PPEditor: 한국어 의존구조 부착을 위한 반자동 말뭉치 구축 도구

김 재 훈† · 박 은 진††

요 약

말뭉치(corpus)는 많은 언어 정보를 포함하고 있으며, 언어처리 및 계산언어학 분야에서 다양한 용도로 사용되고 있다. 그러나 말뭉치에 언어 정보를 부착하는 데는 많은 시간과 인력이 소요된다. 이 문제를 완화시키기 위해서 말뭉치 구축 도구가 반드시 요구된다. 본 논문에서는 한국어 의존구조 부착을 위한 말뭉치 구축 도구의 설계 및 구현에 관해서 기술한다. 가장 이상적인 방법은 주석자가 전혀 개입하지 않고, 말뭉치를 구축하는 것이나 이것은 사실상 불가능하다. 따라서 대부분의 말뭉치 구축 도구는 반자동으로 구성되어 있으며, 본 논문에서 제안된 도구도 반자동이다. 제안된 도구는 언어 분석기의 분석 결과에 내포된 오류를 효과적으로 수정할 수 있고, 또한 가능한 한 반복적인 작업을 피할 수 있으며 쉽게 사용할 수 있도록 인터페이스를 설계하였다. 제안된 시스템을 이용해서 20어절 이상의 1만 문장에 의존구조를 부착해 보았다. 잘 훈련된 8명의 주석자들이 매일 4시간씩 2개월 동안 구축하였으며, 그 결과는 정확하고 일관성 있는 말뭉치를 구축할 수 있었으며, 작업 시간과 인력도 크게 줄일 수 있었다.

키워드 : 한국어처리, 구문분석, 품사태깅, 부분구문분석, 의존구조, 말뭉치 구축 도구

PPEditor: Semi-Automatic Annotation Tool for Korean Dependency Structure

Jae-Hoon Kim† · Eun-Jin Park††

ABSTRACT

In general, a corpus contains lots of linguistic information and is widely used in the field of natural language processing and computational linguistics. The creation of such the corpus, however, is an expensive, labor-intensive and time-consuming work. To alleviate this problem, annotation tools to build corpora with much linguistic information is indispensable. In this paper, we design and implement an annotation tool for establishing a Korean dependency tree-tagged corpus. The most ideal way is to fully automatically create the corpus without annotators' interventions, but as a matter of fact, it is impossible. The proposed tool is semi-automatic like most other annotation tools and is designed to edit errors, which are generated by basic analyzers like part-of-speech tagger and (partial) parser. We also design it to avoid repetitive works while editing the errors and to use it easily and friendly. Using the proposed annotation tool, 10,000 Korean sentences containing over 20 words are annotated with dependency structures. For 2 months, eight annotators have worked every 4 hours a day. We are confident that we can have accurate and consistent annotations as well as reduced labor and time.

Key Words : Korean Language Processing, Syntactic Analysis, POS tagging, Partial Parsing, Dependency structure, Corpus annotation tool

1. 서 론

최근 여러 형태의 말뭉치가 자연언어처리를 포함한 여러 응용 분야에 널리 사용되고 있으며, 이와 같은 말뭉치를 개발하기 위해 많은 인력과 예산이 소요되고 있다[1-4]. 이 문제는 말뭉치 구축 도구를 사용함으로써 어느 정도는 해결할 수 있다[5-10].

말뭉치는 언어 자체(예: 한국어, 영어 등), 언어 정보의 종류(품사, 구문구조 등)와 응용분야(예: 기계번역, 정보검색, 전문용어 분석 등)에 따라 매우 다양하게 분류된다. 이와 같은 다양성에도 불구하고 몇몇 말뭉치 구축 도구[7, 8, 11]는 여러 언어, 여러 언어정보 등을 다루고 있다. 그러나 다양한 말뭉치의 특성을 하나의 말뭉치 구축 도구에서 충분히 다룰 수 없을 것이다. 설령, 다양한 말뭉치의 특성을 수용하더라도 각 언어 혹은 주어진 문제의 특성을 효과적으로 잘 다룰

† 종신회원 : 한국해양대학교 컴퓨터공학과 부교수
†† 정 회 원 : 한국해양대학교 컴퓨터공학과 석사과정
논문접수 : 2005년 8월 4일, 심사완료 : 2006년 1월 31일

수는 없으며, 더구나 새로운 응용분야(예: 웹 문서 분류)가 등장되었을 때, 기존의 도구를 그대로 사용할 수 없다[12].

본 논문은 한국어 의존구조부착 말뭉치를 효과적으로 구축하기 위해 말뭉치 구축 도구의 설계 및 구현에 관해서 기술한다. 한국어 의존구조에는 크게 세 가지의 언어 정보, 즉 형태 정보(morphological information), 구묶음 정보(chunk information), 구문 정보(syntactic information)가 포함되어 있다. 형태 정보에는 형태소 분리 및 복원 정보와 품사 정보가 있으며, 구묶음 정보에는 구묶음의 품사와 경계(boundary) 정보가 있고, 구문 정보에는 중심어 및 의존 관계 정보가 있다. 이들은 서로 밀접하게 연관되어 있어서 어떤 한 정보로 쉽게 무시할 수 없다. 본 논문은 한국어 구문분석에서 널리 사용되고 있는 의존문법을 기반으로 구문구조 정보를 부착하고자 한다. 의존구조로 구문정보를 부착하더라도 서로 변환이 가능하기[13, 14] 때문에 한국어 구문분석 시스템을 개발하는 데 널리 사용될 수 있을 것으로 생각된다. 본 논문에서 제안된 말뭉치 구축 도구는 반자동으로 실행된다. 즉 문장이 주어지면 구문분석기[15]를 이용해서 구문분석을 수행하고, 분석 결과에 포함된 오류를 수정하는 방법을 취한다. 이 방법은 일반적인 말뭉치 구축 도구와 크게 다르지 않다. 제안된 말뭉치 구축 도구가 다양한 언어정보를 효과적으로 수정하고 편집될 수 있도록 설계되었다는 점이 다른 시스템과 크게 구별된다. 예를 들면, 낮은 단계(형태소 분석)의 오류가 수정되면 그 결과가 바로 높은 단계(구문분석)로 전달되어 바로 반영될 수 있도록 하였으며, 필요할 경우에는 낮은 단계의 오류가 높은 단계에 영향을 주지 않도록 다시 작업을 해야 하는 번거로움을 최소화하였다. 더구나 수정과정에서 반복적으로 일어나는 행위를 감시하다가 같은 과정이 어느 수준 이상이 되면 그 행위를 자동적으로 수행할 수 있도록 설계되었다. 그 외에도 제안된 말뭉치 구축 도구는 아래와 같은 특징을 가지고 있다. 1) 특정 응용분야에 관계없이 두루 사용할 수 있다. 2) 분석 단계와 분석 오류를 연계하여 작업의 집중도를 높였다. 3) 가능한 한 오류는 축적되지 않도록 하여 구축된 말뭉치의 질을 크게 개선할 수 있었다. 4) 구축된 정보는 서로 공유할 수 있도록 하여 작업의 일관성을 극대화하였다. 5) 초보자로 사용자가 쉽게 도구를 사용할 수 있도록 인터페이스를 설계하였다.

제안된 도구를 사용해서 20어절 이상으로 구성된 10,000 문장의 의존구조 부착 말뭉치를 구축하였다. 약 2개월 동안, 8명의 연구원이 하루 평균 4시간의 작업으로 의존구조 부착 말뭉치를 구축할 수 있었다. 그 결과는 정확하고 일관성 있는 말뭉치를 구축할 수 있었으며, 작업 시간과 인력도 크게 줄일 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구로서 기존의 말뭉치 구축 도구에 대해서 살펴본다. 3장과 4장에서는 각각 제안된 말뭉치 구축 도구의 설계 및 구현에 대해서 자세히 기술한다. 5장에서 제안된 말뭉치 도구의 적용 사례에 대해서 기술하고, 6장에서는 결론을 맺고 앞으로의 연구에 대해서 기술한다.

2. 관련 연구

2.1 말뭉치 구축 도구의 사용자 요구

본 절에서는 Reidsma와 그의 동료들의 연구[12]를 통해서 말뭉치 구축 도구의 사용자 요구 사항(user requirement)을 살펴보고 본 논문에서 제안된 시스템이 이에 얼마나 부합하는지를 살펴보고자 한다. 말뭉치 구축 도구의 사용자는 다양한 분야(예: 언어학, 자연언어처리, 심리학 등)에 종사하며, 이들의 요구가 매우 다양할 수 있다. 말뭉치 구축 도구의 사용자는 크게 주석자(annotator), 말뭉치 사용자(corpus user), 말뭉치 개발자(corpus developers), 말뭉치 구축 도구 개발자(system developers)로 구분되며, 이들에 따라서도 요구사항이 달라질 수 있다. 제안된 말뭉치 구축 도구는 주로 주석자의 요구 사항을 반영하였다. 즉 효율적으로 말뭉치를 구축하는 데 주안점을 두고 있다. [12]에 따르면 말뭉치 도구에 대한 주석자의 요구사항은 아래와 같이 요약된다.

- **이식성이 좋아야 한다(portability).** 주석자들은 연구실뿐 아니라 자택에서도 작업을 희망한다. 따라서, 여러 시스템에 자유롭게 설치되어야 한다.
- **학습 편의성이 좋아야 한다(learnability).** 언어 정보를 부착하는 일을 고도의 지적인 작업이며 많은 부가적인 지식이 필요하다. 따라서 말뭉치 구축 도구 내에서 원하는 정보를 쉽게 습득하고 익힐 수 있어야 한다.
- **인터페이스가 사용자에게 편해야 한다(user-friendly interface).** 인터페이스는 쉽고 직감적으로 사용할 수 있어야 한다.
- **다루는 내용은 가능한 한 시각화되어야 한다(visualization).** 언어 정보는 다소 복잡하므로 이런 정보를 이해하기 용이하도록 시각화된다면 작업의 능률을 훨씬 더 높일 수 있을 것이다. 또한 다양한 환경 변수를 이용해서 분석 및 수정 결과를 개인의 특성에 맞도록 조절할 수 있어야 작업을 좀더 편하게 할 수 있을 것이다.
- **가능한 한 많은 기능이 (반)자동화되어야 한다((semi-)automatic annotation):** 언어 분석에 대한 기술력이 만족할 수는 없지만 어느 정도까지는 유용하게 사용할 수 있다. 따라서 기존의 언어 분석 기술을 이용해서 어느 정도까지는 (반)자동으로 언어 정보를 부착하거나 수정할 수 있어야 한다.
- **문서화가 잘 되어야 한다(documentation).** 처음으로 언어 정보를 부착하거나 말뭉치 구축 도구를 사용하는 이에게는 매뉴얼이나 지침서가 매우 중요하다. 따라서 이들이 온라인으로 접근하고 검색할 수 있어야 한다.
- **작업에 집중할 수 있도록 설계되어야 한다(attractiveness).** 언어 정보 부착은 지적일 뿐 아니라 매우 집중은 요하는 작업으로 말뭉치 도구에서도 이를 도와줄 수 있는 기능들이 제공되어야 한다.
- **많은 부가정보들도 지원되어야 한다(meta-data support).** 주석자와 날짜 등과 같은 부가적인 정보를 관리해야 한다.

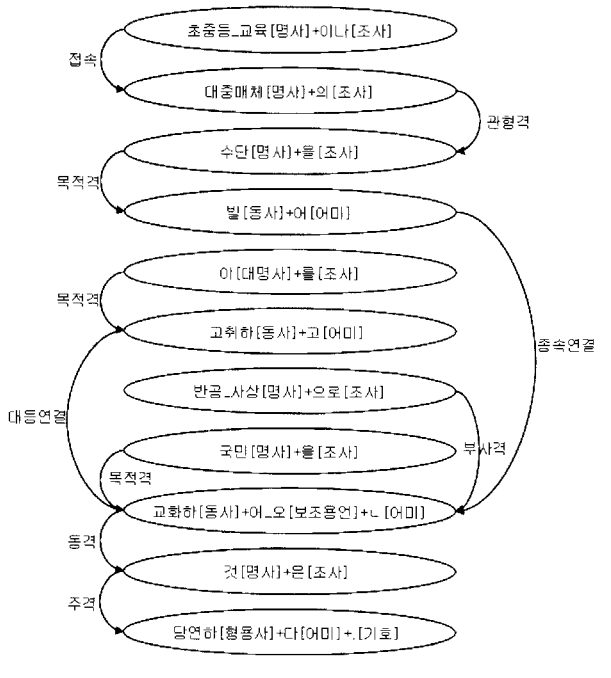
제안된 말뭉치 구축 도구는 이와 같은 사용자 요구사항을 반영하였고, 더구나 구축된 말뭉치의 일관성 유지를 위해서 실시간 정보 공유(instant information sharing) 기능을 추가하였다.

2.2 한국어 의존구조

본 절에서는 제안된 말뭉치 도구에서 다루는 한국어의 구문구조에 대해서 살펴보고자 한다. 본 논문에서 다루는 의존구조에서 형태소(morpheme), 품사(part-of-speech), 구묶음(chunk), 의존관계(dependency relation), 중심어(head) 정보가 포함되어 있다. (그림 1)은 (예문 1)의 의존구조이다[1].

(예문 1) 초중등 교육이나 대중매체의 수단을 빌어 이를 고취하고 반공사상으로 국민을 교화해 온 것은 당연하다.

(그림 1)에서 형태소는 기호 "+"로 구분된다. 예를 들어, "수단+을"은 두 개의 형태소로 구성되었음을 의미한다. 품사는 대괄호([ ])로 구분되며, 독자의 이해를 돕기 위해서 제안된 시스템에서 사용하는 품사가 아니라 학교문법에서 다루는 기본 품사로 표현하였다. 구묶음은 둘 이상의 형태소가 문장에서 하나의 구성성분(constituent) 역할을 할 경우를 말하며, 연결자로 "_"를 사용한다. 예를 들면 두 형태소 "반공"과 "사상"이 하나의 구성성분로 사용되어 구묶음으로 표현하였다. 구문관계는 화살표 위에 표현하고, 중심어는 화살표의 방향에 의해서 표현되며 화살촉이 가리키는 쪽이 중심어가 된다.



(그림 1) 한국어 의존구조의 예

1) 본 논문에서는 독자들이 쉽게 이해할 수 있도록 제안된 말뭉치 구축 도구에서 사용되는 품사명과 의존관계명이 조금 다르다. 또한 제안된 말뭉치 구축 도구에서 의존구조는 (그림 1)과 같은 그래프로 표현하는 것이 아니라 는 표 형식으로 표현하여 데이터베이스에 저장된다.

2.3 기존 말뭉치 구축 도구

앞에서 언급했듯이 말뭉치 구축 도구는 한국어와 영어와 같은 언어(language), 품사와 구문구조와 같은 부착 정보(tagged information), 문장과 음성과 같은 입력 종류(source) 등에 따라 매우 다양하다. 본 절에서는 제안된 말뭉치 구축 도구와 밀접한 관계를 가지고 있는 구묶음[2] 부착 및 구문구조 부착 말뭉치 구축 도구에 대해서 살펴보고자 한다.

Alembic Workbench[8]는 정보추출 시스템의 학습을 위해서 필요한 개체명(named-entity)을 부착하기 위해서 MITRE[3]에서 개발된 말뭉치 구축 도구이다. Alembic Workbench는 주로 개체명과 대용어(anaphora) 부착 도구이다. 주요 기능으로는 다국어 지원 가능하고, 구축된 말뭉치는 SGML 형식으로 출력된다. 또한 말뭉치를 구축하는 과정에서 축적된 패턴에 따라서 어느 정도 반자동 구축이 가능하다. 사용자 인터페이스를 통해서 사용자가 쉽게 개체명(name-entity)을 부착할 수 있도록 설계되었고, 사용자의 행동을 학습하여 부착 작업에 도움을 줄 수 있다. 이 시스템은 2004년에 다국어 지원이 가능하도록 크게 개선하여 Callisto[4]라는 이름으로 공개하였다. 이 시스템을 한국어에 그대로 적용하기 위해서는 형태소 분석과 같은 전처리 작업이 필요하다.

WordFreak[7]도 Alembic workbench와 같이 인터넷[5] 을 통해서 공개된 말뭉치 구축 도구이다. 자바로 구현되어 다국어와 여러 종류의 운영체제에 쉽게 적용될 수 있다. 또한 구문정보, 개체명, 대용어 정보뿐 아니라 구묶음, 구문구조 등과 같은 다양한 언어정보를 쉽게 부착할 수 있도록 설계되었다. 시스템의 많은 구성 요소들은 쉽게 확장할 수 있고 재사용할 수 있도록 설계되었다. 주요 기능으로는 새로운 문서에 쉽고 빠르게 언어정보를 부착할 수 있도록 자동 언어정보 부착 기능이 포함되어 있다.

Annotate[9]은 NEGRA 말뭉치를 구축하기 위해서 개발되었다. NEGRA 말뭉치[6]는 독일어 신문 기사에 구문구조를 부착된 말뭉치이다. Annotate는 품사 태거와 구문분석기를 포함하는 반자동 말뭉치 구축 도구이다. 이 도구는 특별히 품사 태거와 구문분석기의 결과에 대한 신뢰도를 측정하여 신뢰도가 어느 수준 이하일 경우에 주석자들이 즉시 수정할 수 있다. 그 결과 매우 빠른 속도로 언어정보를 부착할 수 있다[7].

구문분석 말뭉치 종합 관리 도구[6]는 세종계획(문화공보부, 2003)에서 구축하는 이진 구구조 구문분석 말뭉치를 변환/검색/수정하는데 편의성을 제공하기 위하여 개발된 종합 관리 도구이다. 기존의 구문분석 말뭉치를 관리하는 작업이

2) 구묶음(chunk)은 개체명(named-entity)와 명사구(noun phrase) 등을 모두 포함한다.

3) http://www.mitre.org/tech/alembic-workbench/

4) http://callisto.mitre.org/

5) http://wordfreak.sourceforge.net/

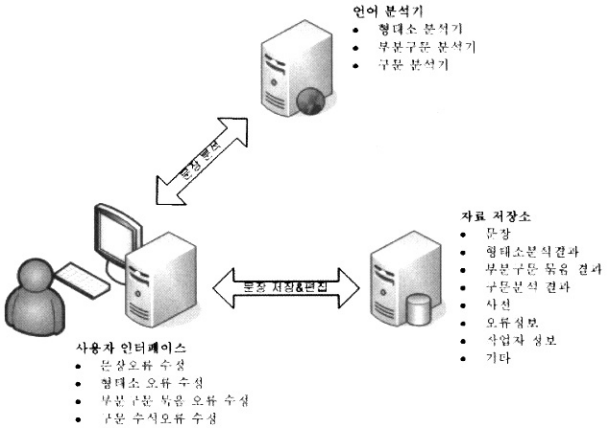
6) http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/

7) (Brants et al, 2000)에 따르면 문장당 평균 50초만에 품사 및 구문 정보를 부착할 수 있다고 한다.

독립적으로 수행되던데 비해서 구문분석 말뭉치를 관리하는 통합 환경을 제공한다. 그러나 다중 작업자 처리와 각 단계별로 문장이 파일로 존재하는 관계로 대용량 말뭉치를 구축하기엔 부적절한 도구이다.

3. PPEditor: 한국어 의존구조 부착 말뭉치 구축 도구

본 절에서는 제안된 말뭉치 구축 도구(PPEditor: (Partial) Parse structure Editor)의 구조에 대해서 자세히 기술하고자 한다. (그림 2)는 PPEditor의 시스템 구조이며, 크게 언어 분석기, 자료 저장소, 사용자 인터페이스로 나눈다. PPEditor을 크게 세 부분으로 나누어 설계한 가장 큰 이유는 언어 분석기가 많은 컴퓨터 자원을 요구하기 때문이다. 그러나 세 하부 시스템의 설치는 제약을 받지 않는다. 즉 세 하부 시스템이 모두 하나의 컴퓨터에 설치될 수도 있고 원하면 각 다른 컴퓨터에 설치할 수도 있다. 개략적인 말뭉치의 구축 과정은 다음과 같다. 1) 언어 분석기를 통해서 주어진 문장을 분석한다. 2) 그 결과를 오류 편집기로 보여준다. 3) 주석가가 잘못된 부분을 수정하여 그 결과를 자료 저장소에 저장한다. 이 방법을 일반적으로 반자동 말뭉치 구축 도구(semi-automatic corpus annotation tool)라고 한다. 이와 같이 언어 분석기의 결과를 수정한다는 의미에서 반자동 말뭉치 구축 도구를 오류 편집기(error editor)라고 부르기도 한다. 이하에서는 주로 오류 편집기와 말뭉치 구축 도구를 서로 혼용해서 사용할 수도 있다. PPEditor는 언어 분석기에 의해서 자동으로 분석된 언어 정보의 오류를 빠르고 쉽게 수정할 수 있도록 설계되었다. 또한 본 시스템은 대량의 말뭉치 분석 작업을 다수의 작업자가 동시에 작업할 수 있도록 중앙에 데이터베이스를 구성해서 말뭉치 구축 작업이 분산되어 수행할 수 있도록 설계되었다.



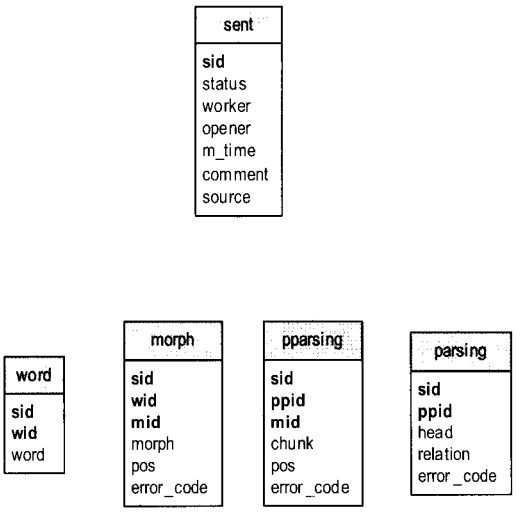
(그림 2) PPEditor의 시스템 구조

3.1 언어 분석기

언어 분석기는 한중기계번역 시스템의 한 부분으로 개발되었으며, 형태소 분석기, 품사 부착기, 부분구문분석기, 구문분석기로 구성되어 있다. 자세한 설명은 (Kim et al, 2004)을 참고하기 바란다.

3.2 자료 저장소(데이터베이스)

자료 저장소는 데이터베이스로 구성되며, (그림 3)은 언어 분석 결과를 저장하기 위한 데이터베이스 구조이다. (그림 3)의 데이터베이스는 총 6개의 테이블로 구성되고, 각 테이블은 기본 키(그림에서는 굵은 글자체)를 포함해서 하나 이상의 피일드로 구성된다. sent 테이블은 문장의 상태(status), 주석자(worker), 소유자(opener), 최종 수정 시간(m_time), 문장의 출처(source) 등이 기록되며, word 테이블은 각 문장에 속한 어절을 저장한다. morph 테이블은 word 테이블의 각 어절의 형태소 결과를 저장한다. pparsing 테이블은 형태소 분석 결과로부터 가능한 구묶음을 저장한다. parsing 테이블은 구묶음을 기준으로 각 구의 중심어와 중심어와의 의존관계를 저장한다. 본 논문에는 (그림 3)에서 보여준 테이블 이외에도 사용자 정보, 형태소 사전과 구묶음 사전, 오류 및 오류 문맥 정보, 문장상태를 관리하기 위한 테이블이 있으나 지면 관계로 구체적인 내용은 다루지 않을 것이다.

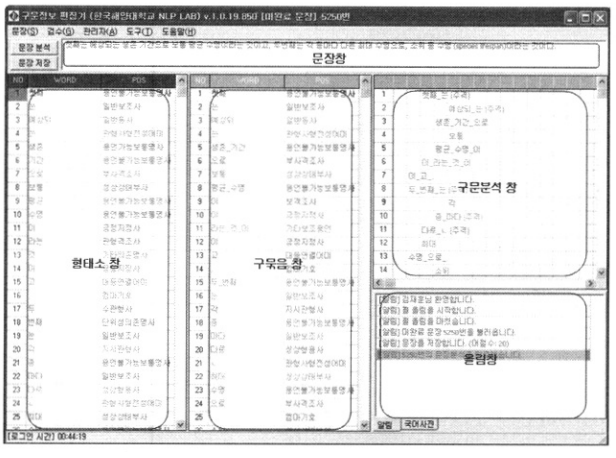


(그림 3) 언어 분석 결과를 저장하기 위한 데이터베이스의 구조

3.3 그래픽 사용자 인터페이스

그래픽 사용자 인터페이스는 (그림 4)에서 보는 바와 같이 문장 창, 형태소 창, 구묶음 창, 구문구조 창, 알림 창으로 구성된다.

알림 창을 제외하고는 말뭉치 구축에서 발생될 수 있는 오류와 밀접한 관계를 가지고 있으며(<표 1>), 또한 각 창은 주어진 문장이나 언어 분석 결과에 존재하는 오류를 수정할 수 있다. 문장 창은 주어진 문장을 보여주면 문장의 오류(철자 오류, 띄어쓰기 오류)를 수정할 수 있다. 형태소 창은 형태소 분석 결과를 보여주며, 형태소 분석 오류(품사 오류, 형태소 분리 오류, 형태소 복원 오류)를 수정할 수 있다. 품사 오류를 수정할 때는 입력 오류를 최소화하기 위해



(그림 4) PPEditor의 사용자 인터페이스

콤보박스를 통해서 구현되었다. 구묶음 창은 구묶음 분석 결과를 보여주고 구묶음 분석 오류(품사 오류, 구묶음 경계 인식 오류)를 수정할 수 있다. 구묶음 창에서 품사 오류 수정은 형태소 창의 기능과 같으며, 구묶음 경계를 잘못 인식하여 구묶음을 해제할 경우 구묶음의 품사는 자동으로 형태소의 품사로 대체된다. 구문구조 창은 구문분석 결과를 보여주고 구문분석 오류(중심어 지정 오류, 의존관계 오류)를 수정할 수 있다. 구문분석의 오류는 주로 장문(long sentence)에서 발생되는데 일반적으로 장문의 구문분석 결과에서는 중심어의 위치를 파악하는 것은 쉽지 않다. 본 논문에서는 언어분석기의 구문분석 결과에서 중심어를 쉽게 찾을 수 있도록 중심어를 강조하여 주석자는 최단 시간에 많은 문장을 작업할 수 있도록 하였다. 또한 문장분석 시간을 단축하기 위하여 각 기능별로 단축키(hot key)를 제공하여 문장 분석 작업의 속도를 높였다.

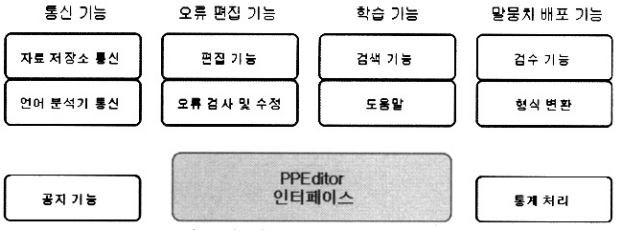
〈표 2〉 창과 오류와의 관계

창	오류
문장	철자, 띄어쓰기
형태소	품사, 형태소 분리, 형태소 복원
구묶음	품사, 구묶음 경계
구문구조	중심어, 의존관계

3.4 PPEditor의 기능

(그림 5)는 PPEditor의 구조를 대표 기능별로 나타낸 것이다. PPEditor의 기능은 크게 외부 시스템과의 통신 기능, 오류 편집 기능, 학습 기능, 말뭉치 배포 기능, 기타 기능으로 분류되며, 모든 시스템의 제어는 사용자 인터페이스를 통해서 이루어진다.

통신 기능은 언어 분석기와 자료 저장소와의 통신을 위해서 필요한 기능이다. 언어 분석기는 문장뿐 아니라 품사 태거와 같은 각 언어 분석기의 분석 결과를 주고받아야 한다. 언어분석기와의 통신 방법으로는 CGI(Common Gateway Interface)를 사용한다. 따라서 여러 운영체제나 컴퓨터 시스템에 자유롭게 설치할 수 있다. 분석 결과는 정해진 형



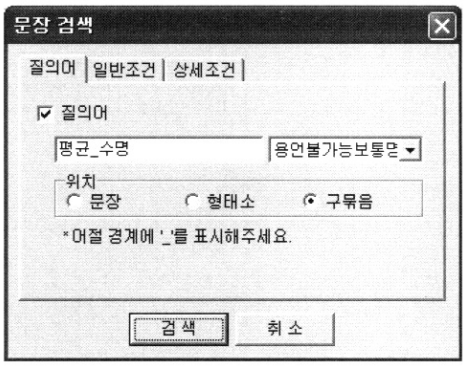
(그림 5) PPEditor의 구조

식에 따라 표현되는데 형식의 구체적인 내용은 지면상의 이유로 본 논문에서는 다루지 않을 것이다. 자료 저장소는 데이터베이스로 구성되어 있기 때문에 자료 저장소와의 통신 방법으로 SQL(Structured Query Language)을 사용한다.

오류 편집 기능은 PPEditor의 주된 기능이다. 일반적인 편집기가 가질 수 있는 기능인 삽입, 삭제, 추가 등과 같은 기능이 기본적으로 제공된다. 그러나 일반적인 편집기와는 몇 가지 다른 특징을 가지고 있다. 첫째, 언어 분석기는 같은 오류를 반복적으로 범한다. 이를 해결하기 위해서는 본 논문에서는 반복적으로 출현되는 오류에 대해서 오류수정규칙[8]을 등록하여 자동으로 해당 오류를 수정할 수 있도록 하였다. 둘째, 낮은 단계(예: 형태소 분석)의 오류는 높은 단계(예: 구문분석)로 전달된다. 예를 들면 문장의 철자 오류는 형태소 분석과 그 이상의 언어 분석에 그대로 영향을 준다. 이를 해결하기 위해서는 본 논문에서는 낮은 단계의 오류 수정 결과가 높은 단계에 그대로 반영될 수 있도록 하였다. 셋째, 자연언어는 본질적으로 모호한 문장이 자주 사용된다. 본질적으로 모호한 표현이라고 하더라도, 구축된 말뭉치 전체에서는 항상 일관성이 유지되어야 한다. 이를 위해서는 본 논문에서는 비슷한 언어 현상이나 문맥을 검색할 수 있도록 하였으며 좀더 자세한 내용은 다음에 기술할 것이다. 넷째, 말뭉치를 구축하는 일은 고도의 지적인 작업으로 작업자가 잠시 방심하면 하찮은 오류를 쉽게 간과할 수 있다. 이를 방지하기 위해 가능한 한 오류가 자료 저장소에 저장되지 않도록 해야 한다. 이를 위해 본 논문에서는 오류 검사 기능을 통해 오류가 말뭉치에 저장되지 않도록 하였다. 다섯째, 원어민이라 하더라도 오류를 판단하기 위해서는 사전 등의 자료가 필요하다. 그만큼 말뭉치를 작성하는 일은 쉬운 일이 아니다. 이를 위해서 본 논문은 사전 검색뿐 아니라 예제 검색을 통해서 오류를 판단하는 데 도움을 주고자 하였다. 여섯째, 하나의 컴퓨터 화면에 여러 가지 정보가 동시에 보여짐으로써 자칫 작업의 위치를 잃어버릴 수 있다. 본 논문에서는 이를 해결하기 위해 여러 가지의 관심집중(focus) 기능을 추가하였다. 즉 작업자가 무슨 일을 하고 있는지를 쉽게 판단할 수 있도록 하였다.

학습 기능은 말뭉치 구축에 필요한 온라인 학습 기능이다. 본 논문에서는 학습 기능으로 말뭉치 구축 도구의 사용법, 말뭉치 구축 지침서, 예제 검색 기능을 제공한다. 도움말

8) 오류수정규칙은 문맥과 변환규칙으로 정의되며, 문맥은 오류가 일어난 주변 단어로 표현되고 변환규칙은 오류의 수정 규칙이다. 문맥은 정규표현식(regular expression)으로 표현된다.



(그림 6) 예제 검색 인터페이스

기능을 통해서 말뭉치 구축 도구의 사용법과 말뭉치 구축 지침서를 제공한다. 예제 검색은 (그림 6)과 같은 인터페이스를 통해서 예제를 검색할 수 있으며, 난해한 분석의 해결뿐 아니라 말뭉치 구축의 일관성 유지에 대단히 중요한 역할을 한다.

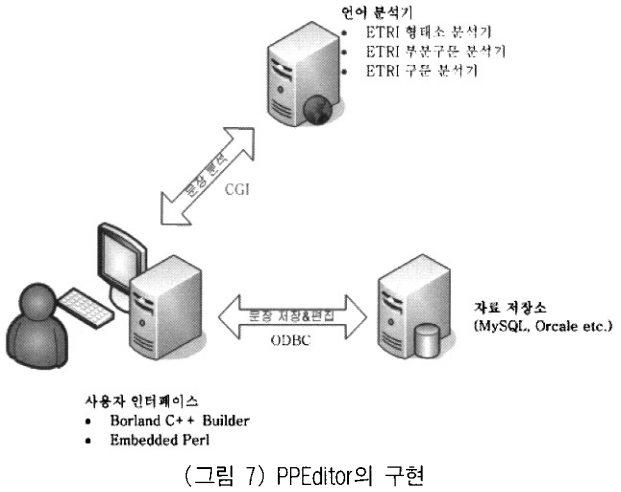
말뭉치 배포 기능은 구축된 말뭉치는 일련의 검수 과정을 통해서 배포된다. 일반적으로 말뭉치의 배포는 SGML(Standard Generalized Markup Language)과 같은 특별한 형식으로 배포된다. 본 논문에서는 지면 관계상 자세한 형식은 다루지 않을 것이다.

그 밖에도 말뭉치 구축이 완료된 문장 수와 같은 통계 정보를 제공하는 통계 기능과 각종 공지사항을 전달하는 공지 기능을 제공한다.

4. PPEditor의 구현

본 절에서는 제안된 말뭉치 구축 도구가 어떻게 구현되었는지를 소개한다. (그림 2)의 PPEditor의 전체 구조를 토대로 PPEditor의 구현 방법을 (그림 7)에서 보여주고 있다.

먼저 앞에서 언급했듯이 언어 분석기는 한중 기계번역 시스템[15]에 사용되었던 형태소 분석기, 품사 부착기, 부분구문분석기, 구문분석기를 본 연구에 맞게 재구성하고, CGI



(그림 7) PPEditor의 구현

인터페이스 기능을 추가하였다. 자료 저장소는 MySQL[9]로 구현되었다. MySQL은 공개된 DBMS 중에서 가장 널리 사용될 뿐 아니라 Windows나 Linux 등과 같은 여러 플랫폼에 자유롭게 설치될 수 있도록 배포판을 제공한다.

그래픽 사용자 인터페이스는 Borland C++ Builder[10]를 이용하여 구현하였다. Borland C++ Builder은 GUI를 작성하는 데 가장 강력한 도구 중 하나이며, 단시간에 GUI를 작성하는데 매우 유용한 도구이다. 사용자 인터페이스는 CGI를 통해서 언어 분석기와 통신한다. CGI는 HTTP 서버와 정보를 교환하기 위한 표준이다. 많은 계산량을 요구하는 시스템을 구축할 때 CGI를 사용함으로써 외부 컴퓨터 시스템을 자유롭게 연결할 수 있다. 언어 분석기의 결과를 자료 저장소에 적합하도록 변환해야 하는데 이를 위해 Perl 언어[11]를 사용한다. Perl 언어는 문자열을 다루기에 매우 효과적인 언어이다. 사용자 인터페이스의 구현이 C++ 언어로 구현되었기 때문에 Perl 언어는 C++ 언어에 내장된 형식으로 처리되기 때문에 두 언어 사이의 통신 부하는 전혀 없다. 사용자 인터페이스와 자료 저장소의 연결은 ODBC(Open DataBase Connectivity)를 사용하였다. ODBC는 1992년 SAG(SQL Access Group)에 의해서 개발된 표준 데이터베이스 근접 방법이며 DBMS(database management system)에 무관하게 자료를 접근할 수 있다. 즉 본 논문에서는 MySQL을 사용하지만 Oracle과 같은 다른 DBMS를 사용하더라도 그대로 사용할 수 있다.

구현된 시스템은 크게 두 종류로 나눈다. 하나는 개발용이고 또 다른 하나는 실제 작업용이다. 두 시스템 모두 언어 분석기는 ETRI 시스템을 사용한다. 원거리에 설치되어 네트워크의 사정이 원활하지 않을 경우는 다소 지연이 생길 수 있으나, 대체로 큰 문제가 없었다. 사용자 인터페이스와 MySQL은 개발용은 한 시스템에 설치하여 실험하였고, 작업용은 서로 다른 컴퓨터에 설치하였다. 왜냐하면 여러 작업자들이 동시에 자료 저장소를 접근하기 때문에 DB를 다른 컴퓨터에 설치하였다.

5. 한국어 의존구조 부착 말뭉치 구축

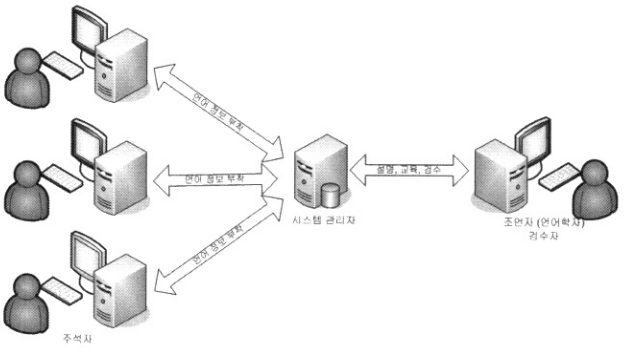
제안된 말뭉치 구축 도구를 사용해서 10,000문장의 의존구조 부착 말뭉치를 구축하였다. 모든 문장은 20어절 이상으로 구성되었으며, 적어도 3개 이상을 단문이 결합된 복문들이다. 의존구조 내에는 단순한 구문구조 정보뿐 아니라, 형태소 분리, 형태소 복원, 품사, 구묶음, 중심어, 의존 관계 정보가 포함되어 있다. 총 작업 시간은 8명 × 4시간/일 × 20일/개월 * 2개월 = 1,280시간이다. 따라서 문장당 0.128시간이 소요되며, 1시간에 약 7.8문장을 작업할 수 있었다. 이 수치는 [9]의 결과와 직접 비교하면 너무나 큰 차이를 보인다. 그 이유는 다음과 같이 요약된다. 첫째, 본 논문에서는

9) http://www.mysql.com/
10) http://www.borland.com/
11) http://www.perl.com/

말뭉치 구축 문장의 길이가 모두 20어절 이상이다. 문장의 작업 속도를 기하급수적으로 떨어뜨린다. 이것은 구문결과인 구문트리의 수를 보아도 충분히 알 수 있다. 둘째, 한국어는 생략된 부분이 많아서 문장의 의미를 정확하게 파악하는데 다소 시간이 걸린다. 셋째, 주석자들이 약 1개월을 훈련했지만 충분한 훈련이 되지는 않았다. 넷째, 말뭉치 구축 중에 지침서가 일부 변경되었다.

5.1 말뭉치 구축 환경

(그림 8)은 한국어 의존구조 말뭉치를 구축하기 위한 환경을 개략적으로 설명하고 있다. 말뭉치 구축 도구의 사용자는 크게 주석자, 조언자, 검수자, 시스템 관리자로 구성된다. 주석자는 말뭉치 구축을 담당한다. 대부분의 사용자가 주석자이므로 말뭉치 구축 도구는 모두 주석자를 중심으로 설계되었다. 조언자는 난해한 언어 분석을 도와주는 언어학자이며, 검수자는 말뭉치 구축 결과가 올바르게 되는지를 검수하는 자이며 일반적으로 조언자가 검수자 역할을 한다. 검수자가 일정한 주기를 가지고 말뭉치 구축 결과로부터 임의로 5%의 문장을 추출하여 오류가 있는지를 검수하고, 그 결과를 주석자들에게 전달하여 유사한 오류가 나타나지 않도록 하였다. 이 작업을 통해서 구축된 말뭉치의 신뢰도를 크게 향상시킬 수 있었다. 시스템 관리자는 데이터베이스의 관리 및 백업 그리고 도구의 사용 및 설치를 돕는다.



(그림 8) 말뭉치 구축 시나리오

5.2 말뭉치 구축 과정

일반적으로는 아래와 같은 순서로 말뭉치 구축 작업이 진행된다.

1. 자료 저장소에서 문장을 선택한다.
2. **언어 분석기에 문장을 보낸다.**
3. **언어분석기가 문장을 분석한다.**
4. **언어분석기로부터 분석 결과를 받는다.**
5. **사용자 인터페이스에 분석 결과를 보여준다.**
6. **오류가 있는지를 찾아는다.**
7. **찾아진 오류를 수정한다.**
8. **모든 오류가 완전히 수정될 때까지 2에서 7 과정을 반복한다.**
9. 최종 결과를 자료 저장소에 저장한다.

정상적인 경우에는 항상 위와 같은 과정이 반복되는 작업이 말뭉치 구축 과정이다. 주석자가 난해한 문장을 접하면 이를 해결하지 위해서 많은 시간을 낭비하는 것보다는 조언자(언어학자)를 통해서 해결하도록 작업 환경을 구성하였다. 주석자가 자료 저장소를 통해서 조언자에게 무엇이 문제인지를 요청하면 조언자는 그 해결 방법을 자료 저장소를 통해서 전달한다. 조언자는 요청된 문제가 모든 작업자가 알아야 할 경우에는 게시판을 이용해서 공지한다.

5.3 말뭉치 구축 오류

제안된 말뭉치 구축 도구는 반자동이며 오류를 수정함으로써 말뭉치가 구축된다. 언어 분석기에서 발생된 오류뿐 아니라 원문 자체에도 오류가 포함되어 있다. 본 절에서는 오류 유형을 살펴보고(<표 2>) 각 오류의 유형이 어떤 방법으로 수정되는지를 살펴보고자 한다. <표 2>에서 "+"는 형태소 분리를 나타내고, "/"는 품사나 의존관계에 해당하는 부가적인 언어 정보를 나타낸다. 또한 "< >"는 하나의 구묶음을 표현한다. 오류는 총 9개이며, 철자, 띄어쓰기, 형태소 분리, 형태소 복원, 품사, 구묶음 인식 및 미인식, 중심어, 의존관계 오류가 있다. 앞에서도 언급했듯이 각 오류의 유형은 언어 분석기의 각 모듈과 사용자 인터페이스의 각 창과 밀접하게 연계되어 있다. 따라서 인터페이스는 각 창에 따라서 수행될 수 있는 행위가 크게 차이가 있을 수 있다.

〈표 2〉 오류의 유형과 예

오류 유형	예제		오류의 출처
	오류	수정	
철자	등안시	등한시	원문
띄어쓰기	저희들 한테	저희들한테	원문
형태소 분리	자유+로	자유로	형태소 분석
형태소 복원	집+을 지+어	집+을 짓+어	형태소 분석
품사	김정일/기타고유명사	김정일/인명고유명사	품사 태깅, 구묶음
구묶음 미인식	먹+게 되+ㄴ다	먹+<게_되>+ㄴ다	부분구문분석
구묶음 인식	<다음, 자기 집 옆>+에	다음, <자기 집 옆>+에	부분구문분석
중심어 지정	철수+가 밥+을 먹+었+다+.	철수+가 밥+을 먹+었+다+.	구문분석
의존관계	밥+을/주격 먹+었+다+.	밥+을/목적격 먹+었+다+.	구문분석

6. 결론 및 앞으로의 연구

본 논문에서는 한국어 의존구조를 부착하기 위한 말뭉치 구축 도구인 PPEditor의 설계 및 구현 방법에 대해서 기술하였다. 한국어 의존구조에는 구문구조뿐 아니라 형태소 분리 및 복원, 품사, 구묶음, 중심어, 의존관계에 관한 정보가 포함되어 있다. PPEditor를 사용해서 말뭉치를 구축하는 일은 언어 분석기의 분석 결과를 수정하는 일과 같다. 따라서 PPEditor을 오류 편집기라고도 한다. PPEditor는 다른 분석기와 비교하면 몇 가지 장점을 가지고 있다. 즉, 응용분야에

무관하며, 말뭉치 구축에 집중할 수 있도록 오류를 지역화 및 국부화시켰으며, 강력한 오류 검사 및 수정 기능을 가지고 있다. 또한 구축된 말뭉치 정보를 바로바로 공유할 수 있으며, 사용자 인터페이스는 최대한 사용자에게 친근하도록 설계되었다.

PPEditor을 사용해서 10,000문장의 의존구조 말뭉치를 구축하였다. 이 말뭉치를 구축하는 데는 8명의 주석자가 하루 4시간씩 2개월이라는 시간이 소요되었다. 이 결과는 정확하고 일관성 있는 말뭉치를 구축할 수 있었으며, 작업 시간과 인력도 크게 줄일 수 있었다.

PPEditor은 자동 수정 기능에서 약간의 오류를 일으킨다. 즉 자동 수정 기능을 사용했을 경우 잘못 수정하는 경우가 종종 발생한다. 가까운 장래에 변환기반 기법[16]과 같은 기계학습 방법으로 이용해서 좀더 나은 성능을 가지도록 개선되어야 한다. 또한 각 언어 부착의 신뢰도[17]를 측정하여 오류의 가능성이 높은 어구에 대해서 특별한 표시를 해줌으로써 작업 효율을 크게 높일 수 있을 것으로 생각한다.

참 고 문 헌

[1] 문화관광부, 21세기 세종계획 국어 기초자료 구축, 연구보고서, 2003.

[2] 문화체육부&과학기술처, 대한민국 국어정보베이스, 연구보고서, 1998.

[3] Marcus, M. P., Santorini, B. and Marcinkiewicz, M. A. "Building a large annotated corpus of english: The Penn treebank", Computational Linguistics, Vol.19, pp.313-330, 1993.

[4] Burnard, L. The British National Corpus(BNC) Users Reference Guide, 2000.

[5] Atalay, N. B., Oflazer, K. and Say, B. "The annotation process in the Turkish treebank," Proceedings of the EACL Workshop on Linguistically Interpreted Corpora, Budapest, Hungary, 2003.

[6] 임준호, 박소영, 곽용재, 임해창, 김의수, 강범모, "구문패턴을 이용한 반자동 구문분석 말뭉치 구축도구", 제14회 한글 및 한국어정보처리 학술발표 논문집, pp.343-350, 2002.

[7] T. Morton and J. LaCivita, "WordFreak: An open tool for linguistic annotation," Proceedings of the NAACL, pp.17-18, 2003.

[8] D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, and M. Vilain, "Mixed-Initiative Development of Language Processing Systems," Proceedings of the ANLP, pp.348-355, 1997.

[9] T. Brants and O. Plaehn, "Interactive corpus annotation," Proceedings of the Second International Conference on Language Resources and Engineering (LREC 2000), pp.453-459, 2000.

[10] J. Carletta, D. McKelvie, A. Isard, A. Mengel, M. Klein, and M. B. Mller, "A generic approach to software support for linguistic annotation using XML," G. Sampson & D. McCarthy (Eds.), Readings in Corpus Linguistics, Continuum International, 2002.

[11] C. Laprun, J. G. Fiscus, J. Garofolo, and S. Pajo, "A practical introduction to ATLAS," Proceedings of the Third International Conference on Language Resources and Evaluation, 2002.

[12] D. Reidsma, N. Jovanovic, D. Hofs, Designing Annotation Tools based on Properties of Annotation Problems, CTIT Technical Reports TR-CTIT-04-45, University of Twente, The netherlands, 2004.

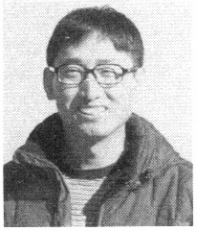
[13] H., Gaifman "Dependency systems and phrase-structure systems," Information and Control, Vol.8, pp.304-337, 1965.

[14] S. Höfler, Link2Tree: A Dependency-Constituency Converter, Ph.D. Dissertation, Institute of Computational Linguistics University of Zurich, 2002.

[15] C. Kim, M. Hong, Y. Huang, Y. K. Kim, S. I. Yang, Y. A. Seo, and S.-K. Choi, "Korean-Chinese Machine Translation Based on Verb Patterns," Proceedings of The 5th Conference of the Association for Machine Translation in the Americas, pp.94-103, 2002.

[16] E. Brill, "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging", Computation Linguistics, Vol.21, No.4, pp.543-565, 1995.

[17] 김재훈, "품사 태깅 시스템의 신뢰도 측정", 정보처리학회논문지 B, 제8-B권, 제4호, pp.365-372, 2001.



김 재 훈

e-mail : jhoon@mail.hhu.ac.kr

1986년 계명대학교 전자계산학과(학사)

1988년 한국과학기술원 전산학과(공학석사)

1996년 한국과학기술원 전산학과(공학박사)

1988년~1997년 한국전자통신연구원 선임연구원

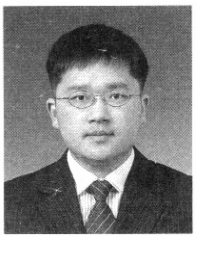
1997년~1999년 한국해양대학교 컴퓨터공학과 전임강사

2000년~2002년 2월 한국과학기술원 첨단정보기술연구센터 연구원

2001년~2002년 2월 USC Information Sciences Institute 방문연구원

1999년~현재 한국해양대학교 컴퓨터공학과 부교수

관심분야 : 자연언어처리, 한국어정보처리, 정보검색, 정보추출



박 은 진

e-mail : bakeunjin@bada.hhu.ac.kr

2003년 한국해양대학교 자동화정보공학부 (학사)

2002년~2004년 (주) 블루코드테크놀로지 사원

2004년~현재 한국해양대학교 컴퓨터공학과 석사과정

관심분야 : 자연언어처리, 한국어정보처리, 정보검색, 정보추출