품사태그부착 코퍼스 구축을 위한 한국어 품사태깅 워크벤치

박영찬*, 김남일, 허욱, 남기춘, 최기선
*대전시 유성구 어은동 1 번지, 시스템공학연구소
자연어정보처리연구부 정보검색연구실
대전시 유성구 구성동 373-1 한국과학기술원 전산학과
E-mail : ycpark@seri.re.kr, {nikim,hook,kichun,kschoi}@world.kaist.ac.kr

The Korean Part-of-speech Tagging Workbench for Tagged Corpus Construction

*Young C. Park, Nam-il Kim, Wook Huh, ki-Chun Nam, KeySun Choi
*Dept. of Natural Language Information Processing, Information Retrieval Lab,
SERI, 1 Eoeun-dong Yusong-gu Taejon
Korea Advanced Institute of Science and Technology
373-1 Kusong-dong Yusong-gu Taejon
E-mail : ycpark@seri.re.kr, {nikim,hook,kichun,kschoi}@world.kaist.ac.kr

요 약

한국어의 언어분석을 위한 가공코퍼스의 하나인 품사부착 코퍼스는 형태소 언어분석의 기초가 되는 자료로서 각종 언어분석 모델의 학습자료와 관측자료 또는 검증자료로서 중요한 역할을 한다. 품사부착 코퍼스의 구축은 많은 노력과 시간이 요구되는 어려운 작업이다. 기존의 구축방법은 자동 태거의 결과를 일일이 사람이 확인해 가면 오류를 발견하고 수정하는 단순 작업이었다. 이러한 단순 작업은 한 번 수정된 자동태거의 반복적 오류, 미등록어에 의한 오류 들을 계속적으로 수정해야하는 비효율성을 내포하고 있었다. 본 논문에서는 HMM 기반의 자동 태거를 사용하여 1 차적으로 한국어 문서를 자동 태깅한다. 자동 태깅 결과로부터 규칙기반의 오류 수정을 추가적으로 행한다. 이렇게 구축된 결과를 사용자에게 제시하여 최종 오류를 수정하고 이를 앞으로의 태깅작업에 반영하는 품사부착 워크벤치에 대해 기술한다.

1. 서론

사람들의 의사 소통을 위해 사용되는 음성 혹은 문자 언어를 분석, 이해, 번역 및 생성하는 자연언어 처리(natural language processing, NLP)는 한 언어의 활용도와 정보화의 근간으로 그 중요성이 부각되고 있다. 그 중 자연 언어 분석은 입력된 문서로부터 내부 표현을 도출하는 과정으로 형태소 분석, 품사 태깅, 구문 분석, 의미 분석, 개념 인식, 화용 인식, 숙어 인식 등 여러 단계를 거친다.

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. 형태소 분석은 자연언어가 가지는 중의성(ambiguity)으로 인해 일반적으로 한 입력 어절에 대해 여러 개의 형태소 열을 결과로 제시하게 된다. 또한 사전 내용의 부족이나 신조어, 고유 명사 등으로 인한 미등록어에 대해서는 올바른 분석 결과를 제시하지 못한다. 형태소 분석 결과의 중의성이나 미등록어에 대한 적절한 처리는 다음 분석 단계의 수행 성능 및 정확도에 영향을 주게 된다. 따라서, 미등록어를 인식하여 올바른 품사를 찾아내고 품사의 중의성을 해결하는 과정이 필요하다. 어떤 형태소가 여러개의 품사를 가질 경우, 그 형태소는 품사 중의성(the ambiguith of part-of-speech)이 있다고 한다. 품사 중의성은 형태소의 주변 문맥 등을 봄으로써 줄일 수 있으며, 이렇게 하여 품사 중의성을 해소하는 과정을 품사 태깅(part-of-speech tagging)이라고 한다. 자동 품사 태깅 시스템은 크게 규칙기반과 확률기반 그리고 두 방법을 혼합한 하이브리드 형태가 존재하는데 대체로 92%~95%의 정확률을 보인다. 그러나, 많은 자연언어 처리 시스템은 나머지 5%~8%의 오류로 말미암아 문장의 해석이 실패되거나 잘못된 분석을 할 수 있다.

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

현재의 일반적인 품사부착 코퍼스 구축 방법은 그림 1과 같은 단계들을 거친다 - 형태소 분석, 자동 품사 태깅, 수동 또는 자동 오류 수정. 그림 1에서 형태소 분석기는 문서를 보고 사전 정보를 사용해서 각 어절의 형태소 분석 결과를 출력한다. 자동 품사 태거는 학습 코퍼스로부터 추출한 통계 정보를 이용해서, 형태소 분석 결과들 중 가장 확률이 높은 분석 결과들을 고른다. 마지막으로, 사람이 수동으로 태깅 결과의 오류들을 수정하거나, 자동 오류 수정 프로그램을 이용해서 오류들을 수정한다.

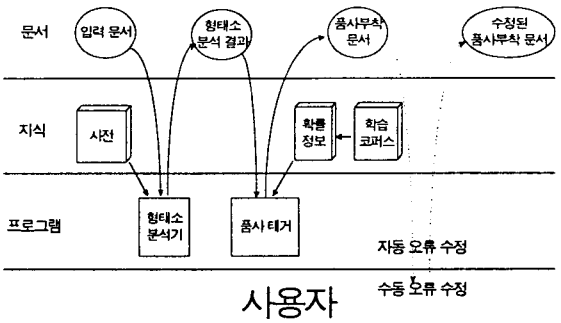

그림 1. 일반적인 품사부착 코퍼스 구축 방법

첫 단계인 형태소 분석에서 미등록어 문제가 발생한다. 미등록어는 태깅 오류의 주원인이 된다. 그리고, 통계 정보를 이용하는 자동 품사 태깅은 지역적인 의존관계를 넘어서는 문제를 해결할 수 없기 때문에 오류가 발생한다. 이러한 태깅 결과의 오류를 수동으로 수정하는 방법은 많은 수작업이 필요하고, 자동으로 수정을 하면 여전히 오류가 존재하게 된다.

본 논문에서는 확률기반의 자동 태깅 시스템을 근간으로 하고, 규칙기반과 사용자 수정정보를 사용하는 품사부착 워크벤치에 대해 기술한다.

2. 품사 태깅 오류의 몇가지 원인

품사부착 코퍼스 구축 과정에서 자동 품사태깅의 오류를 보다 쉽게 수정하기 위해서, 먼저 몇 가지의 오류 발생 원인에 대해 살펴본다.

2.1 품사 미등록어

대부분의 한국어 형태소 분석기는 사전을 기반으로 분석이 이루어 진다. 사전을 이용할 경우 미등록어란 사전에 등록되지 않은 말로 정의한다[1].

대부분의 형태소 분석기는 분석에 실패하면 미등록어가 포함되어 있다고 판단을 하는데, 이 때 미등록어는 고유명사 혹은 명사로 가정한다. 그 이유는 개방어(open class word)에 속하는 대부분의 단어가 이 명사류에 속하기 때문이다. 물론, 대부분의 미등록어는 명사류이다. 그러나, 명사류가 아닌 다른 부류의 단어들도 미등록어로서 쉽게 접할 수 있다. 특히 한국어와 같은 형용적 표현이 발달된 언어에서는 더욱 더 그와 같은 현상을 자

주 접할 수 있다[1].

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', '좌+우명+이나', '좌우+명+이나',... 등으로 분리할 수 있고, 각각에 대해 개방어에 속하는 모든 품사들을 할당해야 한다. 즉, 한 어절에 대한 형태소 분석 결과의 수가 상당히 많아지게 된다. 또, 미등록어에 대한 형태소 정보가 학습 코퍼스에 없기 때문에, 형태소 확률은 0에 가깝고, 단지 태그열의 확률만으로 태깅을 하게 된다. 따라서, 미등록어를 포함하는 어절에 대해서는 옳은 태깅 결과를 기대하기가 어렵다.

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

2.2 마르코프 가정

태깅 문제는 다음과 같이 정의 할 수 있다.

$$\phi(W) = \arg\max_T P(T|W) \quad (1)$$

$$= \arg\max_T \frac{P(W|T)P(T)}{P(W)} \quad (2)$$

$$= \arg\max_T P(W|T)P(T) \quad (3)$$

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 T를 구하는 것을 의미하고, w_i는 i번째 위치한 단어이고, t_i는 그 단어에 해당하는 품사이다. 식(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[5].

$$P(W|T) \cong \prod_{i=1}^{n} P(w_i|t_i) \quad (4)$$

$$P(T) \cong \prod_{i=1}^{n} P(t_i|t_{i-h,i-1}) \quad (5)$$

식(4)와 식(5)를 식(3)에 적용하면, 다음과 같이 된다.

$$\phi(W) = \arg\max_T \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-h,i-1}) \quad (6)$$

식(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, 태깅 식에서는 '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.

2.3 의미 중의성

한 형태소나 어절이 여러 의미로 사용되는 경우, 이들간의 차이는 사용된 의미를 파악해야만 분석할 수 있다.

형태소의 예로서, '말(言)'과 '말(馬)'이 있다. '말(言)'은 동작성 보통명사(ncpa)이고, '말(馬)'은 비동작성 보통명사(ncn)인데, 형태태깅 단계에서는 이 차이를 구별할 수 없다.

어절의 예로서, '한'을 형태소 분석하면 의존명사, 수사, '하(동사)+ㄴ(관형형 어미)' 등의 결과가 나온다. '우리는 누구나 한 가정의 구성원이다.'라는 문장에서 '한 가정'이 '하나의 가정'이라는 의미를 알면 '한'은 수사로 분석될 것이다. 그러나, '누구나 하는'으로 분석을 한다면, 이때의 '한'은 '하(동사)+ㄴ(관형형 어미)'로 분석될 것이다

3. 제안하는 방법

본 논문에서 제안하는 품사부착 코퍼스 구축 방법은 그림 2 와 같다. 그림 2 에서 형태소 분석기는 입력 문서를 보고 분석 실패하면 미등록어가 포함되어 있다고 추정을 한다. 그리고 나서, 미등록어를 포함한다고 추정된 어절들을 사용자에게 제시한다. 사용자는 시스템이 제시한 추정된 미등록어 목록을 보고 미등록어들을 '사전 관리기'라는 도구를 사용해서 사전에 등록한다. 등록이 끝나면, 형태소 분석기는 분석 실패했던 어절들을 다시 형태소 분석한다. 미등록어 처리 과정을 거치면 형태소 분석 결과에 과잉 분석 현상이 줄어들게 되어 품사 태깅의 정확률도 높아지게 된다.

자동 품사 태깅의 오류를 사람이 수동으로 수정하기 전에 본 시스템은 자동으로 오류를 찾아주고 대안을 제시하는 기능을 제공한다. 이 때, 시스템은 오류 규칙과 사람이 이전에 수정했던 내용(수정 로그)을 사용해서 오류를 찾고 대안을 제시한다.

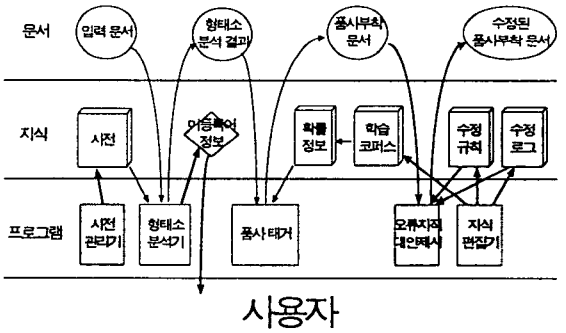

그림 2. 제안하는 품사부착 코퍼스 구축 방법

3.1 태깅워크벤치의 구성

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 3 에 나타나 있다.

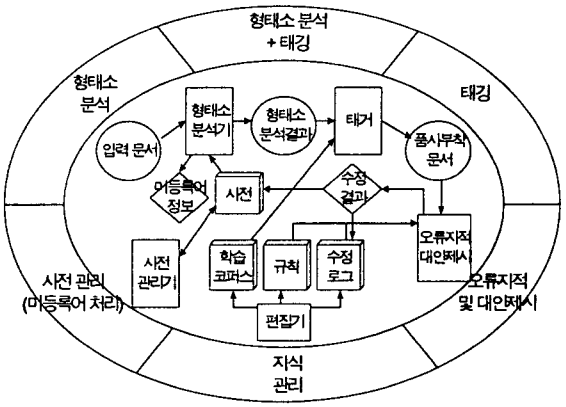

그림 3. 태깅 워크벤치 전체 구조

그림 3 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 적절

히 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 과정을 반복한다.

3.2 미등록어 처리

태깅 오류 중의 대부분은 사전에 없는 단어 즉 미등록어로 인한 것이다. 일반적이로 형태소 분석기에 미등록어가 포함된 어절이 입력되면, 형태소 분석기는 그 미등록어가 어떻게 구분되는지를 추정해서 가능한 모든 태그를 부여한다. 따라서, 분석 결과가 많아지게 되고, 태거는 옳은 태그열이 존재하더라도, 그것을 찾기가 어렵게 된다.

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 추정하고, 이러한 어절들을 사용자에게 제시한다. 그리고, 사용자가 사전 관리기로 미등록어인 형태소들을 사전에 등록한 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 논문에서 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.

3.3 자동 오류 지적 및 대안 제시

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 사용된다.

규칙을 이용하는 방법은, 사용자가 오류인 어절에 대한 규칙과 대안을 기술하여, 그 규칙과 일치하는 어절이 발견되면, 그 규칙에 맞는 대안을 제시하는 것이다. 이 방법에서 사용되는 규칙의 형식은 그림 4 와 같다.

<수정 문맥 형태소> <수정 문맥 품사>)* /
정할 형태소 또는 품사의 위치 /
정 후의 형태소 또는 품사

그림 4. 자동 오류 지적 규칙의 형식

그림 4 에서 <수정 문맥>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 위

치>에서 지정한 위치의 형태소나 품사가 어떻게 수정될 것인가를 나타낸다.

<수정 문맥>에는 Don't Care(*), Closure(+), NOT(!), OR(|) 의 네가지 연산자를 사용할 수 있다.

예를 들어, 동사 '되다' 앞의 주격조사를 보격조사로 바꾸는 규칙은 '* jcs 되 pvg / 2 / jcc'와 같이 기술할 수 있다.

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 결과 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 분석결과를 비교해서 수정 로그의 형식으로 변환해서 즉시, 현재 태깅 결과에 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.

4. 실험 및 평가

본 실험에서 형태소 분석기는 [3]의 형태소 분석기를 사용하였고, 품사 태거는 [4]의 품사 태깅 시스템을 사용하였다. 그리고, 품사 태그 집합으로는 [2]의 태그 집합을 사용하였다.

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 1 에 나타나 있다.

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

문서	어절 수	태깅 오류 수	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 1 실험 문서들의 자동 태깅 결과

오류 수정은 미등록어 입력을 통한 수정, 자동 오류 수정과 수동 오류 수정을 통한 피드백 반영으로 나눌 수 있다.

미등록어 제시를 통한 미등록어 처리를 통해서 전체 오류의 약 7% 정도를 수정할 수 있었다. 그리고, 본 워크벤치를 통해서 문서들을 태깅함으로써 사전 등록 형태소의 수가 점점 증가하게 되므로 새로운 문서를 태깅할 때 미등록어의 수가 점점 줄어들게 될 것이다.

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. 표 2 에서 '미등

록어 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.

미등록어 제시를 통한 미등록어 처리를 통해서 전체 오류의 약 7% 정도를 수정할 수 있었다. 그리고, 본 워크벤치를 통해서 문서들을 태깅함으로써 사전 등록 형태소의 수가 점점 증가하게 되므로 새로운 문서를 태깅할 때 미등록어의 수가 점점 줄어들게 될 것이다.

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 2. 오류 수정 결과

5. 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.

본 논문에 대한 향후 연구 과제로 다음의 두 가지를 들 수 있다.

첫째, 오류에 대한 체계적인 관찰이 필요하다. 이를 통해서 새로운 규칙들을 많이 찾아낸다면 자동 오류 수정의 정확률을 높일 수 있을 것이다.

둘째, 수정 로그를 보다 효율적으로 사용하는

방법에 대한 연구가 필요하다. 본 논문에서는 단순한 패턴 비교를 통해서 수정 로그를 구성하였는데, 수정 로그들을 좀 더 자세하게 표현한 후 수정 로그에서 규칙을 발견할 수도 있을 것이다.

참고문헌

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

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

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

[4] 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.

[5] 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.

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