한국마린엔지니어링학회지 제37권 제2호, pp. 227～235, 2013. 3 (ISSN 2234-8352 Online) / (ISSN 2234-7925 Print)
Journal of the Korean Society of Marine Engineering http://dx.doi.org/10.5916/jkosme.2013.37.2.227

한국어 품사 부착 말뭉치의 오류 검출 및 수정

최명길[1] · 서형원[2] · 권홍석[3] · 김재훈†

(원고접수일 : 2013년 2월 5일, 원고수정일 : 2013년 2월 15일, 심사완료일 : 2013년 2월 28일)

Detecting and correcting errors in Korean POS-tagged corpora

Myung-gil Choi[1] · Hyung-Won Seo2[2] · Hong-Seok Kwon[3] · Jae-Hoon Kim†

요약: 품사 부착 말뭉치의 품질은 품사 부착기를 개발하는데 있어서 매우 중요한 역할을 수행한다. 그러나 세종 말뭉치를 비롯하여 한국에서 구축된 많은 품사 부착 말뭉치들은 여전히 다양한 형태의 오류를 포함하고 있다. 이런 오류들을 살펴보면 품사 부착 오류는 물론이고 철자 오류, 문자의 삽입 및 삭제 등 매우 다양하다. 본 논문에서는 오류 패턴을 이용하여 품사 부착 오류를 검출하고 이를 효과적으로 수정하는 도구를 개발한다. 제안된 방법과 도구를 이용해서 오류를 수정할 경우 평균 9배 이상 빠르게 오류를 수정할 수 있어서 이 방법이 매우 효과적인 방법임을 확인할 수 있었다.

주제어: 품사 부착 말뭉치, 오류 수정, 오류 검출, 말뭉치 수정 도구

Abstract: The quality of the part-of-speech (POS) annotation in a corpus plays an important role in developing POS taggers. There, however, are several kinds of errors in Korean POS-tagged corpora like Sejong Corpus. Such errors are likely to be various like annotation errors, spelling errors, insertion and/or deletion of unexpected characters. In this paper, we propose a method for detecting annotation errors using error patterns, and also develop a tool for effectively correcting them. Overall, based on the proposed method, we have hand-corrected annotation errors in Sejong POS Tagged Corpus using the developed tool. As the result, it is faster at least 9 times when compared without using any tools. Therefore we have observed that the proposed method is effective for correcting annotation errors in POS-tagged corpus.

Keywords: POS-tagged corpus, Error correction, Error detection, Corpus annotation/correction tool

1. 서 론

자연언어처리 분야에서는 대량의 학습 자료를 사용해서 보다 쉽고, 지능적이며, 빠르게 시스템을 개발하고 있다. 자연언어처리 분야에서 대량의 학습 자료를 일반적으로 언어정보 부착 말뭉치라고 한다. 한국어 정보처리를 위해도 다양한 말뭉치**[1]-[3]**가 구축되었으며, 이 중에 한국어 정보처리 연구자가 쉽게 그리고 널리 이용할 수 있는 말뭉치가 세종 말뭉치**[3]**이다. 세종 말뭉치는 원시 말뭉치, 형태분석 말뭉치, 구문분석 말뭉치 등을 포함하고 있다. 특히 세종 형태분석 말뭉치에는 형태소에 품사가 잘못 부착되었거나, 문장 내에서 단어가 잘못 분리된 경우, 그리고 불필요한 단어가 삽입된 경우나 단어가 삭제되는 경우 등의 오류를 포함하고 있다**[4][5]**. 이러한 오류들이 포함된 말뭉치를 학습 자료로 사용할 경우 품사 부착기 등과 같은 자연언어처리 시스템의 좋은 성능을 기대할 수

†교신저자: (606-080) 부산광역시 영도구 태종로 727,
한국해양대학교 IT공학부, E-mail: jhoon@hhu.ac.kr, Tel: 051-410-4574
1 금호마린테크, E-mail: cmg5478@naver.com, Tel: 051-265-8984
2 한국한국해양대학교 컴퓨터공학과, E-mail: wonn24@gmail.com, Tel: 051-410-4896
3 한국한국해양대학교 컴퓨터공학과, E-mail: hong8c@naver.com, Tel: 051-410-4896

없다. 또한 이러한 오류는 다양한 패턴으로 나타날 뿐 아니라 그 수가 매우 많아서 그 오류를 일관성 있게 수정하기란 여간 어려운 일이 아니다. 또한 이러한 오류를 수정하기 위해서는 많은 인력과 시간이 필요하며, 결과적으로 많은 비용이 들게 된다. 그리고 많은 인력을 동원하여 수정한 말뭉치에 또 다른 오류가 발생할 수도 있다**[5][6]**. 왜냐하면 여러 사람들이 동시에 작업하므로 수정의 일관성을 유지하기란 그다지 쉽지 않기 때문이다.

이와 같은 문제점을 해결하기 위해서 본 논문에서는 품사 부착 말뭉치로부터 오류 유형을 분석하고 그 결과에 따른 오류 수정 방법을 제시하고 효율적으로 수정하기 위한 도구를 개발한다. 본 논문에서 오류 검출 방법으로 형태소 생성에 기반한 오류 패턴을 이용한다. 이 방법은 주어진 어절과 형태소 분석 결과의 형태소 생성 결과가 서로 다를 경우, 해당 어절을 오류 가능 어절로 제시하고, 이 어절이 오류이면 오류 패턴을 생성하여 다음에 같은 유형의 오류를 자동으로 검출할 수 있도록 한다. 오류 수정 방법은 GUI(graphical user interface)를 통하여 수동으로 수정되며 가능한 한 반복적인 작업은 수행하지 않도록 설계되었다. 특히 본 논문에서는 일관성 유지를 위해 데이터베이스를 이용해서 모든 정보를 작업자들이 실시간으로 공유할 수 있도록 하였다.

논문의 구성은 다음과 같다. 2장에서 말뭉치 오류 수정 방법 및 도구 그리고 형태소 생성에 대해서 간단히 살펴보고, 3장에서 한국어 품사 부착 말뭉치에서 오류를 분석하여 검출 방법을 제시한다. 4장에서 오류 수정 방법을 바탕으로 구현한 오류 수정 도구에 대하여 설명한다. 5장에서는 오류 수정 도구의 성능에 대하여 평가한다. 마지막으로 6장에서 결론을 맺고 향후 연구 방안에 대하여 방향을 제시한다.

2. 관련 연구

2.1 말뭉치 오류 검출 및 수정

말뭉치가 자연언어처리 시스템의 학습 자료로 널리 사용되나, 다양한 형태의 오류들도 함께 존재한다**[5][7][8]**. 오류가 많지 않을 경우에는 학습 시스템에 커다란 해가 되지 않지만 오류가 많을 경우에는 전혀 엉뚱하게 학습되어 예기치 못한 결과를 얻을 수도 있다. 따라서 이와 같은 오류를 최소화하고자 하는 노력은 꾸준히 이루어지고 있다**[6][9][10]**. 말뭉치 내에 포함된 오류를 찾아내는 오류 검출 방법은 다양하게 제안되었으며, Sparse Markov Transducer를 이용하여 오류에 대한 변칙 탐지 방법**[11]**, SVMs(Support Vector Machines)을 이용한 exceptional elements 검출 방법**[12]**, Directed Treebank Refinement(DTR)을 이용한 오류 검출 방법**[13]**, Modular neural network를 이용한 오류 검출 방법**[14]** 등이 있다. 한국어에 대해서는 오류 검출에 대한 연구가 활발히 진행되지는 않았지만 일부의 연구**[4][5]**에서 세종 말뭉치를 대상으로 진행되었다. 이들은 오류 유형을 정의하여 오류를 검출하였고, 이러한 오류들을 검출하기 위하여 모든 어절에 대하여 어절 분할 오류 검사, 철자 오류 검사, 표지 부착 오류 검사, 일관성 오류 검사를 실시하여 오류인지 아닌지 확인하였다.

2.2 말뭉치 수정 도구

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. 이러한 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 수는 없으며, 더구나 새로운 응용분야(예: 웹 문서 분류)가 등장되었을 때, 기존의 도구를 그대로 사용할 수 없다**[15]**. 한국어의 경우 부분적으로 언어지식 구축 및 수정 도구를 개발하였다**[16]-[18]**. 이와 같은 도구들은 특정 말뭉치와 연구에 매우 밀접한 관계를 가지고 있기 때문에 다른 말뭉치와 연구에 적용하기 매우 어려울 뿐 아니라 공개된 시스템이 없어서 사용할 수도 없다. 외국어 경우에도 다양한 언어지식의 구축 도구를 개발하였다**[19]-[21]**. 이들 도구도 특정 영역이나 특정 프로젝트를 위해서 구축된 것이다. 일부의 시스템**[19]**은 공개되었지만 지금은 사용할 수 없을 뿐 아니라 너무 오래 전에 개발되어 현재의 기술을 적용하기 매우 어렵게 되어 있다.

2.3 형태소 생성

형태소 생성이란 일련의 형태소들로부터 어절을 생성하는 것이다. 대부분의 형태소 생성은 단순한 형태소들의 연접(concatenation)으로 가능하다. 예를 들면 (′학교′, ′을′)은 "학교를"로 생성된다. 그러나 용언의 경우에는 동사나 형용사와 어미가 결합하면서 다양한 음성 현상이 발생한다. 예를 들면 (′아름답다′, ′어′)는 “아름다워”로 생성된다. 형태소 생성에 대한 연구로는 **[22][23]**이 있다.

3. 한국어 품사 부착 말뭉치에서 오류 검출

3.1 한국어 품사 부착 말뭉치의 특징

한국어 품사 부착 말뭉치는 영어 품사 부착 말뭉치와 다르게 어절과 형태소 분석 결과를 함께 저장해야 한다**[1]**. 어절은 형태소 분석 결과의 형태소 생성 결과로 볼 수 있다. 그러나 일반적으로 어절에 대한 형태소 분석 결과가 모호하므로 품사 부착 말뭉치에서는 정확한 형태소 분석 결과를 저장하고 있어야 한다. 따라서 대부분의 한국어 품사 부착 말뭉치**[1]-[3]**들은 **Figure 1**과 같이 어절과 형태소 분석 결과를 함께 저장하고 있다. **Figure 1**에서 <p>는 문장을 표시하는 태그이다. 첫 번째 열은 어절 번호이며 말뭉치 내에서 구별되는 번호를 가지고 있다[1]. 두 번째 열은 어절 자체이고 세 번째 줄은 그 어절의 형태소 분석 결과이다. 세 번째 열에서 형태소는 ‘+’로 구분되며 각 형태소는 형태소 자신과 품사로 구성되어 있으며 세종말뭉치의 품사 집합은 **[3]**을 참조하기 바란다.

3.2 세종 형태분석 말뭉치의 오류 분석

본 논문에서는 세종 형태분석 말뭉치의 오류를 분석하고 그 결과를 바탕으로 오류 검출 방법을 제안하고자 한다**[24][25]**. 세종 형태분석 말뭉치의 3%에 해당하는 450,000 어절에 대하여 오류를 분석하였으며, 그 결과 29,253개의 오류가 발견되어 대략 6.5%의 오류를 포함하고 있음을 알 수 있었다. **Table 1**에서 그 일부를 보여주고 있다. **Table 1**에서 대부분의 오류는 부가적인 설명 없이도 충분히 이해할 수 있다. ‘한자

```
<p>
BTAA0001-00000164 빙수기,        빙수기/NNG + ,/SP
BTAA0001-00000165 샤베트기,      샤베트기/NNG + ,/SP
BTAA0001-00000166 얼음           얼음/NNG
BTAA0001-00000167 물통           물통/NNG
BTAA0001-00000168 등             등/NNB
BTAA0001-00000169 여름           여름/NNG
BTAA0001-00000170 주방           주방/NNG
BTAA0001-00000171 생활           생활/NNG
BTAA0001-00000172 용품이         용품/NNG + 이/JKS
BTAA0001-00000173 인기다.        인기/NNG + 이/VCP + 다/EF + ./SF
</p>
<p>
BTAA0001-00000174 특히           특히/MAG
BTAA0001-00000175 여름철         여름철/NNG
BTAA0001-00000176 더위를         더위/NNG + 를/JKO
BTAA0001-00000177 식히는데       식히/VV + 는데/EC
BTAA0001-00000178 최고인         최고/NNG + 이/VCP + ㄴ/ETM
BTAA0001-00000179 빙수를         빙수/NNG + 를/JKO
BTAA0001-00000180 만드는         만들/VV + 는/ETM
BTAA0001-00000181 기계는         기계/NNG + 는/JX
BTAA0001-00000182 소비자들이     소비자/NNG + 들/XSN + 이/JKS
BTAA0001-00000183 가장           가장/MAG
BTAA0001-00000184 많이           많이/MAG
BTAA0001-00000185 찾고           찾/VV + 고/EC
BTAA0001-00000186 있는           있/VX + 는/ETM
BTAA0001-00000187 품목.          품목/NNG + ./SF
</p>
```

Figure 1: The part of Sejong POS tagged corpus as an example of the Korean POS tagged corpus

변환 오류’는 말뭉치를 구축하는 과정에서 한자 정보가 한글로 변환된 형태로 품사 정보가 부착되었다. 또 ‘다중품사 부착 오류’는 말뭉치 구축 과정에서 품사 부착기의 출력이 수정되지 않은 상태로 존재하는 오류들이고, ‘영어 대소문자 오류’는 어절의 영어 철자와 형태소 분석 결과의 영어 철자가 다를 경우 오류이다.

3.2. 형태소 생성에 의한 오류 검출

3.1절에서 언급했듯이 한국어 품사 부착 말뭉치에는 어절과 형태소 분석 결과를 모두 포함하고 있다. 먼저 형태소 분석 결과에 포함된 형태소들을 결합하여 어절을 생성하면 여러 개의 어절이 생성된다. 그 중에 하나가 원래의 어절과 다르다면 오류일 가능성이 매우 높다. 본 논문에서 형태소 생성이 목적이 아니므로 직접 형태소를 생성하는 것이 아니라 원래의 어절과 문자열의 차이를 구하고, 그 차이와 주변 문맥

[1] 세종 말뭉치 외에 다른 말뭉치들은 일반적으로 어절 번호를 별도로 관리하지 않는다.

Table 1: Examples of annotation errors in the Sejong POS tagged corpus

오류 유형	어절의 예	오류 수정의 예시	
		오류	수정
분석	끔찍한	끔직/XR+ 하/XSA+ ㄴ/ETM	끔찍/XR+ 하/XSA+ ㄴ/ETM
한자 변환	다소비(다소비)	다소비(다소비)	다소비(多消費)
특수 문자	㈜	(/SS+ 주/NNG+)/SS	㈜/SS
다중품사 부착	여권에서	여/NNG+ 권/XSN/NNG + 에서/JKB	여/NNG+ 권/NNG + 에서/JKB
어절 철자	혜택을	혜택을	혜택을
띄어쓰기	디자인세계	디자인세계	디자인 세계
형태소 분리	국내의	국/NNG+ 내의/NNG	국내/NNG+ 의/NNG
영어 대소문자	content	Content/SL	content/SL

이 정당한 형태소 생성이라면 오류로 추정하지 않는다. **Table 2**에서 그 예를 보여주고 있다.

Table 2: An example of morphological generation patterns detected through identification of string difference

왼쪽 문맥	불일치	오른쪽 문맥
ㅇㅏㄹㅡㅁㄷㅏ	ㅇㅜ	ㄴ
ㅇㅏㄹㅡㅁㄷㅏ	ㅂ	ㄴ

Table 2에서 어절 '아름다운'과 이에 대한 형태소 분석 결과에 속한 형태소들 '아름답+ㄴ'를 자소 단위로 비교하여 그 차이를 보이고 있다. 여기서 형태소 생성 패턴은 좌우의 한 자소를 형태소 생성 패턴으로 저장한다. 이렇게 정당한 형태소 생성 패턴을 저장하여 품사 부착 말뭉치의 오류를 검출한다. 한국어는 통상적으로 한 어절에 하나의 이상의 형태적 변이가 발생할 수 있으나 본 논문에서는 하나의 형태적 변이만 가능한 것으로 가정하였다.

4. 한국어 품사 부착 오류의 수동 도구

4.1. 설계 원칙

본 논문에서는 **[15]**에서 제안한 말뭉치 수정 도구의 요구사항을 만족하도록 설계하였으며 아래와 같이 요약된다.

- **이식성**: 이식성을 높이기 위해서 Java 언어를 채택하였다. Java는 Windows뿐만 아니라 Linux에서도 잘 실행된다.
- **학습 편의성**: 처음 사용하는 사용자도 도구의 사용에 전혀 부담을 느끼지 않도록 설계되었으며 혹시라도 부족한 점이 있다면 도움말을 참조하도록 하였다. 특히 말풍선을 이용해서 익숙지 않은 UI에 대한 충분한 설명을 제공하도록 노력하였다.
- **사용자 중심 인터페이스**: 사용자들이 작업할 때 가능하면 적은 노력으로 큰 결과를 얻을 수 있도록 설계하였으며 문장, 어절 형태소 등 언어 정보의 단위에 따라서 각각 다른 작업을 할 수 있도록 설계하였다.
- **시각화:** 대부분의 내용은 직접 눈으로 확인할 수 있도록 하였으며 코드로 표현된 정보는 사용자의 요구에 따라서 적절히 선택할 수 있도록 하였다.
- **(반)자동화:** 가능하다면 자동적으로 언어 정보를 부착할 수 있도록 설계되어 사용자들의 수정 작업을 최소화하도록 설계하였다.
- **집중화:** 작업의 집중도를 높이기 위해 다양한 색깔을 이용해서 문제가 될 수 있는 언어 정보를 한눈에 확인할 수 있도록 하였다.
- **사용자 모델링**: 사용자에 따라 사용할 수 있는 메뉴를 다르게 제공하여 자신의 수준에 맞게 작업할 수 있는 환경을 제공하였다.
- **부가 정보 지원**: 주석자, 수정 날짜, 수정 이력 등이 많은 부가정보를 관리하도록 하였다.

4.2. 시스템 구조

Figure 2는 한국어 품사 부착 말뭉치의 수정 도구(TagBench)의 전체 구조이다**[24][25]**. 관리자는 작업

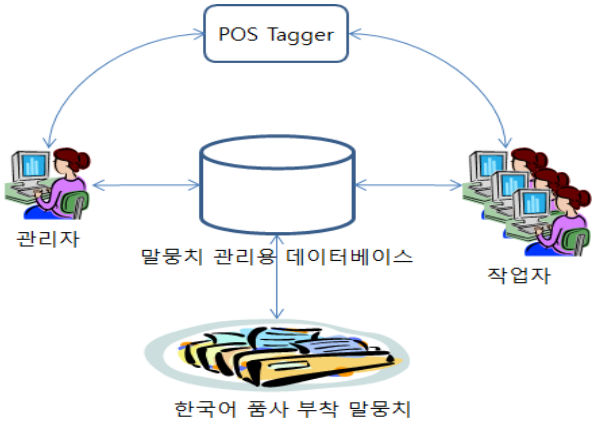

Figure 2: System configuration for detecting and correcting errors in Korean POS-tagged corpora

자에게 수정할 말뭉치를 배정한다. 작업자는 4.3절에서 설명한 사용자 인터페이스(GUI)를 통해서 배정된 말뭉치의 오류를 수정하고 그 결과를 데이터베이스에 저장하며, 그 시나리오는 **Figure 3**과 같다. 작업자에게 처음으로 배정된 문장은 모두 'untagged'로 초기화되어 있다. 정상적으로는 한 문장을 선택하여('ongoing') 오류를 수정하고 이를 저장한다('tagged').그러나 작업자가 복잡한 언어 현상 등으로 수정이 어려울 경우에는 전문가에게 질문할 수 있다('asked'). 전문가는 'asked'된 문장에 주석으로 기재하여 초기 상태('untagged')로 되돌린다. 또한 작업자가 수정 중에 수정을 완료하지 않고 장시간 자리를 비울 경우, 수정 작업을 완료하지 않고 임시 저장할 수도 있으며('deferred') 제자리로 돌아왔을 때 임시 저장된 문장을 바로 수정할 수 있도록 하였다.

저장된 모든 정보는 모든 사용자(관리자, 작업자,

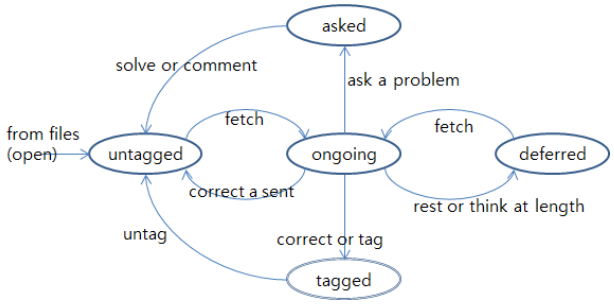

Figure 3: Data flow for detecting and correcting errors in Korean POS-tagged corpora

전문가)에 의해서 언제 어디서나 손쉽게 확인하고, 작업을 진행하는 데 참고할 수 있어서 작업의 능률을 크게 개선할 수 있다. 이와 같은 방법으로 모든 문장의 오류가 수정되면 그 말뭉치에 부착된 정보를 각각의 품사 부착 말뭉치 형식에 따라 출력할 수 있다. 현재는 세종 형태분석 말뭉치와 ETRI 의존구조 말뭉치 형식으로 출력할 수 있으나 다른 형식도 출력 모듈을 수정하면 쉽게 확장할 수 있다.

4.3. 사용자 인터페이스(GUI)

Figure 4는 본 논문에서 구현된 사용자 인터페이스이며 크게 다섯 부분(그림에 ❶ ~ ❺로 표시된 부분)으로 구성되어 있다. **Figure 4**에서 ❶로 표시된 부분은 현재문장창이라고 하며, 현재 작업 문장을 보여주고 있다. 작업자가 문장을 좀 더 쉽게 이해할 수 있도록 전후 문장도 함께 보여준다. 특히 이 문장 자체에 띄어쓰기 오류, 철자 오류 등이 포함되어 있을 경우, 문장 자체를 수정하고 수정된 문장을 한국어 형태소 품사 부착기를 통해서 재분석할 수 있다. 이 경우 작업자가 수정된 부분을 일일이 수정할 필요가 없게 되어 효율적으로 수정이 가능하다. ❷로 표시된 부분은 작업자들이 자주 사용하는 수정 행위를 신속하게 수행할 수 있도록 단추(buttons)으로 구성하여 모아 두었다. 물론 익숙한 사용자들은 단축키(hot keys)도 사용할 수 있다. ❸으로 표시된 부분은 문장창이라고 하며 전체 파일 중에서 얼마나 작업이 진행되었는지를 파악할 수 있다. 또한 문장들의 상태(untagged, tagged, asked 등)을 파악할 수 있으며 주석이 포함되어 있는지 등 문장의 상태를 쉽게 파악할 수 있다. ❹로 표시된 부분은 분석창이라고 하며, 실질적으로 오류 수정 작업은 이곳에서 이루어진다. 작업자의 집중도를 높이기 위해서 오류일 가능성이 높은 형태소는 빨간색으로 표시하여 쉽게 오류에 쉽게 접근할 수 있도록 하였다. 이 오류들은 3장에서 설명한 방법으로 검출되며 오류 검출과 수정의 자동화는 작업이 진행될수록 정확하고 빠르게 수행할 수 있다. ❺로 표시된 부분은 주석창이라고 하며 이 창을 통해서 문장, 어절, 형태소의 품사 부착이나 기타 부가 정보를 기재하고 다른 작업자와 공유할 수 있다. 기타 상태 표시나 메뉴 등은 일반적인 프로그램들과 동일하게 구성하였다.

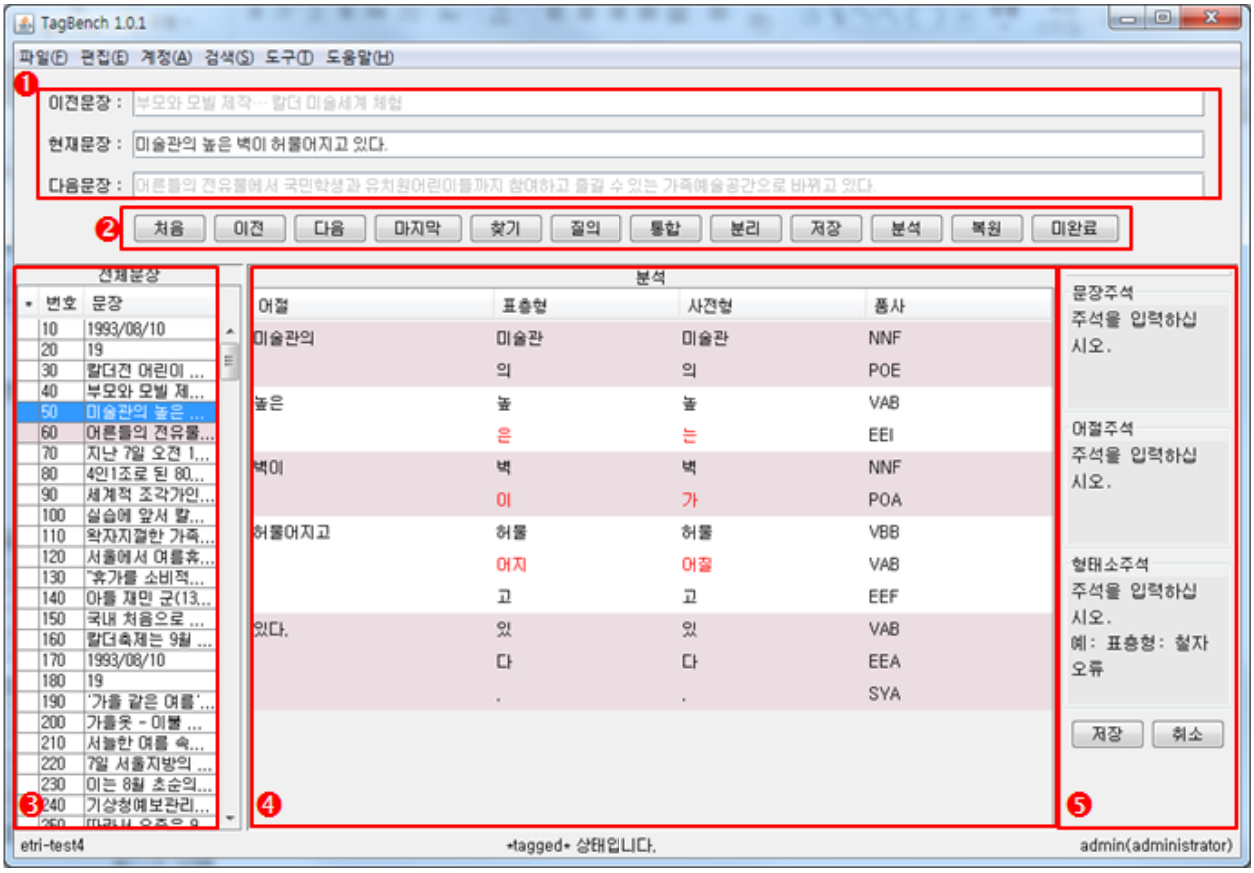

Figure 4: Graphical user interface; ❶ Sub-window for the current sentence, ❷ Useful buttons for correcting errors, ❸ Sub-window for whole sentences, ❹ Main working window, ❺ Sub-window for comments and communication between workers

4.4. 오류 패턴 관리

작업자가 오류를 수정하고 데이터베이스에 저장할 때, 어절과 형태소 분석된 결과가 문자적으로 다를 경우 자동으로 검출하여 오류 패턴으로 등록한다. 문자열이 서로 다르더라도 음운 현상이나 용언의 불규칙 현상이 포함되어 있을 경우에는 오류가 아니므로 이를 작업자가 확인하도록 하였다. 만약 작업자가 실수로 오류 패턴을 저장하였다 하더라도 관리자가 이를 찾아서 수정할 수 있다.

4.5. 구현 환경

한국어 품사 부착 오류의 수정 도구는 Windows7 하에서 개발되었으나 Java로 개발되어 운영체제에 관계없이 어디서나 동작할 수 있다. 데이터베이스는 MySQL server 5.5[2]을 사용하였고, 프로그램이 방대하고 여러 개발자들이 함께 개발하므로 효율적인 프로그램 관리를 위해서 CollabNet Subversion Edge 3.0[3]를 사용하였다. 현재 한국어 품사 부착 말뭉치 중에서 세종 말뭉치와 ETRI 의존구조 말뭉치에 대해서 수정이 가능하도록 입출력 모듈이 구현되어 있다.

5. 실험 및 평가

본 장에서는 개발된 한국어 품사 오류 수정 도구의 성능을 평가하기 위해 세종 형태분석 말뭉치[4]를 대상으로 두 가지 실험을 수행하였다. 첫 번째 실험은 오류 검출에 관한 것이고 두 번째 실험은 오류 수정에 관한 것이다.

첫 번째 실험에서 오류 수정 도구를 사용하여 단계

[2] http://dev.mysql.com/downloads/mysql/

[3] http://www.collab.net/downloads/subversion

[4] 오류를 비교적 쉽게 찾을 수 있는 세종 형태분석 말뭉치를 실험 대상으로 선정했다.

Table 3: Error detection speed according to use of the developed tool

실험자	오류 검출 시간	
	오류 수정 도구 미사용	오류 수정 도구 사용
A	35 min	3 min 33 sec
B	45 min	5 min 08 sec
C	32 min	4 min 21 sec
평균	**37 min**	**4 min 20 sec**

적으로 오류 패턴을 개선하는 경우와 작업자가 말뭉치에서 일일이 오류를 검출하는 경우의 시간을 관찰해 보았다. 대상 문장은 100문장이며 3명의 실험자에 의해서 수행되었으며 그 결과는 **Table 3**과 같다. **Table 3** 에서 보는 바와 같이 100문장에서 오류를 작업자가 직접 검출하는 경우, 평균 37분 정도 소요되었으나 오류 수정 도구를 이용한 경우, 평균 4분 정도 소요되었다. 이 결과를 볼 때, 대략 9배 정도 빠르게 오류를 검출할 수 있었다. 더구나 작업자가 직접 오류를 검출할 경우는 오류를 찾지 못하는 경우도 발생하였다. 오류의 검출이 자동적으로 이루어짐으로 작업의 효율뿐 아니라 작업의 집중도와 작업 시간도 늘릴 수 있다. 일반적으로 장시간 동안 말뭉치를 구축하면 작업자 집중도가 크게 떨어져서 오류를 그대로 방치할 가능성이 매우 높다. 본 논문에서 제안된 방법은 대부분의 오류를 시스템이 검출하므로 이 문제가 크게 개선되었다.

두 번째 실험에서는 세종 형태분석 말뭉치에서 10개의 파일을 임의로 선정해서 숙련된 작업자가 직접 수정하는데 걸리는 시간을 분석해 보았다. 각 파일은 평균 400 문장으로 구성되었으며, 각 파일의 작업 시간은 **Table 4**와 같다. **Table 4**에서 보는 바와 같이 작업이 진행될수록 작업 시간이 단축됨을 알 수 있다. 이는 작업의 숙련도가 증가된다고 볼 수 있다. 그러나 작업의 숙련도보다 오류 수정 패턴이 축적되어 작업자가 직접 오류를 수정해야 하는 횟수가 줄어든 결과이다. 이 결과는 오류 패턴이 어느 정도 축적되면 400 문장에 대해서 평균 15분 정도의 수정 시간이 소요된다. 이는 정확하게 다른 도구와 비교할 수는 없지만 경험에 의하면 비교적 빠른 시간이다.

위의 두 실험 결과를 보아 본 논문에서 제안된 오류 패턴과 수정 도구는 품사 부착 오류를 수정하는데 매우 유용함을 알 수 있었다. 본 논문은 주로 세종 말뭉치를 대상으로 실험해 보았지만 대부분의 한국어 품사 부착 말뭉치가 비슷한 구조를 가지고 있으므로 다른 한국어 품사 부착 말뭉치에도 그대로 적용할 수 있을 것으로 기대된다.

Table 4: Correction time of each file with about 400 sentences, which are selected from Sejong corpus

순번	말뭉치 파일 이름	작업 시간(분)
1	BTBF0269	24
2	BTHO0365	18
3	BTAA0015	16
4	BTJO0444	18
5	BTHO0366	15
6	BTBE0238	14
7	BTHO0442	17
8	BTBE0242	14
9	BTEO0327	15
10	BTAE0204	15
총 작업시간(분)		**168**

6. 결론

본 논문에서는 한국어 품사 부착 말뭉치로부터 오류 유형을 분석하고 그 결과에 따른 오류 수정 방법을 제시하고 효율적으로 수정하기 위한 도구를 개발한다. 본 논문에서 오류 검출 방법으로 형태소 생성에 기반한 오류 패턴을 이용한다. 이 방법은 주어진 어절과 형태소 분석 결과의 형태소 생성 결과가 서로 다를 경우, 해당 어절을 오류 가능 어절로 제시하고, 이 어절이 오류이면 오류 패턴을 생성하여 다음에 같은 유형의 오류를 자동으로 검출할 수 있도록 한다. 오류 수정 방법은 GUI(graphical user interface)를 통하여 수동으로 수정되며 가능한 한 반복적인 작업은 수행하지 않도록 설계되었다. 제안된 방법과 도구를 이용해서 오류를 수정할 경우 평균 9배 이상 빠르게 오류를 수정할 수 있어서 이 방법이 매우 효과적인 방법임을

확인할 수 있었다. 특히 본 논문에서는 일관성 유지를 위해 데이터베이스를 이용해서 모든 정보를 작업자들이 실시간으로 공유할 수 있도록 하였다.

오류 패턴만으로 모든 오류를 검출할 수는 없었다. 따라서 완전히 오류를 수정하기 위해서는 새로운 오류 검출 방법이 필요하다. 따라서 향후 연구로서 기계학습 방법을 이용한 오류 검출 방법을 수정 도구에 접목한다면 오류의 검출이 더욱 정확할 뿐 아니라 다양한 오류들을 쉽게 찾을 수 있을 것으로 생각된다.

후 기

본 연구는 지식경제부의 지식경제 기술혁신사업(10041807)과 한국전자통신연구원 위탁과제인 “자동번역 지식 구축 도구에 관한 연구”의 부분적인 지원으로 수행되었으며, 최명길의 석사학위논문**[25]**과 학술대회 논문**[24]**를 확장하고 개선한 것이다.

참고문헌

[1] J.-H. Kim and G. C. Kim, Guideline on Building a Korean Part-of-Speech Tagged Corpus: KAIST Corpus, Technical Report CS-TR-95-99, Department of Computer Science, KAIST, 1995 (in Korean).

[2] C.-H. Han and N.-R. Han, Part of Speech Tagging Guidelines for Penn Korean Treebank, Technical Report IRCS Report 01-09, Institute for Research in Cognitive Science, University of Pennsylvania, 2001.

[3] H.-G. Kim, 21st Century Sejong Project - Construction of the Primary Data of the Korean Language, Research Report NIKL 2007-01-10, National Institute of the Korean Language, 2007 (in Korean).

[4] M. Lee, H. Jung, W.-K. Sung, and D.-I. Park, “Verification of POS tagged corpus,”, Proceedings of the 17th Annual Conference on Human and Cognitive Language Technology, pp. 145-150, 2005 (in Korean).

[5] J.-H. Kim, H.-W. Seo, K.-H. Jeon, and M.-G. Choi, “Error correction methods for Sejong corpus,” Proceedings of the KOSME Spring Conference, pp. 435-436. 2010 (in Korean).

[6] M. Dickinson, Error Detection and Correction in Annotated Corpora. Ph.D. Thesis, The Ohio State University, 2005.

[7] H. Loftsson, “Correcting a PoS-tagged corpus using three complementary methods,” Proceedings of the 12th Conference of the European Chapter of the ACL, pp. 523-531, 2009.

[8] H. Loftsson, J. H. Yngvason, S. Helgadottir, and E. Rognvaldsson, “Developing a POS-tagged corpus using existing tools,” Proceedings of the 12th Conference of the European Chapter of the ACL, pages 523-531, 2009.

[9] H. van Halteren “The detection of inconsistency in manually tagged text,” Proceedings of the 2nd Workshop on Linguistically Interpreted Corpora, 2000.

[10] M. Dickinson and W. D. Meurers, “Detecting errors in part-of-speech annotation,” Proceedings of the 10th conference on European chapter of the Association for Computational Linguistics pp. 107-114. 2003.

[11] E. Eskin, “Automatic corpus correction with anomaly detection,” Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics pp. 148-153, 2000.

[12] T. Nakagawa and Y. Matsumoto, “Detecting errors in corpora using support vector machines,” Proceedings of the 17th International Conference on Computational Linguistics, pp. 709-715, 2002.

[13] T. Ule and K. Simov, “Unexpected productions may well be errors", Proceedings of 4th International Conference on Language Resources and Evaluation, pp. 1795-1798, 2004.

[14] Q. Ma, B.-L. Lu, M. Murata, M. Ichikawa and

H. Isahara, “On-line error detection of annotated corpus using modular neural networks,” Proceedings of the International Conference on Artificial Neural Networks, pp. 1185‑1192, 2001.

[15] R. Reidsma, K. Tomanek, U. Hahn, and A. Rappoport, “Multi-task active learning for linguistic annotations,” Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 861-869, 2008.

[16] B. G. Chang, K. J. Lee and G. C. Kim, “Design and implement of tree tagging workbench to build a large tree tagged corpus of Korean,” Proceedings of the 9th Annual Conference on Human and Cognitive Language Technology, pp. 421-429, 1997 (in Korean).

[17] Y.-H. Noh, H. A. Lee, and G. C. Kim, “A workbench for domain adaptation of an MT lexicon with a target domain corpus,” Proceedings of the 12th Annual Conference on Human and Cognitive Language Technology, pp. 163-168, 2000 (in Korean).

[18] J.-H. Kim and E.-J. Park, “PPEditor: Semi-automatic annotation tool for Korean dependency structure,” The Transaction of the Korean Information Processing Society, vol. 13-B, no. 1, pp. 63-70, 2006 (in Korean).

[19] D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, and M. Vilain, “Mixed-initiative development of language processing systems", Proceedings of the Applied Natural Language Processing Conference, pp. 348~355, 1997.

[20] T. Morton and J. LaCivita, “WordFreak: An open tool for linguistic annotation,” Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pp. 17-18, 2003.

[21] T. Brants and O. Plaehn, “Interactive corpus annotation,” Proceedings of the 2nd International Conference on Language Resources and Engineering, pp. 453-459, 2000.

[22] S. Chung, T. Kim, D. Hwang, and D.-I. Park, “Morphological generation system in English-Korean Machine Translation System MATES/EK,” Proceedings of the Workshop on Research Projects of the Ministry of Science and Technology, pp. 10-13, 1990 (in Korean).

[23] U. C. Choi, D. U. An, K.-S. Choi, and G. C. Kim, “Design and implementation of Korean generator for English-Korean Machine Translation,” Proceedings of the Autumn Conference of KISS, vol. 17, no. 2, pp. 221-224, 1990 (in Korean).

[24] H.-W. Seo, M.-K. Choi, Y.-R. Nam, H.-S. Kwon, and J.-H. Kim, “TagBench : A tool for building large corpora,” Proceedings of the 24th Annual Conference on Human and Cognitive Language Technology, pp. 126-131, 2012 (in Korean).

[25] M.-G. Choi, Developing a Tool for Detecting and Correcting Errors in Sejong POS Tagged Corpus, Master’s Thesis, Department of Computer Engineering, Korea Maritime University, 2012 (in Korean).

[26] J.-H. Kim, A Study on a Corpus Construction Tool for Machine Translation, Research Report, Electronics and Telecommunications Research Institute (ETRI), 2012.