24 만

표 1. 기존 코드의 비교

그림 2 은 기존의 1:1 코드 변환 방법을 나타내며 이것은 새로운 코드가 나타나면 기존에 존재하는 각각의 코드에 대한 변환 루틴이 필요하다. 그림 3 는 정음형 중심 통합 한글 코드 변환기를 나타내며 이것은 새로운 코드에 대해서 정음형과 상호 변환이 가능하도록 하면 기존의 다른 한글 코드들 간의 변환이 가능해진다.

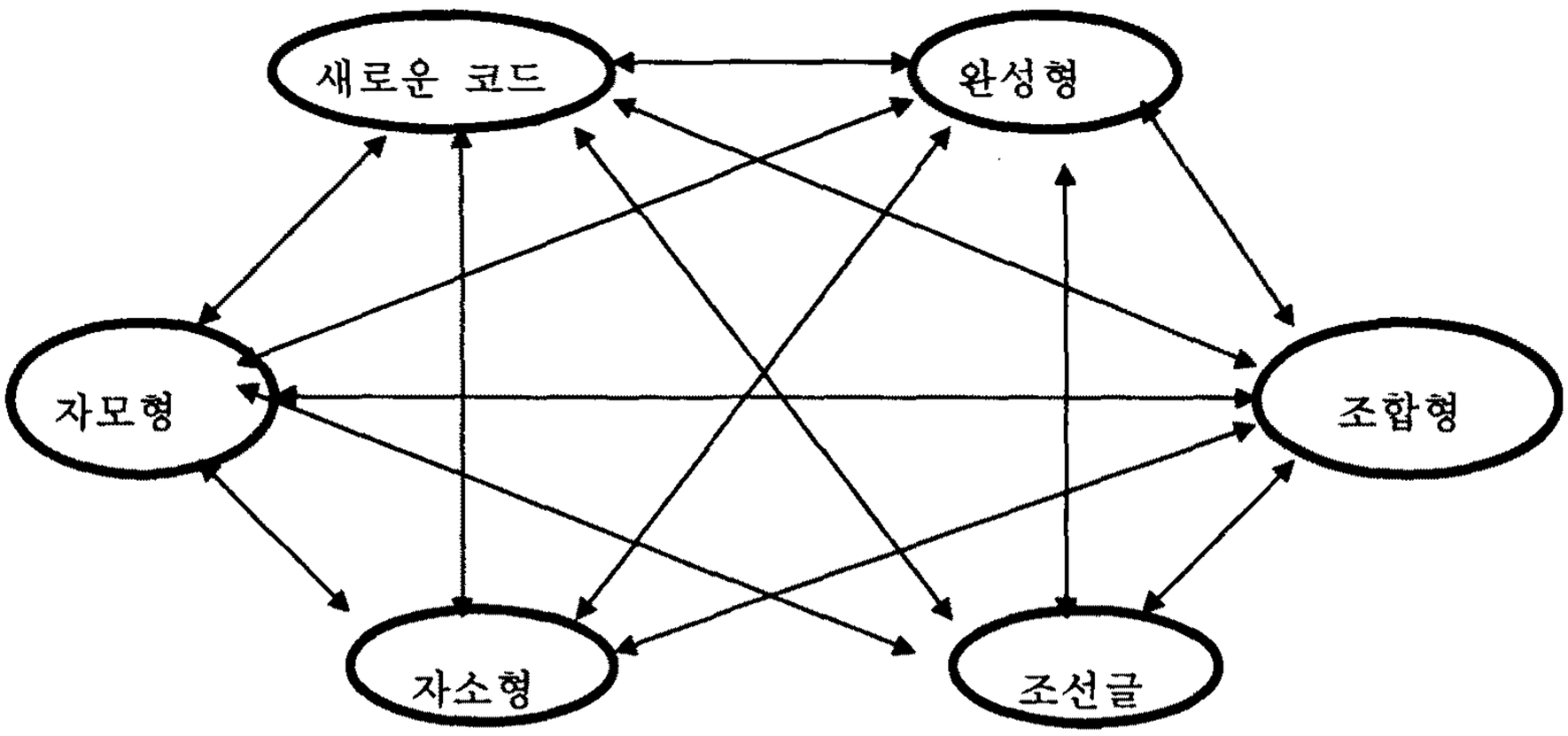


그림 2. 1:1 코드 번역

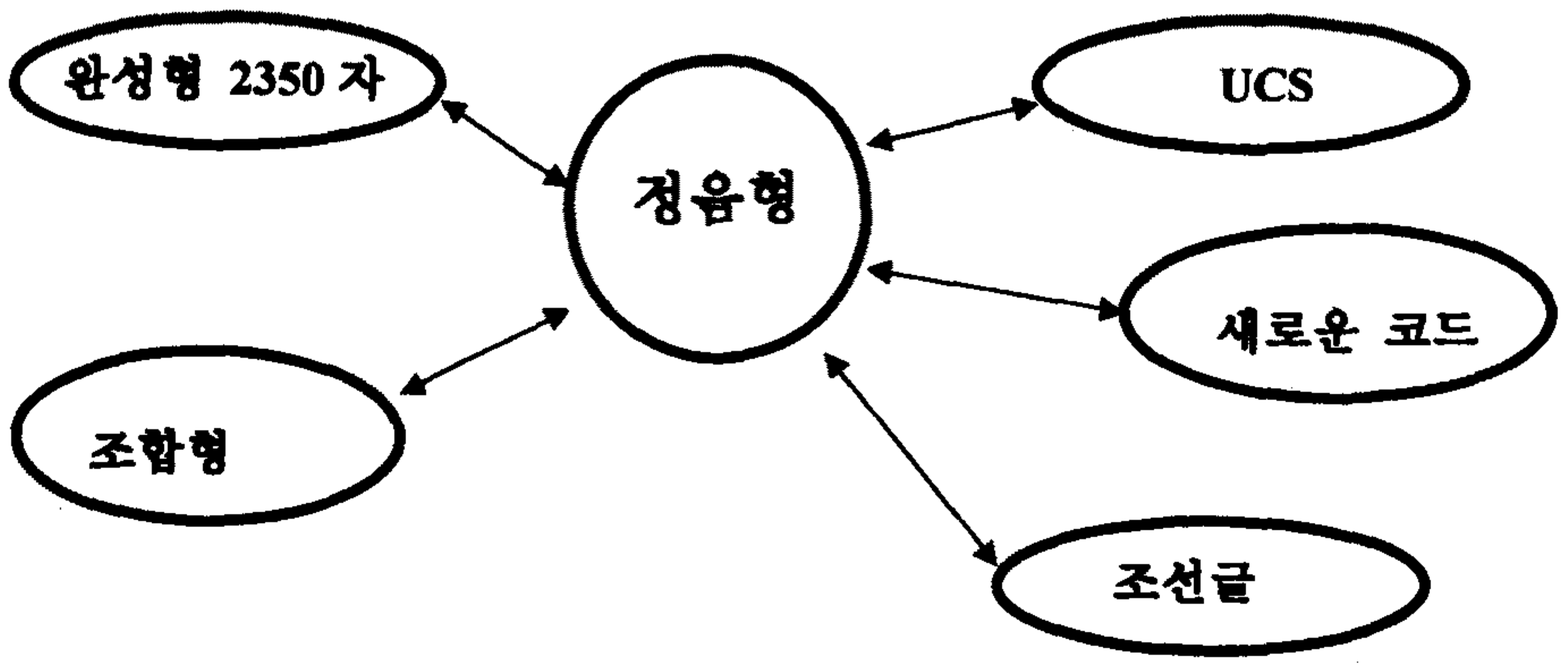


그림 3. 정음형 중심 코드 번역

여기서 정음형이 코드 변환의 축이 될 수 있는 것은 한글 음절의 전체 집합을

표현할 수 있기 때문이다.

2. 배열 초기화

(1) 조합형 코드에 대응하는 정음형 2 차원 배열

초성(3*19), 중성(3*21), 종성(3*28)

(2) 완성형 11172 자에 대응하는 정음형 2 차원 배열

(7 * 11172)

(3) 완성형 2350 자에 대응하는 정음형 2 차원 배열

(7 * 2350)

(4) 완성형 2420 자에 대응하는 정음형 2 차원 배열

(7 * 2420)

3. 코드간 일치화

(1) 정음형-1995 <---> KS C 5601-1982 조합형

1) 정음형-1995 ---> KS C 5601-1982 조합형

정음형 한음절을 초성, 중성, 종성으로 분리하여 끊어 읽는다.

배열에서 같은것을 찾아서 배열번째가 조합형 코드 값이다.

sign + 초성 + 중성 +종성 = 조합형 코드

초성, 중성, 종성중 없는 것이 있으면 fill code 를 출력한다.

(초성 1, 중성 2, 종성 1)

2) KS C 5601-1982 조합형 ---> 정음형-1995

조합형 2byte를 읽어서 초성, 중성, 종성 값을 안다.

초성, 중성, 종성 각각의 값을 배열의 index로 하여 정음형 값을 안다.

정음형은 음소로 나뉘어 지므로 초성, 중성, 종성 각각을 출력할 수 있다.

(2) 정음형-1995 <---> KS C 5700-1995 완성형 11172자

1) 정음형-1995 ---> KS C 5700-1995 완성형 11172자

정음형 한글 한음절을 끊어 읽는다.

배열(7*11172)에서 같은 값을 찾아서 배열에서 몇번째(i)인지 안다.

출력값 = 0xac00 + I

2) KS C 5700-1995 완성형 11172자 ---> 정음형-1995

완성형 2byte를 읽는다.

index = 입력값 - 0xac00

배열(7*11172)에서 index 번째 값을 출력한다.

(3) 정음형-1995 <---> KS C 5601-1987 완성형 2350자

1) 정음형-1995 ---> KS C 5601-1987 완성형 2350자

정음형 한 음절을 끊어 읽는다.

배열(7*2350)에서 같은 것을 찾는다.

첫번째 바이트 = i / 94 + oxa1

두번째 바이트 = i % 94 + 0xb1

완성형에 없는 글자는 풀어 쓴다.

2) KS C 5601-1987 완성형 2350 자 ---> 정음형-1995 완성형 2byte 를 읽는다.

(첫번째 byte - 0xa1) * 94 + (두번째 byte - 0xb1) = index

(4) 정음형-1995 <--->9566-93 조선글 음절자 2420 자

(3)과 같다.

4 장. KS C 5700-1995 에 정음형 코드의 적용

1 절. UNICODE 와 ISO 10646

인터넷과 같은 컴퓨터 통신이 발달하고, 일반 사회의 컴퓨터 사용이 보편화 됨으로써 다국어 문자 처리가 강력하게 요구되었다. 기존의 ISO 646 을 바탕으로 한 ISO 2022 확장 규격은 이들 문자를 표현하고 처리하는 데 한계를 가지게 되었고, 그래서 미국의 유니코드(Unicode)와 ISO 10646 다국어 문자판(BMP: Basic Multilingual Plan)에서는 두 바이트 코드로 전세계의 문자를 하나의 공간에 함께 수용하는 방식으로 발전하여 왔다.

1992 년 ISO/JTC1/SC2 서울회의에서 ISO 10646 의 다국어판에 배치할 한글 코드(부호계)가 제안되었다. 1992 년 3 월에 한국 JTC1/SC2 의 WG2 위원들이 모여서 한국안을 작성하기 위한 회의를 여러 번에 걸쳐서 논의하였다. 회의의 결론은 정음형 56 자(변정용 제안)를 채택하였으나 코드화 대상의 범위를 훈민정음 창제 후 사용한 모든 자소를 찾아서 이들을 코드화 한다는 다수결 의결을 하였다. 그래서 그 후 국립 국어 연구원에서 240 자를 선정하였고, 여기서 완성형은 빼기로 결론을 내렸다. 이 안은 그 후 JTC1/SC2 를 통과하면서 완성형은 그대로 살아 남았는데 기존의 틀을 깨지 않은 범위에서 이 문제를 추진하는 것으로 하여 결국 기존의 KS C 5601-1974 자모형(51 자), 현행 국내 표준인 KS C 5601-1987 완성형 2350 자, 그리고 초성자, 중성자, 종성자를 코드화한 자소형 240 자를 추가하였다. 이때 완성형은 2350 자에 포함되지 못한 차순위의 빈도를 가진 1930 자와 나머지 무순위 2376 자를 배치하여 결국 11172 자 가운데 6656 자인 약 60%를 배치하였다.

완성형의 불완전성을 대하여 노력 끝에 1996 년 봄에 국내 JTC1/SC2 는 11172 자를 BMP 에 배정이 확정되었으며, 기존에 배치된 6656 자를 없애고 현대 한글

완성형 음절 모두를 가나순으로 배정 받게 되었다. 이렇게 빨리 국내 표준으로 수용하게 된 경위는 95 년 후반기 마이크로소프트사가 한글 WIN95 에서 가나다순과 관계없이 11172 자를 편법으로 표현한 소위 '통합형' 한글 코드 때문에 코드 논쟁이 재연되었다. 이것은 현재 2350 자 완성형 한글과 한자 4888 자가 배정되어 있는 0xB0A1 에서 0xFEFE 사이를 제외한 나머지 부분에 11172 자 가운데 2350 자에 포함되지 못한 음절자를 배정하였기 때문에 가나순 정렬이 불가능하게 되었다. 이 때 정부는 UCS 코드를 국내 표준으로 수용하여 KS C 5700-1995 로 발표하자 이내 코드 논쟁은 잠잠해졌다. 그런데 사실 KS C 5700-1995 가 실제로 그 문제를 해결한 것은 전혀 없었다. 우리는 한글 코드 논쟁의 본질 문제 해결에 제대로 접근하지 못하고 있기 때문에 계속 논쟁의 불씨를 남겨두고 있다. 가나다순 배열 문제의 해결로 일단락될 사안이 아니다. 더구나 앞으로 언어처리 응용이 중심이 된다는 사실을 감안하면 그 조치는 미봉에 불과하다. 세 가지 종류의 한글 코드가 부호화 방식이 다르고 자모형이나 자소형은 완성형이 표현할 수 있는 한글의 음절을 표현할 수 있다. 그런데 이렇게 코드를 이중화하고 중복시키는 목적은 분명하지 않다.

2 절. 한글 표현 방식 분석

1. 부호화 대상 관점

한글 문자의 구조 원리를 규정한 훈민정음 해례를 보면 한 소리마디가 초성, 중성, 종성으로 이루어져 있으며 기본 소리에서 초성과 중성 28 자를 정의하고 종성은 초성과 같다고 정의하였다. 그리고 합자해에서 겹소리를 구성하는 방법과 소리 글자를 구성하는 규칙을 규정하였다. 이것은 기존 연구에 따르면 그 조합 가능한 음절의 수가 무려 약 399 억 음절에 이른다고 한다. 이것은 다시 말해서 문자의 생성 원리를 규정한 것이다. 이러한 특성을 가진 한글 문자를 최근 우리가 인식한 것은 1930 년 조선어학회를 근간으로 한 현대 한글 맞춤법에 근거하여 생각을 고정시켜 두고 있다. 1974 년 처음 정보교환용 한글 부호계를 제정할 당시엔 그래도 한글의 특성에 맞게 자음과 모음을 코드화 대상으로 삼았다. 이 때는 컴퓨터의 특성과 한글의 특성이 모두 고려된 제정으로 이해가 된다. 하지만 그 다음 1982 년에는 초성과 중성 그리고 종성자를 코드화 하는 방안을 채택하여 한글의 특성은 더욱 잘 고려된 반면에 컴퓨터의 문자 코드 제정 규격을 따르지 않음으로써 국제화에 어려움을 가지게 되었다. 그래서 1987 년에는 음절을 코드

화 대상으로 삼는 관점으로 바뀌어 현재 까지 한글 코드는 음절을 그 코드화 대상으로 삼아 왔다. 그래서 문자를 단순히 표현하는 분야에선 한자나 가나와 같은 수준에서 편하게 소프트웨어를 개발하게 되었지만 언어 처리를 하는 응용 분야에선 초성, 중성 종성 또는 자음과 모음에 관한 음절자에 대한 속성을 얻을 수 없었기 때문에 자모형이나 조합형 코드를 사용하거나 기존에 존재하지 않는 새로운 코드 체계를 사용하게 되었다.

한글 문자는 훈민정음 해례에서 정의한 생성 원리에 따라서 로마자나 가나와는 다른 글자 구조를 보이기 때문에 코드화 대상을 무엇으로 해야 하는 지에 대한 다양한 시각을 가져 왔다. 자음과 모음을 또는 초성,중성,종성으로 구성된 음절을, 또는 단순히 음절을 그 대상으로 삼아 왔다. 훈민정음의 특성상 두 번째 관점이 가장 적합하지만 컴퓨터의 코드 규격과 상충된다. 최근의 훈민정음의 과학성의 연구 성과에서 코드화 대상을 자연스럽게 낱자로된 초성자, 중성자, 종성자로 보자는 관점의 제안이 있었고, 이를 기반으로 '정음형'이라는 코드가 제안되었다. 한글에서 음절자란 가독성(readability)과 변별력을 위한 수단이라고 할 때 컴퓨터 내부에선 풀어쓰기 형태로 존재하고 화면과 인쇄기에 출력될 때는 초성자, 중성자, 종성자에 내포된 음절자 정보를 가지고 음절자로 표현하면 된다는 것이다.

2. 국내 표준화 과정의 문제

표준화는 다양한 규격을 하나의 규격으로 단일화 함으로써 효과적으로 이룩할 수 있다. 컴퓨터가 도입된 이래로 한글 처리는 하나의 숙제였으며 컴퓨터 회사를 비롯한 국내의 여러 기관이 이에 대한 방안을 추구하여 왔으며, 이러한 과정에서 만들어진 여러 규격에 대한 표준안으로 1974 년 처음 KS C 5601-1974 를 제정하였다.

그 후 1982 년에는 조합형, 그리고 1987 년에는 2350 자 완성형으로 각각 개정한 바 있다. 1992 년에는 국제 규격에 맞지 않는 조합형 코드가 복수 표준이 되었다. 1995 년에는 마이크로 소프사의 통합형 한글 코드 문제가 있었고, 이에 대한 보완책으로 UCS 를 수용하여 KS C 5700-1995 라 하였다. 여기에는 KS C 5601-1974 자모형과 240 자 자소형, 그리고 11172 자 완성형이 들어 있다. 이것은 앞에서 말한 대로 표준화의 근본 취지에 어긋난다.

왜냐면 코드를 단일화 하기 보다는 복수 코드 체계를 사용하도록 권장하고 있기 때문이다. 그래서 음절자 표현 방법이 중복되어 있다. 예를 들면 현대 한글에서 조합 가능한 11172 자는 자모형으로나 자소형으로 조합이 가능하며, 자모형은 자소형에 포함되어 있다. 결국 240 자 자소만으로도 충분하게 표현할 수 있는 음절자를 구태여 여러 가지 방법으로 표현할 수 있도록 한 것은 표준화가 아니라 혼란을 일으킬 것이다. 11172 자 완성형을 주로 사용할 것을 권하고 있는데 그것은 언어 처리 응용에선 무용지물이기 때문에 포함시킬 필요가 없으며, 설사 240 자 자소형이나 자모형도 복자모에 대하여 코드를 부여하였기 때문에 언어 정보를 상당 부분 상실하고 있다. 또한 옛한글 처리가 가능하긴 하지만 복자모를 군을 완전히 찾아 내지 못하였기 때문에 현재 완전한 지원을 할 수 있다고 보기 어렵다.

3 절. 한글 음절자의 표현과 처리에서 문제점

1. 음절자의 복수 부호계 표현

KS C 5700-1995 는 현대 한글에서 표현 가능한 11172 자를 기본 다국어판에 반영되어 있다. 그래서 2350 자 완성형에서 표현할 수 없던 음절자의 문제는 해결이 되었다. 그런데 단자음 또는 모음자를 표현하고자 할 때는 자모형이나 자소형과 코드 체계와 혼합해서 쓰지 않으면 안된다. 예를 들어서 "ㄱ은 어금닛 소리니" 라고 할 때 'ㄱ'은 완성형 코드 그룹에는 없다. 그래서 'ㄱ'을 표현하려면 어쩔 수 없이 자모형이나 자소형에서 빌어 써야 하는데 자모형에서 가져다 쓰야 한다. 그리고 음절을 표현할 때 화면에서는 글자꼴이 같아 보이더라도 대응하는 코드는 여러 가지가 있을 수 있다. 예를 들어서 음절자 '한'을 UCS 에 있는 세 가지의 코드계의 16 진수 코드 값으로 표현하면 다음과 같다.

- 11172 자 완성형(1 바이트): D55C

- 240 자 자소형(3 바이트): 1122 1161 11AB

- 51 자 자모형(3 바이트): 314E 314F 3134

이들 코드에 대하여 출력 자동틀(Automata)에서 모두 '한'이라는 글자꼴을 화면에 표현해 주어야 한다. 그런데 여기서 복잡한 문제는 비교와 패턴 매치의 경우

에 이들이 서로 같은 글자임을 인식하고 그렇게 처리되도록 해 주어야 한다.

2. 한 음절자에 대한 혼합 코드 표현 가능

문제를 보다 확대해서 보면 한 음절자를 표현함에 있어서 코드계를 혼합해서 사용하는 것도 가능하다. 자모형과 자소형, 자모형과 완성형 또는 자소형과 완성형 등을 혼합해서 한 음절자를 조합할 수 있다. 복자음 또는 복모음을 사용하는 경우에는 더욱 더 복잡해 질 수 있다. 이러한 경우를 모두 따져 볼 때 KS C 5700-1995 에서 세 가지 코드를 표현하는 것은 한 편으로 모든 가능성을 부여해 놓은 것 같지만 잘 못 사용하였을 때 그 혼란은 매우 심각해 진다. 예를 들어서 음절 '한'의 경우를 보자.

◆ 완성형-자모형 :

- 완성형(하),자모형 ㄴ : D558 11AB

◆ 완성형-자소형 :

- 완성형(하),자소형 ㄴ : D558 3134

◆ 자모형-자소형 :

- 자소형 ㅎ,자모형 ㅏ,자모형 ㄴ:1122 314F 3134
- 자소형 ㅎ,자소형 ㅏ,자모형 ㄴ:1122 1161 3134
- 자소형 ㅎ,자모형 ㅏ,자소형 ㄴ:1122 314F 11AB
- 자모형 ㅎ,자모형 ㅏ,자소형 ㄴ:314E 314F 11AB
- 자모형 ㅎ,자소형 ㅏ,자소형 ㄴ:314F 1161 11AB
- 자모형 ㅎ,자소형 ㅏ,자모형 ㄴ:314E 1161 3134

이처럼 혼합해서 음절자를 구성할 때 8 종류의 코드가 만들어 지며 이것은 복수 음절자 코드 3 종류를 합하면 무려 11 종류의 코드가 만들어 질 수 있다. 물론 이것은 안내서를 통해서 이러한 혼란을 막을 수 있을 것이다.

3. 음절자 표현 관련 부가 규정 없음

KS C 5700-1995 에는 부록 A 에서 N 까지 14 개의 부록이 있다. 국내 표준이기 때문에 마땅히 한글로 번역되어 있어야 함에도 불구하고 영문으로 그대로 있다. 현재 부록에서 음절자 조합에 관한 규정은 찾아 볼 수 없다. 단지 부록 B (규정)에서 정의하고 있는 조합 문자 목록에 한글이 포함되어 있지 않는 점으로 미루어 보아 한글은 조합하지 않는 것으로 추정할 수 있다. 그럴 지라도 단자모 가운데 어느 것을 쓸 지에 대한 규정을 두어야 할 것이다. 또한 조합하지 않는다면 옛한글은 전혀 조합할 수가 없게 된다. 다국어 문자판(BMP)에서 그렇게 많은 공간을 차지하고서도 옛한글을 표현할 수 없음은 문제가 아닐 수 없다. 앞에서 열거한 문제들을 없애려면 완성형이 가진 음절자만 사용하고 나머지 자모형 또는 자소형은 낱자만을 표기하도록 해야 할 것이다. 하지만 옛한글의 요구를 수용하려면 어쩔 수 없이 자모형이나 자소형으로 조합을 해야 한다. 그러면 위와 같은 조합의 가능성을 배제할 수 없다.

4. 훈민정음 원리에 따른 옛글 조합

옛글 음절자를 완벽하게 조합하려면 훈민정음 창제원리에 따르면 가능하다. 하지만 이 때는 3.2 에서 제기한 것처럼 초성자, 중성자, 종성자가 두자 또는 석자씩 조합할 수 있기 때문에 혼합 코드계 조합을 하게 되어 문제는 더욱 심각해 질 수 있다. 예를 들면 초성자의 석자 합용병서에서 자모형, 자소형, 완성형의 조합이 위의 경우처럼 일어 날 수 있기 때문이다. 그리고 또 한 가지 문제는 옛한글 사전을 만든다고 할 때 가나다순으로 정렬을 할 수가 없다. 왜냐면 현재 옛글의 순서가 현대 한글 다음에 오는데 옛한글은 현대 한글을 포함하고 있기 때문에 그렇다.

5. 언어 처리

이미 여러 논문에서 논의되어 왔듯이 완성형은 자소 정보가 없기 때문에 언어 현상을 처리하거나 분석하는 언어 처리 분야 응용에서는 사용할 수가 없다. 그것은 음소 즉 자소를 부호화 대상으로 삼지 않았기 때문이다. 따라서 부득이 같은 UCS 테이블내에 있는 자모형 또는 자소형을 쓰야 한다. 언어 처리에서는 형태

소 분석과 같은 경우 언어 정보가 글자의 원소 단위에 숨어 있기 때문에 코드 부여 대상이 낱자소가 되어야 한다. 하지만 자모형과 자소형은 모두 복자소를 코드화 대상으로 하고 있기 때문에 자소형의 경우 초성자, 중성자, 종성자의 합이 240자로 클 뿐만 아니라 낱자소 정보를 알아 내기 위하여 부가적인 정보를 코드 밖에서 가지고 있어야 한다. 이것은 매우 불편하며, 훈민정음의 기본 원리에도 어긋난다.

훈민정음의 원리는 낱자소에 대하여 코드로 부여하고 나머지는 규칙으로 처리하도록 한 것이다. 결국 언어처리를 하려면 240자 자소형의 사용이 불가피하기 때문에 음절자 조합은 불가피하다.

4절. 개선 방향

1. 정음형 코드로 개선

훈민정음 해례에서 정의된 문자 구성 원리는 매우 과학적이며 독특하다. 즉 기본적인 문자만 정의하고 나머지 음절자 구성은 모두 규칙으로 처리한 것이 바로 그것이다. 현재 컴퓨터 사용의 확산은 보다 다양한 한글 또는 국어 정보처리를 요구하고 있다. 그간 코드 제정자들은 코드 적용 범위를 파악하지 못하고 단편적으로 그 때 그 때의 요구를 만족할 수 있는 코드를 제정하다 보니 요구가 있을 때마다 임시 변통적인 개정을 거듭해 왔다. 이렇게 시행착오를 거치면서 여러 가지 코드를 제정해 왔고, 그들 코드로 된 자료들을 가지고 있다. 이제 훈민정음 원리가 밝혀지고 이를 공학화하는 연구 결과에 따라서 한글 및 국어 정보처리 전반의 요구를 만족시킬 수 있는 정음형-1996의 사용을 권장한다. 정음형은 ISO 646과 ISO 2022 규격에 매우 적합하며 또한 ISO 10646 BMP에서 조합하는 문자군으로 등록될 수 있다. 정음형을 채택하였을 때 결과는 다음과 같다.

첫째, 훈민정음 원리에 따라서 음절자 구성을 하기 때문에 간단하지만 무려 399억 음절자를 구성한다고 한다. 그래서 표현하고자 하는 문자는 글자꼴이 지원되는 한 모두 표현할 수 있다.

둘째, 또한 기존의 모든 한글 코드의 전체집합이기 때문에 번역시 호환성이 완벽하게 보장된다.

셋째, 정음형은 낱자소 조합으로 음절을 구성하기 때문에 현대 한글이나 옛한글에서 복자소의 완전성을 항상 보장한다.

넷째, 문자 집합이 작기 때문에 국내외 규격을 만족한다.

다섯째, 낱자소에 대한 코드를 가지고 있기 때문에 언어 처리를 하는 한글 및 국어정보처리 전반에 충분한 언어정보를 전달한다.

2. 남북의 가나다순 배열

북한의 국규 9566-93 정보교환용 부호는 한글 음절자 2420 자를 규정하고 있다. 이것은 1 수준에 해당하며 KS C 5601-1987 의 2350 자 보다 70 자 더 많다. 여기서 차이가 많이 나는 것은 배열 순서이다. 북한은 복자음과 모음을 단자모 다음에 배치하는 관계로 이러한 차이가 나는 것이다. 만약 북한 인터넷에 가입한다면 전자우편 서신을 교환하는데는 이 양자를 번역하는 프로그램으로 해결할 수 있다. '96 코리언 컴퓨터 처리 학술대회에서 남북 공동의 자모순을 합의하였다. 그결과는 남북의 어느 안도 아닌 상호 일부를 양보한 내용이다. 여기서 문제의 해결은 정음형을 사용하였을 때 많은 부분이 해결될 수 있다. 주로 문제가 되는 것은 복자모의 위치에 관한 것이다. 정음형은 복자모에 대한 코드를 부여하지 않기 때문에 자형순을 따르는 남쪽안과 음가순을 따르는 북쪽안이 문제가 되지 않는다. 단지 걸리는 문제 하나는 바로 초성자에 포함된 'ㅇ'자이다. 북한은 이것은 음가가 없다하여 초성 ㅎ 다음에 배치하지만 남한은 자형순이기 때문에 ㅅ 다음에 배치하고 있다. 정음형을 가지고 표현을 했다고 할 때 각측에 맞도록 정렬해 주는 정렬 프로그램을 작성하면 된다. 이것에 대한 한 가지 예가 로마자이다. 로마자는 소문자와 대문자가 있으며 코드 배열순으로 할 경우 소문자 a는 영문 Z가 오고 난 다음에 나타난다. 그렇기 때문에 로마자 사전순 정렬을 프로그램에서 이것을 조정하고 있다. 여기서 ISO 10646 에는 남한의 안이 반영되었다. 그래서 통일 이전에 남북이 정보교환을 하려면 남한의 자모순 배열을 따라야 한다. 이것을 북한의 입장에서 받아 들이기 어려운 문제가 될 것이다. KS C 5700-1995 는 남북한 정보교환에서 공동으로 사용할 수 없는 문제를 가지고 있다. 1996 년 중국 연변에서 있은 합의문에 나타나는 자모 순서는 남북의 어느 안에도 맞지 않는다. 그래서 공동안이 될 수 있었다. 현재 상황에서 남북간 정보교환은 불가능할 것이다. 물론 북한에서 남한의 완성형을 북한 국규 9566-93 으로 바꾸는 변환 프로그램을

준비하면 될 것이지만 하여간 이것은 북한이 받아 들이지 않을 것이다.

3. KS C 5700-1995 에서 적용 방안

UCS 코드를 사용함에는 몇 가지 규정을 가져야 앞으로 혼란을 막을 수 있다. 우선은 두 가지 범주를 고려해 볼 수 있다. 하나는 현대 한글만을 취급하며 언어 처리와 무관한 응용이고 다른 하나는 언어 처리와 옛한글 표현을 포함하는 분야이다. 먼저 전자를 고려해 보면 다음과 같다.

- 현대 한글 음절은 반드시 AC00-D7A3 사이의 코드로만 표현한다.
- 낱글자를 표현하고자 할 때는 1100-11F9 까지의 글자만을 쓴다.

여기서 문제는 언어 처리 응용과 옛 한글을 표현하는 응용들의 경우이다. 이 때는 어쩔 수 없이 조합을 해야 하며 다음과 같이 규정한다.

- 완성형 11172 자 AC00-D7A3 사이의 코드는 절대로 쓰지 않는다.
- 모든 글자는 조합하며 240 자 자소 1100-11F9 사이 코드만을 사용한다.
- 이 때 한 음절을 나타내는 바이트의 수는 4 또는 6 바이트이다.

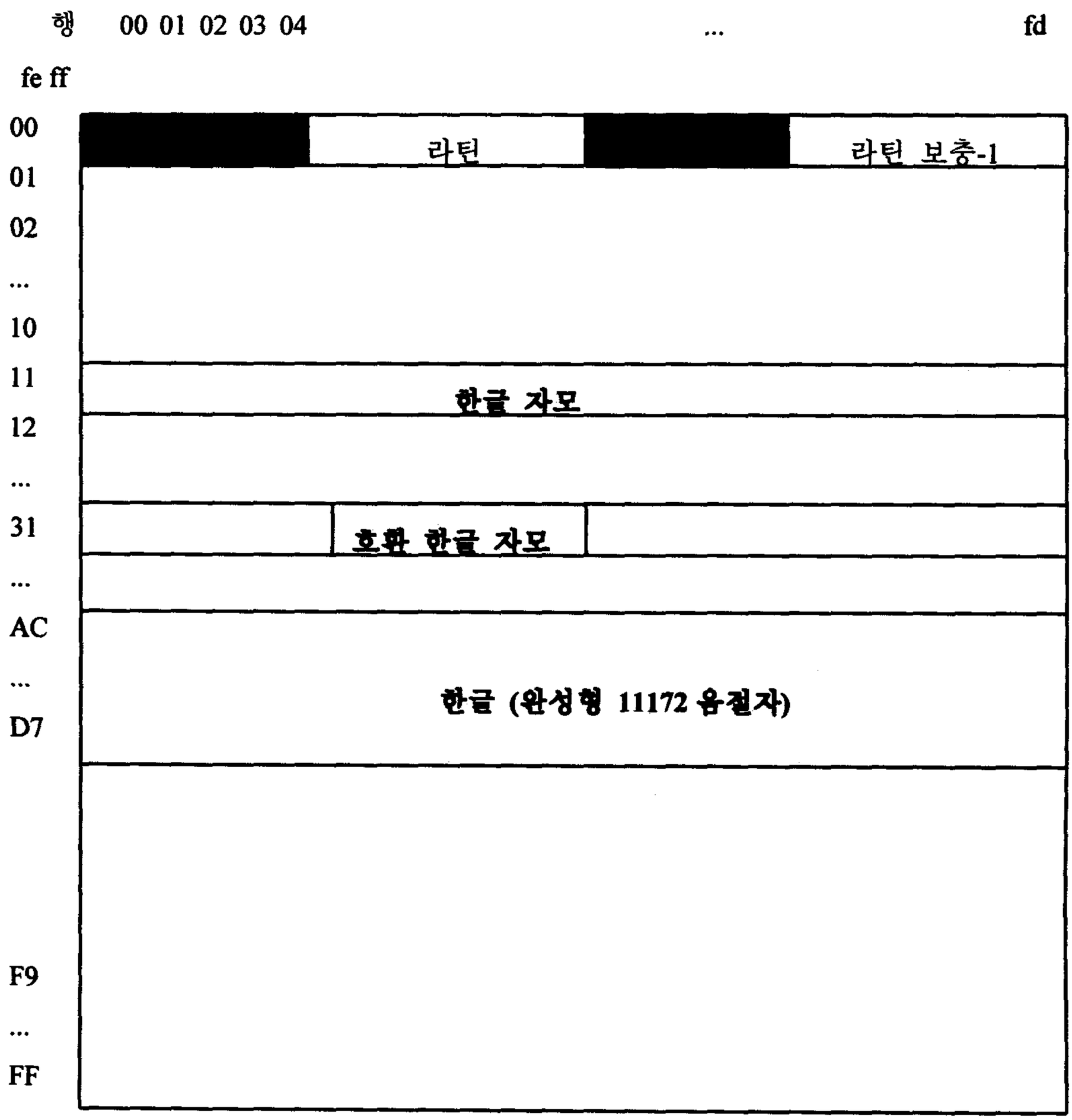


그림 4.1 ISO 10646 BMP

이상의 규격은 복자소에 대한 코드를 사용하였기 때문에 최대 6 바이트로 제한 할 수 있다. 여기서 문제는 240 자에 포함되지 않은 복자소의 구성이 새로이 나타난다면 6 바이트는 초과될 수 있으며 이 때 정렬이 매우 어려워 진다는 점이다. 그리고 언어 처리에서 복자소의 구성 성분에 대한 정보를 요구한다면 이에 대한 별도의 정보를 유지하고 있어야 한다는 점이다.

4. 훈민정음 원리에 맞는 한글 코드 표준화

한글 코드 논쟁의 본질이 코드화 대상에 있기 때문에 올바른 코드화 대상을 찾아냄으로써 해결될 수 있음을 보았다. 또한 국내 표준의 제정에서 최근에는 국내 표준에 우선하여 국제 표준을 정하고 국내 표준은 이를 따라서 채택하고 있음이 문제점으로 지적되었다. 앞으로 국내 표준으로 사용할 KS C 5700-1995 가 가진 문제점 가운데 가장 크게 지적되고 있는 것이 표준화를 위하여 기존의 표준화된 코드를 모두 표준안에 포함시킨 점이다. 물론 그 명분은 완성형이 가지고 있는 문제점을 보완한다는 것이긴 하지만 정당성이 명확하지 않다. 왜냐면 세 가지 종류 코드가 사용할 가능성이 있으며 최악의 경우에는 그들을 혼합해서 사용할 수도 있기 때문이다. 그럴 경우 같은 음절에 대한 표현이 너무 많아서 표현된 글자꼴은 같다고 하더라도 실제 코드는 다름으로 패턴 매치 등에서 일치하지 않게 된다. 또한 완성형을 중심으로 한 코드이기 때문에 언어 처리에 부적합하며, 국제 문자 표준 표에서 두 번째로 많은 영역을 차지하는 결과를 갖게 되어 과학적인 문자라고 자부하여 온 국민적 정서에 상치되는 결과를 초래하고 있다. 따라서 세 가지 종류의 코드를 모두 삭제하고 이를 훈민정음의 원리를 자연스럽게 표현한 '정음형' 코드만을 표준안으로 삼아서 컴퓨터 상에서도 마치 '연필로 글을 쓰듯이 ...'를 이룩할 수 있도록 지원해야 한다.

5 장. 윈도 95 에 정음형 코드 적용

1 절. 윈도 95 의 콘트롤

1. 윈도 95

마이크로 소프트 윈도 95 는 WIN 32 API 의 기능을 대부분 지원하는 윈도 3.1 의 개정판으로 새로운 사용자 인터페이스를 제공한다. 윈도는 기본적인 GUI 에서 보다 객체 지향적인 인터페이스로 변모해 왔다. 이제 응용 프로그램의 역할은 인터페이스를 이용하여 객체를 다루는 도구로 바뀌고 있다. 대부분의 WIN32 API 가 구현되었음은 실행 파일이 윈도 NT 와 윈도 95 에서 실행될 수 있음을 보장한다. 윈도 95 는 32 비트 운영체제이지만 아직 하위 호환성을 위해 16 비트로 남겨진 부분과 메모리 부분과 약간의 부분은 아직 예외이다. 추가 사항은 선점형 멀티태스킹과 다중 쓰레드가 응용 프로그램의 병렬 수행 기능 등이다.

윈도 95는 완성형 한글 코드 KS C 5601-1987을 지원한다. 그리고 선택적으로 유니코드 또는 ISO 10646을 사용할 수 있도록 되어 있다. 정음형 코드를 사용하려면 비쥬얼 C++에서 기본적으로 제공하는 TCHAR 타입을 사용해야 한다. 그런데 프로그램에서 _UNICODE를 정의하면 이것은 wchar_t 즉 16 비트 문자 타입을 지원한다. Char 타입은 8비트 타입을 지원한다. _UNICODE를 사용하지 않으면 멀티 바이트 문자 세트(MBCS)로 기동된다. 윈도 95는 유니코드를 지원하지 않는다. 정음형은 Char 타입이나 CString 클래스를 사용할 수 있다.

참고로 앞으로 국제 문자 표준이 될 유니코드는 윈도 NT 내부에서 기본적으로 구현이 되어 있다. 그러나 입력에서는 ASCII를 받아서 유니코드로 번역하고 이를 출력할 때는 다시 유니코드를 ASCII로 번역하여 뷰로 출력한다. 본 연구의 테스트 베드인 윈도 95에서는 기본이 ASCII 이다. 그래서 윈도 NT와는 반대로 유니코드를 외부적으로 처리하게 해줄 수 있다. 유니코드를 받아 들여서 이를 ASCII 문자열로 변환하여 내부에 맞추고 다시 이를 출력할 때는 ASCII를 유니코드로 변환하여 출력하고 있다.

정음형 코든 이러한 경우와 마찬가지로 윈도 95와 윈도 NT에서 영문 모드를 기본으로 하여 이를 상호 번역하는 기능을 통하여 해결한다. 이 때에도 완성형 코드를 만드는 IME가 계속 작동하기 때문에 이를 사용하지 않도록 해야 한다.

2. 윈도 95 응용 프로그램 조건

마이크로 소프트는 윈도 95 의 진정한 응용 프로그램으로 인정 받기 위한 다음 조건을 제시하였다. 본 연구를 수행하면서 편집기의 호환성을 위하여 다음 사항을 유의한다.

첫째, WIN32에서 실행 가능해야 한다.

둘째, 반드시 UI/Shell을 지원해야 한다.

셋째, 반드시 윈도 NT 3.5에서 검증되어야 한다.

네째, 긴 파일명을 지원하고 모든 문서명 표현에 사용해야 한다.

다섯째, 플러그 앤드 플레이 인식은 권장 사항이지만 필수 요구 사항은 아니다.

3. 콘트롤

콘트롤 (Control)이란 사용자와 상호 동작하거나 응용 프로그램에서 입력을 전달하는 수단으로 사용되는 그래픽 객체라 할 수 있다. 일반적인 다이얼 로그 박스는 직접 정보를 표현하는 대신 콘트롤이란 사용자 인터페이스 오브젝트를 통하여 사용자에게 정보를 보여주고 입력받는다. 콘트롤에게 다이얼 로그 박스는 상위 윈도우이다. 여러 가지 형태의 콘트롤은 고유의 외양과 기능을 갖는다. 비쥬얼 C++ 4.0 에서 사용할 수 있는 콘트롤은 '표준 콘트롤(Standard Control)'과 '커스텀 콘트롤(Custom control)' 등 두 종류를 지원한다.

편리하고 다양한 입력 또는 출력 사용자 인터페이스를 위하여 다이얼로그 박스를 제공하는데 이를 위한 리소스 생성, 편집 기능을 제공한다. 표준 콘트롤은 윈도우 운영체제가 직접 제공한다. 콘트롤이 자료의 입출력에 사용된다는 점에서 만약 편집기에서 이러한 콘트롤을 사용할 경우 예를 들어서 찾기에서 찾기 패턴은 정음형 문자열이어야 한다. 그러면 키보드를 통하여 입력할 때 ANSI 문자열을 사용하고 여기서 정음형 모드가 사실일 때 영문자 코드를 정음형 코드로 변환한다. 이들은 CString 과 같은 문자열 타입에 저장되고 정음형 텍스트에 대하여 패턴 매치를 한다. 따라서 정음형 코드를 입력 받거나 콘트롤을 통하여 출력하려면 정음형 접수기와 변환기가 필요하다.

정음형 코드를 텍스트 박스에 출력하려면 디스플레이 콘텍스트 오브젝트의 멤버 함수인 TextOut()을 이용한 정음형 코드 디스플레이 함수를 개발하여 추가해서 해결해야 한다. 표준 콘트롤 가운데 입력되는 문자열이 정음형 문서와 함께 처리되어야 한다면 이들은 반드시 정음형 코드로 입력되어야 한다. 이처럼 정음형 코드와 밀접한 관계가 있는 콘트롤은 다음과 같다.

- 정적 텍스트 콘트롤(CStatic) : 사용자가 바꿀 수 없는 텍스트 표시
- 편집 복스 콘트롤(CEdit) : 사용자가 텍스트를 입력하거나 표시하는 영역을 제공한다.
- 콤보 복스 콘트롤(CComboBox) : 텍스트 복스 콘트롤과 리스트 복스 콘트롤이 결합된 형태로, 텍스트 복스 콘트롤에서 데이터 직접 입력하거나 리스트 복스에서 하나의 항목을 선택할 수 있게 한다.

4. 디바이스 콘텍스트(Device Context)

윈도 어플리케이션은 다양한 종류의 출력 장치에 정보를 출력하며 장치 독립적이다. 이들은 GDI32.DLL 과 디바이스 드라이브에 의하여 이러한 기능이 이루어 진다. 어플리케이션은 GDI 에게 특정한 디바이스 드라이버를 로드할 것을 요청해야 하며, 일단 디바이스 드라이버가 로드되면 출력에 필요한 정보를 해당 디바이스 드라이브에게 제공해야 한다. 이러한 작업은 디바이스 콘텍스를 생성하고, 관리함으로써 수행된다.

MFC 라이브러리는 디바이스에 출력하기 위한 디바이스 콘텍스트에 관련된 기초 클래스로 CDC 라는 클래스를 제공한다. CDC 클래스의 m_hDC 데이터 멤버에는 디바이스 콘텍스 핸들이 저장된다. CDC 파생 클래스에서 CPaintDC 클래스는 여러 윈도가 겹쳤다가 회복되었을 때 해당 영역을 다시 그려야 하는 무효화 영역에 관한 사각형 정보를 WM-PAINT 메시지에 함께 넣어 준다. BeginPaint API 를 통하여 디바이스 콘텍스를 구하고 EndPaint API 함수를 사용하여 디바이스 콘텍스를 윈도 운영체제에 되돌려 준다.

이러한 DC 에서 TEXTMETRIC 구조체로부터 글꼴 정보를 얻을 수 있다. 글자의 폭과 높이 등에 관한 정보를 이것으로 부터 얻어서 글자를 DC 에 출력하거나 커서를 이동할 때 그 움직이는 단위를 결정한다. 예를 들어서 한글은 글자꼴의 높이와 폭이 32 x 32 인 반면에 영문자는 32 x 16 이다. 영문자는 16 픽셀 단위로, 한글은 32 픽셀 단위로 커서 이동을 하는데 유용한 정보를 제공한다. 그 외에도 글자색이나 배경색 선택이나 배경색의 투명 여부, 텍스트가 출력될 때 정열 방식 등에 관하여 TEXTMETRIC 그조체의 멤버 함수가 결정한다. 그리고 각종 좌표의 원점 위치 변경이나 좌표 계 값의 증감 방향 등을 정의한다.

2 절. MFC 문서-뷰 아키텍쳐

모든 MFC 어플리케이션을 시작할 때 CWinApp 파생 오브젝트를 생성한다. 파생 오브젝트의 역할은 여러 가지 문서 템플릿 오브젝트를 생성하고 등록하는 일이다. 문서 템츨릿은 문서(Document), 뷰(view), 프레임 윈도우(Frame Window), 메뉴, 아이콘, 비트맵 등 디폴트 리소스를 생성하고 이들을 유기적으로 결합시키는 일을 담당한다.

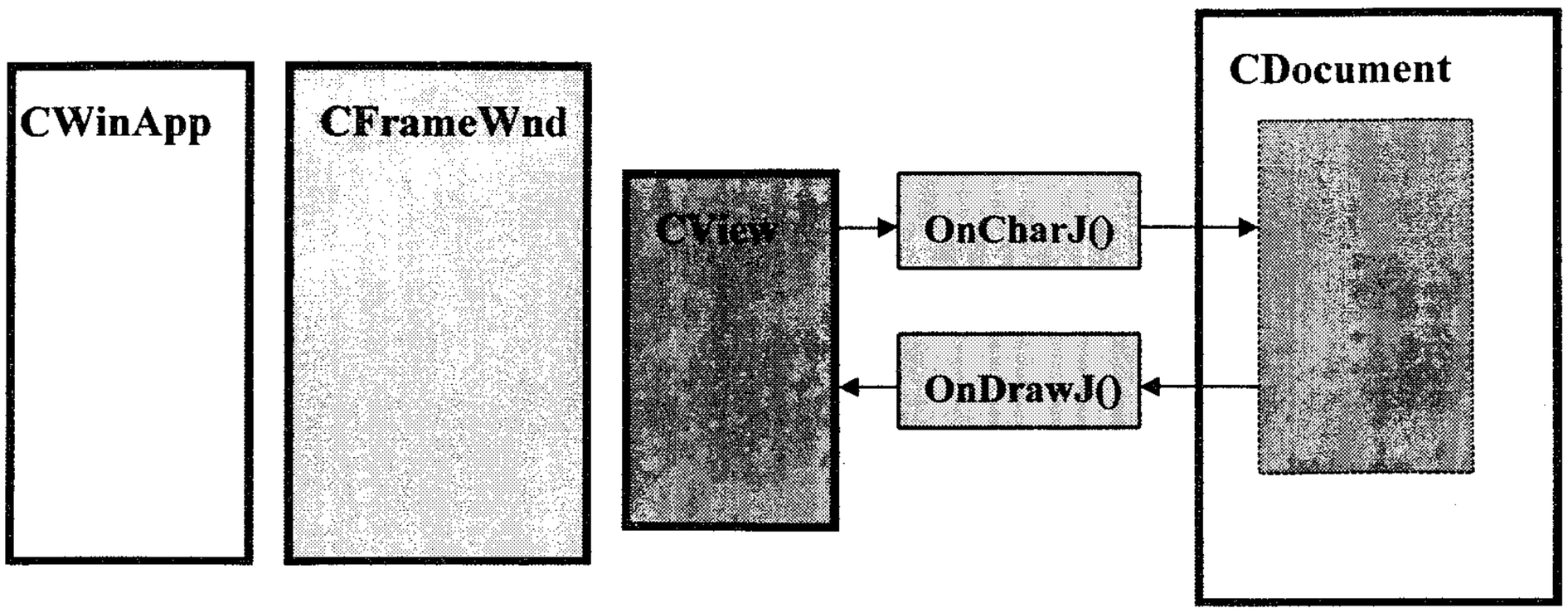


그림 5.1 어플리케이션 프레임워크 개요

어플리케이션이 하나의 문서와 여러 개의 뷰로 구성될 때 각 뷰 클래스마다 하나씩 여러 개의 문서 템프릿 오브젝트가 생성되어야 한다. 어플리케이션이 여러 개의 문서 클래스와 여러 개의 뷰 클래스로 구성될 때는 각 문서/뷰 쌍마다 하나의 문서 템프릿 오브젝트가 생성된다

3절. 정음형 적용 방안

MFC의 문서-뷰 아키텍쳐에서 정음형을 구현하는 것은 임시 방편적이다. 왜냐면 2350 완성형 음절자를 지원하는 표준 한글 코드가 윈도 95 운영체제 수준에서 지원되고 있기 때문이다. 키보드에 있는 한/영 키를 활용하여 한글을 입력하면 IME가 이를 완성형 코드로 변환한다. 그렇기 때문에 영문자 모드에서 정음형 한글 입력 모드를 별도로 만들고 문서-뷰 아키텍쳐의 구조를 활용하면 해결이 될 수 있다. 하나의 문서는 여러 개의 뷰를 가질 수 있기 때문에 현재의 ASCII와 KS C 5601-1987에 기반한 뷰에서 ASCII와 정음형 코드를 출력하는 뷰를 추가한다.

뷰 클래스는 어플리케이션의 물리적인 클라이언트 영역을 나타내며, 논리적으로는 문서 오브젝트에 포함된 정보의 뷰포트가 된다. 또한 뷰 오브젝트는 마우스나 키보드를 통하여 사용자로 부터 입력을 받아 들인다.

입력을 받아 들이는 OnChar()에서 정음형 오토마톤과 코드 변환기가 장착되고 편집계에서 CString 을 확장하여 정음형 문자열을 처리할 수 있는 멤버 함수를 추가한다. 출력에서는 뷰의 멤버 함수인 OnDraw() 함수에서 디스플레이 콘텍스의 멤버 함수인 TextOut()를 확장하여 정음형 코드를 완성형 또는 조합형 글꼴에 연결하게 한다.

프레임 윈도 클래스의 상태 바에는 정음형과 영문자 모드를 표시하고 정음형 한글 모드인 경우에는 다시 음소 모드와 음절 모드를 표시하여 사용자에게 편집기의 현재 모드를 안내한다. 정음형 코드 편집기에서는 완성형 코드 모드가 작동이 되지 않도록 처리해야 한다. 왜냐하면 윈도 95 는 기본적으로 IME 가 작동하여 한글 모드에서 한 음절이 인식되면 해당 음절의 완성형 코드를 반환하게 된다. 이것은 정음형 코드가 완성형과 같이 ISO 646 의 오른 쪽 도형 문자 지역인 GR 에 할당되어 있기 때문에 바이트의 8 번째 비트가 1 로 세트되어 있어서 충돌하며 둘을 구분할 수가 없다.

데이터는 문서 클래스에 저장되고 타입은 CString 인데 이것도 하나의 클래스로 정의되어 있어서 여러 가지로 편리한 멤버 함수를 지원한다. CString 에 대한 몇 가지 예를 들어 본다.

```
CString str = "1233";
str.empty();
```

이것은 str 포인트가 가리키는 문자열을 비우고자 할 경우에 사용한다. 즉 문자열의 첫 인덱스에 널(null)을 채우는 것과 같다.

```
Str[0] = NULL;
```

그리고 정음형 문자열에서 한글 표현을 위하여 필요한 몇 가지 중요한 문자열 처리 함수를 CString 멤버 함수를 사용하여 확장한다. CString 에서 문자열의 일부를 잘라 내려고 할 때 해당 문자열에 커서가 놓인 위치를 중심으로 다음 세가지 함수를 적용한다. Left(n)는 왼쪽부터 n 개 문자 추출하는 것이고, Right(n)는 문자열의 끝에서부터 왼쪽으로 n 개를 추출한다. 그리고 Mid(m,n)는 문자의 m 번 인

덱스부터 n번 인덱스까지 바이트를 추출한다. 이들 문자열 분할 함수는 정음형 코드 처리에서 음절 문자열을 초성자, 중성자, 종성자 문자열로 분할하는 데 그대로 적용할 수 있다. 이것을 적용하기 전에 정음형으로 된 한 음절자에 대한 문자열이 있을 때 자소 경계를 알아 내는 함수가 있어야 한다.

```
CString hanstr( "ㄱ 새 ㄹ" );
Cpoint   border = GetUmjeolBound( hanstr);
```

CPoint는 좌표 값이지만 여기서 두 개의 값을 함수로 리턴 받기 위하여 기존하는 구조체를 사용하였다. 여기서 border.x는 m이 되고, border.y는 n이 된다. 다음과 같이 초성자, 중성자, 종성자 추출 함수를 정의한다.

```
CString cho = hanstr.left(m);
CString jung = hanstr.mid(m,n);
CString jong = hanstr right(n);
```

또 하나 중요한 한글 처리 함수는 어떤 문자열의 길이에 관한 멤버 함수를 한글 음절 길이 함수로 확장하는 것이다. CString 클래스에서 영문자를 기반으로 해서 길이를 계산하는 멤버 함수가 있다.

```
CString strEng ("my daughter");
CString strHan("나의 딸");

m = strEng.GetLength();          // m = 11
n = strHan.GetLength();          // n = 4
```

여기서 이들 각각의 길이를 계산하라고 할 때 그 결과는 11과 4가 된다. 이것은 현재 윈도 95에서도 마찬가지다. 윈도 95에선 앞에서 언급한 대로 두 바이트를 기본 단위로 사용하고 있다. 영문자도 2바이트로 표현하고 있다는 말이다. 그래서 완성형 한글이 두 바이트로 표현되니까 영문과 한글이 길이 계산이

다를 바 없다. 이것은 유니코드가 되어도 마찬가지이다.

그런데 정음형에선 이렇게 자연스럽게 되지 않는다. 그것은 앞서 말한 대로 한 음절자를 구성하는 문자열의 길이가 일정하지 않기 때문이다. 그래서 정음형에 대한 GetLength() 멤버 함수를 새로 개발해야 한다. 물론 같은 이름으로 할 수도 있겠지만 대상 문자열이 기본적으로 다르기 때문에 이름을 달리 한다. 나머지 필요한 함수에 관하여 제 6 장에서 자세하게 다룬다.

```
n = GetLengthHan();
```

6 장. 편집기 "바른글"의 설계

1 절. 적용 분야

'바른글' 편집기는 정음형 코드를 채택하고 있기 때문에 특히 한글 코드에 민감한 언어 처리 분야에서 필요하다. 왜냐하면 언어 처리를 위한 언어 정보를 코드가 담고 있어야 하기 때문이다. 제 2 장에서 이미 언급한 바와 같이 현행 표준인 KS C 5601-1987 완성형은 음절을 대상으로 코드화를 하였기 때문에 코드만 가지고 각 음절자가 어떠한 자소로 구성되어 있는지 알 수 없다. 여기서 요구하는 바는 단순히 초성자, 중성자, 종성자의 정보만이 아니라, 각 자소가 어떠한 낱자소로 구성되었는지를 알아야 완전한 언어 처리를 할 수 있다. 한글에서도 각종의 언어 현상이 존재한다. 구개 음화, 자음 접변, 연음현상 등등이 그것이다. 이러한 요구는 반드시 완벽한 낱자소에 대하여 코드를 부여하여야 한다는 점이다.

현행의 표준은 2350 자 완성형과 조합형이며, 현재 대부분의 편집기나 워드프로세서는 이들을 이용하여 개발되었다. 완성형 코드는 주지하는 바와 같이 자소 정보를 표현하지 못하며 또한 2350 자는 무엇보다도 자소 정보를 가지고 있지 않다는 점이 결정적인 결함이다. 또한 표현할 수 있는 문자에 있어서도 현대 한글 11172 자의 1/4 에 국한될 뿐만 아니라 옛한글의 표현에 있어서 별다른 방법이 없다.

또 다른 표준인 조합형은 현대 한글 11172 음절을 표현할 수 있고 옛한글의

일부도 표현한다. 하지만 우선 문제가 되는 것은 ISO 646 이나 ISO 2022 그리고 최근의 ISO 10646 과 같은 국제 문자 코드 규격에 위반되기 때문에 정보교환에 사용할 수 없다는 점이다.

B16	초성(b15 - b10)	중성(b9 - b5)	종성(b4 - b0)

그림 6.1 조합형 코드의 비트 구조

```
unsigned int sylable;

        int chosung, jungsung, jongsung;

        jongsung = sylable & 0x1f;                  //최하위 5 비트 AND masking

jungsung = (sylable >>5) & 0x1f;

            //오른쪽으로 5 비트 쉬프트 후 5bit AND masking

        jongsung = (sylable >>10) & 0x1f;

            //오른쪽으로 10 비트 쉬프트 후 5bit AND masking
```

그림 6.2 조합형 코드의 비트 연산 프로그램

또한 일반적인 문자 처리 단위인 바이트 단위로 정보가 표현되어 있지 않아서 자소 정보를 인식하기 위하여 16 비트 가운데 상위 한 비트를 제외한 나머지 15 비트를 다섯 비트식 잘라내는 연산을 해야 하고, 잘라 낸 다섯 비트는 복자소에 대하여 코딩이 되어 있기 때문에 이를 다시 구분해야 하는 어려움이 있다. 단지 조합형에서 편리한 점이라면 비록 비트 연산을 통하여 이루어지는 복잡한 연산들이 있긴 하지만 한글의 한 음절이 16 비트로 된 일정한 단위에 표현할 수 있다는 점이다. 그러나 앞에서 언급한 국제 문자 코드 규격에 어긋나기 때문에 이 코드는 제한적으로 적용할 수 밖에 없다. 따라서 이러한 현행 코드를 채택하고 있는 편집기나 워드프로세서는 우선 국어 정보처리의 각 응용 가운데 특히 언어 처리에 관련된 분야에서 사용할 수가 없으며, 이것은 결국 국어 정보 처리 기술 발전

에 상당한 저해 요소로 작용한다.

편집기 '바른글'은 바로 이러한 문제에 대처할 수 있는 편집기이다. '바른글'은 훈민정음 창제 원리에 기반하여 만든 정음형 한글 코드를 채택하고 있기 때문에 언어 정보를 모두 표현할 수 있다. 이러한 언어 처리 분야 기술 개발에서 필요한 것은 자료의 입력과 프로그래밍을 작성할 수 있는 기본 편집기이다. 바른글은 이러한 요구를 만족시킬 목적으로 개발되었다. 바른글의 설치 환경은 PC 이며 32 비트 프로세서를 장착하고 있어야 한다.

2 절. 편집 명령어

1. 자소 모드와 음절자 모드

편집기가 갖추어야 할 기능 가운데 편집 명령어는 다양하다. 그렇지만 이것은 워드프로세서에 비하면 그 수가 매우 적다. 바른글 편집기는 일반적인 편집기의 기능을 기본적으로 가지고 있으며, 여기에 덧붙여 한글의 특성인 자소 및 음절자 문자에 대한 연산 기능을 가지고 있다. 다시 말하면 현재 대부분의 한글 편집기나 워드프로세서는 음절자 중심의 연산만을 가지고 있다. 부분적으로 음절 입력을 할 때 자소 연산이 부분적으로 적용되고 있다. 예를 들어서 '각'이라고 입력할 것을 현재 '간'이라고 입력하고 있으면 이 때 아직 한 음절자가 완전히 인식된 상태가 아니기 때문에 역공간(backspace) 키를 눌러서 'ㄴ'을 지우고 다시 'ㄱ'을 치면 '각'으로 입력할 수 있는 것 정도이다. 바른글에선 자소 연산 모드에선 커서 이동이나 편집기의 일반적인 연산인 삽입, 삭제, 변경 등이 모두 초성자, 중성자, 종성자를 단위로 이루어 진다. 뿐만 아니라 복자모에 있어서 연산 역시 완벽한 낱자 정보가 들어 있기 때문에 마찬 가지로 적용된다. 예를 들어서 'ㅙ'의 경우 커서 이동은 두 번에 걸쳐서 이동한다. 다시 말하면 마치 로마자 문자열을 지날 때처럼 낱자소에 대하여 연산이 이루어 진다. 자소를 입력하거나 출력할 때에도 마찬 가지이다. 예를 들어서 '학'을 '각'으로 변경하려면 '덮어쓰기' 모드에서 커서를 'ㅎ'에 놓고 'ㄱ'을 치면된다. 삽입, 삭제의 경우에는 좀 더 복잡한 연산이 연산 과정에 이루어 진다. 편집기의 명령어는 자소 모드인 경우와 음절자 모드인 경우에 각각 어떻게 행동하는 지에 관하여 구분하여 설명한다.

2. 커서 이동

커서 이동은 상하, 좌우로 이동이 된다. 상하 이동의 경우 자판에서 상향 화살표와 하향 화살표 키를 누르면 된다. 상향 키를 계속 누를 경우 텍스트 버퍼의 첫 줄에 이르면 더 이상 올라 가지 않고 소리를 내어 사용자에게 경고를 한다. 하향 키 역시 마찬 가지로 텍스트 버퍼의 마지막 줄에 이르면 더 이상 내려 가지 않고 경고음을 낸다. 상하 이동은 자소 모드과 음절자 모드의 특성이 별로 다를 것이 없다.

상하 이동에서 고려해야 할 점은 각 줄은 영문자와 한글이 섞여 있고, 영문자 두 글자와 한글 한 음절자의 폭이 같도록 되어 있다는 점이다. 이것은 현재 줄에서 아래 또는 위쪽 줄로 이동할 경우 위쪽에 있는 줄의 어느 글자에 커서를 놓아야 하는 지를 결정해야 한다.

1: 훈민정음은 과학적인 우수한 문자이다.

2: abc 는 알파벳이라고 한다.

3: 세종대왕은 친히 집현전 학사들과 함께 한글을 창제 하셨다.

여기서 현재 커서가 1 의 '훈'에 있다고 할 때 하향 키를 누르면 영어 abc 의 어느 글자에 커서를 놓아야 할까? 1 의 '훈'에 대응하는 2 의 글자는 'ab'이기 때문에 'a'와 'b' 가운데 어느 것도 가능하다. 하지만 이 경우 처음에 커서가 '훈'에 놓여 있다가 'ab'로 갈 경우라면 커서는 'a'로 가야 한다. 하지만 커서가 처음 2 의 'b'에서 1 로 간 다음 다시 2 로 내려 오는 경우 라면 커서는 'b'되 돌아 와야 한다. 커서는 좌표 값은 영문자 단위로 증감한다. 1 의 '훈'은 가로 좌표 값이 1,2 이다. 2 에서 1,2 는 a 와 b 에 해당한다. 그래서 2 에서 a 또는 b 에 커서가 놓여 있으면 1 의 '훈'으로 커서를 옮겨야 한다.

줄간 커서 이동에서 문제가 되는 것은 짧은 줄과 긴 줄간에 커서를 이동할 때이다. 2 는 1 과 3 에 비하여 줄의 길이가 짧다. 만약 커서가 3 의 '... 한글을 ...'의 '한'에 놓여 있다면 실제 가로 좌표 값은 48,49 이다. 줄 1,과 2 의 끝 좌표 값은 줄 3 의 '한'보다 작은 값이다. 그래서 커서를 위쪽으로 이동시키면 줄 2 와 3 의 줄 끝에 커서가 놓인다.

3. 삽입

(1) 기본 모드

정음형 코드에서 한글의 표현은 2 장에서 이미 언급하였듯이 내부에선 일차원, 외부에선 이차원 즉 음절로 구현한다. 삽입 연산은 텍스트 버퍼에 일차원으로 존재하는 문자열에 버퍼 커서가 놓인 위치에 삽입되는 문자를 삽입한다. 편집기에서 기본 모드는 삽입 모드이다. 그래서 자판에서 누르는 문자는 현재 버퍼 커서가 가르치는 위치에 삽입된다. 여기서 삽입 연산은 자소 모드와 음절자 모드로 구분하여 설명한다.

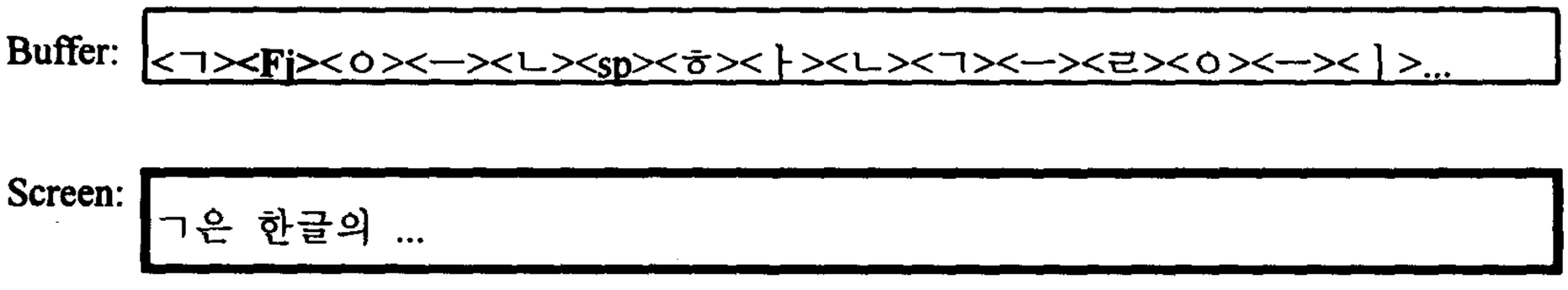


그림 6.3 널중성 삽입

(2) 널 문자 삽입

음절자 모드에서 버퍼 커서는 해당 음절자 문자열의 첫번째 문자에 놓여 있기 때문에 들어 오는 문자는 해당 음절자의 앞에 놓이며 이 때 한글은 음절자를 구성하도록 한다. 예를 들어서 'ㄱ'의 경우 이 문자만이 입력되었다면 뒤에 널 중성자를 삽입하여 이것이 하나의 음절자 형태로 인식되도록 문자열을 구성한다.

(3) 자소 모드에서 삽입 연산

자소 모드에서 버퍼 커서는 한 음절자의 아무런 위치에 놓여 있을 수 있다. 따라서 현재 커서가 놓여 있는 위치 앞에 입력되는 문자를 삽입한다. 이 때 여러 가지 복잡한 문제가 야기될 수 있다. 예를 들어서 '한'은 'ㅎ ㅏ ㄴ'으로 구성되어 있는데 현재 커서 'ㅏ'에 놓여 있다면 여기서 모음 'ㅗ'를 입력하면 '한'은 '환'으

로 바뀌게 된다. 그런데 만약 글자꼴이 지원되지 않는 음절자가 구성되도록 하는 문자가 입력되었을 때는 이들을 적절하게 처리해 주어야 하는데 다음과 같다.

첫째, 기존 문자열과 함께 글자 구성이 되지 않을 경우 아예 입력을 거부한다.

둘째, 음절자가 구성이 되지 않더라도 입력을 한 다음 구성이 되지 않음을 화면에 표시한다.

세째, 음절자 모드처럼 낱자가 입력된 뒤에 음절자 구성이 되도록 처리를 해준다.

위의 세 방법 가운데 둘째 방법은 입력되는 문자를 모두 입력시켜 준다는 점에선 좋으나 화면에서 표현 방법이 어렵고 화면 표현의 일관성 유지가 어렵다. 세째는 음절자 모드의 기능과 같게 하면 일관성 유지 정책이 하나로 유지될 수는 있지만 모드간 역할 분담이 일부 어긋나는 점이 있으나 괜찮은 방법이다. 첫째는 약간 제약적이기 하나 글자꼴이 없는 한 결국 표현할 수 없기 때문에 매우 합리적이다. 이상의 검토에서 첫번째 방법을 지향하면서 셋번째 방법으로 보완하도록 한다.

(4) 버퍼 관리

현재 커서가 놓인 곳에 문자가 삽입되면 뒤에 있는 문자열은 하나 뒤로 밀려 난다. 한 줄이 받아 들일 수 있는 문자의 길이가 결정되어야 한다. 정음형에서 한 줄의 버퍼 크기를 얼마로 해야 할 것인가를 단정적으로 결정하기란 쉽지 않다. 왜냐하면 정음형 코드는 버퍼 내부에서 일차원의 풀어 쓰기 꼴로 들어 있기 때문이다. 화면상의 커서와 버퍼 상의 커서의 위치가 영문자 경우처럼 일치하지 않는다. 그래서 한글이 포함된 텍스트의 경우 한 줄의 길이는 가변적이다.

또한 문자가 계속 삽입되어 버퍼가 일정한 길이를 넘어설 경우 이 문제를 어떻게 처리해야 할 지를 결정해야 한다. 편집기는 워드프로세서와 다르기 때문에 워드프로세서처럼 화면의 폭을 넘는 경우 이를 자동 줄 바꾸기를 할 필요는 없다. 왜냐면 편집기의 기능은 자료를 입력하거나 프로그램을 작성하는 용도로 사용되기 때문이다. 그렇다면 줄의 길이는 한계를 줄 필요 없이 계속 입력할 수가 있다. 이것은 사용자가 화면 뒤에 가리게 되는 부분에 대한 불편 때문에 스스로 엔터키를 입력하여 줄 바꾸기를 하도록 유도한다.

문자 입력에서 한 문자가 입력이 될 때 마다 뒤에 있는 문자를 뒤로 옮기는 것은 시간 소모적이고 효율적이지 못하다. 이를 해결하는 방법은 다음과 같다.

1) 임시 버퍼를 사용해서 입력되는 문자를 받아 들이고 실제 버퍼에는 영향을 미치지 않는다. 그러나 화면에 표시할 때는 실제 버퍼와 임시 버퍼를 통합하여 마치 하나의 버퍼에 있는 것처럼 하는 방식이다.

2) 실제 버퍼의 현재 커서 위치에서 두 개의 버퍼로 분할을 한 다음 커서 앞쪽 문자열이 들어 있는 버퍼 뒤에 입력되는 문자를 계속 추가시켜 나가고 입력이 끝나면 커서 뒤쪽 문자열을 합하여 다시 하나의 버퍼로 만든다.

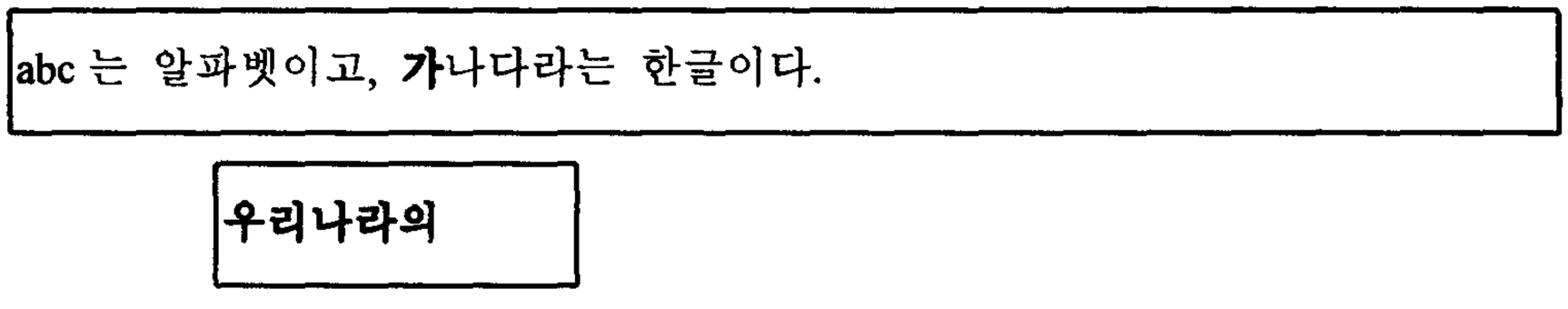


그림 6.4(ㄱ) 임시 버퍼 사용

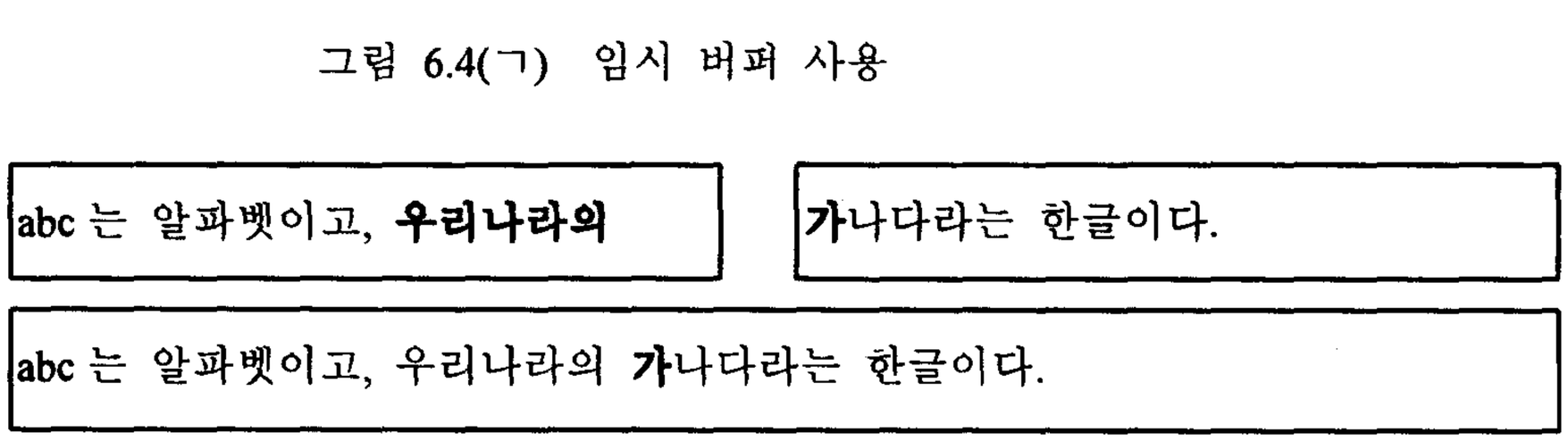


그림 6.4 (ㄴ) 버퍼의 분할과 통합

예를 들어서 그림 6.4 에서 보는 바와 같이 두 방법이 모두 선형 리스트 운용의 문제를 해결하는 방법으로 좋은 방법이다. 앞의 것은 화면에 일치되게 만드는데 있어서 어려움이 있다. 그러나 이 경우는 드문 경우이겠지만 '우리나라의'를 입력하다가 그대로 취소하는 경우 단지 이 임시 버퍼만 버리면 되기 때문에 간단하다. 두 번째, 방법은 첫 번째, 방법 보다 훨씬 간단하다.

줄의 어떤 위치에서건 엔터 키를 치면 새로운 줄이 생기게 된다. 어떤 줄의 가운데서 엔터 키를 치게 되면 두 개의 줄로 분할된다. 이 때 텍스트 버퍼가 링크드 리스트이면 간다하다. 하지만 배열일 경우에 줄과 줄사이에 하나의 새로운 줄을 삽입해야 하므로 연산이 어렵고 시간이 많이 걸린다.

4. 삭제

삭제키(del key)를 치게되면 커서가 놓여 있는 문자가 삭제된다. 커서가 한글의 음절에 놓여 있을 때 음절문자 모드이면 한 음절이 지워진다. 외견상 한 음절이 지워지지만 실제로 해당 음절을 구성하고 있는 문자열이 지워진다. 그러나 음소 모드일 경우에는 좀 더 복잡하다. 예를 들어서 '한'이라는 음절자에서 'ㅎ'에 커서가 놓여 있고 'ㅎ'만이 지워진다고 할 때 남는 글자는 'ㅎ'이 없는 음절을 표시하거나 아니면 음절 구성이 안된다는 경고 문자(예를 들어 가위표)를 표시할 수도 있다. 만약 모음을 없애면 더욱 문제가 될 수도 있다. 그러나 종성을 없애면 이 때는 문제가 없다.

초성자와 중성자의 삭제에서 문제가 되지 않는 경우는 복수의 초성자나 중성자로 구성된 문자를 지울 때는 문제가 되지 않는다. 예를 들어 '과'의 경우 커서가 'ㅗ'에 있다면 삭제 명령이 내려지면 글자는 '가'가 된다. 옛한글에 나오는 음절자인 'ㄳ ㅏ' 의 경우 초성의 'ㅅ'에 커서가 있다면 삭제 명령이 수행되면 '가'가 된다. 그러나 '각'의 경우에 종성 'ㄱ'이 삭제되면 음절자는 '가'가 되어 문제가 되지 않는다.

삭제가 이루어 지면 텍스트 버퍼는 커서가 놓여 있는 뒤에 있는 문자열을 앞으로 이동해야 한다. 이것 역시 선형 리스트의 문제이기 때문에 시간 소모적인 면이 없지 않지만 별다른 방법이 없으며 메모리 연산이기 때문에 크게 문제가 되지 않을 것이다. 삭제의 경우 한꺼번에 많은 문자열을 지우려면 마우스로 블록을 설정한 다음 삭제 명령을 내리는 기능을 두는 것이 바람직하다. 이것은 뒤에서 블록 연산에서 다시 언급한다.

한 줄이 모두 다 삭제가 되면 그 줄을 없애야 한다. 이것은 각 줄이 리스트로 구성이 되어 있기 때문에 링크된 리스트의 삭제 연산으로 해결한다. 이 때 현재 커서가 그 줄의 첫번째 문자에 놓여 있고, 그 줄에는 아직 문자열이 남아 있을 때 역삭제 명령을 내리면 경고음을 내면서 어떠한 행동을 취하지 않는다. 이것은 워드프로세서의 경우 앞 줄로 계속해서 연산이 일어나겠지만 편집기에서 줄의 개념은 정정이어서 화면의 줄과 버퍼의 줄이 일치하고 있기 때문이다. 다시 말해서 화면에 출력되고 있는 줄이 10줄이면 텍스트 버퍼에서 여기에 대응되는 줄 역시 열 줄이다. 마찬가지로 커서가 어떤 줄의 마지막에 놓여 있을 때 거기서 삭제 명령을 내리면 바로 아래에 있는 줄과 별개이기 때문에 경고음이 울리면서 더 이상의 연산은 이루어 지지 않는다.

한 줄의 삭제는 먼저 그 줄에 어떠한 문자도 남아 있지 않은 상태에서 역삭제 명령을 내리거나 그렇지 않으면 해당 줄 삭제 명령을 내렸을 때 이루어 진다.

5. 대치

현재 커서가 놓여 있는 문자가 입력되는 문자로 대치되는 경우이다. 이것은 삽입의 한 종류이다. 즉 수정 또는 덮어 쓰기 모드에서 삽입과 같은 것이다. 이 연산은 버퍼에 있는 해당 위치에 입력되는 문자의 코드와 대치한다. 그러나 정음형의 경우 자소 모드와 음절 모드를 고려하면 조금 복잡한 문제가 야기될 소지가 있다.

음절자 모드에서 화면상에 보이는 음절자의 폭은 일정하지만 버퍼에서 그 문자를 구성하고 있는 문자열의 길이기 다르기 때문에 한 음절식 덮어 쓰기를 하기가 쉽지 않다. 예를 들어서 아래와 같은 경우를 보자.

정인지 선생은 훈민정음해례의 **정**인지 서문을 쓰신 분이다.

현재 커서가 '정인지 서문...'의 '정'에 있다면 여기서 '하위지'를 덮어 쓰기를 해 나간다고 할 때 '정'에 대응하는 입력 문자열 '하'는 두 바이트이다. 이를 때 '하' 입력한 상태에서 화면에 보이는 내용은 '정'의 'ㅇ'이 아직 남아 있기 때문에 '하'가 '항'이 된다. 이것은 입력하는 과정에서 사용자가 사전에 이러한 문제를 인지하고 있을 경우에는 문제가 되지 않겠으나 일반 사용자의 경우에 이것은 오해와 혼란의 소지가 크다. 여기서 음절자 모드에서 '하'의 'ㅎ'이 입력되면 '정'의 문자열이 지워지게 할 수도 있다. 이것은 그 뒤의 문자열과 조정의 문제를 남긴다. 이 경우 '정'과 '하'의 입장이 바뀌었다고 할 때 '하위지'의 '위'자가 깨어지는 문제가 발생한다. 하지만 이것 역시 사용자가 자연스레 받아 들이고 이를 이해한 바탕 위에 입력을 수행한다면 문제가 없다.

6. 블록 연산

한 문자씩 삽입 또는 삭제를 하는 경우가 아닌 한꺼번에 많은 문자열 또는 문자열 블록을 옴기거나 복사와 삭제를 하는 경우에 마우스로 블록을 설정하고 여기

에 맞는 연산을 수행한다. 볼록 연산에는 이동, 삭제, 복사가 있다.

이동 연산은 버퍼의 각 줄이 연결된 리스트로 구성되어 있기 때문에 마우스로 블록을 설정하고 이를 옮기는 것은 연결된 리스트에서 줄의 위치를 바꾸는 연산에 해당한다. 단지 마우스가 어떤 줄의 중간 또는 일부에 걸려서 시작하거나 끝나는 경우만 처리하면 된다. 해당 줄의 처리는 삽입에서 엔터키가 입력되었을 경우와 같이 일단 두 줄로 분할 한 다음 줄 노드를 버퍼의 줄 리스트에 삭제와 삽입을 한다. 블록을 설정할 때 마우스가 읽어 낸 좌표가 어떤 줄에 걸쳐 있으면 그 줄을 포함한다. 이 연산은 화면의 좌표 값으로 그와 관련된 버퍼를 찾아 내는 것이다. 이 연산은 먼저 화면의 첫 번째 줄부터 시작하여 글자의 높이를 계산한다. 예를 들어서 마우스의 상단 왼쪽 좌표가 (35,72)일 때 버퍼의 좌표 값을 계산하는 문제를 고려하자. 글자의 높이가 32 일 때 화면의 첫 줄로부터 64 이면 둘째 줄이고 96 이면 셋째 줄이다. 72 는 그 사이 값이므로 셋째 줄이 된다. 한글의 글자 폭이 32 이고 영문자가 16 일 때 버퍼의 내용이 모두 한글이면 두 번째 셋째 줄의 두 번째 음절에 해당한다.

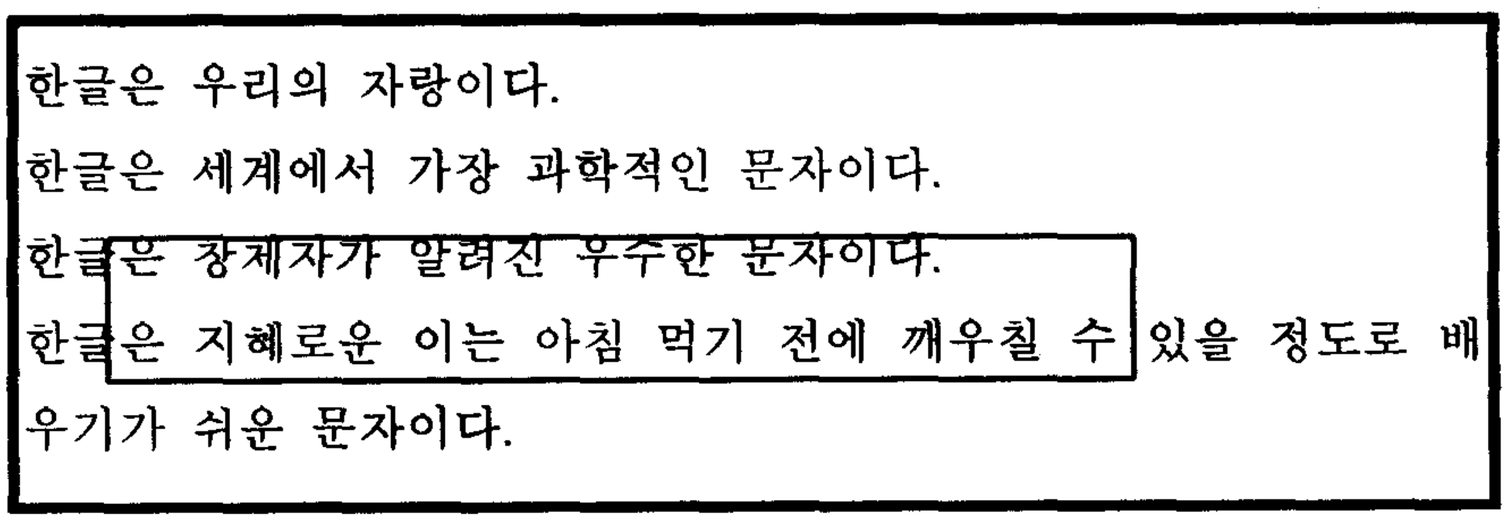


그림 6.4 마우스로 블록 설정

삭제의 경우에는 이동 보다는 좀 더 간단하다. 해당 줄을 리스트에서 제거하는 작업으로 연결된 리스트의 삭제 연산에 해당한다. 단지 이동에서와 마찬가지로 중간에 걸쳐 있는 부분을 분할하여 이들 줄들을 제거한다.

복사는 블록 이동 연산과 유사하며 이동 연산에서 설정된 블록을 삭제하는 부분을 없앤 것과 같다.

7. 파일 읽기

이제 까지 여러 종류의 한글 코드가 제정되었다. 그러나 현재 표준은 완성형과 조합형이다. 그래서 이들 코드로 작성된 많은 자료가 있다. 하지만 정음형 편집기에 올려서 완성형이나 조합형으로 된 자료 또는 프로그램을 수정하려고 한다면 각 코드의 변환기를 거쳐서 입력되어야 한다. 파일 읽기는 다른 경우와 같지만 단지 여기서는 정음형 편집기이기 때문에 완성형을 정음형으로 변환하는 기능을 내포하고 있다.

8. 파일 쓰기

파일 쓰기는 파일 읽기의 반대 기능을 가지고 있는데 여기서 관심 사항은 정음형으로 된 자료 또는 프로그램을 완성형이나 조합형으로 저장하는 것이다. 그런데 정음형이 지원하는 문자의 수가 많기 때문에 완성형이나 조합형으로 번역을 할 때 표현할 수 없는 문자가 많을 수 있다. 이것에 대하여 완성형의 경우 없는 음절에 대하여 자모 코드로 풀어 쓰기를 하는 규칙이 마련되어 있어서 이것을 따른다. 조합형은 현대 한글의 경우는 문제가 없고, 옛한글의 경우 일반적으로 자주 쓰이는 것은 문제가 없다. 단지 잘 쓰이지 않는 것은 현재 자소가 반영되어 있지 않아서 표현할 수가 없지만 현재 우리가 쓰고 있는 음절자의 범위 밖에 있기 때문에 문제가 되지 않을 것이다. 그래서 이 부분은 구현에서 고려하지 않는다.

3절. 내부 자료 구조

1. 텍스트 버퍼

앞에서 서술한 바와 같이 비쥬얼(Visual) C++는 문서-뷰 모델(Document-View Model)을 제공하는데 그 역할은 프로그램에서 처리하는 데이터와 사용자의 입력을 받아 처리하는 작업으로 분리되어 있다. 문서 객체는 프로그램에서 사용하는 데이터의 관리와 처리 작업을 책임지고 있으며, 뷰 객체는 사용자의 입력을 처리하는 작업을 책임지고 있다.

여기서 텍스트 버퍼는 비쥬얼 C++의 Cstring 객체를 기본적으로 사용하며, 이들은 다시 CstringList 클래스 객체를 활용하여 텍스크 버퍼를 구성하여 그곳에 정

의되어 있는 멤버 함수를 활용하도록 한다. CStringList 는 Cstring 객체의 연결된 리스트 구조이며 편집기의 명령에 따라서 텍스트 줄의 삽입, 삭제 또는 이동이 용이하다는 점에서 선택하였다. CStringList 를 활용한 텍스트 버퍼의 자료 구조는 그림 6.5 와 같다.

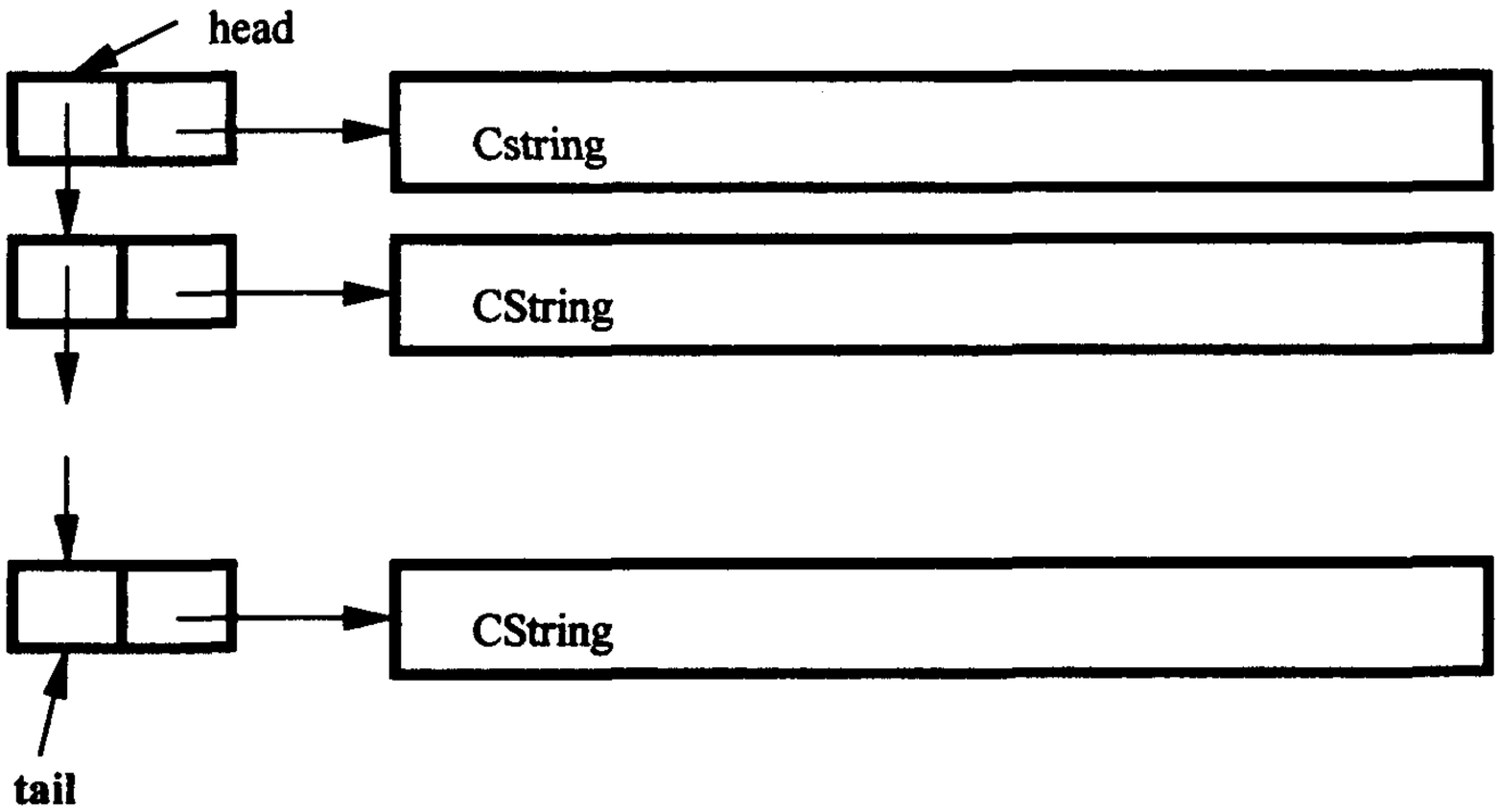


그림 6.5 텍스트 버퍼 구조

MFC(Microsoft Foundation Class) 의 CDocument 라는 클래스는 모든 문서 클래스를 정의하는 원시 클래스이다. 여기에 그림 6.5 와 같이 CDocument 를 이용하여 CStringList 를 선언함으로써 CString 이라는 텍스트 버퍼 구조로 설계한다. Cstring 객체는 새로운 줄이 만들어지면 생성되어 CStringList 라는 리스트에 등록된다. 이것은 연결 리스트이며 제공되는 멤버함수를 이용하여 처리할 수 있다. 이 텍스트 버퍼에는 정음형 코드가 음소문자 특성을 가지고 일차원으로 저장되어 있다. 여기에 저장된 풀어 쓰기로 된 문서를 모아 쓰기하여 화면에 보여 주어야 한다.

2. 화면 버퍼

문서 내용을 윈도우에 표현하는 것은 뷰의 중요한 기능으로 윈도우를 통하여 문서의 내용을 사용자에게 보여 주는 것이다. 문서는 프로그램에서 사용되는 데이터를 저장하고 있으며, 이를 사용자에게 보여 주는 것이 뷰이다. 뷰는 문서의 일부분을 윈도우로 보여 주는 기능과 함께 한글의 경우 다른 기능을 이 사이에 처

리해야 한다. 한글은 훈민정음 창제 원리에 따라서 일차원의 풀어 쓰기 형태로 컴퓨터 내부에 저장되어 있으며 읽기 능률을 높이기 위하여 그리고 소리의 단위인 음절간 구별을 위하여 이차원의 모아 쓰기 형태를 취한다. 문서와 뷰 사이에는 한글의 위상 기하 구조가 존재한다. 이제까지 한글 코드를 제정하면서 우리가 늘 회피하여 왔던 부분이 바로 이 부분이다.

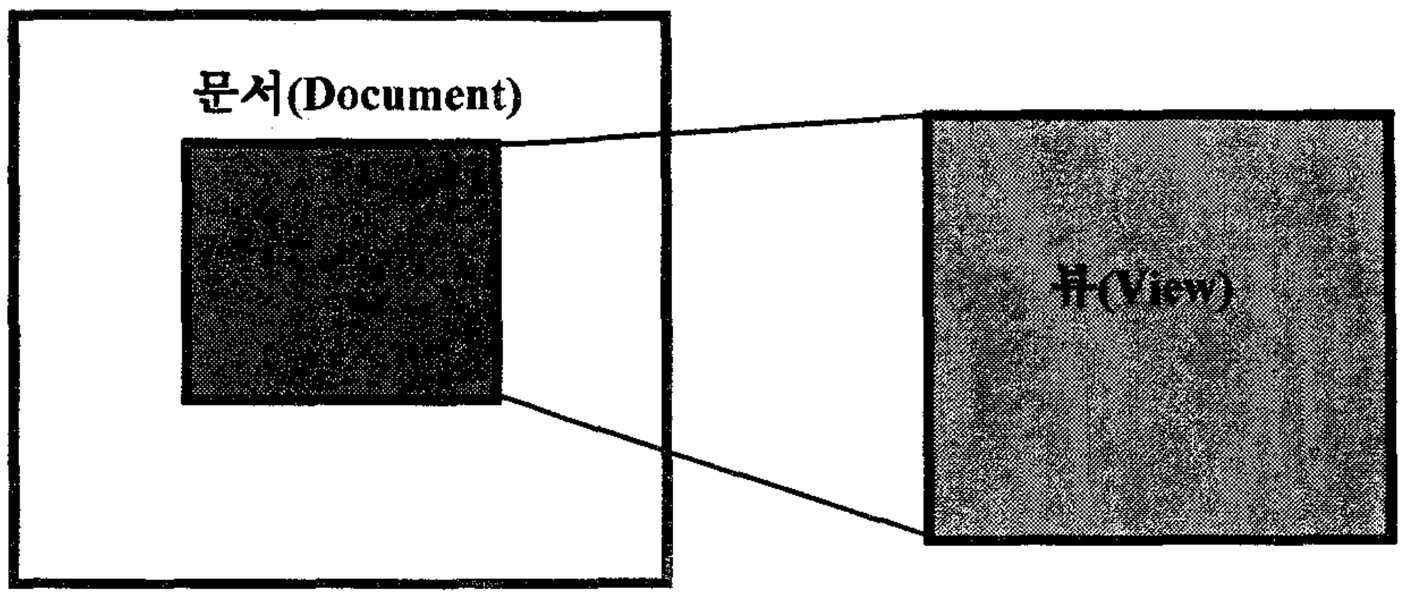


그림 6.6 문서-뷰의 관계

한글이 가지고 있는 음소 및 음절 문자 특성으로 말미암아 음소 문자 특성만을 가진 로마자의 경우와 음절 문자 특성만을 가진 가나에 대하여 한글은 두 개의 특성을 동시에 가짐으로써 그것이 장점으로 파악되지 못하고 거추장스럽고 로마자나 가나 보다 못난 글자로 인식되어 왔다. 적어도 로마자와 가나는 이러한 특성을 가지고 있지 않기 때문에 그 문자와 비교하여 두 가지 특성을 가진 것을 단점으로 파악하여 왔다. 그래서 둘 가운데 음소 문자 특성을 없애면서 만든 것이 완성형이다. 문서-뷰 모델은 바로 이러한 두 가지 특성을 정음형 코드를 통하여 비쥬얼 C++의 특성에 접목하는 포인트이다.

문서에 있는 텍스트를 사용자가 볼 수 있도록 전환하는 과정에서 한글의 음소 문자 특성에서 음절 문자 특성으로 변환하는 전환 기능을 첨가한다. 편집기의 텍스트는 수순한 텍스트 이외의 요소들을 포함하고 거의 있지 않다. 그래서 로마자나 가나의 경우 버퍼의 좌표와 화면의 좌표 값이 거의 일정 비율로 일치하고 있다. 한글의 경우 두 바이트 완성형과 조합형에선 이것이 일정 비율로 일치한다. 그래서 대체로 별 어려움 없이 기존의 로마자 취급 편집기의 내용의 일부분의 수정으로 쉽게 편집기를 개발할 수 있었다.

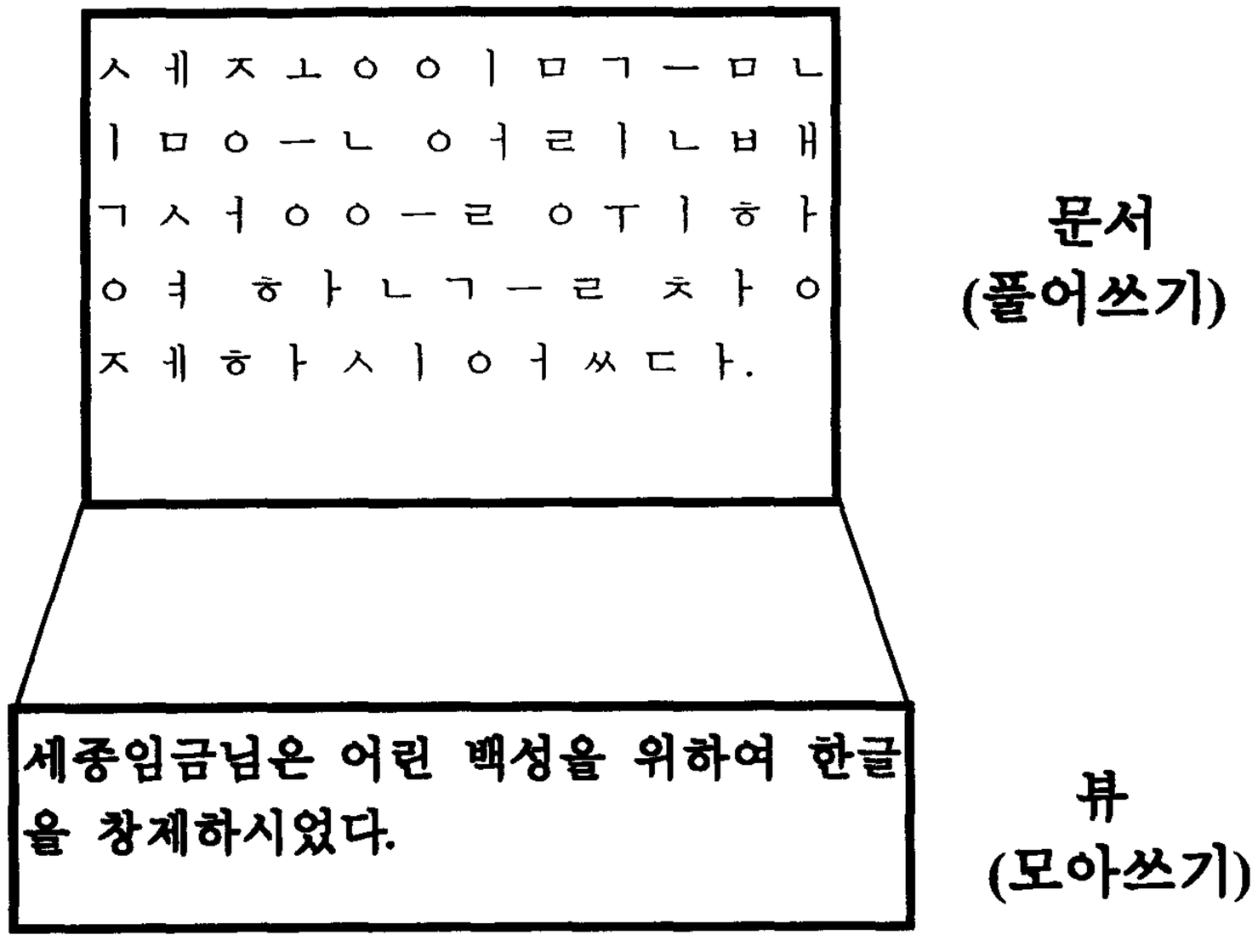


그림 6.7 문서-뷰에서 한글 표현 방법

하지만 기존의 표준이었던 자모형(소위 n-바이트 한글 코드) 한글 코드는 한 음절의 길이가 두 바이트에서 다섯 바이트 까지 일정하지 않아서 버퍼와 화면 좌표 값의 계산이 필요하다는 이유로 사용이 회피 되어 왔다. 정음형은 자모형의 그 차이 보다 더 심하다. 왜냐면 자모형은 그래도 복자모에 대하여 코드화를 하였기 때문에 그래도 최장 음절자의 자모의 수가 5 바이트이었지만 정음형은 낱자소에 대하여 코드화를 하였기 때문에 현대 한글의 경우에 최고 6 바이트 까지 그리고 옛 한글의 경우에는 최고 9 바이트로 구성이 된다. 여기서 음절의 수가 일정하지 않기 때문에 완성형이나 조합형에서처럼 고정 비율의 양자간 일치를 기대하기 어렵다. 그러나 문제는 그렇게 어렵지 않다. 버퍼에 있는 자소의 수가 어떠하든지 해당 줄에 포함된 영문자의 수와 한글의 음절 수만 파악되면 화면에서는 글자꼴을 표현하기 때문에 한글과 영문자의 글자 폭이 2 배라는 점만을 고려하여 계산하면 간단하게 해결된다.

문서-뷰 모델에서 문서와 뷰간에 사상 관계는 문서에서 일차원의 한글 텍스트를 뷰에 이차원으로 보일 때 한글 음절과 로마자의 수를 각각 계산하고 양자간에 2 대 1 의 고정 폭인 글자꼴을 사용하기 때문에 글자 폭에 이것을 반영하여 계산

하면 화면에서 픽셀의 주소가 계산될 수 있으며 버퍼의 좌표 값과 화면의 픽셀 좌표 값간의 사상 관계를 정의할 수 있다. 이러한 함수를 갖추면 편집기의 경우 많은 글자꼴을 사용하는 워드프로세서에 비하여 간단하게 구현할 수 있다.

'바른글'에서는 고정 폭의 글자꼴을 쓰며 이들은 초성자, 중성자, 종성자 글자꼴을 겹쳐쓰기를 통하여 조합하여 화면에 표시한다. 그래서 어떤 특정한 화면의 위치에 대응하는 버퍼에 있는 음절자의 문자열을 추출할 수 있어야 하고, 또한 화면 커서에 대응하도록 버퍼 커서를 이동시킬 수 있어야 한다. 마찬 가지로 버퍼 커서를 이동하였을 때 대응하는 화면의 커서 좌표 값(픽셀 좌표 값)을 계산할 수 있어야 한다. 이러한 사상 관계를 구현하는 함수만 준비하면 정음형의 구현은 그다지 어렵지 않다.

4절. 기본 함수

1. 문자열에서 한.영 글자수 계산

버퍼 커서와 화면 커서가 고정 비율 대응성을 가지지 못하기 때문에 양자간을 일치시켜 주는 장치가 있어야 한다. 한글과 영문자의 폭이 2:1 비율을 가지기 때문에 현재 커서가 있는 버퍼 줄에서 커서가 있는 곳까지 한글 음절자의 수와 영문자의 수가 얼마인지를 알아야 한다. 물론 이 때 한글과 영문 글자꼴들은 모두 고정 길이와 폭을 가진 글자꼴을 사용한다. 화면 커서의 좌표 값은 디스플레이 콘텍스(DC)의 픽셀 좌표 값이다. 픽셀의 x, y 좌표 값을 구하는 식은 다음과 같다.

```
y=tm.height * lineCount_On_Screen;
x=UmjulCount * 32 + EngCount * 16;
```

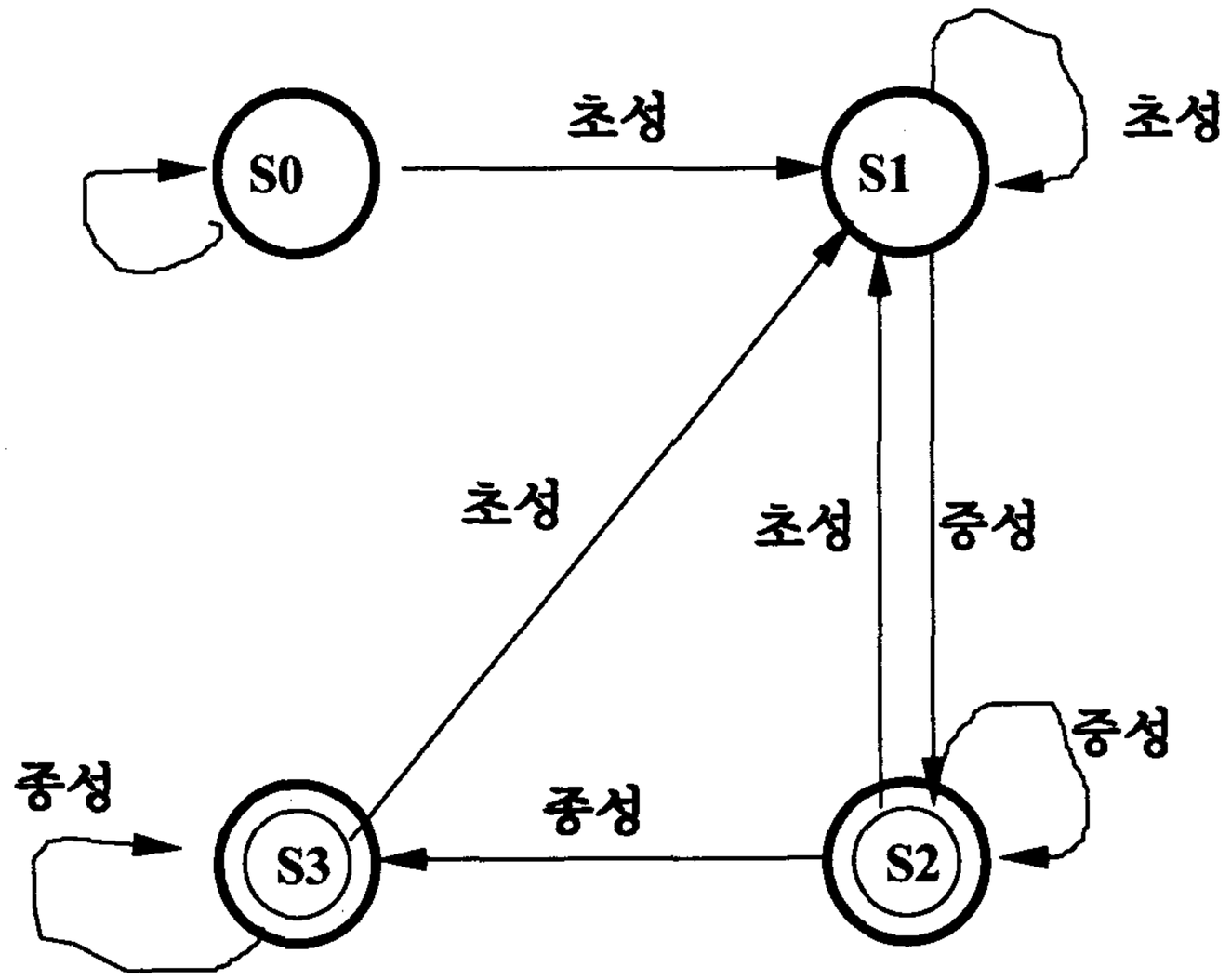


그림 6.9 정음형 접수기

그러면 여기서 주어진 문자열에 내포된 영문자와 한글 음절의 수를 계산하는 방법은 다음과 같다. 여러 가지 방법이 있지만 정음형 한글 코드가 섞여 있기 때문에 이것은 정음형 코드의 음절을 인식하는 간단한 오토마톤(accepter)으로 쉽게 해결할 수 있다. 그 오토마톤의 결과를 Umjul 에 FALSE 또는 TRUE 로 반환한다면 다음과 같은 함수로 정의한다.

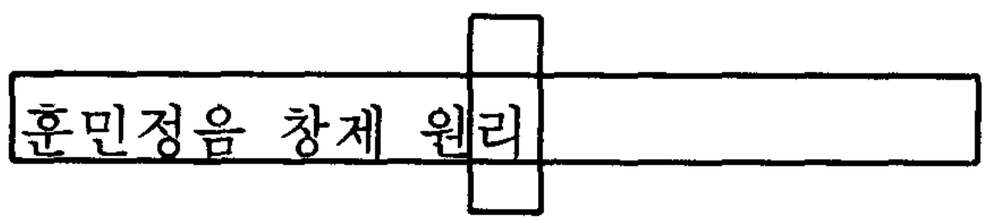
```
CPOINT GetCountHE(Cstring * hanStr, bufferX) {
// 문자열의 bufferX 문자의 시작점까지 픽셀의 x 좌표
    CPOINT hanEng;
    han=eng=0;
For(i=0; i<cursorX; i++){
    Umjul = JUAutomaton(hanStr[cursorX]);
    If (Umjul) ++han;
    else   ++eng;
}
```

```
        hanEng.x=han; hanEng.y =eng;
        return(hanEng)
}
```

그림 6.10 픽셀의 x 좌표 계산 함수

함수 GetCaretX()의 결과는 텍스트 버퍼의 x 좌표 값인 bufferX 가 가리키는 문자를 포함하는 한글 또는 영문 글자꼴의 상단 왼쪽의 픽셀 x 좌표 값이다. 그렇기 때문에 bufferX 가 가리키는 글자꼴 이전까지 한글 음절의 수와 영문자의 수를 얻어서 각각의 글자 폭을 곱하여 계산한다. 다음 예를 보자.

HanStr :



'리' 바로 앞까지 한글 음절 수는 7 개이고, 영문자 수는 2(space)이다. 따라서 글자의 폭이 한글이 32 이면 결과는 7 * 32 + 2 * 16 = 224 + 32 = 256 이다.

```
nBufX = 17;
int CaretX = GetBufX2CaretX(hanStr, nBufX);
...
```

hanStr 버퍼에 대한 커서 위치는 nBufX 에 있다. 여기서 nBufX 를 포함하는 한글 또는 영문자의 픽셀 좌표 값을 구하는 함수의 정의는 다음과 같다.

```
int GetBufX2CaretX(hanStr, nBufX){
   CPOINT HE;
   HE = GetCountHE(hanStr, nBufX);
   caretX = HE.x *32 + HE.y * 16;
   return caretX;
}
```

함수 GetCountHE()의 반환 값은 CPOINT 타입을 활용하여 두 개의 정수 값을

반환하였다. CPOINT 타입을 이용한 것은 단지 두 개의 값을 반환 받기 위한 것 이외의 목적은 없다. 화면 커서의 값은 영문 커서의 값을 기반으로 계산한다. 예를 들어서 위의 '리'에서 논리적 화면 커서 값은 실제 한글을 기준으로 했을 때 다시 말하면 오른 쪽 화살표 키를 눌러서 '리'에 옮기자면 10 번을 눌러야 한다. 그것은 한글 음절이 7 개이고 영문 공간이 2 개이기 때문이다. 하지만 한글은 영문의 두 배 크기이기 때문에 영문자를 기준으로 하면 17 개 눌러야 한다.

2. 문자열에서 음절 문자열의 추출

정음형 코드는 버퍼에서 일 차원인 풀어 쓰기 꼴로 존재한다. 앞에서 언급하였듯이 화면 커서와 버퍼 커서간의 사상 관계를 만들기 위하여 어떤 문자열이 주어졌을 그 문자열에서 몇 번째 음절의 문자열에 관한 다양한 정보가 필요하다. 함수를 정의함에 있어서 다양한 방법을 고려할 수 있다. 함수의 입력 매개 변수로서 문자열 포인터와 글자의 화면상 논리적 위치를 주고, 해당 음절의 문자열을 직접 반환하게 할 수도 있고 그와 관련된 다음 정보를 반환하게 할 수도 있다.

- k 번째 음절자의 시작 주소
- k 번째 음절자의 길이

함수의 일반성을 위하여 두 번째 방법으로 함수를 정의한다. 그러면 위 두 정보를 가지고 주어진 문자열로부터 해당 음절 문자열을 추출할 수도 있고, 다른 용도로 활용할 수도 있기 때문이다. 함수로 정의하면 반환 값이 하나이기 때문에 두 개의 함수로 정의하였다. 반환 값 가운데 하나는 음절의 시작 문자 주소이고, 다른 하나는 음절의 길이이므로 정수이다.

```
CString* GetUmjulBegin(CString *, nPos);
int GetUmjulLength(Csting *, nPos);
```

3. 음절 문자열에서 자소 추출

한글의 한 음절자는 초성자, 중성자, 종성자로 구성이 되어 있다. 이들은 최대 석

자까지 구성될 수 있다. 이들 각각의 문자열 추출이 필요한 이유는 국어 정보처리에서 언어 정보를 분석하기 위해서 이다. 하지만 편집기에선 각 자소 문자열의 글자꼴이 있는 지를 확인하는 과정에서 필요하다. 확인이 되면 이들을 조합하여 화면에 출력한다. 함수의 반환값은 각각의 문자열이다. 이 문자열은 널로 문자열 끝이 처리되어 있다. 함수의 매개 변수는 주어진 문자열 포인터와 자소 구분이다.

Cstring GetJaso(Cstring, nFlag);

*nFlag 값 0 : 초성자, 1 : 중성자, 2 : 종성자

Cstring cho = GetJaso(CStrData, 0);

Cstring jung = GetJaso(CStrData, 1);

Cstring jong = GetJaso(CStrData, 2);

4. 화면 좌표 값과 버퍼 좌표 값의 사상

커서의 성격은 세 가지로 구분할 수 있다. 텍스트 버퍼의 좌표, 화면 픽셀의 좌표, 화면의 논리적 위치 좌표이다. 이것을 보다 명확하게 구분하기 위하여 그림을 통하여 설명한다.

버퍼에서 17 번째 위치에 '정'의 문자열이 시작한다. 그러면 '정'의 버퍼 주소는 16 이고 길이는 3 이다. 그리고 화면에서는 한.영문자를 포함하여 10 번째에 위치해 있으며, 영문자 기준 커서는 15 번째에 위치한다. 그리고 픽셀 주소로는 16*4 + 32 * 5 = 224 이다. 편집기에서 어떤 커서 값이 주어지면 그것을 다른 관련 다른 커서 값으로 모두 계산이 가능하여야 한다. 그와 같은 요구는 다음과 같다.

1) 버퍼의 좌표 값이 주어지면 화면의 픽셀 주소, 논리적 커서 위치 등을 얻을 수 있어야 한다.

2) 화면의 픽셀 주소 값이 주어지면 여기에 해당하는 버퍼의 주소 값을 얻을 수 있어야 한다.

3) 화면의 논리적 커서 값이 주어지면 화면의 픽셀 값과 버퍼의 해당 문자열

의 주소 값을 얻을 수 있어야 한다.

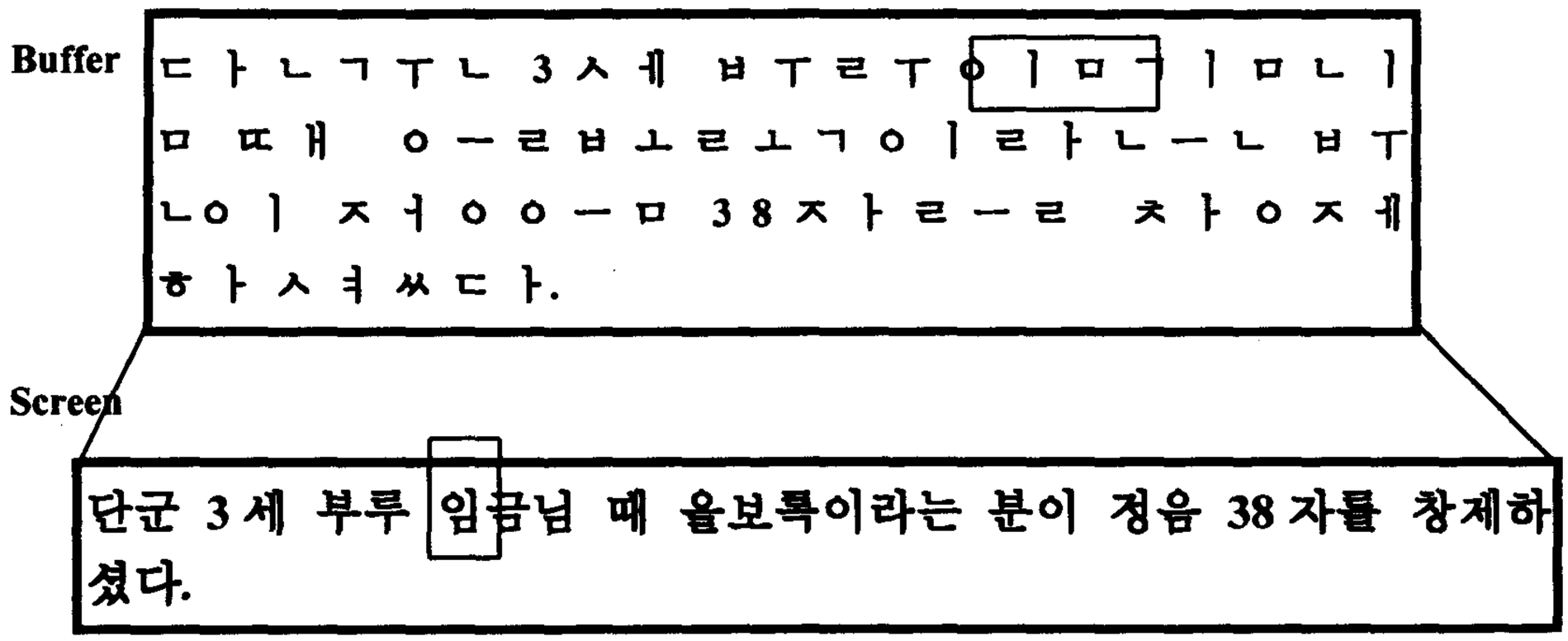


그림 6.12 커서의 구분

이상의 요구를 만족시키는 함수는 앞에서 정의한 함수들을 활용하여 만들 수 있다. 첫번째 요구는 커서를 옮길 때라든지 특정한 위치에 어떤 문자를 삽입하였을 때 요구된다. 예를 들어서 음절 모드 연산에서 오른 쪽 커서를 눌러서 오른쪽으로 한 음절 이동하였다고 하자. 그러면 새로 옮긴 음절의 화면상 위치를 알아야 커서를 놓을 수 있다. 커서를 옮겼을 때 그 글자는 영문자일 수도 있고, 한글일 수도 있다. 이 경우에 화면상 픽셀 주소 계산이 먼저 이루어져야 한다. 그것은 버퍼의 좌표 값을 기준으로 계산할 수 있다. 그래서 다음과 같은 절차를 거친다.

(1) 새로 옮긴 음절 이전까지 한글 음절의 수와 영문자 수를 계산한다.

(2) 이를 이용하여 화면상의 픽셀 주소(caretX)를 계산한다.

(3) 커서의 주소를 셋팅한 다음 커서를 옮겨서 보이기 한다.

```
CPOINT HE = GetCountHE(Cstring, nBufX);
caretX = HE.x * 32 + HE.y * 16;
CaretPos.x = caretX;
CaretPos.y = current_line * 32;
SetCaretPos(CaretPos);
```

```
...
invalidate();
```

(2)에서 화면의 픽셀 주소로써 버퍼의 주소를 추적할 때 여러 가지 상황을 고려해야 한다. 이 때는 커서를 상하로 옮겼을 때를 상정한다. 현재 줄에서 픽셀의 주소가 200 이라면 아래 줄에서 100 에 해당하는 버퍼의 해당 글자는 어느 것인가를 찾아 내야 한다.

```
Int nBufX = GetCaret2BufX(current_line, Caret.x);
```

(3)에서 화면의 논리적 커서 좌표로써 픽셀 주소와 버퍼 주소를 계산하는 함수를 정의한다. 화면의 논리적 주소는 두 가지로 구분이 된다. 예를 들어서 영문자와 한글 음절을 단위로 계산하는 경우인데 이 경우에 커서를 움직이는 회수가 된다. 그런데 영문자의 폭이 한글의 절반이므로 논리적 커서 위치 값은 실제로 사용자 위주로 생각한 것이고 그것은 실제로 여기서 도움이 되지 않는다. 그래서 영문자 위주의 전체 커서 값을 가지고 계산한다.

```
nBufX = GetCursor2BufX(current_line,Cursor);
Caret.x = GetCursor2CaretX(current_line,Cursor);
```

결국 화면 상의 논리적 커서 좌표 값이란 x 의 경우 한글 음절자의 수의 2 배에 영문자의 수를 합한 값이 된다. 예를 들어서 수식은 다음과 같다.

```
int GetCursor2BufX(current_lint, Cursor) {
    CPOINT HE;
    for(int nbufx= 0; nbufx < Sbuf[current_line].length(); nbufx++){
        HE = GetCountHE(Sbuf[current_line], nbufx);
        if (Cursor => (HE.x *2 + HE.y)) break;
    }
    return nbufx;
}
```

픽셀 주소 즉 캐럿 x 값을 얻어려면 화면의 논리적 커서 값으로 먼저 버퍼의 좌표 값을 얻은 다음 다시 이를 이용하여 앞에서 개발된 함수 GetBufX2CaretX() 함수를 활용하여 계산한다.

```
int GetCursor2CaretX(current_line, Cursor) {
    // 논리적 화면 커서 값을 픽셀 값으로 변환한다.
    Int nbufx = GetCursor2CaretX(current_line,Cursor);
    return GetBufX2CaretX(pDoc->hanStr[current_line], nbufx);
}
```

이미 앞에서 개발된 함수들을 활용하여 화면상 논리적 커서 값을 버퍼와 픽셀의 x 값을 얻는 함수와 알고리즘의 일부를 보았다. 이러한 번거로운 작업이 이루어져야 하는 것은 정음형 코드가 영문자처럼 일차원 꼴을 하고 있음에 연유한다. 그런데 이것이 영문자에선 문제가 되지 않지만 한글에선 문제가 된다. 그것은 영문자에서는 음절의 구조가 글자 구조에 나타나지 않지만 한글은 그것을 글자에 반영해 두고 있기 때문이다.

비쥬얼 C++에서 문서-뷰 모델을 제공하고 있어서 개념상 훈민정음 창제 원리를 적용하기가 쉽다. 그러나 우리가 어렵고 귀찮게 여기는 이유는 단지 이러한 문자의 특성을 모르기 때문이며, 이러한 일이 한글에서는 당연한 일로 받아 들이고 있지 못한데 원인이 있다.

전체적인 설계는 커서의 이동, 문자의 삽입, 삭제, 변경 연산에서 화일에 저장, 하고 읽어 오는 기능들이다. 그리고 한글은 자소와 음절 모드 두 가지를 가지고 있어서 기존의 다른 한글 편집기와 그 개념을 달리할 뿐만 아니라 훈민정음 창제 이래로 사용한 모든 옛 한글을 모두 표현할 수 있는 기틀을 가지고 있다는 점에서 본 설계의 의미를 높이 두고자 한다.

7장. 편집기 '바른글'의 구현

1절. 정음형 오토마톤 접수기(Accepter)

1. 함수 및 변수 작명법

프로그램을 개발할 때 함수 이름, 변수 이름 등에서 이름 그 자체가 뜻을 가지고 코멘트를 대신할 수 있으면 좋다. 그래서 본 연구에선 프로그램을 개발하면서 작명하는 작명법을 제정하기로 하였다. 일단 MFC 라이브러리 관련 함수는 모두 대문자로 시작하고 함수 이름에 나열된 단어의 첫 글자는 대문자로 쓴다. 그리고 변수의 경우 윈도우에서 관례적으로 붙이는 접두 규칙을 따를 것이다. 또한 각 클래스에서 정의하여 객체 지향 개념의 구현 방식을 그대로 따른다.

접 두 어	의 미	접 두 어	의 미
a	배열(array)	l	정수
b	BOOL	m_	클래스의 데이터 멤버
c	char	n	int
cr	색상 참조값	p	pointer
cx, cy	x, y 길이 계수	pt	point(x,y 좌표)
dw	dword	s	문자열
f	flag(BOOL)	sz	널로 끝나는 문자열
fn	함수(필요한 경우만)	tm	TEXTMETRIC
h	핸들	w	워드

표 7.1 윈도 변수의 접두어 관례

예를 들어서 문자열 포인터 변수나 CString의 포인터 변수는 p를 접두어로 하여 작명한다. 특히 인덱스 변수의 경우 단순히 j, k, m, n 등을 많이 쓰는 데 이것만으로는 구분이 어려울 경우가 많다. 이렇게 함으로써 소스 문서화를 별도로 하는 것을 줄일 수 있고 소스를 수정하는데 크게 도움이 되도록 한다.

```
Char * pChar;
Cstring* pStr;
```

2. 한글 오토마타 개발

한글은 조직적이고 규칙적인 문자 구조를 가지고 있다. 그래서 보통 문자에서 볼 수 없는 편리함을 제공한다. 한글은 자판에서 입력할 때 일정 수 이상의 문자를 연속해서 입력할 수 없다. 왜냐면 글자의 구성이 되지 않기 때문이다. 한글은 기본 자소와 모아 쓰기의 규칙에 의거하여 음절자를 생성하도록 설계되었다. 그래서 통상적으로 우리가 한글 오토마타(Automata)라고 하는 것은 세 가지의 오토마톤 가운데 접수기(accepter)를 지칭한다. 접수기는 입력되는 선형 문자열을 읽고 정해진 규칙에 따라서 on, off를 응답하는 역할을 한다. 접수기는 입력된 자료를 어떠한 형태로든 번역하거니 변경시키지 않는다. 한글에서 접수기는 일차원의 문자열이 자판으로부터 입력이 되면 이들을 읽고 음절 구성 여부를 판단해서 on, off 로 응답을 한다. 따라서 요즈음 대부분의 오토마톤에서는 음절 구성이 되지 않을 때 경고음을 울리지 않고 있는데 이것은 수정되어야 한다. 변환기(Transducer)는 입력되어 들어 오는 문자열을 일정한 규칙에 따라서 변환하는 기능을 가지고 있다. 또한 생성기(Generator)는 신호나 자극이 있을 때 문자열을 생성해 준다. 이들 삼자를 각각 칭하여 오토마톤(Automaton)이라고 하고, 통틀어 복수형이 될 때 오토마타(Automata)라고 부른다. 한글 편집기에선 접수기와 변환기를 사용한다. 이들은 입력계, 편집계, 출력계에 걸쳐서 적용한다.

1) 입력계에서 자판 KS C 5715(자모형)에서 타자되어 들어 오는 문자열이 음절 구성이 되는지 여부를 검사하고, 자모형을 자소형인 정음형으로 변환한다. 앞에서 하는 역할은 접수기 역할이고, 뒤에서 하는 역할은 변환기 역할을 한다.
2) 편집계에서 정음형으로 변환되어 버퍼에 저장되어 있으며, 이들은 한글의 음절을 계산할 때는 접수기가 주로 활용된다.
3) 출력계에선 정음형 문자열을 이미지 문자꼴로 변환한다. 이것은 화면에 이미지 글자꼴(font)을 출력하는 것을 말한다.

정음형 편집기에서 사용되고 있는 정음형 오토마톤이 자판에서 읽어 들인 문자열을 화면이나 인쇄기에 출력할 때까지 접수기와 변환기를 활용한다. 그림으로 이들의 기능을 그리면 다음과 같다.

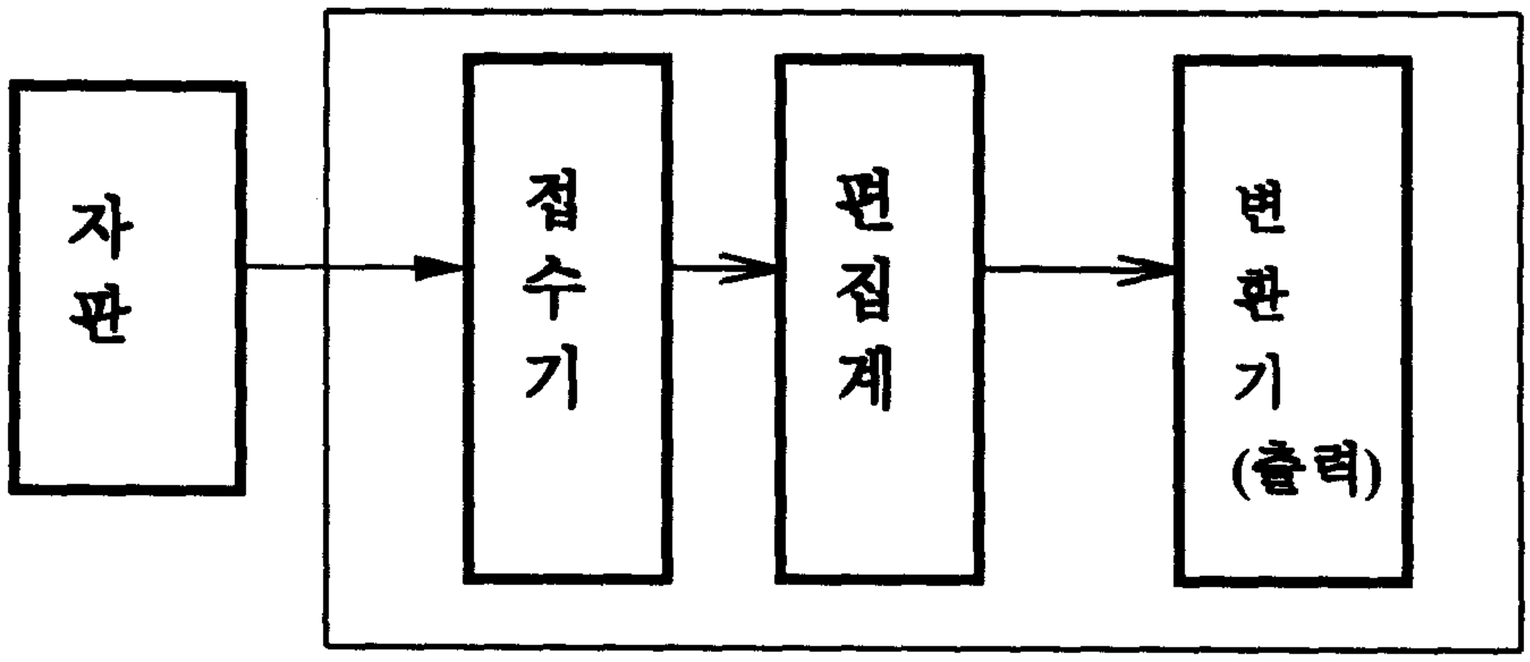


그림 7.1 편집기 '바른글'에서 오토마톤

그런데 여기서 자모형 자판이 아니라 자소형 자판의 경우에는 매우 간단해 진다. 왜냐면 단순히 변환기의 기능만 가지게 되며, 본 연구의 초기 단계에선 접수기 없이 소문자 자음은 초성으로, 모음은 그대로, 대문자 자음은 종성으로 하여 구현하였다. 이것은 단순히 자판에서 들어 오는 ASCII 값을 정음형 코드로 사상하는 정도의 역할을 하였다. 또한 한글 코드가 자소형이 되면 자모형이 자음에 대하여 모호성을 가지고 있어서 한 글자 앞을 내다 보아야 하는 (lookahead one)을 해야 하는 부담을 가지지 않는다. 그러므로 비쥬얼 C++의 뷰 클래스의 멤버 함수로 OnChar() 함수에서 코드 변환만으로 간단하게 해결된다. 코드 변환은 대응 배열을 만들어서 해결하였다.

```
Char aAsciiCho[]        = {'r','s','e',  ...,  'g'};
char aJeongEumCho[]     = {'ㄱ','ㄴ','ㄷ', ..., 'ㅎ'}
char aAsciiJoong[]      = {'k','o','O','y',  ....,  'l'}
char aJeongEumJoong[]   = {'ㅏ','ㅐ','ㅒ','ㅑ',...., 'ㅣ'}
char aAsciiJong[]       = {'R','S','E',  ...,  'G'};
char aJeongEumJong[]    = {'ㄱ','ㄴ','ㄷ', ..., 'ㅎ'}
```

3. 정음형 오토마톤: 접수기와 변환기

정음형 코드는 초성자, 중성자, 종성자 코드로 45 자에 불과 하지만 여기에 규칙을 적용하여 생성할 수 있는 음절자는 약 399 억이나 된다. 기본 45 자와 음절자 생성 규칙은 유한하지만 이들은 거의 천학적인 수의 음절자를 생성해 낸다. 정음형 오토마톤 접수기는 입력 자판의 종류에 따라서 변환기로 역할할 수도 있다.

KS C 5715 자모형 (통상적으로 2 벌식) 자판은 자음과 모음을 배치하여 입력 글자를 33 자로 단순화하였다. 하지만 정음형 코드는 자소형이기 때문에 자모가 입력되면 이를 오토마톤에서 음절자의 각 자소를 인식한 정보를 활용하여 정음형 코드로 변환한다. 음절자 구성에 관한 인식은 접수기 오토마톤이 하는 데 접수기는 자모형 자판에서 들어온 자음의 속성을 판단한다. 한글의 음절자에서 자음은 초성이 될 수도 있고 종성이 될 수도 있다. 그런데 첫 음절자의 자음 1 은 그대로 그 음절의 초성이 되지만 모음 다음에 오는 자음 2 는 그 음절의 받침이 되거나 다음 음절의 초성이 된다. 그 결정은 자음 2 다음에 오는 문자가 자음이냐 모음이냐에 따라 결정된다.

자음 1+ 모음 + 자음 2+ 자음 3 또는 모음

이러한 정보를 활용하여 자모형 코드를 자소형 즉 정음형으로 변환하는데 이 기능이 곧 변환기(transducer) 이다.

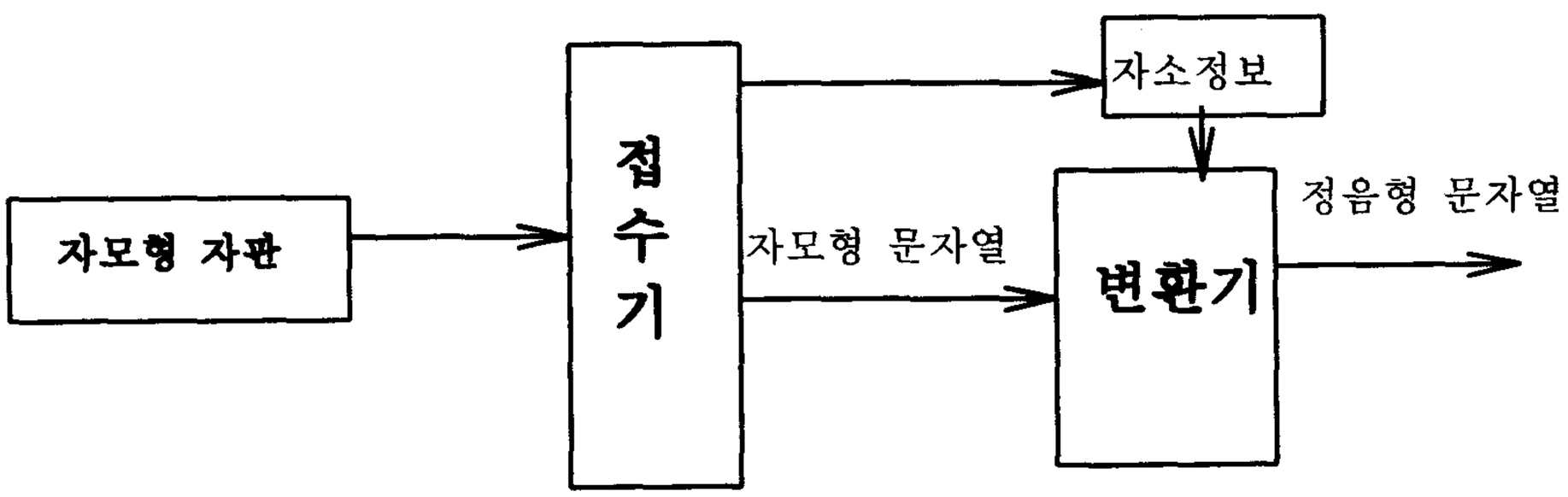


그림 7.2 입력계의 코드 변환기

입력계에는 접수기와 변환기가 있어서 들어오는 문자열의 음절자 인식 여부를 검사하고 또한 그들을 정음형 코드로 변환해 주는 역학을 한다는 점에서 오토마

톤의 복수형인 오토마타로 불러도 좋다.

오토마톤은 이미 2 장에서 보았듯이 밀리형으로 4 상태 천이도로 되어 있으며 이는 하나의 함수로 구현된다. 입력 자료는 표준 자판 KS C 5715 로 부터 입력된 33 자 자모형을 45 자로 된 정음형 코드 값을 반환한다. 이것은 비쥬얼 C++의 뷰 클래스에서 정의된 OnChar() 안에 구현하였다.

4. 윈도우의 에디터 콘트롤

윈도 95 는 기본적으로 완성형을 지원하기 때문에 정음형 코드 구현에서 위와 같은 기능을 지원하려면 에디터 콘트롤이나 리치 텍스트 에디터 콘트롤을 사용하기가 어렵다. 왜냐면 그들은 IME 라는 콘트롤에 의하여 자판의 자모형 코드가 입력되면 이들을 완성형 코드로 변환하여 주는 역할을 담당하기 때문이다. 이러한 문제를 피하기 위하여 좀 더 별도의 정음형 한글 입력 모드를 정의하여 상태의 자판에서 입력하는 영문자의 내용을 그대로 받을 수 있는 상태에서 구현하였다.

주지하는 바와 같이 비쥬얼 C++는 MFC(MicroSoft Foundation Class) 라이브러리와 윈도 API 를 결합한 것이다. 비쥬얼 시 다블 플러스의 통합 개발 환경은 입력된 자료의 ASCII 값을 리턴하는 OnChar() 과 OnKeydown()이라는 View 클래스의 멤버 함수를 WizardApp 를 통하여 쉽게 그 프로토타입을 부여 받는다. 후자는 버츄얼 키값을 반환해 준다. 버츄얼 키 값은 33 자 판 이외에 쉬프트, 알터, 콘트롤 키, 펑션 키와 그들의 조합에 대한 스캔 코드를 리턴한다.

정음형 편집기 '바른글'은 여기서 출발한다. 다시 말해서 리치 에디터 콘트롤을 사용하면 편집기의 기능을 바로 얻을 수 있지만 그것은 이미 완성형 코드를 기반으로 하고 있기 때문에 되지 않는다.

비쥬얼 C++ 프로그램은 4 가지 주요 생성 요소로 되어 있다. 즉 애플리케이션 객체, 메인 윈도 객체, 그리고 뷰 객체이다. 이들은 모두 묶여 있으며 각각은 자체의 기능을 가지고 있다.

윈도우는 사용자의 요구에 대한 메시지를 프로그램에게 메시지로 보내어 연산을 처리한다. 예를 들어서 사용자가 키를 누르면 WM-KEYDOWN 이라는 메시지를 프로그램에 보낸다. 또한 키를 읽을 때 메시지를 읽지 않고 WM_CHAR 이라는 메시지를 가로챈다. 메시지 가로채기는 classwizard 를 이용하여 OnChar()라는

함수 포로토 타입을 호출한다.

```
Void JungEumView::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)
{
        Cview::OnChar(nChar, nRepCnt, nFlags);
}
```

이것은 OnChar()의 프로토 타입이다. 정음형 오토마톤 접수기는 바로 이 안에서 구현되어야 한다. 매개 변수 가운데 nChar 는 ASCII 값을 가지고 있다. 윈도 95 는 기본적으로 영문 모드와 완성형 한글 모드를 가지고 있다. 정음형 에디터 콘트롤은 여기에 정음형 모드를 추가한다. 정음형과 영문 모드는 함수 키 F12 또는 버추얼 키 VK-F12 이다. 그런데 버추얼 키는 OnChar() 함수에서 인식되지 않고, OnKeydown() 함수에서 인식된다. 이 함수에서는 모든 버추얼 키를 처리한다. 예를 들어서 방향 키라든지 알터 키, 콘트롤 키, 쉬프트 키, 함수 키 등을 인식한다. OnKeydown() 함수의 프로토타입은 다음과 같다.

```
Void JungEumView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
  CScrollView:: OnKeyDown(nChar, nRepCnt, nFlags);
 }
```

여기서 정수형 변수인 nChar 에는 버추얼 키 값이 매개 변수로 전달된다. 버추얼 키 값은 부록 나에서 보는 바와 같이 VK_ 접두어로 시작한다. 다음 절에서 다루는 방향 키는 VK-UP, VK-DOWN, VK_LEFT, VK_RIGHT 로 정의되어 있다. 이들을 여기서 switch-case 문으로 각각에 대한 연산을 처리한다. 윈도우에서 사용자가 자판의 제어 키를 누르면 메시지가 바로 이 함수에 전달되고 OnKeyDown() 함수는 이를 처리한다.

2절. 조합 글꼴 개발 및 활용

1. 완성형 코드와 글꼴

현대 한글에서 사용하는 음절자 11172자 가운데 완성형 KS C 5601-1987은 2350 음절자만 포함하고 있다. 그래서 워드 프로세서 시자에선 이를 모두 지원하는 제품을 선호하여 왔다. 따라서 MS 가 측은 이를 다 지원하는 방식을 윈도 95에 적용하려는 노력을 보인 결과는 통합형 코드로 나타났고, 두 바이트 코드계를 채택하고 있는 윈도 95에서 다음과 같이 11172자 가운데 2350자에 포함되지 못한 8822자를 256 x 256 코드표에 할당하여 해결하려고 하였다. 이러한 방식을 취한 것은 유니코드를 국내 표준으로 채택하고 있지 않은 한국의 사용자를 의식한 것이고, 또 다른 하나는 적어도 기존의 표준인 KS C 5601-1987을 사용하고 있는 사용자의 불편을 줄이면서 현대 한글 음절자를 모두 지원한다는 방식을 취한 것이다. 이것은 하나의 모험이었고, 한시적인 조치였다. 하지만 이것은 MS의 상술의 극치를 보여 주는 하나이다. 만약 이러한 통합형 코드를 사용하여 한글 자료가 많이 만들어 진다면 결국 나중에 유니코드가 나왔을 때 불규칙하게 배열된 그 자료를 모두 변환해야 한다. 다른 한편으로 생각해 보면 이것도 큰 문제가 될 것은 없다. 왜냐면 완성형 2350자도 결국은 유니코드의 한글에 대하여 변환을 해야 하기 때문에 누가 좀 수고를 하여 이 프로그램만 준비되면 공통으로 사용할 수 있다.

95년도 전반기부터 통합형 코드로 인하여 불거져 나온 한글 코드 논쟁은 다시 불을 당겨서 한 참을 논쟁의 상태로 있었다. 결국 이 문제에 대하여 정부에서 유니코드를 수용한다는 발표를 함으로써 일단락을 지었다. 이 코드는 1996년 봄에 ISO에서 통과되었고 국내 표준의 이름은 KS C 5700-1995 가 된 것이다.

그러면 글자꼴은 완성형 코드에 대응하여 제공되고 있다. 편집기를 개발하면서 글자꼴을 새로 개발해서 사용하는 것과 완성형 글자꼴을 그대로 사용하는 문제를 검토하였다. 완성형 글자꼴을 그대로 사용할 경우 정음형 코드의 정신을 그대로 살리는 데 제약이 있지만 미려한 글꼴을 활용할 수 있다는 측면에서 활용하기로 하였다. 완성형은 2350 음절자만 활용한다. 통합형인 경우 11172자를 사용할 수는 있지만 2350 자 이외의 글자가 불규칙하게 배정이 되어 있기 때문이다. 그래서 2350 음절자 이외의 글자는 조합형 글꼴을 사용하기로 하였다. 조합형 글꼴이 중심이고 완성형 글꼴은 보조 글꼴이다. 조합형 글꼴 개발을 함에 있어서 다행히도 글자꼴을 개발하는 툴이 제공되고 있으며, 188자에 대한 사용자 구역이 비어

있어서 이를 이용함으로써 시간과 노력을 줄이기로 하였다. 물론 이들 사용자 구역을 이용함에는 완성형 코드를 기본으로 채택하고 있는 MFC 의 환경을 그대로 활용하기 때문에 별도의 다른 노력이 필요하지 않는 것이 이점이다.

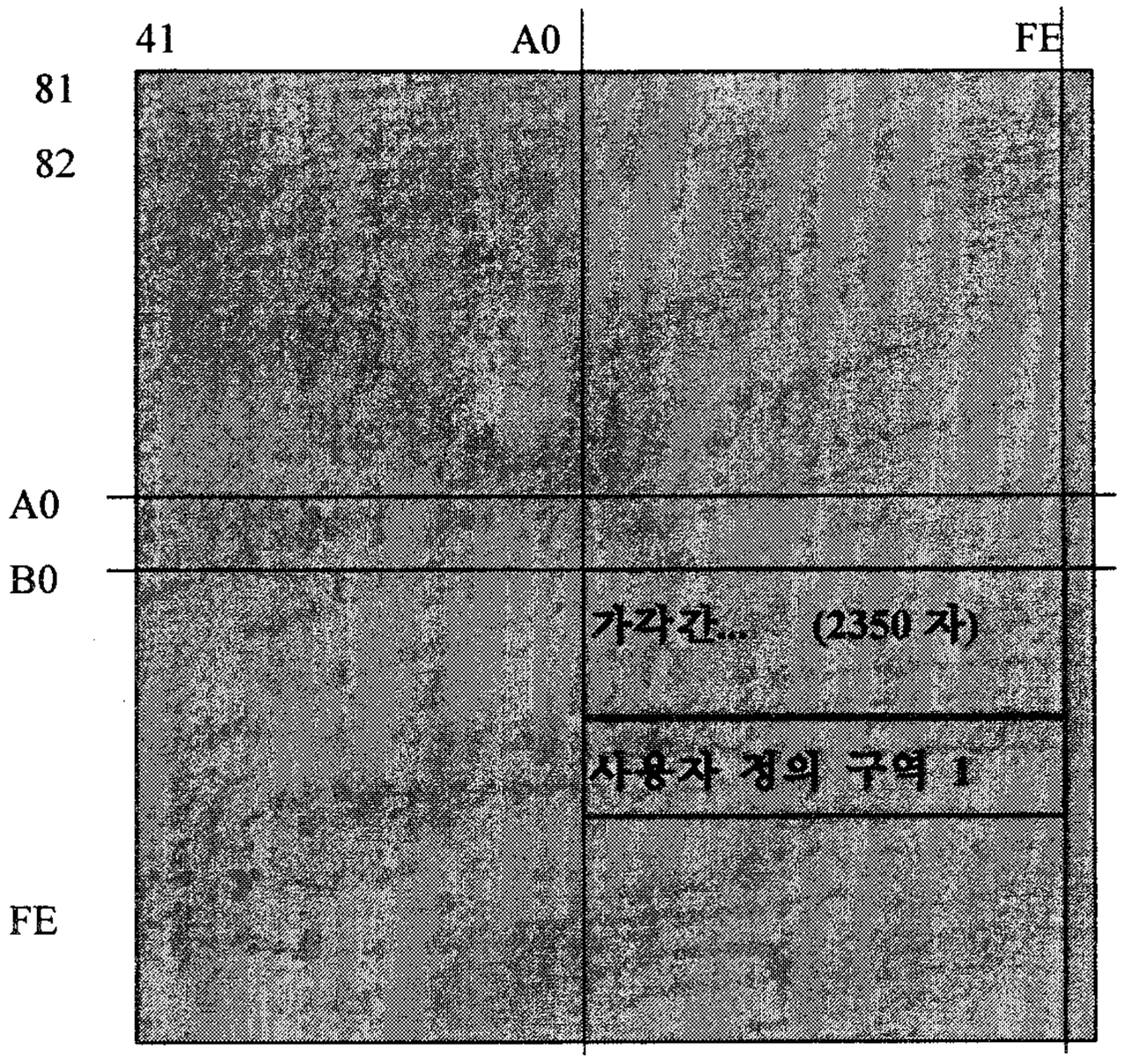


그림 7.3 ISO 2022 에 기반한 완성형 한글 구역

2. 조합 글꼴 개발

정음형 코드는 훈민정음 원리에 따라서 음절자를 조합한다. 음절자를 조합한다는 관점에서 기본적으로 글자를 조합하는 것이 원칙에 충실하므로 조합 폰트를 쓰기로 한다. 현재 윈도 95 에는 완성형 코드에 기반한 완성형 폰트를 바탕으로 하기 때문에 조합형 글꼴을 새로 만들기로 하였다. 여기서 조합형 글꼴은 미려함은 둘째로 하고 우선 조합하는 알고리즘 개발에 초점을 맞추었다.

조합형 글꼴을 사용하기 위하여 앞에서 언급한 대로 윈도 95 에 사용자 정의구역 편집기를 활용하여 EUDC 표준 글꼴의 사용자 구역 1 을 사용하기로 한다. 사용자 구역은 원래 목적이 다르긴 하지만 현재로서 별다른 사용 목적을 가지고 있

지 않다. 왜냐하면 사용자 구역에 배정할 음절자의 범위가 크고 설사 배정을 하였더라도 가나다순 문제가 있기 때문에, 그리고 그것은 표준화에 문제를 일으킬 소지가 크기 때문에 그렇다. 그래서 사용자 구역에 배정할 수 있는 글자꼴의 종류는 94 자이다. 정음형 코드가 요구하는 글자꼴의 범위는 너무 커서 감히 그 모두를 지원한다는 것은 엄두를 낼 수 없다. 그러나 완성형 글꼴을 생각했을 때는 그렇지만 그들을 조합한다면 상당 부분 해결이 가능하다. 훈민정음 해례에서 초성자, 중성자, 종성자가 각각 2, 3 자 조합이 가능하다 하였기 때문에 조직적으로 조합형 글꼴을 만든다면 단순화 시킬 소지가 있다. 하지만 글자꼴의 아름다움은 추구하기가 힘든다.

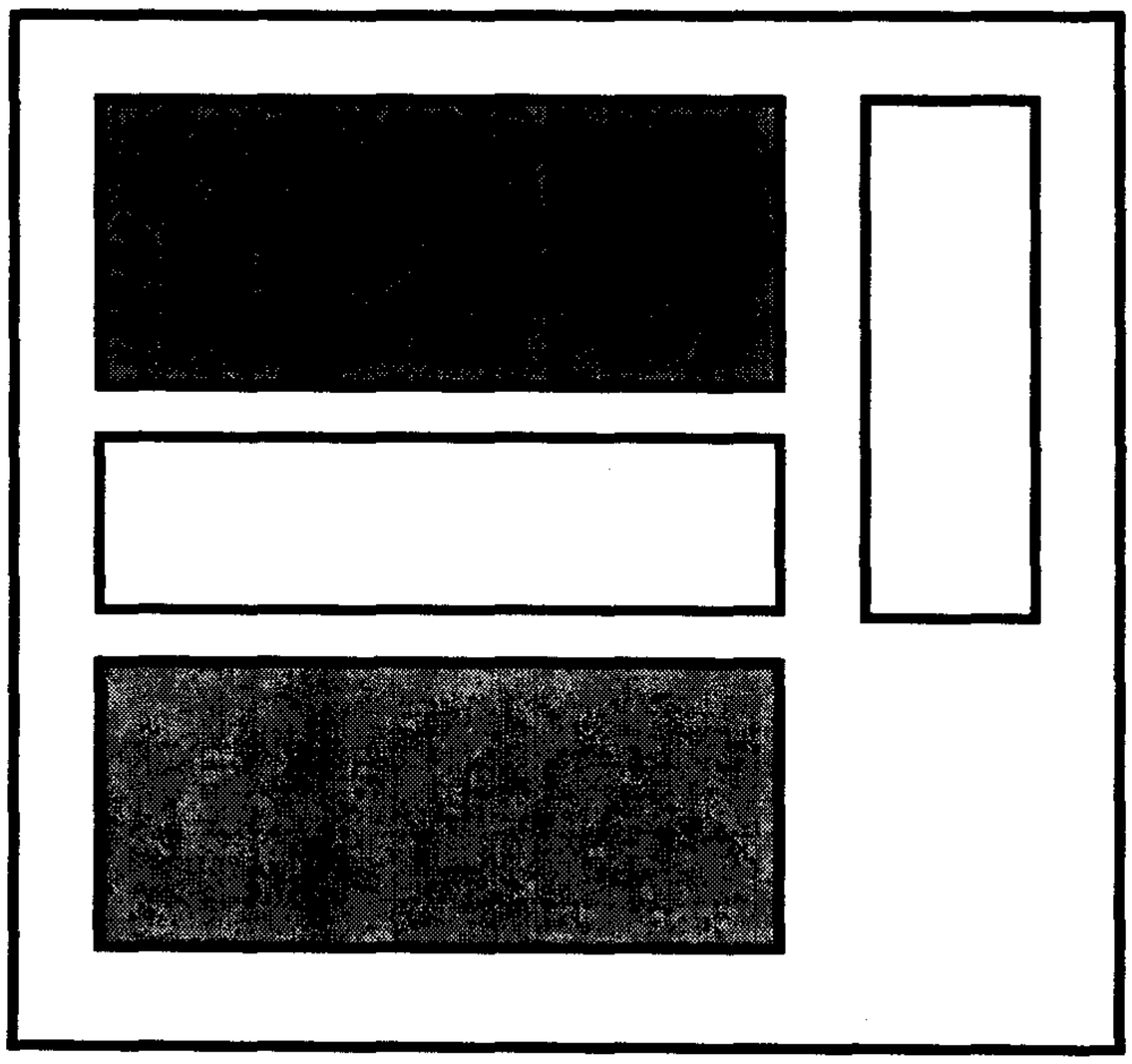


그림 7.4 조합형 글꼴 구조

조합형 글꼴 개발에서 기본 글자 구조는 위와 같다. 위의 구조에서 초성자, 중성자, 종성자는 각각 한 벌식으로 한다. 그래서 가로 모음을 가진 음절자는 세로 모음을 가진 음절자 보다 글자의 폭이 작다. 그리고 받침이 없는 음절자는 다른 글자 보다 작다. 정음형은 옛 한글의 쉽게 지원한다는 장점을 가지고 있다. 지원 방식은 정음형에 현재 쓰지 않는 넉자를 훈민정음 창제 당시와 같은 자격으로 포함시켰다. 그리고 모든 문자열 처리에서 그 자격을 현대 한글과 똑 같은 수준에서 처리하도록 하였다. 석자 조합을 할 경우 초성자의 예를 들면 'ㅺㄱ'의 경우 위의 그림처럼 초성자 글자꼴의 전체 크기를 결정하고 이를 이를 삼등분하여 그 시작 좌표 값을 결정하고 삼등분된 크기에 맞도록 글자를 설계한다. 그리

고 석자 조합을 할 때는 해당 글자꼴을 해당 좌표에 위치시키면 자연히 글자를 조합할 수 있다. 이것은 두자 조합일 때도 똑 같이 적용할 수 있다. 물론 두 자 조합일 경우에 각 자소의 글자꼴의 크기는 석자일 때 보다 크다. 이것도 역시 좌표 값을 만든다면 조합을 할 수 있다. 글자의 아름다움을 위해서는 전문가의 식견에 맡기면 더 좋은 결과가 있을 것으로 본다.

자소의 개발은 기본 45 자를 기본으로 해서 순경음을 비롯하여 두 자 조합이나 석자 조합의 글자꼴을 완성형 형태로 일단 개발을 하였다. 앞에서 언급한 각 자소의 조합은 궁극적으로 해결해야 할 과제로 보고 여기서는 그 가능성만을 제시한다. 그렇다고 한글의 2 자 또는 3 자 조합 자소들을 모두 개발하는 것도 불가능하다. 그래서 일단 그 범위를 훈민정음 창제 이후에 사용한 적이 있는 자소 240 자로 하였다. 하지만 이것 역시 모두 포함이 불가능하기 때문에 합용병서된 모음은 현대 한글에서 사용하고 있는 것으로만 한정하였다. 그리고 두자 조합은 모두 포함하고 석자 조합은 특별한 기준없이 일부를 포함시켰다.

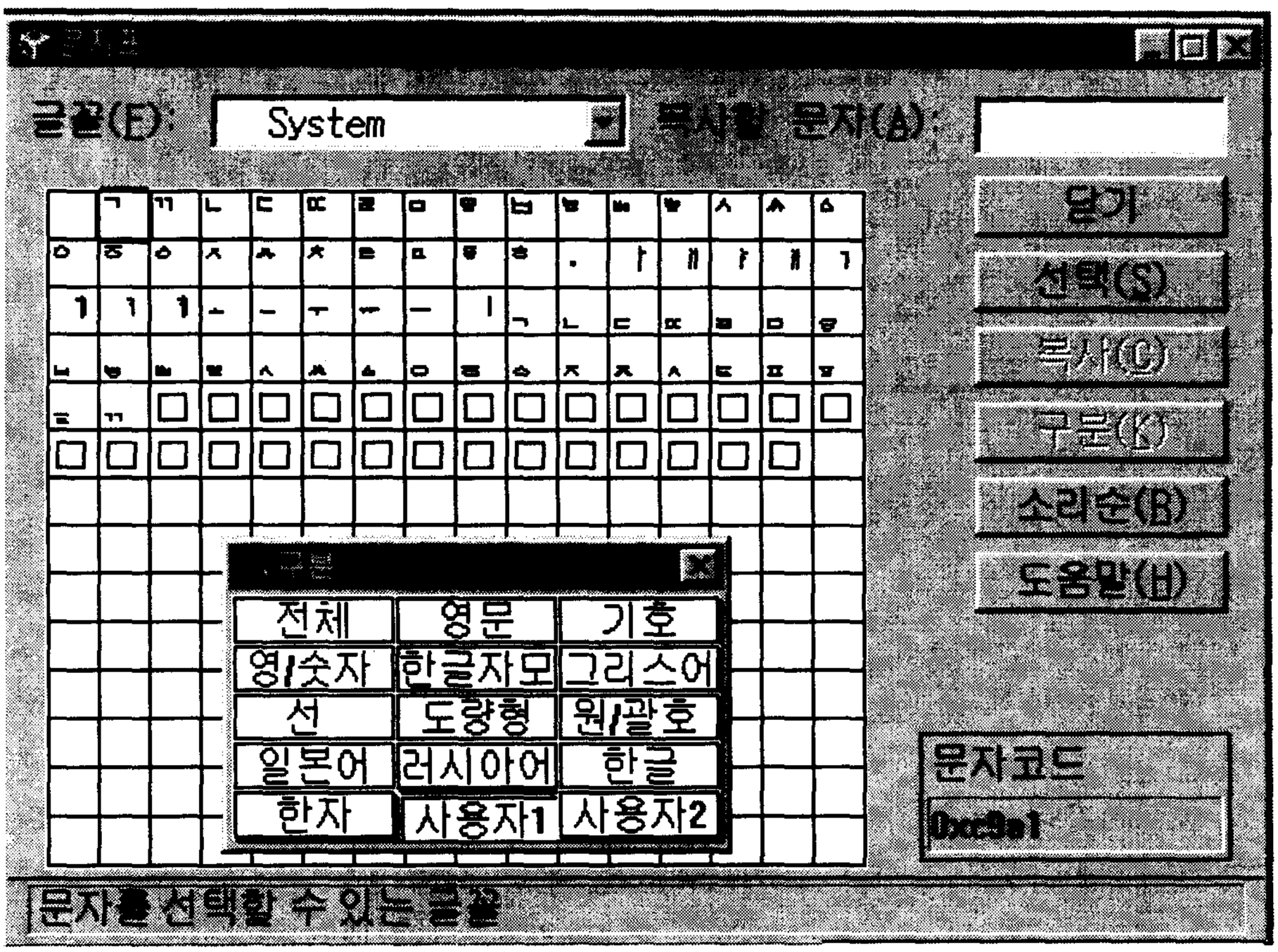


그림 7.5 문자표

본 연구의 목표는 일단 정음형 윈도우 콘트롤을 만드는 것이기 때문에 그것의 방법론을 개발하고 프로토타입을 개발하는 데 주력하였다. 다음은 문자표와 사용자 정의 구역 편집기의 모습이며, 글자를 개발하는 일면이다.

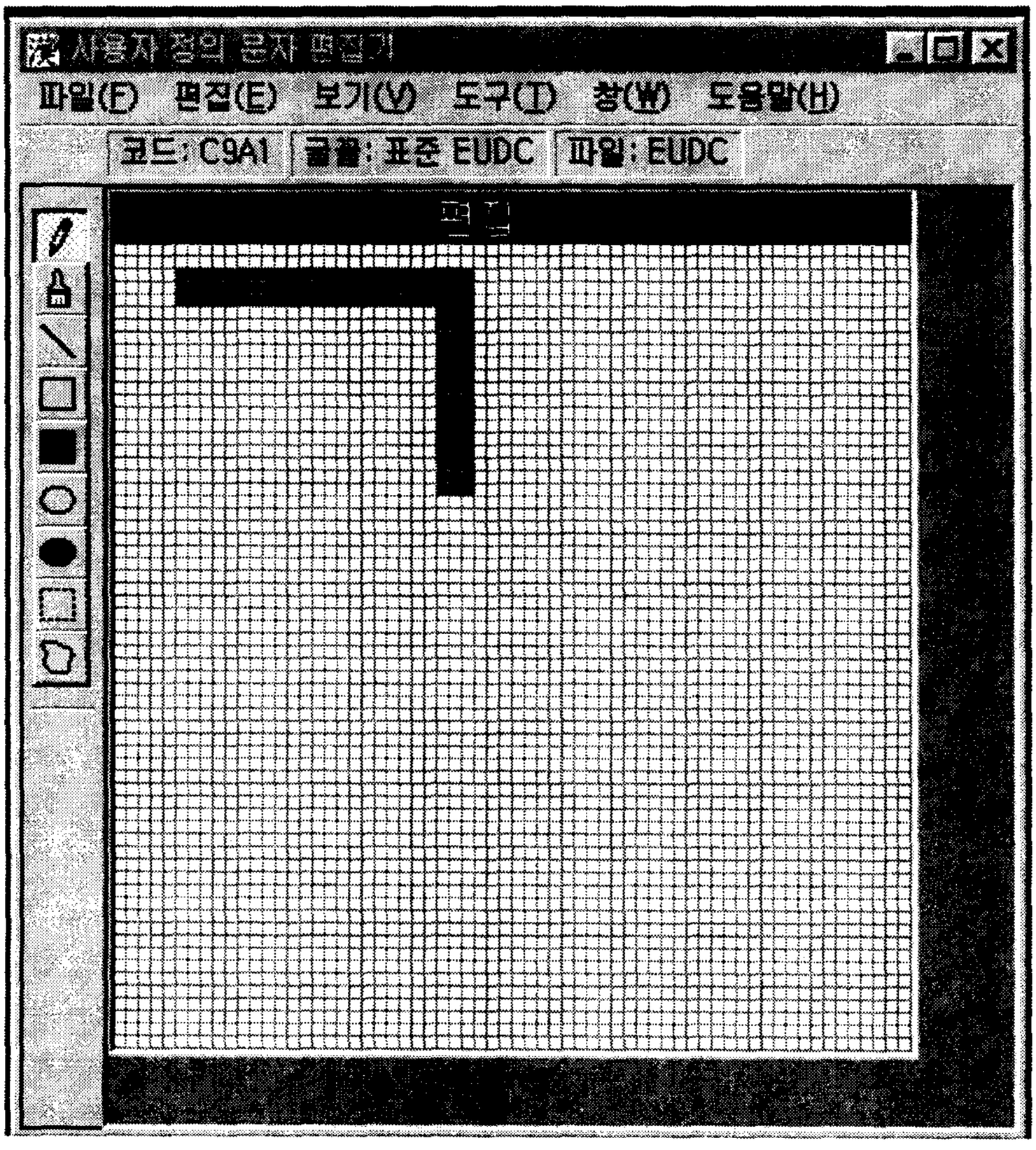


그림 7.6 사용자 정의 구역 편집기

3. 조합형 글꼴의 활용

비쥬얼 C++에서 뷰 클래스에서 글꼴의 출력에 관계되는 멤버 함수가 있다. OnDraw() 함수는 디스플레이 콘텍스(DC) 의 포인터를 매개 변수로 받아서 텍스트 버퍼가 있는 문서(document)의 접근 포인트를 활용해서 문서의 코드에 대응하는 글자꼴을 화면의 적절한 위치에 뿌려 준다.

```
Void CJeongEumView::OnDraw(CDC* pDC)
{
        CJeongEumDoc* pDoc = GetDocument();
        ...
      pDC->TextOut(CaretPos.x, CaretPos.y, HanFont[k]);
        ...
    SetCaretPos(CaretPos);
    }
```

화면에 글자꼴을 뿌린다는 것은 실제로 매개 변수인 CDC* pDC 가 가리키는 곳이 바로 화면 버퍼이다. 화면 버퍼는 화소로 구성된 출력 공간이다. 여기에 문서의 내용을 적절하게 출력한다. 한글은 글자폭이 32 이고 영문은 16 이다. 그런데 정음형은 조합형 글자꼴을 사용해서 출력하기 때문에 예를 들어서 '각'은 'ㄱ ㅏ ㄱ'을 출력한다. 그런데 조합형 글꼴은 자소 각각으로 디자인이 되었기 때문에 이들을 모두 겹쳐 쓰기를 한다. 디스플레이 콘텍스 클래스의 멤버 함수인 SetBkMode() 함수를 하여 해결하였다. 이 함수는 매개변수로서 글자의 배경을 투명하게 할 것인가 불투명하게 할 것인가를 결정한다. 불투명하게 하는 값은 OPAQUE 이고, 투명하게 하는 값은 TRANSPARENT 이다. 글자를 디스플레이 콘텍스트에 뿌리는 절차는 다음과 같다.

1) 먼저 커서를 해당 위치에 놓는다.
2) 글자의 배경색 모드를 TRANSAPARENT 로 한다.
3) 문서에서 출력할 글자 코드 값을 얻는다.
4) 초성자를 TextOut() 함수를 통하여 출력한다.
5) 중성자와 종성자를 차례로 출력한다.

이를 비쥬얼 시 플러스플러스로 표현하면 다음과 같다.

```
CJeongEumDoc PDoc = GetDocument();
...
SetCaretPos(x,y);
pDC->SetBkMode(TRANSPARENT);
...
```

```
pDC->TextOut(CaretPos.x, CaretPos.y, sCho);
pDC->TextOut(CaretPos.x, CaretPos.y, sJoong);
pOC->TextOut(CaretPos.x, CaretPos.y, sJong);
...
```

정음형 코드는 자모형처럼 자소에 대한 모호성이 없기 때문에 자소 문자열로 써 해당 폰트를 찾아서 곧 바로 출력한다. 여기서 우리가 고려할 것은 자료를 단순히 입력할 때와 편집을 할 때 조금 차이가 있다. 윈도가 보여 주는 내용은 문서의 일 부분이다. 그리고 자료의 입력과 편집에서 수정되는 부분은 일 부분이다. 처음 편집기를 고려할 때 화면에 사상되는 버퍼의 내용을 전부 출력하는 방식을 고려해 볼 수 있다. 이것은 가장 간단한 방법이다. 하지만 화면의 일부가 수정되었음에도 이것을 계속해서 화면 전체를 새로 출력해야 한다는 것은 문제가 있다. 그래서 수정된 부분만을 고치는 방법을 고려한다. 그것은 명령에 따라서 다음과 같다.

1) 새로운 줄에 또는 줄의 맨 뒤에 입력을 하고 있으면 그 줄이나 입력된 문자 가운데 마지막 글자 다음의 캣럿 값을 얻어서 해당 글자만 출력하면 된다.
2) 하나의 줄이 추가되었으면 줄이 추가된 그 아래 부분은 전부 새로 출력한다.
3) 특정한 줄의 가운데 추가하였으면 입력된 글자 이후의 모든 글자를 새로 출력한다.
4) 스크롤은 편집 콘트롤의 기능에 의존한다.

과거 덤 터미널의 마이크로 이맥스의 경우 제임스 고슬링이 만든 화면 버퍼의 수정을 최소화하는 고슬링 알고리즘이 있었다. 현재 피시가 예전 보다 속도 높아져서 별반 요구는 없지만 처리 시간을 최소화 하는 것은 우리의 궁극적인 목표이다. 디스플레이 콘텍스 수정을 최소화하는 것이 또 하나의 목표이다.

4. 옛 한글의 입력

현재 한글 편집기나 워드 프로세서에서 옛 한글은 마치 특수 문자로써 취급하고 있다. 그러나 옛 한글은 1930 년도에 조선어학회에서 조선어 맞춤법 통일안을 만들면서 지금 옛 한글 자모라는 그 넉자를 제외시켰다.

그림 7.7 옛 한글 키 할당

나머지 24 자와 차별하지 않고 훈민정음 창제 당시의 28 자의 대접을 하고 있다. 이것을 가능하게 하는 것은 컴퓨터 내부에서 음소 문자를 구현하고 컴퓨터 외부에 보여질 때 2 차원의 음절문자 특성을 구현하기 때문이다. 다시 말하면 문서 상태로 있을 때는 1 차원의 풀어 쓰기 상태로 뷰로 보일 때는 2 차원의 모아쓰기 형태로 변경한다. 그러면서도 코드화 대상은 완벽한 낱자소 하였고, 훈민정음 해례에서 정의한 대로 2 또는 3 자 합용 병서를 하도록 허용하였다. 그러다 보니 입력할 때도 자판에서 아무 것도 할당되어 있지 않은 쉬프트 문자 부분에 각각 할당하였다.

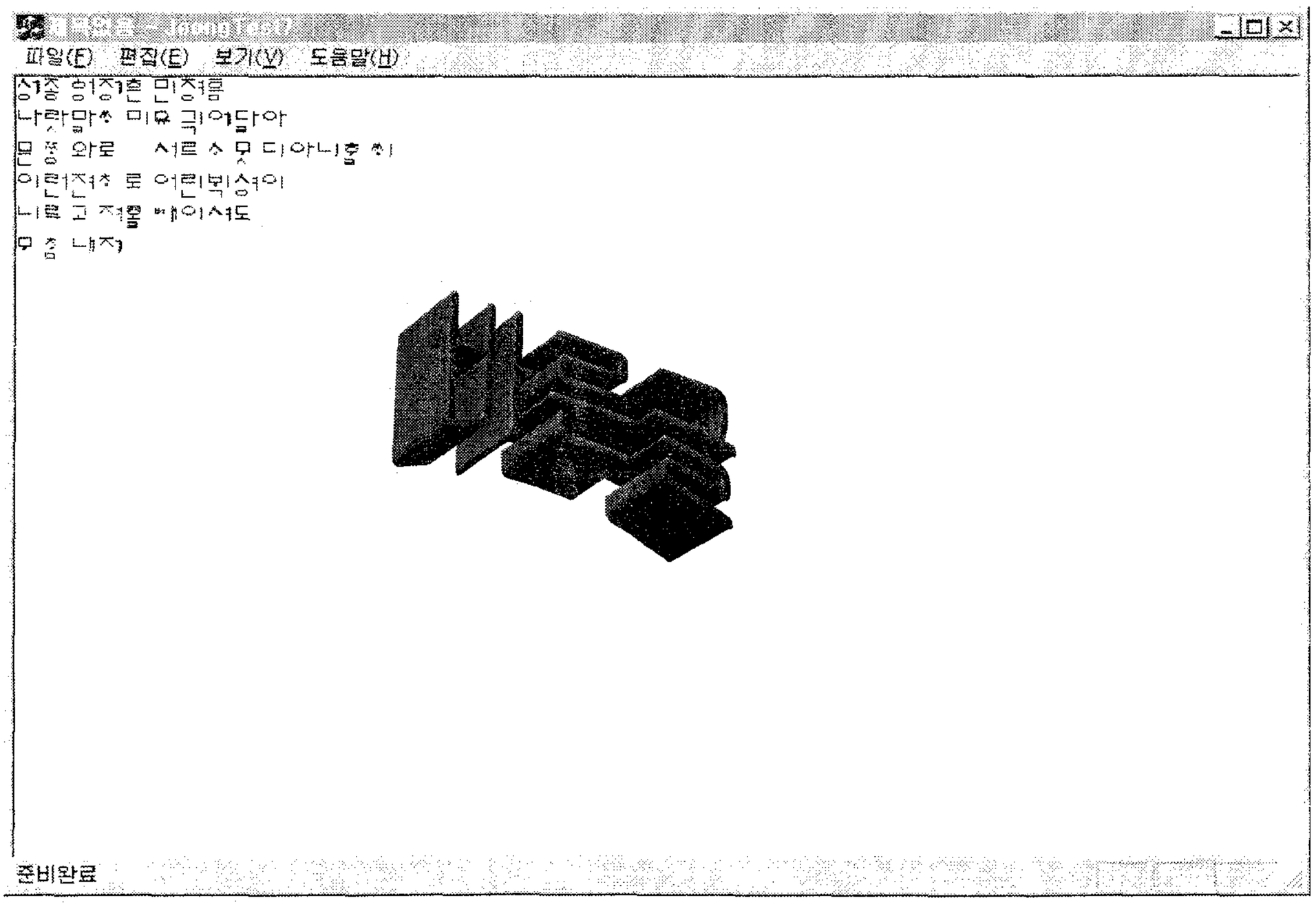


그림 7.8 '바른글' 화면에 표현된 옛글

옛 한글의 키를 위의 그림과 같이 배정하였다. 이렇게 배정한 것은 비슷한 음가를 가진 것과 위치 등을 고려한 것이다. 그런데 옛한글의 경우 현재 자모형 자판으로 입력을 할 경우에 입력에서 음절 끊기에 문제가 있다. 그래서 구분자로서 사용하도록 쉬프트 엘(SHIFT L)을 배정하였다. 옛 한글의 경우 중성인 경우는 문제가 없지만 자모형 자판에서 초성과 종성인 경우 이들이 3 개 까지 합용 병서 할 수 있기 때문에 앞 음절의 종성과 뒤 음절의 초성 문자들의 구분이 입력하는 과정에서 필요하다. 그래서 이들을 구분해서 넣기 위하여 구분자 키를 특별히 할당하였다. 그러면 글자꼴이 지원되는 한 옛 한글은 모두 표현이 가능하다. 그리고 우리가 이상적으로 생각하는 것은 컴퓨터는 우리의 연필과 종이를 대신할 도구이기 때문에 컴퓨터가 우리의 사고를 제약하지 않고 도리어 사고를 확장시켜주는 역할을 하여야 한다. 그런 점에서 "마치 연필로 글을 쓰듯이 ..." 를 구현할 수 있어야 한다. 바로 '바른글' 편집기는 이와 같은 역할을 할 수 있다.

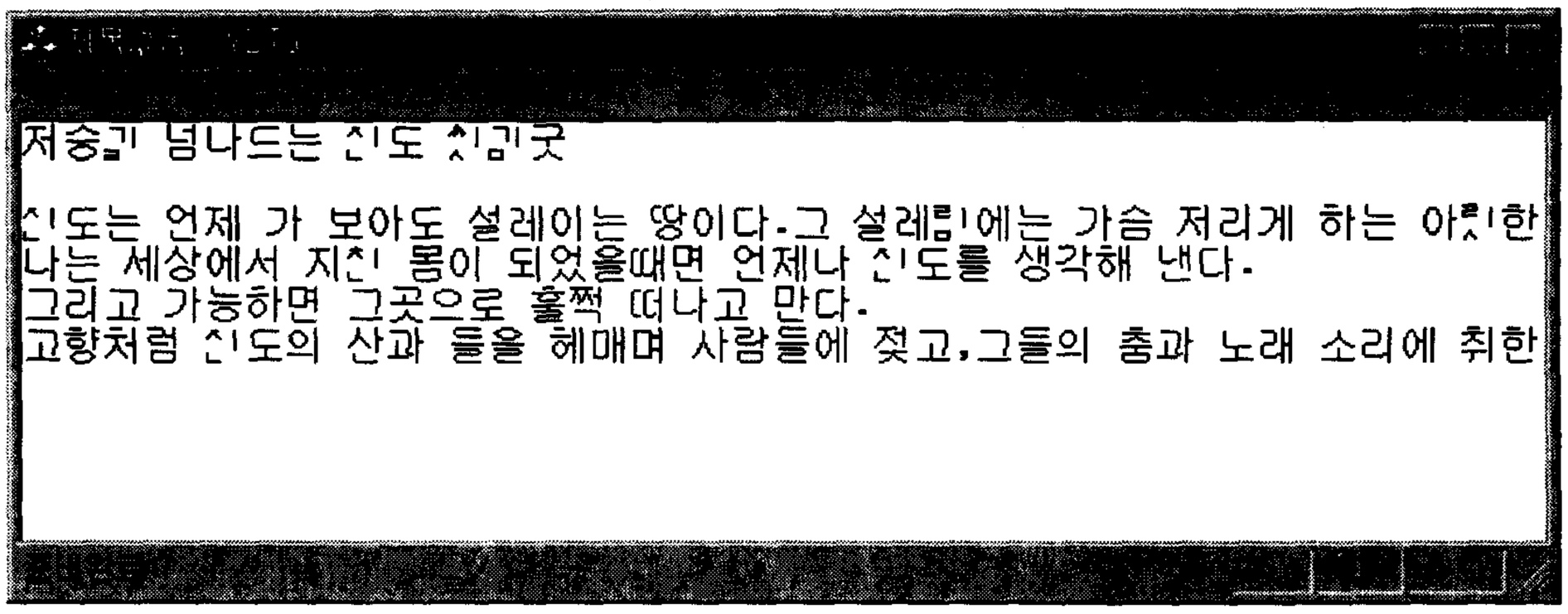


그림 7.9 완성형 글꼴과 조합형 글꼴의 혼합 사용 예

3 절. 완성형 글꼴의 사상

1. 정음형-완성형 사상 표

바른글 편집기에서 글꼴은 기본적으로 조합형 글꼴로 하기 때문에 이것을 비전문가가 개발하였기 때문에 미려한 글씨가 될 수 없이 조잡하다. 그런 측면에서 완성형 글꼴을 사용할 방안을 생각하였다. 현재 우리가 사용할 대부분의 글자가 현대 한글일 것이라는 점에서 완성형 글꼴을 사용하면 일반적인 문서에서 거의

해결이 될 것이다.

정음형 코드를 완성형에 사상시키는 함수는 별도로 개발하여야 한다. 이것은 3장에서 사용하였던 정음형-완성형 간 상호 변환 표를 이용하여 만들 수 있다. 상호 변환표에서 2350 자에 해당하는 완성형 각 음절이 정음형의 어떠한 문자열로 구성되었는지를 배열로 표현하였다. 이 배열은 세로 인덱스가 2350 이고 가로 인덱스가 7 이다. 즉 한 음절은 최고 6 자소로 구성되었다는 데 마지막에는 널(NULL)로 채웠다.

```
JE_WS_TBL[2350][7] = { {0xb1,0x00,0x00,0x00,0x00,0x00,0x00},
                       {0xb1,0x01,0x00,0x00,0x00,0x00,0x00},
                        ...
                        {0xc1,0xcf,0xe2,0x00,0x00,0x00,0x00}
                       }
```

정음형을 완성형으로 변환하는 방법은 먼저 정음형의 각 자소를 널로 끝나는 문자열로 해서 패턴 매치를 한다. 그래서 완성형 2350 음절 가운데 하나가 일치하면 이 때 얻어지는 값은 0 에서 2349 이다. 이것은 다음과 같이 계산하여 완성형 코드 값을 얻을 수 있다.

1) 그 값을 94 로 나눈다.
2) 이 때 얻은 나머지와 몫을 가지고 계산한다.
   (1) 몫은 완성형의 앞 바이트를 결정하는 값이 된다.
   (2) 나머지는 완성형의 뒤 바이트를 결정하는 값이 된다.
   (3) 앞 바이트의 범위는 b0 에서 c8 까지이다.
   (4) 뒤 바이트의 범위는 a1 에서 ff 까지이다.

예) 96 이면 몫이 1 이고 나머지가 2 이다. 완성형 값은 0xb0a2 이다.

2. 없는 글꼴의 표현

다만 옛 한글과 같이 완성형에 없는 글꼴은 조합형 글꼴을 그대로 사용하도록 한다. 이것은 정음형 코드를 글꼴로 사상시키는 함수를 훨씬 더 복잡하게 만든다.

그래서 먼저 앞에서 만든 사상 함수를 활용하여 글꼴을 찾아 보고 없으면 조합형 글꼴 함수를 실행하여 찾도록 한다. 그렇기 때문에 여기선 별도의 함수를 사용하지 않아도 좋다. 단지 양쪽에 모두 없는 글꼴에 대한 처리의 원칙이 정해져야 한다. 그럴 경우에는 가위표 도형을 해당 위치에 표현하여 글꼴이 없을 사용자에게 알리도록 한다.

그림 7.10 없는 글꼴

3. 저장과 출력

정음형 자료를 저장할 때 별도의 다른 처리는 없다. 문서 클래스는 화일에서 자료를 읽거나 화일로 자료를 출력하는 일을 담당한다. 그래서 멤버 함수의 지원을 받는다. 여기서 정음형인 경우 확장자를 자동으로 붙여 주는 역할을 한다. 정음형의 확장자는 JEU 또는 jeu 이다. 다른 자료와 썩이지 않도록 하기 위하여 확장자로 명확하게 하고 파일을 마우스로 드래그 했을 때 자동으로 '바른글' 편집기가 뜨도록 해야 한다.

8장. 결론

국어 정보처리에 최적한 한글 코드를 개발하고, 그 코드를 기반으로 한 입력 도구 및 관련 환경을 개인용 컴퓨터(PC)의 중심 운영체제인 마이크로 소프트의 윈도 95 상에서 개발하였다. 최적한 한글 코드의 개발은 이미 1, 2 차년도 연구 결과를 활용하는 측면이 컸다. 그래서 부가적인 연구는 없었지만 본 보고서에서는 그것을 재검토하였다. 본 연구의 핵심은 윈도 95 환경에 맞는 정음형 코드를 기반으로 하는 국어 정보처리 기반 환경을 개발하는 것이다. 따라서 윈도 95 의 기본 프로그래밍 환경인 WIN32 API 를 활용하고 MFC 을 바탕으로 하는 비주얼 C++를 개발 언어로 삼고서 정음형 코드를 기본 코드계로 하는 정음형 편집기 "바른글"를 개발하였다. 그리고 기존의 한글 코드와 호환성을 위하여 통합 코드 변환기를 개발하였고, 이것을 편집기의 입출력 환경에 부착하였다. 또한 조만간

유니코드가 표준 코드로 될 것이므로 유니코드를 수용한 국내 표준인 KS C 5700-1995 구현상 문제점을 지적하고 정음형 코드가 그 문제를 해결할 수 있음을 보였다. 뿐만 아니라 정음형 코드를 유니코드에 적용할 때 유니코드의 어느 위치에 놓을 수 있으며 기존의 코드는 없애도 된다는 실증적인 증거를 보였다.

윈도 95 는 ISO 646 과 ISO 2022 를 기반 코드 체계로 하여 개발되었기 때문에 정음형을 구현함에는 제약이 많았다. 우선 한글/영문 전환 키에는 KS C 5601-1987 완성형 한글 코드가 할당되어 있기 때문에 정음형 코드 모드는 별도의 키를 할당해야 했다. 물론 이 키를 정음형 코드에서 슬리킹할 수는 있지만 API 수준에서 지원하기 때문에 일단 그대로 두고 F12 를 할당하였다. 이와 같은 문제를 비롯하여 기존하는 완성형 환경을 그대로 인정하면서 정음형은 영문 모드에서 별도의 모드를 활용하였다. 그리고 글자꼴은 일단 완성형 코드 표에서 사용자 정의 구역의 제 1 구역 94 자에 정음형의 주요 글자꼴을 배정하였다. 여기서 정음형 글자꼴은 초성자, 중성자, 종성자로 된 세벌식으로 설계하였으며, 여기서 글자의 미려함은 차순으로 하였다. 물론 선택적으로 글자꼴의 미려함을 고려하여 완성형 글자꼴을 일차적으로 매치하여 쓰고 없는 글자에 한하여 세벌식 합성 글자꼴을 쓸 있도록 하였다. 비주얼 C++에서 문서-뷰 모델은 정음형 코드를 활용한 편집기를 개발하기에 편리한 환경을 제공하였다. 그러나 문서, 뷰 클래스들은 모두 정음형과 같은 완전 풀어 쓰기 형태의 음절 문자열을 처리하도록 되어 있지 못하여 관련 멤버 함수를 추가로 개발하여 사용하였다.

'바른글' 편집기를 활용하면 국어 정보처리에서 필요로 하는 형태소 정보를 비롯한 세세한 언어 정보들을 담을 수 있다. 신기하게도 코드화 대상의 문자 집합이 적을 수록 한글 코드는 더욱 더 많은 언어 정보를 담을 수 있다. 바른글 편집기는 한글 문자 처리 뿐만 아니라 국어 정보처리에 필요한 자료를 입력하는 가장 강력한 도구이다. 또한 정음형 코드를 활용하는 각 종의 프로그램을 작성할 수 있다. 편집기를 개발하는 과정에서 정음형 한글 처리와 관련되 파생클래스와 멤버 함수들을 라이브러리로 구성하여 제공함으로써 국어 정보처리 기술 개발에 기초 환경을 제공하여 개발에 경제성을 제공한다.

정음형 코드를 윈도 95 환경에 개발하는 것은 편법이다. 그것은 윈도 95 가 기본적으로 지원하고 있는 완성형과 정음형은 코드 체계 자체가 확연하게 다르기 때문이다. 완성형은 음절을 코드화 대상으로 하였고, 2 바이트를 코드의 기본 체계로 하고 있다. 반면에 정음형은 자소를 코드화 대상으로 하였으며, 일 바이트

체계이다. 그리고 완성형은 모든 음절이 이 바이트로 표기되지만 정음형은 한 음절을 구성하는 문자열의 길이가 가변적이다. 완성형의 절대적인 결함은 자소 정보를 가지고 있지 못하여 국어 정보처리에서 사용할 수 없다는 점이고, 정음형은 이러한 모든 점을 가지고 있는 이상적인 코드이긴 하나 국가 표준 코드가 아니기에 기업들이 이러한 코드를 활용하여 본격적인 소프트웨어 개발을 못하고 있다는 현실이다. 이러한 완성형의 문제점에 대하여 유니코드에서 이러한 문제가 해결될 수 있다고 하나 유니코드는 표준에 역행하는 요소들을 또한 가지고 있다. 그것은 세 가지 종류의 코드 즉 자모형, 자소형, 완성형을 모두 포함하고 있기 때문이다. 현대 한글 표기법이 규정한 자모의 집합과 훈민정음 해례가 규정한 모아 쓰기에 의하여 조합되는 11172 음절자를 완성형 코드로 하여 가나다순으로 유니코드에 배정하였다. 자음과 모음에 대한 코드와 훈민정음 창제 이래 사용한 자소들을 조사하여 240 자소를 또한 포함시켰다. 국제 표준위원회에서 우리의 이러한 요구를 관철시킨 것은 우리의 힘이다. 하지만 상대적으로 한글의 우수성 특히 과학적인 문자라는 점에 먹칠을 한 것이다. 이것은 한자의 수준으로 한글을 격하시킨 것이다. 이것은 한글 정보처리 소프트웨어가 단순 문자 처리 수준에 있는 우리의 현실을 입증하는 것이며, 또한 단견적인 국내외 소프트웨어 업자들의 이익에 편승한 결과이다. 현재 통상부의 품질관리원에 소속되어 있는 해당 위원회의 구성원 가운데 이러한 정보처리 업자들의 입김이 상당히 크게 작용하고 있다. 특히 마이크로 소프트의 힘은 절대적이다. 국내의 소프트웨어 관련 기업이라 하더라도 현실적으로 윈도 95 를 중심으로 소프트웨어를 개발하여야 하기 때문에 어쩔 수 없을 것이다. 이러한 현실에 의하여 결정된 유니코드에 포함된 한글 코드는 어려운 과정을 거쳐서 유니코드에서 한자 다음의 큰 영역을 차지한 만큼 이를 우리가 스스로 포기하는 것은 쉽지 않을 것이다. 하지만 우리의 현실을 볼 때 국어 정보처리에서 완성형을 사용하여 정보처리를 할 수 없다는 사실이 존재하는 만큼 큰 일이 아닐 수 없다. 우선 이 문제를 해결하려면 품질 기술원에 해당 위원회 구성을 새로이 해야 한다. 국어 정보처리는 분야가 넓고 다양하다. 그러므로 이러한 문제를 다각적으로 검토할 수 있도록 위원회를 구성해야 한다. 지금 처럼 정부의 안에 항상 긍정적으로 동조할 위원들로는 정보화 시대의 국가의 이익을 충분하게 담아 내는 결정을 할 수 없을 것이다. 특히 이 위원회를 구성하면서 기업의 대표는 국어 정보처리의 한 분야 이하로 그 구성원의 수를 줄여야 한다. 국가의 어떠한 결정이 특정 기업의 이익에 직결되어서는 안되기 때문이다. 그 기업이 국내 기업이어도 그러할텐데 외국 기업이라면 우리의 문화와 정신적 피해를 감수하면서 그들에게 이익을 주어야 할 이유는 없다. 유

니코드는 올바른 코드이다. 하지만 유니코드에 포함된 한글 코드는 한글의 특성과 한글 및 국어정보처리 기술 개발에 나쁜 방향으로 결정되어 있다. 유니코드의 장점과 그 속에 포함된 한글 코드의 단점을 구별해야 하며 국어 정보처리에서 발생할 문제를 호도하는 설명에 현혹되어서는 안된다. 한글은 우리의 문화이며 한글이 가진 과학성은 닥아올 21 세기 첨단 정보화 사회에 대비하여 우리의 조상이 물려준 값진 유산이며 국제 경제사회의 경쟁에서 우리의 경제적 입지를 넓혀줄 값진 무기이다. 정부는 이러한 점에 유의하여 KS C 5700-1995 의 문제점을 검증하고 이를 개선하는 정책을 추진해야 할 것이다.

우리는 한글을 위한 컴퓨터을 개발할 것인지 컴퓨터를 위한 한글을 만들 것인지에는 이론이 없을 것으로 본다. 하지만 우리의 현재 입장은 우리의 기대와 상식과는 반대로 컴퓨터를 위한 한글의 방향으로 진행하여 왔으며 앞으로 그렇게 가도록 예정되어 있다. 이러한 과정에서 우리가 간과한 것은 공학을 위한 과학을 검증하지 않았던 점이다. 올바른 공학은 과학의 바탕에서 이루어 질 수 있다. 과학이 없는 공학은 사상누각에 다를 바 없다. 한글의 컴퓨터 공학은 한글의 과학성과 컴퓨터의 과학성이 무엇인가를 도출하고 이들 양자간의 과학성의 부합 여부를 검증하고 상호 적응화 과정을 통하여 원만한 공학을 낳을 수 있다. 하지만 이제까지 우리는 그것에 관심을 갖지 못하고 있다. 컴퓨터에서 과학의 존재는 인정하면서 한글의 과학성은 부분적인 발견에서 머물고 있다. 한글의 과학성을 바탕으로 한글 공학화를 추진해야 한다. 과학이 무시된 공학은 시행착오를 거듭할 뿐이다. 이제 그 시학 착오의 긴 터널에서 벗어나야 하며 무지에 비롯되는 시행 착오의 고리를 끊어야 한다. 본 연구 결과가 이러한 사업에 대한 하나의 씨알이 될 것으로 믿는다.

참고문헌

[1] 변정용, 한글 정보처리를 위한 표준 라이브러리 개발, 한국과학기술원, 1996

[2] 이동휘 역, 고급사용자를 위한 ADVANCED VISUAL C++ 4, 삼각형, 1997

[3] 한기용, 한번더 생각한 비주얼 C++와 MFC 4.0, 도서출판 대림, 1996

[4] 유석, MFC 사용자를 위한 비주얼 C++ 4.2 프로그래밍, 정보문화사, 1997

[5] 전병선, 비주얼 C++ 4.0 MFC 윈도우 95 프로그래밍, 심양출판사, 1997

[6] 강신항, 훈민정음연구, 성균관대학교출판부, 1990

[7] 문영호, "국제표준글자부호계에 등록할 조선글자체계의 자모순서문제에 대하여", '96 코리언 컴퓨터 처리 국제학술대회 논문집, 1996

[8] 최병수, "콤퓨터 조선글 부호체계에 넣어야 할 옛글자에 대한 연구", '96 코리언 컴퓨터 처리 국제학술대회 논문집, 1996

[9] 변정용, "훈민정음 원리에 따르는 우리글 코드 제정 방향", '96 코리언 컴퓨터 처리 국제 학술대회, 1996

[10] 합의문, '96 Korean 컴퓨터처리 국제 학술대회,1996

[11] 변정용, 한글정보처리를 위한 표준 입력 라이브러리 연구, 한국과학기술원 보고서, 1996

[12] 한국표준협회, KS C 5700-1995

[13] 조선인민민주주의공화국 국규 9566-93

[14] 변정용,"훈민정음 창제 원리의 공학화에 기반한 한글 부호계의 발전 방향", 한국정보과학회지, V12, N2, 1994

[15] 변정용.강진곤, "한글문자의 음소 및 음절 문자 특성의 구현 방안", 제 6 회 한국어정보처리 학술대회,1994

[16] 강진곤.함경수.변정용, "훈민정음이 살아 있는 hunterm", 한국정보과학회 '95 봄 학술대회, V22, N1, 1995

여백

10. 확장품사사전규칙과 보급패키지

전북대학교
안동언

여 백

10. 확장품사사전규칙과 보급패키지

1장. 서론

1절. 연구의 배경 및 필요성

국어정보베이스는 한국어의 체계적인 연구를 위한 언어 정보 및 기반 기술을 연구하고 개발함으로써 컴퓨터를 이용한 한국어의 처리와 한국어의 기초 연구를 촉진하고자 하는 목적을 가지고 기초 자료의 축적과 기반 연구를 일관된 환경에서 집중적으로 수행하기 위한 연구 모델이다. 따라서, 세부 과제에서 개발되는 시스템들은 각각의 결과물을 서로 이용하고 공유할 수 있어야 하며 구축된 지식베이스를 같이 사용하여야 한다.

이를 위해서는 우선 각 세부과제에서 사용하고 있는 지식베이스의 각종 태그가 표준화되어야 한다. 태그에 있어서 가장 기본이 되는 것은 품사 태그이다. 2차년도의 연구에서는 이 품사 태그를 "국어 형태·통사 태그 규격"이라는 이름으로 표준안을 마련하여 "1996년도 제1회 우리말 정보처리 규격 심포지움"에서 발표하였다. 이 태그는 주로 코퍼스를 자동으로 태깅하기 위해서 만들어진 것이다. 그러므로 이 태그를 형태소 해석이나 생성에 이용하기 위해서는 확장되어야 한다.

품사론은 문장론과 더불어 문법의 근간을 이루고 있다. 더구나 문장론도 품사론을 기반으로 연구된다. 이처럼 품사에 대한 연구는 한국어 연구의 가장 중요한 부분이며, 한국어정보처리의 출발점이 된다. 즉 형태소 해석/생성, 구문 해석/생성을 하는데 있어서 품사 정보는 올바른 해석/생성을 위한 가장 중요한 정보이다.

품사는 공통된 문법적 성질을 가진 단어끼리 모아 놓은 단어의 갈래를 말한다. 따라서 공통된 문법적 성질이 무엇이냐에 대한 국어학자들의 의견에 따라 품사의 분류가 달라지게 진다. 또한 한국어 정보처리의 용도, 해석/생성의 정도, 응용 분야에 따라서도 요구되는 품사 분류의 폭과 깊이가 다양하다. 현재 한국어정보처리 시스템을 구축하는 연구마다 나름대로 각자 품사 분류를 하고 있다. 시스템을 구축하는 모든 사람이 전문가가 아니므로 시스템의 신뢰성이 떨어지고 다른 시스템과의 호환성도 보장하지 못한다. 그러므로 한국어 정보처리를 위한 품사 분류

에 대한 기준과 표준화가 요구된다. 계층적 품사 분류에 의한 품사 사전을 구축하고 품사 분류를 위한 규칙을 찾는다.

신뢰성 있는 품사의 분류는 형태소 해석/생성, 태깅, 구문 해석/생성 등의 한국어 정보처리 시스템의 성능을 향상 시킨다. 즉 품사 분류는 한국어 정보처리의 기반 기술이자 요소 기술이므로 이의 표준화를 통해서 한국어 언어 처리 기술의 향상을 가져오고 한국어 정보처리 기술의 고도화를 이룩할 수 있다. 또한, 전자사전 구축에 기여하고자 한다.

한국어 언어 정보의 기반 확립을 위한 국어정보베이스의 여러 시스템에서도 품사의 분류가 시스템 구축의 출발점이 되고 있다. 이 시스템들간의 정보 및 결과의 교환과 시스템의 통합을 위한 품사사전규칙이 확립되어야 하며 보급을 위한 패키지가 제공되어야 한다.

2절. 연구의 목표 및 범위

본 연구의 목표는 국어정보베이스의 세부 과제의 시스템들이 정보를 공유하고 한 과제의 결과를 다른 과제에서 사용할 수 있도록 확장품사사전규칙을 제시하고자 한다. 또한, 2차년도의 품사전규칙을 형태소 생성과 해석에 사용할 수 있도록 확장하고자 한다. 연구의 내용 및 범위는 다음과 같다.

- ■ 확장품사사전규칙 제시
  - ● 기존 품사사전규칙의 검토
  - ● 형태론적 품사 분류체계 정립
  - ● 통사론적 품사 분류체계 정립
  - ● 다양한 국어정보처리 분야에 적용 검토
- ■ 보급 패키지 개발
  - ● 다양한 국어정보처리 분야에 맞는 품사사전규칙 제공
  - ● 확장품사사전규칙의 보급 패키지 개발

2장. 관련 연구

태깅의 목적은 가공되지 않은 코퍼스에 언어학 지식을 추가하여 언어에 관한 현상을 연구하는데 있다. 따라서 코퍼스와 태깅은 밀접한 관계를 가지고 있다. 따라서, 관련연구에서는 코퍼스와 태깅에 대해서 알아본다. 또한, 국어학에서의 품사 분류에 대하여 조사하여 본다. 한국어정보처리의 품사 분류는 국어학의 연구가 바탕이 되어야 한다.

1절. 코퍼스 언어학

코퍼스를 분석하여 언어를 연구하는 코퍼스 언어학은 19세기말부터 시작되었다. 대표적인 연구로는 언어 습득, 철자 통계, 언어 교육법, 비교 언어학 등에 관한 분야에서 이루어졌다.

이러한 코퍼스를 이용한 연구들은 Chomsky의 연구로 인해 주춤해 졌다. Chomsky는 언어학의 연구 방향을 경험주의에서 합리주의로 바꾸어 놓았다. Chomsky는 코퍼스는 언어학자에게 유용한 도구가 아니며 언어 수행(performance)보다는 언어 능력(competence)을 규명하기에는 적당하지 않다고 주장하였다. 또한, 코퍼스를 통계적으로 분석하는 작업 과정들이 모두 사람의 눈에 의해서 이루어졌다. 이러한 요인들은 코퍼스를 기반으로 하는 많은 언어학 연구들에 부정적인 영향을 주었다.

이러한 어려움에도 불구하고 코퍼스 데이터가 가지고 있는 장점 때문에 코퍼스 언어학은 다시 각광받게 되었다. 코퍼스는 과학적 방법에서 볼 때 가장 강력한 방법론이며 정량적인 데이터이므로 언어학자에게 매우 유용하다. 또한 강력한 컴퓨터와 소프트웨어의 등장으로 인해 시간이 많이 걸리고 오류가 많던 코퍼스 관련 작업이 매우 편리해졌다.

컴퓨터로 읽을 수 있는 코퍼스와 컴퓨터 성능의 폭발적 증가에 힘입어 1980년대 이래로 코퍼스 언어학이 매우 발달하였다.

2 절. 코퍼스

코퍼스는 하나 이상의 문장이 모여서 이루어진 것이다. 이러한 코퍼스는 다음의 4 가지 특징에 의해서 설명될 수 있다.

- **대표성**: 모든 문장을 가진 코퍼스를 가질 수는 없다. 따라서 다양한 저자와 분야의 문장들로 구성된 대표성 있는 코퍼스를 구축해야 한다.
- **유한 크기**: 모니터 코퍼스를 제외하고는 유한한 크기를 가진다. 코퍼스를 기반을 둔 연구나 기법에 대한 상호 평가가 이루어질 수 있다.
- **가독성**: 빠른 시간내에 코퍼스를 분석하거나 새로운 정보를 쉽게 추가하기 위해서는 코퍼스를 컴퓨터로 읽을 수 있어야 한다.
- **표준성**: 코퍼스 관련 연구의 비교를 위해서 표준이 될 수 있는 코퍼스가 마련되어야 한다.

코퍼스는 문장 코퍼스와 음성 코퍼스로 나눈다. 또한 문장 코퍼스는 문어체 문장 코퍼스와 구어체 문장 코퍼스로 나누어 진다. 본 연구에서는 문어체 문장 코퍼스에 주로 관심을 가진다.

문장 코퍼스는 많은 기관에서 만들어 졌다. 또한, 이러한 코퍼스들을 몇몇 기관에서 제공하고 있다. 기관들과 그 기관들이 제공하는 코퍼스는 다음과 같다.

1. ICAME (International Computer Archive of Modern English)

1997 에 설립된 노르웨이의 Bergen 에 있는 NCCH (the Norwegian Computing Centre for the Humanities)의 조직이다. (http://nora.hd.uib.no/corpora.html/) 영어 코퍼스와 관련 소프트웨어의 보급 기관의 역할을 하고 있다. 제공하는 코퍼스는 다음과 같다.

- Brown
- LOB
- London Lund Corpus
- Lancaster/IBM Spoken English Corpus

- Kolhapur Corpus
- Melbourne-Surrey Corpus
- Polytechnic of Wales Corpus

2. LDC (The Linguistic Data Consortium)

1992 년에 ARPA (Advanced Research Projects Agency)로부터 지원을 받아 설립된 대학교, 기업체, 정부 간의 공개 협의체이다. 현재 LDC 의 주기관은 펜실바니아 대학교이다. (http://www.ldc.upenn.edu/) 음성 및 문장 데이터베이스 및 사전을 만들고 수집하고 배포하고 있다. 여기에서 가지고 있는 코퍼스는 다음과 같다.

- TIPSTER: Information Retrieval Text Research Collection
- Japanese Business News Text: Nihon Kezai Shimbun
- Penn Treebank
- Spanish News Text Collection
- United Nations Parallel Text Corpus: English/French/Spanish

3. UCREL (University Centre for Computer Corpus Research on Language)

영국 Lancaster 대학에 있는 연구기관으로 1970 년에 Geoffrey Leech 에 의해서 시작되었다. (http://www.comp.lancs.ac.uk/computing/research/ ucrel/) UCREL 의 목표는 자연어 처리에 있어서 컴퓨터와 코퍼스를 기반으로 연구하는데 있다.

- BNC

4. OTA (Oxford Text Archive)

영국 Oxford 대학의 조직으로 20 여년간 다양한 종류의 문장들을 다양한 원천으로부터 수집하여 왔다. (http://ota.ox.ac.uk/) 36 개 언어의 약 2,000 여 종류의 다양한 텍스트를 가지고 있다.

- SUSSANE

위에서 제시된 여러 가지 코퍼스들 중에서 대표적인 코퍼스 몇 가지를 살펴보겠다.

5. Brown Corpus

1960 년에 W. N. Francis 와 H. Kucera 에 의해서 시작되어 1964 년에 완성된 것으로 100 만 단어의 코퍼스를 구축하였다. 하나의 텍스트는 약 2,000 단어로 이루어 지도록 하였기 때문에 전체는 500 개의 텍스트로 이루어져 있다. 이 텍스트들은 1961 년에 미국에서 인쇄된 것들로 15 개 분야에서 모아졌다. 대화가 50%가 넘는 드라마와 소설은 제외되었다.

이 코퍼스에 대해서 1970 에서 1980 년까지 품사 태깅을 하였다. 태그 세트는 87 개에서 187 개로 확장하였다. 품사 태깅 시스템은 TAGGIT 으로서 77 ~ 78%의 정확도를 보였다. TAGGIT 은 Greene 과 Rubin 에 의해서 만들어졌다.

6. LOB Corpus (the Lancaster/Oslo-Bergen Corpus of British English)

Brown 코퍼스에 대항하여 영국 영어 코퍼스를 구축하고자 하는 목적에서 1970 년에 영국 Lancaster 대학의 G. Leech 교수에 의해서 시작되었다. 그 후 노르웨이의 S. Johansson 과 J. Hauge 과 K. Hofland 등이 도와 주어서 1978 년에 종료되었다. 15 개 장르의 100 만 단어로 이루어졌다.

1978 년에서 1983 년까지는 이 코퍼스에 대해서 품사 태깅을 하였다. 이 사용된 태그 세트가 CLAWS1 (the Constituent Likelihood Automatic Word-tagging System) 으로 132 개로 되어 있다. (부록 A) 이 CLAWS1 은 CLAWS2 로 확장되었는데 태그 세트의 수효는 166 개이다. (부록 B) 최근의 태그 세트는 CLAWS7 이다. (부록 C)

7. Lancaster-Leeds Treebank, Lancaster Parsed Corpus

Lancaster-Leeds Treebank 는 LOB 코퍼스의 일부분을 손으로 구문 태깅한 것으로 1983 년에서 1985 년 사이에 Leeds 대학의 G. Sampson 교수에 의해서 이루어졌다.

45,000 단어를 가진 코퍼스로서 각 문장은 괄호 매기기 형태로 태깅되었다. 태깅의 단위는 문장, 구, 절 등이며 태그 세트는 일반적인 구문 기호를 사용하였다.

Lancaster Parsed Corpus 는 LOB 코퍼스의 일부분인 14 만 단어의 문장을 자동으로 파싱 태깅한 것이다.

8. SUSANNE (Surface and Underlying Structural Analysis of Natural English)

1992 년과 1993 년에 G. Sampson 교수에 의해서 이루어 진것으로 Brown 코퍼스의 일부분에 태깅을 한 것이다. 500 단어를 가진 64 개의 화일에 대해서 태깅을 하였으므로 전체는 128,000 단어로 이루어져 있다. 태그는 크게 Form 태그와 Function 태그의 두 종류가 있다. (부록 H)

9. BNC (the British National Corpus)

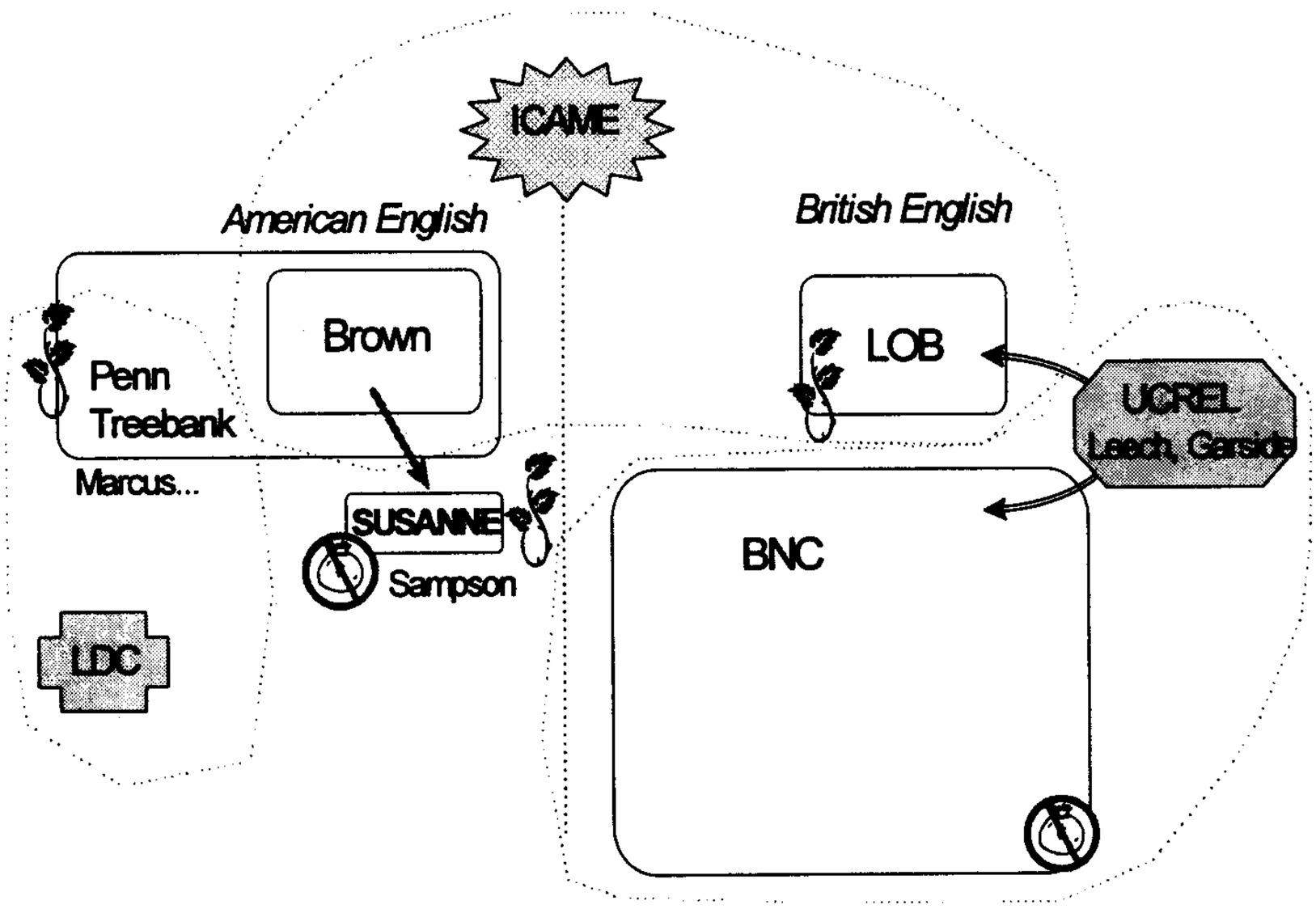
1991 년에 시작되어 1994 년에 종료된 코퍼스로써 90%의 문어체 문장과 10%의 구어체 문장으로 이루어진 것으로 1 억 단어를 가지고 있다. 이 코퍼스는 현대 영국 영어의 모든 분야를 포함하도록 계획되었다. (http://info.ox.ac.uk/bnc/) Lancaster 대학의 UCREL, OUP (Oxford University Press), OUCS (Oxford University Computing Service), 영국 국립도서관 등이 참여하였다. 품사 태깅은 65 개의 태그로 이루어진 C5 로 CLAWS 에 의해서 자동 태깅되었다. (부록 D) 이 태그 세트는 C7 으로 확장되었다. (부록 E) 특히 코퍼스의 코딩에 있어서 ISO standard 8879 (SGML)을 사용하여 TEI (Text Encoding Initiative)의 지침에 따라 이루어졌다. OUCS 가 이 일을 주로 담당하였으며 CDIF (Corpus Document Interchange Format)을 사용하였다.

10. Penn Treebank

1989 년에 Pennsylvania 대학에서 시작된 것으로 450 만 단어로 이루어진 미국 영어이다. 품사 태깅은 48 개의 태그를 가지고 Church 의 PARTS 를 이용하여 이루어졌다. 구문 태깅은 Hindle 의 Fidditch 를 사용하였다.

위와 같은 코퍼스와 관련 기관과의 관계를 [그림 1]로 표현하였다. 코퍼스 간

의 전후 관계, 해당 코퍼스를 가지고 있는 기관들의 관계가 나타나고 있다. 또한, 구문 태깅을 한 코퍼스를 구별하였고 구입시 비용이 필요한가에 대한 표시도 하였다.



[그림 1] 코퍼스와 관련 기관과의 관계 (출처: 이공주, KAIST)

3절. 태깅

태깅이 된 코퍼스는 일반 문장에 언어 지식을 추가한 코퍼스이다. 이러한 태깅은 코퍼스가 가지고 있는 정보를 쉽게 분석하고 검색할 수 있게 도와준다.

Leech는 다음과 같이 태깅의 7원칙을 제시하였다.

1. 태깅된 코퍼스에서 태그를 제거하면 쉽게 가공되지 않은 코퍼스를 얻을 수 있어야 한다.
2. 문장에서 스스로 태깅을 추출할 수 있어야 한다.
3. 태깅은 일반 사용자가 이해할 수 있는 지침에 근거하여 이루어져야 한다.

4. 누가 어떻게 태깅을 했는지 명확하게 밝혀져야 한다.
5. 일반 사용자는 코퍼스 태깅은 전혀 잘못이 없는 것이 아니라 잠재적으로 유용한 도구라는 것을 인식하도록 해야 한다.
6. 태깅 계획은 널리 동의되고 이론 중립적인 원칙에 최대한 근거를 두어야 한다.
7. 표준이 될 것이라는 사전 권리를 가지는 태깅 계획은 없다.

태깅에는 다음과 같이 여러 가지 종류가 있다.

1. 품사 태깅

코퍼스 태깅에서 가장 기본이 되는 것이 품사 태깅이다. 코퍼스에서 특정한 데이터를 검색하는데 도움을 주고 더 나아가 구문이나 의미 태깅의 기본이 된다. 또한 동형이의어를 구별하는데 필요하다.

품사 태깅은 컴퓨터에 의해 자동으로 이루어지고 있다. 1971 년에 Greene 과 Rubin 은 TAGGIT 이라는 태깅 프로그램으로 71%의 정확성을 이루었다. 그 후 많은 발전을 이루어 1980 년초에는 Lancaster 대학의 UCREL 팀은 CLAWS 프로그램으로 95%의 정확도를 이루었다.

품사 태깅을 위해서는 우선 품사 태그를 결정하여야 한다. 영어에 대한 여러가지 태그를 부록에 수록하였다. 또한 한국어에 대해서도 "1996 년도 제 1 회 우리말 정보처리 규격 심포지움에서 발표된 "국어 통사.형태 태그 규격"을 수록하였다. 또한 국어정보베이스의 세부과제에서 사용하고 있는 품사 태그를 비교하였다.

- CLAWS1 (부록 A)
- CLAWS2 (부록 B)
- CLAWS7 (부록 C)
- C5 (부록 D)
- C7 (부록 E)
- Penn Treebank Parts-of-Speech Tags (부록 G)
- 국어 통사.형태 태그 규격 (부록 J)

CLAWS 는 UCREL 에서 개발한 자동 태깅 시스템이다. CLAWS1 은 LOB 코퍼스를 태깅하기 위해서 사용된 태그 세트이며 CLAWS2 로 확장되었다. 현재는 CLAWS7 을 사용하고 있다. C5 와 C7 은 BNC 를 태깅하기 위한 것이다. 이들은 모두 UCREL 가 주도하여 개발하였기 때문에 매우 유사하다. 주목할 것은 확장할 때 어떠한 요소들을 고려하였는가이다.

Penn TreeBank 는 구문 태깅을 하는 것이 목적이지만 이것으로부터 구문 규칙을 찾기 위해서는 우선, 품사 태깅이 선행되어야 한다.

현재 국어공학센터와 한국과학기술원에서는 약 1,000 만 어절의 코퍼스를 태깅하였다. 이중에 500 만 어절은 전에 사용하던 KAIST 태그를 사용하였고 500 만 어절은 "국어 통사.형태 태그 규격"을 사용하였다. 현재 추가로 4,000 만 어절의 가공되지 않은 코퍼스를 구축하고 있으며 여기에서 800 만 어절에 품사를 태깅하고 있다.

2. 구문 태깅

구문 태깅은 품사 태깅이 이루어진 후에 행해진다. 구문 태깅이 된 코퍼스를 흔히 트리뱅크(treebank)라고 한다. 이것은 태깅된 결과가 트리 형태로 나타나기 때문이다. 이 구문 태깅된 결과를 그림으로 표현하기는 힘들기 때문에 괄호를 이용하여 트리 구조를 나타낸다.

[S[NP Claudia_NP1 NP][VP sat_VVD [PP on_II [NP a_AT1 stool_NN1 NP] PP] VP] S]

이러한 괄호 매기기 태깅은 들여쓰기를 이용하면 그림과 같은 효과를 낼 수 있다. 이러한 방법은 Penn Treebank 에서 사용하고 있다.

```
[S
[NP Claudia NP]
[VP sat
        [PP on
                [NP a stool NP]
        PP]
VP]
```

S]

자동 구문 태깅은 자동 품사 태깅보다 적중률이 낮기 때문에 후편집을 하거나 일일이 손으로 한다. 이를 위해서는 명확한 지침서가 필요하다.

구문 태깅된 코퍼스는 문법 이론을 연구하는데 필요한 실제 데이터이므로 매우 중요하다. 구문 해석에 필요한 문법을 만들거나 또는 만들어진 문법을 점검할 수 있다.

구문 태깅을 위해서도 구문 태그가 필요하다. 다음과 같은 구문 태그를 부록에 수록하였다.

- UCREL Skeleton Paring Tagset (부록 F)
- Penn Treebank Tagset (부록 G)
- SUSANNE Tagset (부록 H)
- 국어 구문 태그 규격 (부록 M)

UCREL Skeleton Parsing Tagset 은 LOB 에 구문 태깅하기 위해서 사용하였다. 또한 SUSANNE Tagset 은 Brown 코퍼스에 태깅하기 위해서 사용하였다. 구문 태깅에 있어서 가장 유명한 것은 Penn Treebank 이다. 구문 태그에 문장에서의 기능을 표현하는 기능 태그가 추가된다.

국어 구문 태그도 규격 심포지움에서 발표된 것이다. 국어 형태·통사 태그 규격으로 품사 태깅이 된 코퍼스에서 일부분을 구문 태깅하고 있다. 현재 3 만 문장을 태깅하였고 올해 12 만 문장을 태깅할 계획으로 있다. 한 문장이 평균 11.35 어절로 구성되어 있으므로 현재 약 35 만 어절을 구문 태깅하였다.

3. 의미 태깅

문장의 의미 태깅에는 두 가지 방법이 있다.

1. 문장에서의 문장 요소들간의 의미적 관계를 표시하는 것이다. 예를 들어서 격과 같은 것으로 구문 태깅에서 수용할 수 있다.
2. 문장의 각 단어의 의미자질을 표시하는 것이다.

의미 태깅에는 품사 태깅이나 구문 태깅과는 다르게 여러 작업들간의 유사성이 별로 없다. Sedelow 는 1969 년에 Roget's Thesarus 를 사용하였다. 또한 Wilson 은 Schmidt 의 계층적 의미 구조를 8 자리 숫자로 표시하여 의미 태깅을 하였다.

또한 의미 태깅으로 많이 사용하고 있는 것이 WordNet 의 의미 분류이다. 이 분류는 계층적 의미 분류로 명사는 25 개로 동사는 15 개로 최상위 분류가 되어 있다. 이 최상위 분류를 부록 I 에 수록하였다.

4. 담화 태깅

Stenstr 는 London-Lund spoken corpus 에 16 가지의 종류의 담화 태깅을 하였다. 예를 들어 'apologies', 'greetings', 'hedges', 'politeness', 'responses' 등이다. 담화 분석에서의 잠재적 역활에 비해서 널리 사용되지는 못하였다. 이러한 분류는 문맥에 의존적이고 또한 다른 태깅에 비해서 논란의 소지가 많기 때문이다.

5. 발음기호 태깅

음성 코퍼스에는 발음기호 태깅을 할 수 있다. 이러한 작업은 사람이 직접 해야 하기 때문에 문장 코퍼스에 비해서 수효가 매우 적다. 또한 음성 신호는 하나의 신호로 명확하게 나누어 지지 않는다. 따라서 같은 소리가 문맥에 따라서 다른 기호로 표현될 수 있다.

대표적인 것으로 MARSEC 코퍼스가 있다. 이 코퍼스는 Lancaster/IBM Spoken English Corpus 이다.

4 절. 태깅 형식

태깅 형식에 있어서 표준은 없다. 가장 오래된 태깅 형식은 COCOA 이다. COCOA 는 OCP(Oxford Concordance Program), Longman-Lancaster Corpus, Helsinki Corpus 등에서 사용되었다. COCOA 는 두 개의 인수를 가진 괄호 집합으로 이루어진다. 하나는 태그이고 또 하나는 해당 스트링이다. 다음의 예에서 A 는 저자를 나타낸다.

<A CHARLES DICKENS>

<A HOMER>

현재는 좀 더 형식화된 국제적 표준을 취하려고 하고 있다. 대표적인 것이 TEI(Text Encoding Iniative)이다. TEI는 문서에 필요한 각종 태그 세트를 추가하여 문서화하고 원할하게 주고 받을 수 있는 방안을 연구하는 국제적인 프로젝트이다. TEI에서는 SGML 형식을 사용한다. SGML은 다음과 같은 장점이 있다.

- 명확성
- 간단성
- 형식적 엄밀성
- 국제적 표준

TEI의 저자, 제목, 일자 등을 가진 헤더와 텍스트로 이루어진다. 텍스트에서 태그는 &에 의해 단어와 구별된다.

polished&vvd

이 개체는 FSD(feature system declaration) 형식으로는 다음과 같이 표현된다.

```
<fs  id=vvd  type=word-form>
        <f  name=verb-class><sym  value=verb>
        <f  name=base><sym  value=lexical>
        <f  name=verb-form><sym  value=past>
</fs>
```

이러한 표기는 Lancaster_IBM Spoken English Corpus에서는 다음과 같이 표현된다.

polished_vvd

BNC에서는 TEI의 지침에 따라 태깅된 코퍼스를 CDIF (Corpus Document

Interchange Format) 형식으로 표현하였다. 다음은 하나의 예이다.

```
<!DOCTYPE cdif SYSTEM "cdif1.2.dtd" [ ]>
    <cdif><header>
    A sample text containing required, recommended and optional mark-up
    </header><text>
    <div1><head>A sample text</head>
    <p><div>s contain <hi r=it>paragraphs</hi>;
    they may also contain
    <list><label>a)</label><item r=it>Lists</item>
    <label>b)</label><item><hi r=it>Poems</hi> &mdash; such as
    <quote><poem><l>It's only words,
    <l>and words are all I have&hellip;
    </poem></quote></item>
    <label>c)</label><item r=it>Lower-level <div>s</item></list>
    <div2><head>A sub-section</head>
    <p>Contents of the sub-section
    <note place=foot>Two more levels are allowed</note>.
    </text></cdif>
```

국어정보베이스에서는 다음과 같은 품사 태그 형태를 가지고 있다.

014373200 일반적으로 일반/ncn+적/xsn+으로/jca
014373300 한국의 한국/nq+의/jcm
014373400 산세는 산세/ncn+는/jxc
014373500 험준하여 험준하/paa+어/ecs
014373600 계곡을 계곡/ncn+을/jco
014373700 흐르는 흐르/pvg+는/etm
014373800 냇물은 냇물/ncn+은/jxc
014373900 빨리 빨리/mag
014374000 흘러 흐르/pvg+어/ecx
014374100 내려가 내리/pvg+어/ecx+가/px+아/ecx

014374200 버린다 버리/px+ㄴ다/ef

014374300 . ./sf

또한 이 문장에 대한 구문 태깅은 다음과 같다. 괄호 매기기 형태를 취하고 있다.

; 일반적으로 한국의 산세는 험준하여 계곡을 흐르는 냇물은 빨리 흘러 내려가 버린다.

```
(S
   (VP
      (ADJP (NP  일반/ncn+적/xsn )+으로/jca
            (ADJP
                  (NP (NP  한국/nq )+의/jcm
                        산세/ncn )+는/jxt  험준하/paa ))+어/ecs
      (VP
            (VP
                  (VP
                        (NP
                              (VP (NP  계곡/ncn )+을/jco
                                    흐르/pvg )+는/etm  냇물/ncn )+은/jxt
                        (VP
                              (VP (ADVP  빨리/mag )  흐르/pvg )+어/ecs
                              내리/pvg  ))+(AUXP  어/ecx+가/px  ))+(AUXP  아/ecx  버리
/px )))+ㄴ다/ef+./sf )
```

5절. 국어학에서의 품사 분류

단어는 구문 요소의 최소 단위이며 품사란 단어를 문법적 성질의 공통성에 따라 몇 갈래로 묶어 놓은 것이다. 단어 정립은 전통 문법에서의 품사 설정을 위한 선행 작업이며 단어의 내부 조직 분석이 형태론의 연구 분야이며, 단어의 외부 조직, 곧 단어 상호 관계의 구조 분석이 통사론의 연구 분야이다. 즉, 품사의 기본

문제는 재료인 단어의 설정과 분류 원리인 기준의 책정이다. 문법적 성질의 공통성과 단어를 보는 관점에 따라 국어학자들은 적게는 5 품사에서 13 품사까지에 이르기까지 서로 다른 품사 분류론을 펼쳐 왔다. 어디에 기준을 두고 분류하는 것이 문법 기술에 가장 유용성을 가지느냐 하는데 그 주안점을 두어야 할 것이며 문법 모형에 따라 달라질 수 있다.

국어학에서 연구해 온 품사 분류는 크게 구분지어 보면 조사와 어미의 허사에 대한 독립 품사 설정 여부에 따라 변천하여 왔다.

■ **분석 체계**: 조사와 어미를 모두 독립된 품사로 인정한다. 문법 요소들을 여느 낱말과 대등하게 독립된 품사로 인정한 것이다. 의존 형태인 문법 요소들의 중요성을 한결 덧보이게 하고 있을 뿐 아니라 우리말이 지닌 첨가어/교착어로서의 특성을 돋보이게 하는 것이다. 주시경(1910)에서 비롯된 분석적 관점의 품사 분류는 초기의 문법서들에서 널리 채택되어 계승되었다. 김두봉(1946), 이규영(1920), 김원우(1922), 이규항(1922), 이상춘(1925), 김윤경(1948), 장지영(1932), 심의린(1935), 박승빈(1935), 홍기문(1947) 등은 모두 이 분석 체계를 바탕으로 쓰인 문법서들이다.

■ **절충 체계**: 조사만을 품사로 인정한다. 절충 체계는 최현배(1937)에서 비롯되었으며, 그 뒤 박창해(1946), 이희승(1956), 정인승(1956) 그리고 현행 학교 문법 등 거의 모든 문법서에서 통용되어 왔다. 품사 체계라 하면 거의 이 절충 체계를 내세울 정도로 널리 퍼졌고 교육과 맞춤법 등 실제 언어 생활의 기틀이 되어 왔다. 그러나 이 절충 체계는 우리말이 지닌 첨가어로서의 특질과 굴절 언어인 인구어가 지닌 특질을 혼합한 것으로서의 이론적인 면에서는 여러 가지로 문제점이 있다.

■ **종합 체계**: 조사와 어미를 모두 품사로 인정하지 않고 굴절 어미로 처리한 것이다. 어미뿐 아니라 조사도 독립된 품사 자격이 없고 앞 체언에 부속되는 요소로서 격변화 또는 어형 변화에서 나타나는 형태라는 것이다. 이는 우리말이 지닌 첨가어적인 특성을 무시하고 굴절 언어인 서양말 등과 같은 방식으로 처리한 것이다. 이 체계를 맨 먼저 내세운 정렬모(1946)를 비롯하여 장

하일(1947), 김민수(1957), 이숭녕(1956) 등에서 이 체계에 따른 문법 서술을 보인 바 있다. 그러나 이 체계는 우리말을 서구의 굴절어와 같은 차원에서 다룬 것으로서 문제점이 많다. 무엇보다도 그 숱한 문법 요소를 모두 어형 변화로 다룸은 무리한 일이다. 이런 점 때문에 문법 교육이나 일반 언어 생활에서는 이 체계를 거의 외면하고 있다.

국어학에서 품사 분류는 [표 1]과 같이 연구되어져 왔다.

학자	년도	수효	품사 분류
최광옥	1908	8	명사, 대명사, 형용사, 동사, 후사, 접속사, 부사, 감탄사
유길준	1909	8	명사, 대명사, 형용사, 동사, 조동사, 접속사, 후사, 감탄사
김희상	1909	7	명사, 대명사, 동사, 형용사, 부사, 감탄사, 토
주시경	1910	9	임, 엇, 움, 겻, 잇, 언, 억, 놀, 끗
남궁억	1913	9	명사, 대명사, 동사, 토, 형용사, 부사, 접속사, 후치사, 감탄사
김두봉	1916	9	임, 얻, 움, 겻, 잇, 맺, 언, 억, 늑
안곽	1917	10	명사, 대명사, 수사, 부사, 접속사, 감동사, 동사, 형용사, 조사, 조동사
이규항	1923	11	명사, 동사, 형용사, 조동사, 조사, 부사, 접속사, 감탄사, 금지사, 부정사, 호응사
강매	1925	7	이름말, 꼴말, 움즉임말, 꿈임말, 도음말, 잇음말, 늑임말
이상춘	1925	10	명사, 대명사, 동사, 형용사, 조사, 접속사, 종지사, 관사, 부사, 감탄사
홍기문	1927	9	명사, 동사, 형용사, 부사, 감탄사, 격사, 후계사, 접속사, 종결사

이완응	1929	11	명사, 수사, 대명사, 동사, 형용사, 존재사, 조용사, 조사, 부사, 접속사, 감동사
이병기	1929	7	명사, 형용사, 동사, 조사, 접속사, 부사, 감동사
박상준	1932	9	명사, 대명사, 수사, 동사, 형용사, 부사, 접속사, 감동사, 조사
최현배	1935	10	이름씨, 대이름씨, 셈씨, 움직씨, 어떻씨, 잡음씨, 어떤씨, 어찌씨, 느낌씨, 토씨
박승빈	1935	12	명사, 대명사, 존재사, 지정사, 형용사, 동사, 조용사, 조사, 관형사, 부사, 접속사, 감탄사
심의린	1935	10	명사, 대명사, 동사, 형용사, 존재사, 조동사, 조사, 부사, 접속사, 감탄사
권영달	1941	6	명사, 동사, 형용사, 부사, 감발사, 조사
이상춘	1946	6	명사, 동사, 형용사, 부사, 감탄사, 토
정렬모	1946	5	명사, 동사, 관형사, 부사, 감동사
홍기문	1947	10	명사, 대명사, 수사, 동사, 형용사, 부사, 감탄사, 접속사, 후치사, 종결사
김근수	1947	13	명사, 대명사, 수사, 동사, 형용사, 존재사, 지정사, 관형사, 부사, 접속사, 감탄사, 조용사, 조사
김윤경	1948	9	이름씨, 그림씨, 움직씨, 겻씨, 이음씨, 맺음씨, 매김씨, 어찌씨, 느낌씨
박태윤	1948	8	명사, 대명사, 동사, 형용사, 관형사, 부사, 감동사, 조사
심의린	1949	13	명사, 대명사, 수사, 동사, 형용사, 존재사, 지정사, 조용사, 관형사, 부사, 감동사, 조사, 접속사

이인모	1949	6	임자씨, 풀이씨, 매김씨, 어찌씨, 느낌씨, 토씨
정경해	1953	9	명사, 수사, 대명사, 동사, 형용사, 부사, 접속사, 감탄사, 토
정인승	1956	8	이름씨, 움직씨, 그림씨, 매김씨, 어찌씨, 느낌씨, 토씨
이숭녕	1956	8	명사, 대명사, 수사, 동사, 형용사, 관형사, 부사, 감탄사
이희승	1957	10	명사, 대명사, 동사, 형용사, 존재사, 관형사, 부사, 감탄사, 접속사, 조사
최태호	1957	7	명사, 대명사, 동사, 형용사, 관형사, 부사, 감탄사
김민수	1960	7	동사, 형용사, 명사, 부사, 관형사, 접속사, 환투사
학교 문법	1963	9	명사, 대명사, 수사, 조사, 동사, 형용사, 관형사, 부사, 감탄사

[표 1] 국어학에서의 품사 분류

단어는 일정한 음형(형태)을 가지고 일정한 의미를 가지며, 문의 구성요소(기능)가 된다. 그러므로 위와 같은 품사 분류의 기준으로 기능(function), 형식(form), 의미(meaning)의 세 가지가 사용되고 있으며 기준 적용의 차이와 기준의 비중에 따라 품사 분류가 달라지게 된다.

기능은 한 단어가 문장 가운데서 다른 단어와 맺는 관계를 가리키는 통사적 기능을 말한다. 품사의 설정 및 그 분류의 방법이 단어의 문법적 성질 및 그 기능에 의존할 수 있다. 통사적 기능에 따라 체언, 관계언, 용언, 수식언, 독립언 등의 분류가 있다.

형태는 단어의 형태적 특징을 의미한다. 어형이 변화하는가에 따라 굴곡어와 비굴곡어로 나눌 수 있다. 형태와 기능은 표리 관계를 가지고 있으므로 형태는 기능 분석이 전제될 때 보다 유용하다. 즉 통사론적 분류 기준이 형태론적 분석에 상치되지 않고 일관성이 있다.

의미는 개별 단어의 어휘적 의미나 상위개념이 아니라 형식적인 의미이다. 즉 속성개념으로 어떤 단어가 사물의 이름을 나타내느냐 그렇지 않으면 움직임이나 성질, 상태를 나타내느냐 하는 것이다. 이 기준은 객관성과 명확성이 희박하고 모호성이 있기 때문에 품사 분류의 객관적 기준이 될 수 없다. 따라서, 속성개념은 형태, 통사적 구조와 유관성을 가져야 한다.

기능, 형태, 의미가 공통적이면 같은 품사로 묶여 질 수 있는 것이다. 세 가지 기준은 기능, 형태, 의미의 순위로 비중을 두어 분류한다. 품사의 설정은 형태론이나 통사론의 기술에 모두 공통성을 가진 문법 범주이므로, 그 어느 쪽에서든지 이론적으로 타당하고, 그 두 분야에 일관성 있는 규정을 가지고서, 문법 기술에 보다 효용성 있는 방향으로 설정되어야 한다.

국어학자들의 연구 관점에 따라 다양한 품사 분류를 보인다. 그러나, 전반적으로 많은 공통점을 가지고 있으며 약간의 상이점을 보일 뿐이다. 품사 분류는 말 연구의 편의상으로 하는 것이니 일정불변한 표준이 있는 것이 아니고 상이한 분류가 생기는 것은 당연한 것이다. 다만, 국어의 성질에 맞도록 분류하는 것이므로 그 말의 이해와 실용에 가장 편의한 분류를 해야 한다. 궁극적으로 많은 연구의 공통점을 도출하여 중지를 모아 가장 타당성 있게 품사를 분류한 것이 학교 문법의 품사 분류이다. 이 학교 문법이 많은 연구의 기준이 되고 있다.

3장. 확장 품사 사전 규칙

2차년도에서 제시한 "국어 형태·통사 태그 규격"의 확장 방안을 모색한다. 이 규격은 처음의 의도가 코퍼스의 품사 태깅과 자동 품사 태깅을 위한 태거 개발에서 품사의 모호성을 해소하기 위하여 만들었다.

3차년도에서는 이 "국어 형태·통사 태그 규격"을 확장하여 형태소 해석/생성에 사용할 수 있는 태그 세트를 제시한다. 또한 이 품사 태그는 구문 태깅의 기본이 된다. 물론 이 태그로 구문 태깅을 할 수는 없지만 기본적인 구문 규칙을 형성하는 하위 요소가 된다.

1 절. 국어 형태·통사 태그 규격

1996 년도 제 1 회 우리말 정보처리 규격 심포지움에서 발표된 국어 형태·통사 태그 규격은 현재 국어정보베이스 과제에서 코퍼스 태깅에 이용되고 있다. 1,000 만 어절이 태깅이 되어 있다. 이중에서 500 만 어절은 KAIST 태그를 사용하였으며 500 만 어절은 국어 형태·통사 태그 규격을 사용하였다. KAIST 태그는 국어 형태·통사 태그 규격이 정해지기 전의 태그이지만 국어 형태·통사 태그 규격의 기반이 되었기 때문에 큰 차이가 없다. 따라서, KAIST 태그를 사용한 코퍼스도 2 차년도에 개발된 변환기와 변환 사전에 의해서 쉽게 국어 형태·통사 태그로 태깅된 코퍼스로 바꿀 수 있다. 또한, 올해 4,000 만 어절의 가공되지 않은 코퍼스를 구축하며 이 중에서 800 만 어절을 태깅할 계획이다.

현실적으로 이렇게 구축된 코퍼스의 태그를 바꾸는 것은 쉬운 일이 아니다. 물론, 태그에 문제점이 있다면 당연히 수정이 가해져야 한다. 그러나, 이를 위해서는 기존 태그의 문제점을 지적할 수 있는 연구와 실험 및 평가가 이루어져야 한다. 이러한 연구로 생각할 수 있는 첫번째는 태깅의 정확도 분석이다. 다른 태그 세트로 코퍼스를 태깅한 후에 태거를 실행시켜서 비교해야 할 것이다. 또 한 가지 형태·통사 태그 규격의 문제점을 알 수 있는 방법은 구문 태깅을 해 보는 것이다. 구문 태깅은 구문 태그만을 가지고 태깅하는 것이 아니라 품사 태그가 필요하기 때문에 구문 태깅 중에 품사 태그에서 문제점이 발견되면 문제점을 지적할 수 있다. UCREL 의 CRAWS 태그 세트에서 보듯이 지속적인 연구가 필요하다. 다양한 태그 세트에 대한 비교 연구가 있어야 한다.

컴퓨터를 통하여 한국어를 처리하기 위해서는 애매성이 없고 풍부하고 다양한 정보를 가진 품사 분류가 필요하다. 따라서, 이 품사 분류는 당연히 세분되고 기능, 형태, 의미의 기준이 상호 보완적으로 적용되어야 한다. 국어정보베이스에서는 한국어 처리는 물론 한국어에 관심 있는 모든 연구자들에 의해서 두루 사용할 수 있도록 품사 태그에 대한 분류 기준을 명확히 해야 한다. 기본적인 원칙은 다음과 같다.

- 한국어를 대상으로 하며, 한국어 문장에서 두루 사용되는 외국어나 특수기호들도 대상으로 삼는다.
- 품사 태그 집합의 설정은 학교 문법을 최대한 반영하며, 통사론적 분석 위주보다는 형태론적 분석 위주로 한다.

- 품사 태그는 여러 분야에서 다양한 용도로 사용할 수 있도록 계층적으로 분류한다.

위와 같은 원칙에 입각하여 [표 2]와 같은 한국어 품사 태그 집합을 설정하였다. 최상위 9 개 부류(기호, 외국어, 체언, 용언, 수식언, 관계언, 독립언, 어미, 접사)로 나누었고, 다음으로는 세분류로 총 54 개의 품사 태그를 설정하였다.

상위 분류			태그	
기호(s)			sp 쉼표 sl 여는 따옴표 및 묶음표 sd 이음표 su 단위 기호	sf 마침표 sr 닫는 따옴표 및 묶음표 se 줄임표 sy 기타 기호
외국어(f)			f 외국어	
체언(n)	보통명사(nc)	서술성 명사(ncp)	ncpa 동작성 명사	ncps 상태성 명사
		비서술성 명사(ncn)	ncn 비서술성 명사	
	고유명사(nq)		nq 고유명사	
	의존명사(nb)		nbu 단위성 의존명사	nbn 비단위성 의존명사
	대명사(np)		npp 인칭대명사	npd 지시대명사
	수사(nn)		nnc 양수사	nno 서수사
용언(p)	동사(pv)		pvd 지시 동사	pvg 일반 동사

	형용사(pa)	pad 지시형용사 paa 성상형용사
	보조용언(px)	px 보조용언
수식언(m)	관형사(mm)	mmd 지시관형사 Mma 성상관형사
	부사(ma)	mad 지시부사 Maj 접속부사 mag 일반부사
독립언(i)	감탄사(ii)	ii 감탄사
관계언(j)	격조사(jc)	jcs 주격조사 Jco 목적격조사 jcc 보격조사 Jcm 관형격조사 jcv 호격조사 jca 부사격조사 jcj 접속격조사 jct 공동격조사 jcr 인용격조사
	보조사(jx)	jxc 통용보조사 jxf 종결보조사
	서술격조사(jp)	jp 서술격조사
어미(e)	선어말어미(ep)	ep 선어말어미
	연결어미(ec)	ecc 대등적 연결어미 Ecs 종속적 연결어미 ecx 보조적 연결어미
	전성어미(et)	etn 명사형 전성어미 Etm 관형사형 전성어미
	종결어미(ef)	ef 종결어미
접사(x)	접두사(xp)	xp 접두사

	접미사(xs)	xsn 명사파생접미사 xsm 형용사파생접미사	xsv 동사파생 접미사 xsa 부사파생 접미사

[표 2] 국어 형태·통사 태그 규격

이 태그 집합이 완전하거나 모든 것은 아니다. 다만, 정보의 공유와 교환을 위한 하나의 기준이다. 시스템의 용도에 따라서, 이 분류를 세분하거나 또는 추가할 수 있다. 그렇지만, 이 경우에도 여기에서 제시한 규격을 토대로 이루어진다면 소기의 목적을 이룰 수 있다.

제 1 회 우리말 정보처리 규격 심포지움에서 발표된 국어 형태·통사 태그 규격을 검토하고 보완하기 위한 "한국어 품사 태그 세트 검토 및 보완 전문가 회의"가 1997 년 5 월 30 일 ~ 31 일에 대전에서 개최되었다. 남영준 교수(전주대)가 1 차 기준안에 대한 설명을 하였고 안동언 교수(전북대)와 강승식 교수(한성대)가 1 차 기준안에 대한 보완책을 제안하였다. 또한 권혁철 교수가 검토 의견을 내놓았다. 그 후, 전체 회의와 분임조 토의를 가졌다.

전문가 회의에서 1 차 기준안에 대하여 연구자들의 관심분야에 따라서, 국어정보처리 분야에 따른 다양한 의견과 언어학 관점에 따른 다양한 의견을 제시하였다. 이 회의에서 지적된 1 차 기준안의 문제점은 다음과 같다.

- 1 차 기준안에 대한 검토 미흡
  - 다양한 의견 수렴 부족
  - 국어 형태 통사 태그의 보급 부진
  - 태깅 코퍼스의 미보급
  - 실험 및 평가 미비
- 규격 제정의 기준 미비
  - 일관성 결여
  - 지침서 미비

위의 문제점은 전체적인 관점에서 바라본 것이며 각 세부 태그에 대한 다양한 의견들이 제시되었다.

- 기호
  - 기호 자체가 태그의 역할을 하므로 소분류는 불필요하다.
- 외국어
  - 외국어의 기준이 모호하다.
- 체언
  - 수사의 소분류인 양수사에 수관형사를 포함하는 것은 문제가 있다.
- 용언
  - 동사의 소분류에서 자동사와 타동사의 분류가 필요하다.
  - 보조용언의 소분류에서 보조동사와 보조형용사의 분류가 필요하다.
  - 지시사의 분류가 불필요하다.
- 수식언
  - 보통명사의 소분류와의 일관성이 결여되어 있다.
    - 서술성 부사와 비서술성 부사
    - 동작성 부사와 상태성 부사
  - 관형사에서 수관형사의 소분류가 필요하다.
- 관계언
  - 격조사의 소분류가 모호하다.
- 접사
  - 명사와 동사와의 결합에 따른 접두사의 소분류가 필요하다.
  - 결합 특성 때문에 접미사의 소분류에서 복수접미사를 구분해야 한다.

분임 토의와 전체 회의를 통하여 다음과 같은 사항을 도출하였다.

- 1 차 기준안의 대분류와 중분류는 의견 일치
- 일단 사용후 문제점 도출
- 표준안은 권고안이어야 한다.

- 워킹 그룹 결성
  - ■ 각 연구기관의 태그 세트 수집 및 분석
  - ■ 지속적인 연구
- 지식베이스 및 도구의 보급

위에서 제시한 문제점 및 결정사항은 1997년도 제2회 우리말 정보처리 규격 심포지움에서 "국어 형태 통사 태그의 표준화"라는 제목으로 발표되었다. 앞으로의 진행 방향은 다음과 같다.

- 국어 형태 통사 태그의 표준화를 계속 추진한다.
  - ■ 최소 범위의 태그 세트
    - 각 연구기관에서 확장하여 사용
  - ■ 최대 범위의 태그 세트
    - 각 연구기관에서 선택하여 사용
- 응용 분야 따른 태그 세트 제공
- 진행 과정, 결정 사항 및 지침서의 문서화에 노력한다.
- 워킹 그룹을 결성하여 지속적으로 연구한다.

2절. 형태소 해석과 생성을 위한 확장 품사 태그

"국어 형태·통사 태그 규격" 만을 가지고 형태소 해석과 생성을 하기에는 부족하다. 조사 및 어미의 이형태, 동사와 형용사 어미의 구분 등 필요한 정보를 프로그램 내에서 처리할 수도 있지만, 이러한 형태적 정보 및 용도 등의 정보를 태그에 표현해 보고자 하였다.

이와 관련된 가장 적합한 형태소 처리 방법은 접속정보를 이용하는 것이다. 물론 procedural 하게 처리하는 시스템에서는 이 접속정보를 그대로 사용할 수가 없거나 필요가 없을 것이다. 그러나 이 접속정보에 표현된 형태적 정보를 프로그램 작성에 반영할 수 있다. 규칙이나 고려해야 할 사항을 미리 태그로 표현한 것이다.

다음과 같은 두 가지 종류의 품사 분류를 제시한다. 자세한 내용은 부록에 첨부하였다.

- 형태소 해석(정보검색)을 위한 품사 분류 (부록 L)
- 형태소 생성(기계번역)을 위한 품사 분류 (부록 K)

이 품사 분류는 단순히 분류 및 태그 설정에 그친 것이 아니라 정보검색 시스템의 한국어 형태소 해석기와 기계번역 시스템의 한국어 형태소 생성기에서 사용되어졌다. 여러 번의 수정을 거쳐서 완성되어 진 것이다.

좌, 우접속정보는 한 어절 내에서 형태소 간의 접속 여부를 판단할 수 있는 정보이므로 유종성/무종성 및 양성모음/음성모음 등의 형태적 정보가 많이 포함되어 있다. 또한, 용언의 불규칙정보도 태그에 포함시켰다.

3 절. 국어 구문 태그 규격

국어 구문 태그는 한국어 문장 분석과 밀접한 관계를 가진다. 문장 분석의 단위는 구, 절, 그리고 문장이다. 따라서, 구문 태그도 이 문장 분석 단위를 중심으로 만들어 졌다. 또한, 구문 트리 태깅을 위해서는 우선 기본적인 문법 형식이 있어야 하며 여기에서는 구구조 문법을 사용한다.

구구조 문법은 영어와 같은 구성적인 언어에 적합하다고 알려져 있기 때문에 흔히들 구구조 문법이 한국어 분석에 적합하지 않다고 말한다. 그러나, 한국어가 완전히 비구성적인 언어가 아니며, 부분적으로는 매우 제약적인 어순을 지니고 있기 때문에 구구조 문법을 사용함으로써 얻는 장점 또한 무시할 수 없다.

구문 태그 설정을 위한 품사 태그는 기존의 품사 태그를 그대로 사용하지만 다만, 다음의 품사 집합은 조정한다. 주제화를 수행하는 보조사 '는', '은', 'ㄴ'과 그 외의 통용보조사(jxc)로 세분한다. 주제화보조사(jxt)는 문장의 화제가 되며 주로 문미의 술어와 조응관계를 보인다. 따라서, 주제화를 수행하는 '는'과 '은'을 다른 보조사와 분리하여 문미 술어와의 호응 관계를 규정짓도록 한다.

한국어 구문 태그를 부록 M 에 수록하였으며 한국어 문장 성분에 해당하는 것이다. S(문장), NP(명사구절), VP(동사구절), ADJP(형용사구절), ADVP(부사구절), MODP(관형사구절), IP(독립구절), AUXP(보조용언구절) 등이 있다. 부록 M 에서는 문장 성분과 구분 태그간의 관계도 표시하였다.

1996 년에 3 만 문장의 구문 태깅을 끝냈으며 올해에는 12 만 문장의 구문 태깅

을 계획하고 있다. 3 만 문장의 구문 태깅 작업 중에 구문 태그에 있어서는 별다른 문제점을 발견하지 못했다. 다만, 품사 태깅의 중요성을 발견한 것이 수확이라면 수확이다. 이 구문 태그의 문제점도 이렇게 구축된 구문 태깅 코퍼스를 가지고 구문 규칙을 추출하고 구문 해석 및 생성을 통하여 문제점을 찾아야 할 것이다.

4 장. 보급 패키지

확장 품사 사전 규칙을 보급하는 보급 패키지에는 다음과 같은 것들이 포함되어야 한다.

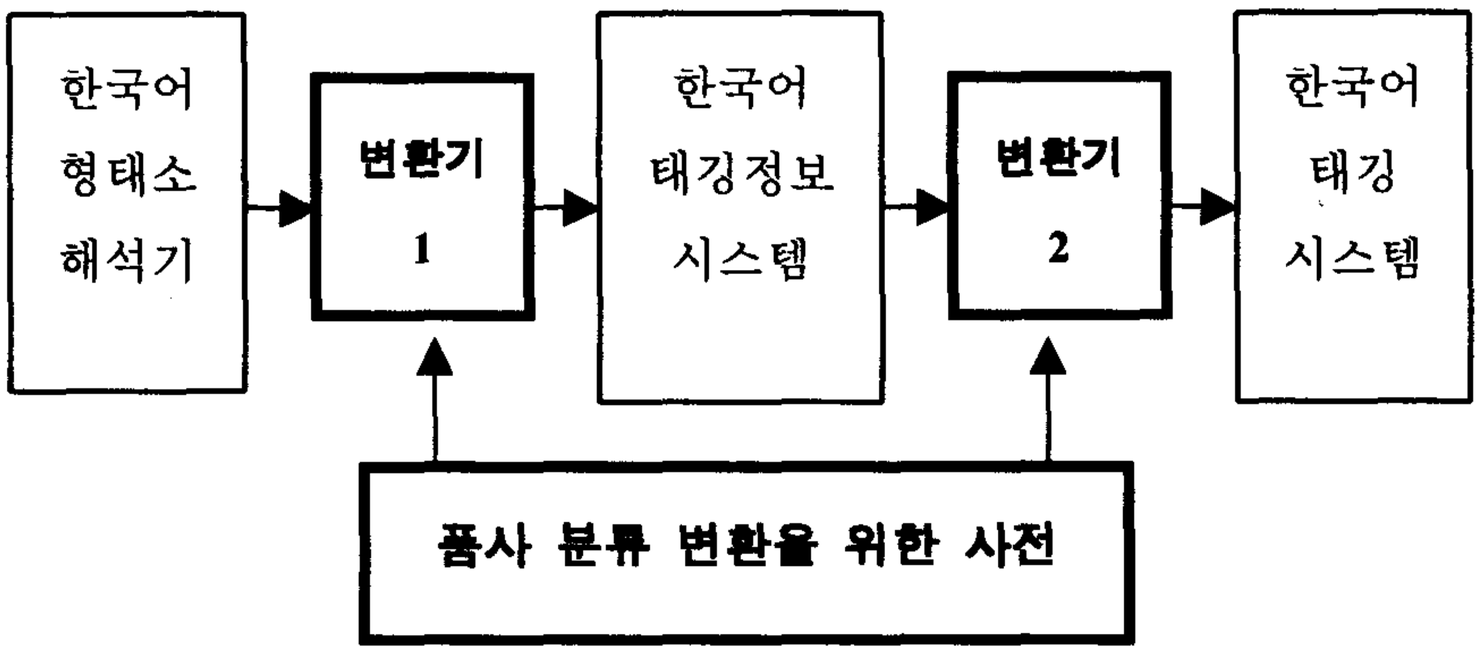
- 확장 품사 사전 규칙이 태깅된 사전
- 기존의 다른 품사 태그와의 변환을 위한 사전
- 기존의 다른 품사 태그와의 변환을 위한 변환기

이중에서 무엇보다도 중요한 것은 확장 품사 사전 규칙이 태깅된 사전의 보급이다. 그러나 아직 확장 품사 사전 규칙이 표준화되지 못했다. 따라서, 제 1 회 우리말 정보처리 규격 심포지움에서 발표된 국어 형태·통사 태그 규격을 기준으로 태깅된 사전을 만들었다.

국어정보베이스의 세부 과제들인 한국어 형태소 해석기, 한국어 태깅 시스템, 한국어 정보 획득 도구들이 아직 각자의 품사 태그를 그대로 사용한다면 다음과 같은 2 개의 품사 분류 변환기가 필요하다.

- **변환기 1**: 한국어 형태소 해석기(포항공대)의 출력 파일 -> 한국어 태깅 시스템(한국과학기술원)의 입력 파일

- **변환기 2**: 한국어 태깅 시스템(한국과학기술원)의 출력 파일 -> 한국어 정보 획득 도구(고려대)의 입력 파일



[그림 2] 품사 분류 변환기

품사 분류 변환기는 한 시스템의 출력 파일에서 해당 품사 분류를 다른 시스템의 품사 분류로 바꾸어 주어서 한 시스템의 출력 파일을 다른 시스템의 입력 파일로 사용할 수 있게 한다. 품사 변환에 있어서 변환 테이블을 이용하면 좋겠지만 2 장의 품사 분류의 비교 분석에서 보았듯이 일대일 변환이 되지 않는다. 더구나 형태소 해석기의 품사 분류는 매우 다르다. 따라서, 품사 분류 변환을 위한 변환 사전을 이용한다.

1 절. 품사 분류 변환을 위한 변환 사전

품사 분류 변환을 위한 사전은 단순히 표제어와 각 세부 과제의 품사 분류로 구성되어 있다. [그림 3]은 하나의 예를 보여 준다.

표제어	한국어 형태소해석기의 품사	한국어 태깅 시스템의 품사	한국어 정보획득도구의 품사	국어 형태·통사 태그 규격
학생	MCNCjy	nc	nncg	ncn

[그림 3] 사전 표제어의 예

변환 사전의 표제어는 한국어 형태소 해석기를 작성한 포항공대에서 제공한 것이다. 한국어 형태소 해석기의 사전에는 표제어와 한국어 형태소 해석기에서 사용하는 품사만이 들어 있다. 이 사전에 한국어 태깅 시스템의 품사, 한국어 정보 획득 도구의 품사, 국어 형태·통사 태그 규격을 첨가한 것이다.

총 표제어의 수효는 62,164 개이며 각 품사 태그별 수효는 [표 3]과 같다.

품사 태그	수효	품사 태그	수효
Ncpa 동작성 명사	8,343	ncps 상태성 명사	210
Ncn 비서술성 명사	36,149	nq 고유명사	8,100
Nbu 단위성 의존명사	325	nbn 비단위성 의존명사	105
Npp 인칭대명사	73	npd 지시대명사	31
Nnc 양수사	67	nno 서수사	0
pvd 지시 동사	0	pvg 일반 동사	3,339
pad 지시형용사	16	paa 성상형용사	2,800
px 보조용언	13	mmd 지시관형사	10
mma 성상관형사	111	mad 지시부사	9
maj 접속부사	0	mag 일반부사	2,078
ii 감탄사	207	Jcs 주격조사	3
jco 목적격조사	2	Jcc 보격조사	2
jcm 관형격조사	1	Jcv 호격조사	6
jca 부사격조사	8	jcj 접속격조사	8
jct 공동격조사	4	jcr 인용격조사	3
jxc 통용보조사	9	jxf 종결보조사	5
jp 서술격조사	0	Ep 선어말어미	8
ecc 대등적 연결어미	28	ecs 종속적 연결어미	10
ecx 보조적 연결어미	5	etn 명사형 전성어미	3
etm 관형사형 전성어미	14	ef 종결어미	27

xp	접두사	0	xsn	명사파생접미사	30
xsv	동사파생접미사	0	xsm	형용사파생접미사	2
xsa	부사파생접미사	0			

[표 3] 표제어의 품사별 수효

2절. 품사 분류 변환기

품사 분류 변환기 1은 한국어 형태소 해석기(포항공대)의 출력 파일의 품사를 변환하여 한국어 태깅 시스템(한국과학기술원)의 입력 파일로 만드는 것이다. 이때, 품사 분류 뿐만 아니라 파일 형식도 시스템에 맞게 변환한다. 명령어와 실행 결과는 [그림 4]와 같다.

품사 분류 변환기 2는 한국어 태깅 시스템(한국과학기술원)의 출력 파일의 품사를 변환하여 한국어 정보 획득 도구(고려대)의 입력 파일로 만드는 것이다. 이때, 품사 분류 뿐만 아니라 파일 형식도 시스템에 맞게 변환한다. 명령어와 실행 결과는 [그림 5]와 같다.

```
% conv1    postech.out    kaist.in
% cat   postech.out
입력 어절(No. 1) : 나는
[1] 나(MPNsm 고유명사) + 는(jCm 격조사)
[2] 나(MPNsm 고유명사) + 는(jSOm 보조사)
[3] 나(MCNCOm 보통명사) + 는(jCm 격조사)
[4] 나(MCNCOm 보통명사) + 는(jSOm 보조사)
[5] 날(DIYarR 동사) + 는(mjPmhn 전성어미)
[6] 나(TOm 대명사) + 는(jCm 격조사)
[7] 나(TOm 대명사) + 는(jSOm 보조사)
입력 어절(No. 2) : 학교에
[1] 학교(MCNCjm 보통명사) + 에(jCCN 격조사)
[2] 학교(MCNCjm 보통명사) + 에(jJm 접속조사)
입력 어절(No. 3) : 간다.
[1] 가(DImagR 동사) + ㄴ(mjpmge 무종성현재결합형전성어미) +
```

```
다(mTCCCCm 종결서술형어말어미) + .(gD 기호)
% cat   kaist.in
@
나 nq 는 jc
나 nq 는 jx
나 nc 는 jc
나 nc 는 jx
날 pv 는 exm
나 np 는 jc
나 np 는 jx
@
학교 nc 에 jca
학교 nc 에 jj
@
가 pv ㄴ exm 다 ef . s.
@
```

[그림 4] 품사 분류 변환기 1 의 명령어와 실행 결과

```
% conv2    kaist.out     korea.in
% cat   kaist.out
나/np+는/jx
학교/nc+에/jca
가/pv+ㄴ/exm+다/ef+./s.
% cat   korea.in
나는        나_npp+는_jx
학교에      학교_nncg+에_jca
가ㄴ다      가_vv+ㄴ_efd+다_eff+._ss.
```

[그림 5] 품사 분류 변환기 2 의 명령어와 실행 결과

5장. 결론

본 연구에서는 국어정보베이스의 세부 과제들인 한국어 형태소 해석기, 한국어 태깅 시스템, 한국어 정보 획득 도구 등이 서로 정보를 공유하고 한 과제의 결과를 다른 과제에서 사용할 수 있도록 확장 품사 사전 규칙을 제시하였다.

2차년도에 발표된 국어 형태·통사 태그 규격의 문제점을 살펴보고 다양한 의견을 제시하였다. 그동안 있었던 심포지움과 전문가 회의의 결정 사항을 보여 주고 앞으로의 진행 방향을 제시하였다.

2차년도의 품사 사전 규칙을 형태소 생성과 해석에 사용할 수 있도록 확장하였으며 실제 시스템에 적용하였다.

확장 품사 사전 규칙이 태깅된 사전과 품사 분류가 서로 달라서 한 시스템의 결과를 다른 시스템에서 사용하지 못하는 문제점을 해결할 품사 분류 변환기가 포함된 보급 패키지를 개발하였다.

품사 분류 변환기 1은 한국어 형태소(포항공대) 출력 파일의 품사 분류를 한국어 태깅 시스템(한국과학기술원) 입력 파일의 품사 분류로 변환한다. 품사 분류 변환기 2는 한국어 태깅 시스템(한국과학기술원) 출력 파일의 품사 분류를 한국어 정보 획득 도구(고려대) 입력 파일의 품사 분류로 변환한다. 이때, 품사 분류뿐만 아니라 파일 형식도 시스템에 맞게 변환한다. 이 변환을 위하여 사전을 작성하였다. 사전에는 표제어와 각 시스템의 품사 분류가 기재되어 있으며 국어 형태·통사 태그 규격이 첨부되어 있다.

이제 이 규격의 사용을 강력히 권고하여야 정보의 공유라는 국어정보베이스의 목적을 이룰 수 있을 것이다. 이를 위해서는 사전과 코퍼스의 보급이 우선적으로 이루어져야 한다.

6장. 참고문헌

[1] 고영근, 국어문법의 연구 -그 어제와 오늘-, 탑출판사, 1987, pp.31-98

[2] 김민수, 국어문법론, 일조각, 1984, pp.29-32, pp.74-79

[3] 김재훈, 서정연, "자연언어 처리를 위한 한국어 품사 태그", 한국과학기술원

인공지능연구센터, CAIR-TR-94-55, 1994

[4] 김희보, 한글 바로쓰기, 종로서적, 1987

[5] 남기심, 고영근, 표준 국어문법론, 탑출판사, 1985, pp.54-62

[6] 남영준 외, "국어 형태·통사 태그 규격," 1996 년도 제 1 회 우리말 정보처리 규격 심포지움, 1996, pp.37-46

[7] 서정수, 국어문법, 한양대, 1996

[8] 시스템공학연구소, 1996 년도 제 1 회 우리말 정보처리 규격 심포지움, 한국과학기술원 인공지능연구센터, 시스템공학연구소 국어공학센터, 1996

[9] 한국어정보처리연구회, 한국어 품사 태그 세트 검토 및 보완 전문가 회의, 한국정보과학회 한국어정보처리연구회, 1997

[10] 시스템공학연구소, 1997 년도 제 2 회 우리말 정보처리 규격 심포지움, 한국과학기술원 인공지능연구센터, 시스템공학연구소 자연어정보처리연구부, 1997

[11] 안동언, 품사사전규칙과 시범패키지, 국어정보처리기술 개발 제 2 차년도 최종보고서, 1996

[12] 이공주, 김재훈, 장병규, 최기선, 김길창, "한국어 구문 트리 태깅 코퍼스 작성을 위한 한국어 구문 태그," KAIST TR-96-102, 1996

[13] 이길록, 국어문법연구, 일신사, 1983, pp.69-146

[14] 이상주, 임해창, "언어정보획득도구에서 사용하는 품사태그집합", 고려대학교, 1996

[15] 이을환, 이철수, 한국어문법론, 개문사, 1985

[16] 이익섭, 남기심, 국어문법론(I), 한국방송통신대학, 1987

[17] 이종혁, "형태론적 품사분류(접속정보) 체계표", 포항공대, 1996

[18] 이희승, 안병희, 한글 맞춤법 강의 - 한글 맞춤법/표준어 규정/외래어 표기법, 신구문화사, 1989

[19] 한국과학기술원, 국어정보베이스, 국어정보처리기술 개발 제 1 차년도 최종보고서, 1995

[20] A.Bies, M.Ferguson, K.Katz, R.MacIntyre, V.Tredinnick, G.Kim, M.A.Marcinkiewicz, and B. Schasberger, "Bracking Guidelines for Treebank II Stype Penn Treebank Project," ftp://ftp.cis.upenn.edu/pub/treebank/doc/manual/root.ps.gz

[21] EAGLES, "Preliminary Recommendations on subcategorisation," EAGLES Document EAG-CLWG-SYNLEX/P, ftp://ftp.ilc.pi.cnr.it/pub/eagles/lexicons/synlex.ps.gz

[22] EDR, EDR Electronic Dictionary Technical Guide, Japan Electronic Dictionary Research Institute Ltd., 1993

[23] R.Garside, G.Leech, and G.Sampson, "The Computational Analysis of English: a Corpus-Based Approach," LONGMAN, 1987

[24] G.A.Miller, R.Bechwith, C.Fellbaum, D.Gross, and K.Miller, "Introduction to WordNet: An On-line Lexical Databse," Report of WordNet, Princeton University, 1990
ftp://clarity.princeton.edu/pub/wordnet/5papers.tar

[25] M.P.Marcus, M.A.Marcinkiewicz, and B.Santorini, "Building Very Large Natural Language Corpora: the Penn Treebank," Computational Linguistics, vol.19, no.2, pp.313-330

[26] T.McEnery, A.Wilson, Corpus Linguistics, Edinburgh University Press.

[27] "List of Tags in the BNC Enriched Tagset," http://info.ox.ac.uk/bnc/c7spec.html

[28] TEI, "Guidelines for Electronic Text Encoding and Interchage," Electronic Text Center at the University of Virginia, http://etext.virginia.edu/TEI.html

[29] "The BNC Basic (C5) Tagset," http://info.ox.ac.uk/bnc/c5spec.html

[30] "UCREL CLAWS1 (LOB) Tagset,"
http://www.comp.lancs.ac.uk/computing/research/ucrel/claws1tags.html

[31] "UCREL CLAWS2 Tagset,"
http://www.comp.lancs.ac.uk/computing/research/ucrel/claws1tags.html

[32] "UCREL Skeleton Parsing Tagset,"
http://www.comp.lancs.ac.uk/computing/research/ucrel/skeletontags.html

[33] M.Wynne, "A Post-editor's Guide to CLAWS7 Tagging,"
http://www.comp.lancs.ac.uk/computing/users/eiamjw/claws/claws7.html

부록 A

UCREL CLAWS1 Tagset (LOB Tagset)

출처: http://www.comp.lancs.ac.uk/computing/research/ucrel/claws1tags.html

.	punctuation tag - full stop
...	punctuation tag - ellipsis
(	punctuation tag - left bracket
!	punctuation tag - exclamation mark
&FO	formula
&FW	foreign word
'	punctuation tag - close quotes
-	punctuation tag - dash
`	punctuation tag - open quotes
)	punctuation tag - right bracket
;	punctuation tag - semicolon
-----	punctuation tag - new sentence marker
,	punctuation tag - comma
?	punctuation tag - question mark
:	punctuation tag - colon
ABL	pre-qualifier
ABN	pre-quantifier
ABX	pre-quantifier/double conjunction
AP	post-determiner
AP$	genitive post-determiner
APS	plural post-determiner
APS$	genitive plural post-determiner
AT	singular article

ATI	article
BE	(be)
BED	(were)
BEDZ	(was)
BEG	(being)
BEM	(am)
BEN	(been)
BER	(are)
BEZ	(is)
CC	co-ordinating conjunction
CD	cardinal number
CD$	genitive cardinal number
CD-CD	hyphenated cardinal number
CDS	plural cardinal number
CDS$	genitive plural cardinal number
CD1	(one, 1)
CD1$	(one's)
CD1S	(ones)
CS	subordinating conjunction
DO	(do)
DOD	(did)
DOZ	(does)
DT	singular determiner
DT$	genitive singular determiner
DTI	singular or plural determiner
DTS	plural determiner
DTX	determiner/double conjunction
EX	existential THERE
HV	(have)
HVD	(had) (past tense)
HVG	(having)

HVN	(had) (past participle)
HVZ	(has)
IN	preposition
JJ	general adjective
JJB	attributive adjective
JJR	comparative adjective
JJT	superlative adjective
JNP	adjective with word initial capital
MD	modal verb
NC	cited word
NN	singular common noun
NN$	genitive singular common noun
NNP	singular common noun with word initial capital
NNP$	genitive singular common noun with word initial capital
NNPS	plural common noun with word initial capital
NNPS$	genitive plural common noun with word initial capital
NNS	plural common noun
NNS$	genitive plural common noun
NNU	abbreviated unit of measurement unmarked for number
NNU$	genitive abbreviated unit of measurement unmarked for number
NNUS	plural abbreviated unit of measurement
NNUS$	genitive plural abbreviated unit of measurement
NP	singular proper noun
NP$	genitive singular proper noun
NPL	singular locative noun with word initial capital
NPL$	genitive singular locative noun with word initial capital
NPLS	plural locative noun with word initial capital
NPLS$	genitive plural locative noun with word initial capital
NPS	plural proper noun
NPS$	genitive plural proper noun
NPT	singular titular noun with word initial capital

NPT$	genitive singular titular noun with word initial capital
NPTS	plural titular noun with word inital capital
NPTS$	genitive plural titular noun with word initial capital
NR	singular adverbial noun
NR$	genitive singular adverbial noun
NRS	plural adverbial noun
NRS$	genitive plural adverbial noun
OD	ordinal number
OD$	genitive ordinal number
PN	nominal pronoun
PN$	genitive nominal pronoun
PP$	prenominal possessive personal pronoun
PP$$	nominal possessive personal pronoun
PPL	singular reflexive personal pronoun
PPLS	plural reflexive personal pronoun
PP1A	(I)
PP1AS	(we)
PP1O	(me)
PP1OS	(us)
PP2	(you, thou, ye, thee)
PP3	(it)
PP3A	(he, she)
PP3AS	(they)
PP3O	(him, her)
PP3OS	(them)
QL	qualifier
QLP	post-qualifier
RB	adverb
RB$	genitive adverb
RBR	comparative adverb
RBT	superlative adverb

RI	adverb (homograph of preposition)
RN	nominal adverb
RP	adverb which can also be a particle
TO	infinitival TO
UH	interjection
VB	base form of lexical verb
VBD	past tense of lexical verb
VBG	present participle of lexical verb
VBN	past participle of lexical verb
VBZ	s-form of lexical verb
WDT	wh-determiner
WP	wh-pronoun, neutral between nominative and objective
WP$	possessive wh-pronoun
WPA	nominative wh-pronoun
WPO	objective wh-pronoun
WQL	wh-qualifier
WRB	wh-adverb
XNOT	
ZZ	letter(s) of the alphabet

부록 B

UCREL CLAWS2 Tagset

출처: http://www.comp.lancs.ac.uk/computing/research/ucrel/claws2tags.html

!	punctuation tag - exclamation mark
"	punctuation tag - quotation marks
$	germanic genitive marker - (' or 's)
&FO	formula
&FW	foreign word
(	punctuation tag - left bracket
)	punctuation tag - right bracket
,	punctuation tag - comma
-	punctuation tag - dash
-----	new sentence marker
.	punctuation tag - full-stop
...	punctuation tag - ellipsis
:	punctuation tag - colon
;	punctuation tag - semi-colon
?	punctuation tag - question-mark
APP$	possessive pronoun, pre-nominal
AT	article
AT1	singular article
BCS	before-conjunction
BTO	before-infinitive marker
CC	coordinating conjunction
CCB	coordinating conjunction
CF	semi-coordinating conjunction

CS	subordinating conjunction
CSA	'as' as a conjunction
CSN	'than' as a conjunction
CST	'that' as a conjunction
CSW	'whether' as a conjunction
DA	after-determiner (capable of pronominal function)
DA1	singular after-determiner
DA2	plural after-determiner
DA2R	comparative plural after-determiner
DAR	comparative after-determiner
DAT	superlative after-determiner
DB	before-determiner (capable of pronominal function)
DB2	plural before-determiner (capable of pronominal function)
DD	determiner (capable of pronominal function)
DD1	singular determiner
DD2	plural determiner
DDQ	wh-determiner
DDQ$	wh-determiner, genitive
DDQV	wh-ever determiner
EX	existential 'there'
ICS	preposition-conjunction
IF	'for' as a preposition
II	preposition
IO	'of' as a preposition
IW	'with'; 'without' as preposition
JA	predicative adjective
JB	attributive adjective
JBR	attributive comparative adjective
JBT	attributive superlative adjective
JJ	general adjective
JJ	general comparative adjective

JJT general superlative adjective

JK adjective catenative

LE leading co-ordinator

MC cardinal number neutral for number

MC$ genitive cardinal number, neutral for number

MC-MC hyphenated number

MC1 singular cardinal number

MC2 plural cardinal number

MD ordinal number

MF fraction, neutral for number

NC2 plural cited word

ND1 singular noun of direction

NN common noun, neutral for number

NN1 singular common noun

NN1$ genitive singular common noun

NN2 plural common noun

NNJ organization noun, neutral for number

NNJ1 singular organization noun

NNJ2 plural organization noun

NNL locative noun, neutral for number

NNL1 singular locative noun

NNL2 plural locative noun

NNO numeral noun, neutral for number

NNO1 singular numeral noun

NNO2 plural numeral noun

NNS noun of style, neutral for number

NNS1 singular noun of style

NNS2 plural noun of style

NNSA1 following noun of style or title, abbreviatory

NNSA2 following plural noun of style or title, abbreviatory

NNSB preceding noun of style or title, abbr.

NNSB1 preceding sing. noun of style or title, abbr.

NNSB2 preceding plur. noun of style or title, abbr.

NNT temporal noun, neutral for number

NNT1 singular temporal noun

NNT2 plural temporal noun

NNU unit of measurement, neutral for number

NNU1 singular unit of measurement

NNU2 plural unit of measurement

NP proper noun, neutral for number

NP1 singular proper noun

NP2 plural proper noun

NPD1 singular weekday noun

NPD2 plural weekday noun

NPM1 singular month noun

NPM2 plural month noun

PN indefinite pronoun, neutral for number

PN1 singular indefinite pronoun

PNQO whom

PNQS who

PNQV$ whosever

PNQVO whomever, whomsoever

PNQVS whoever, whosoever

PNX1 reflexive indefinite pronoun

PP$ nominal possessive personal pronoun

PPH1 it

PPHO1 him, her

PPHO2 them

PPHS1 he, she

PPHS2 they

PPIO1 me

PPIO2 us

PPIS1 I

PPIS2 we

PPX1 singular reflexive personal pronoun

PPX2 plural reflexive personal pronoun

PPY you

RA adverb, after nominal head

REX adverb introducing appositional constructions

RG degree adverb

RGA post-nominal/adverbial/adjectival degree adverb

RGQ wh- degree adverb

RGQV wh-ever degree adverb

RGR comparative degree adverb

RGT superlative degree adverb

RL locative adverb

RP prep. adverb; particle

RPK prep. adv., catenative

RR general adverb

RRQ wh- general adverb

RRQV wh-ever general adverb

RRR comparative general adverb

RRT superlative general adverb

RT nominal adverb of time

TO infinitive marker

UH interjection

VB0 be

VBDR were

VBDZ was

VBG being

VBM am

VBN been

VBR are

VBZ is

VD0 do

VDD did

VDG doing

VDN done

VDZ does

VH0 have

VHD had (past tense)

VHG having

VHN had (past participle)

VHZ has

VM modal auxiliary

VMK modal catenative

VV0 base form of lexical verb

VVD past tense form of lexical verb

VVG -ing form of lexical verb

VVN past participle form of lexical verb

VVZ -s form of lexical verb

VVGK -ing form in a catenative verb

VVNK past part. in a catenative verb

XX not, n't

ZZ1 singular letter of the alphabet

ZZ2 plural letter of the alphabet

부록 C

UCREL CLAWS7 Tagset

출처: http://www.comp.lancs.ac.uk/computing/users/eiamjw/claws/claws7.html

A Post-editor's Guide to CLAWS7 Tagging

!	punctuation tag - exclamation mark
?	punctuation tag - quotation marks
(	punctuation tag - left bracket
)	punctuation tag - right bracket
,	punctuation tag - comma
-	punctuation tag - dash
-----	new sentence marker
.	punctuation tag - full-stop
...	punctuation tag - ellipsis
:	punctuation tag - colon
;	punctuation tag - semi-colon
?	punctuation tag - question-mark
APPGE	possessive pronoun, prenominal
AT	article
AT1	singular article
BCS	before-conjunction
BTO	before-infinitive marker
CC	coordinating conjunction
CCB	coordinating conjunction
CS	subordinating conjunction
CSA	*as* as a conjunction
CSN	*than* as a conjunction
CST	*that* as a conjunction

CSW	*whether* as a conjunction
DA	after-determiner, capable of pronominal function
DA1	singular after-determiner
DA2	plural after-determiner
DAR	comparative after-determiner
DAT	superlative after-determiner
DB	before-determiner, capable of pronominal function
DB2	plural before-determiner, capable of pronominal function
DD	determiner, capable of pronominal function
DD1	singular determiner
DD2	plural determiner
DDQ	wh-determiner
DDQGE	wh-determiner, genitive
DDQV	wh-ever determiner
EX	existential *there*
FO	formula
FU	unclassified
FW	foreign word
GE	germanic genitive marker - (*'* or *'s*)
IF	*for* as a preposition
II	preposition
IO	*of* as a preposition
IW	*with*; *without* as preposition
JJ	general adjective
JJR	Rgeneral comparative adjective
JJT	general superlative adjective
JK	adjective catenative
MC	cardinal number neutral for number
MCGE	genitive cardinal number, neutral for number
MCMC	hyphenated number
MC1	singular cardinal number
MC2	plural cardinal number
MD	ordinal number
MF	fraction
ND1	singular noun of direction

NN	common noun, neutral for number
NNA	following noun of title
NNB	preceding noun of title
NN1	singular common noun
NN2	plural common noun
NNL1	singular locative noun
NNL2	plural locative noun
NNO	numeral noun, neutral for number
NNO2	plural numeral noun
NNT	temporal noun, neutral for number
NNT1	singular temporal noun
NNT2	plural temporal noun
NNU	unit of measurement, neutral for number
NNU1	singular unit of measurement
NNU2	plural unit of measurement
NP	proper noun, neutral for number
NP1	singular proper noun
NP2	plural proper noun
NPD1	singular weekday noun
NPD2	plural weekday noun
NPM1	singular month noun
NPM2	plural month noun
PN	indefinite pronoun, neutral for number
PN1	singular indefinite pronoun
PNQO	*whom*
PNQS	*who*
PNQV	*whoever*, *whomever*, *whomsoever*, *whosoever*
PNX1	reflexive indefinite pronoun
PP	nominal possessive personal pronoun
PPH1	*it*
PPHO1	*him*, *her*
PPHO2	*them*
PPHS1	*She*, *she*
PPHS2	*they*
PPIO1	*me*

PPIO2	*us*
PPIS1	*I*
PPIS2	*we*
PPX1	singular reflexive personal pronoun
PPX2	plural reflexive personal pronoun
PPY	*you*
RA	adverb, after nominal head
REX	adverb introducing appositional constructions
RG	degree adverb
RGA	post-nominal/adverbial/adjectival degree adverb
RGQ	wh- degree adverb
RGQV	wh-ever degree adverb
RGR	comparative degree adverb
RGT	superlative degree adverb
RL	locative adverb
RP	prep. adverb; particle
RPK	prep. adv., catenative
RR	general adverb
RRQ	wh- general adverb
RRQV	wh-ever general adverb
RRR	comparative general adverb
RRT	superlative general adverb
RT	nominal adverb of time
TO	infinitive marker
UH	interjection
VB0	*be*
VBDR	*were*
VBDZ	*was*
VBG	*being*
VBM	*am*
VBN	*been*
VBR	*are*
VBZ	*is*
VD0	*do*
VDD	*did*

VDG	*doing*
VDN	*done*
VDZ	*does*
VH0	*have*
VHD	*had* (past tense)
VHG	*having*
VHN	*had* (past participle)
VHZ	*has*
VM	modal auxiliary
VMK	modal catenative
VV0	base form of lexical verb
VVD	past tense form of lexical verb
VVG	-ing form of lexical verb
VVN	past participle form of lexical verb
VVZ	-s form of lexical verb
VVGK	-ing form in a catenative verb
VVNK	past part. in a catenative verb
XX	*not, n't*
ZZ1	singular letter of the alphabet
ZZ2	plural letter of the alphabet

부록 D

The BNC Basic Tagset (C5 Tagset)

출처: http://info.ox.ac.uk/bnc/c5spec.html

A users guide to the Grammatical Tagging of the BNC

앞의 두 자는 일반적인 문법범주를 나타내고 세번째 자는 세부범주를 나타낸다. 일반적이며 대표적인 세부범주는 0 으로 나타낸다.

AJ0	Adjective (general or positive)
AJC	Comparative adjective
AJS	Superlative adjective
AT0	Article
AV0	General adverb: an adverb not subclassified as AVP or AVQ
AVP	Adverb particle
AVQ	*Wh*-adverb
CJC	Coordinating conjunction
CJS	Subordinating conjunction
CJT	The subordinating conjunction *that*
CRD	Cardinal number
DPS	Possessive determiner
DT0	General determiner: i.e. a determiner which is not a DTQ.
DTQ	*Wh*-determiner
EX0	Existential *there*
ITJ	Interjection or other isolate
NN0	Common noun, neutral for number
NN1	Singular common noun
NN2	Plural common noun

NP0	Proper noun
ORD	Ordinal numeral
PNI	Indefinite pronoun
PNP	Personal pronoun
PNQ	*Wh*-pronoun
PNX	Reflexive pronoun
POS	The possessive or genitive marker *'s* or *'*
PRF	The preposition *of*
PRP	Preposition (except for *of*)
PUL	Punctuation: left bracket - i.e. (or [
PUN	Punctuation: general separating mark - i.e. . , ! , : ; - or ?
PUQ	Punctuation: quotation mark - i.e. ' or "
PUR	Punctuation: right bracket - i.e.) or]
TO0	nfinitive marker *to*
UNC	Unclassified items which are not appropriately classified as items of the English lexicon.
VBB	The present tense forms of the verb BE, except for *is, 's*
VBD	The past tense forms of the verb BE: *was* and *were*
VBG	The *-ing* form of the verb BE: *being*
VBI	The infinitive form of the verb BE: *be*
VBN	The past participle form of the verb BE: *been*
VBZ	The *-s* form of the verb BE: *is, 's*
VDB	The finite base form of the verb BE: *do*
VDD	The past tense form of the verb DO: *did*
VDG	The *-ing* form of the verb DO: *doing*
VDI	The infinitive form of the verb DO: *do*
VDN	The past participle form of the verb DO: *done*
VDZ	The *-s* form of the verb DO: *does, 's*
VHB	The finite base form of the verb HAVE: *have, 've*
VHD	The past tense form of the verb HAVE: *had, 'd*
VHG	The *-ing* form of the verb HAVE: *having*
VHI	The infinitive form of the verb HAVE: *have*

VHN	The past participle form of the verb HAVE: *had*
VHZ	The *-s* form of the verb HAVE: *has, 's*
VM0	Modal auxiliary verb
VVB	The finite base form of lexical verbs
VVD	The past tense form of lexical verbs
VVG	The *-ing* form of lexical verbs
VVI	The infinitive form of lexical verbs
VVN	The past participle form of lexical verbs
VVZ	The *-s* form of lexical verbs
XX0	The negative particle *not* or *n't*
ZZ0	Alphabetical symbols

부록 E

The BNC Basic Enriched Tagset (C7 Tagset)

출처: http://info.ox.ac.uk/bnc/c7spec.html

A users guide to the Grammatical Tagging of the BNC

200 만 단어의 영어 코퍼스를 태깅하기 위한 태그세트이다. Brown Corpus 나 LOB Corpus 의 태그세트와 유사하다.

AT1	Singular article
BCS	"Before-conjunction"
BTO	"Before-infinitive-marker"
CC	Coordinating conjunction, general
CCB	Coordinating conjunction *but*
CS	Subordinating conjunction, general
CSA	*As* as conjunction
CSN	*Than* as conjunction
CST	*That* as conjunction
CSW	The conjunction *whether*, or *if* when it is equivalent in function to *whether*.
DA	"After-determiner" (or postdeterminer), neutral for number
DA1	Singular "after-determiner" (or postdeterminer)
DA2	Plural "after-determiner" (or postdeterminer)
DA2R	Plural "after-determiner", comparative form
DA2T	Plural "after-determiner", superlative form
DAR	Comparative "after-determiner", neutral for number
DAT	Superlative "after-determiner", neutral for number
DB	"Before-determiner" (or predeterminer), neutral for number
DB2	Plural "before-determiner" (or predeterminer)
DD	Central determiner, neutral for number

DD1	Singular central determiner
DD2	Plural central determiner
DDQ	*Wh*-determiner
DDQGE	*Wh*-determiner, possessive
DDQV	*Wh-ever* determiner
EX	Existential *there*
IF	*For* as a preposition
II	Preposition (general)
IO	*Of* as a preposition
IW	*With* and *without* as prepositions
JJ	Adjective (general or positive)
JJR	General comparative adjective
JJT	General superlative adjectives
JK	Catenative adjective (with a quasi-auxiliary function
LE	"Leading coordinator": a word introducing correlative coordination
MC	Cardinal number, neutral for number
MC-MC	Two numbers linked by a hyphen or dash
MC1	Singular cardinal number
MC2	Plural cardinal number
MD	Ordinal number
MF	Fractional number, neutral for number
ND1	Singular noun of direction
NN	Common noun, neutral for number
NN1	Singular common noun
NN2	Plural common noun
NNJ	Human organization noun
NNJ2	Plural human organization noun
NNL	Locative noun, neutral for number
NNL1	Singular locative noun
NNL2	Plural locative noun
NNO	Numeral noun, neutral for number (cf. MC above)

NNO2	Plural numeral noun
NNSA	Noun of style or title, following a name
NNSB	Noun of style or title, preceding a name
NNT1	Singular temporal noun
NNT2	Plural temporal nouns
NNU	Unit-of-measurement noun, neutral for number
NNU1	Singular unit-of-measurement noun
NNU2	Plural unit-of-measurement noun
NP	Proper noun, neutral for number
NP1	Singular proper noun
NP2	Plural proper noun
NPD1	Singular weekday noun
NPD2	Plural weekday noun
NPM1	Singular month noun
NPM2	Plural month noun
PN	Indefinite pronoun, neutral for number
PN1	Singular indefinite pronoun
PNQO	*Wh*-pronoun, objective case (*whom*)
PNQS	*Wh*-pronoun, subjective case (*who*)
PNQVS	*Wh-ever* pronoun, subjective case (*whoever*)
PNX1	Reflexive indefinite pronoun, singular (*oneself*)
PP$	Nominal possessive pronoun
PPH1	Singular personal pronoun, third person (*it*)
PPHO1	Singular personal pronoun, third person, objective case (*him, her*)
PPHO2	Plural personal pronoun, third person, objective case (*them*)
PPHS1	Singular personal pronoun, third person, subjective case (*he, she*)
PPHS2	Plural personal pronoun, third person, subjective case (*they*)
PPIO1	Singular personal pronoun, first person, objective case (*me*)
PPIO2	Plural personal pronoun, first person, objective case (*us*)
PPIS1	Singular personal pronoun, first person, subjective case (*I*)
PPIS2	Plural personal pronoun, first person, subjective case (*we*)

PPX1	Singular reflexive pronoun (*myself, yourself, herself*)
PPX2	Plural reflexive pronoun (*ourselves, yourselves, themselves*)
PPY	Second person personal pronoun (*you*)
RA	Adverb, after nominal head
REX	Adverb introducing appositional constructions
RG	Positive degree adverb
RGA	Post-modifying positive degree adverb
RGQ	*Wh-* degree adverb
RGQV	*Wh-ever* degree adverb
RGR	Comparative degree adverb
RGT	Superlative degree adverb
RL	Locative adverb
RP	Adverbial particle
RPK	Catenative adverbial particle
RR	General positive adverb
RRQ	*Wh-* general adverb
RRQV	*Wh-ever* general adverb
RRR	Comparative general adverb
RRT	Superlative general adverb
RT	Nominal adverb of time
TO	The infinitive marker *to*
UH	Interjection, or other isolate
VB0	*be* as a finite form
VBDR	*were*
VBDZ	*was*
VBG	*being*
VBI	*be* as an infinitive form
VBM	*am, 'm*
VBN	*been*
VBR	*are, 're*
VBZ	*is, 's*

VD0	*do* as a finite form (in declarative and imperative clauses)
VDD	*did*
VDG	*doing*
VDI	*do* as an infinitive form
VDN	*done*
VDZ	*does, 's*
VH0	*have, 've* as a finite form (in declarative and imperative clauses)
VHD	*had, 'd* as a past tense form
VHG	*doing*
VHI	*have* as an infinitive form
VHN	*had* as a past participle
VHZ	*has, 's*
VM	Modal auxiliary verb
VMK	Catenative modal auxiliary
VV0	The base form of the lexical verb as a finite form (in declarative and imperative clauses)
VVD	The past tense form of the lexical verb
VVG	The *-ing* form of the lexical verb
VVGK	The *-ing* form as a catenative verb
VVI	The base form of the lexical verb as an infinitive
VVN	The past participle form of the lexical verb
VVNK	The past participle as a catenative ver
VVZ	The *-s* form of the lexical verb
XX	*not, -n't*
ZZ1	Singular letter of the alphabet
ZZ2	Plural letter of the alphabet

부록 F

UCREL Skeleton Parsing Tagset

출처: http://www.comp.lancs.ac.uk/computing/research/ucrel/skeletontags.html

Fa	Adverbial Clause
Fc	Comparative Clause
Fn	Noun Clause
Fr	Relative Clause
G	Genitive
J	Adjective Phrase (predicative)
N	Noun Phrase
Nn	Metalinguistic Constituent
Nr	Adverbial Noun Phrase (temporal)
Nv	Adverbial Noun Phrase (non-temporal) (not in AP or SEC corpora)
P	Prepositional Phrase
S	Sentence (used eg in quoted speech, also with + and & as co-ordinates)
Si	Interpolated or appended sentences
Tg	-ing Clause
Ti	Infinitive Clause
Tn	Past Participle Clause
V	Verb Phrase
&	First conjunt
+	Second, (etc.) conjuct
@	Discontinuity marker

부록 G

Penn Treebank Tagset

출처: ftp://ftp.cis.upenn.edu/pub/treebank/doc/manual/root.ps.gz

Bracketing Guidelines for Treebank II Style Penn Tree Bank Project

■ Syntactic Tags

1. Bracket labels

1.1 Clause level

S Simple declarative clause

SBAR Clause introduced by a (possibly empty) subordinating conjunction.

SBARQ Direct question introduced by a *wh*-word or *wh*-phrase.

SINV Inverted declarative sentence

SQ Inverted yes/no question, or main clause of a *wh*-question

1.2 Phrase level

ADJP Adjective Phrase

ADVP Adverb Phrase

CONJP Conjunction Phrase

FRAG Fragment

INTJ Interjection

LST List marker

NAC Not A Constituent

NP Noun Phrase

NX Used within certain complex noun phrases to mark the head of the noun phrase

PP Prepositional Phrase

PRN Parenthetical

PRT Particle

QP Quantifier Phrase

RRC Reduced Relative Clause

UCP Unlike Coordinated Phrase

VP Verb Phrase

WHADJP *Wh*-adjective Phrase

WHADVP *Wh*-adverb Phrase

WHNP *Wh*-noun Phrase

WHPP *Wh*-prepositional Phrase

X Unknown, uncertain, or unbracketable

2. Function tags

2.1 Form/function discrepancies

-ADV adverbial

-NOM nominal

2.2 Grammatical role

-DTV dative

-LGS logical subject

-PRD predicate

-PUT put

-SBJ surface subject

-TPC topicalized

-VOC vocative

2.3 Adverbials

-BNF benefactive

-DIR direction

-EXT extent

-LOC locative

-MNR manner

-PRP purpose or reason

-TMP temporal

2.4 Miscellaneous

-CLR closely related

-CLF cleft

-HLN headline

-TTL title

3. Null elements

T trace of A'-movement

(NP *) arbitrary PRO, controlled PRO, and trace of A-movement

0 the null complementizer

U unit

? placeholder for ellipsed material

NOT anti-placeholder in template gapping

4. Pseudo-attach

EXP Expletive (extraposition)

ICH Interpret Constituent Here (discontinuous dependency)

PPA Permanent Predictable Ambiguity (ambiguity)

RNR Right Node Raising (shared complements)

■ Part of Speech Tags

CC Coordinating conjunction

CD Cardinal Number

DT Determiner

EX Existential *there*

FW Foreign word

IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	predeterminer
POS	Possesive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superative
RP	Particle
SYM	Symbol
TO	*to*
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past particle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possesive wh-pronoun
WRB	Wh-adverb

부록 H

SUSANNE Tagset

출처: ftp://ota.ox.ac.uk/pub/ota/public/susanne/

1. Syntactic Lable

1.1 Rootlevel Formtags

O	paragraph
Oh	heading
Ot	title (e.g. of book)
Q	quotation
I	interpolation
Iq	tag question
Iu	scientific citation

1.2 Clauselevel Formtags

S	main clause
Ss	quoting clause embedded within quotation
Fa	adverbial clause
Fn	nominal clause
Fr	relative clause
Ff	"fused" relative
Fc	comparative clause
Tg	present participle clause
Ti	infinitival clause

Tn	past participle clause
Tf	"for-to" clause
Tb	"bare" nonfinite clause
Tq	infinitival relative clause
Z	reduced ("whiz-deleted") relative clause
L	other verbless clause
A	special "as" clause
W	"with" clause

1.3 Phraselevel Formtags

N	noun phrase
V	verb group
J	adjective phrase
R	adverb phrase
P	prepositional phrase
D	determiner phrase
M	numeral phrase
G	genitive phrase

1.4 Symbols

?	interrogative clause
*	imperative clause
%	subjunctive clause
!	exclamatory clause or other item
"	vocative item

1.5 Coordination

+	subordinate conjunct introduced by conjunction

- subordinate conjunct not introduced by conjunction

@ appositional element

& co-ordinate structure acting as first conjunct within a higher co-ordination

2. Function Tag

2.1 Complement Functiontags

s logical subject

o logical direct object

S surface (and not logical) subject

O surface (and not logical) direct object

i indirect object

u prepositional object

e predicate complement of subject

j predicate complement of object

a agent of passive

n particle of phrasal verb

z complement of catenative

x relative clause having higher clause as antecedent

G "guest" having no grammatical role within its tagma

2.2 Adjunct Funtiontags

p place

q direction

t time

h manner or degree

m modality

c contingency

r respect

w	comitative
k	benefactive
b	absolute

부록 I

WordNet Lexical Category

출처: ftp://clarity.princeton.edu/pub/wordnet/5papers.tar

Design and Implementation of WordNet Lexical Database and

Searching Software

noun.act	nouns denoting acts or actions
noun.anmal	nouns denoting animals
noun.artifact	nouns denoting man-made object
noun.attribute	nouns denoting attributes of people and objects
noun.bodynouns denoting body parts	
noun.cognition	nouns denoting cognitive processes and contents
noun.communication	nouns denoting communicative processes and contents
noun.event	nouns denoting natural events
noun.feeling	nouns denoting feelings and emotions
noun.food	nouns denoting foods and drinks
noun.group	nouns denoting groupings of people or objects
noun.location	nouns denoting spatial position
noun.motive	nouns denoting goals
noun.object	nouns denoting natural objects (not man-made)
noun.person	nouns denoting people
noun.phenomenon	nouns denoting natural phenomena
noun.plantnouns denoting natural phenomena	
noun.possession	nouns denoting possession and transfer of possession
noun.process	nouns denoting natural processes
noun.quantity	nouns denoting quantities and units of measure
noun.relation	nouns denoting relations between people or things or ideas

noun.shpae	nouns denoting stable states of affairs
noun.substance	nouns denoting substances
noun.time	nouns denoting time and temporal relations
verb.body	verbs of grooming, dressing and bodily care
verb.change	verbs of change of size, temperature, intensity, etc.
verb.cognition	verbs of thinking, judging, analyzing, doubting, etc.
verb.communication	verbs of telling, asking, ordering, singing, etc.
verb.competition	verbs of fighting, athletic activities, etc.
verb.consumption	verbs of eating and drinking
verb.contact	verbs of touching, hitting, tying, digging, etc.
verb.creation	verbs of sewing, baking, painting, performing, etc.
verb.emotion	verbs of feeling
verb.motion	verbs of walking, flying, swimming, etc.
verb.perception	verbs of seeing, hearing, feeling, etc.
verb.possession	verbs of buying, selling, owning, and transfer
verb.social	verbs of political and social activities and events
verb.stative	verbs of being, having, spatial relations
verb.weather	verbs of raining, snowing, thawing, thundering, etc.
adj.all	all adjective clusters
adj.pert	relational adjectives (pertainyms)
adv.all	all adverbs

부록 J

국어 형태·통사 태그 규격

상위 분류			태그	
기호(s)			sp 쉼표	sf 마침표
			sl 여는 따옴표 및 묶음표	sr 닫는 따옴표 및 묶음표
			sd 이음표	se 줄임표
			su 단위 기호	sy 기타 기호
외국어(f)			f 외국어	
체언(n)	보통명사(nc)	서술성 명사(ncp)	ncpa 동작성 명사	ncps 상태성 명사
		비서술성 명사(ncn)	ncn 비서술성 명사	
	고유명사(nq)		nq 고유명사	
	의존명사(nb)		nbu 단위성 의존명사	nbn 비단위성 의존명사
	대명사(np)		npp 인칭대명사	npd 지시대명사
	수사(nn)		nnc 양수사	nno 서수사
용언(p)	동사(pv)		pvd 지시 동사	pvg 일반 동사
	형용사(pa)		pad 지시형용사	paa 성상형용사
	보조용언(px)		px 보조용언	
수식언(m)	관형사(mm)		mmd 지시관형사	mma 성상관형사
	부사(ma)		mad 지시부사	maj 접속부사
			mag 일반부사	
독립언(i)	감탄사(ii)		ii 감탄사	

관계언(j)	격조사(jc)	jcs 주격조사 jco 목적격조사 jcc 보격조사 jcm 관형격조사 jcv 호격조사 jca 부사격조사 jcj 접속격조사 jct 공동격조사 jcr 인용격조사
	보조사(jx)	jxc 통용보조사 jxf 종결보조사
	서술격조사(jp)	jp 서술격조사
어미(e)	선어말어미(ep)	ep 선어말어미
	연결어미(ec)	ecc 대등적 연결어미 ecs 종속적 연결어미 ecx 보조적 연결어미
	전성어미(et)	etn 명사형 전성어미 etm 관형사형 전성어미
	종결어미(ef)	ef 종결어미
접사(x)	접두사(xp)	xp 접두사
	접미사(xs)	xsn 명사파생접미사 xsv 동사파생접미사 xsm 형용사파생접미사 xsa 부사파생접미사

부록 K

형태소 생성(기계번역)을 위한 품사 태그 (좌,우접속정보)

체언

국어 형태·품사 태그 규격			좌접속정보	우접속정보
보통명사(nc)	서술성명사(ncp)	동작성명사(ncpa)	동작성보통명사(ncpa)	무종성(ncpa_n)
				유종성(ncpa_y)
				ㄹ종성(ncpa_l)
		상태성명사(ncps)	상태성보통명사(ncps)	무종성(ncps_n)
				유종성(ncps_y)
				ㄹ종성(ncps_l)
	비서술성명사(ncn)	비서술성명사(ncn)	비서술성명사(ncn)	무종성(ncn_n)
				유종성(ncn_y)
				ㄹ종성(ncn_l)
고유명사(nq)	고유명사(nq)		고유명사(nq)	무종성(nq_n)
				유종성(nq_y)
				ㄹ종성(nq_l)
의존명사(nb)	단위성 의존명사(nbu)		단위성 의존명사(nbu)	무종성(nbu_n)
				유종성(nbu_y)
				ㄹ종성(nbu_l)
	비단위성의존명사(nbn)	서술성비단위성의존명사(nbn_p)		무종성(nbn_p_n)
				유종성(nbn_p_y)
				ㄹ종성(nbn_p_l)
		비서술성단위성의존명사(nbn_n)		무종성(nbn_n_n)
				유종성(nbn_n_y)
				ㄹ종성(nbn_n_l)

대명사(np)	인칭대명사(npp)	인칭대명사(npp)	무종성(npp_n)
			유종성(npp_y)
			ㄹ종성(npp_l)
	지시대명사(npd)	지시대명사(npd)	무종성(npd_n)
			유종성(npd_y)
			ㄹ종성(npd_l)
수사(nn)	양수사(nnc)	양수사(nnc)	무종성(nnc_n)
			유종성(nnc_y)
			ㄹ종성(nnc_l)
	서수사(nno)	서수사(nno)	무종성(nno_n)
			유종성(nno_y)
			ㄹ종성(nno_l)

수식언

국어 형태·품사 태그 규격		좌접속정보	우접속정보
관형사(mm)	지시관형사(mmd)	지시관형사(mmd)	지시관형사(mmd)
	성상관형사(mma)	성성관형사(mma)	성성관형사(mma)
		수관형사(mmn)	수관형사(mmn)
부사(ma)	지시부사(mad)	지시부사(mad)	
	일반부사(mas)	일반부사(mas)	무종성(mas_n)
			유종성(mas_y)
			ㄹ종성(mas_l)
	접속부사(maj)	접속부사(maj)	

용언

국어 형태·통사 태그 규격		좌접속정보	우접속정보	
동사(pv)	지시 동사(pvd)	동사(pv)	규칙 (pv_rg)	무종성/양성 (pv_rg_n_b)
	일반 동사(pvg)	보조동사(pvx)		무종성/음성 (pv_rg_n_d)
				유종성/양성 (pv_rg_y_b)
				유종성/음성 (pv_rg_y_d)
			ㅅ불규칙 (pv_s)	유종성/양성 (pv_s_y_b)
				유종성/음성 (pv_s_y_d)
			ㄷ불규칙 (pv_d)	유종성/양성 (pv_d_y_b)
				유종성/음성 (pv_d_y_d)
			ㅂ불규칙 (pv_b)	유종성/양성 (pv_b_y_b)
				유종성/음성 (pv_b_y_d)
			ㄹ불규칙 (pv_reu)	무종성/양성 (pv_reu_n_b)
				무종성/음성 (pv_reu_n_d)
			우불규칙 (pv_woo)	무종성/음성 (pv_woo_n_b)

			여불규칙 (pv_yeo)	무종성/양성 (pv_yeo_n_b)
			러불규칙 (pv_reo)	무종성/음성 (pv_reo_n_d)
			거라 (pv_geo)	무종성/양성 (pv_geo_n_b)
			너라 (pv_neo)	무종성/양성 (pv_neo_n_b)
			으탈락 (pv_eu)	무종성/양성 (pv_eu_n_b)
				무종성/음성 (pv_eu_n_d)
			ㄹ탈락 (pv_l)	ㄹ종성/양성 (pv_l_l_b)
				ㄹ종성/음성 (pv_l_l_b)
형용사 (pa)	지시형용사 (pad)	형용사(pa) 보조형용사 (pax)	규칙 (pa_rg)	무종성/양성 (pa_rg_n_b)
	성상형용사 (paa)			무종성/음성 (pa_rg_n_d)

	유종성/양성 (pa_rg_y_b)
	유종성/음성 (pa_rg_y_d)
ㅅ불규칙 (pa_s)	유종성/양성 (pa_s_y_b)
	유종성/음성 (pa_s_y_d)
ㅂ불규칙 (pa_b)	유종성/양성 (pa_b_y_b)
	유종성/음성 (pa_b_y_d)
ㄹ불규칙 (pa_reu)	무종성/양성 (pa_reu_n_b)
	무종성/음성 (pa_reu_n_d)
여불규칙 (pa_yeo)	무종성/양성 (pa_yeo_n_b)
러불규칙 (pa_reo)	무종성/음성 (pa_reo_n_d)
ㅎ불규칙 (pa_h)	유종성/양성 (pa_h_y_b)
	유종성/음성 (pa_h_y_d)
으탈락 (pa_eu)	무종성/양성 (pa_eu_n_b)
	무종성/음성 (pa_eu_n_d)

			ㄹ탈락 (pa_l)	ㄹ종성/양성 (pa_l_l_b)
				ㄹ종성/음성 (pa_l_l_d)
보조용언(px)	보조용언(px)			

독립언

국어 형태·통사 태그 규격		좌접속정보	우접속정보
독립어(i)	감탄사(i)	감탄사(i)	감탄사(ie)

관계언

국어 형태·통사 태그 규격		좌접속정보		우접속정보
격조사(jc)	주격조사(jcs)	격조사(jc)	무종성형(jc_n)	무종성(jc_n)
	목적격조사(jco)		유ㄹ종성형(jc_yl)	유종성(jc_y)
	보격조사(jcc)		무ㄹ종성형(jc_nl)	ㄹ종성(jc_l)
	관형격조사(jcm)		유종성형(jc_y)	
	호격조사(jcv)		공용형(jc_c)	
	부사격조사(jca)			
	인용격조사(jcr)			
	접속격조사(jcj)			
	공동격조사(jct)			
보조사(jx)	통용보조사(jxc)	보조사(jx)	무종성형(jx_n)	무종성(jx_n)
	종결보조사(jxf)		유ㄹ종성형(jx_yl)	유종성(jx_y)
			공용형(jx_c)	ㄹ종성(jx_l)
			종결어미형(jx_f)	

서술 격조 사(jp)	서술격조사(jp)	서술격조사(jp)	서술격조사(jp)

어미

국어 형태·품사 태그 규격		좌접속정보		우접속정보
선어말어미 (ep)	선어말어미(ep)	존칭 (ep_h)	무종성형 (ep_h_n)	무종성/음성 (ep_h_n_d)
			유ㄹ종성형 (ep_h_yl)	
		과거시제 (ep_p)	양성모음형 (ep_p_b)	유종성/양성 (ep_p_y_b)
			음성모음형 (ep_p_d)	유종성/음성 (ep_p_y_d)
		미래시제(ep_f)		유종성/음성 (ep_f_y_d)
		회상시제(ep_r)		무종성/음성 (ep_r_n_d)
연결어미(ec)	대등적 연결어미(ecc)	대등적 (ecc)	무ㄹ종성형 (ecc_nl)	무종성 (ecc_n)
			유종성형 (ecc_y)	유종성(ecc_y)
			공용형 (ecc_c)	
	종속적 연결어미(ecs)	종속적 (ecs)	무종성형 (ecs_n)	무종성 (ecc_n)
			유ㄹ종성형 (ecs_yl)	유종성(ecc_y)

			무ㄹ종성형 (ecs_nl)	
			유종성형 (ecs_y)	
			ㄹ종성형 (ecs_l)	
			양성모음형 (ecs_b)	
			음성모음형 (ecs_d)	
			공용형 (ecs_c)	
	보조적 연결어미(ecx)	보조적 (ecx)	양성모음형 (ecx_b)	무종성 (ecx_n)
			음성모음형 (ecx_d)	
			공용형 (ecx_c)	
선어말어미 (ep)	선어말어미(ep)	존칭 (ep_h)	무종성형 (ep_h_n)	무종성/음성 (ep_h_n_d)
			유ㄹ종성형 (ep_h_yl)	
		과거시제 (ep_p)	양성모음형 (ep_p_b)	유종성/양성 (ep_p_y_b)
			음성모음형 (ep_p_d)	유종성/음성 (ep_p_y_d)
		미래시제(ep_f)		유종성/음성 (ep_f_y_d)
		회상시제(ep_r)		무종성/음성 (ep_r_n_d)

연결어미(ec)	대등적 연결어미(ecc)	대등적 (ecc)	무ㄹ종성형 (ecc_nl)	무종성 (ecc_n)
			유종성형 (ecc_y)	유종성(ecc_y)
			공용형 (ecc_c)	
	종속적 연결어미(ecs)	종속적 (ecs)	무종성형 (ecs_n)	무종성 (ecc_n)
			유ㄹ종성형 (ecs_yl)	유종성(ecc_y)
			무ㄹ종성형 (ecs_nl)	
			유종성형 (ecs_y)	
			ㄹ종성형 (ecs_l)	
			양성모음형 (ecs_b)	
			음성모음형 (ecs_d)	
			공용형 (ecs_c)	
	보조적 연결어미(ecx)	보조적 (ecx)	양성모음형 (ecx_b)	무종성 (ecx_n)
			음성모음형 (ecx_d)	
			공용형 (ecx_c)	
전성어미(et)	명사형 어미(etn)	명사형 (ctn)	무ㄹ종성형	무종성 (etn_n)

			유종성형	유종성 (etn_y)
			공용형	
	관형사형 어미(etm)	관형사형 (etm)	유종성/현재/동사형 (etm_y_r_pv)	
			유종성/현재/형용사형 (etm_y_r_pa)	
			유종성/과거/동사형 (etm_y_p_pv)	
			유종성/과거/형용사형 (etm_y_p_pa)	
			유종성/미래 (etm_y_f)	
			무ㄹ종성/현재/동사형 (etm_nl_r_pv)	
			무ㄹ종성/현재/형용사형 (etm_nl_r_pa)	
			무ㄹ종성/과거/동사형 (etm_nl_p_pv)	
			무ㄹ종성/과거/형용사형 (etm_nl_p_pa)	

			무종성/미래형 (etm_n_f)	
			ㄹ종성/미래형 (etm_l_f)	
종결어미(ef)	종결어미(ef)	평서형 (ef_d)	동사/어간/무ㄹ종성형 (ef_d_pv_st_nl)	평서형 (ef_d)
			동사/어간/유종성형 (ef_d_pv_st_y)	
			동사/선어말/무종성형 (ef_d_pv_ep_n)	
			동사/선어말/유종성형 (ef_d_pv_ep_y)	
			형용사/무ㄹ종성형 (ef_d_pa_nl)	
			형용사/유종성형 (ef_d_pa_y)	
		감탄형 (ef_e)	동사/과거/미래형 (ef_e_pv_p_f)	감탄형 (ef_e)
			동사/무ㄹ종성형 (ef_e_pv_nl)	
			동사/유종성형 (ef_e_pv_y)	

			형용사/무ㄹ종성형 (ef_e_pa_nl)	
			형용사/유종성형 (ef_e_pa_y)	
		의문형 (ef_q)	동사/무ㄹ종성형 (ef_q_pv_nl)	의문형 (ef_q)
			동사/유종성형 (ef_q_pv_y)	
			형용사/무ㄹ종성형 (ef_q_pa_nl)	
			형용사/유종성형 (ef_q_pa_y)	
		명령형 (ef_i)	동사/양성모음형 (ef_i_pv_b)	명령형 (ef_i)
			동사/음성모음형 (ef_i_pv_d)	
			동사/무ㄹ종성형 (ef_i_pv_nl)	
			동사/유종성형 (ef_i_pv_y)	
			동사/존칭형 (ef_i_pv_h)	

		청유형 (ef_p)	동사/무ㄹ종성형 (ef_p_pv_nl)	청유형 (ef_p)
			동사/유종성형 (ef_p_pv_y)	

접사

국어 형태소 태그 규격		좌접속정보	우접속정보
접두사 (xp)	접두사(xp)	접두사(xp)	접두사(xp)
접미사 (xs)	명사파생접미사 (xsn)	명사피생접미사(xsn)	무종성(xsn_n)
			유종성(xsn_y)
			ㄹ종성(xsn_l)
	동사파생접미사 (xsv)	동사파생접미사(xsv)	규칙/무종성/음성 (xsv_rg_n_d)
			규칙/유종성/양성 (xsv_rg_y_b)
			여불규칙/무종성/양성 (xsv_yeo_n_b)
	형용사파생접미사 (xsm)	형용사파생접미사 (xsm)	여불규칙/무종성/양성 (xsm_e_n_b)
			ㅂ불규칙/유종성/양성 (xsm_b_y_b)
			ㅂ불규칙/유종성/음성 (xsm_b_y_d)
	부사 파생접미사 (xsa)	부사파생접미사(xsa)	무종성(xsa_n)
			유종성(xsa_y)
			ㄹ종성(xsa_l)

부록 L

형태소 해석(정보검색)을 위한 품사 분류 (좌,우접속정보)

체언

국어 형태-통사 태그 규격			좌접속정보		우접속정보
보통명사(nc)	서술성명사(ncp)	동작성명사(ncpa)	서술성보통명사(ncp)		무종성(ncp_n)
		상태성명사(ncps)			유종성(ncp_y)
					ㄹ종성(ncp_l)
	비서술성명사(ncn)	비서술성명사(ncn)	비서술성명사(ncn)		무종성(ncn_n)
					유종성(ncn_y)
					ㄹ종성(ncn_l)
고유명사(nq)	고유명사(nq)		고유명사(nq)	인명(nq_per)	무종성(nq_per_n)
					유종성(nq_per_y)
					ㄹ종성(nq_per_l)
				지명(nq_loc)	무종성(nq_loc_n)
					유종성(nq_loc_y)
					ㄹ종성(nq_loc_l)
				기타(nq_etc)	무종성(nq_etc_n)
					유종성(nq_etc_y)
					ㄹ종성(nq_etc_l)
의존명사(nb)	단위성 의존명사(nbu)		단위성 의존명사(nbu)		무종성(nbu_n)
					유종성(nbu_y)
					ㄹ종성(nbu_l)
	비단위성의존명사(nbn)		서술성비단위성의존명사(nbn_p)		무종성(nbn_p_n)
					유종성(nbn_p_y)
					ㄹ종성(nbn_p_l)
			비서술성단위성의존명사(nbn_n)		무종성(nbn_n_n)
					유종성(nbn_n_y)

			ㄹ종성(nbn_n_l)
대명사(np)	인칭대명사(npp)	대명사(np)	무종성(np_n)
	지시대명사(npd)		유종성(np_y)
			ㄹ종성(np_l)
수사(nn)	양수사(nnc)	양수사(nnc)	무종성(nnc_n)
			유종성(nnc_y)
			ㄹ종성(nnc_l)
	서수사(nno)	서수사(nno)	무종성(nno_n)
			유종성(nno_y)
			ㄹ종성(nno_l)

수식언

국어 형태·통사 태그 규격		좌접속정보	우접속정보
관형사(mm)	지시관형사(mmd)	지시관형사(mmd)	지시관형사(mmd)
	성상관형사(mma)	성성관형사(mma)	성성관형사(mma)
		수관형사(mmn)	수관형사(mmn)
부사(ma)	지시부사(mad)	부사(ma)	무종성(ma_n)
	일반부사(mas)		유종성(ma_y)
	접속부사(maj)		ㄹ종성(ma_l)

용언

국어 형태-통사 태그 규격		좌접속정보	우접속정보
동사(pv)	지시 동사(pvd)	동사(pv)	규칙(pv_rg)
	일반 동사(pvg)		ㅅ불규칙(pv_s)
			ㄷ불규칙(pv_d)
			ㅂ불규칙(pv_b)
			르불규칙(pv_reu)
			우불규칙(pv_woo)
			여불규칙(pv_yeo)
			러불규칙(pv_reo)
			거라(pv_geo)
			너라(pv_neo)
			으탈락(pv_eu)
			ㄹ탈락(pv_l)
형용사(pa)	지시형용사(pad)	형용사(pa)	규칙(pa_rg)
	성상형용사(paa)		ㅅ불규칙(pa_s)
			ㅂ불규칙(pa_b)
			르불규칙(pa_reu)
			여불규칙(pa_yeo)
			러불규칙(pa_reo)
			ㅎ불규칙(pa_h)
			으탈락(pa_eu)
			ㄹ탈락(pa_l)
보조용언 (px)	보조용언(px)		

독립언

국어 형태·통사 태그 규격		좌접속정보	우접속정보
독립어(i)	감탄사(i)	감탄사(i)	감탄사(ie)

관계언

국어 형태·통사 태그 규격		좌접속정보		우접속정보
격조사(jc)	주격조사(jcs)	일반조사(j)	무종성형(jc_n)	일반조사(j)
	목적격조사(jco)		유ㄹ종성형(jc_yl)	
	보격조사(jcc)		무ㄹ종성형(jc_nl)	
	관형격조사(jcm)		유종성형(jc_y)	
	호격조사(jcv)		공용형(jc_c)	
	부사격조사(jca)			
	인용격조사(jcr)			
	접속격조사(jcj)			
	공동격조사(jct)			
보조사(jx)	통용보조사(jxc)			
	종결보조사(jxf)			
서술격조사(jp)	서술격조사(jp)	서술격조사(jp)		서술격조사(jp)

어미

국어 형태·통사 태그 규격		좌접속정보	우접속정보
선어말어미(ep)	선어말어미(ep)	선어말어미(ep)	선어말어미(ep)
연결어미(ec)	대등적 연결어미(ecc)	어말어미(ef)	어말어미(ef)
	종속적 연결어미(ecs)		
	보조적 연결어미(ecx)		
전성어미(et)	명사형 어미(etn)		
	관형사형 어미(etm)		
종결어미(ef)	종결어미(ef)		

접사

국어 형태·통사 태그 규격		좌접속정보		우접속정보	
접두사(xp)	접두사(xp)	접두사(xp)		보통명사용(xp_nc)	
				고유명사용(xp_nq)	인명(xp_nq_per)
					지명(xp_nq_loc)
					기타(xp_nq_etc)
접미사(xs)	명사파생접미사(xsn)	보통명사용(xsn_nc)		무종성(xsn_n)	
		고유명사용(xsn_nq)	인명(xsn_nq_per)	유종성(xsn_y)	
				ㄹ종성(xsn_l)	
			지명(xsn_nq_loc)		
			기타(xsn_nq_etc)		

	동사파생접미사(xsv)	서술어파생접미사(xsp)	규칙(xsp_rg)
	형용사파생접미사(xsm)		여불규칙(xsp_yeo)
			ㅂ불규칙(xsp_b)
	부사 파생접미사(xsa)		

부록 M

● 국어 구문 태그 규격

구문 태그	설명
S	문장
NP	명사구절
VP	동사구절
ADJP	형용사구절
ADVP	부사구절
MODP	관형사구절
IP	독립구절
AUXP	보조용언구절

● 문장 성분과 구문 태그간의 관계

문장 성분	구문 태그
문장	S
주어	NP+jcs
서술어	VP
	ADJP
목적어	NP+jco
보어	NP+jcs
관형어	MODP
	NP+jcm
	VP+etm
	ADJP+etm
부사어	ADVP
	NP+jca
독립어	IP
기타	AUXP

11.한국어 음성 DB 구축에 관한 연구

원광대학교
이용주

여 백

11. 한국어 음성 DB 구축에 관한 연구

1장. 서 론

음성인식 및 합성 등 우리말 음성정보처리시스템의 개발을 위해서 가장 먼저 확보해야 할 것이 다양한 사람이 발성한 대량의 음성데이타베이스이다. 이 음성데이타베이스는 인식연구에서는 알고리즘의 훈련 및 평가용, 합성에서는 합성단위 제작을 위한 기본자료이며 음운 및 운율규칙의 생성을 위한 기본적인 분석자료의 대상이 된다.

따라서 음성 관련 연구자들에게는 모두 이러한 음성데이타베이스가 연구초기부터 필수적으로 필요하나 이의 확보에는 많은 시간과 예산, 그리고 노력이 필요하며 또한 전문적 기술이 필요하다. 따라서 국가적인 차원에서 체계적으로 구축하여 공동으로 활용케하는 것이 바람직스러우며 각국에서도 국가나 공공연구기관이 주축이 되어 활발하게 구축, 활용하고 있다. 그러나 우리 나라는 아직 거기까지는 이르지 못하고 있으며 대부분의 기관에서 나름대로 자체 제작하여 내부적으로만 사용하고 있어 데이타량 및 이용형태가 제한되고 또한 각 연구자가 발표한 인식시스템의 성능 및 분석방식의 평가를 각 연구자의 데이터에 의존하고 있어서 객관적으로 이루어지지 않고 있는 실정이어서 음성데이타베이스의 체계적인 구축이 시급하다.

음성 DB 가 체계적으로 구축되면 먼저 이를 토대로 한 한국어 음성의 체계적이고 정량적인 분석이 가능하고, 이에 따라 축적된 한국어 음성의 분석자료는 인식 및 합성과 같은 공학적 응용의 기반이 된다. 즉, 인식시스템 개발에 있어서는 음성 DB 가 직접적으로 시스템의 훈련 및 평가용으로 활용될 것이며, 합성시스템의 개발에서는 규칙의 고도화에 따른 합성품질의 개선 및 평가에 필수 요소가 될 것이다. 이러한 음성 DB 를 국가적 차원에서 체계적으로 구축하여 보급하는 일은 관련산업계의 기술 개발 촉진뿐만 아니라 한국어 음성연구의 학술적 발전에도 기여하는 바가 대단히 크다. 즉 모든 음성 관련 연구 및 기술 개발은 체계화된 음성자료인 음성 DB 의 확보에서 출발하므로 본 연구의 필요성은 크다.

본 연구는 이러한 음성 DB 를 체계적이고 지속적으로 구축하고, 이를 효율적으로 공동 이용할 수 있도록 유지, 관리, 보급하는 체계를 갖추는 것을 목표로 하는 3 개년의 연구의 3 차년도이다.

본 3 차년도에는 기 구축된 어절레벨의 PBW 을 대상으로 음소단위로 레이블링된 DB 를 구축하였고 다양한 음소환경이 포함된 문장세트인 PBS(Phonetically Balanced Sentence)를 대상으로 이의 음성 DB 를 구축하였다. 이로서 단어 및 문장 단위의 음성에 대한 기본적인 음성 DB 가 확보되어 음성 연구에 널리 이용될 수 있게 되었다.

2 장. 연구개발의 목표 및 내용

1 절. 연구개발의 최종목표

한국어 음성정보처리연구에 활용할 수 있는 음성 DB 를 체계적이고 지속적으로 구축하여, 국가적 차원에서 이를 효율적으로 공동 이용할 수 있도록 유지, 관리, 보급하는 체계를 확보한다.

2 절. 연차별 연구개발 목표 및 내용

본 연구의 연차별 연구개발 목표 및 내용은 다음 표 1.1 과 같다.

3 장. 음성 DB 의 음소 레이블링

본 장에서는 2 차년도 최종연구보고서에서 제안한 음성 DB 의 음소 레이블링 기준 중에서 3 차년도 연구를 수행하면서 변경된 부분을 기술한다. 그리고 3 차년도 연구에서 수행한 10 명분 화자의 PBW 452 어절을 음소단위 레이블링한 결과를 기술한다.

구 분	연 구 개 발 목 표	연 구 개 발 내 용 및 범 위
1 차년도 (1994)	-음성DB구축환경정비 -공통음성DB의 연차별 확보 계획 수립 -낭독음성DB 시험판 구축 단어레벨 단독,연결숫자:4명분*4회 PBW: 4명분*2회 문장레벨 단문: 4명분*2회	- 음성DB제작환경 구축 - 장기 확보 계획서 작성 - 시제품 개발
2차 년도 (1995)	-낭독음성DB 보급판 구축 단어레벨 단독,연결숫자: 70명분 PBW: 70명분 고빈도어: 2명분 문장레벨 문장음성DB 발성목록설계 - 레이블링 기준 작성 PBW 1명분 레이블링	- 시제품 개발(단어 음성 CD-ROM) - 발성 대상목록 작성 - 기준안 작성
3차 년도 (1996)	단어레벨 레이블링: PBW 10명분 문장레벨 PBS -설계목록 보완 -PBS 음성 DB : 50명분 Dictation용 설계항목 검토	- 시제품(레이블링 DB) - 설계목록 - 시제품 개발(음성DB CD-ROM) - 설계항목 검토서

표 1.1 연차별 연구개발 목표 및 내용

1 절. 레이블링 기준의 보완

1. 음성층과 음소층의 표기법의 재정립

2 차년도 보고서에서는 음성층에는 음소의 변이특성, 즉 파열음이나 파찰음의 폐쇄구간과 파열구간, 유성음화, 비파열, 유음의 변이음만을 표기하였다. 그러나 음성층의 트랜스크립션 결과가 인식이나 합성을 위한 훈련 데이터로 단독으로 사용될 수 있으므로, 이런 변이형뿐만 아니라 그외 음소도 음소층과 동일한 방식으로 음소 표기하였다.

그림 3.1 은 2 차년도 보고서에 기술한 음성층과 음소층의 표기법이고 그림 3.2 는 수정된 음성층과 음소층의 예이다.

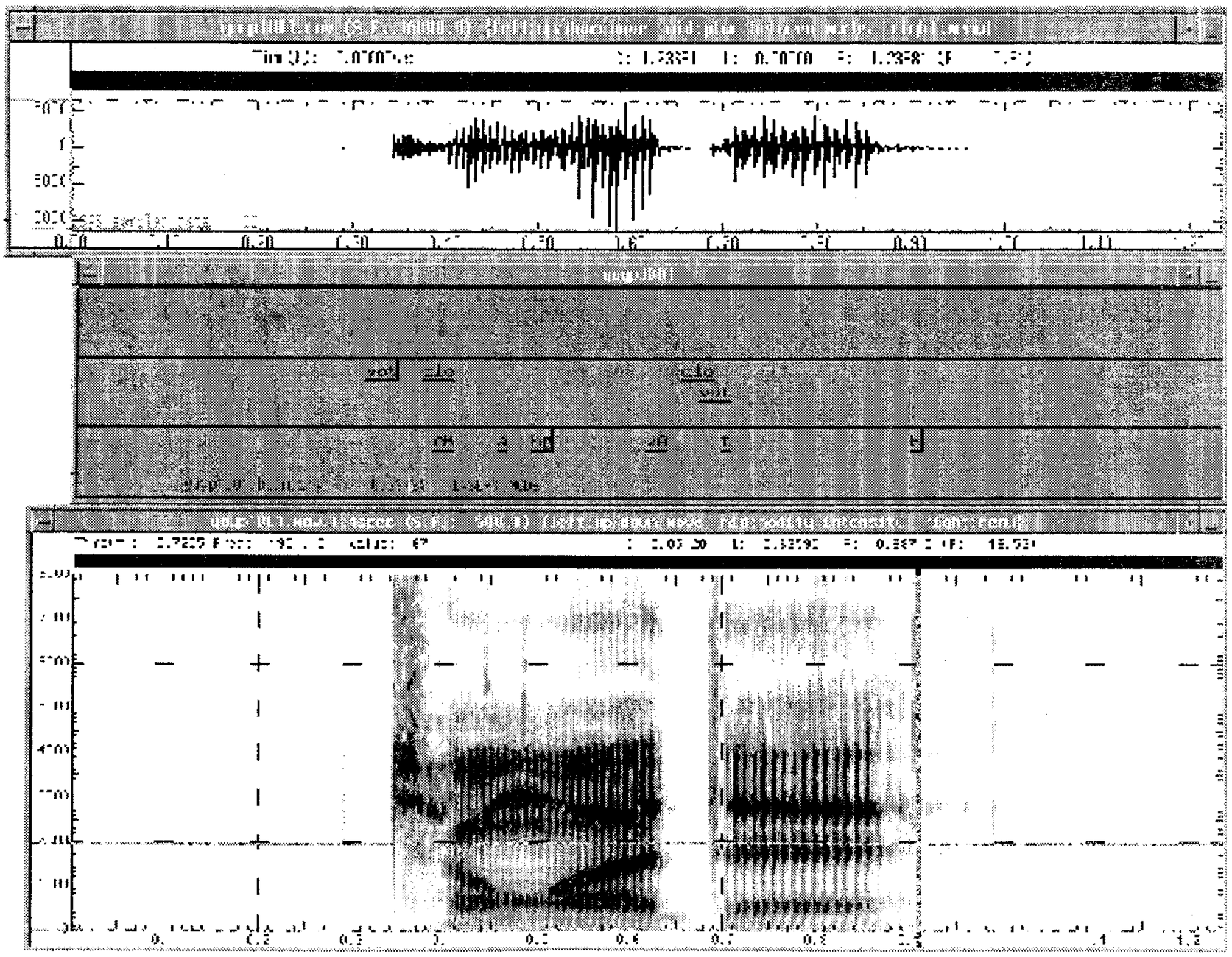


그림 3.1 변경전 음성층/음소층 표기법

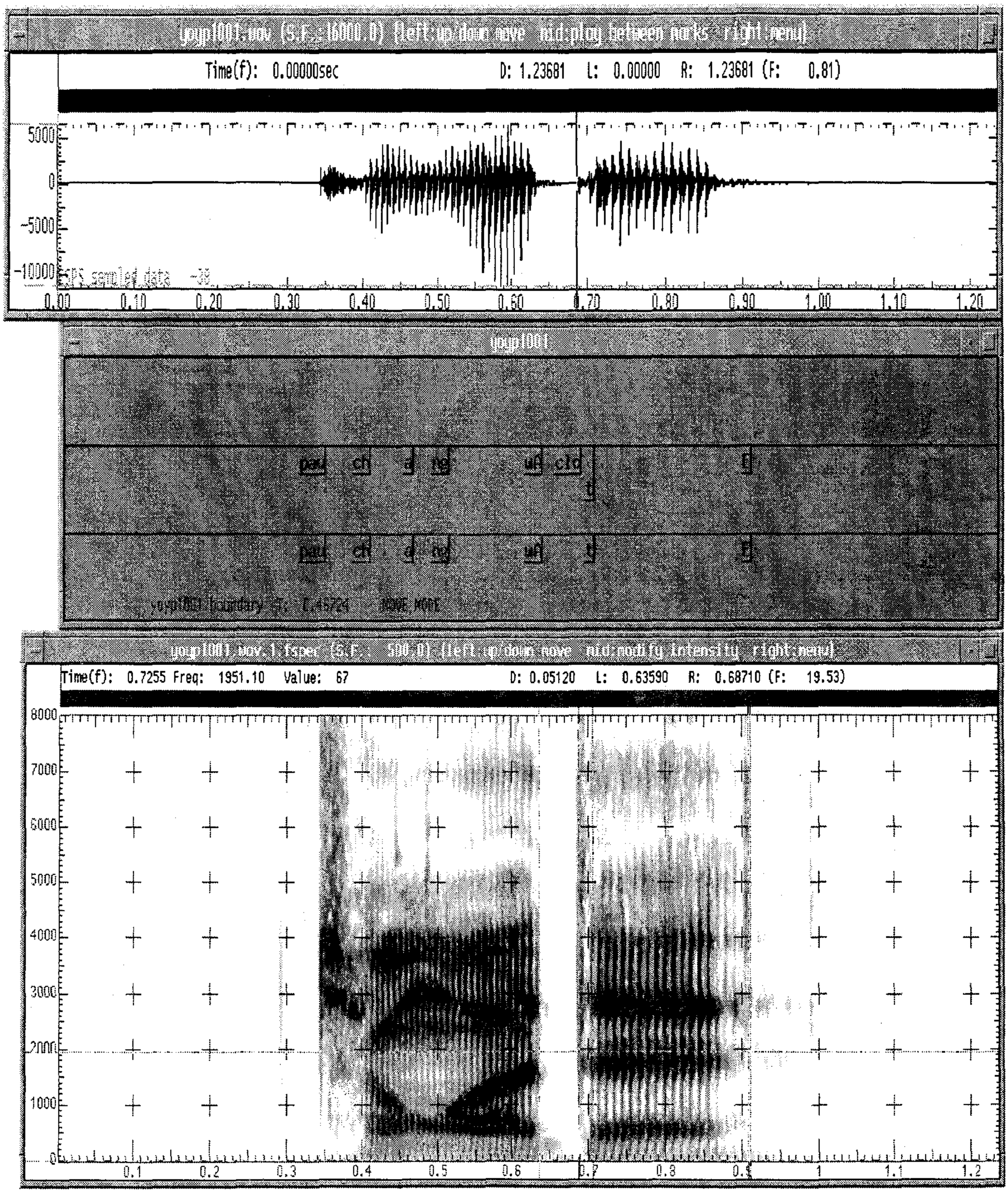


그림 3.2 변경된 음성층/음소층 표기법

2. 탄설음으로 나타나는 유음의 예

유음이 음절 초에서 탄설음으로 실현되는 경우 파열음과 유사한 특징을 보여 멈춤과 폐쇄구간을 보이는 경우, 폐쇄구간과 파열구간으로 분할하여 폐쇄구간은 'cl'로 파열구간은 'r' 로 표기하였다. 이에 대한 예는 그림 3.3 과 같다.

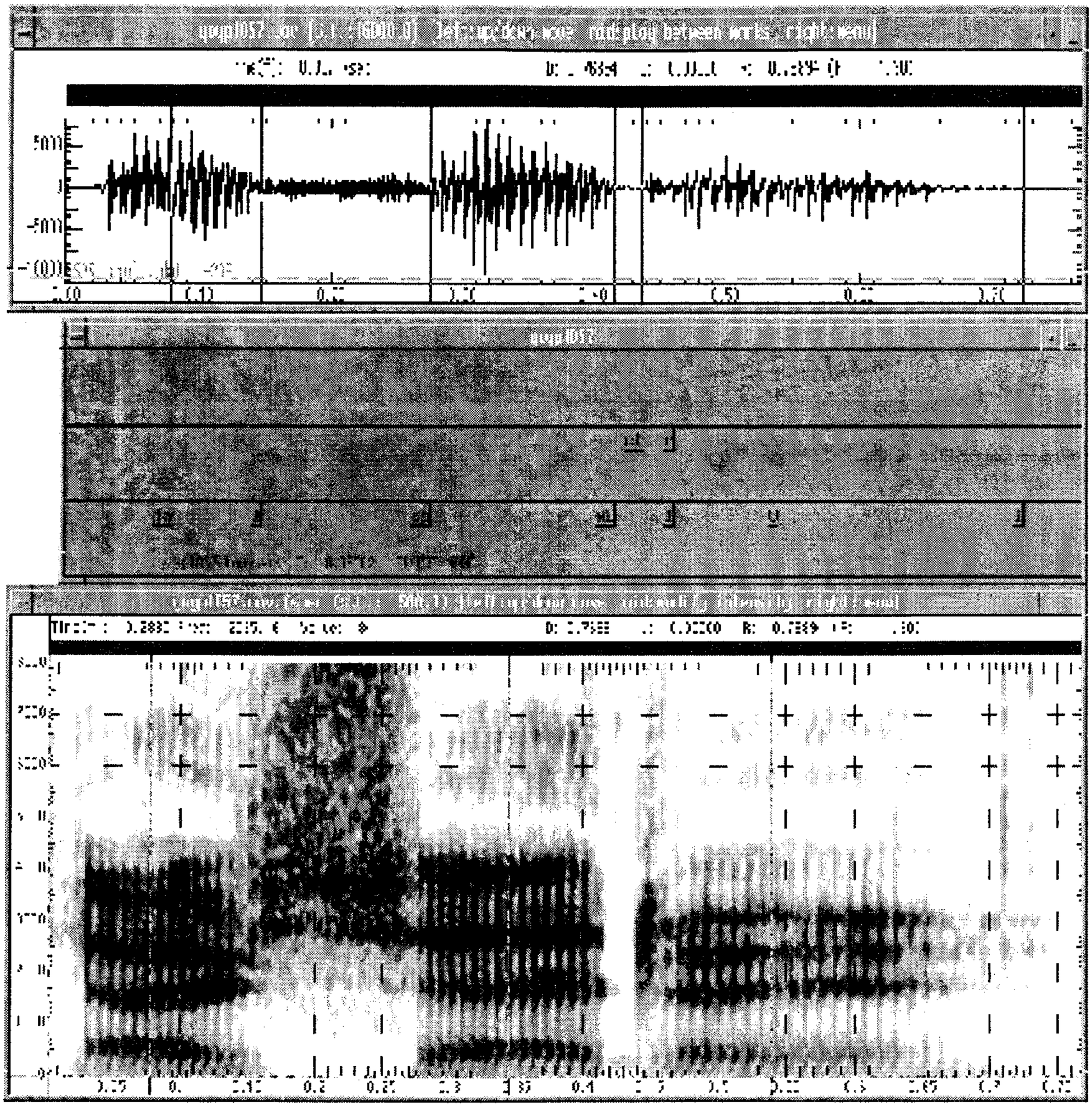


그림 3.3 탄설음의 레이블링

3. 모음 + 모음의 천이구간의 별도 표기

인간의 조음특성상 음소의 연속체에 나타나는 음소간 경계를 분할하기가 매우 어렵다. 이로 인해 음소단위 모델을 이용한 인식시스템이나 자동 레이블링 시스템에서 음소간 경계를 정확히 결정짓기 어렵다. 또한 음성합성에서는 음소연속체를 생성할 때 그 경계 부분에서의 포만트 주파수 불일치에 인해 부자연스러운 기계음으로 들리게 한다. 이러한 문제점을 해결하기 위해 여러 가지 대안(bigram, trigram)들이 제안되고 있으나 근본적인 문제점에는 접근하지 못하고 있는 실정이다. 이러한 문제점들에 해결하기 위해서는 음소연속체구간에서의 천이구간을 대상으로 조음환경에서의 천이모델(rising/falling model)에 대한 연구가 진행되고 있으며 이는 후속 연구로 계속 되어야 한다. 이를 위해서는 먼저 천이구간을 별도의 방법으로 트랜스크립션한 음성 데이터베이스가 절실히 필요하다. 그러므로 다음 그림과 같은 트랜스크립션 규칙을 검토하였으며 후속 연구에서 이를 더욱 확장하여야 한다.

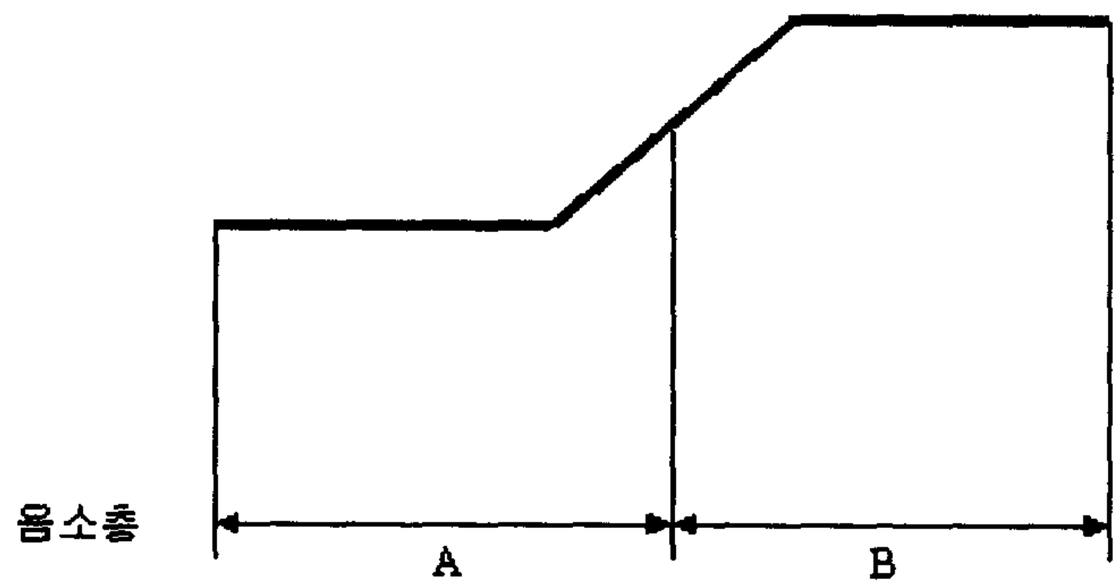


그림 3.4 현재 트랜스크립션 기준

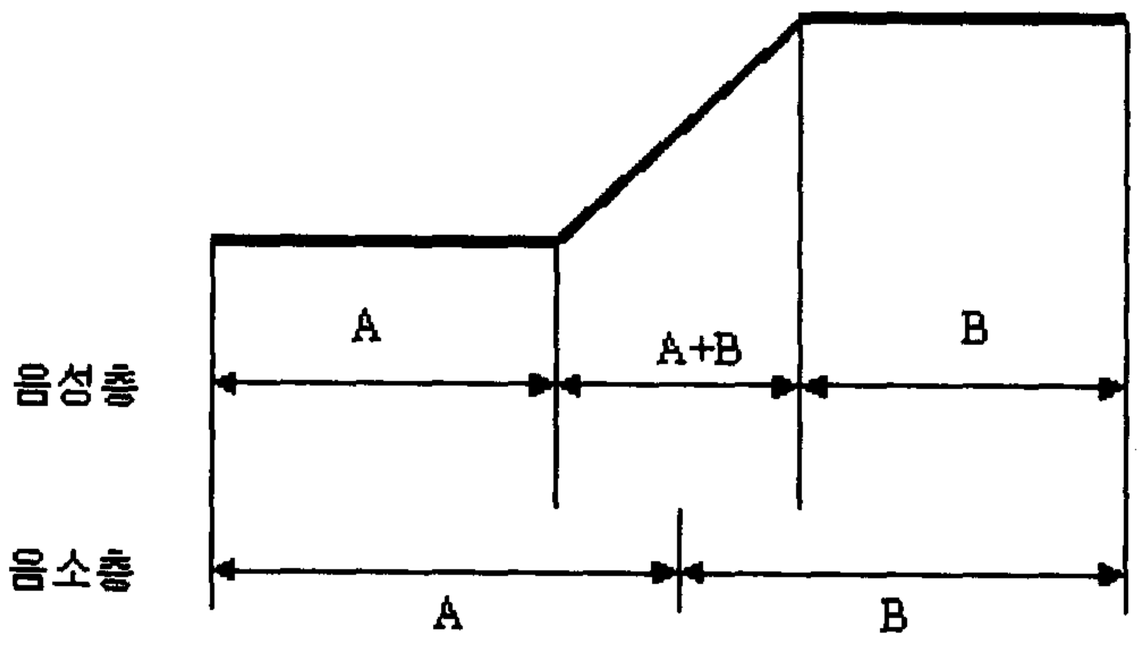


그림 3.5 검토중인 트랜스크립션 기준

2 절. 음소 레이블링 DB

1. 작업 절차

본 3 차년도 연구에서는 2 차년도 연구에서 작성된 반자동 레이블링 시스템을 사용하였다. 작성된 반자동 레이블링 시스템의 인식 모델 최적화나 성능평가는 아직 이루어지지 않았으나 수작업 량을 상당부분 감소시킬 수 있었다. PBW 452 어절 10 명분의 레이블링 작업은 진행은 다음과 같은 작업 절차에 따라 진행되었다.

가. 반자동 레이블링 시스템을 이용하여 10 명분의 데이터를 레이블링.

나. 훈련된 레이블러에 의한 1 차 에러 수정

다. 음성전문가의 2 차 에러 수정

2. 레이블링 결과

남성 5, 여성 5명이 1 회씩 발성한 총 4520 어절의 PBW에 대해서 반자동 레이블링시스템으로 레이블링하고 남성 1, 여성 3 명으로 구성된 전문 레이블러들의 수정 작업을 통하여 음소 레이블링이 완료되었다.

3 절. 향후 계획

음소 레이블링은 실제 데이터를 대상으로 레이블링 해 가면, 기존의 기준으로는 판단하기 어려운 많은 예들이 발생하고 이에 따라 기준의 수정이 이루어져야 한다. 본 연구에서 10명분의 PBW를 대상으로 레이블링이 이루어 졌으나 지속적으로 대량의 음성 데이터를 레이블링 하기 위해서는 기준이 보완되어야 한다.

향후 연구에서는 현재까지 작성된 10 명분의 레이블링 결과를 토대로 음성 연구자들이 공통으로 쓸 수 있도록 한국어 음성의 음소단위 레이블링 기준의 보완 및 표준화와 전문가 그룹에 의한 검토가 이루어져야 한다. 이를 위해서는 국내의 음성학자 및 관련 연구가들을 대상으로 한 컨소시엄 형태의 의견수렴의 장이 마련되어져야 하며, 장기적으로는 보다 체계적인 활동을 위해 소위원회 형태의 상

시 전문가 그룹이 결성되어야 할 필요가 있다.

4장. 문장 음성 DB의 구축

본 장에서는 문장 음성 DB의 발성목록으로 사용된 PBS를 균형 코퍼스(Balanced Text Corpus)에서 추출하는 절차와 추출된 PBS를 사용하여 구축된 문장 음성 DB에 대하여 기술한다.

1절. PBS(Phonetically Balanced Sentence)의 설계

1. PBS의 정의

음성 인식 및 합성 시스템의 개발 등 음성 연구에 있어서 다종 다양한 음운 현상을 포함한 음성 데이터 베이스의 구축은 중요한 과제의 하나로서 많은 시간과 노력이 요구된다. 개별적 음성 데이터 베이스의 구축에 따른 중복투자를 줄이고 분석, 합성, 인식의 각종 알고리즘을 적절히 비교 평가하기 위해서도 공통의 음성 데이터 베이스는 필수적이다.

이러한 목적으로 사용 될 문장 음성 데이터 베이스는 적은 문장으로 실제 한국어의 발성에 나타나는 음운현상을 가능한 많이 포함하며, 특정 태스크에 집중되지 않는 것이 바람직하다.

음운 밸런스가 취해진 상태란 음운의 출현빈도가 같은 상태를 말한다. 이러한 상태는 음운을 확률사상으로 했을 때 엔트로피가 최대인 상태이다. 문장세트에 나타난 각 음운 환경의 출현확률을 p_n , $(1 \le n \le N)$라고 할 때, 문장 세트의 음운 엔트로피 S는 다음 식 (1)에 의해 구할 수 있다.

$$S = \sum_{n=1}^{N} - p_n \log_2 p_n \qquad (1)$$

이때 엔트로피 S는 음운환경의 출현빈도가 모두 동일할 때 최대치 $\log_2 N$이 된다. 그러나 PBS를 추출할 때, 실생활에서 사용되고 있는 문장들을 모집단으로 하여 추출하기 때문에, 사용되지 않는 무의미한 문장들을 임의로 만들어 내지 않

는 한 PBS를 구성하고 있는 각 음운 환경이 완벽히 고른 확률 분포를 가질 수는 없다. 다만 실제 사용되는 유의미한 문장들을 엔트로피가 최대가 되도록 구성함으로써, PBS에 포함된 음운 환경들간의 균형을 최대한 유지할 수 있다.

또한 음성DB를 위한 발성 목록은 그 양이 많아지게 되면 발성자의 실제 발성 시료를 수집하는 데 많은 어려움이 따르게 된다. 따라서 PBS가 발성 목록으로 사용되기 위해서는 엔트로피를 최대화하면서도, 최소 문장들의 집합으로 구성되어야 한다.

따라서, PBS는 모집단에 나타난 모든 음운 현상을 포함하며, 각 음운 환경들이 고른 확률 분포를 갖는 최소 문장들의 집합이라고 할 수 있다.

2. PBS 추출을 위한 모집단의 선정

(1) PBS 추출을 위해 사용한 텍스트 코퍼스

PBS를 추출하는 목적이 특정 태스크에 집중되지 않으면서 한국어의 발성에 나타나는 음운현상을 최대한 포함한 발성목록을 구성하는 것이기 때문에, PBS 추출의 모집단으로 어떠한 코퍼스를 사용할 것인가 하는 점도 PBS를 추출하기 위한 중요한 요인중의 하나이다.

PBS 추출의 모집단이 특정 태스크에 편중되어 있으면 그로부터 추출된 PBS 역시 특정 태스크에 편중되는 결과가 발생하므로 모집단이 실제 한국인의 언어생활을 충분히 반영할 수 있도록 각 분야별로 균형있게 구성된 대량의 텍스트 코퍼스에서 PBS를 추출하는 것이 바람직하다.

본 연구에서는 시스템 공학 연구소(SERI)의 국어공학센터(KLE)에서 각 장르별로 균형 있게 구성하여 구축한 100만 어절의 한국어 텍스트 코퍼스를 PBS 추출의 모집단으로 사용하였다. 텍스트 코퍼스를 구성하고 있는 각 분야별 분포는 다음 표 4.1과 같다.

(2) 모집단 10,000 문장의 선정

텍스트 코퍼스에는 신문기사의 제목, 특수기호가 포함된 문장 등 발성목록으로는

적절치 않은 문장이 존재한다. 이러한 문장을 제외하고 PBS 추출의 모집단으로 사용할 10,000 문장을 무작위로 선정한다. 10,000 문장을 선정하는 기준은 다음과 같다.

범 주	장 르	출전 종류 수	어절 수
소 설	일반 소설	30종	162,000
	꽁트, 우화	3종	16,000
	어린이	4종	19,000
	역사 소설	8종	43,000
	공상 과학	2종	10,000
학술 산문	학술 산문	40종	208,000
예술 산문	수 필	21종	114,000
	전 기	7종	37,000
비학술, 비예술 산문	실용, 지침서	16종	75,000
	해설서	19종	96,000
	보고, 실록	11종	46,000
	시 론	11종	47,000
	기 타	3종	16,000
사적 저술	일 기	2종	16,000
	회고록	7종	38,000
	기행문	3종	16,000
	편지	3종	24,000

표 4.1 텍스트 코퍼스의 분야별 구성

(1) 문장의 내용

- 외국어가 직접 사용된 문장은 제외한다.

- 비어, 속어가 포함된 문장은 제외한다.

(2) 문장의 길이

적은 수의 문장으로 한국어의 음운 현상을 최대한 많이 포함시키기 위해서는 문장의 길이가 긴 것이 바람직하겠지만, 문장이 너무

길면 발성자가 발성하기에 용이하지 않다. 따라서 문장의 길이를 10어절에서 20어절의 사이로 제한한다.

(3) 기사의 제목, 학술 논문의 장, 절 제목 등 종결되지 않은 문장은 제외한다.

(4) 한자는 일괄적으로 한글로 변환한 후에 처리한다.

(5) 인용부호의 처리

인용 부호(‘’, “”, 「」, 『』)속의 내용은 인용부호만을 삭제하고 취급한다. 예를 들면 “진짜로 사촌이 땅 산 것만큼이나 배아파하는 ‘사촌’들을 많이 볼 수 있는 것이다.”와 같은 문장은 인용 부호(‘’)만을 삭제하고 올바른 문장으로 취급하였다.

(6) 괄호의 처리

괄호 속에 앞에서 서술한 용어 등에 대한 부가적 서술이 들어가 있는 경우에는 괄호 속의 내용을 무시하고 문장으로 취급한다. 예를 들어 “*평활근(smooth muscle)은 장기(intestines, 소화중 음식물을 운반한다), 한선(sweat glands; 피부 밖으로 액체를 밀어내어 체온을 낮춘다) 및 부신(adrenal glands, 최대 운동시 혈류 속으로 아드레날린을 주입한다)과 같은 혈관과 내장기관에 있다.*”와 같은 문장은 괄호와 괄호 속의 내용을 무시하고 나머지만을 문장으로 취급하다.

(7) 기타 특수기호가 포함된 문장은 제외한다.

이상과 같은 과정을 거쳐 선정된 모집단 10,000 문장의 어절 수별 분포는 다음 그림 4.1 과 같다.

(3) 읽기 규칙의 적용

한국어의 음운 규칙에는 읽기 규칙(음소 변동 규칙)과 변이음 규칙이 있다. 모집단 10,000 문장에 읽기 규칙을 적용하여 소리나는 형태로 자동 변환한다.

음운 변환기는 한글을 소리나는 대로 바꿔주는 읽기 규칙을 구현한 것이다. 읽기 규칙은 교육부에서 고시한 표준어 규정의 표준 발음법을 따르고 있다. 읽기 규칙의 적용에 대한 자세한 사항은 1 차년도 최종 보고서를 참조 할 것.

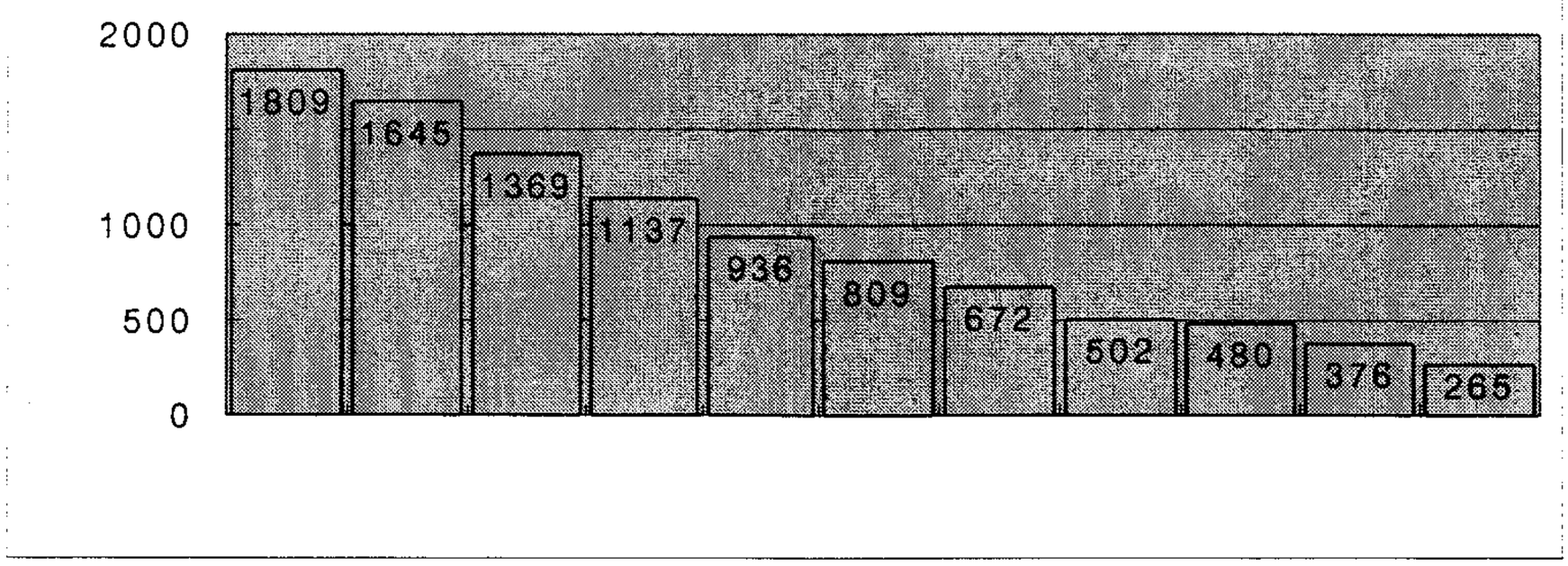


그림4.1 모집단 10,000문장의 어절 수별 분포

읽기 규칙을 적용했을 경우 변환된 문장의 예를 보이면 다음과 같다.

변환 전 :

절룩거리는 걸음걸이여서 발을 떼어놓을 때마다 발뒤꿈치를 따라 흙탕물 방울이 유난스럽게 튀어올랐다

변환 후:

절룩꺼리는 거름거리여서 바를 떼어노을 때마다 발뒤꿈치를 따라 흑탕물 방우리 유난스럽께 튀어올랃따

3. PBS 추출 절차

(1) 음운 환경의 정의

음운 규칙 중 읽기 규칙(음소 변동 규칙)은 읽기 규칙 변환을 통하여 적용되었으나, 변이음 특성은 모집단에 반영되지 않았다. 변이음 특성은 전후 음소의 영향을 받아서 발생하기 때문에 이를 반영하기 위해서는 변이음이 발생하는 음운 환경(음소열)을 정의하고 이를 단위로 하여 PBS를 추출하여야 한다.

음운 환경은 음소열의 길이를 길게 할 수록 여러 음운 현상을 충실히 반영할 수 있다. 그러나 음소열의 증가에 따라 다루어야 하는 단위의 수가 기하급수적으로 증가하므로 처리하기가 곤란하게 되며, 단위의 수가 증가함에 따라 추출되

는 PBS의 문장수도 증가하게 됨으로써 발성목록으로 사용하기에 곤란하다는 단점이 있다.

따라서 본 연구에서는 한국어의 발성에 나타날 수 있는 모든 2음소열을 PBS 추출의 기본 단위로 하고, 특히 3음소열로 고려해야 할 유성음화, 탄설음화의 경우를 유형별로 클러스터링 하여 음운 환경으로 정의하였다.

1) 2음소열

본 연구에서 사용하는 대상 음소로는 자음 19개, 모음 21개로 구성하고 종성의 자음에 /ㅅ/, /ㅊ/, /ㅌ/ 등은 모두 대표 자음으로 모아 다루었으며, 반모음은 후속 모음과 합하여 하나의 음소로 취급하였다. 예를 들면 /ㅑ/, /ㅕ/, /ㅘ/, /ㅟ/등이 그러한 경우이다.

대상 2음소열은 위와 같이 분류된 음소의 쌍으로 구성하였으며, 문장을 발성할 경우 같은 음소라도 문중, 문두, 문미의 위치에서 음향적 특성이 서로 다를 가능성이 있기 때문에 문두, 문미의 공백소도 하나의 음소로 취급하여 공백소와 해당 음소가 쌍을 이루는 형태를 2음소열의 종류에 포함하였다. (즉, /B ㄱ/은 /ㄱ/이 문두에 오는 경우, /ㄱ B/는 /ㄱ/이 문미에 오는 경우를 말한다. 다음 표4.2에 대상 음소를 나타내었다.

자 음	ㄱ,ㄲ,ㄴ,ㄷ,ㄸ,ㄹ,ㅁ,ㅂ,ㅃ,ㅅ,ㅆ,ㅇ,ㅈ,ㅉ,ㅊ,ㅋ,ㅌ,ㅍ,ㅎ
모 음	ㅏ,ㅐ,ㅑ,ㅒ,ㅓ,ㅔ,ㅕ,ㅖ,ㅗ,ㅘ,ㅙ,ㅚ,ㅛ,ㅜ,ㅝ,ㅞ,ㅟ,ㅠ,ㅡ,ㅢ,ㅣ
공백소	B

표 4.2 대상 음소

이와 같이 정의된 2음소열에서 반영할 수 있는 변이음 특성은 다음과 같다.

(가) 고모음의 무성화

(나) 목젖 소리 되기

(다) 경구개음화

(라) 원순음화

(마) 연구개음화

(바) 비음화

(사) 미파화

(아) 기타 2 음소열간 발생할 수 있는 변이음 특성

2) 3 음소열

PBS 추출의 기본 단위로는 위에 기술한 2 음소열을 그 대상으로 하였으나, 실제 발화시 자신의 앞과 뒤에 있는 음소들의 영향을 받아 음성 현상이 변화하는 경우도 있기 때문에 3 음소열 역시 단위로 포함되어야 한다. 그러나 CVC, VCV, CCV 등 3 음소열 전부를 단위로 하기에는 그 종류수가 방대해져서 처리하기가 곤란하게 된다.

따라서 본 연구에서는 한국어의 발화에서 특히 3 음소열을 고려해야 할 대상으로서 다음과 같은 두개의 3 음소열 그룹을 PBS 추출의 단위로 추가하여 사용한다.

(가) 유성음화 및 자음 약화

자음 /ㅂ, ㄷ, ㄱ, ㅈ, ㅎ/는 유성음(모음, 비음 /ㅁ, ㄴ, ㅇ/, 유음 /ㄹ/) 사이에서 유성음화 되거나 약화될 수 있다. 따라서 이를 3 음소열로 설정하여야 하나 이들 모두를 단위로 하기에는 그 양이 너무 많아진다(유성음 25 종 * 자음 5 종 * 유성음 25 종 = 3125 종). 따라서 유성음을 고모음(/ㅣ, ㅡ, ㅜ/, /ㅚ, ㅟ, ㅠ/)과 저모음(나머지 모음), 유음, 비음의 4 종류로 클러스터링 하여 처리함으로써 단위의 수를 80 종(유성음 4 종 * 자음 5 종 * 유성음 4 종 = 80 종)으로 줄였다.

(나) 유음의 탄설음화

유음 /ㄹ/은 모음과 모음 사이에서 탄설음화 된다. 이러한 경우 역시 모음을 전체 모음으로 클러스터링 하여 1 종으로 취급하였다. 또한 종성 유음 /ㄹ/은 뒤에 초성 /ㅎ/가 탈락 되면서 다음 음절의 초성으로 발성되어 탄설음화될 가능성이 있다. 이러한 경우를 반영하기 위해 /ㄹ/뒤에 /ㅎ/이 왔을 경우도 탄설음화의 경우에 포함하였다.

(2) PBS 추출 알고리즘

모집단에서 발생한 음운환경 n 의 수를 N_{pn} , PBS 후보 세트에서의 수를 N_n , 모집단에서 발생한 음운환경의 종류 수를 T_p , PBS 후보 세트에서의 수를 T 라고 한다면 PBS 추출 알고리즘은 다음과 같다.

1) 초기 PBS 후보 세트의 구성

단계 1. 모집단의 모든 문장들을 조사하여 N_{pn}=1 인 음운환경을 갖는 문장들을 모두 PBS 후보 세트에 포함시키면서 모집단에서 제거한다.

2) 추가 과정

단계 2. 모집단의 나머지 문장들을 모두 조사하여, 각 문장에 포함된 음운 환경 중 N_n=0 인 음운 환경을 가장 많이 포함하고 있는 문장을 임시 선택한다.

단계 3. 선택된 문장이 복수일 경우, 한 문장 씩 PBS 후보 세트에 추가하여 PBS 후보 세트의 엔트로피를 계산하였을 때, PBS 후보 세트의 음운 엔트로피가 최대화되는 문장을 PBS 후보 세트에 포함시키면서 모집단에서 삭제한다. 이때, PBS 후보 세트의 엔트로피는 식(1)에서와 같이 구할 수 있으나, 식(1)을 그대로 적용하였을 경우에는 문장이 추가되거나 삭제될 때 각 음운 환경의 상대 출현 빈도를 전부 재계산 하여 엔트로피를 구해야 하므로 음운 환경의 종류가 증가함에 따라 계산량이 급격히 증가하게 된다. 따라서 다음과 같이 수식을 변경하여 계산량을 감소시켜 사용한다.

PBS 후보 세트의 음운 환경의 총 출현빈도, 즉 $\sum N_n$ 을 N_T , $S_n=-N_n\log_2 N_n$ 이라고 한다면 식(1)은 다음과 같이 쓸 수 있다.

$$S = \log_2 N_T + \frac{1}{N_T}\sum_{n=1}^{N} S_n \qquad (2)$$

식 (2)를 이용하면 PBS 추출 시 문장이 추가되거나 삭제될 경우 각 음운

환경에 대해 상대 출현 확률을 모두 계산할 필요가 없고, 단지 추가되거나 삭제되는 문장에 포함된 음운 환경에 대해서만 S_n 을 재계산 하면 엔트로피를 구할 수 있으므로 식(1)을 직접 사용한 것 보다 계산량을 현저하게 줄일 수 있다.

단계 4. 단계 2. ~ 단계 3.의 과정을 T 가 T_p 와 같아질 때까지 반복한다. 단계 1. ~ 단계 4.를 통하여 모집단에서 발생한 모든 음운 현상이 포함되게 된다.

단계 5. PBS 후보 세트에 포함되지 않은 모집단의 나머지 문장들을 한번에 한 문장씩 PBS 후보 세트에 추가하여 PBS 후보 세트의 엔트로피를 계산해 보고, 추가했을 경우 PBS 후보 세트의 엔트로피를 최대화 할 수 있는 문장을 PBS 후보 세트에 추가하고 이 문장을 모집단에서 삭제한다.

단계 6. 단계 5.의 과정을 PBS 후보 세트의 엔트로피를 증가시키는 문장이 더 이상 없을 때까지 반복한다.

3) 삭제 과정

단계 7. PBS 후보 세트를 구성하고 있는 문장들 중, $N_n \geq 2$ 인 음운 환경으로만 구성된(즉, 2 회 이상 출현한 음운 환경으로만 이루어진) 문장을 모두 임시 선택한다.

단계 8. 선택한 문장들을 한번에 한 문장씩 PBS 후보 세트에서 삭제해서 PBS 후보 세트의 엔트로피를 계산해 보고, 삭제시 PBS 후보 세트의 엔트로피를 최대화시키는 문장을 PBS 후보 세트에서 삭제한다.

단계 9. 단계 7. ~ 단계 8.의 과정을 $N_n \geq 2$ 인 음운 환경으로만 구성된 문장이 더 이상 존재하지 않거나, 존재하더라도 삭제하였을 경우 PBS 후보 세트의 엔트로피를 증가시키는 문장이 더 이상 없을 때까지 반복한다. 단계 5. ~ 단계 9.의 과정을 통하여 최소한의 문장들의 집합에 모든 음운 환경의 출현 확률을 최대한 고르게 구성할 수 있다.

4. 추출된 PBS

앞에서 기술한 알고리즘에 의거하여 본 연구에서 PBS를 추출하는 과정에서 초기 후보 세트로 81문장이 추출되었으며, 단계 2. ~ 단계 4.의 과정을 통하여 271문장, 단계 5 ~ 단계 6.의 과정을 통해서는 390문장이 추출되었다. 이렇게 해서 추출된 총 661문장 중 단계 7. ~ 단계 9.의 삭제과정을 통하여 총 59문장이 삭제되어 602문장의 PBS를 추출하였다. 추출된 602문장 중 다음과 같이 발성 목록으로 쓰이기에 부적절하다고 판단되는 문장들은 삭제하였다.

(1) 얼른 보아 의미를 알 수 없는 단어가 쓰였거나 의미가 모호하다고 판단되는 문장

예) "정양 실컨 하고선 애들 며칠 가르치다 밤새 몰래 도망치지는 않겠지"

"도행병은 복숭아와 살구를 물을 조금 넣고 믹서에 간 뒤 쌀가루 빻은 것에 섞어서 시루나 찜통에 얹어 쪄 낸다"

"섬유공업 비누는 원료 방적 섬유와 그 정련에서 그후 처리 공정에 널리 이용됩니다"

(2) 장, 절 등의 제목이 삭제되지 않고 문장으로 편입된 것

예) "견갑대 관절 운동 견갑골 관절이 움직일 때마다 겹갑골과 쇄골이 움직인다"

"둘째마당 재료 찾아내기 첫째갈래 뜻밖의 경우 비눗방울의 재료를 찾아 떠나오긴 했는데 어디서 어떻게 찾아야 할지 막막하군요"

"빛 해양에서의 먹이상은 녹색식물에 의해 광합성이 이루어지는 동안 흡수되고 후에 먹이의 형태로 다른 생물들에 의해 섭취되는 에너지와 물질에 의존한다"

"거짓말쟁이 먼 나라들을 여행하고 갓 귀국한 어떤 양반이 친구와 들길을 걸으면서 자기가 방문한 나라들에 대해 자랑을 섞어가며 얘기했다"

(3) 명백히 맞춤법이 틀렸거나 코퍼스 입력자의 실수라고 판단되는 문장

예) "언덛 위에 올라서 보니 큰 범 두 마리가 바위 아래서 싸우고 있었다"

"다구나 피아프는 극도의 영양부족으로 열 일곱 살이 될 때까지 시력이 온전치 못했다"

"단순한 플래쉬 카드 교습중의 하나의 악보 이름대기 프로그램은 음악 보표에 음표를 보여주고 학생이 이룸을 입력하도록 한다"

"어머님이 교육 방법은 요새 젊은 엄마들의 눈에는 매우 위험스럽게 보일지도 모릅니다"

"그 머리 위로 때묻은 갈매기 한 마리가 허준의 비웃듯이 기웃기웃거리다가 문득 바람에 날리며 높게 낮게 한껏 자유롭게 바다 위로 떠갔다"

"또한 가을철에 국화꽃과 잎을 말렸다가 베게 속에 넣어 잠자리에세 꽃향기가 나도록 했다"

"그는 도중에 차를 세우고 봉투 속의 수표액을 확인해볼까 하다가 일단 안전한 곳까지 가지로 했다"

(4) 인명 등 고유명사가 잘못 표기된 문장

예) "새로운 왕조의 태조가 될 꿈을 굳힌 이씨는 다섯째 아들 병원으로 하여금 포은 정몽주의 의향을 타진하게 하였다"

이상과 같이 발성 목록으로 부적절 하다고 판단되는 20 문장을 삭제하고 단계 2. ~ 단계 9.의 과정을 다시 수행하여 최종적으로 PBS 589 문장을 추출하였다. 그림 4.2 에 602 문장을 추출하는 과정의 엔트로피 변화를 나타내었다.

모집단 10,000 문장과 PBS 에는 모두 서로 다른 1,091 종의 음운 환경을 포함하고 있으며, 출현한 음운환경의 총 빈도수는 모집단에서 692,454, PBS 에서 40,129 이다. 또한 모집단에서의 음운 엔트로피는 7.786559 인데 반해 8.295044 로 음운 환경의 균형이 모집단에 비해 좋아 졌다. 추출된 PBS 의 음절수 평균은 37.83, 어절수 평균은 12.96 으로 평균 1 어절은 2.92 음절을 갖고 있는 것으로 나타났다.

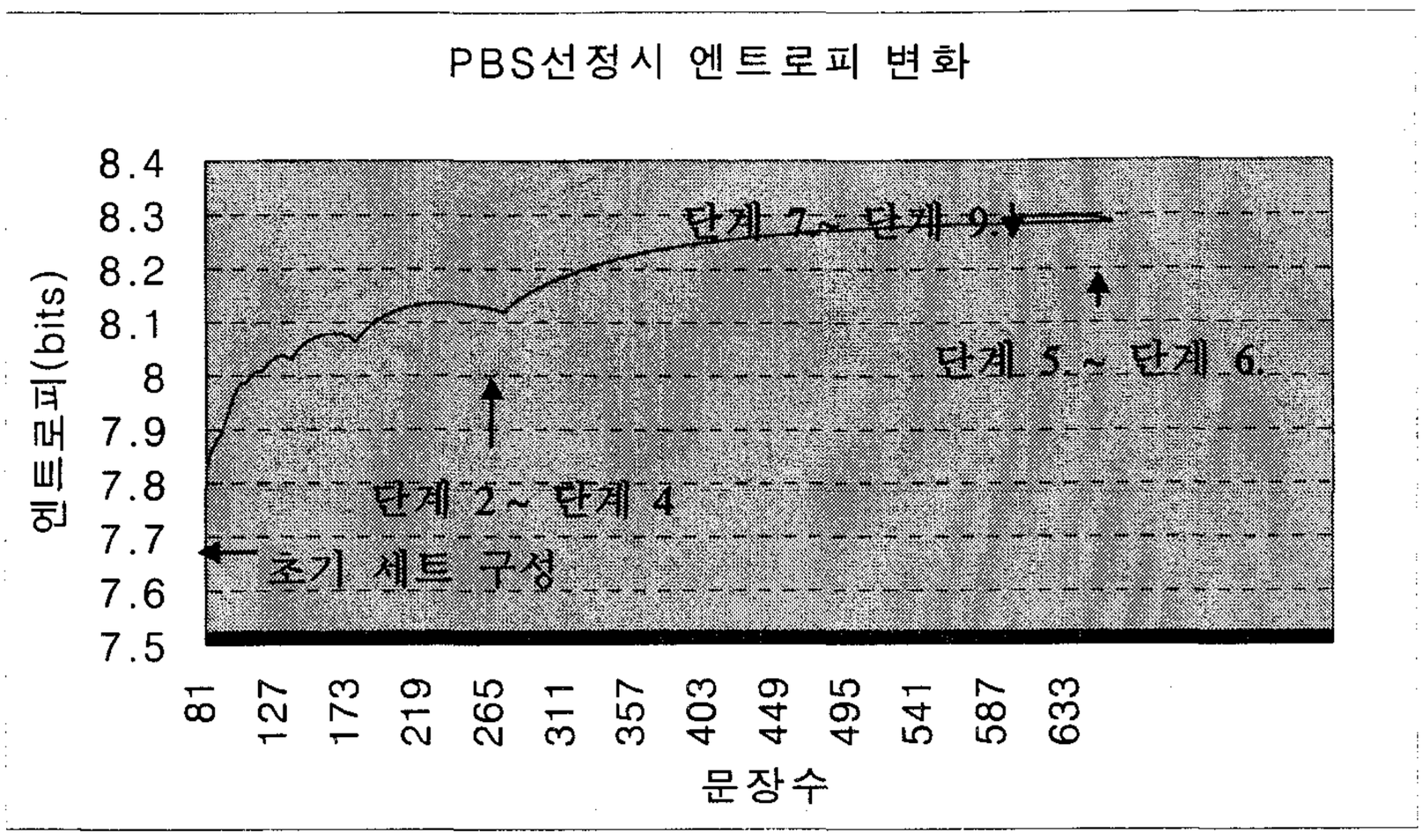


그림 4.2 PBS 추출 과정의 엔트로피 변화

그림 4.3 과 그림 4.4 는 모집단 10,000 문장과 PBS 에서의 음운 환경의 출현 빈도를 나타낸 것이다. PBS 에서 각 음운 환경의 출현 빈도가 모집단에 비해서 고른 분포를 보이고 있다.

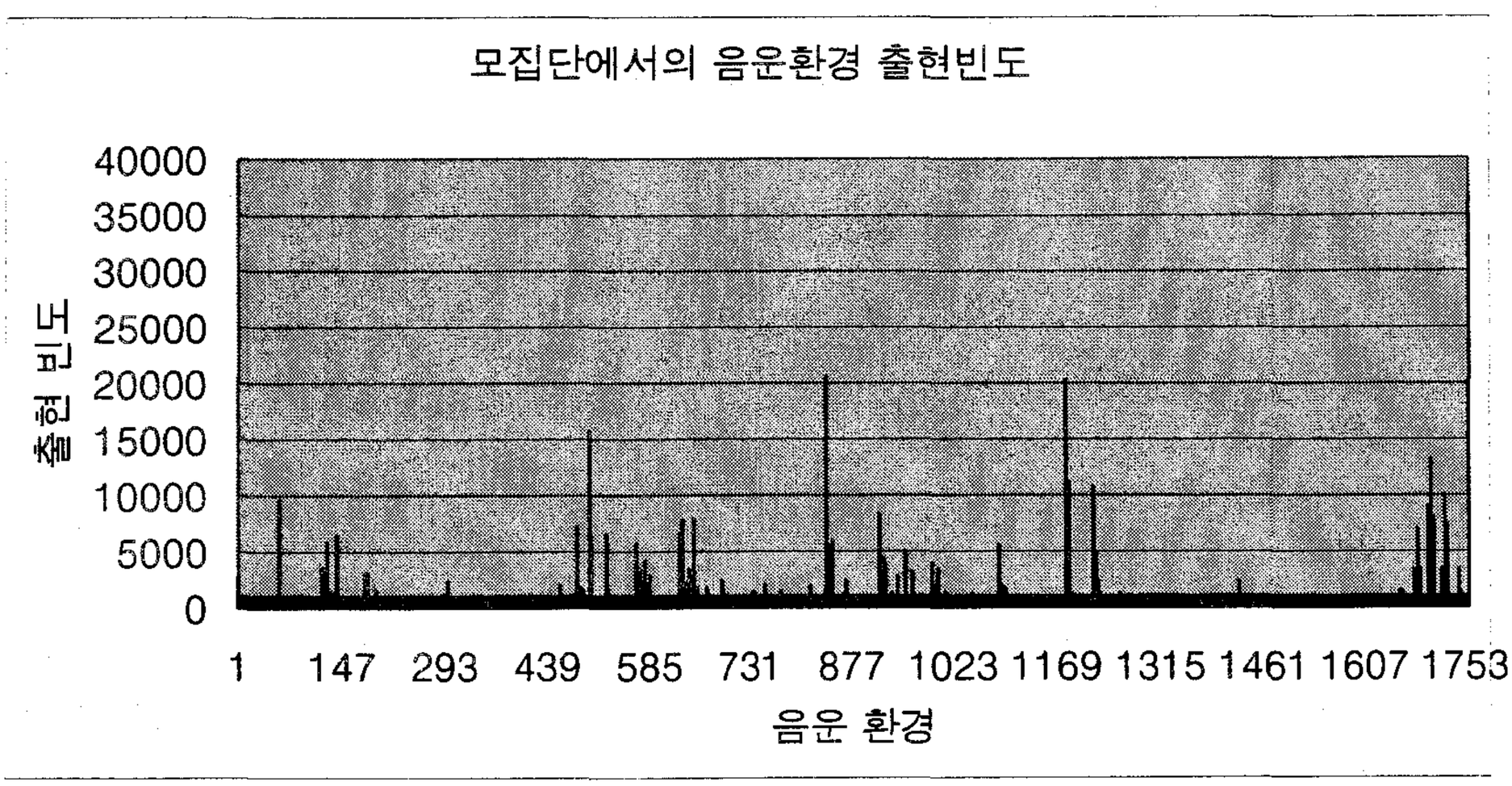


그림 4.3 모집단 10,000 문장에서의 음운 환경 출현 빈도

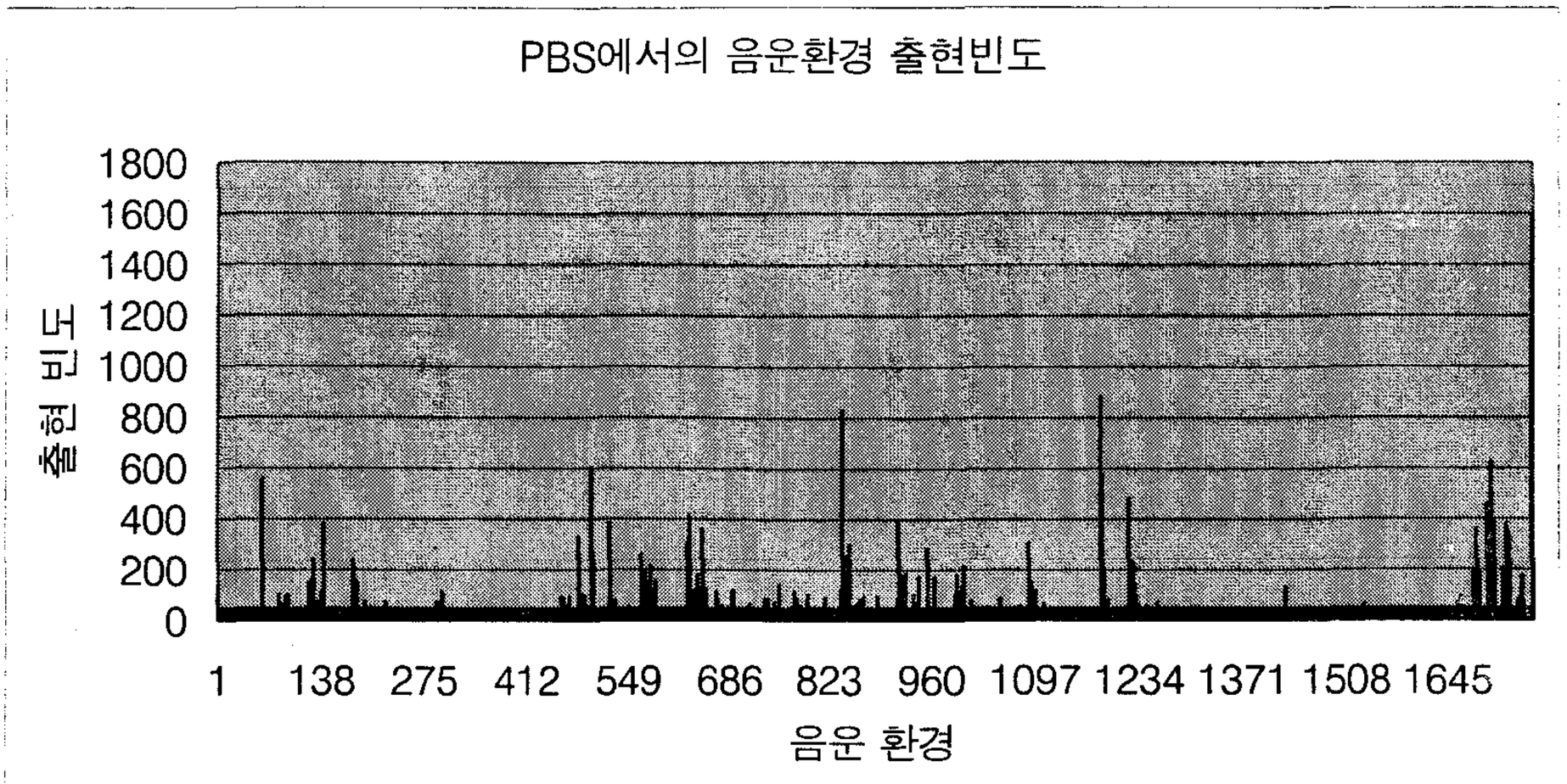


그림 4.4 PBS 에서의 음운 환경 출현 빈도

추출된 PBS 목록은 부록 1.에 그리고 음소별 출현빈도 및 음운 환경별 출현 빈도에 대한 상세한 자료는 부록 2.에 실려 있다.

2 절. 문장 음성 DB 의 구축

본 절에서는 추출된 PBS 를 발성 목록으로 사용하여 음성 시료를 수집하고 이를 음성 DB 로 구축하는 과정 및 구축된 음성 DB 를 효율적으로 검색하는 방안에 대하여 기술한다.

1. 발성 목록의 구성

추출된 PBS 는 589 문장으로 한사람의 화자가 발성하기에는 너무 많은 양이다. 따라서 이를 몇 개의 세트로 나누어서 여러 화자의 음성시료를 수집하여야 한다. 그러나 화자 독립 음성 인식기의 훈련을 위해서는 동일한 내용을 가급적 많은 화자가 발성한 데이터도 필수적이다. 따라서 본 연구에서는 추출된 589 문장의 PBS 에서 50 문장의 공통 세트를 선정하고, 나머지 539 문장에 한하여 각 문장 당 10 명의 화자가 발성할 수 있도록 발성 목록 세트를 구성하였다.

추출된 589 문장의 PBS 중에서 삭제될 경우 PBS 의 엔트로피를 상승시키거나 가장 적게 떨어뜨리는 문장을 차례로 삭제하여 50 문장을 공통 세트로 선정하였다. 그림 4.5 에 공통 세트 50 문장을 선정하는 과정에서의 엔트로피 변화를 나타내었다.

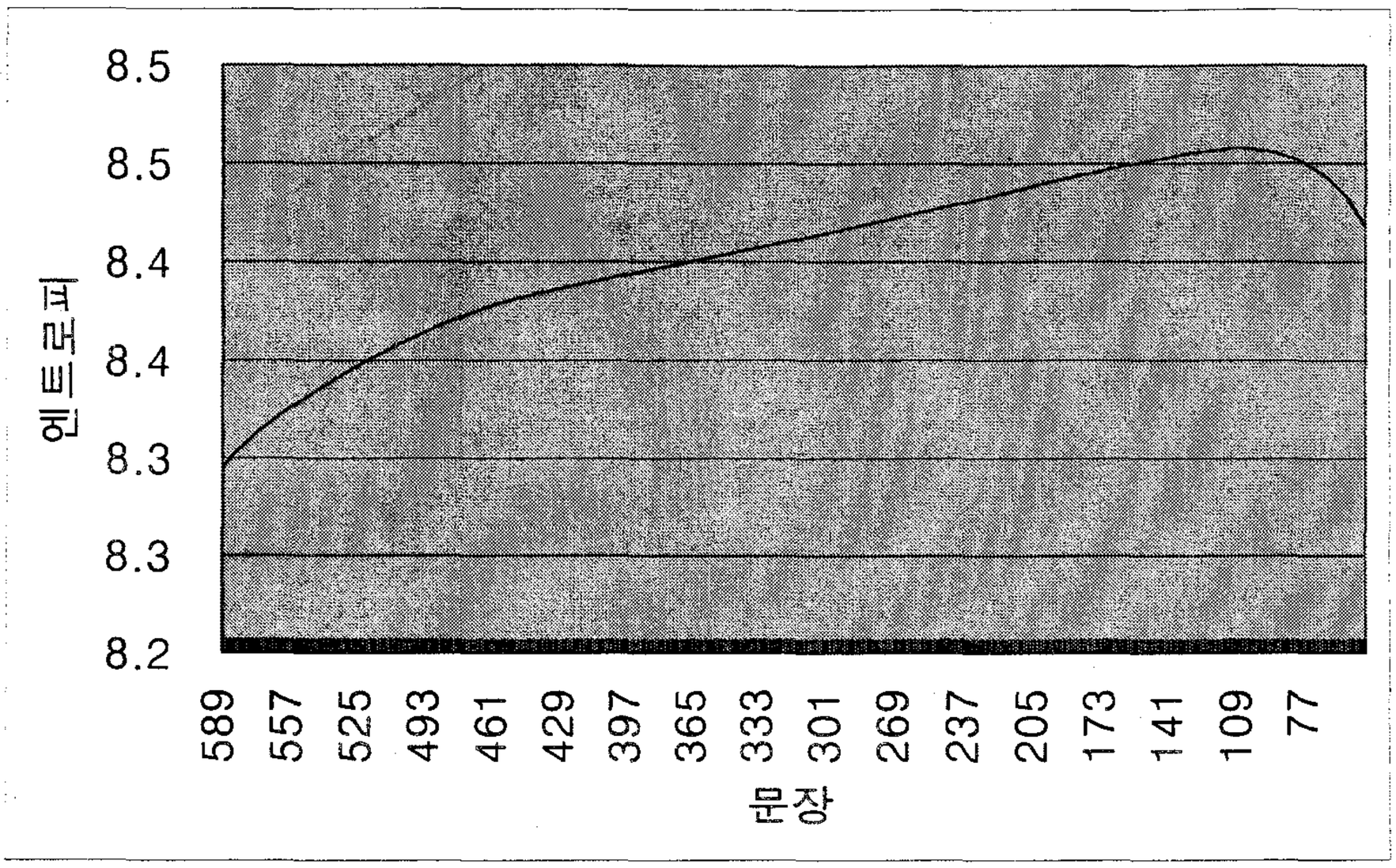


그림 4.5 공통 세트 선정 과정에서의 엔트로피 변화

이와 같이 선정된 50 문장을 제외한 나머지 539 문장은 각 문장 당 10 명의 화자가 발성할 수 있도록 108 문장으로 구성된 4 개의 세트와 107 문장으로 구성된 1 세트로 구분하였다. 따라서 각 화자는 공통 세트 50 문장과 나머지 5 세트 중 1 세트를 1 회 발성하여 평균 158 문장을 발성하게 된다.

2. 음성 시료의 수집 및 A/D 환경

선정된 발성 목록을 사용하여 표준어를 사용하는 일반인을 대상으로 음성 데이터를 수집하였다. 50 명분의 음성 시료 수집을 완료하였으며, 음성은 방음 부스에서 Senheizer HMD224X 를 사용하여 녹음하였으며, 발성된 데이터는 디지털 오디오 테

이프에 저장하였다. 디지털 오디오 테이프에 저장된 음성 시료는 SUN SPARCStation 20 상에서 DATLink 를 통하여 16kHz, 16Bit 로 양자화 하였다.

3. 음성 데이터의 편집

A/D 하여 컴퓨터의 하드 디스크에 저장된 음성 데이터는 그 양이 많으므로 각기 필요 없는 부분은 제외하고 사용할 수 있는 부분만으로 다시 편집을 하여야 한다. 그 과정은 우선 양자화 된 음성 데이터를 대상으로 자동 끝점 추출 알고리즘을 사용하여 음성 데이터 편집 툴을 구현한다. 그리고 이를 이용하여 필요한 음성 데이터를 찾아내고, 불량 데이터의 제거, 수정 데이터의 삽입, 음성 데이터의 앞뒤에 일정한 길이의 무음구간의 확보 등 전반적인 사항을 수정한다.

음성 구간의 자동 끝점 검출을 위한 알고리즘은 여러 가지가 있으나 발성 데이터가 외부 잡음이 없는 방음실에서 수록된 것이기 때문에 음성 구간의 에너지와 영교차율만을 이용한 알고리즘을 사용하여 끝점을 검출해도 좋은 성능을 나타낸다

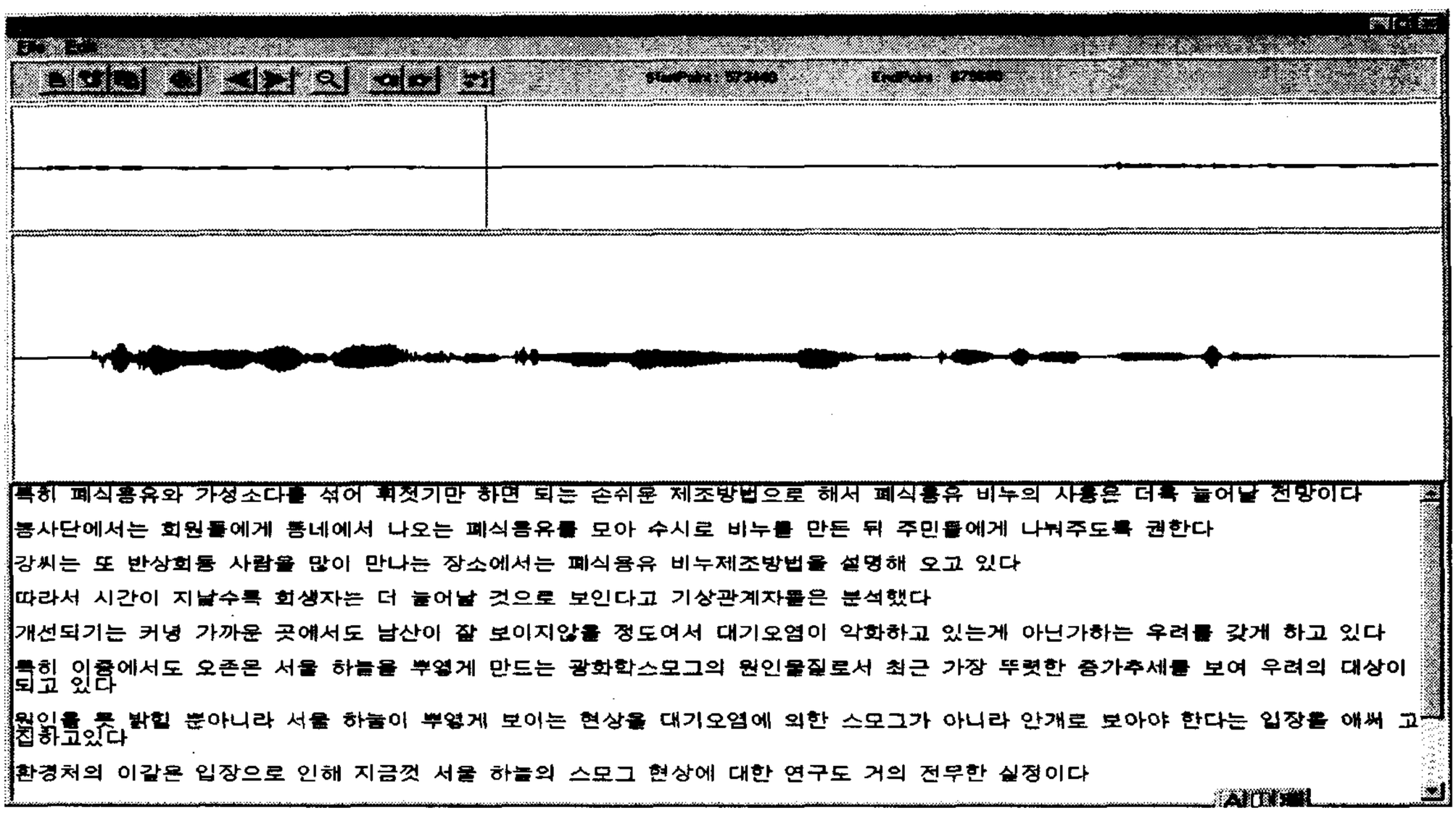


그림 4.6 자동 끝점 검출 알고리즘을 이용한 음성 데이터의 편집

이와 같은 과정을 거친 각각의 데이터파일을 디렉토리 구조로 저장하여 CD-ROM 형태로 제작되었을 때 효율적으로 검색하고 활용할 수 있도록 하였다. 그림 4.6 에 자동 끝점 검출을 통하여 음성 데이터를 편집하는 과정을 나타내었다.

4. 음성 데이터 파일의 구조 및 디렉토리 구조

음성 데이터의 수정 및 편집 과정을 거친 후 음성 데이터는 각각 파일로 저장한다. 일반적으로 파일을 저장하는 방법은 편집된 데이터를 하나의 큰 데이터로 묶어서 필요한 파일만을 찾아낼 수 있는 dictionary 파일과 같이 사용하면 데이터를 관리하기가 편하지만 개개의 파일을 복사할 때 번거로움이 따른다.

반대로 음성 데이터 파일 개개에 하나씩 이름을 붙여서 사용하면 파일을 관리하기는 어렵지만 필요한 파일을 복사하거나 이용하기가 용이하다. 여기서는 각각의 데이터를 디렉토리 구조로 저장함으로써 효율적으로 데이터를 검색할 수 있다. 이는 CD-ROM 화하기에도 용이한 방법이다. 음성 데이터 파일이 저장된 디렉토리 구조와 파일 구조는 2 차년도에 제안된 방법을 그대로 따르고 있으므로 파일 및 디렉토리 구조에 대한 자세한 사항은 2 차년도 최종 보고서를 참조할 것.

5. 음성 DB 의 검색

구축된 음성 DB 총 50 명의 화자가 평균 108 문장을 1 회 발성하였다. 따라서 모두 5400 여개의 음성 데이터 파일로 이루어져 있다. 따라서 단순히 디렉토리 구조만을 가지고 효율적으로 원하는 음소 환경을 갖는 문장을 검색하기는 어렵다.

본 연구에서는 구축된 음성 DB 의 효율적 검색을 위하여 음성 DB 검색 툴을 구현하였다. 음성 DB 검색 툴에서 음성 DB 의 내용을 검색할 수 있는 방법은 다음의 3 가지이다.

(1) 발성 목록의 검색

음성 DB 에서 직접 철자(grapheme)의 형태로 발성 목록을 검색하는 방법이다. 찾으려고 하는 철자를 포함하고 있는 문장을 검색하고 그 결과를 사용자에게 제시한다.

(2) 음소열 검색

음성 DB 에서 찾으려고 하는 음소(phoneme)의 형태로 발성 목록을 검색하는 방법이다. 찾으려고 하는 음소열을 포함하고 있는 문장을 검색하고 그 결과를 사용자에게 제시한다.

(3) 유형별 검색

음성DB를 다양한 방법으로 검색할 수 있는 방법이다. 음성DB에 포함된 음소열을 여러 가지의 유형별로 3 음소열까지 검색할 수 있도록 해준다.

1) 자음의 유형

자음을 검색하기 위한 유형 표기의 프로토 타입은 다음과 같다.

Prototype : c(phoneme, manner, position, strength)

- phoneme : 자음 음소를 직접 표기할 경우 사용한다. 생략할 수 있다
- manner : 조음 방법을 나타낸다. 생략할 수 있으며 다음과 같은 5가지의 토큰이 있다.
  - nasal : 비음
  - stop : 파열음
  - fricative : 마찰음
  - affricate : 파찰음
  - liquid : 유음
- position : 조음점을 나타낸다. 생략할 수 있으며 다음과 같은 5개의 토큰이 있다.
  - bilabial : 양순음
  - dental : 치경음
  - palatal : 경구개음
  - velar : 연구개음
  - glottal : 성문음
- strength : 기식의 정도를 나타낸다. 생략할 수 있으며 다음과 같은 3개의 토큰이 있다.
  - lenis : 연음

- **aspirated** : 기식음
- **fortis** : 경음

위와 같은 유형 표기를 적용하여 자음을 효율적으로 검색하기 위한 방법은 예를 들어 연음만을 찾고 싶은 경우 "c(, , , lenis)"로 표현하면 연음만을 검색하고, "c(,,,)"와 같이 표기하면 모든 자음을 다 검색한다.

2) 모음의 유형

모음을 검색하기 위한 유형 표기의 프로토 타입은 다음과 같다.

Prototype : v(phoneme, horizontal position, vertical position)

- phoneme : 모음 음소를 직접 표현할 경우 사용한다. 생략할 수 있다.
- horizontal position : 수평 조음 위치를 나타낸다. 생략할 수 있으며 수평 조음 위치는 다음과 같은 3개의 토큰을 갖는다.
  - front : 전설 모음
  - mid : 중설 모음
  - back : 후설 모음
- vertical position : 수직 조음 위치를 나타낸다. 생략할 수 있으며 수직 조음 위치는 다음과 같은 3개의 토큰을 갖는다.
  - high : 고모음
  - mid : 중모음
  - low : 저모음

위와 같은 유형 표기를 적용하여 모음을 효율적으로 검색하기 위한 방법은 예를 들어 전설 모음만을 찾고 싶은 경우 "v(, front,)"로 표현하면 전설 모음을 검색하고, "v(,,)"와 같이 표기하면 모든 모음을 다 검색한다.

위에서 정의된 유형 표기 프로토 타입을 따라 음소열을 검색하는 방법은 각 음소 유형 사이에 "+" 토큰으로 연결하여 표기하면 된다. 예를 들어 "v(,,) + c(, , , lenis) + v(,,)"와 같이 표기하여 검색하면 모음과 모음사이에서 유성음화될수 있는 연음을 갖고 있는 모든 발성목록을 검색하고 그 결과를 사용자에게 제시한다. 다음 그림 4.7 에 본 연구에서 구현된 음성 DB 검색 툴을 이용하여 2 차년도에 구축된 PBW 음성 DB 를 검색하는 과정과 그림 4.8 에 그 결과를 나타내었다.

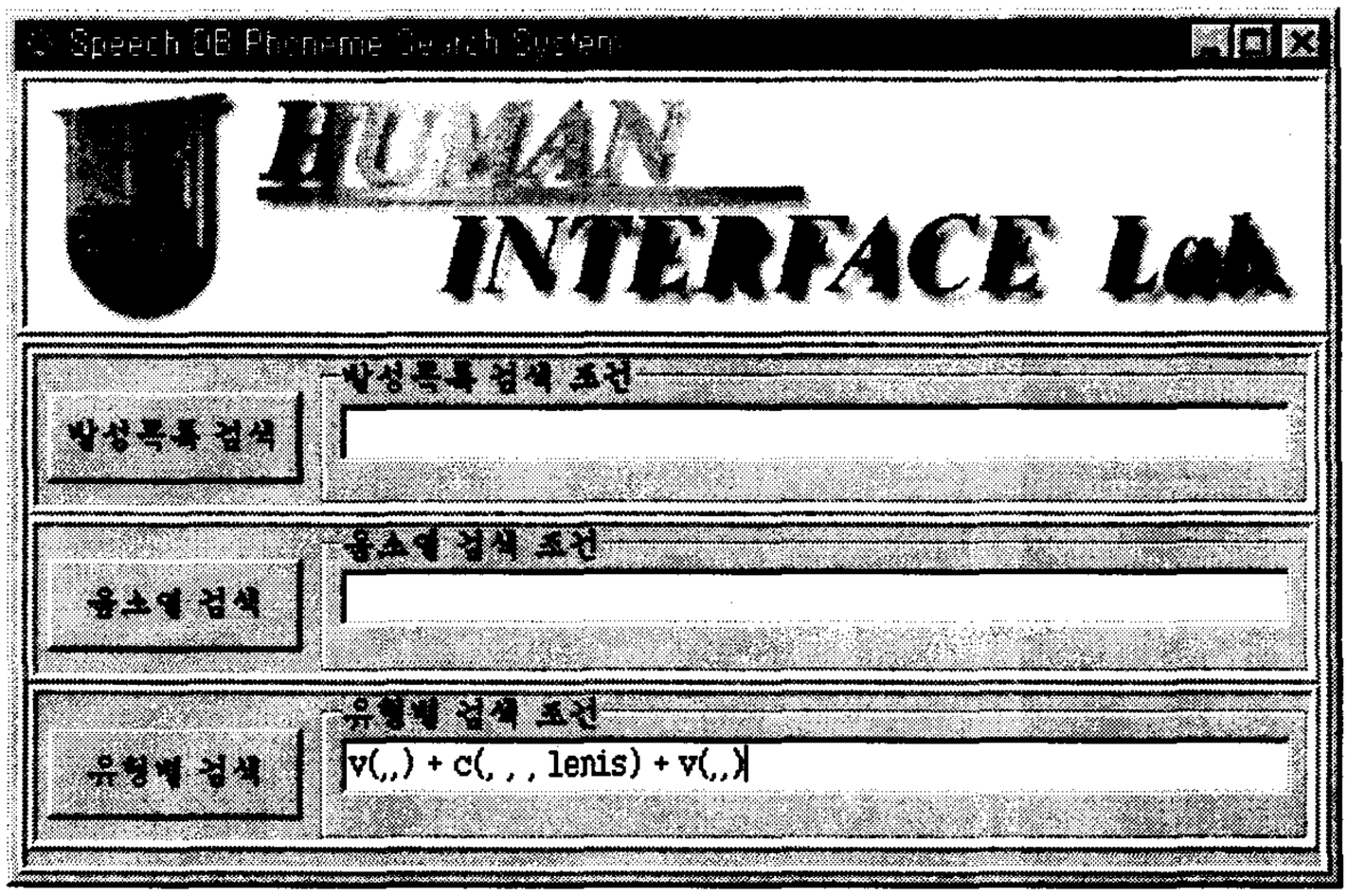


그림 4.7 음성 DB 의 검색

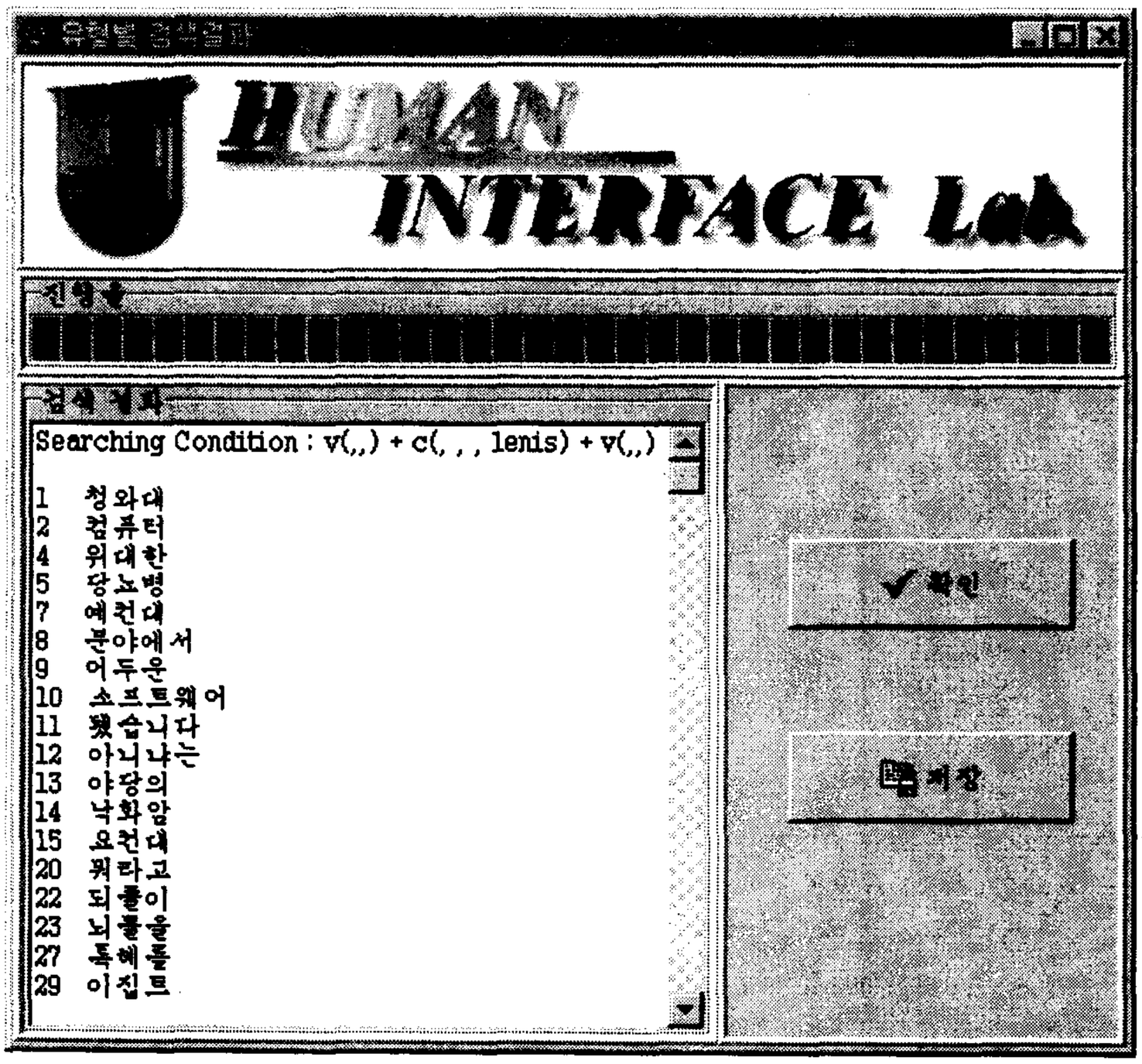


그림 4.8 음성 DB 의 검색 결과

5장. Dictation 시스템을 위한 DB의 검토

1절. Dictation 시스템 개요

최근의 음성인식연구의 방향을 크게 두 가지로 나눈다면 하나는 자연스러운 대화를 인식하는 대화시스템(dialog system)과 낭독문을 중심으로 대어휘 연속음성을 인식하는 구술시스템(dictation system)의 연구가 될 것이다. 특히 구술시스템은 이른바 받아쓰기기계 또는 음성타이프라이터와 같은 개념으로 대어휘의 문장을 연속적으로 발성하여 이를 인식해 내는 기술이 근간이 된다.

문장단위로 발성된 음성을 연속적으로 인식하는 "연속음성인식"기술은 또한 음성인식기술의 중심적 과제이다. 특히 수만 단어 이상의 어휘를 다루는 대어휘 연속음성인식은 일상적으로 우리 인간이 발성하는 음성언어의 거의 모두를 처리대상으로 하므로 음성인식기술의 응용 분야 확대에 매우 중요한 과제이기도 하다. 최근, 미국이나 유럽에서는 음성/언어의 데이터 및 모델을 공통화시켜 대어휘 연속음성인식시스템의 성능을 비교하고 이를 통해서 각각의 요소기술을 평가하므로서 시스템 성능을 비약적으로 향상시키고 있다. 최근의 보고에 의하면 2만단어의 어휘를 갖는 신문 낭독문의 인식을 범용 워크스테이션을 이용하여 실시간의 수배 정도로 실행하여 단어에러율 10%이하의 인식정확도를 달성하는 것이 표준적인 시스템의 성능이다. 국내의 경우 대어휘연속음성의 연구 예나 이를 위한 평가방법, 그리고 이를 평가하기 위한 공통의 데이터에 대한 연구가 아직 없다. 특히 텍스트 및 음성 코퍼스는 이러한 기술 개발에 있어서 초기의 훈련과정에서뿐만 아니라 객관적인 성능평가를 위해서 초기부터 확보하여야할 연구환경이다. 본 연구에서는 대어휘연속음성인식에 포함된 여러 요소기술의 성능을 신속하고 엄밀하게 평가하기 위한 기반으로써 텍스트/음성 코퍼스의 확보를 위한 사항들을 검토하였다. 특히 언어모델 학습용의 대규모 텍스트코퍼스를 정비하여 음향음성모델의 연구를 위한 표준적인 언어모델을 제공하고, 언어모델의 연구에는 표준적인 음향음성모델을 제공할 수 있도록 고려한다. 따라서 이러한 것은 대어휘연속인식뿐만 아니라 음성인식기술 전반에 걸쳐서 다음과 같은 효과가 기대된다.

가. 요소기술간의 상호비교를 쉽게하므로서 첨단 연구영역의 기술진보를 가속시킨다.

나. 요소기술이 공통화되므로 음성인식기술을 응용한 제품 및 서비스

개발이 신속하게 이루어질 수 있다.

다. 음성응용시스템의 실용상의 문제점을 손쉽게 요소기술로 피드백 시켜 해결할 수 있게 한다.

2 절. Dictation 시스템을 위한 음성/텍스트 코퍼스의 연구동향

유럽에서는 신문낭독문의 대어휘 연속음성인식(dictation)을 통한 기술평가가 최근 수년동안 활발히 이루어져왔고 이들 기술개발 및 평가에 있어서 여러 종류의 텍스트코퍼스가 담당한 역할은 크다.

미국에서는 1992 년경에 DARPA(Defence Advanced Research Project Agency)의 주도하에 신문기사(Wall Street Journal)를 대상으로한 대어휘 연속음성인식 연구가 시작되었다. 어휘사이즈를 5K, 20K 로 설정하여 기본적인 평가조건을 Hub, 음향모델/언어모델의 적응화에 의한 평가 등 부대적인 평가조건을 Spoke 라고 부르며 여러 평가 조건아래에서 활발한 연구가 진행되고 있다. 1992 년에는 "5K 어휘/미지어 불포함" 의 신문기사 태스크 상에서 16.6% 의 단어에러율이 보고되었다. 그후 2 년간 에러율이 1/3 에서 1/4 까지 줄었다. 최근에는 어휘제한이 없는(20K 어휘에 1 단어 이상의 미지어가 포함된) 태스크에서 6.6%의 에러율이 보고되고 있다. 이들 결과는 n-gram 이라고 하는 단순한 것임에도 언어모델을 도입하므로서 수만단어의 어휘를 가진 연속음성을 인식할 수 있다는 것을 보인 것이다.

한편, 유럽에서도 SQALE(Speech recognition Quality Assessment for Linguistic Engineering)프로젝트에서 영어(미국식, 영국식), 불어, 독어에 관한 연속음성인식을 평가하였다. 이 프로젝트에서는 언어모델 및 음소모델의 학습데이타량을 언어간에 거의 균등하게 하여 상호비교를 쉽게 함과 동시에 연구기관간에도 단어의 음소사전을 공통으로 사용하도록 하였다. SQALE 의 결과는 영어 이외의 유럽어에 있어서도 영어와 똑같은 방법에 의해 대규모의 연속음성인식이 가능하다는 것을 보이고 있다. 일본의 경우는 최근 음성인식의 대규모화에 대응하여 일본음향학회에서 연속음성데이타베이스가 구축되어 음소모델 학습용의 데이터베이스로서 널리 이용되고 있다. 그러나 연속음성인식용 텍스트데이터에 관해서는 정비가 이루어지고 있지 않다가 일본 정보처리학회의 음성언어연구연락회가 중심이 되어 "대어휘 연속음성인식연구를 위한 데이터베이스 정비 워킹그룹"이 발족되어 1995

년 11 월부터 1997 년 10 월까지를 목표로 대어휘의 텍스트코퍼스, dictation 을 위한 음성데이타베이스 그리고 Dictation 을 위한 기본모델 및 개발도구의 구축을 목표로 활발한 활동을 보이고 있다. 특히 이 그룹에서는 마이니치신문 4 년분의 기사를 이용하여 선정된 문장을 대상으로 150 명 정도의 화자가 총 15000 문장 정도의 신문기사 낭독음성을 수록하는 것을 목표로 추진중이며 낭독 대상문은 대어휘용(2 만단어급), 중어휘용(5000 단어급)의 단어세트를 기본으로 문장의 길이나 복잡성을 고려하여 선택하였다. 아울러서 신문기사의 텍스트를 이용하여 bigram, tri-gram 과 같은 기본적인 언어모델도 정비하고 있다.

그러나 국내의 경우 대어휘 연속음성인식 특히 dictation 시스템 연구의 예가 아직 없으므로 이를 위한 텍스트 및 음성코퍼스의 연구 예가 없다.

3 절. 텍스트 및 음성코퍼스 구축을 위한 기초검토

Dictation 시스템과 같은 대어휘 연속음성인식을 위해서는 먼저 언어모델을 훈련할 텍스트코퍼스를 확보하고 이 코퍼스로부터 평가를 위한 음성 DB 를 제작하여야 한다. 따라서 그중 먼저 텍스트코퍼스를 어떻게 설계할 것인가를 검토한다.

1. 코퍼스의 선택

대어휘 연속음성인식 연구를 위해 텍스트데이타를 정비하는 목적은 언어모델 학습을 위한 데이터정비와 평가용 낭독문의 선정에 있다. 문체와 내용이 어느 정도 통일된 대량의 텍스트가 전자적으로 이용 가능하기 때문에 신문기사를 대상으로 하는 것이 대부분이다. 미국은 월스트리트저널, 프랑스는 르몽드지, 일본의 경우는 마이니치신문을 그 대상으로 하는 것이 좋은 예이다.

이러한 신문기사는 문자언어이므로 자연스럽게 낭독하는데는 곤란한 표현이 존재할 수 있다. 따라서 이러한 문장은 평가용 낭독문의 후보에서는 제외하여야 한다. 또 이들 표현에 통계적 특징이 있을 경우에는 언어모델에 악영향을 미칠 수가 있으므로 언어모델의 학습용 데이터로부터도 제외하는 것이 좋다. 따라서 낭독을 염두에 둔 신문기사의 코퍼스화를 위해서는 다음과 같이 적절치 못한 표현을 제외하여야 한다.

가. 문장이 아닌 기사 및 단락

주식시황의 표, 인사정보, 스포츠 결과 등과 같이 기사나 단락자체가 문장이 아닌 경우에는 삭제

나. 괄호나 특수기호 등

괄호의 경우는 상황에 따라 삭제할 경우와 그렇지 않은 경우가 있다. 따라서 텍스트의 모든 경우를 면밀히 검토하여 삭제여부를 판정한다. 기호의 경우도 텍스트상에서의 용례에 따른 검토가 필요하다.

2. 낭독문장세트의 구성

앞서 논의한바와 같은 처리를 거친 코퍼스로부터 음성인식의 평가를 위한 낭독용 문장세트를 선정하여야 한다. 그 기준으로는

- 문장에 포함된 단어의 종류(어휘)
- 문장의 길이
- 문장의 복잡도

등을 고려하여 밸런스가 고려된 문장을 선택하여야 한다.

(1) 어 휘

어휘의 경우, 외국의 예에서는 주로 고빈도 5000 단어수준의 것과 20000 단어수준의 것을 고려하는 경우가 많다. 이는 실제적으로 대어휘연속음성인식의 훈련 및 평가목적으로 쓰이므로 이 정도가 일반적이다. 여기에 미지어의 문제를 고려하여 5000 단어세트나 20000 단어세트에 해당단어세트에 포함되지 않은 한 두개 정도의 단어를 포함한 문장세트도 구성하고 있다. 우리말의 경우도 이런 기준을 참고하여야 할 것으로 생각된다.

(2) 문장의 길이

너무 긴 문장이나 짧은 문장은 형태소 해석에러가 포함될 경우가 많고 낭독도 곤란할 경우가 많으므로 통상 20 단어(코퍼스의 문장을 통계적으로 검토후 조정)를 중심으로 긴 것과 짧은 것을 적절히 배합할 필요가 있다. 또한 문장길이의 상한과 하한도 전체 코퍼스의 통계적인 자료를 토대로 설정하여야 한다.

(3) 문장의 복잡도

문장의 복잡도는 언어모델 학습용의 코퍼스를 이용하여 학습한 bigram 언어모델에 의해 계산된 단어 perplexity 를 기준으로 하는 것이 일반적이다. 이 기준에 따라 낭독문을 저/중/고 의 3 종류로 분류한 후 이들을 적정한 비율(예를 들면 1:3:1)로 선택한다.

3. 기타 사항

이와 같은 텍스트 및 음성코퍼스의 설계를 위해서는 충분한 양의 전자화된 신문기사 모음이 필요하고 이들은 결국 단어단위의 해석이 필요하므로 형태소단위로 태깅되어야한다. 신문기사의 양은 외국의 경우 일간신문의 5 년분 정도를 사용하는 것이 보통이나 국내의 경우 최근부터 전자화되기 시작하여, 현시점에서는 2~3 년 정도의 기사가 확보 가능하다. 따라서 이를 대상으로 우선 검토가 이루어질 필요가 있다. 물론 여기에는 사용에 따른 지적소유권의 문제 등도 해결되어야 한다. 형태소 태깅의 경우도 대상으로 하는 기사의 양이 많아 상당한 노력과 시간이 소요될 것이다. 물론 자동 태깅후 수정 작업을 거치는 방법이 고려될 수 있을 것이다. 보다 상세하고 실질적인 검토는 후속 연구로 계속되어야 할 것이다.

6 장. 결 론

지금까지 우리말 음성 DB 구축을 위한 3 개년간의 연구의 마지막 연구 결과를 정리하였다. 본 연구에서는 기 작성된 음소 레이블링 기준 초안에 따라 10 명분의

PBW 452 어절을 레이블링 하였고, 이 과정 중에 레이블링 기준 초안도 보완되었다. 이 보완된 기준을 이용하여 레이블링 DB 의 제작이 더욱 확대되어가야 할 것이다.

또한 다양한 음운 환경을 포함한 문장 세트인 PBS 를 균형된 텍스트 코퍼스(Balanced text corpus)로부터 추출하여, 50 명이 각각 개별 발성 문장과 공통 문장을 포함하여 총 5890 문장을 발성한 데이터가 음성 DB 로 구성되었다. 또한 원하는 음소 환경을 가진 문장을 신속하게 검색할 수 있는 시스템을 구성하여 이용의 효율성을 높였다.

이로서 음소 환경이 고루 포함된 어절 및 문장 음성 DB 가 본 연구를 통해서 확보되어 음성 연구자들이 공통으로 사용할 수 있게 되었다. 또한 향후 추진될 dictation 시스템의 훈련 및 평가용 텍스트 및 음성 DB 를 구축하기 위하여 예비적인 검토가 이루어 졌다. 이 결과는 후속연구로 이어질 것이다.

음성DB 관련 참고 문헌

[1] Shuichi Itahashi, "Recent Speech Database Project in Japan," Proc., 24.1.1, ICSLP 90.

[2] J. Miwa and K. Kido, "Spoken word data collecting System," (in Japanese), Preprints Spring Meeting Acous. Soc. Japan, 1982. 3

[3] NIST: Speech Copora Produced on CD-ROM Media by The National Institute of Standards and Technology(NIST), 1991. 4

[4] Seiichi Nakagawa, "Assessment and Database of Speech Recognition /Understanding Systems," (in Japanese), 일본전자정보통신학회, 1990. 12

[5] S. Itahashi, "Speech Database," (in Japanese), 일본전자정보통신학회, 1987. 4

[6] Y. Sagisaga, et al, "A Large-Scale Japanese Speech Database," Proc., 24.4.1, ICSLP 90.

[7] Joon-Hyuk Choi, et al, "Construction of A Large Korean Speech Database and Its Management System In ETRI," Proc., 24.2.1, ICSLP 90.

[8] S. Itahashi, et al, "A Japanese Language Speech Database," Proc. ICASSP86, Paper 7.4, 1986. 4

[9] S. Itahashi, et al, "Speech database of discrete words," (in Japanese), J. Acous. Soc. Japan, 41, 10, 1985. 10

[10] K. Iso, et al, "Design of a Japanese Sentence List for a Speech Database," Proc. ASJ. 2-2-19, 1988. 3

[11] T. Ehara, et al, "ATR Dialogue Database," Proc., 24.5.1, ICSLP 9012] S. Itahashi, et al, " A Collection and Editing System of Speech Data for Research," 일본음향학회강연논문집, 1982. 10

[13] R. Mizoguchi, et al, "A Subsystem for Supporting the Load of Speech Data in

SPEECH-DB," 일본음향학회강연논문집, 1981. 10

[14] S. Itahashi, et al, "On the Composition of Speech DataBase - Examination of Sampling Rate Transformation Methods," 일본음향학회강연논문집, 1981. 10

[15] M. Hamaguchi, et al, "A Prototype System of Speech Database "SPEECH-DB"- Command Structure and its Examples," 일본음향학회강연논문집, 1981. 5

[16] N. Maeda, et al, "Speech Database "SPEECH-DB" ," (in Japanese), EA81-5617] N Maeda, et al, "A Prototype System of Speech Database "SPEECH-DB" - Its Design Policy and Data Model," 일본음향학회강연논문집, 1981. 5

[18] A. Kurematsu, "ATR Japanese speech database as a tool of speech recognition and synthesis," Speech Communication, 1990. 9

[19] R. Mizoguchi, et al, "Speech Database with an Intelligent Access Mechanism - SPEECH-DB"일본 정보처리학회논문집, 1983. 5

[20] J. Gauvain, et al, "Design Considerations and Text Selection for BREF, a large French read-speech corpus," Proc., 24.6.1, ICLSP 90

[21] K. Tanaka, "The Speech Database for Speech Analysis and Recognition Research," Proc., 24.7.1, ICLSP 90

[22] S. Makino, et al, "A Distributed Speech Database with an Automatic Acquisition System of Speech Information," Proc. 24.91, ICLSP 90

[23] T. Nakajima, et al, "Data File Control Sytem for Speech Research," 일본 음향학회음성연구회자료집 S73-07, 1973. 7

[24] K. Takeda, et al, "Acoustic Labeling in a Japanese Speech Database," 일본 음향학회강연논문집, 1987.3

[25] 이용주, 이정철, 김경태, "음성 데이타베이스 구축에 관하여," 음향학회지, 제7권 제5호, 1988

[26] K. Takeda, et al, "A Japanese speech database for various kinds of research purpose," 일본음향학회지, 44권 10호, 1988

[27] K. Takeda, et al, "Acoustic-phonetic transcripton in a Japanese speech database," Proc. 1st Eur. Conf. Speech Technology, 1987. 10

[28] 이용주, 김경태 외,"단어음성 데이타 수집 및 DB 구성 시스템," 대한전자공학회 추계 종합학술대회 논문집, Vol. 9, NO.2, 1986. 12

[29] 이용주, 김경태 외, "대용량 발음사전 표제어에 나타난 음소의 통계적 성질," 대한전자공학회 통신교환연구회 발표논문집, 1987. 11

[30] K.Takeda, et al, "Construction of an Acoustically-Phonetically Transcribed Japanese SpeechDatabase," (in Japanese), SP87-19, 1987

[31] 김경태, 최준혁, 이용주, "음성 데이타베이스 관리 시스템의 구축,"Korea-Japan JointSymposium on Acoustics, 1991. 7

[32] K. Shirai, et al, "Speech database projects in Japan - present and future," Proc. of ESCA Workshop, Speech I/O Assessment and Speech Database, 1989. 9

[33] K. Tanaka, et al, "A Demiphoneme Network Representation of Speech and Automatic LabellingTechniques for Speech Database Construction," Proc., 7.1.1, ICASSP 86

[34] F. Guyote, et al, "A Speech database at the United States Air Force Academy," Proc., 7.2.1, ICASSP 86

[35] S. David, "A PCM/VCR Speech Database Exchange Format," Proc., 7.3.1, ICASSP 86

[36] K. Shikano, "Phonetically balanced word list based on information entropy," Preprints Autumn Meeting Acous. Soc. Japan, Paper 3-3-10, 1984. 3

[37] S. Hayamizu, et al, "Generation of VCV/CVC Balanced Word Sets for Speech Database," 일본 전자기술총합연구소, 제49권 제10호, 1985

[38] R. Mizoguchi, et al, "A Speech Labeling System Based on Knowledge Processing," (in Japanese) SP89-85, 1989

[39] J. CUI, et al, "The Phonetic Database of the Chinese Speech Sounds," 일본음향

학회강연논문집 1991. 3

[40] K. Akiba, et al, "Speech Database for Research of Japanese Speech Recognition," 일본음향학회 강연논문집, 1982.3

[41] M. Hamaguchi, et al, "V/U/S Classification Algorithm and its Performance Evaluation using Speech Database," (in Japanese), EA81-57

[42] G. Doddington, "The next generation DARPA speech recognition/natural language database," Proc. of the ESCA Workshop, Speech I/O Assessment and Speech Database, 1989. 9

[43] H. Fujisaki, "Overview of the Japanese national project on advanced man-machine interface through spoken language," Eurospeech 87

[44] J.m. Baker, et al, "Speech Recognition Performance Assessments and Available databases," Proc.,ICASSP 83

[45] 정유현, 최준혁 외, "공통 음성 데이타베이스 구축을 위한 사전 조사 연구," 전자공학회 하계 학술 대회, 1992. 6.

[46] 한국전자통신연구소, *자동통역전화를 위한 요소 기술 개발*, 최종보고서, 1991. 12.

[47] Michel Weintraub, et al, "Constructing Telephone Acoustic Models from a High-Quality Speech Corpus," 1994 IEEE ICASSP, Vo1, pp I85-I88.

[48] Thomas Staples, et al, "The Voice Across Japan Database-The Japanese Language Contribution to POLYPHONE," 1994 IEEE ICASSP, Vo1, pp I89-I92.

[49] Ronald Cole, et al, "Towards Automatic Collection of the U.S. Census," 1994 IEEE ICASSP, Vo1, pp I93-I96

[50] Bruce Millar, et al, "The Australian National Database of Spoken Language," 1994 IEEE ICASSP, Vo1, pp I97-I100

[51] John garofolo, et al, "The Development of File Formants for Very Large Speech Corpora:Speech Corpora: SPHERE and SHORTEN," 1994 IEEE ICASSP, Vo1, pp I113-I116

[52] Carlos Ribeiro, et al, "A Software Tool for Speech Collection, Recognition and Reproduction," EUROSPEECH-93, Vo1, pp 179-182, Berlin 1993.

[53] Christoph Draxler, et al, "Prolog Tools for Accessing the PhonDat Database of Spoken German," EUROSPEECH-93, Vo1, pp 191-194, Berlin 1993.

[54] G.Castagneri, G.Di Fabbrizi, A.Massone, M.oreglia, "SIRVA-A Large Speech Database Collected on the Italian Telephone Network," EUROSPEECH-93, Vol, pp 199-201, Berlin 1993.

[55] 이용주, 김종진 외, "자유발화음성 및 텍스트코퍼스 구축에 관한 검토," 한국음향학회 제11회 음성통신 및 신호처리 워크샵 논문집, 1994.10.

[56] 최인정, 권오욱 외, "자동통역용 한국어 음성 데이타베이스," 한국음향학회 제11회 음성통신 및 신호처리 워크샵 논문집, 1994. 10

[57] 최승호, 김형근 외, "전화음성데이타 수집에관한 연구," 한국음향학회 제11회 음성통신 및 신호처리 워크샵 논문집, 1994. 10

[58] Yeonja Lim, Yonggjik Lee, "Implementation of the POW Algorithm for Speech Database," ICASSP-95, Vo1, pp 89-92, Detroit 1995.

[59] 이용주, "한국어 음성언어정보처리와 음성 데이타베이스," 한국어정보처리 소식 제2권 제 1,2호 1994. 10

[60] 이용주, 김봉완 외, "해외 음성DB 구축 동향," 한국음향학회 제12회 음성통신 및 신호처리 워크샵 논문집, 1995. 6

[61] 조철우, "음성DB구축을 위한 국제간 활동현황," 한국음향학회 제12회 음성통신 및 신호처리 워크샵 논문집, 1995. 6

[62] 이영직, 류준형 외, "ETRI의 음성 데이타베이스 구축 현황," 한국음향학회 제12회 음성통신 및 신호처리 워크샵 논문집, 1995. 6

[63] 도삼주, 구명완, "한국통신의 음성DB 현황," 한국음향학회 제12회 음성통신 및 신호처리 워크샵 논문집, 1995. 6

[64] 최인정, 박종렬 외, "KAIST 통신연구실의 음성 데이타베이스 구축 현황," 한국음향학회 제12회 음성통신 및 신호처리 워크샵 논문집, 1995. 6

[65] 이용주, 김봉완 외, "국어공학센터의 한국어 음성DB 구축 계획," 한국음향학회 제12회 음성통신 및 신호처리 워크샵 논문집, 1995. 6

[66] 김락용, 김민성 외, "LG전자의 음성 DB 구축 현황," 한국음향학회 제12회 음성통신 및 신호처리 워크샵 논문집, 1995. 6

[67] 김상룡, "삼성종합기술원의 음성DB 구축현황," 한국음향학회 제12회 음성통신 및 신호처리 워크샵 논문집, 1995. 6

[68] 이용주, "음성 데이터베이스의 현황과 과제," 한국음향학회 제13회 음성 통신 및 신호처리 워크샵 논문집, 1996. 8

[69] 김봉완, 이용주 외, "공동이용을 위한 단어음성DB 구축 및 PBS설계에 관한 검토," 한국 음향학회 제13회 음성 통신 및 신호처리 워크샵 논문집, 1996. 8

[70] 김종진, 이용주 외, "한국어 음성DB 구축을 위한 한국어 레이블링 기준에 관한 연구," 한국음향학회 제13회 음성 통신 및 신호처리 워크샵 논문집, 1996. 8

[71] 이용주, 김봉완 외, "음운 환경을 고려한 음성DB 단어세트의 설계," 원광대 논문집 제30편, 1995. 11

[72] 김종진, 이용주 외, "한국어 음성의 음성 자동 분할에 관한 연구," 원광대 논문집, 제32-2편, 1996. 12

[73] 이용주, "우리말의 지역차 연구를 위한 음성DB 구축의 검토," 한국음향학회 호남 충청지회 합동 음향 워크샵 논문집, 1996. 6

[74] 김종진, 이용주 외, "음성DB구축을 위한 한국어 음소 레이블링의 기준에 관한 연구," 한국언어학회 음성학연구회 제4차 학술발표대회, 1996. 6

[75] 이용주, 김봉완, "음성 연구 및 음성 데이터베이스," 대한음성학회 음성학 학

술대회 자료집, 1996. 2

[76] 한국과학기술원, *국어정보처리개발 - 한국어 음성DB 구축에 관한 연구*, 1차년도 최종 보고서, 1995. 7

[77] 한국과학기술원, *국어정보처리개발 - 한국어 음성DB 구축에 관한 연구*, 2차년도 최종 보고서, 1996. 7

[78] Steve Young, and Gerrit Bloothooft eds, *Corpus-Based Methods in Language and Speech Processing*, Kluwer Academic Publishers, Dordrecht, 1997

[79] 이숙향, 고현주 "한국어 ToBI 기준안 및 세그멘테이션 기준," 한국전자통신연구소 최종보고서, 1997. 1

[80] 이기영, 최창식, "포만트천이의 변경규칙을 적용한 반음절 단위 규칙합성," 한국음향학회 제13회 음성통신 및 신호처리 워크샵 논문집, 1996. 8

[81] 武田一載, *大語彙 連續音聲認識用 新聞記事 讀み 上げコ-パス*, 人文學 情報處理 No.12, 勉誠社, 1996.12 (Japanese)

[82] Michael Riley, et al, "The AT&T 60,000 Word Speech-to-Text System," Proc. Vol. 1. EUROSPEECH '95, pp. 207 ~ 210

[83] Jung-Kuei Chen, et al, "Large Vocabulary, Word-Based Mandarin Dictation System," Proc. Vol. 1. EUROSPEECH '95, pp.285 ~288

[84] Xavier Aubert, et al, "Improved Acoustic-Phonetic Modeling in Philips' Dictation System by Handling Liaisons and Multiple, Pronunciations," Proc. Vol. 1. EUROSPEECH '95, pp.767 ~770

[85] J. L. Gauviain, et al, "Design considerations & text selection for BREF, a large French read-speech corpus," ICSLP-90

[86] D. Paul, et al, "The Design for the Wall Street Journal-based CSR Corpus," DARPA speech & Nat. Lang. Workshop, ArdenHouse, NY, 1992. 2

부록 1. PBS 목록

1.공통 세트(50 문장)

001 기자가 몇 개월 동안 쉼 끝에 다시 신문사에서 일하게 됐다.

002 위의 예에서 어떤 대우 표현을 선택하여 쓰느냐 하는 것은 말할이의 교양이나 품위와 밀접한 관련이 있다.

003 조붓한 허리와 함께 얄깃얄깃 노는 그 몸놀림은 웬만큼 흉내를 낼 수도 없었다.

004 낙동역과 한림정역을 거쳐 목적지인 진영에 도착될 때까지 허정우의 마음은 들떠 있었다.

005 거기서 집을 내려다보면 볼품없이 삐쭉삐쭉 솟은 연탄고래 굴뚝이 너덧 개가 보인다.

006 침팬치가 흥겨워 뛸 때에는 한쪽 다리에 힘을 주어 두 박자의 트롯이 된다.

007 요원들이 체포되고 특히 당내투쟁에 의해서 대중과 유리되어 조직은 매우 약화되었던 것이다.

008 민족주의와 절충주의의 두 성격은 러시아의 현대음악에서도 쉽게 찾아볼 수 있다.

009 러시아의 옛 수도원과 사원에서는 러시아 최초의 연대기들이 발견되는데 이들은 전부 이때부터 기록이 시작된 것이다.

010 그는 태어날 때부터 농노였으나 자유의 몸이 되었고 왕립 예술학교에 보내졌다.

011 꿈에서 깬 듯 황급히 지폐와 티켓을 바꿔들고 돌아서던 윤종혁은 뒤에서 급히 달려들던 흑인과 정면으로 부딪혔다.

012 이상이 전력 계통이나 동력 계통이 아니라 전자뇌와 위상 동기 장치에 있음은 확실했다.

013 복숭아빛 피부가 햇빛에 반짝이고 푸른 수염이 이 순간에도 몇 밀리의 몇천 분의 일씩 자라고 있다.

014 학교 당국의 지나친 관료주의적 교육 방법과 뚜렷한 차별 교육은 없는 집 아이들을 소외시키고 미래의 꿈을 잃게 만들었다.

015 뒤를 힐끗 돌아보던 김주사는 못 볼 것이라도 본 듯 황망히 범바우쪽으로 피한다.

016 쉴 새 없는 질문 끝에 마침내 내가 일부러 맨 마지막으로 빼놓았던 질문이 튀어 나왔다.

017 제법 재미가 있을 것 같아 그 책을 좀더 자세히 살펴본다.

018 그 어린이의 답의 정확성이 마이크로 컴퓨터에 부착된 음성 식별기에 의해 평가된다.

019 네이비 블루 빛깔의 시트가 씌워진 트윈 베드와 커다란 창문이 바다쪽을 향해 시원스레 틔어 있어 분위기가 좋은 방이었다.

020 공기 중의 먼지와 땀으로 더러워지기 쉬운 피부에는 무엇보다 깨끗한 세안이 중요하다.

021 우유가 너무 뜨거우면 화상을 입기 쉽고 또 너무 차가우면 모공이 열리지 않아 때를 없애기가 어렵다.

022 무기질 특히 활동성 결핵 환자의 경우 칼슘 소모가 많아 칼슘 평형 유지를 위해 칼슘이 많이 든 음식을 먹어야 한다.

023 어쨌든 경북궁을 짓고 성을 쌓고 성 주위에는 문을 여덟개나 세웠지요.

024 임진각 망향의 제단 앞에서 북녘을 향해 엎드려 울고 있는 둘째 사위의 모습이 눈앞 가득히 보였기 때문입니다.

025 영양분이 자꾸 씻겨 가는데 어떻게 깨끗한 물이 있을 수 있습니까?

026 밤낮 휴일 없이 직장만 알고 직장 일에 전념해 온 직장 생활풍토가 점차 약화됨에 따라 직장풍속도도 크게 변화하고 있다.

027 음극선은 두꺼운 검은색 종이에 싸여 있었으므로 그 빛이 음극선일리가 없었다.

028 그리고 바로 이런 절차를 통해 왼쪽 뇌 우위의 상태에서 오른쪽 뇌 우위의 상태로 바뀐다고 말할 수 있겠다.

029 어떤 사람을 기억해 낼 때 왼쪽 뇌는 그 사람의 얼굴 모습보다는 그의 이름을 기억해 낸다.

030 아들 똘이 녀석과 같은 유치원에 다니는 이웃집 현아 엄마의 전화였습니다.

031 세제를 절약하려면 와이셔츠의 칼라나 양말의 발뒤꿈치는 세탁기에 넣기 전에 부분 세탁이 필요하다.

032 꽃 빛깔로 치자면 싸리꽃의 보라빛이 조금은 더 붉은 빛을 띠었다.

033 저 멀리 회양의 넓고 넓은 눈 덮인 광야가 햇빛을 받아 찬란하게 펼쳐져 있었다.

034 컴퓨터 시스템도 이 권위주의 조직 문화의 도구로 이용되었기 때문에 독재형 패턴을 가질 수밖에 없다.

035 가겟방 여자의 말마따나 물통은 그 색깔과 크기 또 모양새 등 여러 종류였다.

036 불쌍한 물고기들은 너무나 뜨거워서 각자 있는 힘껏 프라이팬 위에서 펄쩍펄쩍 뛰고 있었다.

037 부처 앞에는 위패나 향로 이외에 빨간 초가 산처럼 녹아 있어서 우리를 놀라게 했다.

038 그 후에 이들 부부 사이에 열 명의 아이가 생겨 모두 열다섯 명의 어머니가 되었다.

039 적도지역은 태양 에너지의 순흡수 지역이며 극지역은 태양 에너지의 순방출 지역이다.

040 그래서 견우 아저씨께 큰공을 세우시게 하여 옥황 상제님의 용서를 받게 하고 싶어요.

041 이 집 저 집으로부터 방아 찧는 소리가 쿵덕쿵 쿵더쿵 들려 왔습니다.

042 돌아오셔서는 큰 병이 심해져 그 외투를 입고 외출할 기회도 별로 없었으니까요.

043 그 예쁜 외투가 아마 아직도 수옥이 옷장 어디엔가 걸려 있을 겁니다.

044 세제의 세척작용은 주로 계면활성제의 역할에 의한 것으로 빨래에 붙어 있는 때와 세척액 사이의 표면장력을 줄여 세제의 습윤침투작용에 의해 때를 분리시킵니다.

045 성상세포의 축색돌기는 추체세포의 축색돌기와는 달리 백색질까지 뻗치지 않고 회색질내에서 연장이 그친다.

046 특히 유충의 타액 섭취에 의한 영양 저하에 관한 생리학적 증거가 입증되지 않고 있다.

047 해양오염 문제에 대한 연구와 바다에서 생명에 기초에 대한 연구는 공히 밀접하고 복잡하게 얽혀 있다.

048 왜냐하면 원래의 가트 회원국은 기존의 자국 무역관련 법규를 유지할 수 있기 때문이다.

049 그런데 요즈음 매우 주목할 만한 경향이 우리 학계의 일부에 싹트고 있음을 본다.

050 국민 앞에서의 약속을 지키지 못할 경우라면 그 이유와 책임소재를 공식 기구를 통해 충분히 논의해야 한다.

2.세트 A(108 문장)

001 이후 새로운 학교 교육 제도의 도입으로 국어가 정식 과목이 되었다.

002 얼마전 모 봉사회에 하소연해 온 어느 교포 여성의 얘기를 들으면 더욱 기가 막힌다.

003 두 가지 일들이 내내 번갈아 떠오르며 아쉬움과 분함이 자꾸 교차된다.

004 아울러 능력과 실력보다는 이름 석 자를 먼저 앞세워 평해 본 일이 없는지 반성해 봐야겠다.

005 당연히 그 자리는 앉으면서부터 일어설 때까지 같은 처지의 동료 애기와 함께 원맺힌 한으로 일관해 왔었다.

006 살림이 어려울 대로 어려워졌으나 그는 고문과 숱한 회유에도 굴하지 않고 지금껏 버텨왔다.

007 또 문연은 조직강령으로 고유문화의 정당한 계승 및 세계문화의 비판적 섭취 그리고 인민의 민주주의적 교육 및 과학적 계몽 등을 채택하고 있었다.

008 대우법에서의 상하관계는 언어적 표현 방식과 연관된 몇 갈래의 높낮이 구분에 불과하다.

009 대우법의 사회학적 측면은 언어적 측면과 밀접하게 연관된 사회적 관계에 한정되어 있다.

010 그것은 대우법이 사회적 계층 구별이 엄격한 사회에서 발달되어 왔기 때문이다.

011 위와 같은 경우는 들을이에 대한 직접적 대우 표현과 관련된 품위 문제이다.

012 그뿐 아니라 화제의 인물에 관한 대우 표현에서도 말할이의 인품이 드러나게 된다.

013 세 명의 육중한 사나이들 밑에 납짝하게 깔린 기태는 숨을 쉴 수가 없었다.

014 그들은 납치 장면을 목격하기 전에 남의 집 대문 앞의 어두운 처마 밑에서 뜨거운 입맞춤을 나누고 있었다.

015 그의 인상과 어울리게 가로등 아래에 세워져 있는 그의 자가용도 멋진 코발트색이었다.

016 머리와 양쪽 팔뚝이 잘려나가고 없기 때문에 그런 생각이 든 것 같았다.

017 피가 엉겨붙어 있기에 이상하다 생각하고 더 헤쳐 봤더니 어깨가 나오고 가슴이 나오더군요.

018 절룩거리는 걸음걸이여서 발을 떼어놓을 때마다 발뒤꿈치를 따라 흙탕물 방울이 유난스럽게 튀어올랐다.

019 그래서 우리는 어떤 대상에 대한 목표와 쟁취 욕구를 가지며 인간 대 인간은 끝없는 약육강식을 시도하는 것이다.

020 그 자존심 앞에서 나는 가끔 왜소한 존재가 되어 숨을 헉헉거렸다.

021 어떤 피상을 통해서 느끼는 마음의 작용이기 때문에 안개처럼 서서히 포말될 수밖에 없다.

022 인과의 연쇄와 목적의 연쇄 사이의 차이는 다음과 같은 것에 있다.

023 기침을 하고 외짝 방문을 열어 보니 컴컴한 방안은 텅 비어 있었다.

024 낡은 반닫이 궤 앞에 목침 하나만 눈에 뜨일 뿐 사람이 기거치 않는 썰렁한 냉기만 느껴졌다.

025 거기다 날로 더해 가는 건강의 나쁜 조짐이 그의 의욕과 활동을 더 이상 지탱시켜 주지 못했다.

026 그녀는 검정색 광목 보퉁이의 벌어진 틈 사이에다 기차를 탄 뒤부터 읽어 왔던 얄팍한 신약전서를 끼워넣었다.

027 쫑쫑 땋은 두 갈래의 머릿단이 남색 호박단 저고리 앞으로 드리워져 있었다.

028 여기 진영벌과 대산벌이 합쳐진 이 앞쪽 평야가 김해평야와 맞먹는 큰 곡창지대란 걸 모르는 모양이군.

029 내 한 팔이 자네 두 팔 힘을 감당할 수 있어.

030 고향인 평양서부터 따지자면 밀려 내려오기도 조선 팔도 중 거의 끝에서 끝이었다.

031 꿩요리는 물론이고 정성껏 만든 갖가지 안주가 상에 가득 놓여 있었다.

032 수줍음과 부끄러움으로 두 볼이 발그레해진 징재를 가만히 요 위에 앉힌 뒤 여주가 말했다.

033 그것은 백제가 고구려와 동맹을 맺을 수 없게 하여 백제의 세력을 약화시키

는 동시에 대륙과 직접 교역을 할 수 있었던 것이다.

034 왜냐하면 남생의 편에 설 수도 없고 그렇다고 형을 내쫓고 막리지가 된 아우 남건의 편에 설 수도 없는 진퇴양난의 처지였던 것이다.

035 안시성 성주 대결걸중상은 곧 결의 내용과 양만춘의 국장 거행 전말 등을 복명서에 기록하여 왕성으로 떠나는 특사편에 급히 띄웠다.

036 국내성의 남생 장군에게 밀사를 보내야 할텐데 누구를 보내면 좋겠느냐?

037 모든 걸 다 알고 묻는데야 연헌성도 할 말을 잃고 말았다.

038 아버님의 밀서를 전했더니 몹시 기뻐하셨고 날더러는 직접 황제를 뵈오러 가지 않아도 된다고 하셨소.

039 고생 얘기가 나오면 끝이 없을 것 같아 다시 아름다운 시절로 말머리를 돌렸다.

040 꼭 맞추어진 그 오랜 시간속의 약속을 이제서야 나누는 몸짓 같았다.

041 우리가 계약한 아파트가 분양액의 몇 배가 될지는 모르겠지만 우리는 이미 엄청난 이익을 보고 있는 거예요.

042 기업을 하고 있다는 그것 하나로 남의 집에 파탄이 나는 현상을 바라보면서도 법대로 하라구?

043 기씨 문중에서 처녀 하나가 공녀에 뽑히어 인당수에 뛰어든 심청이처럼 눈물을 흘리며 연경으로 끌려왔다.

044 이곳에 오 년간이나 있으면서 연회에 참석한 것은 다섯 번이 못 되었다.

045 마침 계집종이 상을 들고 오자 기정은 눈을 번뜩이며 재빨리 손목을 나꿔챘다.

046 순서에 따라 위계에 맞춰 황제 앞에 나가 절을 올리며 봉축의 예를 갖췄다.

047 삼촌과 같이 왔습니다만 삼촌은 지금도 탑돌이에 정신을 놓고 있을 거예요.

048 작가 동맹대회 및 기타모임에서 공식적으로 채택되어 공산주의 문화예술의 기본원리로 되어 있다.

049 종자에 있어서 기본은 사상에 두어야 하고 소재와 주제의 요소들은 사상적 알맹이에 의하여 제약되며 거기에 복종된다.

050 예술적 가공에 있어 예술적 세부들과 형상들을 종자에 집중시키고 복종시킴으로써 작품의 대를 튼튼히 세워야 한다.

051 이 단체의 조직은 중앙위원회 위원장을 정점으로 약간 명의 부위원장을 두고 있으며 각 도에는 지부를 설치하고 있다.

052 이 작품은 특히 한국의 대학생들에게 짓밟히면 짓밟힐수록 굴하지 않고 더욱 더 싸워 나가는 투사적 기질과 영웅적 기백을 가질 것을 강조했다.

053 걷기와 같이 앞으로 발을 내딛기 위해 발을 들어 올리거나 윗몸 일으키기에서 몸통을 들어올릴 때 이들 근육이 동원된다.

054 무릎관절 근육이 이축관절 근육이기 때문에 엉덩이 관절 작용근과 무릎관절 작용의 명확한 구분은 애매하다.

055 가장 중요한 목 근육은 목을 굴곡시키는 흉쇄 유돌근과 목을 신전시키는 경부 비근과 두부 비근이다 .

056 이것은 당시의 개혁에 있어 어떠한 나라도 생산수단의 사회주의적 소유원칙에 대해 중요한 변화를 가져오는 개혁방안을 제시한 적이 없다는 점에서 분명하다.

057 급진적인 개혁이 시도되었던 유고와 폴란드에 있어서도 매우 제한된 범위내에서의 개인농이 부분적으로 허용되었을 뿐이다.

058 유아의 발음이 정확하지 않거나 잘못 사용하는 말들은 그때 그때 고쳐주어야 한다.

059 유아교육 기관에서 가르칠 수 없는 내용은 가정교육을 통하여 보충해 주어야 한다.

060 아무튼 동양이나 서양을 막론하고 옛부터 문학의 중추로서 시가 그 으뜸을 차지하면서 흘러 왔다.

061 지면 관계상 실제 작품의 인용은 가능한 한 생략했음을 밝혀 둔다.

062 결국 건축에 있어서의 고전주의도 계몽주의의 사상적 토대로부터 그 철학적 기반이 형성되었다고 할 수 있다.

063 그 효과는 재료 자체의 표현력에 걸맞게 커다란 원아치창을 이용하여 폐쇄성은 약한 반면 기념비적 성격이 뚜렷하여 리처드슨의 취향을 잘 나타내고 있다.

064 이렇게 발달된 형태언어에 관해서 두 가지 형태의 특성을 지적할 수 있다.

065 이 단계에서는 흔히 국왕과 성직자의 사이에 협력관계가 이루어져서 물리적 협박과 종교적 협박이 서로 보완한다.

066 많은 도시에서 사회민주주의자들은 매우 급속히 영향력을 획득해 가는 이 사회운동에 적극 개입할 필요를 느꼈다.

067 해외에 망명해 있던 멘셰비키들은 처음에 오히려 이 계획에 대해 반대했다.

068 러시아 교회음악에 갑작스런 화성음악의 도입은 자연스러운 변화가 아니었기 때문에 그 화성개념의 적용에 대해서는 많은 학자들이 회의적인 의견을 표한다.

069 이런 노래는 곧 러시아 전역으로 퍼지고 러시아 교회에서는 어디서나 들을 수 있게 된다.

070 이들의 활동영역은 퍽 넓어서 초원지방 끝까지 미쳤고 삼림지역의 외딴 구석의 마을에까지 미쳤던 것으로 보인다.

071 이곳에서 상연되는 연극은 처음에는 독일말로 이루어지다가 차츰 러시아말로 바뀌게 된다.

072 그리고 역 구내의 철로 위에 세워져 있는 객차 안의 포근함도 알고 있었다.

073 이튿날 아침 일찍이 오원은 그 아버지를 찾아가 그 앞에 엎드렸다.

074 두 번이나 뛰어 들었으나 번번이 동편에 칼끝과 말머리가 맞았을 뿐이다.

075 특히 장거리 여행을 하는 경우에는 거의 예외 없이 토요일 밤 비행기 편을 이용했다.

076 그 흑인이 시야에서 사라지고 나서야 윤종혁은 뉴욕에 범죄가 많다는 것이 생각나서 자신을 살펴보았다.

077 온 몸에 그늘이 덮여오는 것 같은 스산함을 느끼면서 어깨를 움츠리고 있는데 늙은 여자의 목쉰 소리가 바로 곁에서 들려왔다.

078 흑인답지 않게 눈이 비교적 작고 몸집이 중키쯤 되어보이는 그 사내는 코밑에 수염을 기르고 있었다.

079 그러니까 지금 그 동양인과 흑인 사내와 윤종혁은 모두 이어져 있는 세 개의 칸에 나눠타고 어두운 땅 속을 달리고 있는 셈이었다.

080 윤종혁도 거기서 내려야 했기 때문에 차에서 나와 출구를 향해 걷기 시작했다.

081 갑자기 웬 나이 타령이냐는 얼굴이었지만 이내 알 만하다는 얼굴로 그는 고개를 끄덕이고 있었다.

082 맨 위쪽에는 활의 탄력성을 높이기 위해 묶어 놓은 물소뿔들을 넣었고 그 아래칸에는 민어부레나 소심줄 화피 등속을 넣어두곤 하였다.

083 정임이 대문을 열자 연산댁과 아랫마을에서 뽕을 치는 율촌댁이 서 있었다.

084 오영미가 이 동네에 오면 그녀의 최후의 보루가 무너져 버리고 만다.

085 그런데 지금 와서 곰곰 생각해보면 저와 그 후배 남학생이 서로 사랑하고 있는 게 확실한 것 같아요.

086 집 근처의 어스름한 곳에서 그는 갑자기 저의 손을 잡고 키스를 했어요.

087 그는 열렬한 팝송팬이어서 어머님이 잠시 비울 때는 음악을 크게 틀어놓고 둘이서 어울려 서툴지만 디스코 풍의 춤을 추며 즐거워하기도 했지요.

088 나에게 쪽지를 보낸 그 남학생은 공부도 잘하고 또한 제가 평소에 그리던 이상형의 남학생이었으니 제 마음이 오죽했겠어요.

089 그는 제 친구의 남자친구보다 키가 훨씬 컸고 믿음직스러웠으며 머리 모양도 멋지다고 생각했고 무엇보다도 다정하기 짝이 없었기 때문이에요.

090 그는 저와 함께 벤취에 앉기가 바쁘게 제 어깨로 팔을 돌리더니 재빨리 제 턱을 끌어당겨 키스를 하려고 했어요.

091 저는 전혀 예기치 못한 그의 행동에 당황하여 몸을 빼내려고 했지만 그의 완력에 못 이겨 그만 입술을 빼앗기고 말았어요.

092 저는 참기 어려운 모욕스런 그 말에 홱 뿌리치고 일어나고 싶었지만 한편으로 묘한 기분에 몸이 자지러지는 느낌이었어요.

093 하지만 제 남자친구는 교환데이트 시일이 끝나서 이제 나와 만나야 하는데도 저를 본체 만체할 뿐 제 친구애와 계속해서 사귀는 것 같았어요.

094 넓은 마루에는 고목나무 뿌리로 만든 탁자와 웅접세트가 깨끗하게 정돈되어 있었다.

095 그런데 그녀는 들어오지 않고 우뚝 멈추어 서며 깜짝 놀라는 것이었다.

096 그렇게 서 있던 다혜는 텐트 안에서 누군가 갑자기 뛰어나오며 왜 지켜보느냐고 소리칠 것만 같은 생각에 겁이 났다.

097 이 때를 놓치지 않겠다는 듯이 바짝 마르고 광대뼈가 튀어나온 이방이 곁달았다.

098 현감 최동진은 약간 자리를 고쳐 앉으며 위엄스럽게 부하 관리들과 향중의 유지들을 휘둘러 본 뒤에 큰 소리로 외친다.

099 왼쪽 여자는 마흔 쯤 되어 보이고 오른쪽 여자는 몸매까지 앳된 것이 열 대여섯 살의 소녀이다.

100 나졸의 방망이가 아들의 어깨를 잡은 어머니의 팔을 힘껏 내려쳐 떼어 낸다.

101 여삼의 두 눈에서는 소리없이 굵은 눈물방울이 떨어져 족쇄에 걸린 무쇳덩이 위를 적신다.

102 좌중은 조금 전에 본 임여삼의 힘과 늙은 여자의 야수성에 대해서 웃기도 하고 혀를 차며 욕을 하기도 했다.

103 새로 등적된 노예를 매매할 때는 그 매매금 모두를 국고에 넣기로 돼있는 것이오.

104 안팎 궂은 일을 다 할 수 있으니 일석이조의 사용가치가 있지 않소?

105 창대수염은 속이 타는 듯 낮은 소리로 부르짖으며 임여삼의 뺨을 갈겼다.

106 앰뷸런스 안에 있는 시체는 밤새도록 퍼마신 위스키의 위력으로 주체하지 못한 조 중위를 비웃는 듯했다.

107 내리쬐는 폭양 속에 시체는 금새 풀어져 악취를 내고 있기 때문이다.

108 당시 자유당과 내각과 군의 일부 요직에도 모두 이기붕의 입김이 미치고 있었다.

3.세트 B(108 문장)

001 조만호 중위는 옆에 꾸벅대고 졸고 있는 운전병을 한 팔뒤꿈치로 툭 쳐댔다.

002 어디가 됐든 금테를 둘러 표시한 것도 아닌데 특별시 여자들은 왜 표가 날까.

003 닷새만에 투항하듯 낙향할 수밖에 없었던 그에게 특별시가 깊이 새겨 준 예감의 하나는 사실은 그곳이 어쩌면 전쟁터일는지도 모른다는 것이었다.

004 이제까지 잠들어 있던 바바리코트의 남자가 잠을 깨고 불쑥 던진 첫마디였다.

005 불현듯 식장 쪽으로 기사와 함께 먼저 올라간 여자의 모습이 궁금해졌다.

006 그 애의 입술이 마치 잘 썩힌 퇴비를 한줌 쥐었을 때처럼 부드럽고 따뜻이 그의 입술에 짚였던 것이었다.

007 뜀박질하여 휴게소를 한바퀴 휘젓고 다녔으나 바바리코트의 모습은 흔적조차 찾을 수 없었다.

008 택시는 한꺼번에 여러 대의 트럭을 추월해서 곧장 앞으로 빠져 나왔다.

009 게다가 바람막이조차 없었으므로 휑 열린 고속도로에서 불어닥치는 바람 때문에 빗물이 채 마르지 않은 그의 몸은 곧 한기에 시달려야 했다.

010 택시는 빗속에서도 팔십 킬로 이상의 속력을 냈으니까 십 분쯤은 충분히 따라잡을 수 있었을 것이다.

011 톨게이트로 완전히 진입해 들어왔을 때에야 남자가 고개를 들었으며 똑바로 앞을 바로보았다.

012 트럭의 속력이 그 순간 뚝 떨어졌고 때맞추어 유조차가 휭 하고 지나가 버렸다.

013 총열 끗에 매달린 태극기의 선명한 빛깔들이 그의 얼굴을 갑자기 환하게 밝혀 준다.

014 노상 아침밥 숟갈을 놓기가 바쁘게 동생 몽치와 함께 대문간을 기웃거리곤 했던 삼방우다.

015 전경대원들은 낮이면 송전학교 운동장을 빌려 하루 종일 군사 훈련을 받곤 했다.

016 그러나 월전에 담임선생으로부터 그 얘기를 듣고 나서부터는 완전히 달라지고 말았다.

017 관섭은 오줌을 다 누지도 못하고 옆에 있는 바위 위로 올라가 엉덩이를 깐다.

018 신숫점을 보러 아침나절에 시내촌으로 넘어간 며느리가 여태 왜 안 돌아오는지 추곡댁은 조바심이 난다.

019 두칠의 마음속을 밝혀 말하면 논을 소진사에게 빼앗기듯이 어머니를 빼앗길 수는 없다고 생각했다.

020 하루 이틀이 지나면 꽉 찰 것 같은 이지러진 달이 허공에 걸려 뿌연 빛을 쏟아붓고 있었다.

021 원채에서 꽤 떨어져 사랑채가 있었으나 원채를 마주 보게끔 좌향을 틀었다.

022 두칠은 꽁무니에다 찔렀던 낫을 쨉싸게 뽑아 들고 온몸에다 힘을 주면서 소곤거리며 말했다.

023 아래턱 수염이 몽탕 빠져 나오는 바람에 살점이 젖혀져 군데군데 피가 비주

룩 돋았다.

024 귀에서 웡 소리가 일고 눈을 부릅뜨고 있었으나 전혀 앞뒷일이 똑똑히 분별이 되지 않아 얼굴이 일그러져 몸을 떨고 있었다.

025 흩껍데기에 말아싼 소진사의 머리를 쥔 채 두칠은 숲속에 들어가자 그 자리에 자빠져 누웠다.

026 함안고을에서 북으로 사십 리 길을 가야 용화산이 있고 그것도 올라가야 초막이 있었다는 어릴 때 기억이 아슴푸레 떠올랐다.

027 두칠은 두 주먹을 꽉 잡고 속으로 역정을 내어 스스로 물었다.

028 흰 머리칼이 죽은 피에 웅겨 붙고 눈을 반쯤 뜨고 있었다.

029 철학은 모진 비바람과 폭풍우 속에서도 예쁜 꽃을 피워내는 들꽃이어야 한다.

030 그러나 그 자극제 뒤에 보이지 않게 자리해 있는 것은 맹목적인 물량 추구와 이윤 추구이다.

031 청량 음료의 쾌감은 이러한 쾌적한 삶에 한몫 낀다는 자기 높임의 느낌에 연결된다.

032 바깥 문을 닫자 회색빛의 을씨년스러운 서구의 세계는 무대 뒤로 사라진다.

033 굉장히 큰 호랑이의 가죽이 벽난로 앞에 한껏 뻗어 누워 있다.

034 왜냐하면 그의 얼굴은 평온하고 신비에 싸인 표정이며 두 눈은 코끝을 향해 내리깔고 있기 때문이다.

035 부속 기술이 숙달화가 되면 다른 상위의 지적 활동에 집중할 수 있고 방해를 덜 받게 된다.

036 앞면에 배울 것이 적혀있고 뒷면에 정답이 있는 카아드가 학습자에게 하나씩 하나씩 제시된다.

037 컴퓨터의 이용은 매우 복잡한 방법을 학습자의 관점에서 쉽게 사용할 수 있도록 만들 수 있다.

038 없어진 음표라고 불리우는 교습은 음악 보표에 네 개의 음표와 함께 없어진 음표의 공란과 함께 보여준다.

039 가장 좋은 상황에서도 연습 및 훈련 학습은 학생들에게 매우 흥미로운 학습 활동은 아니다.

040 컴퓨터 보조 연습 및 훈련 프로그램은 섬세한 수업전략의 특징을 잘 적용한다면 플래쉬 카드나 연습지보다 훨씬 효과적일 수 있다.

041 웨이터는 마치 흐린 날씨가 제 탓이기라도 한 듯이 송구스러운 표정이었다.

042 커피 두 잔만 달랑 들고 온 웨이터가 빙싯거리며 웃는 민희에게 얼른 말씨를 놓았다.

043 엘리베이터 안으로 들어서는 두 여자의 뒤통수에다 대고 이렇게 말끝을 맺지 못한 채 사내는 돌아섰다.

044 연미색 커튼은 솜씨있게 주름잡혀 있어 따스한 방안의 온기를 감싸고 있는 듯 하였다.

045 워낙 몸매가 좋고 감각이 있어 무엇이든 마음먹고 대들면 남보다 한 박자쯤 빠른 여자였다.

046 한국 물정 모르는 두 여자를 어떻게 구워 삶나 궁리하고 있겠지.

047 이층과 연결된 계단은 샹들리에 불빛이 휘황했고 붉은 포도주 빛깔의 카펫이 깔려 있었다.

048 복도 끝 쪽엔 분위기 좋아 보이는 바가 숨어 있듯이 자리잡고 있었다.

049 장식 등에 뒷받침된 건축 공간자체의 아름다움이 설령 미완성일망정 크게 돋보이고 있다.

050 여자는 스커트 속으로 팬티가 보일 정도로 널브러졌고 사내의 코피가 여자의 흰 옷에 튀어 붉은 반점을 남겼다.

051 남북교류 얘기가 요란하게 번질수록 그 친척은 전영달의 토끼같은 가족에게 시한폭탄 격이 될 것이 당분간은 자명했다.

052 부안의 김낙철 접주와 고부 전봉준 접주는 오지 않고 그 대리가 왔다.

053 박형사는 보일듯 보일듯하는 예감의 불빛을 따라 조금씩 암흑 속을 헤쳐나간다.

054 그 다음 만경대에서부터 쓰고온 토끼털로 만든 귀덮개를 벗으시어 동생의 양손에 끼워 주시었다.

055 그 초가지붕밑에는 너무도 일찍 두고 오신 유년시절과 무지개를 잡으시려던 꿈이 파묻혀있다.

056 그의 옆에는 쌀함박과 바가지 짝을 올려놓은 시꺼먼 궤짝 하나가 놓여 있다.

057 그러다가도 이따금씩 무릎위에 얼굴을 틀어박고 새우잠을 자는 어린 소녀애의 포대기를 여미어준다.

058 무성이는 자기때문에 외삼촌이 자꾸 왼심을 쓰는 것이 마음에 걸렸던지 다른 데로 가지 않고 모닥불곁에 쭈그리고 앉았다.

059 발뒤꿈치의 색보다 더 진한 자주빛을 띤 크고 뭉투룩한 발가락이 해진 양말 앞코숭이로 비쭉 내밀렸다.

060 마부와 철주의 어깨사이에서 털목도리에 휘감긴 텁석부리 얼굴이 미간을 찡그리며 천천히 일어섰다.

061 지금의 아버님 모습은 평양감옥을 나서시던 그때의 모습보다 한결 더 파리하고 수척해지신 것 같았다.

062 발통이 땅에 닿을 때마다 편지밑에서는 보랏빛과 연분홍빛으로 눈부시게 채색된 눈가루가 뽀얗게 피어올랐다.

063 보위색바지에 검은 솜동복을 받쳐입고 수달피 털모자를 꾹 눌러쓴 날씬한 몸매의 기수가 말안장에 앉아있었다.

064 즉 객관적 현실 중에 본래 미가 존재하고 있으며 미는 바로 객관적 사물 자체에 있다.

065 철학상에서 유심주의와 유물주의 사상의 근본 대립은 종종 미학상의 유물주의와 유심주의 관점의 원칙의 분기를 야기시켰다.

066 당연히 마르크스주의 이전의 유물주의도 다소 형이상학적 국한성을 띠고 있으며 사회 역사관에 있어서는 여전히 기본적으로 유심주의적이다.

067 이렇게 미학사에 있어서의 유물주의 학파 역시 심각한 결함과 부족을 면하기 어려웠다.

068 권력의 전략에 있어서 정신의학과 형법학은 동일한 지평 위에 세워져 연관성을 맺는다.

069 감시와 처벌은 고전주의 시대로부터 끊임없이 쉬지 않고 전개된 권력과 불법적 행위와의 투쟁과 충돌의 역사를 서술한 책이다.

070 왕이 군림하던 과거의 사회와 훈련과 규율로 특징되는 현대사회는 여러가지 점에서 대비된다.

071 이는 문학의 합리성 사실성의 본질론적 그 원리를 갈파한 신재효의 혜안일 것이다.

072 끝으로 그의 여섯 마당 특징을 간단히 요약하여 보면 다음과 같다.

073 현관을 나서는 연규의 등뒤로 강씨의 염려 섞인 말이 날아와 귓바퀴에 내려앉았다.

074 그런데 이 사정을 나중에야 알게 된 유라가 못내 아쉬워하며 연규에게 타박을 했다.

075 호텔 방으로 들어올 때와는 달리 연규의 가슴 깊은 곳에서는 욕망이 꿈틀거리기 시작했다.

076 연규는 옥빛 대리석으로 빚어낸 유라의 나신을 금방이라도 부숴뜨릴 듯한 격한 몸짓으로 탐욕스럽게 끌어안았다.

077 유라는 자신의 내밀한 수밀도의 꽃샘 속에서 터지는 뜨거운 용암의 분출을 느꼈다.

078 그 부분에 활동전위가 전도해 오면 탈분극과 겹쳐서 활동전위의 높이가 작아진다.

079 이 과산화효소가 뇌의 어떤 부위에 주사되면 그곳에 있는 축색말단에서 흡수

되어 역행성 수송에 의해서 그 모체가 되는 뉴론세포체를 검색하게 되는 방법이다.

080 그리하여 군국주의자들의 대륙정책은 착착 진행되고 드디어는 중국과 러시아와 싸워 이겨 국위를 크게 떨쳤다.

081 점차로 세계의 여러 지역이 여행자들과 선교사들의 탐험에 의하여 알려지면서 인류 사회의 다양성이 밝혀지기 시작하였다.

082 그리고 그런 행위 유형은 고고학자들이 찾아내고 기록하는 특수한 인공 유물의 유형과 지표면에서의 여러 징후를 통해 고고학적인 기록으로 나타난다.

083 첼로는 바이올린의 확대물이라 그 주법이 바이올린과 흡사하여 성과는 매우 좋았다.

084 여기서 예술적 경험이 인식에 중요하고 또 기초라고 하는 점에 유의해야 한다.

085 꽃도 꼭 튜울립이 아니고 그 어린이가 꽃이라고 지각한 일반적인 어떤 꽃이다.

086 그 회사들은 간부직원의 경영기술을 강화하기 위하여 집단역동 원리를 이용한 세미나와 휴양소를 기본으로 사용하고 있다.

087 각 구성원들은 집단의 모든 구성원 앞에서 그들의 특이한 대인관계적 문제를 행동화하기 시작하고 동료 구성원들의 집단적인 세밀한 탐색하에서 그들의 왜곡들을 영속시킨다.

088 찹쌀을 절구에 넣어 떡메로 쿵쿵 찧을 때 생기는 압력 에너지가 몸을 따뜻하게 해주는 것이다.

089 동양이 서양에 매혹되고 서양이 동양에 매혹되는 것은 양자의 성격이 완전히 다르기 때문이다.

090 이때 팩을 하면 모공이 완전히 열린 상태이므로 흡수가 잘 된다.

091 또 율무는 특히 피부에 윤기와 탄력을 주며 골다공증을 예방한다고 알려져 있다.

092 그러나 식물성 섬유를 자칫 잘못 섭취하면 소화 흡수가 어려운 까닭에 설사하기 쉽다.

093 싼값에 손쉽게 구해서 쓸 수 있는 수세미 즙은 겨울철 건조하기 쉽고 거칠어지기 쉬운 피부에 효과적인 영양제가 된다.

094 수세미 즙으로 맨살을 가꾸면 천연의 보습 효과가 피부를 촉촉히 해주므로 피부 노화를 예방할 수 있다.

095 예부터 사람들은 더러워진 피부를 깨끗이 하고 젊음을 유지하기 위해 온천욕과 냉수욕을 해왔다.

096 목이나 어깨에 거미줄 모양의 반점이 생기고 손바닥 혈관이 확장되어 뻘겋게 보이기도 한다.

097 피로가 쌓이지 않게 하고 싱싱한 과일과 야채를 많이 섭취하여 예방하는 것이 최선이다.

098 오부의 경락은 음으로 대체로 목 부근에서 끝나며 육부는 양으로 얼굴까지 뻗쳐 있다.

099 무의 마른 잎을 잘게 썰어 욕조에 넣으면 부인병과 대하에도 좋다.

100 또 서울시청 뒷골목에서 전에 미대사관이었던 미문화원 뒷골목이 본래는 하천이었는데 무교동 쪽에서 시청쪽으로 가는 길에 뫼청다리가 있었습니다.

101 옛 서울의 아름답던 기와집들은 모두 불에 타 없어지고 초가집으로 바뀔 수밖에 없었습니다.

102 프랑스의 침략 만행에 격분한 조선 민중은 여러 곳에서 의용군을 조직하여 맞서 싸웠다.

103 수출품은 민중의 일상 생활에 꼭 필요한 쌀 콩 쇠가죽 등이었다.

104 이제 조선은 외세 침략자들의 세력 확장을 위한 싸움터가 되고 말았다.

105 열 석달 만에 월급으로 받은 쌀이 분량도 제대로 채워지지 않았고 겨까지 섞여 있었다.

106 이때 환자는 퇴원 후의 병에 대한 주의사항 또는 약을 받고 입 퇴원과에 입원비를 지불하고 퇴원하게 된다.

107 환자는 오히려 이 시기에 심심해서 얘기 상대가 필요한데도 문병객은 뜸해진다.

108 환자의 병 상태나 연령 등을 고려하여 선물을 선택하여야만 환자에게 도움이 된다.

4.세트 C(108 문장)

001 감귤류처럼 비교적 산도가 높은 과일은 오히려 위장을 자극할 염려가 있어 위궤양 환자에게는 금물이다.

002 다만 위액 분비가 대단히 나쁜 무산증 환자에게는 귤과 같이 천연적으로 산도가 높은 과일이 치료와 회복에 도움이 될 수도 있다.

003 예를 들면 모병원 산부인과에서 태아를 출산하려다 사고로 인하여 태아가 사망하게 되었다면 가족과 병원측과의 싸움이 생기게 된다.

004 할 일도 없고 심심해서 자기 집 옆 공터를 주인의 양해 없이 밭을 일구고 호박을 심어 정성껏 가꾸었다.

005 남성우위 남성지배의 사회는 불평등사회 이전의 장구한 인류사의 전 과정에 비추어볼 때 불과 수천 년에 불과한 짧은 역사를 가졌을 뿐입니다.

006 이런 경로를 통해 확립된 부권제와 또 남성의 손에 모인 비교적 커다란 부의 뒷받침을 받아 남성우위 남성지배의 일부 일처제가 시작되었습니다.

007 남성에 소유된 여성 쪽 역시 유감스럽게도 개인적인 성애를 마음속에 품고 있을 수 없었겠지요.

008 대부분의 국가의 헌법이나 가족법 등을 보면 다음과 같은 내용을 포함하고 있습니다.

009 즉 여러 가지 사회생활상의 규칙 중에서 가장 구속력이 적고 자연스러운 것이 풍습이지요.

010 캠퍼스에서 쌓아왔던 삶의 신념들을 펼쳐 나갈 수 있는 직장은 과연 어떤 곳일까요?

011 고등학교 때부터 사귄 여자 친구를 대구에 두고 올라온 학생이 한명 있었다.

012 연세대학교와 이화여자대학교 사이에 자리잡은 우리 집 근처에는 대학이 개강하자 양쪽 학교에서 학생들이 차를 몰고 와 이 좁은 길가에 주차한다.

013 차가 있고 돈이 있으면 자연 예쁜 여학생을 옆에 태우고 놀러 다니고 싶어지는 법이다.

014 의리가 없는 세상 약속도 지키기 어려운 세상에 어떻게 의리까지 지키라고 할 수 있느냐고 따질 사람도 있을 것이다.

015 현관에 들어서니 왼쪽에는 주욱 방들이 열지어 있고 오른쪽은 대청겸 복도였다.

016 요즘은 손이 떨려 원고도 쓰기 힘들고 기억력도 예전 같지 않다.

017 형편이 어려워 우리가 살던 적수부락 인근에 모신 아버지의 유해를 본촌인 북청군 하거서면 선산으로 이장하기 위해서였다.

018 그날 밤 말을 타고 왜놈을 뒤쫓는 독립군의 꿈을 꾸었던 듯싶다.

019 형사들이 책더미를 꾸린 후 나를 가자고 끌어내자 외할머니가 나를 끌어안고 그자들에게 욕설을 퍼부으며 덤비셨지만 무지막지한 그들의 완력을 막아낼 수는 없었다.

020 이렇게 되니 유치장이 턱도 없이 모자라 이들의 발목에 족쇄를 채워 무작정 사무실에까지 집어넣었다.

021 그립고 그립던 선배들과 함께 있게 되니 더할 나위 없이 좋았는데 특히 정운길 선배와 같이 있게 되어 더욱 기뻤다.

022 귀리밭의 어머니 사체검안 후 우리들 다섯 명 중 한 명은 기소되고 네 명은 기소유예로 풀려났는데 나는 나가는 쪽에 끼게 되었다.

023 처음에는 물가부터 얼음이 얼기 시작해서 차차 얼음판이 두꺼워지면서 중앙부 쪽으로 확산된다.

024 이 장면은 밤새워 얼음낚시를 하는 사람들이나 운 좋게 볼 수 있는 진기한 광경이다.

025 그러나 이 방법은 여러 가지 후유증이 예상돼 큰 진전은 없는 듯하다.

026 그로부터 시작해서 이듬해 봄까지 미국의 태평양 연안에 자리잡고 있는 지방에는 전에 볼 수 없었던 큰 규모의 폭풍우가 여러 차례 몰아닥쳤다.

027 마치 사과 크기의 지구에 도너츠 모양의 바람 띠가 걸쳐 있어 이것이 끊임없이 돌고 있는 형상에 비할 수 있다.

028 실제적으로 이 병에 걸렸던 환자 가운데 한적한 교외로 갔을 때 병세가 훨씬 좋아진 사람들이 많다.

029 그래서 햇빛이 비치지 않기 때문에 몇 도 정도 기온이 낮아질 것으로 예측이 되기도 합니다.

030 쿠웨이트시는 사흘중에 이틀은 바로 앞도 보이지 않는 캄캄한 밤과 같습니다.

031 또 이라크가 유정을 파괴하고 쏟은 기름이 이 지역의 토양을 다 덮어 버렸습니다.

032 또 시커먼 비가 내려서 쿠웨이트나 인근 지역의 물뿐 아니라 토양이 다 오염되어 버렸습니다.

033 왜냐하면 여기는 영양분이 흙과 함께 더 쉽게 유실될 수 있기 때문에 그렇습니다.

034 첫째는 흙이나 영양분이 자꾸 씻겨 가기 때문에 비료를 주어야 합니다.

035 그런데 잔디뿌리는 너무 얕기 때문에 그 얕은 뿌리 아래로는 영양소가 빗물에 다 씻겨 내려갑니다.

036 물이 왜 이렇게 더러운가 하면 해수욕장 인근의 하천이 오염되었기 때문입니다.

037 이와 같은 가정에서의 심리적 갈등이 밖으로 표출될 때 문제행동을 수반하게 되니 각별한 교육적 배려가 요청된다.

038 그리고 출납계에서는 보고받은 지출금액 중 부족분을 다시 소액현금 담당자에게 지급한다.

039 횡선수표는 수표의 표면에 두 줄의 평행선을 긋고 그 사이에 은행 또는 이와 동일한 의미가 있는 문자를 기재한다.

040 주당 평균노동시간의 감소는 불완전 취업의 증대와 실업 증가의 또 다른 표현형태일 수 있다.

041 독점자본은 생산영역뿐만 아니라 유통과 소비영역에까지도 침투하여 잉여의 환수와 가치의 증식에 골몰한다.

042 기존 사회운동의 침체와 시민운동 활성화의 배경은 무엇보다도 먼저 객관적 현실 자체의 변화에서 찾아진다.

043 말할 것도 없이 대중 매체의 발전과 유포는 이들 변화의 일부였다.

044 그것을 대체하는 것이 대중 매체에 의한 의사 소통 즉 매스컴이다.

045 오늘날 대중 매체는 크게 인쇄 매체와 전자 매체로 나눌 수 있다.

046 대중 매체는 사회에 의해 영향을 받으면서 또 사회에 영향을 미친다.

047 국가는 어떤 내용이 대중 매체에 나올 수 있고 어떤 내용이 나올 수 없는지를 결정한다.

048 과거에는 매체가 특정 내용을 공표하지 못하도록 개입하는 적극적인 검열이 주였다.

049 이 때문에 본래 오락성이 강한 매체의 내용은 말할 것도 없고 보도 매체의 내용조차도 오락화하는 경향이 있다.

050 먼저 정치부는 중앙청 외무부 통일원 등의 정부 행정부서를 돌아본 행정팀과 정당팀이 하나 둘씩 속속 들어와서 뉴스로 채택된 기사의 편집을 시작한다.

051 네트워크 뉴스에서는 해외특파원 이외에 미국내 각 지국의 선임기자와 전문분야를 취재영역으로 갖고 있는 기자들을 특파원으로 부른다.

052 친구와의 대화에서 상대방의 말뜻을 헤아려서 적절한 대답을 할 수 있는 것

도 왼쪽 뇌의 활약 덕분이다.

053 어쨌든 오른쪽 뇌의 언어 능력은 미미하며 뇌에 비해 월등히 부족한 것이 사실이다.

054 특히 예술적인 뇌인 오른쪽 뇌와 정교함을 좋아하는 왼쪽 뇌로부터 비롯되지 않았을까?

055 가사는 전혀 슬프지 않지만 단조의 애조를 띤 분위기의 악곡 때문에 좋아했었다.

056 그러므로 한 가정을 이루는 구성원은 항상 자신의 위치를 바로 세워 원만한 가정을 꾸며나가야 할 공통적인 사명을 띄고 있는 것이다.

057 사회는 바야흐로 물질적인 풍요 때문에 과소비 풍조가 발생되어 바른 윤리관을 크게 위협하고 있다.

058 죄를 범하고도 인정하지 않는 잘못과 용서를 비는데 받아들이지 않는 잘못도 어리석음을 범하는 것이다.

059 와이셔츠를 세탁기에 넣을 때에는 가급적 그물망에 넣어서 옷감이 상하지 않도록 한다.

060 흰 실크옷은 세탁하기 전에 잠시 우유에 담가두면 변색을 방지할 수 있다.

061 스타킹은 매우 얇아 자칫 잘못하면 코가 빠져 줄이 가 그만 못 신게 되고 만다.

062 넥타이는 반드시 도마같이 평평한 곳에 펴놓고 솔질을 해서 세탁을 해야 한다.

063 세탁할 때 쉽게 빨 수 있고 입었을 때 때가 덜 타게 하는 방법이 있다.

064 우선 빨 때 샴푸를 손에 묻혀 발라 두었다가 세탁을 하면 찌든 때가 깨끗이 빠진다.

065 목욕하러 들어갈 때나 세수할 때 빨아서 목욕탕의 유리창이나 욕조의 타일면에 반듯하게 펴서 붙여 두면 마르고 난 뒤에 다릴 필요가 없다.

066 급할 때는 선풍기 바람을 쐬면서 중기 다림질을 하는 것이 효과가 있다.

067 그런 뒤 물감을 처음부터 큰 그릇에 넣어 탄 다음에 큰 그릇에 전부 옮겨야 한다.

068 조제는 옷이나 천이 적당히 염색되었을 때 일단 천이나 옷을 꺼내놓고 넣어야 하며 그 다음에 천이나 옷을 넣어야 한다.

069 옷을 종이에 싸거나 종이봉지에 넣을 때 하얀 양지에 싸는 것은 좋지 않다.

070 여름철에는 특히 곰팡이류가 가죽 제품에 피기 쉬우므로 건조제와 좀약을 넣어둔다.

071 이것은 스웨터나 얇은 천의 블라우스 등에 보통 옷에 단추를 달 듯이 달기 때문에 생기는 것이다.

072 꽃은 나무의 여기저기에 지극히 감미로운 선율의 음악을 펼쳐 두고 있는 듯 보였다.

073 첫번째 시집에서 떨어진 빛의 가루들이 두 번째 시집의 책갈피를 잠깐씩 반짝이게 하는 격이었다.

074 선전이나 계몽도 삶의 어떤 표피에 부딪쳐 올 때는 의외의 정직한 힘을 발휘한다.

075 법정 안은 학생들의 우렁찬 구호와 합창의 열기로 뜨겁게 달아올라 있었다.

076 위의 예에서 본 바와 같이 원시인 사회에서는 남녀의 뚜렷한 분업상태는 거의 확실해져가고 있다.

077 날이 가물면 물이 없어 애태우다 장마가 지면 홍수로 물난리를 겪어야 했다.

078 뽕잎이 활짝 핀 산뽕나무를 만나면 어머님의 옛모습이 떠올라 코허리가 아릿해진다.

079 이제 붙잡힐 염려가 없다는 안도감에 마음이 놓이자 감자밥 한 그릇으로 태산준령을 넘어 밤새도록 걸어온 배가 대뜸 휑하니 고파오기 시작했다.

080 주소를 들고 친구가 취직해 있는 시계포 찾기는 별로 어렵지 않았다.

081 고원에 그대로 눌러 있게 해달라는 내 호소에 아버님은 객주집 어두운 방에 나와 함께 누우셔서 사정하듯 말씀하셨다.

082 앞에서도 얘기했지만 우리 마을은 눈이 오면 넉가래로 우물길을 쳐놓아야 할 만큼 적설량이 많은 곳이다.

083 길을 뜨기 전 세 공모자가 노자를 맞춰보니 약속이나 한 듯 셋이 다 삽십 몇 전씩이엇다.

084 시서화에 대한 화가의 부계쪽 취향을 간접적으로 증언해 줄 물증은 지금 하나 남아 있다.

085 화가의 부모는 이들 형제의 교육을 뒷바라지하기 위해 함께 서울로 이사왔다.

086 학생 미전에는 경기여자 고등 보통학교에 재학 중이던 장차의 화가 부인도 수예를 출품했다.

087 이 제도는 양털을 해외에 팔아먹기 위해 농민들을 농토에서 쫓아내고 그곳에 울타리를 쳐서 양을 놓아 먹인 제도이다.

088 화폐는 외국과의 무역 통상을 했을 때의 결재나 청산 등에 사용하는 세계화폐의 기능을 한다.

089 이런 상업망을 통해서 상품유통이 이루어지고 있기 때문에 자본주의에서는 유통비가 있게 된다.

090 이때에 직접 빌리기는 어렵고 하여 은행이 그 중개역할을 맡아 수행한다.

091 그렇다면 이 평균이윤을 제외한 나머지 부분 즉 초과분은 어떻게 생길까?

092 공업화 사회 초기의 공급자 주도형 시장이나 공업화 사회 후기의 소비자 그룹 주도형 시장 구조는 이미 무너졌다.

093 수산 사업부와 패션 사업부가 다 같이 전산 센터의 호스트 컴퓨터 내에 있는 데이터 베이스를 사용하고 있었다.

094 개인용 컴퓨터들을 근거리 통신망으로 네트워크화하면 소규모 회사도 충분히 정보화할 수 있다.

095 또한 데이터 베이스의 구축과 관리를 간편하게 할 수 있는 관계형 데이터 베이스 관리 시스템이 개인용 컴퓨터에도 탑재되고 있다.

096 개인 차원을 넘어선 조직 차원의 데이터 베이스나 지역 차원의 데이터 베이스를 서버 시스템에서 지원해 주고 있다.

097 시스템에 장애가 발생하였을 때 중앙집중식 시스템에서는 전 시스템이 눈앞에 보이기 때문에 쉽게 그 사태를 파악할 수가 있다.

098 시스템의 변신 능력을 설계 및 구성시에 확보하거나 시스템 관리상에서 확보해야 한다.

099 잘 생기고 공부도 잘한다면 하드웨어와 소프트웨어가 둘 다 좋은 것이다.

100 컴퓨터를 처음 켤 때는 디스크 드라이브에 반드시 도스 프로그램 있는 디스켓이 꽂혀 있어야 한다.

101 이렇게 컴퓨터에 반 미치도록 열심인 사람을 컴퓨터 매니어 또는 컴퓨터 해커라고 부른다.

102 개인용 컴퓨터도 값비싼 컬러 모니터를 옵션이라면 이해가 가지만 단색 모니터는 당연히 기본 품목이어야 한다.

103 컴퓨터 내부에 꽂혀진 그래픽 카드와 특성이 맞는 모니터를 골라야 한다.

104 박부장 역시 개인컴의 하드웨어부터 프로그램까지 통달해 있는 컴도사인데 어떻게 해서 아들을 낳을 수 있었을까?

105 이와 같이 추론의 토대가 되는 사실이나 규칙의 집적을 지식베이스라고 부른다.

106 그녀의 가슴에는 아직도 질병으로 노랗게 야윈 아들의 얼굴이 선명하게 새겨져 있었다.

107 죽기 직전 자신의 품에 꼭 매달려 있던 아들의 체온이 지금도 느껴지는 듯했다.

108 결국 컴퓨터는 이 현상을 단순한 우연의 일치로 결론지을 수밖에 없었다.

5.세트 D(108 문장)

001 대학 시절에는 상당한 실력을 갖춘 스쿼시 선수이기도 했으며 지금도 일주일에 세번정도 운동을 하였다.

002 한쪽 벽에는 대영 백과사전이 꽂혀 있었고 반대편 벽에는 원소 기호가 적힌 도표가 붙어 있었다.

003 몰래 넘겨다본 책 윗부분에는 예상치도 않은 공식들이 잔뜩 적혀 있었다.

004 그래도 열심히 노래를 하며 웨이터 아저씨들과 주방 아저씨들이 들락거리며 분주한 틈틈이 휘파람도 불며 박수도 쳐주곤 했다.

005 얘기 분위기도 심상찮아서 인사만 하고 돌아나오는데 그 중에서 제일 뚱뚱하고 얼굴이 거무튀튀한 아줌마가 나를 불러세우더니 서슬이 퍼렇게 다그치는 것이었다.

006 속수무책으로 앉아 있는 엄마와 넋이 나간 내 앞에서 빚쟁이 아줌마들은 한치 양보도 없이 어떻게 할거냐고 몰아세웠다.

007 그들 덕에 나는 비교적 쉽사리 첫 걸음을 뗄 수 있었다.

008 특히 기체의 경우는 분자 자체의 크기보다 분자와 분자 사이의 거리가 훨씬 크다.

009 환기가 안 되는 실내의 흡연과 탁한 공기도 에어컨 두통과 감기를 더욱 부채질한다.

010 그들의 얘기를 들으면서 물이 나빠지지 않도록 하는 방법은 왜 연구하지 않는가 궁금해졌다.

011 위에 열거한 유기 주석 화합물이 일으키는 환경 오염은 이러한 위험의 전형적인 예이다.

012 머지않아 개방과 함께 몰려올 외국의 우량기업들과의 한판 싸움에서 그 승부의 열쇠는 결국 고객이 쥐고 있다.

013 하긴 그 표어를 써 붙인 주인이 그렇게 실천하고 싶지 않을리야 없을 것이다.

014 하지만 실천하기 쉬운 목표라면 구태여 표어까지 써 붙일 필요도 없었을 것이다.

015 어쩌면 오장에 끼여 있는 뗏국이 아직도 덜빠진 탓이 아닌가도 싶었다.

016 그러나 가까이 가자 옷깃 밑의 자주색 턱받이를 발견할 수 있었다.

017 그리고 그 집 뒤뜰의 빨랫줄에는 언제나 더러운 옷가지들이 걸려 있었다.

018 그는 앞에 책을 기대어 세워놓고 책상 앞에 앉아서 중얼거리고 있었다.

019 침대 옆의 작은 탁자에는 안경과 기도서와 골짜기의 백합들을 꽂아놓은 꽃병이 놓여 있었다.

020 생각에 잠긴 듯한 그녀의 눈길이 얼마 전에 세탁 비용을 지불했음을 나타내는 깨끗한 탁자보에 머물러 있었다.

021 그때 위층으로부터 물이 세차게 쏟아지는 소리와 발로 쿵쾅거리는 소리가 들려왔다.

022 그 방법을 알려고 할진대 대략 그 요령을 배워야 할 것이다.

023 진영의 일원으로서 연합국측과 결정적으로 대립하게 되었고 특히 미국과의 관계를 한층 더 악화시키게 되었다.

024 은백색의 뱅어떼가 갈 곳을 잃고 우왕좌왕하는 모습이 보이면 구경꾼들까지 합세해서 그물을 끈다.

025 중국의 양자강에서 초어가 상류까지 올라가서 산란을 하는 것도 물에 뜨는 알을 낳기 때문이다.

026 봄에 세줄의 붉은 띠를 걸치고 상류 쪽으로 이동하는 물고기라면 황어밖에 없다.

027 그러나 왠지 갑자기 할아버지의 장례식 때 달았던 뻣뻣한 검정 리본이 생각났다.

028 좁은 틀 안의 현실적 범위 안에서 사고하며 아들이 목사가 되기를 소망했다.

029 양친은 웬 뚱뚱한 남자와 둥근 식탁에 앉아 포도주를 마시고 있었다.

030 또 이 한적한 길 위에서 빼놓을 수 없는 것은 예쁜 아가씨처럼 웃는 붉은색 덩굴장미와 수줍은 신부와도 같은 흰 덩굴장미 꽃송이다.

031 필요악이란 말도 이러한 모순된 삶의 구조에서 온 것이 아닐까하고 나는 매일 아침 새벽 산을 내려오면서 생각한다.

032 그리고 조심조심 표충사 경내를 빠져나와 왼쪽으로 다시 외원길을 타고 올라갔다.

033 윤처사가 웬만큼만 단속을 덜 했어도 그새 어떻게든 발길이 한 번쯤 미쳤을 곳이었다.

034 벼르고 별러온 수수께끼의 문을 바야흐로 눈앞에 하고 서게 된 성취감 때문이었다.

035 그는 재빨리 다시 불빛을 죽이고 쫓기듯 문 앞을 물러 내려왔다.

036 그 옆 벽면 한 귀퉁이에 작은 출입문이 하나 틀어박혀 있었다.

037 상대쪽은 그간 이편의 동태를 속속들이 모두 지켜보고 있었을 것임에 반하여 자신은 아직도 그쪽이 어떤 위인인지를 알 수 없었기 때문이었다.

038 도섭은 이내 자기 몫의 일을 알아차리고 불길이 한창인 그 쌍아궁이 앞으로 자리를 잡아 앉았다.

039 도섭은 애초 우봉 스님을 직접 만나뵙고 그의 도피행의 사연을 고해 올릴 계획이었다.

040 도섭은 의기양양 윤처사의 참견을 밀쳐버리고나서 소상한 뒷사연을 마저 엮어나가기 시작했다.

041 옷을 건네주고나서 계속 골방 문턱께서 도섭의 옷매무새를 살피고 있던 윤처사가 그 새 옷에 따른 특혜조치를 뒤늦게 알려왔다.

042 쇼크를 받은 나는 가게 안으로 들어가 주인인 듯한 중국인에게 물어 보았다.

043 무료 식당이나 무료 빵 배급소에 사람들이 길게 줄지어 늘어서 있곤 하였다.

044 혼자 외롭게 내려오는 사람도 이따금 눈에 띄었고 다정한 부부의 모습도 심

심찮게 보였다.

045 그럼에도 그가 억지를 쓰듯 큰 물통을 산 것은 젊음에 대한 항거였다.

046 언제나 그는 사위가 출근을 하고 외손자 외손녀가 한바탕 북새통 끝에 학교로 떠난 뒤 밥상 앞에 앉았다.

047 그러나 서로 자기들끼리 얘기꽃을 피우고 있었으므로 약수터의 위치를 묻기가 난처해 그는 그냥 길을 따라 올라갔다.

048 잠시 후 타박타박 걷는 여자의 발짝 소리가 바로 등뒤에서 들려 왔으므로 그는 흘낏 뒤돌아 보았다.

049 꽃들이 제각기 특별한 향기를 내뿜듯 그 냄새는 그녀 특유의 여자냄새였다.

050 골짜기의 앞이 트여 벌어진 것은 물론이요 바위 양쪽이 갈수록 점차로 높아지다가 뒤가 완전히 우긋하게 막힌 것이 갈데없는 삼태기 꼴이었다.

051 염노인은 마영감과 입씨름 끝에 윗도리 안주머니에서 뭔지 비닐봉투에 넣어 싼 것을 꺼내 들었다.

052 노트 한 페이지에 한사람씩 그 인적사항이 큰 글씨로 적혀져 있었다.

053 여태껏 한마디도 않고 그냥 흐믈흐믈 웃고만 있던 노파가 마영감의 말을 뚝 자르며 쏘아붙였다.

054 우리가 살고 있는 태양계 우리가 살고 있는 지구도 하나의 별입니다.

055 우리 지구에 에너지를 공급하여 따뜻하게 해 주는 태양도 언젠가는 차가운 별의 찌꺼기로 변해 마지막을 맞게 될 것입니다.

056 반짝거리는 밥공기와 녹슨 남비가 탁자 위에 미술품처럼 깔끔히 놓여 있다.

057 꿰뚫을 것처럼 빛나는 눈이 나의 등골에 박혀 세상을 보고 있었다.

058 때마침 눈을 떠보면 불빛 아래 책을 펼쳐놓은 당직보모의 작은 등이 보였다.

059 빛을 보면 곧 소란을 피워 아이들이 당황해서 달아날 때까지 대들었다.

060 달아낸 처마 위로 먼지투성이의 녹색 유리 기와가 얼굴을 희미하게 비치고

석양이 내리쬐면 겨울에 말라버린 잡초 너머로 번쩍번쩍 빛을 냈다.

061 고요한 밤에 울려퍼진 기계소리는 북경 사람들의 잠을 깨웠고 도시가 꿈에서 깨어났을 때는 완전히 나체로 변해 있었다.

062 게다가 엉덩이 위에서 춤추는 어깨 가방도 더러웠지만 세탁한 적은 물론 없다.

063 가끔씩 밤비에 젖어 냉기를 한껏 머금은 바람이 집들 사이를 통해 휘몰아쳐 왔다.

064 한 남자가 쓰러질 때 모습 그대로 팔다리를 쭉 뻗은 채 누워 있었다.

065 환자의 비웃음 섞인 얼굴에 얼핏 불신의 그림자가 스쳐 지나갔다고 느낀 것은 그의 착각이었을까.

066 그러나 현재 고등학교와 대학에 다니는 자식을 둔 나 자신도 돌이켜 보면 어머니와 같은 방법으로 자식들을 키워 온 것 같다.

067 영어에서 일상어는 고유어가 우세하고 학술어는 라틴어 계통의 어휘가 우세한 것과 매우 비슷한 구조를 보여준다.

068 한글 맞춤법은 표음문자인 한글로 국어를 기록할 때 적용되는 규칙을 말한다.

069 이 단어들을 찾아내어 표준어에 편입시키면 국어의 어휘는 더욱 풍부하고 아름답게 될 것이다.

070 사이시옷 표기는 우리 맞춤법 중에서 가장 어려운 것으로 통일안은 복합어에서 앞 단어에 받침이 없을 때에 한해서 사이시옷을 표기하도록 규정했었다.

071 이에 견주어 원시인들은 자연의 위대한 힘에 부딪쳤을 때 착잡한 반응을 보였다.

072 그리고 그 수체계는 완전히 가산적이어서 곱셈이나 나눗셈도 결국 덧셈으로 환원되었다.

073 갑자기 샛별에서 두 개의 눈부신 푸른 빛줄기가 땅으로 내리 뻗쳤던 것이다.

074 벌이 집 지을 무렵이 되자 떼를 지어 버드나무에 와서 평화롭게 한때를 살았다.

075 하지만 한꺼번에 데려갈 수가 없으니까 한 마리씩 데려다 줄 생각인데 어떨까?

076 이 지상에서 제가 사랑하는 모든 것이 왜 위협을 당해야 한단 말입니까?

077 늦어도 금요일 저녁에는 회신을 받아야 일요일에 할 일을 결정할 수 있기 때문입니다.

078 왜냐하면 음모와 반음모가 뒤얽힌 매우 복잡한 이유 때문에 곧 해임되었기 때문입니다.

079 왜냐하면 구름들은 밑으로부터 올라오는 열을 차단하여 가두고 또한 그 상부 표면에 내려 쪼이는 햇빛들은 반사시키기 때문이다.

080 그러나 일정 지역이 지구상의 어느 위치에 있는가에 따라 에너지의 흡수와 방출량 간의 균형이 서로 다르다.

081 그같은 생산의 효용 덕분에 시인은 산채에서 군사나 막빈에 못지않게 귀한 대접을 받았다.

082 이윽고는 언제까지고 끝날 것 같지 않던 겨울도 가고 봄이 왔다.

083 실제에 있어서도 그는 헤어진 지 몇해 안 돼서부터 취옹을 만나고 싶어했다.

084 이미 그 며칠 여러 가지로 놀라운 경험을 한 터라 그 웅얼거림에도 무언가 예사롭지 않은 뜻이 있을 것 같았다.

085 그들은 서푼어치 과학적 지식인이고 싶었고 그 유리한 대열에 계속해서 껴안고 싶었던 것이다.

086 엄격한 중앙집권의 알맹이를 겉으로 부드럽게 포장하기 위해 민주를 갖다 붙인 것이지요.

087 국회에서 크게 문제가 돼 공비토벌 촉구 결의안이 통과되고 이승만 대통령이 직접 내게 공비토벌을 지시했다.

088 사령관의 성을 부대 이름에 넣는 것은 전례없던 일로서 개인적 영광에 앞서 책임감이 어깨를 무겁게 눌렀다.

089 개전 이래 촌각의 여유도 없이 전투에 전투를 거듭해 온 그때의 내 눈에는 적어도 그렇게 비쳤다.

090 포위망이 좁혀지면 그 안에 보통 대여섯 마리의 토끼들이 갇혀 몽둥이 세례를 받는다.

091 또한 주로 토벌을 담당했던 각 지역의 경찰대와 빨치산들 간에는 뿌리 깊은 증오가 자라 있었다.

092 따라서 일단 빨치산과 함께 있다 붙잡힌 사람들은 모두 수용소를 거친 다음에야 풀려날 수 있었다.

093 처음에는 제대로 적응이 안 돼 철모도 안 쓴 맨머리를 비행기 천장에 부딪치면서 마구 구토를 해댔으나 차츰 요령도 생기고 익숙해졌다.

094 자식의 목숨을 끊어 달라는 부탁은 미움보다는 어차피 병사할 자식을 자기 손으로 묻기라도 하겠다는 한서린 부성애에서 비롯됐을 것이다.

095 그 곳에 누워 예쁜 라일락꽃을 바라다보는 것은 매우 즐거운 일이었습니다.

096 융단을 깔아 놓은 듯 부드럽게 느껴지는 구름 위를 날며 연이와 비둘기는 천궁과 옥황 상제님에 대하여 이야기했습니다.

097 뾰족하게 치솟은 높다란 망루는 사파이어나 루비 같은 보석으로 장식되어 있었습니다.

098 금빛 날개를 펼쳐 번개같이 빠르게 하늘을 날아다니고 무예 또한 비범하였습니다.

099 그윽한 두 눈에 온 몸의 중심을 싣고 있는 듯 했다.

100 어쩌면 이렇게도 쉽게 마음을 내놓으라고 할 수 있을까 원망하고도 싶었다.

101 초췌한 모습으로 사흘만에 나타난 아들을 붙잡고 어머니는 한숨부터 쉬었다.

102 일본의 주요 일간지에 매일같이 실리는 똑같은 내용의 기사 아닌 기사가 있

다.

103 본래는 기자들이 직접 취재해야 할 것이로되 똑같은 내용을 앵무새처럼 수십 번씩 되풀이하는 게 피차 번거로워 이쪽에서 아예 통보하게 되었다는 설명이었다.

104 주머니에 웬만큼 여유가 생기니까 몸생각을 하지 않을 수 없는 노릇이다.

105 그리고 또 추운 겨울에는 따뜻하게 덥혀온 돌을 내 손위에 놓아 주기도 하였습니다.

106 물 밖의 따가운 여름 햇살과 차가운 겨울 바람 때문에 그는 차츰 지치고 있었습니다.

107 그리고 뒷얘기를 더 들을 것도 없이 그 길로 곧 자신의 예감을 좇아 나선 것이었다.

108 사내는 그때 과연 몸을 불태울 듯이 뜨거운 어떤 태양의 불볕을 견디고 있었다.

6.세트 E(107 문장)

001 그렇게 잘 맡아 오실 분이 이 세상에 다시 있을 것 같지 않습니다.

002 아버님께서 시골 면장 노릇이나 조용히 하셨던들 저희 집안이 그런 풍랑이야 겪었겠습니까?

003 궂은 날이나 악조건 속에서도 살아갈 수 있어야만 씩씩한 비눗방울이 될 수 있지요.

004 처음 사용하는 플라스틱 컵 등은 아무리 깨끗하게 보여도 제조 공정에서 특수 기름이 묻었을 수도 있기 때문이지요.

005 칠엽수 열매의 껍질을 벗기고 얇게 썰어 즙을 내어 주둥이가 큰 병에 넣습니다.

006 떠나는 배를 보며 그의 무릎이 결코 추위 때문이 아닌 자신의 내부로부터의

충격에 의해 떨기 시작했다.

007 문득 허준의 눈속에 작은 연민의 정이 지나갔을 때 끝내 어머니가 무너졌고 어머니의 어깨에 아들의 손이 조용히 놓여졌을 뿐이었다.

008 지구의 운동 방향으로 잰 빛의 속도는 이에 수직으로 잰 빛의 속도보다 더 큰 값이 될 것이다.

009 잘 봐 주세요라는 태도를 표방하여 특별 대우를 받기 원하는 타입이다.

010 노란 편지 봉투에 싼 미라사탕 아니면 잔칫상에서 염치 불구하고 집어넣었음직한 약과나 다식 따위였다.

011 내 유년기의 기억의 첫 장을 꽉 채우다시피한 기다림은 그리 오래 가지 않았다.

012 뒷간도 재미있지만 뒷간에서 너무 오래있다 나왔을 때의 세상의 아롬다움은 유별났다.

013 채찍처럼 세차고 폭포수처럼 시원한 빗줄기가 복더위와 달음박질로 불화로처럼 단 몸뚱이를 사정없이 후려치면 우리는 드디어 폭발하고 만다.

014 내가 최초로 맛본 비애의 기억은 앞뒤에 아무런 사건도 없이 외따로인 채 다만 풍경만 있다.

015 그때만 해도 엄마 등에 업혔을 때하고는 달리 서러움을 적당히 고조시키고 싶어 꾀까지 썼다.

016 서민 중에서는 힘센 그의 남편의 힘을 얻고 싶어 마냥 허위적거리는 산부의 손을 꼭 쥐어 주기도 한다.

017 경종비는 왕이 죽은 후 왕의 숙부와 통하여 아들을 낳았고 인종은 자기 이모와 결혼하였다.

018 곧 자녀의 개념은 이 같은 성정자연주의의 테두리 밖에서 이해되어야 했던 것이다.

019 저의를 눈치 못 챈 노총각은 마구 뛰어올라가 시비를 걸고 마침내 엎치락뒤치락 싸우게 됐다.

020 한국 여성사의 내면적 저력은 곧 이 옛터주의에서 새터주의로의 탈출 과정이랄 수가 있다.

021 돼지코가 흰 운동화를 사면 나는 어머니에게 한사코 졸라서 돼지코의 흰 운동화와 똑같은 것을 사서 신었다.

022 적십자부의 간부도 맡고 교회에서 고등학생부의 간부까지 맡아 매우 바쁘게 지내고 있었다.

023 독일남부나 스위스의 경치는 말로 다 할 수 없을 정도로 풍요롭고 아름다우며 산뜻하다.

024 고등학교의 교육 내용은 오히려 부정적인 요소가 긍정적인 요소보다 더 많다.

025 언 귀를 세우고 창 밖을 내다보면 칼끝 같은 바람이 불고 눈 덮인 우리 시대의 끝이 보인다.

026 그는 각각 구별되는 위치에 있는 열개의 바퀴에 숫자를 기억시킬 계획이었다.

027 주먹을 불끈 쥐면 포도송이를 꽉 잡는 기분도 한껏 맛볼 수 있다.

028 이와 같은 남녀에 따른 뇌 크기의 차는 원숭이한테서도 볼 수 있다.

029 이 여섯 층으로 된 세포체의 층이 대뇌피질 기능의 기본단위 역할을 한다.

030 성상세포는 축색돌기의 연장하는 상태에 따라 중총세포와 단축세포 두 종류로 구분한다.

031 따라서 사람의 대뇌피질내에 있는 수백억 개의 추체세포들도 수십 또는 수백의 범주로 분류된다.

032 무아지경에서 깨어 머리속으로 가계부 정리와 같은 계산을 하기 시작하면 뇌파는 곧 베타타입으로 된다.

033 이는 받은 자극을 뇌포에 전하는 신경세포의 축색돌기의 직경의 크기가 다양하기 때문이다.

034 그런데 왜 이토록 중요한 노동시간 단축 투쟁이 우리나라에서는 아직 본격적으로 전개되지 않고 있는 것일까?

035 흰개미는 분류학상 바퀴벌레와 그 공통 시조로부터 유래된 것으로 알려져 있다.

036 이러한 꿀의 농축과정과 꿀 자체의 효소작용에 의하여 당이 주성분인 꽃꿀은 포도당과 과당으로 분리된 벌꿀로 되는 것이다.

037 초파리의 복안은 좌우쌍으로 존재하며 크기는 두부 표면적의 약 반을 차지한다.

038 선생님의 편지에는 제가 가야 할 길들이 몇 갈래 적혀 있더군요.

039 태양 조석의 평균 높이는 평균 태음 조석의 약 반 정도가 되며 이는 지구로부터 태양까지의 거리가 훨씬 더 멀기 때문이다.

040 해수의 염을 구성하는 용존물질의 조합이 매우 복잡하기 때문에 염분을 직접 측정하기는 어렵다.

041 동물 플랑크톤은 대부분 그들의 먹이가 되는 식물 플랑크톤의 분포 때문에 해양의 표층 가까이에서 발견된다.

042 일차 초식동물은 이차 초식동물에 의해 먹히고 이것은 다시 삼차 육식동물에 의해 먹힐 수 있다.

043 예컨대 온대 해양에서는 소위 식물 생물량의 대증식이 수온과 평균 일조량의 증가가 나타나는 봄철에 일어날 수 있다.

044 비록 이러한 원칙에 예외는 있지만 개념에 영향을 줄 만큼 많지는 않다.

045 그 때 세상 경험을 널리 하려고 숲에서 나왔던 풍뎅이란 놈이 그 승용차의 앞 유리창에 부딪쳐 박살이 났다.

046 승용차의 주인은 언제 죽어도 죽을 까짓 풍뎅이의 죽음에 눈하나 깜빡 안 했다.

047 우리의 마음속은 아직도 옛날 빈민굴의 뒷골목에서와 같이 떨고 있다.

048 훨씬 멀리까지 전망이 좋았으나 서쪽만은 저무는 태양의 아지랑이가 온갖 것의 형태를 녹여 내고 있었다.

049 특히 내가 이 전략에 동의할 수 없는 것은 그 여의사의 처방근거였다.

050 사실은 몇 년 전부터 우리 부부는 그 문제에 관해 이야기를 나눠 왔다.

051 두통이 심할 때는 손톱 주위를 볼펜자루 끝으로 꼭꼭 눌러서 대단히 아픈부위를 바늘로 찔러 피를 살짝 빼 준다.

052 피부색소의 자외선 저항력이 약한 흰 피부 소유자일수록 위험성은 크다는 것이다.

053 군에 남아 곡절 끝에 육군 총수에 올랐던 김씨는 이제 무대의 뒷면으로 사라지게 됐다.

054 대통령 선거이후 국민의 관심이 애정의 영역 밖으로 밀려나 있었던 셈이다.

055 농어민 후계자로 선정되면 특례 보충역으로 편입되는 병역 혜택과 함께 각종 영농정착금이 지원된다.

056 이 때문에 항공촬영을 할 때 큰 헬기가 필요하지만 아직 국내에서는 이 방식으로 촬영한 것은 없다.

057 벌써 주유소에서 휘발유 값을 결제할 수 있는 카드를 정유사들이 앞다투어 내고 있다.

058 성장률뿐 아니라 수출 증가율도 최하위를 기록하는 등 성장부문에 관한 한 지난해 우리의 경제 성적표는 최악의 상황을 나타냈다.

059 그러나 개별 주식들의 변동은 아주 활발해 최대수익주와 최대 손실주의 격차는 매우 컸다.

060 일본은 근대역사에서 왜곡된 한국인상을 어떻게 바로잡을 수 있을 것인가를 깊이 생각해 볼 필요가 있다.

061 최씨에 따르면 이들은 우리의 어깨를 쫙 펴게 해주고 기분을 방방 뜨게 해주는 사람들이다.

062 두달여 동안 그렇게 애를 먹인 피의자를 구속하는 개가를 올렸지만 박검사의 마음은 의외로 착잡했다.

063 유씨는 엄마의 행동 패턴을 변화시키면서 자연스럽게 어린이의 격리불안 장애를 극복하는 방법을 권한다.

064 충분한 산소를 들이마시면 신체의 불순물 제거가 촉진돼 요통치료에도 도움이 된다고 이원장은 말한다.

065 해수욕을 하기전과 마친 후 뜨거운 물로 샤워하거나 뜨거운 모래속에서 찜질을 하면 불필요한 근육의 긴장이 풀려 더욱 좋다.

066 이런 진흙팩 제품은 피부 노폐물의 제거에 효과적이라는 것이 업계의 주장이다.

067 국산 중대형 컴퓨터는 국내업체들이 막대한 연구비와 인력을 동원해 개발한 제품으로 성능이나 품질면에서 외국제품과 손색이 없는 제품이다.

068 그런 중대형 컴퓨터의 수출이 많이 돼야 관련업체의 매출이 확대되고 이를 바탕으로 더욱 성능이 향상된 제품을 만들 수 있다.

069 제조업의 경쟁력상실은 크게 생산제품의 성능이나 품질저하 또는 영업활동의 열세로 구분할 수 있다.

070 펜티엄으로 개인용 컴퓨터는 물론 중대형 컴퓨터에 들어가는 칩까지 독식하려 하고 있다.

071 펜티엄이 나온지 몇 달 뒤에야 간신히 샘플 몇 개를 얻은게 우리의 현실이다.

072 디지털기술의 발전은 또 컴퓨터를 인간과 대화하는 멀티미디어 컴퓨터로 변모시켰다.

073 업계는 독자개발 했다고 떠들지만 기껏해야 외국에 주문한 도면을 약간 고친 수준에 불과하다.

074 시제품을 빨리 만들어 봤자 어차피 최종제품의 규격만은 덩치 큰 미국시장에 맞출 수밖에 없음을 간파한 전략이다.

075 방풍식 라이터 핵심부품인 노즐 분사장치 기술을 보유하고 있는 일본업자들은 한사코 한국업체에 기술이전을 거부했다.

076 세밀한 현실 분석에 뿌리를 둔 정확한 미래예측 역시 정부의 임무이다.

077 고열의 아연파편에 화상만 잔뜩 입고 집에 들어올때면 어깨가 축 늘어졌다.

078 그럴 경우 공격대상은 이라크의 전투기 추적 레이더 기지와 공군 시설 등이다.

079 미국은 또 아예 이라크의 공군기지에 착륙해 있는 이라크 전투기를 몽땅 파괴할 것도 구상하고 있다.

080 양측은 상대방이 이 문제를 확대하면 본격 대응하겠다며 엄포만 놓고 있는 자세다.

081 실상 이런 유입은 학문 분야뿐 아니라 일반 어휘에서도 자주 볼 수 있다.

082 여기서 빈칸 띄어쓰기와 함께 독립신문의 또 하나의 특징인 옆줄치기에 대해서 몇 마디 덧붙이기로 한다.

083 서재필은 특히 일반 민중을 생각하여 전통적인 맞춤법을 고집했을 것으로 추측된다.

084 더욱이 우리나라처럼 예술전용채널이 없는 실정에서는 예술프로그램을 심야로만 내몰게 아니라 가족시간 대나 골든아워에도 편성할 필요가 있다.

085 그러나 왜 이런 고급프로그램은 심야에 중복편성돼 있어 밤잠을 못자게 하고 아까운 내용들을 놓쳐야만 하는지 애호가들은 고개를 갸웃거린다.

086 정부의 예산 뒷받침이 부족해 프랑스 지방도시 및 유럽주요도시 순회공연으로 이어지지 못했다.

087 서양의 웬만한 도시에 마련되어 있는 조각있는 광장은 동시에 어린이 놀이터이기도 하다.

088 일본에서는 또 웬만한 소도시마다 어린이들이 놀면서 과학적 탐구심을 키우고 지적인 꿈을 가꾸도록 하는 과학관들이 들어서 있다.

089 그 대신 생겨난 프로그램들이 채널선택기회가 많은 청소년과 주부 대상의 연예오락들이다.

090 장기적으로 우리의 민족예술이 숨을 쉴 수 있으려면 국민적인 관심과 애정이 없으면 안된다.

091 공급자 중시의 산업정책은 궁극적으로 기업의 창의력과 자주적 적응능력을 쇠퇴시켜 경쟁력 약화라는 유산을 남겨놓았다.

092 그렇다고 자금력 기술력 마케팅면에서 우수한 대기업집단의 활동을 억제만 해서도 안된다.

093 농림수산부문의 무역적자가 국제수지악화의 큰 요인이 되고 있음을 여실히 보여주고 있다.

094 대기업과의 계열화에 성공하면 그래도 생존의 길이 열리나 대금지불 등에서의 불리한 조건을 수용해야 한다.

095 아니 더 큰 장사를 위해 권력 앞에 무릎을 꿇었다고 보아야 옳을 것이다.

096 그러나 웬만한 정치가나 전직고관의 영전에서 으레 볼 수 있는 대통령의 조화며 조문이 있었다는 얘기가 없다.

097 처음 구도에서는 청와대에는 또 다른 분야의 특별 보좌역들은 있어도 문화담당자는 끼어 있지 않았다.

098 그의 이러한 정치우위특징은 공약했던 정책개혁 계획도 민자당의 정책기구를 통해서 추진하겠다고 밝힌데서 극명하게 확인된다.

099 척결의 전권을 주어 독립적인 활동으로 국민들이 사정기구를 신뢰하고 협조할 수 있도록 해야 할 것이다.

100 따라서 대표와 최고위원 경선에 나서는 후보들은 한자리에 모여 최고위원과 당무위원 대의원수를 대폭 줄이는 절충을 벌여야 한다.

101 서울과 평양을 몇 차례씩 오가며 출범시켰다는 남북 합의시대의 소득은 바로 이런 것이었다.

102 국정업무의 인수인계에 참여하고 요직의 인선자료까지 만들게 된다면 여기에

줄을 대려는 철새들의 접근이 있을 수 있기 때문이다.

103 또한 사실이 아니라면 정 대표는 이 대표의 명예를 크게 훼손시킨 것이다.

104 득표를 위한 대회가 감표를 가져오는 역효과를 낳지 않기 위해서는 맑고 산뜻한 집회였다는 인상을 남겨야 한다.

105 하지만 최근 복잡해진 환경분쟁은 환경조정위에 골프장과 돼지의 관계까지 과학적으로 풀도록 했다.

106 오존층의 구멍과 사막의 크기가 산업폐유로 오염되어 가는 연못처럼 가속적으로 확산된다면 정말로 끔찍한 일이다.

107 벌써 국내 화학회사들은 우리의 연구진을 곁눈질로 바라보며 외국기술의 도입을 타진하고 있다.

부록 2. PBS 에서의 음운 환경 기초 통계

1. PBS 에서의 길이별 분포

가. 음절수 및 어절수에 따른 분포

표 A.1 PBS에서 음절수에 따른 분포 ——▶

음절	문장수	%
16	1	0.17%
19	1	0.17%
21	1	0.17%
22	3	0.51%
23	2	0.34%
24	3	0.51%
25	11	1.87%
26	17	2.89%
27	17	2.89%
28	21	3.57%
29	21	3.57%
30	33	5.60%
31	28	4.75%
32	32	5.43%
33	32	5.43%
34	27	4.58%
35	26	4.41%
36	31	5.26%
37	30	5.09%
38	19	3.23%
39	24	4.07%
40	12	2.04%
41	21	3.57%
42	20	3.40%
43	12	2.04%
44	17	2.89%
45	13	2.21%
46	14	2.38%
47	10	1.70%
48	10	1.70%
49	8	1.36%
50	11	1.87%
51	9	1.53%
52	6	1.02%
53	7	1.19%
54	3	0.51%
55	1	0.17%
56	4	0.68%
57	6	1.02%
58	8	1.36%
59	3	0.51%
60	1	0.17%
61	4	0.68%
62	1	0.17%
64	4	0.68%
66	2	0.34%
67	1	0.17%
71	1	0.17%
계	589	100.00%

표 A.2 PBS에서 어절수에 따른 분포

어절수	문장수	%
10	123	20.88%
11	104	17.66%
12	97	16.47%
13	60	10.19%
14	51	8.66%
15	42	7.13%
16	37	6.28%
17	23	3.90%
18	17	2.89%
19	20	3.40%
20	15	2.55%
계	589	100.00%

나. PBS에서의 음소별 출현 빈도

표 A.3 음소별 출현 빈도

자음	출현횟수	%	모음	출현횟수	%
ㄱ	3379	6.97%	ㅏ	4349	8.97%
ㄲ	550	1.13%	ㅐ	1058	2.18%
ㄴ	4272	8.81%	ㅑ	228	0.47%
ㄷ	2209	4.56%	ㅒ	12	0.02%
ㄸ	776	1.60%	ㅓ	2409	4.97%
ㄹ	3292	6.79%	ㅔ	1065	2.20%
ㅁ	1770	3.65%	ㅕ	1123	2.32%
ㅂ	1268	2.62%	ㅖ	118	0.24%
ㅃ	149	0.31%	ㅗ	2091	4.31%
ㅅ	1737	3.58%	ㅘ	475	0.98%
ㅆ	460	0.95%	ㅙ	47	0.10%
ㅇ	1395	2.88%	ㅚ	291	0.60%
ㅈ	1504	3.10%	ㅛ	244	0.50%
ㅉ	280	0.58%	ㅜ	1634	3.37%
ㅊ	618	1.27%	ㅝ	141	0.29%
ㅋ	388	0.80%	ㅞ	24	0.05%
ㅌ	517	1.07%	ㅟ	231	0.48%
ㅍ	439	0.91%	ㅠ	179	0.37%
ㅎ	1194	2.46%	ㅡ	2706	5.58%
			ㅢ	559	1.15%
			ㅣ	3303	6.81%
계	26197	54.03%		22287	45.97%

2.2. PBS 에서의 음운 환경 출현 빈도

가. PBS에서의 2음소열 출현 빈도

(1) VV

표 A.4 VV 출현 빈도

	ㅏ	ㅐ	ㅑ	ㅒ	ㅓ	ㅔ	ㅕ	ㅖ	ㅗ	ㅘ	ㅙ	계
ㅏ	15	1	18	0	24	22	60	3	23	22	2	190
ㅐ	9	0	21	0	15	21	11	1	9	10	1	98
ㅑ	1	0	0	0	1	2	1	0	1	1	0	7
ㅒ	0	0	0	0	0	0	0	0	0	0	0	0
ㅓ	11	1	20	1	18	14	9	4	18	10	1	107
ㅔ	17	1	8	0	27	9	8	1	6	6	1	84
ㅕ	9	1	7	0	1	2	4	2	10	11	1	48
ㅖ	1	0	1	0	1	7	1	0	1	1	0	13
ㅗ	36	1	11	1	27	15	32	6	13	9	1	152
ㅘ	1	1	4	0	1	6	9	1	1	2	0	26
ㅙ	1	0	1	0	0	1	1	0	0	1	0	5
ㅚ	1	0	2	0	38	14	2	0	1	3	0	61
ㅛ	1	0	2	1	2	5	1	0	0	6	0	18
ㅜ	3	1	3	0	35	19	15	2	4	13	0	95
ㅝ	1	0	2	0	1	1	0	1	1	1	0	8
ㅞ	1	0	1	0	3	0	0	0	0	0	0	5
ㅟ	1	1	1	0	15	26	3	0	1	4	0	52
ㅠ	2	0	1	1	1	3	1	1	0	1	0	11
ㅡ	6	1	1	1	2	5	3	1	2	2	0	24
ㅢ	12	1	8	2	18	2	22	8	2	5	1	81
ㅣ	72	4	9	1	96	59	29	3	13	29	2	317
B	11	1	2	1	13	1	14	5	7	2	6	63
계	212	15	123	9	339	234	226	39	113	139	16	1465

	ㅢ	ㅛ	ㅜ	ㅝ	ㅞ	ㅟ	ㅠ	ㅡ	ㅢ	ㅣ	B	계
ㅏ	4	14	22	13	2	11	6	24	37	71	559	763
ㅐ	5	12	28	8	0	1	1	3	29	29	0	116
ㅑ	1	1	1	1	0	1	1	0	1	1	1	9
ㅒ	0	0	0	0	0	0	0	0	0	0	0	0
ㅓ	1	9	35	8	0	4	3	12	22	37	1	132
ㅔ	2	4	6	11	1	2	4	2	35	48	0	115
ㅕ	1	1	12	5	1	2	3	1	7	40	0	73
ㅖ	2	2	1	0	0	0	1	0	4	2	0	12
ㅗ	6	6	12	5	1	6	9	12	14	133	3	207
ㅘ	1	1	1	1	0	1	5	1	14	10	0	35
ㅙ	0	1	0	0	0	1	1	0	0	5	0	8
ㅚ	0	0	4	5	0	1	1	1	6	2	0	20
ㅛ	1	1	1	1	0	1	9	1	6	5	22	48
ㅜ	1	8	7	6	2	12	4	1	46	69	1	157
ㅝ	0	0	1	1	0	0	0	0	0	4	0	6
ㅞ	0	0	0	0	0	0	0	0	0	5	0	5
ㅟ	0	1	5	7	0	1	1	0	7	1	0	23
ㅠ	0	1	1	1	0	0	1	0	5	2	0	11
ㅡ	1	1	1	3	4	3	1	3	25	3	0	45
ㅢ	3	4	3	2	1	4	8	3	4	23	0	55
ㅣ	5	16	15	16	1	11	8	5	49	36	1	163
B	1	2	6	2	1	4	7	2	1	51	0	77
계	35	85	162	96	14	66	74	71	312	577	588	2080

(2) VC

표 A.5 VC 출현 빈도

	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	계
ㅏ	127	36	832	86	13	232	257	47	9	1639
ㅐ	48	13	68	63	4	3	86	1	6	292
ㅑ	16	0	4	1	0	6	5	0	1	33
ㅒ	0	0	0	0	0	0	0	0	0	0
ㅓ	88	24	394	192	30	109	168	116	3	1124
ㅔ	1	17	101	11	13	5	63	0	6	217
ㅕ	41	5	284	78	3	55	38	12	1	517
ㅖ	0	1	8	4	1	0	8	0	5	27
ㅗ	103	11	186	36	13	79	118	9	4	559
ㅘ	18	1	76	22	4	27	13	0	1	162
ㅙ	1	0	8	4	1	0	1	0	0	15
ㅚ	2	1	88	1	3	11	11	1	0	118
ㅛ	10	1	3	0	2	0	5	0	1	22
ㅜ	58	7	309	17	4	156	92	2	1	646
ㅝ	0	0	70	9	2	17	1	0	0	99
ㅞ	0	0	8	0	1	0	0	0	0	9
ㅟ	0	4	15	13	2	5	7	11	0	57
ㅠ	11	1	21	0	1	9	11	0	0	54
ㅡ	42	10	886	21	9	580	203	24	1	1776
ㅢ	0	6	23	0	4	0	38	0	2	73
ㅣ	65	26	481	145	53	143	229	46	9	1197
계	631	164	3865	703	163	1437	1354	269	50	8636

	ㅅ	ㅆ	ㅇ	ㅈ	ㅉ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	계
ㅏ	119	16	304	0	5	64	57	56	54	0	675
ㅐ	59	12	98	0	3	24	11	14	13	0	234
ㅑ	1	0	98	0	0	2	12	1	1	0	115
ㅒ	0	0	0	0	0	0	0	0	0	0	0
ㅓ	159	35	190	0	8	18	48	17	23	0	498
ㅔ	178	10	2	0	2	18	6	27	31	0	274
ㅕ	31	12	172	0	1	8	12	12	9	0	257
ㅖ	22	1	0	0	1	3	1	5	1	0	34
ㅗ	143	3	214	0	2	47	34	34	34	2	513
ㅘ	30	6	28	0	0	8	6	5	8	0	91
ㅙ	4	1	1	0	0	1	1	0	1	0	9
ㅚ	15	1	2	0	1	6	1	4	5	0	35
ㅛ	10	0	56	0	0	5	2	1	2	0	76
ㅜ	60	3	125	0	2	27	7	46	10	0	280
ㅝ	3	1	1	0	0	1	2	1	1	0	10
ㅞ	1	0	1	0	1	0	0	1	0	0	4
ㅟ	7	0	2	0	2	6	1	4	2	0	24
ㅠ	11	1	3	0	1	7	1	23	1	0	48
ㅡ	40	1	81	0	1	11	30	46	13	0	223
ㅢ	49	1	0	0	1	22	9	19	15	0	116
ㅣ	150	105	17	0	9	67	46	51	41	0	486
계	1092	209	1395	0	40	345	287	367	265	2	4002

(3) CV

표 A.6 CV 출현 빈도

	ㅏ	ㅐ	ㅑ	ㅒ	ㅓ	ㅔ	ㅕ	ㅖ	ㅗ	ㅘ	계
ㄱ	28	11	0	0	8	2	5	5	16	7	82
ㄲ	87	25	0	0	33	63	14	4	98	45	369
ㄴ	330	111	35	1	103	102	100	10	96	5	893
ㄷ	10	11	0	0	6	1	0	0	12	0	40
ㄸ	387	117	0	0	83	6	0	0	79	0	672
ㄹ	44	15	9	0	24	2	59	1	32	0	186
ㅁ	262	57	3	0	63	43	204	1	150	1	784
ㅂ	16	2	0	0	7	1	3	0	11	0	40
ㅃ	40	7	1	0	14	1	10	0	20	0	93
ㅅ	296	87	4	1	413	91	10	1	123	0	1026
ㅆ	54	23	0	0	130	8	1	0	19	0	235
ㅇ	22	5	20	1	16	63	24	1	5	1	158
ㅈ	18	1	0	0	18	6	0	0	7	1	51
ㅉ	69	10	0	0	54	6	0	0	51	1	191
ㅊ	87	38	0	0	89	42	34	0	33	2	325
ㅋ	80	25	1	0	41	36	22	1	36	10	252
ㅌ	118	53	0	0	90	28	11	0	63	0	363
ㅍ	53	15	0	0	20	27	58	6	35	4	218
ㅎ	93	30	4	0	2	2	13	1	4	18	167
계	2094	643	77	3	1214	530	568	31	890	95	6145

	ㅙ	ㅚ	ㅛ	ㅜ	ㅝ	ㅞ	ㅟ	ㅠ	ㅡ	ㅢ	ㅣ	계
ㄱ	0	1	3	16	1	0	2	1	99	0	15	138
ㄲ	1	1	13	36	4	1	3	1	55	1	65	181
ㄴ	2	20	18	60	9	3	17	30	602	60	267	1088
ㄷ	1	2	0	8	0	0	4	0	6	0	1	22
ㄸ	1	19	0	14	0	0	10	0	50	0	10	104
ㄹ	0	1	1	2	0	0	0	5	2	0	55	66
ㅁ	1	1	4	213	1	0	5	2	102	31	156	516
ㅂ	0	0	0	11	0	0	0	0	0	0	6	17
ㅃ	0	0	1	34	0	0	0	0	10	0	11	56
ㅅ	5	4	1	183	1	0	24	3	129	1	360	711
ㅆ	2	1	0	20	0	0	1	0	117	1	83	225
ㅇ	0	1	10	17	9	0	8	4	121	55	120	345
ㅈ	0	1	0	14	0	0	1	0	4	0	10	30
ㅉ	0	2	0	20	0	0	0	0	7	0	60	89
ㅊ	0	14	0	73	3	1	15	0	38	5	144	293
ㅋ	2	2	1	9	1	0	4	0	55	0	62	136
ㅌ	0	6	0	22	0	0	6	1	105	2	12	154
ㅍ	0	1	43	53	0	0	1	20	30	2	71	221
ㅎ	0	2	1	2	1	0	2	1	4	0	8	21
계	15	79	96	807	30	5	103	68	1536	158	1516	4413

(4) CC

표 A.7 CC 출현 빈도

	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	ㅅ	계
ㄱ	45	105	0	17	71	0	0	19	38	18	313
ㄲ	0	0	0	0	0	0	0	0	0	0	0
ㄴ	0	20	154	0	23	4	161	0	6	243	611
ㄷ	25	117	0	6	387	0	0	10	28	10	583
ㄸ	0	0	0	0	0	0	0	0	0	0	0
ㄹ	0	49	55	0	64	240	94	0	14	158	674
ㅁ	0	25	78	0	6	1	33	0	2	56	201
ㅂ	7	55	0	5	40	0	0	3	8	1	119
ㅃ	0	0	0	0	0	0	0	0	0	0	0
ㅅ	0	0	0	0	0	0	0	0	0	0	0
ㅆ	0	0	0	0	0	0	0	0	0	0	0
ㅇ	0	6	76	0	7	1	69	0	1	117	277
ㅈ	0	0	0	0	0	0	0	0	0	0	0
ㅉ	0	0	0	0	0	0	0	0	0	0	0
ㅊ	0	0	0	0	0	0	0	0	0	0	0
ㅋ	0	0	0	0	0	0	0	0	0	0	0
ㅌ	0	0	0	0	0	0	0	0	0	0	0
ㅍ	0	0	0	0	0	0	0	0	0	0	0
ㅎ	0	0	0	0	0	0	0	0	0	0	0
B	143	9	22	34	15	2	24	25	2	42	318
계	220	386	385	62	613	248	381	57	99	645	3096

	ㅆ	ㅇ	ㅈ	ㅉ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	B	계
ㄱ	100	0	28	102	22	6	13	24	23	0	318
ㄲ	0	0	0	0	0	0	0	0	0	0	0
ㄴ	14	0	0	28	78	47	65	49	0	1	282
ㄷ	27	0	6	41	29	1	1	4	11	0	120
ㄸ	0	0	0	0	0	0	0	0	0	0	0
ㄹ	25	0	0	14	44	20	27	40	0	0	170
ㅁ	7	0	0	2	23	3	8	35	0	0	78
ㅂ	76	0	7	43	15	2	1	2	4	0	150
ㅃ	0	0	0	0	0	0	0	0	0	0	0
ㅅ	0	0	0	0	0	0	0	0	0	0	0
ㅆ	0	0	0	0	0	0	0	0	0	0	0
ㅇ	1	0	0	9	36	14	25	12	0	0	97
ㅈ	0	0	0	0	0	0	0	0	0	0	0
ㅉ	0	0	0	0	0	0	0	0	0	0	0
ㅊ	0	0	0	0	0	0	0	0	0	0	0
ㅋ	0	0	0	0	0	0	0	0	0	0	0
ㅌ	0	0	0	0	0	0	0	0	0	0	0
ㅍ	0	0	0	0	0	0	0	0	0	0	0
ㅎ	0	0	0	0	2	0	0	0	0	0	2
B	1	0	40	1	24	8	10	8	39	0	131
계	251	0	81	240	273	101	150	174	77	1	1348

나. PBS에서의 3음소열 출현 빈도

(1) 유성음화 및 자음 약화의 경우

표 A.8 유성음화 및 자음 약화의 출현 빈도

음소	H+C+H	H+C+L	H+C+N	H+C+Q	L+C+H	L+C+L	L+C+N	L+C+Q	계	%
ㄱ	194	363	12	1	461	623	20	1	1675	22.82%
ㄷ	74	264	8	0	150	348	12	1	857	11.67%
ㅂ	116	122	3	1	159	204	2	0	607	8.27%
ㅈ	120	190	0	0	299	278	0	0	887	12.08%
ㅎ	21	169	0	0	36	389	0	0	615	8.38%
계	525	1108	23	2	1105	1842	34	2	4641	63.22%
%	7.15%	15.09%	0.31%	0.03%	15.05%	25.09%	0.46%	0.03%	63.22%	

음소	N+C+H	N+C+L	N+C+N	N+C+Q	Q+C+H	Q+C+L	Q+C+N	Q+C+Q	계	%
ㄱ	208	387	0	0	83	175	0	0	853	11.62%
ㄷ	105	371	0	0	38	73	0	0	587	8.00%
ㅂ	105	119	0	0	31	80	0	0	335	4.56%
ㅈ	133	290	0	0	43	70	0	0	536	7.30%
ㅎ	59	330	0	0	0	0	0	0	389	5.30%
계	610	1497	0	0	195	398	0	0	2700	36.78%
%	8.31%	20.39%	0.00%	0.00%	2.66%	5.42%	0.00%	0.00%	36.78%	

- H는 고모음을 의미함.

- L은 저모음을 의미함.

- C는 대상 자음의 위치를 나타냄.

- N은 비음의 의미함.

- Q는 유음을 의미함.

(2) 탄설음화의 경우

유음이 모음과 모음사이에서 탄설 음화되는 경우는 1,603종이 포함 되었으며, 이 출현 빈도에는 종성 유음 /ㄹ/뒤에 다음 음절의 초성 /ㅎ/가 탈락되어 종성 유음이 다음 음절의 초성으로 실현될 수 있는 경우도 포함된 것임.

12. 오프라인 한글 글씨 데이터베이스 구축

고려대학교
이성환

여 백

12. 오프라인 한글 글씨 데이터베이스 구축

1장. 서론

1절. 연구 개발의 목적 및 내용

본 연구 개발의 목적은 한글 글씨 인식 분야의 연구자들에게 공동으로 필요한 다양한 변형을 포함하는 대용량의 한글 글씨 데이타베이스를 구축 및 제공하여 한글 글씨 인식과 관련한 연구를 활성화시키고, 효과적인 한글 글씨 인식 알고리즘의 개발을 유도하며, 개발된 인식 알고리즘의 객관적인 성능 평가를 가능하게 함으로써 한글 글씨 인식 시스템의 상용화를 촉진시키는 데 있다.

본 연구 개발에서는 한글 글씨 인식에 관한 연구자들이 필요로 하는 가장 기본적이고 필수적인 한글 글씨 데이타베이스를 구축하기 위해 1, 2 차년도에는 KS C 완성형 한글 2,350 자 중에서 사용 빈도순 상위 1,000 자 1,000 벌을 수집하였으며, 이번 3 차년도에는 사용 빈도순 차상위 500 자에 대한 1,000 벌의 필기 데이타를 수집하여 데이타베이스를 구축하였다. 이로서 KS C 완성형 한글 2,350 자 중에서 사용 빈도순 상위 1,500 자 1,000 벌을 수집하였다. 3 차년도의 주요 연구 개발 내용은 다음과 같다.

■ 한글 글씨 데이타베이스 설계

- 대상 문자 및 수집 인원 선정
- 수집 용지 설계
- 필기자, 필기 도구 및 필기 유형 선정
- 데이타베이스 구성 방식 설계

■ 한글 글씨 데이타 수집

■ 필기 데이타 수집 용지의 스캔 및 저장

• 수집 용지 스캔, 문자 단위 분할 및 저장

■ 데이타베이스의 검증 및 보완

•데이타베이스 검증 및 보완

■ 데이타베이스의 통계적 특성 분석

■ 사용자 인터페이스 개발

2 장. 3 차년도 한글 글씨 데이타베이스 설계

1 절. 데이타베이스 구축 시의 고려 사항

한글 글씨 데이타베이스는 다양한 필기 변형을 충분히 수용하는 방대한 양의 데이타베이스이어야 한다. 즉, 성별, 연령, 지역, 그리고 직업 등에 편중되지 않도록 광범위한 계층의 필기자를 선정함으로써 특정 요소에 편중되어 발생될 수 있는 국부적인 필기 특성을 배제해야 한다. 이와 같은 다양한 필기 변형을 포함하는 대용량의 한글 글씨 데이타를 사용해야만 효과적인 한글 글씨 인식 시스템을 개발할 수 있고, 그 성능을 객관적으로 평가하는 것이 가능하다.

3 차년도에는 수집 지역 선정에 있어서 다양한 필기 형태를 수집한다는 취지 아래 1, 2 차년도와는 다른 지역을 선정하도록 고려하였다. 이에 반해, 수집 용지 및 필기구 등은 1, 2 차년도와 일관성을 유지하기 위해 동일하게 양면 아트지, 건식 복사지, 갱지 그리고 사인펜, 볼펜, 수성 마킹펜 0.5mm 를 사용하였다. 또한, 수집된 필기 수집 용지로부터 문자 영상을 스캔하여 문자 단위로 분할하고 저장하는 방법에 있어서도 1, 2 차년도와 동일하게 명암 영상(256 단계)으로 저장하였다.

결론적으로, 3 차년도에는 1, 2 차년도와의 연계성을 염두에 두어 수집 지역 선정에 있어서는 다양한 한글 글씨 데이타를 수집하기 위해 중복되는 지역을 피하고, 수집 용지 및 필기구, 영상 저장 방법 등은 데이타베이스 품질의 일관성을 위해 1, 2 차년도와 동일하게 사용하였다.

2 절. 데이타베이스 설계

1. 설계 방법

본 연구 개발에서는 국내외의 5 종류의 오프라인 한글 글씨 데이타베이스 구축의 사례 연구[김문현 92, 방승양 92, Fenri93, Saito85, Wilso90]를 통하여 한글 글씨 데이타베이스가 가져야 할 기준을 정립하고 효과적인 한글 글씨 데이타베이스를 설계하였고, 국내의 학계, 연구소, 산업체 등에서 한글 글씨 인식에 관한 연구를 수행하고 있는 연구자들을 초청, 한글 글씨 데이타베이스 구축을 위한 자문회의를 개최하여 데이타베이스의 설계 내용에 대한 토의를 거쳐 일부 바람직한 의견을 반영하였다.

또한, 데이타베이스의 설계 시에 다음과 같은 평가의 착안점을 고려하였다.

- 필기 환경의 적합성
  - 수집 용지의 재질, 두께, 특성 및 형식
  - 필기구의 종류 및 필기 형태
- 필기자의 다양성
- 데이타베이스의 완전성
  - 다양한 필기 형태의 포함 여부
  - 데이타베이스의 품질 및 사용 시의 편의성
- 인식 알고리즘 개발 및 성능 평가에의 활용도

2. 설계 내용

3 차년도에 구축된 한글 글씨 데이타베이스의 세부 설계 내용은 다음과 같다.

(1) 대상 문자

KS C 완성형 한글 2,350 자 중 사용 빈도순 차상위 500 자

(2) 필기 대상자

다양한 계층, 직업, 연령, 지역 분포를 고려한 1,000 명 이상

(3) 데이타베이스의 크기

500 자 1,000 벌

(4) 수집 문자의 필기 형태

정서체와 자유 필체 두 종류의 필기 형태

(5) 수집 용지(그림 2.1 참조)

- 크기 : A4
- 색상 : 백색
- 재질 : 양면 아트지, 건식 복사지, 갱지 세 종류를 사용
- 형식 :
  - 문자 수 : 500 자 1 벌을 2 장(256 자 1 장, 244 자 1 장)의 용지에 수록
  - 구성 : 10 종류의 서로 다른 문자 배열로 구성
- 필기칸 :
  - 크기 : 문자당 9mm x 9mm 의 사각형
  - 색상 : 적색
  - 필기칸 간의 간격 : 가로 2mm, 세로 5mm
- 예시 문자 :
  - 크기 : 10pt 고딕체
  - 색상 : 적색

- 위치 : 필기할 칸의 상단에 인쇄

• 수집 용지에 포함된 필기자 정보 : 성별, 좌우 손잡이 여부, 필기구 종류, 필기 형태, 지역, 직업, 나이, 이름

(6) 필기구

• 색상 : 흑색

• 종류 : 사인펜, 볼펜, 수성 마킹펜 세 종류를 사용

(7) 수집 용지의 스캔 및 저장

• 문자 추출 영역 : 필기칸 보다 우측과 하단이 1mm 큰 10mm x 10mm

• 문자 영상 스캔 : 300 DPI 해상도, 256 단계 (1 pixel 당 1 byte) 명암 영상

• 저장 :

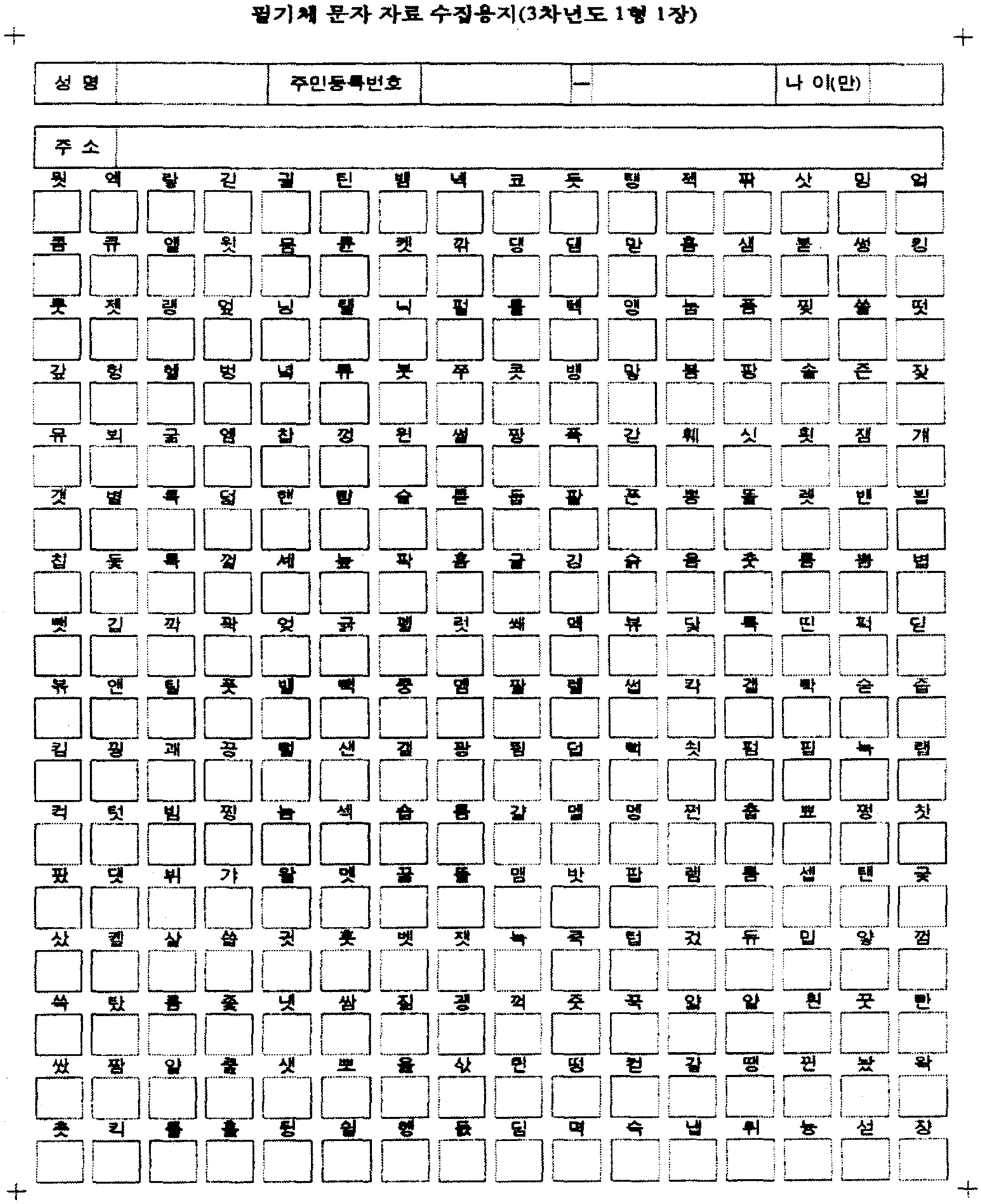
- 단위 문자에 대하여 위에서 아래로, 좌에서 우로 저장

- 수집 년도 별로 500 자 1 벌을 1 개의 화일에 저장

- 완성형 한글 코드 순으로 저장

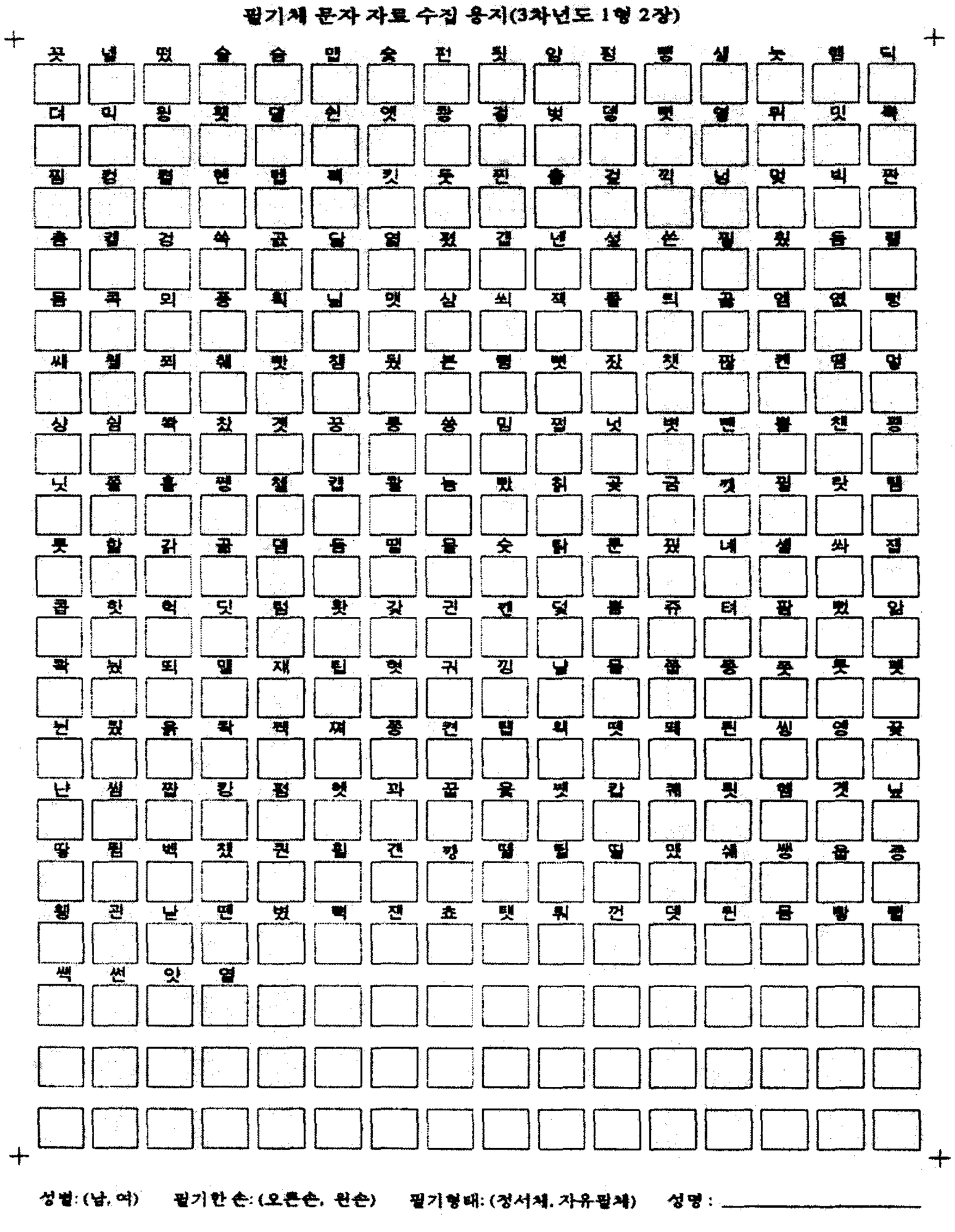
• 압축 : compress 로 압축

• 저장 매체 : 고용량 하드 디스크



(a) 수집 용지 첫번째장 (60% 축소 인쇄)

그림 2.1 3 차년도 한글 글씨 데이타 수집을 위한 수집 용지



(b) 수집 용지 두번째장 (60% 축소 인쇄)

그림 2.1 3 차년도 한글 글씨 데이타 수집을 위한 수집 용지(계속)

(8) 데이타베이스 구성

3 차년도 데이타베이스는 500 자 한 벌을 완성형 한글 코드순으로 하나의 화일에 저장하였다. 데이타베이스를 구성하는 각 화일에 대한 작명 규약은 그림 2.2 와 같다.

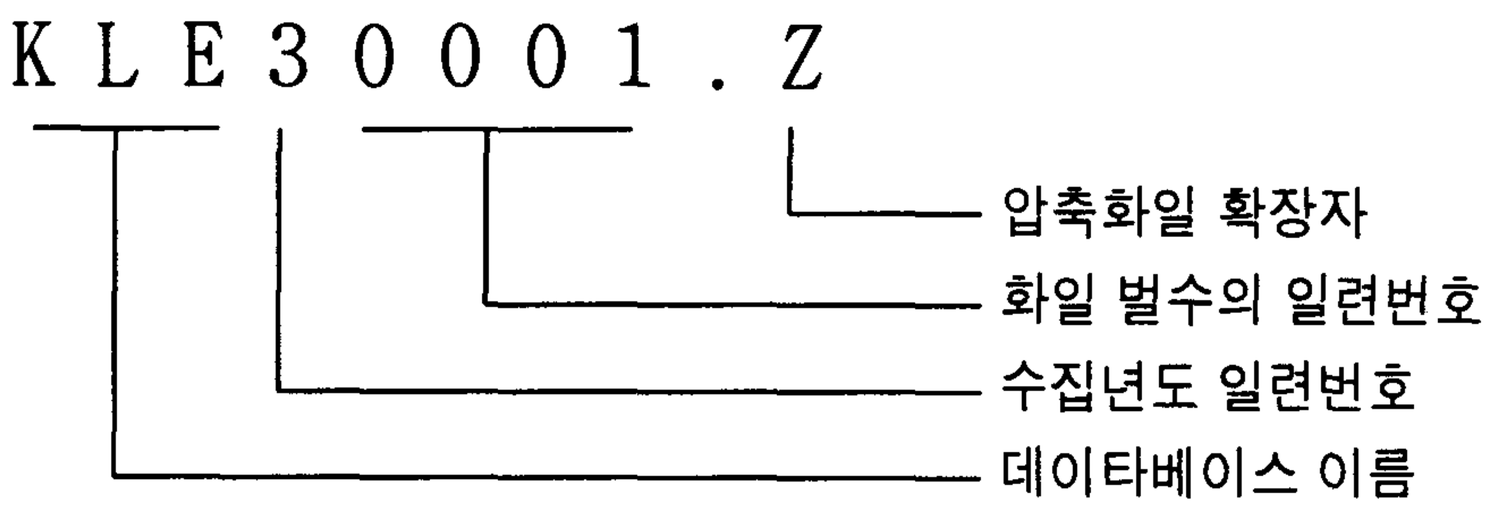


그림 2.2 화일 이름 작명 규약

그림 2.3 은 500 자 한 벌이 하나의 화일에 저장되는 순서를 나타내 준다.

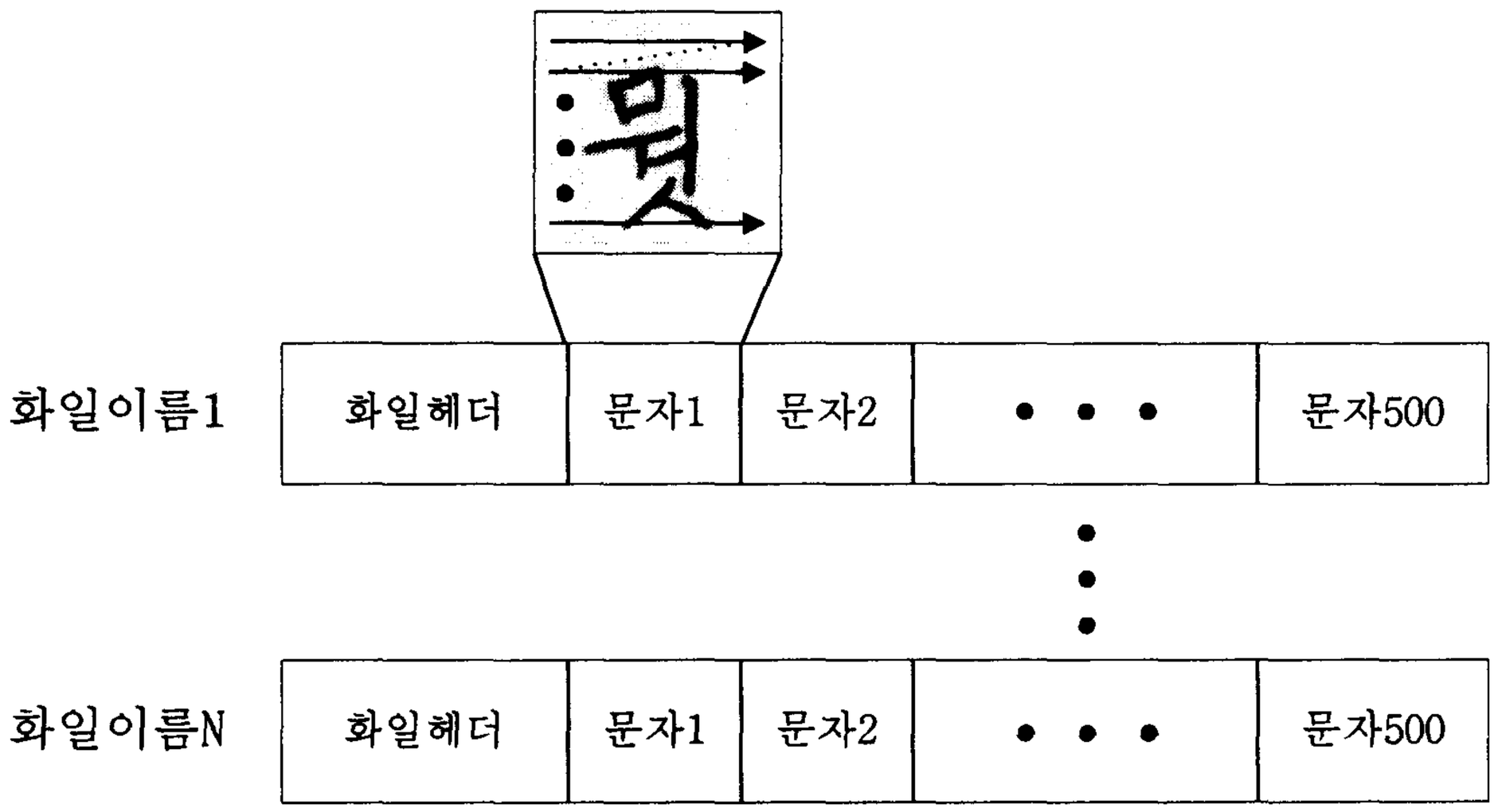


그림 2.3 3 차년도 데이타베이스 화일의 구성 예

그림 2.3 에서 각 화일의 화일 헤더는 다음과 같이 구성된다.

1byte - 수집 년도의 차수(1: 1995 년, 2: 1996 년, 3: 1997 년)

2byte - 벌수의 일련 번호

1byte - 문자의 필기 형태(1 : 정서체, 2 : 자유 필체)

1byte - 문자의 품질 표시(1 : 양호, 2 : 보통, 3 : 불량)

2byte - 수집 년도의 전체 문자 수

1byte - 문자의 가로 픽셀 수

1byte - 문자의 세로 픽셀 수

1byte - 성별 구분(1 : 남자, 2 : 여자)

1byte - 필기한 손(1 : 왼손, 2 : 오른손)

1byte - 필기 용지 종류(1: 양면 아트지, 2: 건식 복사지, 3: 갱지)

1byte - 필기구 종류(1: 사인펜, 2: 볼펜, 3: 수성 마킹펜 0.5mm)

1byte - 스캐너 종류의 코드

1byte - 스캐너 brightness

1byte - 스캐너 contrast

2byte - 스캐너 해상도(dpi)

7byte - reserved zone

3 장. 3 차년도 한글 글씨 데이타 수집

3 차년도 한글 글씨 데이타는 KS C 완성형 한글 2,350 자 중에서 사용 빈도순 차상위 500 자를 선정하여 수집하였다. 필기자는 1, 2 차년도의 수집 지역과는 다른 지역에서 1,000 명 이상을 선정하였다. 한글 글씨 데이타의 수집 용지 및 필기 도구는 1, 2 차년도와 일관성을 유지하기 위해 동일하게 사용하였다. 한글 글씨 데이타의 수집 기간 및 수집된 데이타의 양은 다음과 같다.

- 수집 기간 : 2 개월
- 수집된 데이타의 양 : KS C 완성형 한글 사용 빈도순 차상위 500 자 1,000 벌

필기자에게는 정성스럽게 필기하도록 한 정서체와 본인의 평소 필기 형태로 자연스럽게 필기하도록 한 자유 필체 등 두 종류의 필체로 필기하도록 하였고, 그림 3.1 과 같

은 필기시의 주의 사항을 제시하여 필기 전에 숙지하도록 함으로써 데이타베이스의 품질을 유지하고자 하였다. 그림 3.2는 필기된 수집 용지의 예를 보여 준다.

필기 시의 주의사항

필기체 문자 자료 수집에 협조해 주셔서 대단히 감사합니다. 이 자료는 문서의 자동 입력을 위한 필기체 문자 인식 연구의 기초 자료로 사용하기 위한 것으로, 데이타베이스로 저장되어 영구히 보존되며 대학 및 연구 기관에 배포되어 우리나라 사람들의 필기 습관 파악과 문자 인식 연구에 활용됩니다.

수집 문자는 KS C 5601 한글 사용빈도순 차상위 500자로서, 2장의 용지로 나누어 필기하도록 용지를 구성하였으므로 필기자는 정서체 1벌(2장)과 평소 본인의 자유필체 1벌(2장)을 필기하도록 준비하였습니다. 먼저 정서체로 500자 1벌을 필기한 다음, 자유필체로 500자 1벌을 필기하여 주십시오. 아래의 주의사항과 예를 참조하시어 수집 용지에 필기해 주시면 감사하겠습니다.

- 주 의 사 항 -

1. 수집 용지의 첫째장과 둘째장이 같은 형인지 확인해 주십시오.

2. 박스 상단의 예시 문자를 참조하여 박스를 벗어나지 않는 적당한 크기로 박스 안에 필기해 주십시오.

3. 만약 잘못 필기한 경우는 고치지 말고 잘못 필기한 글자 위에 'X' 표시를 한 다음, 둘째장 하단의 빈 박스에 잘못 필기한 글자를 다시 필기해 주십시오.

4. 반드시 배포된 검정색 필기구만을 사용하여 필기해 주십시오.

5. 볼펜의 경우는 다른 종이에 충분히 써 본 다음 잉크가 확실히 나오는 경우에 필기해 주십시오.

6. 수집 용지는 컴퓨터로 처리되므로 용지가 구겨지거나 더럽혀지지 않도록 주의해 주십시오.

7. 되도록 평평한 것을 받치고 필기해 주시고 수집 용지 두장을 겹쳐 놓고 필기하지 말아 주십시오.

8. 첫번째 수집 용지의 성명, 주민등록번호, 나이, 주소를 반드시 필기하여 주시고, 각 용지 하단의 성별, 필기한 손, 필기구의 종류, 필기형태를 반드시 표시해 주십시오.

그림 3.1 필기 시의 주의 사항

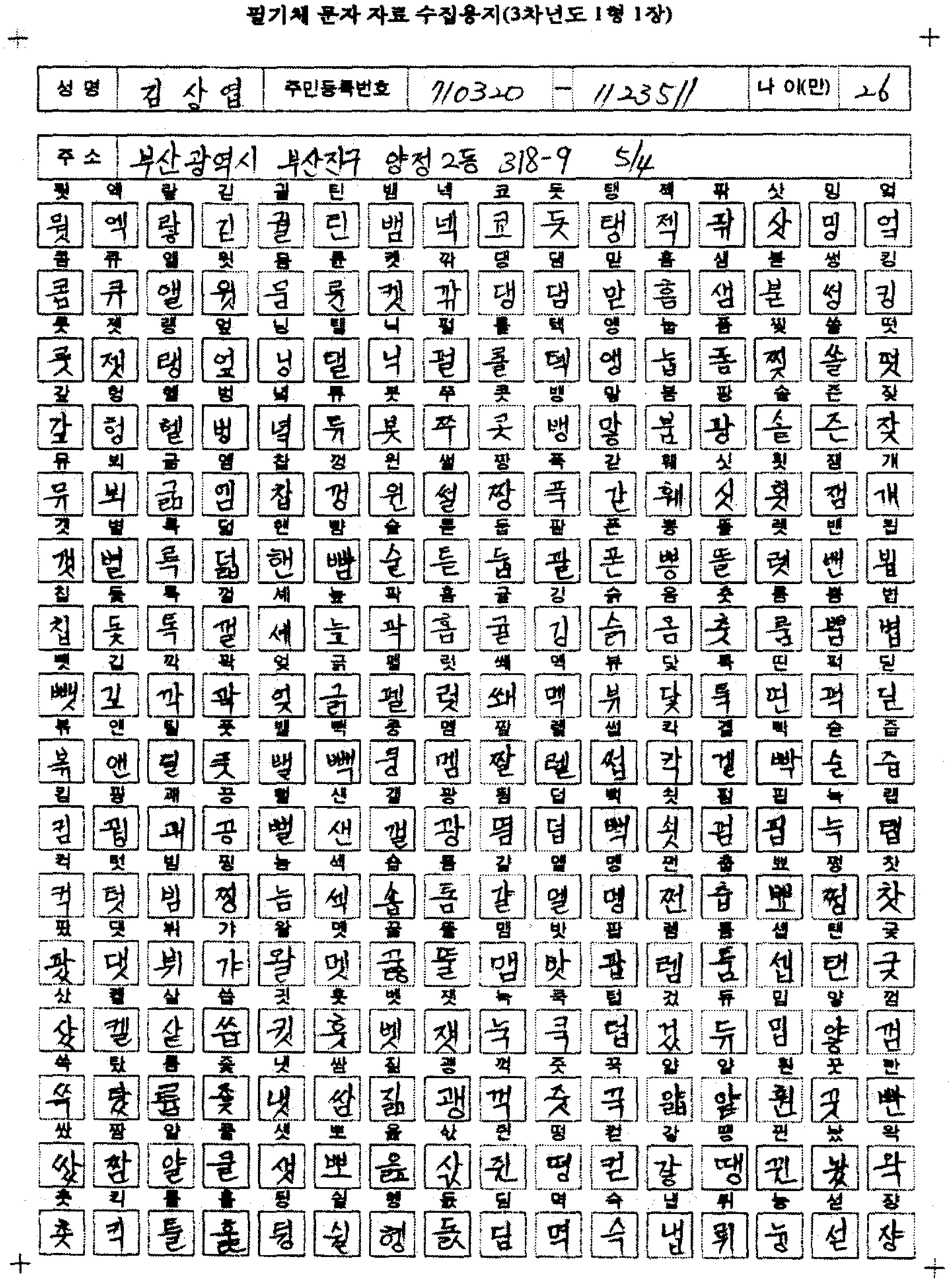


그림 3.2 필기된 수집 용지의 예

1 절. 필기자

필기자는 1, 2 차년도와 다른 지역을 선정하고 다양한 계층의 연령, 성별 분포를 고려하여 1,000 명 이상을 선정하였다. 필기자의 성별 분포, 연령별 분포를 그림 3.3, 3.4 에 각각 나타내었다.

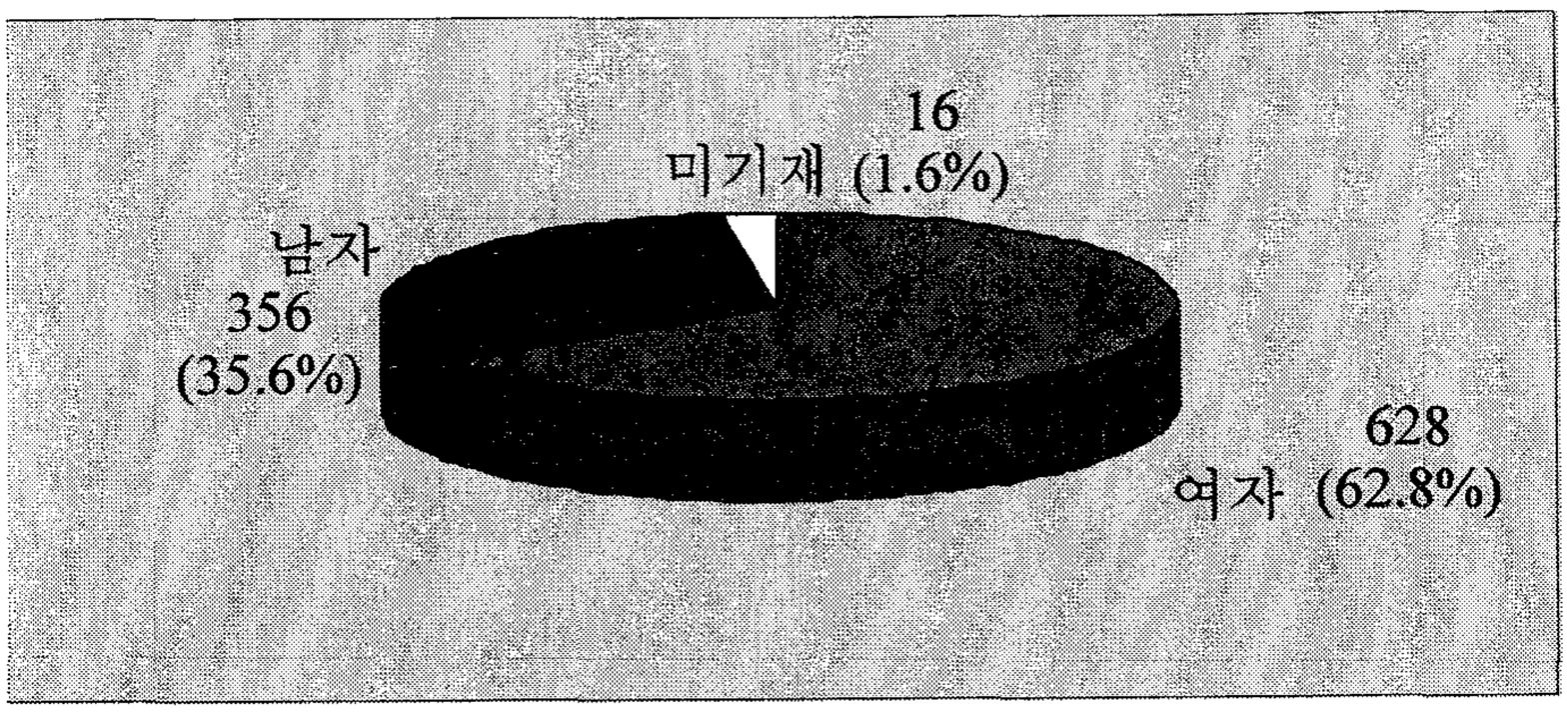


그림 3.3 필기자의 성별 분포

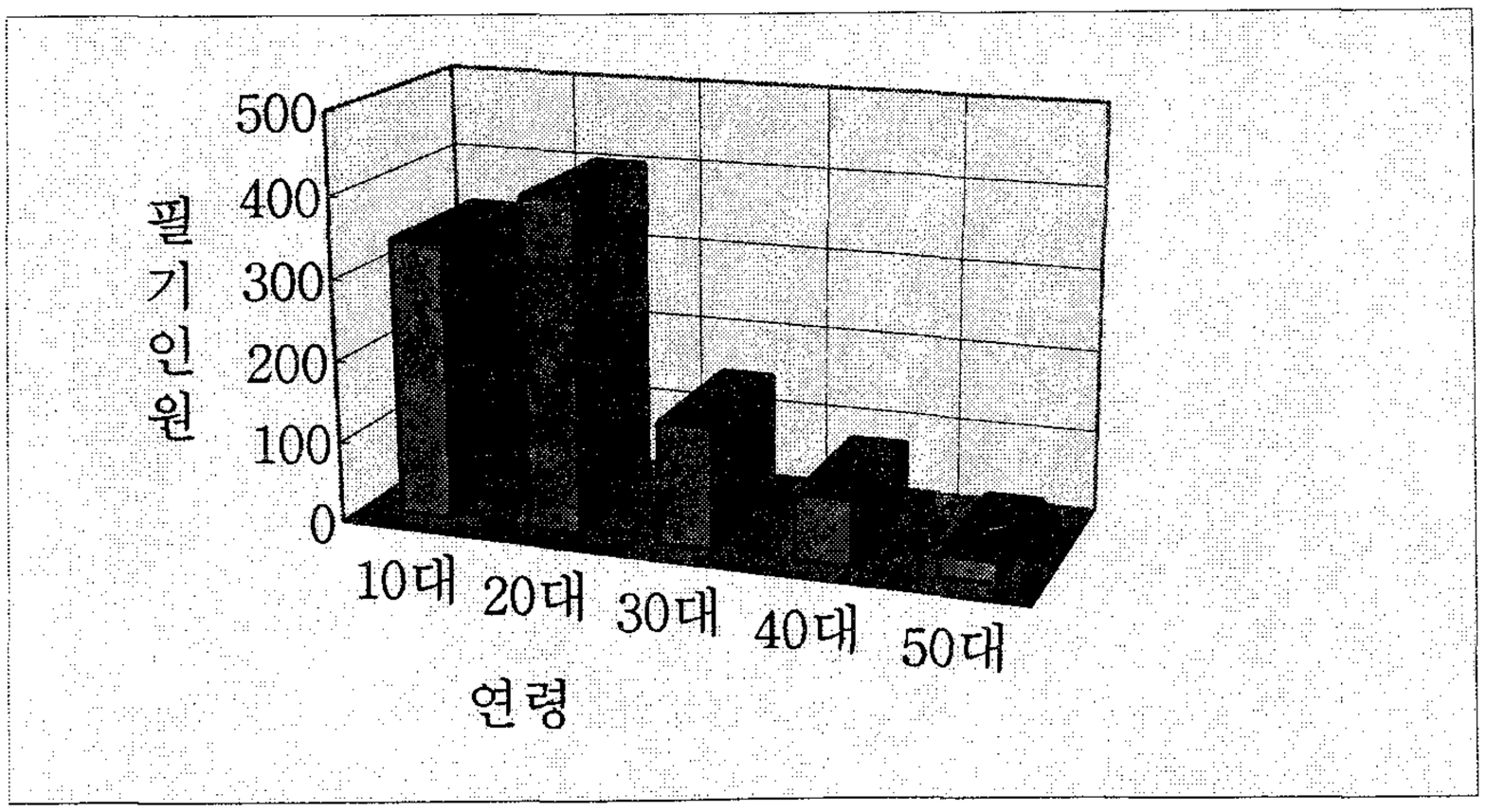


그림 3.4 필기자의 연령별 분포

2 절. 한글 글씨 데이타 수집 용지

한글 글씨 데이타 수집 용지는 2 장에서 설명한 설계 내용을 바탕으로 A4 크기의 백색

용지를 선택하였고 양면 아트지, 건식 복사 용지, 갱지 등 세 종류의 재질을 사용하였다. 그림 3.5 는 한글 글씨 데이타의 수집 시에 사용된 수집 용지의 재질별 분포를 보여 준다.

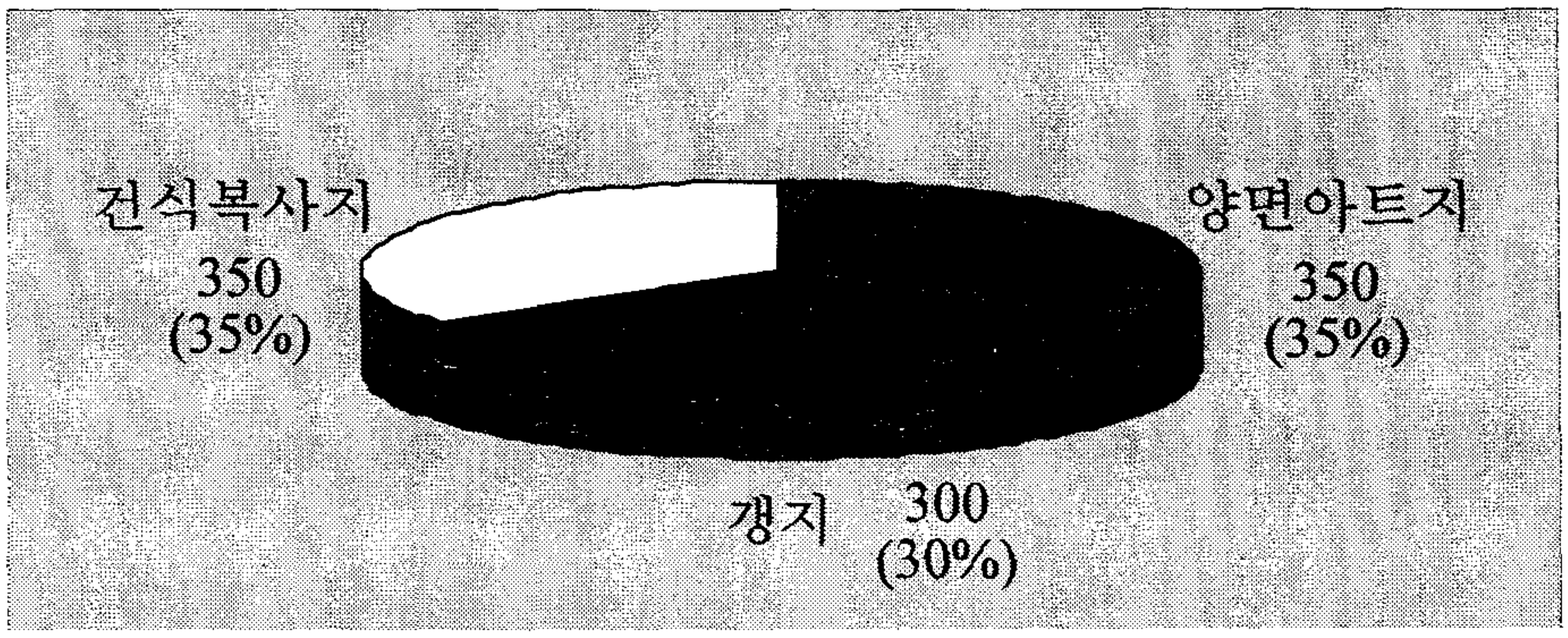
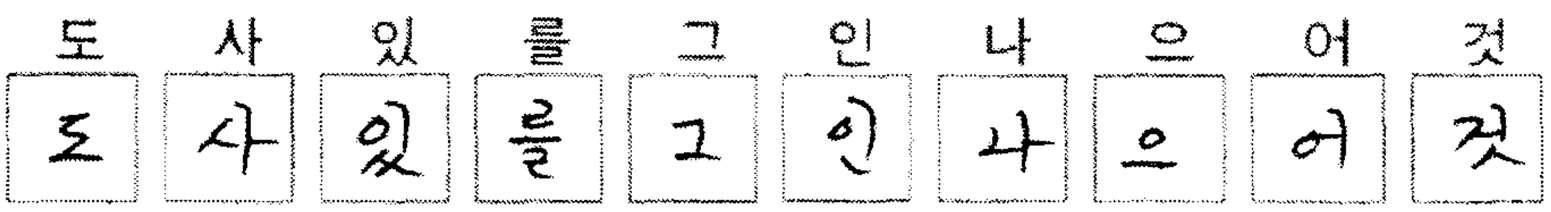
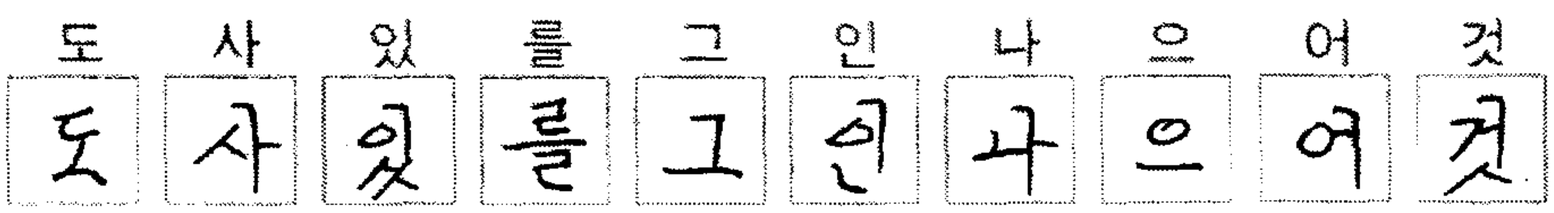


그림 3.5 수집 용지의 재질별 분포

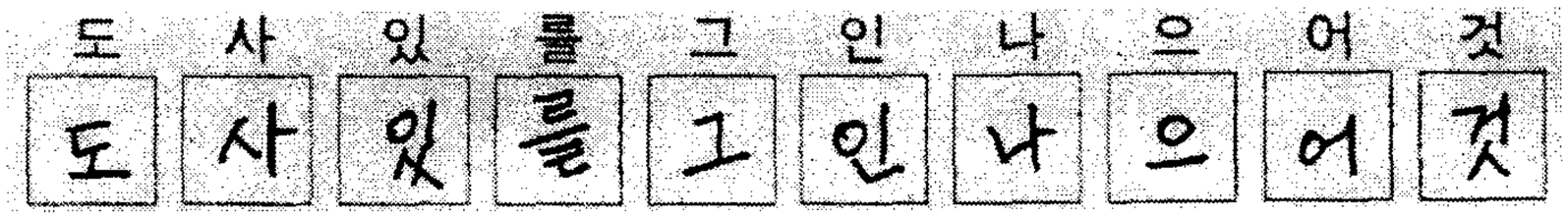
그림 3.6 은 수집된 한글 글씨 데이타 중에서 수집 용지의 재질 별로 스캔된 문자 영상의 상태를 나타낸다. 그림에서 보는 바와 같이 종이의 재질이 갱지인 경우 명도 영상으로 스캔시 배경이 엷게 포함됨을 보여준다.



(a) 양면 아트지



(b) 건식 복사 용지



(c) 갱지

그림 3.6 필기된 수집 용지의 재질별 명도 영상의 예

또한 수집 용지의 설계시에 필기자의 인적 사항으로 성명, 주민등록번호, 나이, 주소 등을 필기하도록 하였다 이러한 자료들은 연속 필기 유형의 데이타로 활용이 가능하다. 그림 3.7 은 수집된 용지중에서 사용자의 인적 사항이 필기된 연속 필기 데이타의 예를 보여준다.

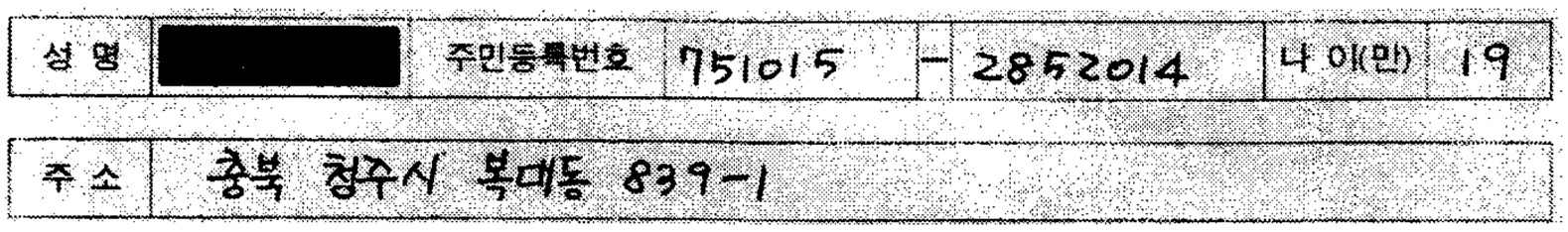


그림 3.7 연속 필기된 필기자 인적사항의 예

3 절. 필기 도구

한글 글씨를 필기하는 데는 일상 생활에서 일반적으로 많이 사용하는 볼펜, 싸인펜, 수성 마킹펜 등 세 종류의 필기 도구로 제한하였고, 색상은 흑색으로 정하였다. 필기시에 사용된 필기 도구는 일괄적으로 필기자에게 제공하여 필기하도록 함으로써 필기 문자의 품질을 균일하게 하도록 유도하였고 각 필기 도구의 종류별 분포를 균일하게 유지하였다. 사용된 필기 도구별 분포는 그림 3.8 과 같다.

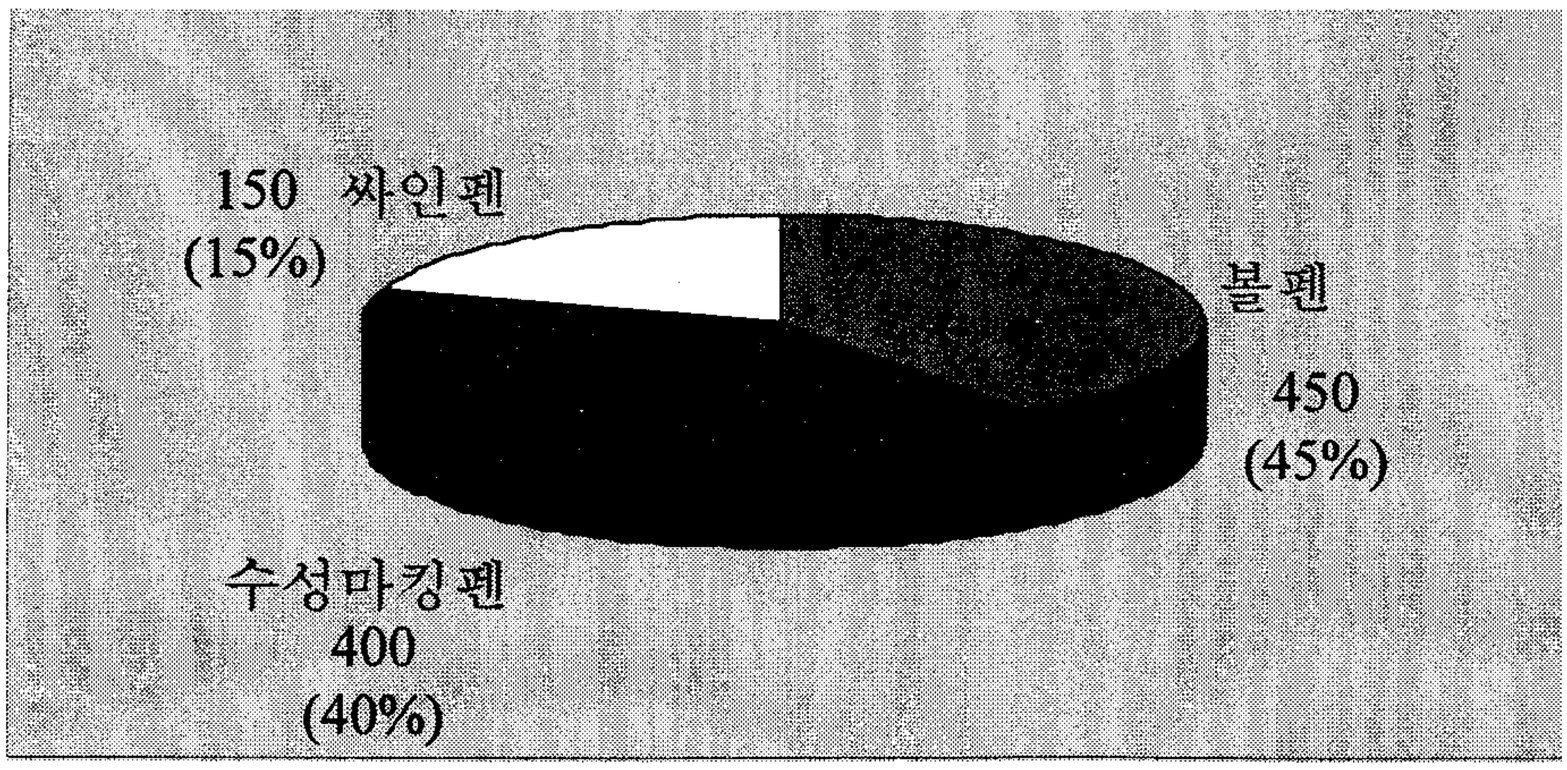


그림 3.8 수집 용지의 재질별 분포

또한 필기시에는 수집 용지에 처음 부터 끝까지 필기하는데 있어서 가능한한 필기 문자의 선폭이 균일하게 필기하도록 유도하였다. 그림 3.9 는 필기시에 사용된 필기 도구별 필기 결과를 보여 준다.

(a) 볼펜

(b) 싸인펜

(c) 수성 마킹펜

그림 3.9 필기 도구별 필기 결과의 예

4 장. 3 차년도 한글 글씨 데이타 수집 용지의 스캔 및 저장

1 절. 서론

한글 글씨 데이타베이스를 구축하기 위해서는 불특정 다수에 의해 필기된 글씨 데이타가 필요하다. 그런데 한글 글씨 데이타베이스를 구축함에 있어서 가장 중요한 과정 중의 하나는 필기된 수집 용지내에 존재하는 문자 단위 영상을 분할하는 데에 있다. 문자 분할을 보다 쉽게 하기 위한 방법으로는 수집 용지에 한 문자를 필기할 수 있는 영역을 미리 그려주는 방법을 들 수 있는데 본 연구 개발에서는 수집 용지에 붉은색으로 사각형의 필기 영역을 표시하여 문자 분할의 효율을 높이고자 하였다.

2 절. 스캔 및 저장

수집된 용지내의 문자 영상 데이타는 1 차년도와 마찬가지로 명도 영상으로 스캐너를 통하여 입력되어 문자 단위의 분할 과정을 거쳐 일련의 구조를 갖는 화일로 저장된다.

그러나 명도 영상으로 스캔된 수집 용지는 붉은색으로 인쇄된 부분이 어느 정도의 명도값을 가지게 되어 문자 단위 분할에 많은 문제점을 야기시킨다.

명도 영상에서의 문자 분할을 어렵게 만드는 경우는 다음과 같이 분류할 수 있다.

- 필기 칸에 접촉 되거나 벗어나게 필기된 경우
- 수집용지의 재질이 갱지인 경우
- 스캔된 영상이 기울어진 경우
- 잘못 필기된 경우

그림 4.1 에서는 필기칸에 접촉되거나 벗어나게 필기된 예를 보여준다. 이 그림에서는 '가' '어' '국' '전' 자가 필기칸에 접촉되어 있는 경우에 해당하며 '과' '라' '부' '여' '주' 자는 필기칸을 벗어난 경우에 해당한다. 이러한 경우에는 문자 분할이 매우 어렵게 되는데 이러한 경우에 대한 처리 방법으로 필기 영역 외부에 존재하는 획들은 저장하지 않는 방법과 외부의 획들까지 모두 저장한 다음 수작업으로 박스 부분을 제거하는 방법을 고려할 수 있다.

그림 4.1 필기 칸에 접촉되거나 벗어나게 필기된 경우의 예

그림 4.2 는 수집 용지의 재질이 갱지인 경우의 필기된 수집 용지의 예를 보여 주는데 여기서 바탕에 엷게 깔린 명도값들을 확인할 수 있다. 또한 갱지의 특성상 바탕 부분의 명도값이 일정하지 않기 때문에 문자 단위 분할을 더욱 어렵게 한다. 이와 같은 문제를 해결하기 위해서는 문자 단위 분할에 사용되는 여러 매개 변수들의 값을 적당히 조절해야 한다.

그림 4.2 갱지일 경우 스캔된 명도 영상의 예

그림 4.3 은 수집 용지를 스캔 할 때 기울어져 스캔된 경우를 보여준다. 이 경우에는 문자 단위 분할 시에 필기 칸이 문자 영상에 포함되는 선 부분이 잡영으로 첨가되어 이를 제거해야 하며 이와 함께 이러한 기울어진 영상으로 인하여 분할 영역이 문자 영상을 모두 포함하지 못하는 경우가 생길 수 있다. 특히 이러한 경우는 필기된 수집 용지내의 글자들의 크기가 클 때 어려움이 더하게 된다. 이러한 문제점을 해결하기 위해서는 분할 영역의 크기를 좀더 크게 하고 분할 영역내에 존재하는 필기 칸들을 효과적으로 제거하는 방법이 고려되어야 한다.

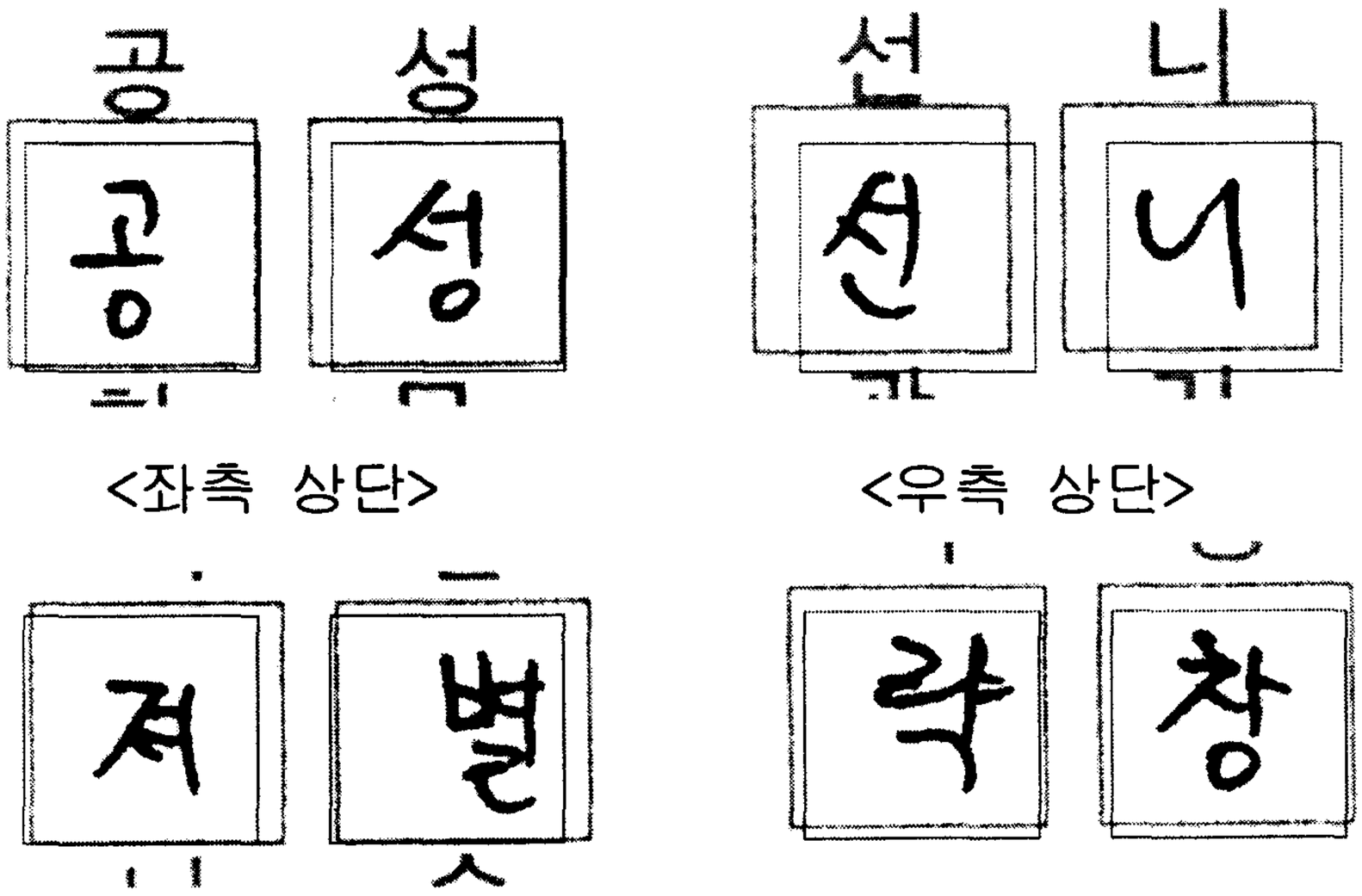


그림 4.3 스캔된 영상이 기울어진 경우의 예

그림 4.4 는 필기시에 필기자의 부주의로 인한 예시 문자와는 다른 문자를 필기하고 이를 수정하기 위하여 X 표로 표시한 경우를 보여주고 있다. 이 경우에는 수집 용지의 여분 필기 칸에 다시 필기된 문자를 대치시켜야 하므로 복잡한 처리 과정을 필요로 하게 된다.

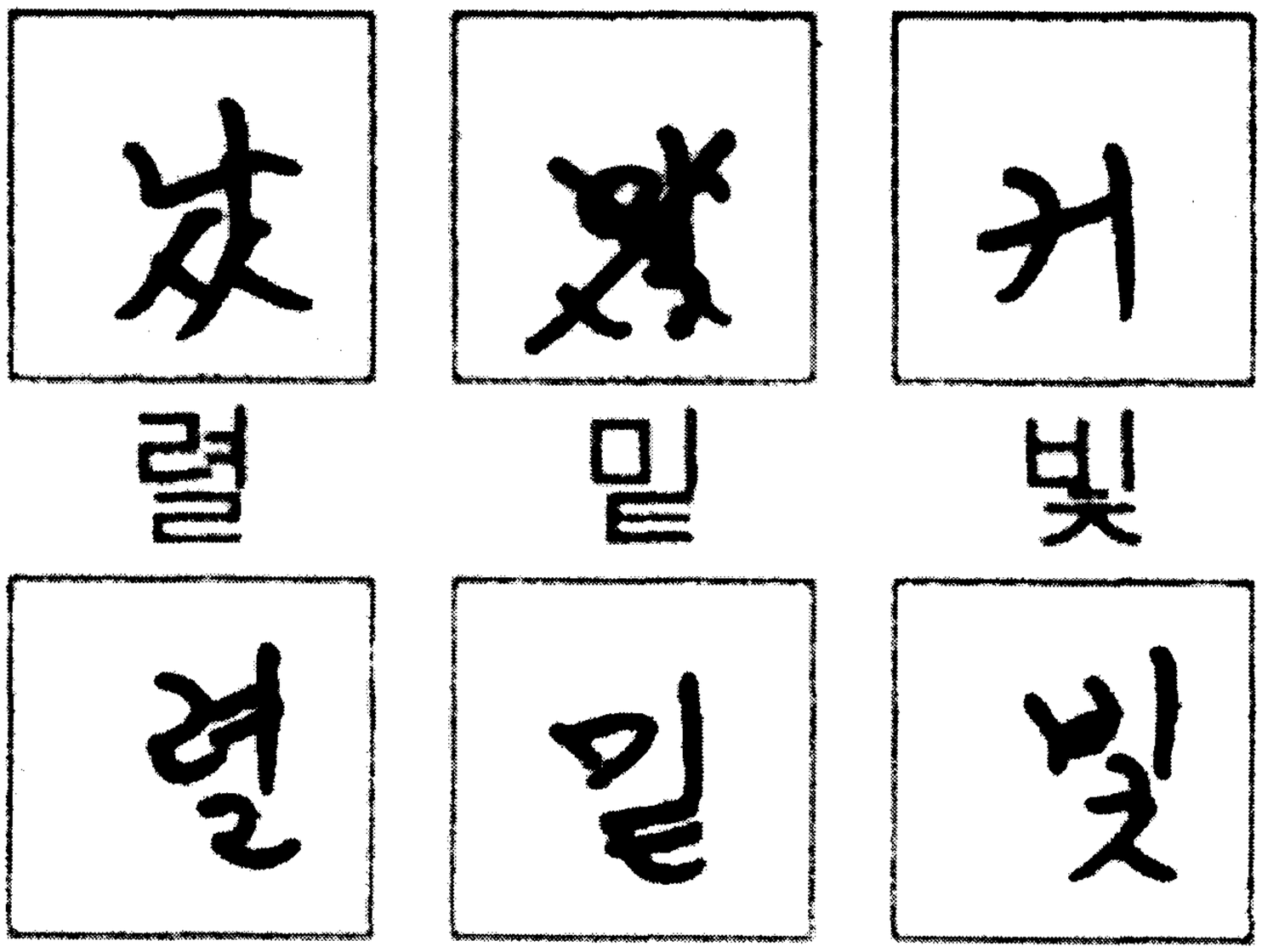


그림 4.4 필기자의 부주의로 인하여 잘못 필기된 문자 영상의 예

또한 방대한 문자 영상 데이타를 다루기 위해서는 수집자로 하여금 보다 쉽게 문자 영상을 저장할 수 있도록 최대한 자동화시켜야 하며 이와 함께 잘못 분할된 문자 영상을 쉽게 교체할 수 있는 문자 분할 시스템이 필요하게 된다. 따라서 본 과제에서는 한글 글씨 데이타베이스 구축을 위한 효과적인 문자 분할 시스템을 구현하여 과제를 수행하였다.

5 장. 데이타베이스 검증 및 보완

1 절. 데이타베이스에 대한 검증 및 보완의 필요성

한글 글씨 데이타 수집 용지는 스캐너를 통하여 입력되고 문자 분할 과정을 거쳐 낱자 단위의 문자 영상 데이타로 저장된다. 그러나 문자 분할 시스템을 통하여 얻은 문자 영상 데이타들이 수집 용지의 필기될 영역을 나타내는 사각형을 제거하고, 필기자의 실수로 인하여 잘못된 문자들을 교체하였다 하더라도 저장된 문자 영상 데이타들에는 많은 문제들을 있기 때문에 단지 문자 분할 과정을 거친 문자 영상 데이타들을 직접 문자 인식 연구에 사용하기에는 많은 어려움을 갖는다. 예를 들면, 수집된 용지를 스캔하여 저장하는 과정에서 실수로 데이타베이스의 헤더 정보를 잘못 입력하였거나, 필기된 문자가 필기될 영역을 나타내는 사각형에 접촉된 경우 문자 영상만을 정확하게 분리할 수 없기 때문에 분할 오류가 발생한다. 또한 필기자가 두개의 서로 다른 문자를 습관적으로 유사하게 필기하거나 또는 전혀 다른 문자로 필기하였을 경우, 문자 영상 데이타와 레이블된 코드가 서로 맞지않는 레이블링 오류가 발생하게 된다. 따라서 이러한 단점을 극복하기 위하여 문자 분할 시스템을 통하여 저장된 문자 영상 데이타를 검증하고 보완함으로써 데이타베이스의 품질을 향상시키는 것이 필요하다.

특히, 3 차년도에는 완성될 데이타베이스의 효율적인 사용을 도모하기 위하여 이미 구축된 데이타베이스내의 화일뿐 아니라 당해년도에 처리된 화일중 레이블링 오류 및 분할 오류로 데이타로서의 가치가 없다고 판단되는 문자에 대해 별도의 처리 과정을 두어 재편집함으로써 데이타베이스의 품질을 향상시키고자 하였다.

2 절. 오류의 형태

구현된 데이타베이스 검증 및 보완 시스템에서 처리하는 오류의 형태는 다음과 같다.

- 헤더 정보 오류
  - 문자 영상을 저장하고 있는 화일의 헤더 정보가 잘못된 경우로서 데이타 스캔 과정에서 발생한다.
- 문자 단위 분할 오류
  - 문자 영상의 일부가 잘려서 저장된 경우로서 문자 분할 시스템의 임계값이 스캔

된 문자 영상에 적합하지 않을 경우 발생한다.

- 잡영 또는 필기될 영역을 나타내는 사각형의 일부가 남아 불필요한 공백이 들어간 경우 문자 분할 시스템은 이러한 부분도 문자 영상으로 간주하여 데이타베이스에 저장한다.

● 레이블링 오류

- 필기자가 두 개의 서로 다른 문자를 습관적으로 유사하게 필기한 경우 발생한다.

- 필기자의 실수로 전혀 다른 문자로 필기한 경우 발생한다.

1. 헤더 정보 오류

헤더 정보 오류는 저장된 데이타베이스 화일의 헤더에 잘못된 정보가 입력된 경우로서 그림 5.1 에서 볼 수 있는 바와 같이 필기한 손에 대한 정보에 오류가 발생하였을 경우와 같이 헤더에 전혀 다른 정보가 입력되었을 경우, 초기 화면의 헤더 정보를 출력하는 부분에 '오류가 발생하였습니다'라는 메세지가 출력된다. 헤더 정보 오류가 발생하였을 경우 초기 화면의 하단부에 있는 헤더 정보 오류 메뉴를 선택하여 헤더 정보 오류가 발생하였음을 표시한다.

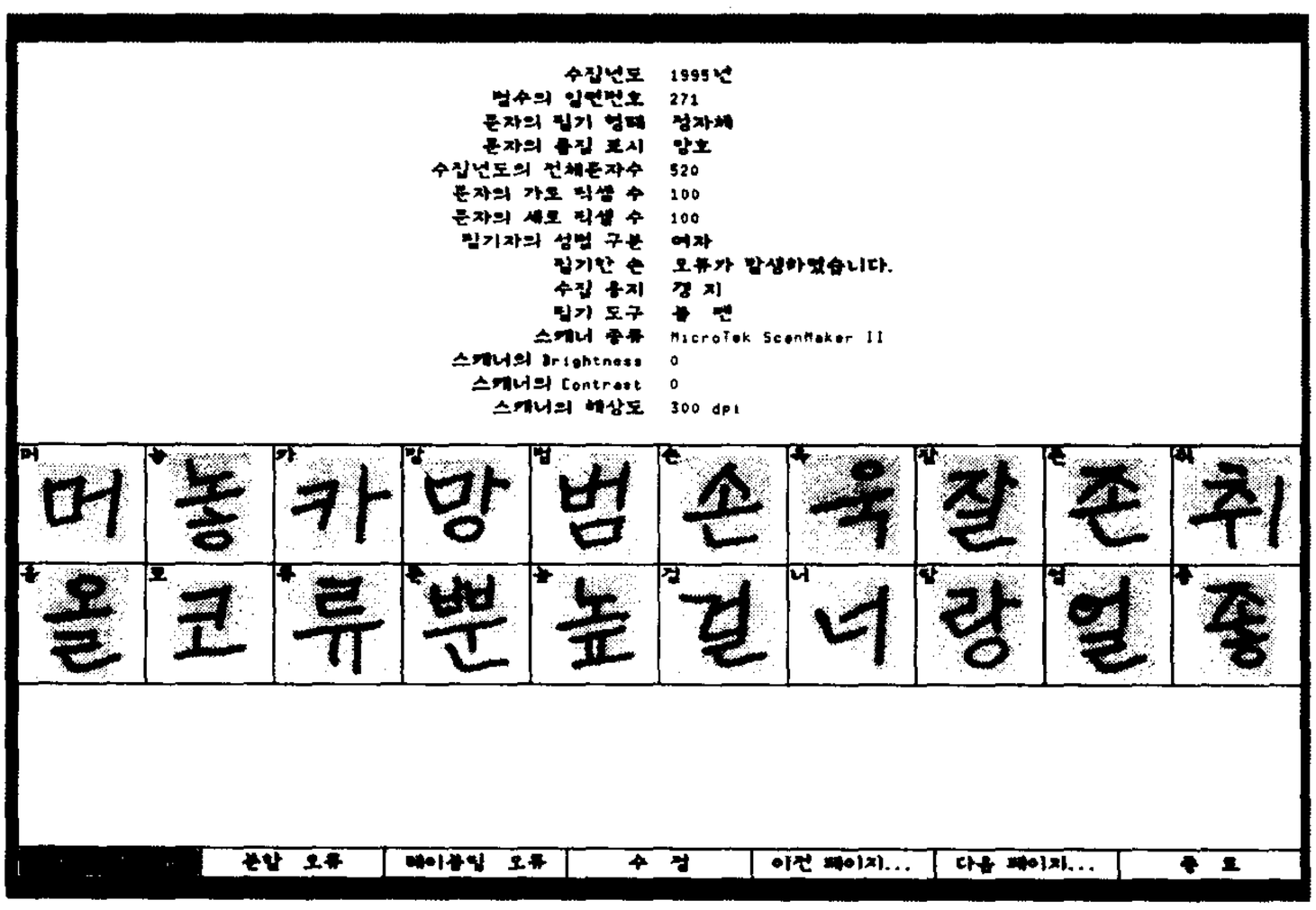


그림 5.1 헤더 정보 오류의 예

2. 분할 오류

문자 단위 분할 오류는 문자 분할 시스템에서 문자 영상을 저장할 때, 문자 영상의 일부분이 잘려서 저장되거나, 또는 잡영에 의해 문자 데이타 영상에 불필요한 영상이 삽입된 경우이다.

(1) 문자 영상의 일부가 잘린 경우

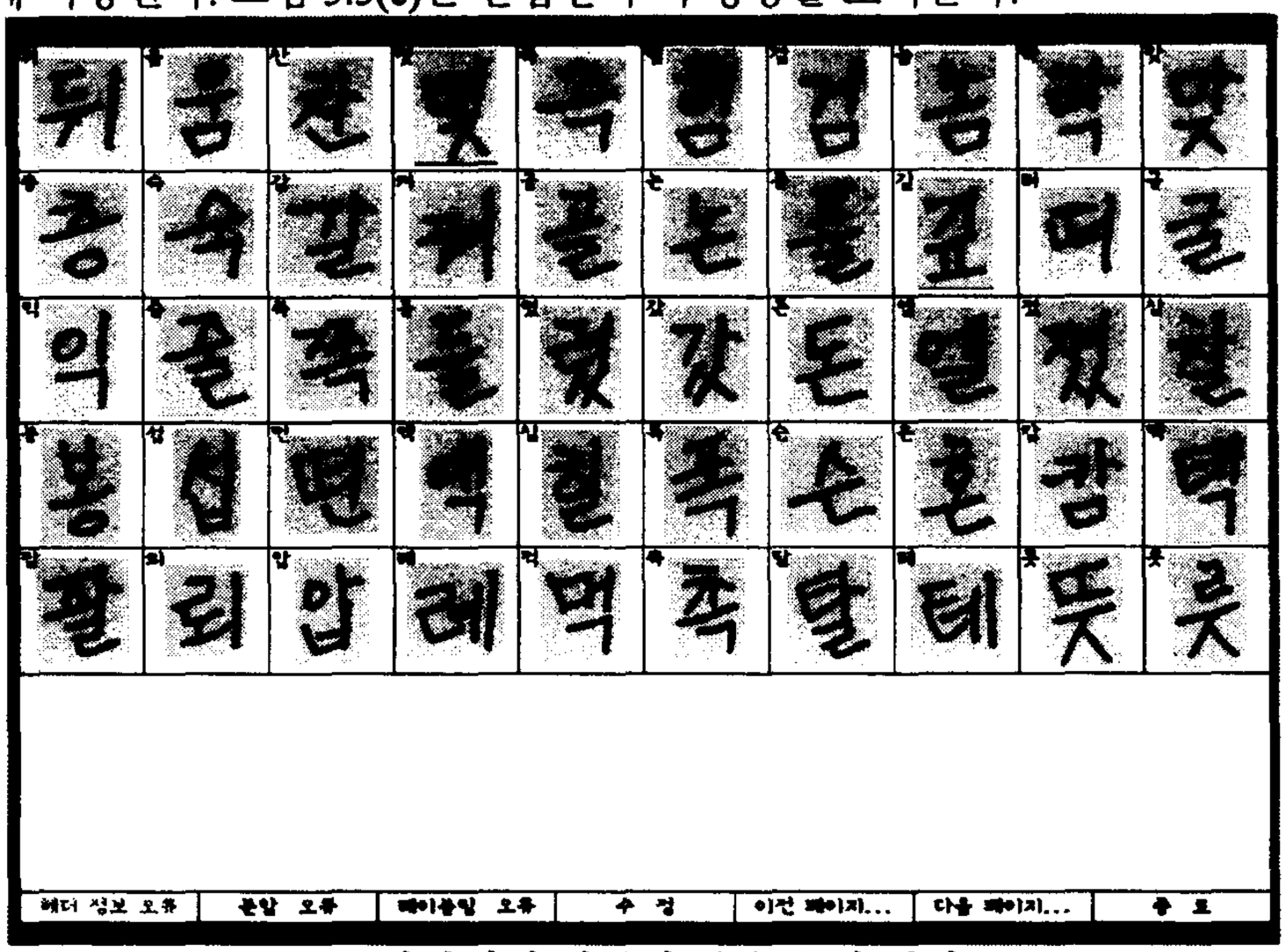
문자 분할 시스템은 문자 영상의 명도값과 필기될 영역을 나타내는 사각형의 명도값에 대한 임계값을 이용하여 문자를 분할하는데, 설정된 임계값이 부적당한 경우 그림 5.2에서 보는 바와 같이 '총', '렇', '창' 자의 경우와 같이 문자 영상의 일부가 잘려 저장된다.

그림 5.2 문자가 잘린 경우의 예

(2) 잡영 등이 삽입된 경우

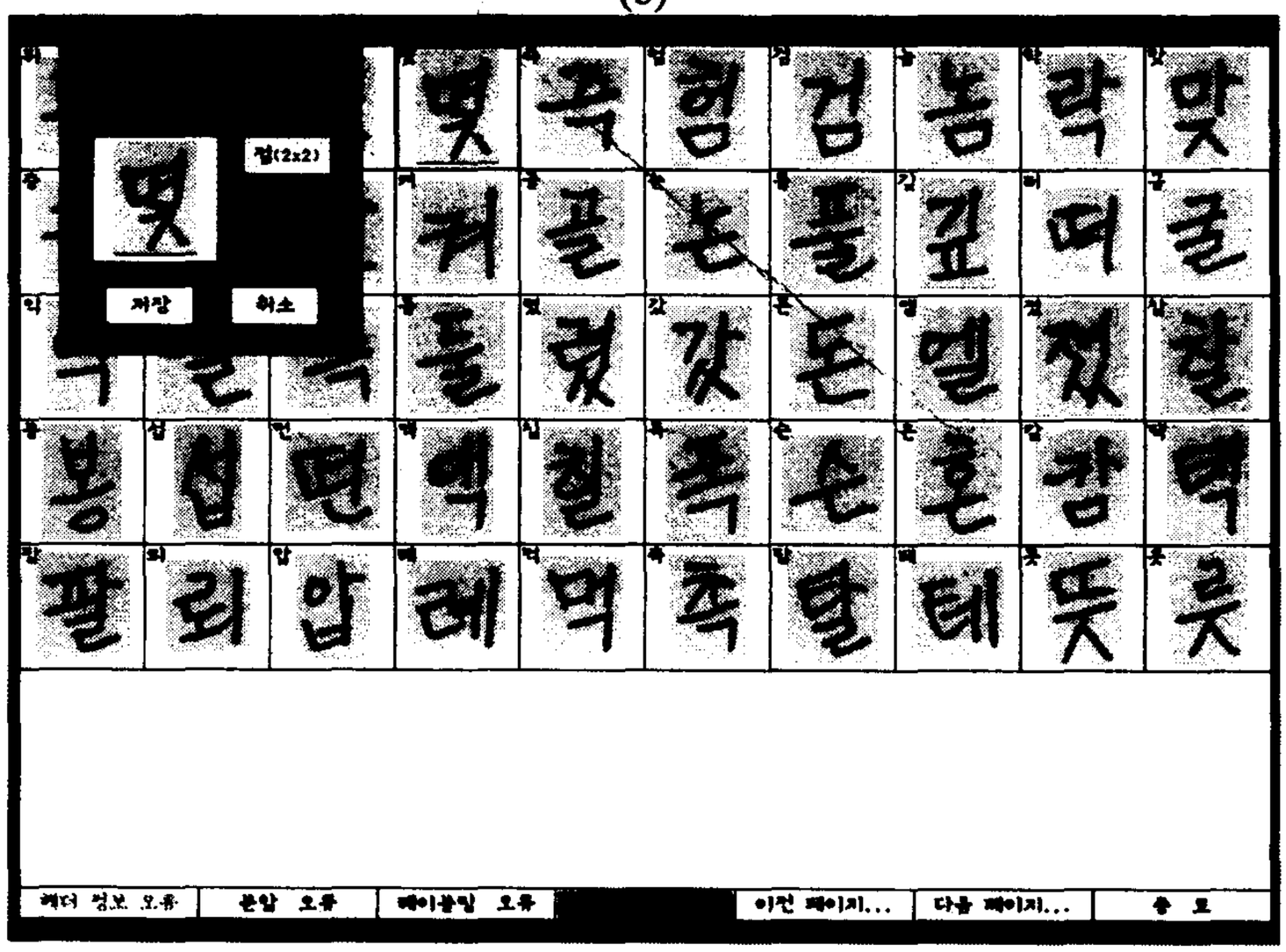
문자 영상의 명도값과 임계값을 비교할 때 스캔된 수집 용지에 남아있는 잡영이나 필기될 영역을 나타내는 사각형의 명도값이 문자 영상의 명도값과 유사할 경우, 그림 5.3(a)의 '몇'자와 같이 잡영이나 필기 영역을 나타내는 사각형의 일부가 문자 영상으로 간주되어 저장 된다. 이러한 잡영이나 사각형의 일부가 문자 영상에 삽입된 경우 화면

하단부의 수정 메뉴를 선택한 후, 수정할 문자 영상을 선택하면 그림 5.3(b)와 같은 편집 화면이 나타난다. 편집 화면에서 불필요한 영역을 지운 후 저장을 선택하면 편집된 문자 영상에 저장된다. 그림 5.3(c)는 편집한 후의 영상을 보여준다.

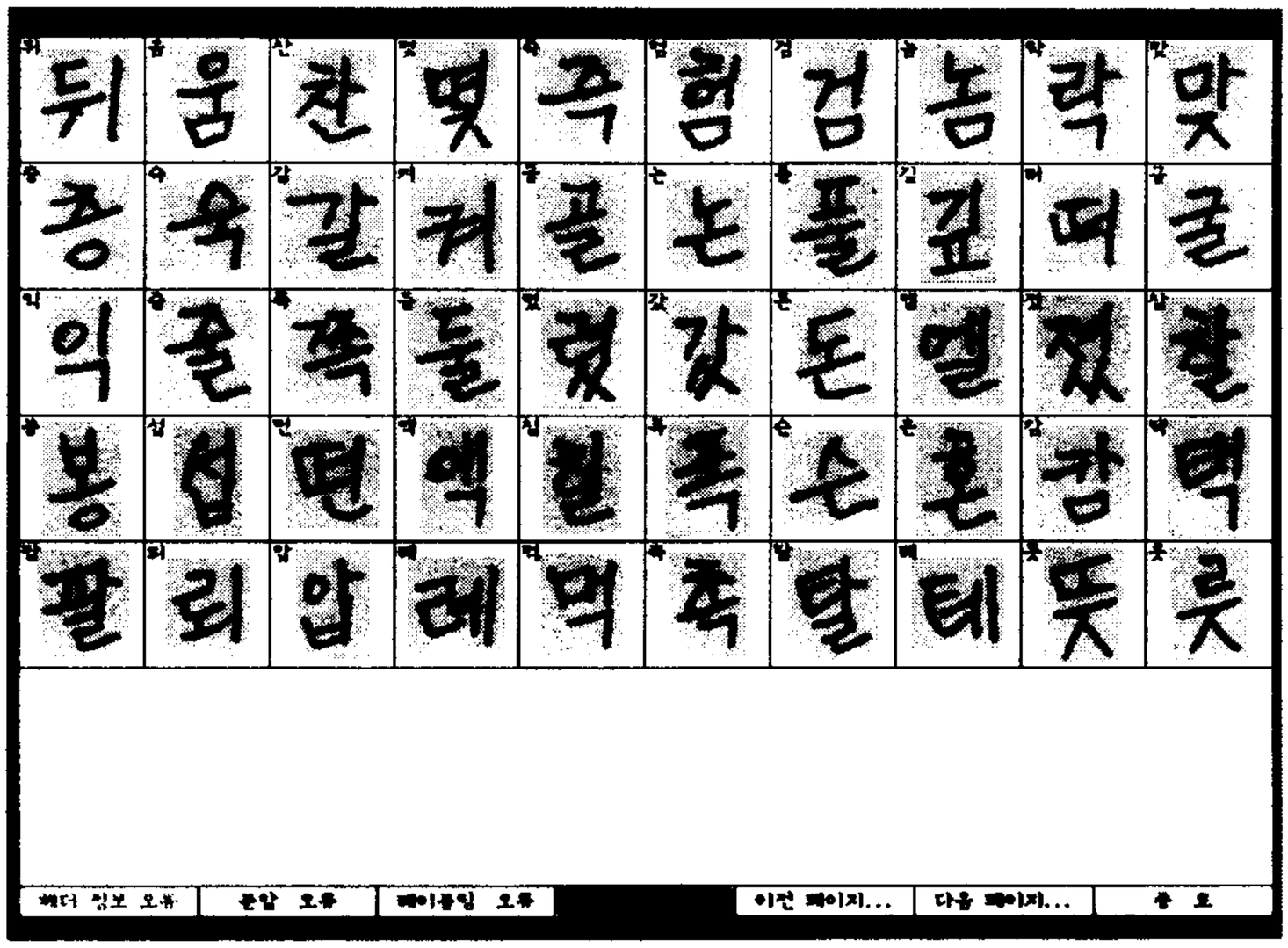


(a) 사각형의 일부가 남은 문자 영상

(b)



(b) 편집 화면



(c) 편집 후의 문자 영상

그림 5.3 잡영 등이 삽입된 경우의 예

(3) 레이블링 오류

레이블링 오류는 그림 5.4 에서와 같이 필기자의 실수로 전혀 다른 문자로 필기하였거나, 필기자가 습관적으로 전혀 다른 문자와 매우 유사하게 필기함으로서 발생한다. 그림 5.4(a)에서는 예시 문자가 '답'자를 필기자의 실수로 'ㄷㅏㅂ'자로 필기한 경우의 예를 보여주며, 그림 5.4(b)에서는 예시 문자 '랄'자의 경우 필기자의 평소 필기 습관으로 '달'자와 매우 유사하게 필기하는 경우의 예를 보여주는데, 그림 5.4(b)와 같이 필기자의 필기 습관에 의한 경우는 수정을 가급적 피함으로써 다양한 필기 유형의 범주를 포함시켰다.

(a) 필기자의 실수에 의한 경우

(b) 필기자의 필기 습관에 의한 경우

그림 5.4 레이블링 오류의 예

3 절. 오류의 처리 방법

문자 영상 데이타 검증 과정에서 헤더 정보 오류는 수정이 쉽게 이루어질 수 있으나, 분할 오류 및 레이블링 오류의 경우는 그렇지 못하다. 분할 오류의 경우에 있어서 잡영 등이 삽입된 경우에는 그림 5.3 과 같은 편집 과정을 거쳐서 수정이 가능하지만 그림 5.2 와 같이 문자 영상의 일부가 잘린 경우에는 수정이 불가능하다. 또한, 그림 5.4 와 같이 필기자의 실수에 의한 경우나 필기자의 필기 습관에 의해서 대부분 발생하는 레이블링 오류 역시 편집 작업을 통한 수정이 불가능하기 때문에 별도의 처리 과정이 필요하다.

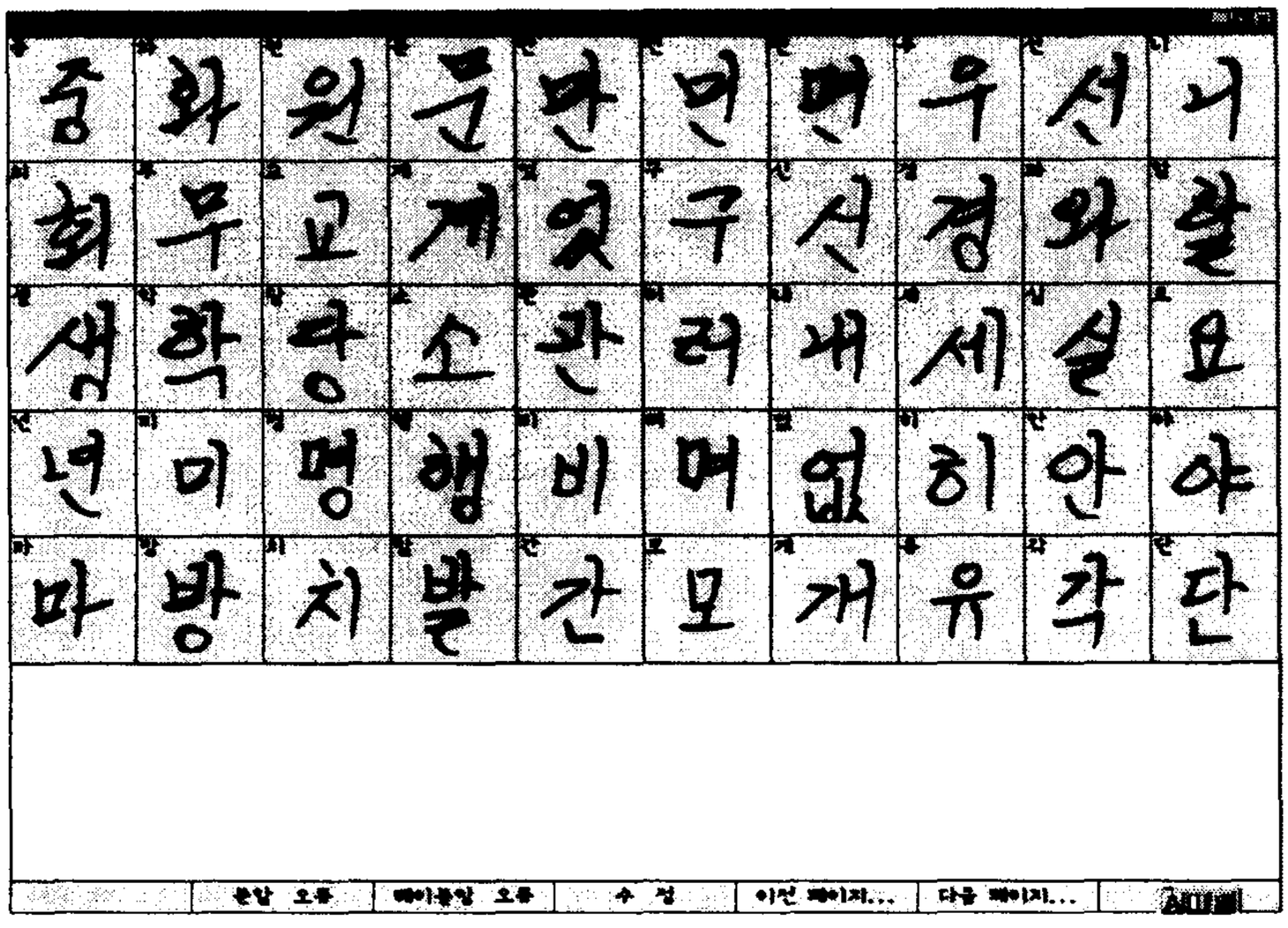
본 연구에서는 이들 오류 데이타를 데이타베이스 내에 그대로 저장해 두기 보다는 올바른 문자만을 데이타베이스에 저장해 둠으로써 데이타베이스의 품질을 유지함은 물론 문자 인식 시스템 개발자들이 잘못된 문자에 대한 훈련이나 인식을 통하여 인식 시스템의 성능을 저하시키는 것을 방지하고자 하였다. 이를 위해 본 연구에서는 오류 부분에 해당하는 문자 영상을 공백으로 처리하였다.

또한, 이들 오류 화일에 대한 정보를 제공함으로써 데이타베이스 사용자가 오류 영상을 제외한 올바른 문자 영상만을 사용할 수 있도록 하였다. 데이타베이스에 대한 오류 화일의 예가 다음과 같다.

화일명	오류 문자 수	위치 코드			
460	1	506	화		
461	1	20	견		
462	2	211	발	486	할
464	1	239	빨		
465	1	448	킬		

그림 5.5 오류 화일의 예

여기서, 460 은 오류가 발생한 화일명(kle10460) 을 의미하며, 다음에 오는 숫자는 오류가 발생한 문자의 갯수를 의미한다. 그리고 다음의 506 은 오류가 발생한 문자 영상의 인덱스를 나타내며, '화'는 문자 영상에 해당하는 완성형 한글 코드이다.



(a) 공백 처리 전

(b) 공백 처리 후

그림 5.6 오류가 발생한 문자 영상에 대한 공백 처리 과정

6 장. 사용자 인터페이스 보완

본 연구개발에서 구축된 한글 글씨 데이타베이스는 한글 글씨의 인식에 관한 연구를 수행하는 연구자들이 사용하기 편리한 형태로 구성되었다. 구축된 데이타베이스는 새로운 데이타의 삽입 또는 기존 데이타의 삭제 등이 거의 발생하지 않는 특성을 갖는다. 따라서, 사용자들이 필요로 하는 데이타의 검색 뿐만 아니라 데이타베이스에 대한 일반적인 특성들이 이해하기 쉬운 형태로 구성되어야 한다.

본 연구에서는 WWW(World Wide Web)의 HTML(Hyper Text Markup Language)을 이용하여 편리한 사용자 인터페이스를 구축함으로써 본 데이타베이스에 대한 소개 및 제반 정보 뿐 만 아니라 한글 글씨 영상 데이타를 제공할 수 있도록 하였다. 본 장에서는 1, 2 차년도에 구축된 사용자 인터페이스에 대한 보완으로 1, 2 차년도에 미 구축된 검색 기능을 보완하여 KSC 완성형 한글 사용빈도순 상위 1,500 자중 사용자들이 필요로 하는 각 문자에 대한 50 개의 영상 데이타를 보여줌으로써 본 데이타베이스에 대한 품질을 평가할 수 있도록 하였다. 또한 본 데이타베이스에 대한 통계적 특성을 분석하여 제공함으로써 한글 글씨 인식 알고리즘 개발에 유용한 정보를 제공해 주는 기능을 갖도록 하였다. 본 장에서는 이들 두 기능을 실제 WWW 의 화면과 함께 간략히 설명한다.

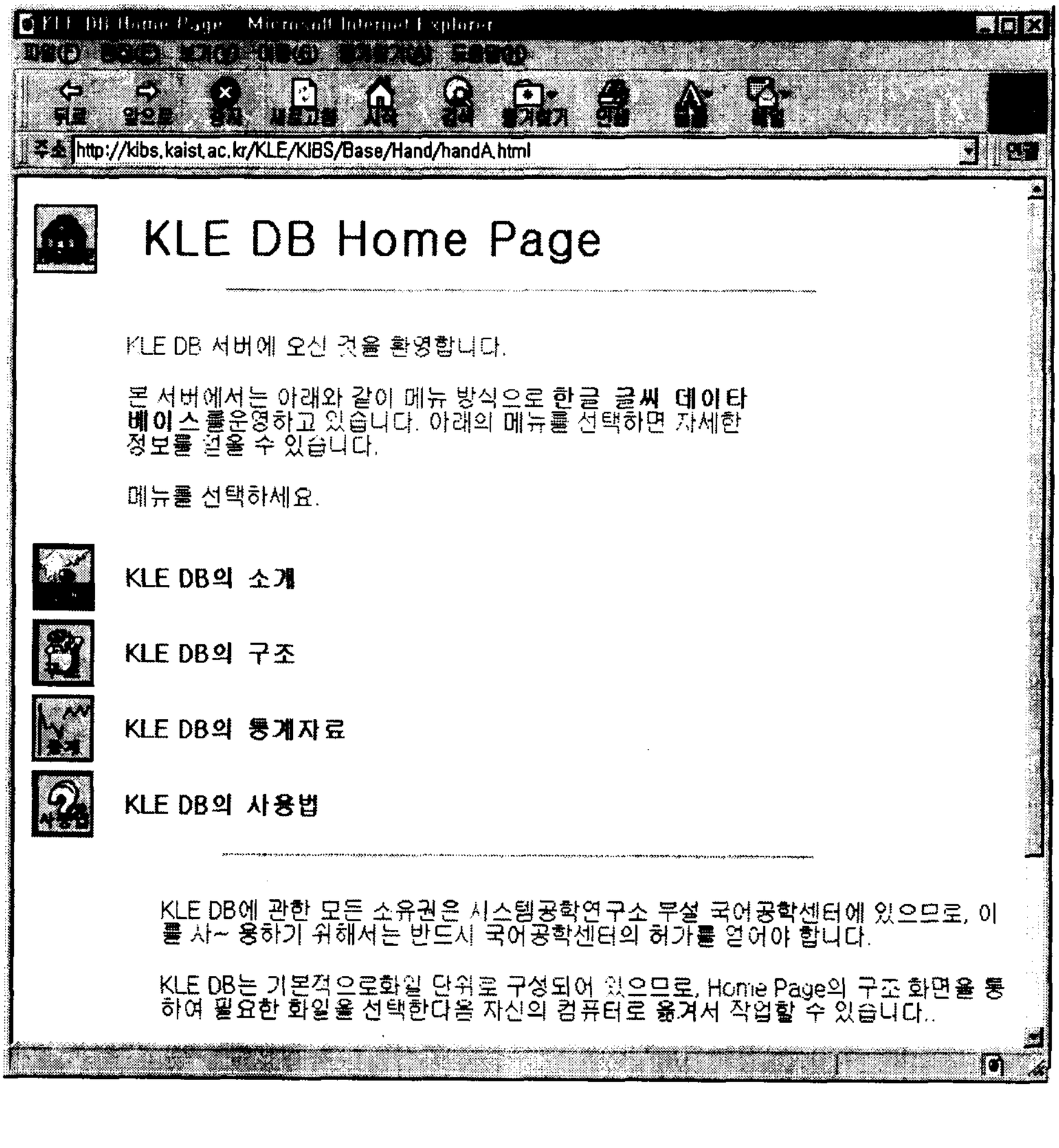


그림 6.1 한글 글씨 데이타베이스 WWW 서버의 홈 페이지

1 절. 문자 영상 검색 기능

한글 글씨 영상의 검색은 그림 6.2 에서 볼 수 있듯이 사용자가 필요로 하는 문자를 입력하게 되면 그림 6.3 과 같이 50 개의 데이타 영상을 보여줌으로써 사용자로 하여금 데이타베이스에 대한 평가를 가능하게 하였다.

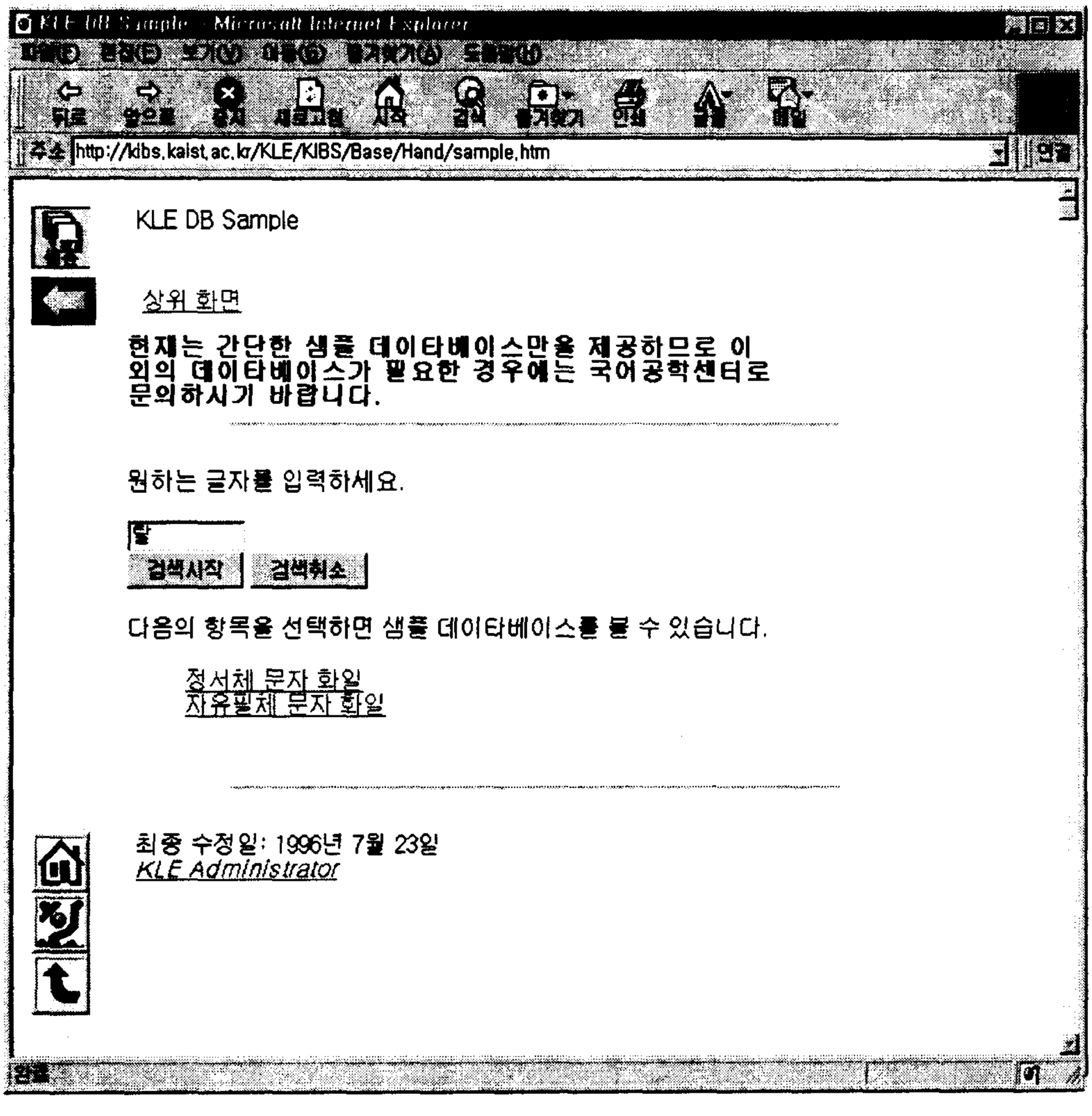


그림 6.2 문자 영상의 검색 기능을 제공하는 페이지

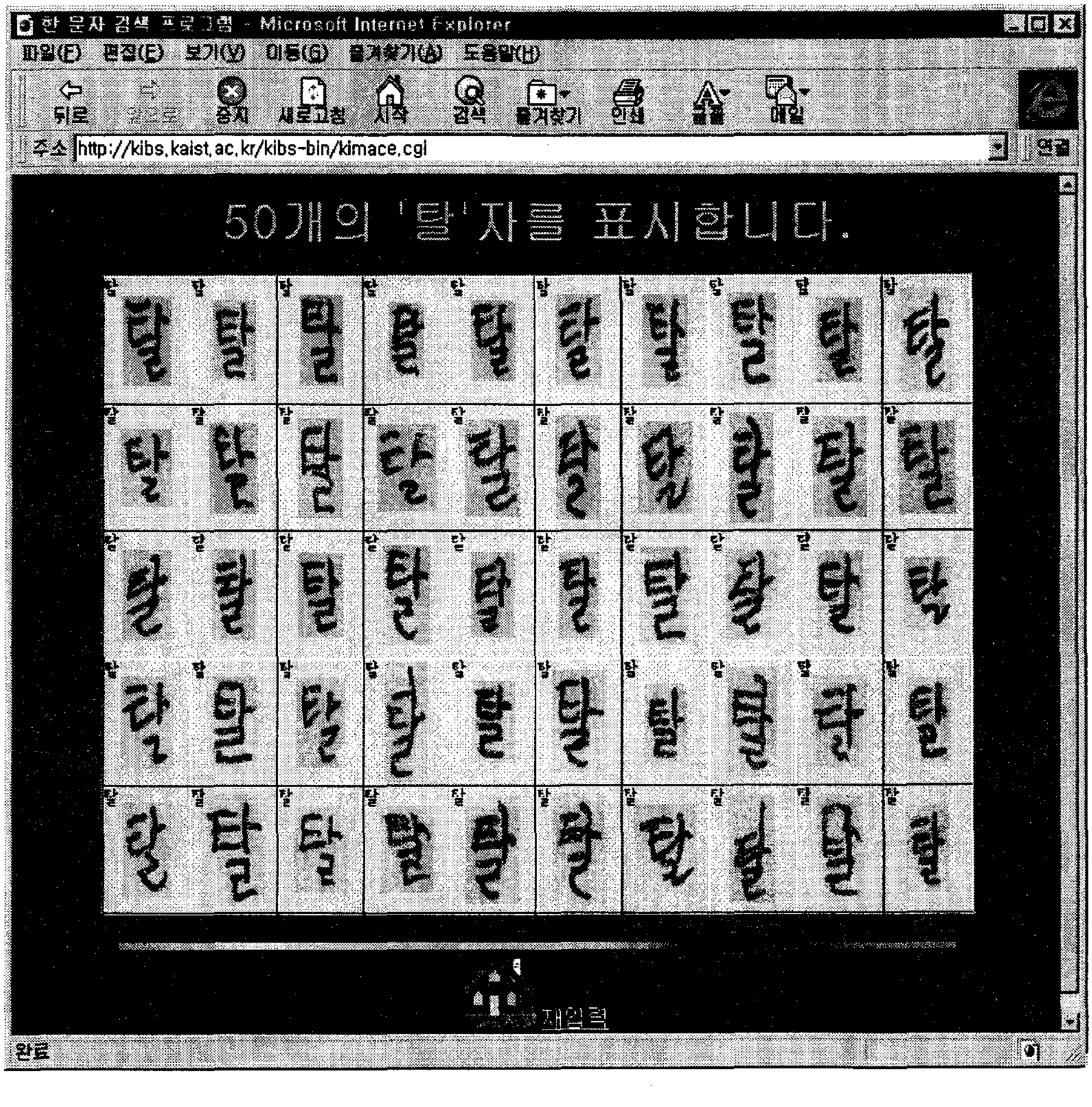


그림 6.3 50 개의 '탈'자에 대한 데이타 영상

이와 더불어 사용자 인터페이스에서는 본 데이타베이스에 대한 통계적 특성을 분석하여 제공함으로써 한글 글씨 인식 알고리즘의 개발에 유용한 정보를 사용자들에게 제공해 주는 기능을 갖는다. 그림 6.4 는 이러한 통계적 특성에 대한 정보를 제공하는 페이지를 보여 주고 있다.

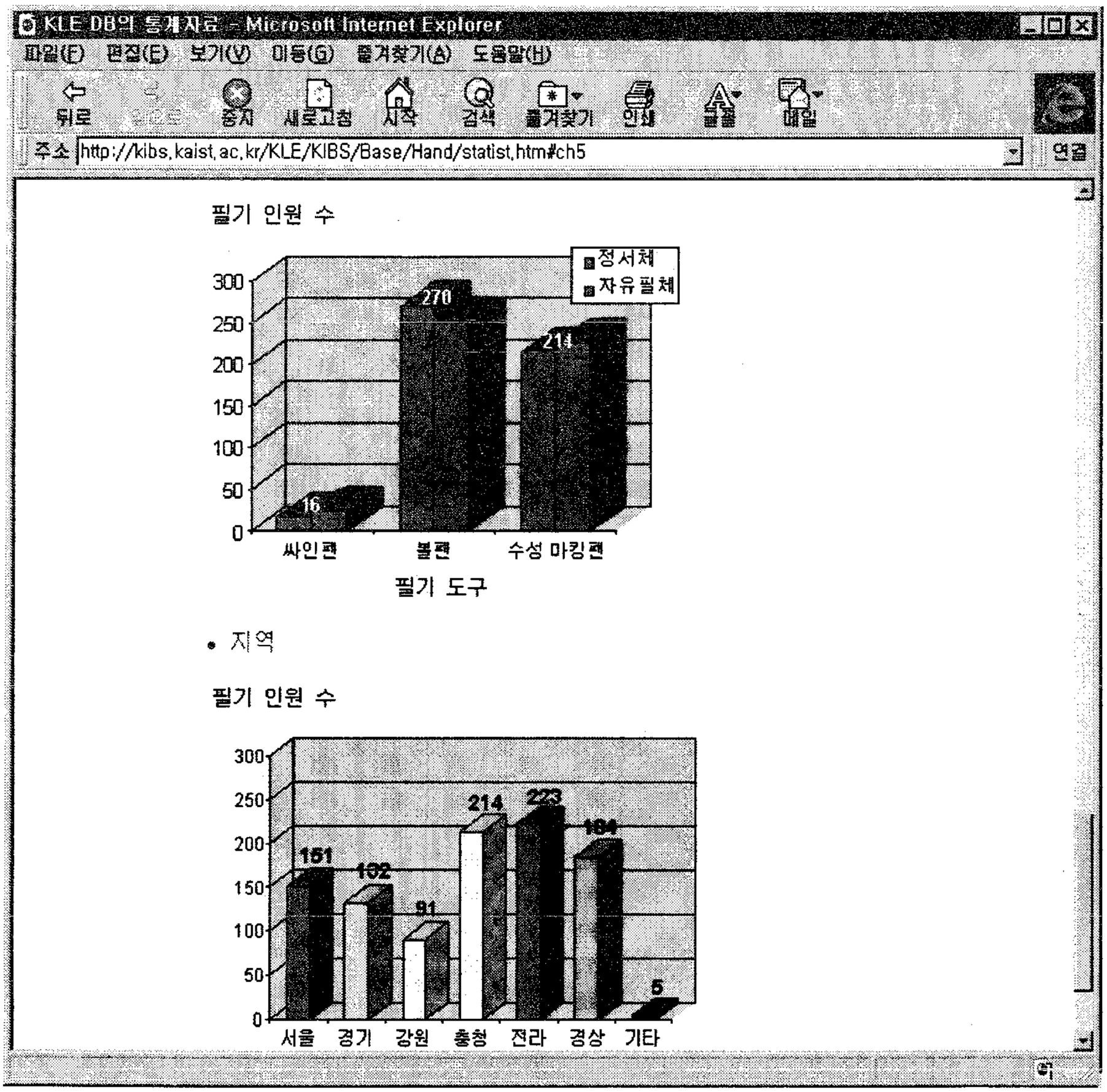


그림 6.4 데이타베이스의 통계적 특성에 대한 정보를 제공하는 페이지

7장. 결론

본 연구 개발의 목적은 한글 글씨 인식에 관하여 연구하는 연구자들이 한글 글씨 인식에 적합한 인식 알고리즘의 개발을 유도하고 개발된 인식 시스템의 성능을 객관적으로 평가해 볼 수 있는 자료로 공동으로 사용될 수 있는 다양한 변형을 포함하는 대용량의 한글 글씨 데이타베이스를 구축하는 것이다.

한글 글씨 데이타의 수집은 한글 글씨 인식에 관한 연구를 수행 중인 연구자들의 협조로 전국의 지역 분포, 다양한 계층의 연령, 성별, 직업별 분포를 고려하여 1,000 명 이상의 필기자들을 선정하고 이들로부터 정서체와 본인의 평소 필기 습관대로 필기하도록 한 자유 필체 등 두 종류로 필기하도록 함으로써 다양한 변형이 포함된 한글 글씨 데이타를 수집하도록 하였다.

본 연구 개발의 3 차년도에는 KS C 완성형 한글 2,350 자 중에서 사용 빈도순 차상위 500 자에 대한 한글 글씨 데이타 1,000 벌을 수집함으로써 최종적으로 사용 빈도순 상위 1,500 자에 대한 1,000 벌의 데이타베이스를 구축하였다. 3 차년도에는 수집 지역 선정에 있어서는 다양한 필기 형태를 수집하기 위해 1, 2 차년도에 선정된 지역과는 다르게 수집 지역을 선정하였으며, 수집 용지 및 필기구 선정, 문자 영상 저장 측면에 있어서는 1, 2 차년도와의 일관성을 유지함으로써 데이타베이스의 질적 측면을 향상시키고자 하였다. 또한, 수집된 데이타베이스에 대한 최종 마무리 작업을 수행하였다. 즉, 1, 2 차년도에 수행된 데이타베이스 검증 및 교정 작업 이후에도 레이블링 오류 및 분할 오류가 남아 있는 데이타 화일들에 대해서 오류 문자를 공백으로 처리함으로써, 오류 문자들로 인해 데이타베이스의 품질이 저하되지 않도록 하였다. 당해년도 데이타베이스에도 같은 후처리를 함으로써 데이타베이스의 전체적인 일관성면을 고려하였다. 또한, 이러한 오류 화일들에 대한 정보를 사용자에게 제공함으로써 데이타베이스를 사용한 문자 인식 시스템의 개발에 불편함을 덜어주고자 하였다. 이로써 사용자는 편리하게 사용 빈도순 상위 1,500 자 1,000 벌의 고품질 데이타베이스를 사용할 수 있게 되었다.

본 연구 개발을 통해서 구축된 KS C 완성형 한글 사용 빈도순 상위 1,500 자 1,000 벌에 대한 데이타베이스를 WWW 상에 올려 놓음으로써 이를 사용하기 원하는 연구자가 ftp 를 이용하여 쉽게 접근할 수 있도록 하였다. 따라서 한글 글씨 인식 분야의 국내 연구자들에게 공동으로 필요한 다양한 변형을 포함하는 대용량의 한글 글씨 데이타베이스를 구축하여 제공함으로써 국내의 한글 글씨 인식 연구를 활성화시키게 될 것이며 한글 글씨 인식 알고리즘의 개발을 유도하여 한글 글씨 인식 시스템의 상용화를 촉진

시키는 데에도 크게 기여하게 될 것이다.