자율 학습에 의한 실질 형태소와 형식 형태소의 분리

조 세 형†

요 약

본 논문은 태그가 없는 단순 말뭉치만을 가지고 자율학습을 이용하여 정보 검색을 위한 색인어의 추출 등에 이용될 수 있도록 한국어의 실질 형태소와 형식 형태소를 분리해내는 기법에 대하여 기술한다. 본 기법은 사전 등의 언어 관련 지식을 요구하지 않으며 오직 단순 말뭉치만을 필요로 한다. 또한 자율학습을 이용함으로써 사람의 간섭이 필요하지 않아 학습에 필요한 시간과 노력이 거의 들지 않는다. 본 방식은 잘 확립된 통계적 방법론을 이용하기 때문에 일반적인 휴리스틱과는 달리 이론적인 기반이 확고하여 확장 및 발전이 용이하다. 본 결과는 한국어에 우선 적용되었으나 한국어에 종속적인 방법이 아니어서 다른 교착어에도 쉽게 적용될 수 있을 것이다.

A Korean Language Stemmer based on Unsupervised Learning

Se Hyeong Cho†

ABSTRACT

This paper describes a method for stemming of Korean language by using unsupervised learning from raw corpus. This technique does not require a lexicon or any language-specific knowledge. Since we use unsupervised learning, the time and effort required for learning is negligible. Unlike heuristic approaches that are theoretically ungrounded, this method is based on widely accepted statistical methods, and therefore can be easily extended. The method is currently applied only to Korean language, but it can easily be adapted to other agglutinative languages, since it is not language-dependent.

키워드 : 자연어(natural language), 형태소(morphology), 자율학습(unsupervised learning), stemmer

1. 서 론

형태소 분석 및 품사 태깅은 대부분 사전을 필요로 한다. 이러한 방식으로는 새로운 어휘가 지속적으로 생성되는 경우 분석에 어려움이 따른다. 이러한 문제를 해결하기 위해서는 지속적인 사전 재구축 작업이 필요한데 이는 많은 비용과 시간을 소모하는 작업이다. 본 논문에서는 사전 없이 한국어의 실질 형태소와 형식 형태소를 분리해내는 기법에 대하여 기술한다. 본 연구에는 두 가지 목적이 있다. 첫째는 정보 검색기에 사용될 사전 없는 실질 형태소 분리기(lexicon-free stemmer)를 구현하는 것이다. 이 실질 형태소 분리기는 사전이 없이 작동이 가능하여야 하며 특별한 분야에 적용이 가능하기 위해서는 대규모의 말뭉치를 요구하지 않는 것이 바람직하다. 사전을 필요로 하지 않음으로써 국어사전에 대한 저작권 문제를 야기하지 않으며 또한 지속적으로 변하는 어휘에 대응할 수 있게 될 것이다. 둘째는 기계학습 기술에 의한 자연 언어 처리의 가능성을 연구하고자 하는 것이다. 이를 위해서는 형태소 학습, 구문 학습, 의미 학습 등이 모두 필요하나 본 논문에서는 그 첫 단계라고 할 수 있는 형태소 학습에 치중한다.

학습은 그 방식에 따라서 자율 학습(unsupervised learning)과 지도 학습(supervised learning)으로 구분할 수 있다. 본 논문에서는 학습의 지도자나 품사 분석자료가 없이 단순 말뭉치(raw corpus)만을 이용하는 자율 학습을 선택하였다. 그 이유는 형태소 학습의 이론적, 실질적 목적이 미지의 언어, 또는 매우 많은 수의 미지 어휘를 포함한 언어를 컴퓨터로 하여금 학습하게 하는데 있으므로 사람의 간섭이나 미리 준비된 정답이 요구되는 지도 학습을 사용함이 적절치 않기 때문이다.

2절에서는 관련 연구에 대해서 논한다. 3절에서는 한국어 형태소의 특징을 알아보고 여기에서 나타나는 몇 가지 통계적 특성을 살펴보며, 이러한 통계적 특성을 형태소 분리 알고리즘에 사용할 수 있는지 가능성을 알아본다. 4절에서는 이러한 통계적인 특성을 기초로 한 다단계 형태소 학습 방법과 그 이론적 당위성을 소개하고 5절에서는 연구 결과에 대한 분석과 아울러 향후 연구 방향을 논한다.

† 정 회 원 : 명지대학교 교수
논문접수 : 2001년 4월 4일, 심사완료 : 2001년 10월 20일

2. 관련 연구

근자에 와서 형태소 학습에 관한 많은 연구가 이루어져 왔다[10-14]. 그러나 아직까지 대부분이 영어 또는 프랑스어 등의 굴절어에 대한 연구에 국한되어 있다.

굴절어에 대한 자동 형태소 학습 방법은 상당히 오래 전부터 시도되어 왔다[9]. 초기의 연구는 미리 준비된 접미사 사전과 어근에 관련된 규칙을 이용한 것들이다. 최근에 들어서 언어에 독립적인 형태소 학습 방법이 연구되고 있다[10, 11]. Marquez[12] 등은 품사 태깅을 위한 기계학습 기반 방식을 개발하였으나 이것은 사전이 주어진 상태에서 모호성 해소에 관한 지식을 학습하는 것이다. Porter[16]는 사전이 필요 없는 실질 형태소 분리기(stemmer)를 개발하였다. Porter의 알고리즘은 단순히 흔한 접미사들을 찾아서 굴절 또는 파생된 단어에서 떼어내는 것으로 영어에서는 비교적 이러한 접사의 수효가 적기 때문에 가능하며 품질은 그리 좋지 않으나 단순하고 속도가 빨라 많이 사용되고 있다. Gaussier[13]는 활용 사전(inflectional lexicon)을 이용하여 어미 활용을 학습하는 자율 학습기를 개발하였다. 여기서는 단어간의 앞머리가 일정 길이 이상 같은 단어들을 후보로 분류하고 클러스터링을 이용하여 단어들을 관련 군(family)으로 묶는 방법을 제안하였다. 그러나 "일정 길이"에 대한 기준이 매우 임의적이라는 것과 활용 사전이 있어야 하는 단점이 있다.

Goldsmith[11], DeJean[14] 등도 형태소 자율 학습기를 개발하였으나 굴절어에 국한되어 있다. 이들은 통계적인 방법을 이용하여 후보 어근들을 찾아내고 적합한 활용어미를 찾아내는 방식을 사용한다. 여기에서는 적절한 접미사가 부적절하게 적용되거나, 형태소 상의 중의성(ambiguity)이 있는 등의 부작용이 있다. 후보 접미사(candidate affix)를 선택하는 기본 방법에 있어서 Goldsmith[11], Gaussier[13], Schone[10]등은 모두 p-유사도(p-similarity)를 사용하였다. P-유사하다는 것은 두 어휘의 첫 p글자가 같다는 것이다. 즉 어두가 길이 p 이상만큼 공통적인 단어들의 어미를 모아 이들 중 빈도가 많은 K개의 어미를 후보로 선택한다. 여기서 p의 크기는 매우 임의적이며 대상 언어에 따라 달라진다. K의 크기도 역시 임의적이며 Goldsmith[11]는 상위 100개를, Shone[10]은 200개를 선택하였으나 그 선택의 이론적 근거는 없으며 단지 해당 언어에 대한 연구자들의 사전 지식(흔히 쓰이는 접미사의 수효)을 근거로 임의 선택한 것이다. 이러한 과정은 미지의 언어에 대한 자동 분석을 위해서는 결정적인 이론적 약점으로 작용한다.

Schone[10]은 Goldsmith, Gaussier등의 방법에 더하여 행렬의 Singular Value Decomposition을 이용하여 차원을 줄이는 방식으로 LSA(Latent Semantic Analysis) 기법을 이용하여 중의성 문제를 해결하는 학습 방법을 시도하였으며 이 방법은 본 논문에서 제안된 방법과 병합될 수 있다.

한국어의 형태소 분석 및 품사 태깅의 수준은 이미 상용 제품에 사용되는 수준에 이르러 있다. [1]에 의하면 한국어 태깅의 정확도는 89~97%에 달하는 것으로 보고되어 있는데, 실상 황금 표준이라고 할 수 있는 수작업 태깅도 수 퍼센트의 오류를 가지기 때문에 이 정도의 오류는 완벽에 가깝다고도 볼 수 있다. 그러나 이들은 사전에 의존하는 방식이기 때문에 본 논문에서의 목적에는 적합하지 않다. [2]는 명사 파생 접미사의 사전 정보 구축을 위하여 통계적인 방식을 이용하였는데 접미사는 이미 사전에 나온 것을 이용하였다.

[3]은 한국어의 음절 특성을 형태소 분석에 이용한 바 있다. 예를 들어 '은/는'과 같은 끝 음절을 가진 어절은 단일어로 분석될 가능성이 거의 없다는 등의 휴리스틱을 이용하였다. 이러한 휴리스틱은 사람이 가진 구체적인 언어 지식을 이용하는 것으로, 본 논문의 목적상 적합하지 않다. 그 외의 형태소 분석 연구로는 효율적인 사전의 구축[4, 5]이라든가 특정 언어(한국어)에 의존적인 지식을 이용하여 형태소 분석을 향상하는 연구[3, 6], 통계적 방식과 규칙기반 방법의 접목 방식[1] 등에 관한 연구들이 이루어져 왔다. [7]은 음절간의 상호정보(mutual information)를 이용한 자동 띄어쓰기 기법을 연구하였다. 자동 띄어쓰기를 위해서는 어휘 지식과 휴리스틱을 이용하는 방법과 어휘 지식 없이 통계적 정보(mutual information)[8]만을 이용한 방식이 가능한데 여기서는 후자를 이용함으로써 사전 없는 방식을 사용하였다. 그러나 그 연구 목표가 어절과 어절의 분리라는 점에서 한 어절내의 형태소와 형태소의 분리라는 본 논문의 연구 목표와는 차이가 있다.

한국어 형태소에 대한 이러한 많은 연구에도 불구하고 한국어 형태소 학습에 대한 연구는 찾아보기 어려운 실정이다.

3. 한국어 형태소의 특징 및 분석 방법의 비교

본 절에서는 사전 없는 형태소 분리를 가능하게 하는 한국어의 특징을 살펴 보고 이를 분석할 간단한 방법 몇 가지를 비교하였다. 그러나 이러한 방법들은 단독으로 사용되었을 때 어느 것도 형태소 분리기로 사용할 만한 수준이 되지 못하고 있음을 보여주고 있다.

3.1 한국어 형태소의 특징

교착어(agglutinative language)인 한국어는 단어에 조사, 어미 등의 접사가 교착(agglutination) 된다. 이는 영어에서 명사의 수, 인칭 등을 바꾸는 굴절(inflection) 및 동사의 명사화 등을 이루는 파생(derivation) 등과 접사가 붙는다는 점에서 유사하지만 한국어의 형식 형태소는 굴절어에서보

다 문법적으로 더 중요한 역할을 한다. 예를 들어 형식 형태소인 조사는 체언의 뒤에 붙어서 선행하는 체언의 문법적인 위치를 결정한다. 또 다른 형식 형태소인 어미는 공대, 시제, 서법, 종결의 형태 등을 나타낸다. 우리말에서는 띄어쓰기의 대상이 단어가 아닌 어절인데, 많은 경우 한 어절 내에는 실질 형태소와 형식 형태소가 결합되어 있기 때문에 이들을 분리하여야 구문의 분석이나 색인어의 추출 등이 가능하다. 실질 형태소는 열린 집합(open class)으로 지속적으로 어휘가 추가되는 반면 형식 형태소는 특별한 경우를 제외하고는 닫힌 집합이어서 크게 수효가 늘어나지 않으며 이에 따라 실질 형태소는 형식 형태소에 비해 현격하게 많은 수효를 가지고 있다.

3.2 형태소의 결합이 나타내는 통계적 특징

형식 형태소인 조사와 어미가 가지는 공통적인 특징은 어절의 끝에 위치한다는 것이다. 따라서 형식 형태소는 어절의 끝에 자주 출현하게 되며 역으로 어절의 끝에 자주 출현하는 어말[1](suffix)을 찾아서 형식 형태소일 것으로 추정하는 것이 가능하다. 여기서 "자주"라는 말을 통계학적으로 여러 가지로 해석해 볼 수 있다. 첫째는 절대 빈도이다. 예컨대 음절 '다'는 거의 모든 문장에서 종결 어미로 사용되고 있다. 따라서 만일 한 문장의 평균 길이가 10어절이고 90%의 문장이 "다"로 종결된다면 "다"의 어말 출현 빈도는 9%이다. 흔히 사용하는 한글의 음절 종류를 대략 2000자[2] 정도로 볼 때 한 음절의 평균 출현 빈도는 1/2000 = 0.05% 이므로 "다"의 빈도는 평균의 약 180배에 달한다. 따라서 각 음절의 어말 출현 빈도를 보아 빈번한 순으로 형식 형태소일 가능성이 많은 것으로 볼 수 있다.

둘째는 상대 빈도, 즉 전체 출현 빈도에 대한 어말 출현 빈도의 비율이다. 음절에는 형식 형태소의 여부와 상관 없이 빈도가 큰 음절이 있다. 따라서 이러한 음절은 당연히 끝 음절에도 많이 나타난다. 따라서 이러한 단점을 보완하기 위해서 상대적 빈도(즉, 끝 음절 빈도/총 빈도)를 사용할 수 있다. 상대 빈도에서 가장 문제가 되는 것은 낮은 빈도의 어절 중에 상대 빈도가 높은 것들이다. 이들은 비록 최우추정(maximum likelihood estimation)에 의한 확률은 높지만 통계량의 부족으로 인하여 신뢰도 높은 결과를 제공하지 못한다.

셋째는 활용 빈도로서 주어진 어말을 포함하는 어절의 종류 수를 말한다. 만일 "아파트"라는 어절이 100회 출현했더라도 "트"의 활용 빈도는 1회로 간주한다.

마지막으로 주어진 어말이 우연히 어말에 등장했을 가능성이 있는가를 가릴 수 있는 가설 검정의 통계치이다. 문장의 생성은(단순화하면) 음절의 확률적 선택으로 볼 수 있다. 또 특정 음절을 기준으로 보면 성공/실패(즉, 출현/비출현)의 베르누이 실행으로 볼 수 있으며 이 경우 이항 검정을 이용할 수 있다[8]. 임의의 지점에서 어느 음절이 선택될 확률을 p_0 라고 하고 이 음절이 어절 끝에 나타날 확률을 p 라고 하자. 또한 형식 형태소가 아닌 음절들은 무작위로(random) 선택된다고 가정하면 형식 형태소는 어절 끝에 나타날 확률 p 가 p_0 보다 클 것이다. p 를 표본에서의 어절 끝 출현 확률, p_0 를 귀무 가설(null hypothesis)에 의한 음절의 출현 확률, N 을 음절 수(시행 횟수)로 볼 때, T-test의 공식은 식 (1)과 같다.

$$z_0 = \frac{p - p_0}{\sqrt{\frac{p_0(1-p_0)}{N}}} \quad (1)$$

위 식 (1)을 이용하여 예컨대 유의 수준 0.05로 검정할 때, Z_0 값은 1.96이상이 되어야 귀무 가설을 기각할 수 있다. 다시 말해서 Z_0 값이 1.96이상이 된다면 95% 신뢰도를 가지고 당 음절은 형식 형태소라고 판단할 수 있다. <표 1>은 15,000음절의 신문 기사(동아일보 2001년1월25일)에서의 음절 "는"이 나타내는 네 가지 통계치를 보인 것이다.

〈표 1〉 음절 "는"의 4가지 통계치 및 순위

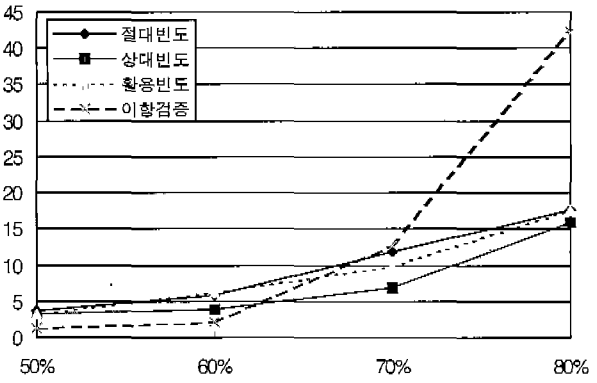
통계 방식	통계 값	순위
절대빈도	1091	2
상대빈도	0.93	42
활용빈도	702	1
이항검증	21.2	1

위의 각 통계치들을 형태소 분리 기준으로 삼았을 때, 각 방식들의 우열을 비교해 보았다. 각 방식에서 실제 형식 형태소로 사용되는 음절을 x%를 포함하도록 값(threshold)을 정했을 때, 이 후보 명단 가운데 잘못 포함된 음절의 백분율(y%)을 나타내었다. 즉 포함률 대비 오류율을 도식하였다. (그림 1)에서 보는 것처럼 이항 검정법은 70% 이상의 형식 형태소를 후보에 포함시키도록 Z_0 값을 정하면 오류율이 급격히 증가함을 볼 수 있다. 이 관점에서 보면 가능한 한 많은 형식 형태소를 찾아내고자 한다면 상대 빈도 방식이 가장 좋은 방식이라고 볼 수 있다. 반면에 낮은 포함률(50~60%)에서 보면 이항 검증법이 가장 오류가 적은 것을 볼 수 있다. 상대 빈도 방식의 경우는 저빈도(4회이하)의 음절은 대상에서 제외하였다. 형식 형태소의 목록은 [15]를 참조하였다.

이러한 통계치를 가지고 형식 형태소 여부를 가려내는 기준으로 삼기에는 부적절함을 알 수 있다. 그래프에서 보듯이 낮은 통계치를 기준하면 형식 형태소가 아닌 것이 형식

1) "어미"가 이미 특정한 용어로 사용되기 때문에 "어미" 대신에 "어말"을 사용하였다. 여기서 어말이란 단순히 어절의 임의 후반부 음소열을 뜻한다(즉 rear substring).

2) KS5601의 경우 2350글자임.



(그림 1) 방식별 형식 형태소의 포함률 대비 오류율.

형태소로 잘못 분류될 것이며 반대로 높은 통계치를 기준으로 한다면 형식 형태소 중에 발견되지 못하는 것들이 많이 생겨나게 된다. 따라서 이러한 단순한 통계만으로 형태소를 분리하기는 어렵다.

4. 다단계 형태소 학습/분리 방식

앞 절에서 설명하였듯이 한 가지의 통계치 만을 가지고 어떤 스트링이 형식 형태소인지를 판단하기는 어렵다. 따라서 본 절에서는 통계적 방식을 기초로 한 다단계 형태소 학습/분리 방식을 제안한다. 이 방식에서는 형식 형태소 후보의 파악을 위해 이항 검정(T-test)을 사용하였으며 동일 어절의 반복적 출현은 1회로 간주하였다. 즉, 활용 빈도와 이항검정을 조합하였다. 앞 절에서 보았듯이 동일 어절의 반복적 출현은 그 어미의 뒷부분이 형식 형태소인가 아닌가를 판별하는데 더 이상의 정보를 주지 못한다. 예를 들어 "아파트"라는 동일한 어절이 100회 출현했다고 하여도 이것만으로는 "파트"나 "트"가 형식 형태소라는 근거가 되지 못한다. 왜냐하면 본래 형식 형태소가 어절 끝에 많이 등장할 것이라는 가정은 "여러 가지 어두와의 결합 때문"이라는 추론에 의한 것이기 때문이다. 또한 실제 형태소는 음소 단위로 이루어지므로 음절 단위가 아닌 음소단위로 분석하였다. 이항 검정을 기본 방법으로 한 다단계 형태소 학습/분리 방법은 다음과 같은 6 단계로 구성된다.

1. 이항 검정에 의한 형식 형태소 후보 생성 단계
2. 잘못 포함된 형식 형태소 후보의 배제 단계
3. 실질 형태소 후보의 생성 단계
4. 실질 형태소와의 결합 빈도에 의한 형식 형태소의 결정 단계
5. 말뭉치의1차 형태소 분리 및 형식 형태소간 바이그램 추출 단계
6. 형식 형태소간 바이그램 확률을 고려한 2차 형태소 분석 단계

각 단계의 대략의 역할은 다음과 같다. 1 단계에서는 앞 절에서 예시한 T-test를 모든 어절의 모든 어말에 적용하여 귀무가설을 반박할 수 있는 어말만을 선정한다. 이 방법에서는 형식 형태소가 아닌 어말이 많이 있으므로 2 단계에서 이러한 어말들을 걸러내는 작업을 2 단계에서 수행하게 된다. 여기서 만들어진 후보 형식 형태소들은 매우 확률이 높은 것들이다. 3 단계에서는 이들과 자주 결합하는 어두를 찾아내어 실질 형태소의 후보로 등록한다. 역으로 이 실질 형태소 목록을 이용하여 실질 형태소와 결합을 빈번히 이루지 않는 형식 형태소 후보를 삭제하는 것이 4 단계이다. 5 단계에서는 1차적으로 말뭉치에 있는 문장들에 대하여 어절 분리를 시도한다. 또한 여기서는 형식 형태소가 공기(co-occurrence) 관계에 있는 것을 이용하기 위하여 바이그램을 추출하며 마지막으로 6 단계에서는 이를 이용하여 분석 대상 문장의 어절들의 형태소 분리를 하게 된다. 학습 단계는 5 단계까지이며 6 단계는 학습의 일부가 아니고 실제 형태소 분리기로서의 역할을 한다. 4.1절 이하는 각 단계에 대한 상세한 설명과 알고리즘을 기술한다.

4.1 이항 검정에 의한 형식 형태소 후보 생성 단계

1 단계에서는 모든 어절의 모든 어말에 대해 이항 검정을 실시하여 후보를 선정한다. 중복된 어말에 대한 중복 분석을 배제하고 효율적인 분석을 하기 위하여 역방향 트라이를 사용한다. 이 단계에서 트라이 구조의 특성상 동일한 어절은 두 번 입력되지 않으므로 중복된 어절은 자동으로 제외된다. 트라이의 모든 내부 노드는 실제 출현한 어말에 해당한다. 조합형으로 나타내되 받침이 없는 음절도 보이지 않는 받침을 가진 것으로 간주하였다. 이항 검증을 위한 Z_0 값의 계산을 위해서는 선험 확률(prior probability)과 표본에서의 출현 확률을 구해야 한다. 이론적인 확률이 존재하지 않으므로 출현 확률은 최우추정으로 구한다. 최우추정 확률은 어떤 스트링이 나타날 가능성이 있는 곳의 총 수효에 대한 실제로 나타난 횟수의 비율로 계산한다. 즉 베르누이 시행이 가능한 위치에 대한 실제 출현 횟수의 비율이다. 예를 들어보자. 2음절로 구성된 어절에서는 1음절짜리 스트링에 대한 2회의 베르누이 시행이 가능하다. 만일 2음절 어절이 총 1000어절이 있었는데 이중 100곳에서 "다"가 출현했다면 $100 \div (2 \times 1000) = 0.05$의 출현 확률을 가진다고 볼 수 있다. 또 다른 예를 들어보자. "ㅂ니다"라는 스트링이 나타날 수 있는 곳은 최소한 3음절 이상을 가진 어절에서만 가능하다. 따라서 2음절 짜리 어절에서 "ㅂ니다"가 나타날 확률을 추정한다는 것은 전혀 의미가 없다. 4음절로 구성된 어절에서는 "ㅂ니다"가 나타날 가능성이 두 곳에 있다. 즉, 1, 2, 3음절과 2, 3, 4음절이다[3]. 만일 전체 말뭉치가 2음절 짜

3) 4음절 어절에서 3음절 후보의 출현은 엄밀한 의미에서 베르누이 시행은 아니다. 만일 1, 2, 3음절 위치에 나타난다면 2, 3, 4음절 위치에는 나타날 수 없다("하하하" 와 같은 음절은 두 번도 가능하다). 그러나 출현 확률이 1보다 매우 작다는 가정에서 이 점은 무시하고 베르누이 시행인 것으로 근사한다.

리 1000음절과 4음절 짜리 1000 어절이라고 한다면 의미 있는 확률로 베르누이 시행을 할 수 있는 것은 2000회이다. 이 중 20 곳에서 실제 출현했다면 1회의 베르누이 실행 때 "ㅂ니다"가 나타날 확률은 $\frac{20}{2000}=0.01$로 추정함이 타당하다. 출현하지 않을 것이 확실한 2음절 어절 내에서는 확률을 따질 필요가 없는 것이다. 어떤 스트링 w의 출현 확률 $p_0(w)$는 식 (2)과 같이 추정한다.

$$p_0(w)=\frac{\#(w, C')}{Pos(|w|, C')} \tag{2}$$

여기서 C'은 원시 말뭉치 C에서 중복 출현 어절을 제외한 가상적 말뭉치이며 $\#(x, C')$는 스트링 x가 말뭉치 C'에 포함된 빈도를 말한다. 또, $Pos(l, c)$는 말뭉치 c와 어절 길이 분포가 같은 임의의 말뭉치에서 길이 l인 스트링이 출현할 가능성이 있는 위치의 수효이며 식 (3)와 같다. 여기서 l_{max}는 가장 긴 어절의 음절수이며 $\#_i(c)$는 길이 i음절인 어절의 c내의 수효이다.

$$Pos(l, c)=\sum_{i=l}^{l_{max}}(i-l+1)\times\#_i(c) \tag{3}$$

어절 끝 확률 $p(w)$는 식 (4)과 같이 계산하며 여기서 \$는 어절의 끝을 나타내는 가상적인 길이 0의 음소이다.

$$p(w)=\frac{\#(w\$, C')}{\sum_{i\geq|w|}\#_i(C')} \tag{4}$$

즉 $p(w)$는 w의 실제 어절 끝 출현횟수에 의한 최우추정치이다. 위의 식을 이용하여 식 (5)의 Zo 값을 계산하여 유의수준 0.1로 추정하여 후보 목록을 만들었다.

$$z_0(w)=\frac{p(w)-p_0(w)}{\sqrt{\frac{p_0(w)(1-p_0(w))}{N}}} \tag{5}$$

비교적 작은 2만5천 어절(중복 제외)의 말뭉치를 분석한 결과 4만여종의 어말 중에서 유의 수준 0.1로 귀무가설을 반박하는 517종의 후보가 선택되었다.

1 단계 알고리즘은 (알고리즘 1)에 기술되어 있다.

```
L←{}
for each suffix w in corput C'
    z0(w) ← (p(w) − p0(w)) / sqrt(p0(w)(1 − p0(w))/N)
            // p(w) and p0(w) as defined by (2), (4)
    if Z0 > T1, L←L∪{w}
endfor
```

(알고리즘 1) 제 1 단계 알고리즘

여기서 T_1은 상수값(threshold)이며 1.3을 택하였는데 이는 유의수준 0.1에 해당한다. N은 베르누이 시행의 총 시행 횟수, 즉 $\sum_{i\geq|w|}\#_i(C')$이다.

4.2 잘못 포함된 형식 형태소 후보의 배제 단계

이항 검정에서 어절 끝 출현 확률이 임의 확률보다 충분히 높은 후보들이 1단계에서 선택되었으나 이들 중에서는 실제로는 후보형태소 자신 때문이 아니라 후보의 일부분(rear substring) 때문에 선택되는 경우가 흔히 있다. 예를 들어 "이다"는 형식 형태소이며 따라서 어절 끝에 출현하는 빈도와 확률이 매우 높다. 이렇게 빈도가 큰 경우 "ㅁ이다", "ㄹ이다", "ㄴ이다" 등은 비록 "이다"의 빈도에는 못 미치지만 타 어말에 비해서 높은 빈도를 보이게 된다. 게다가 이렇게 어말이 길어질수록 이 음소열이 어절의 중간에 나타날 확률은 더욱 작게 된다. 따라서 p_0값은 더욱 작아지고 Z_0의 값은 커지게 된다. 이와는 달리 "에서는"의 경우는 "는"이라는 어미의 빈도 때문에 비례적으로 많아진 것이 아니다. "이다" 앞에 나타나는 "ㅁ"의 확률과 "ㄹ"의 확률 등은 각 자음의 받침 출현 확률에 대략 비례하여 나타난 것인 반면에 "는" 앞에서 나타난 "에서"의 확률은 단순히 "에서"라는 음소열이 출현할 임의의 빈도에 비례한 것이 아니며 이보다 훨씬 크다. 따라서 이러한 경우 "에서"의 일반적인 확률과 "는"의 앞에서 나타날 확률을 비교함으로써 이것이 우연인지를 밝힐 수가 있다.

즉, w와 δ를 중복 없는 말뭉치 C'내의 음소열이라 할 때, $Z_0(w)>T_1$이고 $Z_0(\delta w)>T_1$인 경우

$$p'(\delta, w)=\frac{\#(\delta w\$, C')}{\#(w\$, C')} \tag{6}$$

$$p_0'(\delta)=\#(\delta, C')/\mathrm{Pos}(\delta, c') \tag{7}$$

식 (6)와 식 (7)을 이용하여 다시 이항 검정을 하면 "ㅁ이다"의 경우처럼 실제로는 의미 없는 경우를 제외시킬 수 있을 것이다. 실제 실험에서 "ㅁ이다"의 1단계 이항검정에서의 Z_0(ㅁ이다) 값은 1.4였으나 2 단계 분석에서는 Z_0(ㅁ,이다) = 0.68로 감소하였다. 반면에 "에서는"의 경우 1단계에서 Z_0(에서는) = 3.46을 보였으며 2 단계에서는 Z_0(에서, 는) = 20.0으로 여전히(오히려 더 확실한) 형식 형태소로 분류되었다. 결과적으로 1 단계에서 사용되었던 2만5천 어절 말뭉치를 분석하여 1 단계에서 선택되었던 517종의 후보 가운데 308종이 탈락하고 219종만이 남았다. 이 단계의 알고리즘은 (알고리즘 2)에 나타내었으며 여기에서 N_w는 w의 출현횟수이다.

```
for each suffix v = δw ∈ L        // L is from stage 1 algorithm
    Z0(δ, w) ← (p'(δ, w) − p0'(δ)) / sqrt(p'(δ)(1 − p0'(δ))/Nw)
                                  // p'(δ, w) and p0'(δ) as
                                  // defined by (6), (7)
```

```
if Z0(δ, w) < T1, L ← L − {v}
endfor
```

(알고리즘 2) 제 2단계 알고리즘

4.3 실질 형태소 후보의 생성 단계

어느 어절의 형식 형태소를 안다면 실질 형태소를 알 수 있고 거꾸로 실질 형태소를 안다면 형식 형태소를 알 수 있다. 완벽하지는 않지만 단계 1, 2를 통하여 많은 형식 형태소들을 찾아내었으므로 이제는 이를 이용하여 다시 실질 형태소를 구분해 낼 수 있다. 그러나 주어진 어절에서 형식 형태소의 집합만을 가지고 (만일 완벽한 집합이라고 하여도) 실질 형태소를 분리해 내기는 어렵다. 그 이유는 첫째, 어느 어절의 어말 부분이 우연히 어떤 실질 형태소와 같을 수가 있기 때문이다. 예를 들어 "ㄴ"은 많은 경우에 형식 형태소로 쓰이지만 그렇지 않은 경우도 많다. 예를 들어 "대문"의 경우가 그러하다. 둘째 이유로는 형식 형태소들끼리는 상호 포함 관계(rear substring)에 있을 수가 있으며 이 경우 두 가지 이상으로 형태소 분리가 가능한 애매성의 문제(morphological ambiguity)가 존재하기 때문이며 마지막으로는 주어진 형식 형태소 후보에 여전히 오류가 있을 수 있기 때문이다.

이 문제를 해결하기 위해 이 단계에서는 실질 형태소로서의 가능성이 있는 단어들을 추출한다. 여기에서 "실질 형태소는 단독으로서 어절을 구성하거나 형식 형태소와 결합하여 어절을 구성한다"라는 사실을 이용할 수 있다. 실질 형태소의 후보 w는 $|\{\delta \mid w\delta \in C', \delta \in L\}|$ 즉 형식 형태소 후보 δ와 결합된 어절이 말뭉치 C'내에서 출현한 빈도에 의해 결정된다. 이론적으로는 이 또한 T-test에 의하여 판정하는 것이 바람직한 듯이 보인다. 그러나 일반적으로 실질 형태소는 그 빈도가 형식 형태소와 비교할 때 매우 낮다는 특징 때문에 통계적 검정의 의미가 적어진다. 즉, 최우추정을 하기에 통계량이 매우 부족하다. 이러한 이유로 본 논문에서는 2단계에서 선택한 형식 형태소 중 일정한 수 (threshold) T_2개 이상 결합 사용된 어두를 실질 형태소 후보로 선정하였으며 $T_2=3$으로 실험하였다.

```
M ← {}, ∀x, count(x) = 0
for all eojul w = w1 ··· wk
  for i = 2 to k-1
    if wi ··· wk ∈ L then
    count(w1 ··· wi-1)++
  endfor
endfor
for all strings w
if count(w) > T2 then M ← M ∪ {w}
endfor
```

(알고리즘 3) 실질 형태소 후보의 생성 알고리즘

4.4 실질 형태소와의 결합 빈도에 의한 형식 형태소의 결정

앞의 단계에서 형식 형태소와의 결합에 의해 실질 형태소를 찾아내었듯이 여기서는 역으로 실질 형태소와 몇 가지 형태로 결합하느냐에 의해 형식 형태소 후보를 배제한다. 예를 들어 "그토록" 등에 쓰이는 "록"은 높은 Z_0 값을 가지고 있어서 자칫 독립된 형식 형태소로 인식되기 쉽다. 그러나 "록"의 앞에 결합된 어두들은 실질 형태소로 구분되는 것이 없다. 본래 "형식 형태소는 여러 종류의 실질 형태소와 결합하여 사용되기 때문에 어절 말에 자주 출현한다"라는 가정에서 출발하였으므로 비록 어절 말에 자주 출현하더라도 실질 형태소와의 결합이 빈번하지 않으면 형식 형태소로 인정할 수 없다. 학습 단계는 4단계로서 끝나게 된다. 알고리즘은 (알고리즘 4)에 나타나 있으며 T_3의 상수 값은 4로 실험하였다.

```
for each w ∈ L
  for each v ∈ M
    if v · w ∈ C then count(w)++ // C is the corpus
  endfor
endfor
for each w ∈ L
  if count(w) < T3, then L ← L - {w}
endfor
```

(알고리즘 4) 실질 형태소와의 결합 빈도에 따른 형식 형태소의 재결정

4.5 1차 형태소 분리 및 바이그램의 획득

우선 어말이 형식형태소로 사용될 확률과 어두가 실질형태소로 사용될 확률을 구해보자. 어느 스트링 w가 나타날 확률을 p라고 하고 이것이 어미에서 나타날 확률을 p_s라고 하자. a는 어절말에 나타난 w의 수효, b는 어절말이 아닌 다른 위치에 나타난 수효, a'은 어절 말에 나타난 w 중 형식 형태소로 사용된 수효라고 하자. 또, A는 어절의 수, B는 어절말 이외에 w가 나타날 수 있는 위치의 총합이라고 하자. 최우추정에 의하면 w의 확률은 $p=\frac{a+b}{A+B}$이며 $p_s=\frac{a}{A}$이다. 형식형태소로 쓰일 수 있는 w가 어절 말에 있을 때 이것이 형식 형태소로 사용되었을 확률(MLE)은 식 (8)과 같다.

$$p_{GM}=\frac{a'}{a}=\frac{a-\frac{A}{B}b}{a}=1-\frac{Ab}{Ba} \tag{8}$$

예를 들어 3음절짜리 어절로만 구성된 1,000,000 어절이 있다고 하자. "는"의 횟수가 75,000회이며 이중에 어절 끝에 나타난 횟수가 70,000회라고 하자. 이 경우 어절 앞 및 중간 부분에 나타난 횟수는 5,000회이며 $p_{GM}=\frac{a'}{a}=1-\frac{1000000\times5000}{2000000\times7000}$ $=1-0.0357=0.9643$이다. 물론 이 확률은 형식 형태소로

사용되는 일이 있는 어말에만 적용되는 것이며 그렇지 않은 어말에 대한 확률은 평탄화에 입각한 최소 확률만을 부여하였다.

어두 w가 실질 형태소일 확률(P_{CM})은 이와는 달리 앞에서 실질 형태소와의 결합 빈도에 의해서 그 확률을 3단계로만 양자화(quantize)하였다. 즉, M에 포함된 실질 형태소와 포함은 되지 않았으나 형식 형태소($\exists_w, w \in L$)와 결합한 예가 1회라도 있는 경우, 그리고 전혀 결합한 예가 없는 경우이다. 이것을 양자화한 이유는 형식 형태소는 그 종류가 많지 않아 출현 빈도가 높은 반면 실질 형태소는 그 종류의 다양함으로 인하여 출현 빈도가 낮고 특히 저출현 빈도의 형태소가 많기 때문이다. 이것은 출현횟수가 빈도 순위에 반비례한다는 Zipf의 법칙[17]을 보면 쉽게 짐작할 수 있다. 따라서 실질 형태소의 경우 충분한 통계량을 제공하지 못하게 되므로 이들의 확률을 추정하여 사용하는 것이 큰 의미가 없고 오히려 큰 오차를 야기하게 된다. 세 경우에 대한 확률은 실험적으로 0.9, 0.1, 그리고 1/|C|로 주었다.

$Cm(w)$ 을 "w is used as a content morpheme" 이라고 하고 $Gm(w)$를 "w is used as a grammatical morpheme" 이라고 하면 실질-형식 형태소의 분리는 식 (9)을 만족하는 m을 구하는 것으로 나타낼 수 있다.

$$m = \arg\max_k P(Cm(w_1 \cdots w_k) \& Gm(w_{k+1} \cdots w_n) \mid w = w_1 \cdots w_n) \quad (9)$$

Bayes의 법칙에 의해 식 (9)를 다시 쓰면 :

$$m = \arg\max_k \frac{P(w = w_1 \cdots w_k \mid Cm(w_1 \cdots w_k) \& Gm(w_{k+1} \cdots w_n)) P(Cm(w_1 \cdots w_k) \& Gm(w_{k+1} \cdots w_n))}{P(w = w_1 \cdots w_n)} \quad (9')$$

식 (9')과 같이 되며 여기서 분자의 전반부는 1이며 분모는 모든 k에 대해 같으므로 식 (9")과 같이 되고 실질 형태소의 선택과 형식 형태소 선택을 독립 사상으로 가정하면 최적해 m은 식 (10)과 같이 된다.

$$m = \arg\max_k P(Cm(w_1 \cdots w_k) \& Gm(w_{k+1} \cdots w_n)) \quad (9")$$

$$m = \arg\max_k P(Cm(w_1 \cdots w_k) \& Gm(w_{k+1} \cdots w_n)) = \arg\max_k P_{Cm}(w_1 \cdots w_k) P_{Gm}(w_{k+1} \cdots w_n) \quad (10)$$

즉, 1차 형태소 분석은 어두가 실질 형태소일 확률과 어말이 형식 형태소일 확률의 곱이 최대인 것으로 선택한다.

4.6 형식 형태소 바이그램 확률을 이용한 2차 분석

독립된 어절의 경우는 애매성(ambiguity)으로 인하여 정확한 분리에 한계가 있다. 예를 들어 "순은"의 경우 "올해 순은 값은 하락세를 면치 못하고 있다"에서와 "그러자 순은 말했다"에서의 경우 전자는 형식 형태소가 없는 경우이며 후자는 있는 경우이다. 이러한 문제를 해결하고 향상된 형태소 분리을 위해서는 형식 형태소간의 바이그램 확률을 이용하면 좀더 정확한 확률을 얻을 것으로 예측할 수 있다. 예를 들어 "순은 값은 하락세이다"라는 문장에서 P(순•은|순은) = 0.55, P(순은•|순은) = 0.45이라고 가정해 보자. 여기서 기호 "•"은 분리 위치를 나타낸다. 이 경우 전후를 고려하지 않은 형태소 분석은 "순•은"으로 분석하게 될 것이다. 그러나 주격 조사인 "은"이 연달아 출현할 확률은 상대적으로 매우 적다[4]. 이러한 바이그램 확률을 고려한다면 많은 경우에 애매성을 해소하거나 근소한 차이로 잘못 판단될 수 있는 결론을 고칠 수 있게 될 것이다. 만일 형식 형태소 "은"이 연달아 나올 확률이 한 번 나올 확률의 10분의 1이라면 우리는 쉽게 "순은"의 "은"은 형식 형태소가 아니라고 결론 내릴 수 있다.

그러나 이것은 상호 의존적인 문제로서 형태소 분리가 되어야 바이그램을 얻을 수 있고 바이그램이 있어야 형태소 분리를 할 수 있다. 만일 어느 정도 정확성을 가진 초벌 형태소 분리기가 있고 오류가 있다고 하더라도 오류가 특정 경우로 집중되지 않는다고 가정하면 (이 가정은 충분히 큰 말뭉치에 대해서 성립한다) 바이그램은 형태소 분리기의 개선에 유용하게 된다. 한 가지 문제는 형식 형태소가 없이 실질 형태소 단독으로 어절을 이루는 경우이다. 이러한 경우에는 예를 들어 "나는 여행을 좋아한다."라는 문장에서는 "는"과 "을"의 바이그램을 이용할 수 있지만 (실지로 말뭉치의 분석 결과 바이그램 확률이 가장 높은 문장은 "~은 ~을 ~다"의 형태이다) "나는 겨울 여행을 좋아한다"에서는 "나는"과 "여행을" 사이에 실질 형태소만으로 구성된 어절이 끼어있기 때문에 <는, 을>의 바이그램의 확률을 이용할 수 없게 된다. 따라서 이 단계에서는 어미가 없는 실질 형태소 만으로된 어절(ε-어절이라고 하자)을 제외한 나머지 어절만으로 바이그램 확률[5]을 계산한다. 즉, 원거리 바이그램(long-distance bigram)을 이용한다. $P_b(<w, v>)$는 문장에서 주어진 어절 왼쪽 어절 중 가장 가까운 형식 형태소가 w일 때 당 어절에서 v가 형식 형태소로서 나타날 확률이라고 하자. 즉 형식 형태소 바이그램의 확률인 것이다. 문장 $E = <e_1, e_2, \cdots, e_n>$(각 e_i는 어절)에 있어서 형태소 분리 해(solution)를 $S = <s_1, s_2, s_n>$(각 s_i는 어절 i의 분리 위치)이라고 할 때 *last(E, S, k)*는 k번째 어절까지 중 가장 우측에 있는 non-ε 실질 형태소로 정의한다. 예를 들어 "올해 순은 값은 하락세이다." 라는 문장에서 S = <6, 3, 3, 9> 라는 해는 (음절을 초, 중, 종 3음소로 표현) "올해• 순•은 값•은 하락세•이다"와 같이 분리하는 해이다. *last*(*E*, *S*, 1)

4) 이러한 예로는 "스님은 산은 산이요 물은 물이라 했다"를 들 수 있다.

5) 단, 확률은 곱할수록 작아지는 특징이 있어 ε어미의 수효에 따라 확률 값이 차이나는 것을 방지하기 위해 평균값을 곱해준다. 즉, $P_b(<x, \varepsilon>) = \text{average}(P_b(<x, y>))$.

은 ε이며 $last(E,S,2)$는 "은" 이고 $last(E,S,3)$도 "은"이다. 바이그램 확률을 고려한 해는 식 (11)과 같이 정의된다. 여기서 e_j^i는 어절 j의 i 번째 음소이며 ^는 문장의 시작을 나타내는 가상적인 형식 형태소이다.

$$S_{\max} = \arg\max\{P(Cm(e_1^1\cdots e_1^{s_1}))P(Gm(e_1^{s_1+1}\cdots e_1^{|e_1|}))P_b(\langle \hat{}, e_1^{s_1+1}\cdots e_1^{|e_1|}\rangle)\times P(Cm(e_2^1\cdots e_2^{s_2}))P(Gm(e_2^{s_2+1}\cdots e_2^{|e_2|}))P_b(\langle last(E,S,1), e_2^{s_2+1}\cdots e_2^{|e_2|}\rangle)\times \cdots \quad (11)$$

$$P(Cm(e_n^1\cdots e_n^{s_n}))P(Gm(e_n^{s_n+1}\cdots e_n^{|e_n|}))P_b(\langle last(E,S,n-1), e_n^{s_n+1}\cdots e_2^{|e_n|}\rangle)\}$$
$$= \arg\max \prod_{i=1}^{n} P_{CM}(e_i^1\cdots e_i^{s_i})P_{GM}(e_i^{s_i+1}\cdots e_i^{|e_i|})P_b(\langle last(E,S,n-1), e_i^{s_i+1}\cdots e_i^{|e_i|}\rangle)$$

<표 2>는 문장 "순은 값은 하락세이다"에 대한 확률 값을 두 해 S_1 = <3, 3, 9>와 S_2 = <6, 3, 9>에 대하여 비교하고 있다. (여기서 확률 값은 이해를 돕기 위한 인위적인 값이며 실제 값과는 차이가 있다)

〈표 2〉 수식에 의한 두 가지 해의 확률 값의 비교

어절	"순은"	"값은"	"하락세이다"
S_1	3 (즉, Gm("은")	3	9
P_{CM}	0.9	0.9	0.9
P_{GM}	0.9	0.9	0.9
$last$ (E, S_1, i)	^	"은"	"은"
S_1	P_b(^,은>) = 0.1	P_b(<은, 은>)=0.01	P_b(<은, 이다>) = 0.2
P_b의 확률	0.9×0.9×0.1×0.9×0.9×0.01×0.9×0.9×0.2 = 0.0001062		
S_2	6 (즉, Gm(" "))	3	9
P_{CM}	0.9	0.9	0.9
P_{GM}	0.3	0.9	0.9
$last$ (E, S_2, i)	ε	"은"	"은"
P_b	P_b	P_b(<^, 은>) = 0.1	P_b(<은, 이다>) = 0.2
S_2의 확률	0.9×0.3×0.1×0.9×0.9×0.1×0.9×0.9×0.2 = 0.0003542		

이 수식의 평가를 위하여 모든 조합을 계산 할 경우 시간적인 복잡도(time complexity)는 $\prod_{i=1}^{n}|e_i|$로서 $|e_i| \le m$, 즉 한 어절의 최대 길이(음소의 수)가 m이고 문장 내 어절의 개수가 n일 때 $O(m^n)$이며 이는 실시간에 계산해 내기 어려운 수치이므로 상당히 큰 n값에 대해서는 보다 효율적인 계산 방법이 요구된다. 본 연구에서 분석에 사용된 5만 어절 짜리 말뭉치는 문장의 평균 길이가 15.0 어절이며 최대 56 어절이었다. 실지로 23 어절로 구성된 문장에 대해 위 식에 의해 최대치를 계산해 내는데 Pentium III 700에서 약 1 시간 가량이 소요되었다. 상기 수식의 효율적인 계산을 위해서는 잘 알려진 다이나믹 프로그래밍 기법을 사용하면 $O(nm^2)$의 복잡도로서 계산이 가능하다. 실제 구현한 결과 한 문장에 대한 계산 시간은 무시할 정도로 작았다. 다이나믹 프로그래밍 알고리즘은 교과서적으로 변환이 가능하므로 여기에서는 서술을 생략한다.

5. 결과의 분석 및 향후 연구 방향

비교적 작은 말뭉치인 2만5천 어절의 말뭉치에서 통계를 얻어 동일한 말뭉치에 있는 문장을 임의 추출하여 분석하였다. 즉 학습 말뭉치와 평가 말뭉치는 중복 어절의 차이를 제외하고는 같은 말뭉치이다.[6] 이 말뭉치는 2001년1월 제4주의 동아일보 기사에서 추출한 것이다. 본 연구에서는 아직 복합 어미에 대한 분석을 하지 않고 있으며 형태소에 대한 품사 붙이기(tagging)도 하지 않으므로 성공과 실패의 표준을 올바른 실질 형태소의 분리로 삼았다. 이러한 기준으로 1 단계 분석에 의해 선택한 분리는 74%의 성공률을 보였다. 2 단계 분석으로 오류 후보 상당수가 제외된 후 다시 분석한 결과 81.5%의 성공률을 보였으며 다시 4단계 알고리즘에 의해 실질 형태소를 찾아낸 후 재분석 결과 85%의 성공률을 보였다. 성공률의 측정은 난수 발생기에 의하여 선택된 1000개 어절에 대한 것이며 오차의 범위는 1%, 신뢰도는 95%이다[18].

또, 동일한 문장에 대해 학습용 말뭉치의 크기를 2배로 하여 학습한 후 시도한 결과 성공률은 85%에서 87%로 향상되었다. 그러나 바이그램의 확률을 고려한 방식은 예상과 달리 가시적인 향상을 보이지 않았으며 오히려 일부 경우에 바이그램 확률을 고려하지 않았을 때 분석에 성공했던 어절에 대해 오답을 내는 결과를 가져왔다. 이러한 원인 중 가장 큰 것은 말뭉치의 크기에서 기인된 것으로 분석된다. 2만5천 어절에서 약 200개의 형식 형태소에 대해 바이그램을 구하였으므로 200×200 = 40,000 종류가 있는 반면 형식 형태소가 있는 어절은 대략 1만 5천 정도에 지나지 않으므로 1쌍 당 평균 출현 횟수가 0.5회에도 미치지 못하는 미미한 수치이다. 따라서 연어 정보를 활용하기 위해서는 획기적으로 큰 말뭉치를 이용하여야 할 것으로 생각되며 특정 집단의 언어와 같이 대규모 말뭉치가 가용하지 않은 경우에는 바이그램의 적용이 어려울 것으로 생각된다.

위 성공율은 단순히 실질 형태소와 형식 형태소(뭉치)를 분리해 내는 초보적인 형태소 분리로서, 품사 태깅의 정확도와 직접 비교하기는 어렵다. 그러나 이 정도의 성공률로서도 문서의 인덱싱을 위한 정보로 사용하는 것은 가능할 것으로 보이며 향후 여러 가지 시도를 통하여 향상할 수 있는 여지를 가지고 있다. 그 중 몇 가지를 나열하면 형식 형태소와 실질 형태소간의 바이그램 정보 이용, 복합 형식 형태소의 분리, 복합 실질 형태소의 분리, 실질 형태소 및 형식 형태소의 클러스터링에 의한 품사 학습, 말뭉치의 오류 대응 등이 있다.

본 논문에서는 형식 형태소간의 바이그램 정보만을 이용

6) 본 연구의 목적상 대규모 말뭉치의 사용을 지양하고 있기 때문에 별도의 학습용 대규모 말뭉치를 사용하지 않고 분석할 대상 자체를 학습 말뭉치로 사용하였다.

하였다. 그러나 때로는 실질 형태소와 형식 형태소간에 큰 바이그램 확률을 보임으로써 형식 형태소의 파악에 영향을 미칠 수 있다. 예를 들어 "~과는 달리" 라든가 "~하기 때문" 과 같은 바이그램은 <형식 형태소, 실질 형태소> 간의 바이그램으로서 뒤좇아오는 실질 형태소에 대한 정보가 앞 어절의 형식 형태소의 결정에 도움을 주게 될 것으로 예측된다.

현재로서는 형식 형태소 중 가장 확률이 높은 후보를 선택하게 되어 있다. 그러나 한국어에서는 실제로 형식 형태소 여러 개가 연속하여 출현하는 현상을 볼 수 있다. 따라서 본 알고리즘에 의한 분리 결과를 보면 형식 형태소를 분리하였음에도 불구하고 어두에 여전히 형식 형태소가 남아있는 현상을 볼 수 있다. 따라서 분리된 어두에 대한 반복적인 형태소 분석에 의해 추가적인 형태소를 분리함으로써 정확도를 높일 수 있을 것으로 생각된다.

실질 형태소의 분리도 영향을 미칠 것이다. 예를 들어 "예산결산위원회"같은 복합 명사가 있을 경우 "예산" "결산" "위원회"라는 단어에 대한 지식이 있더라도 현재로서는 하나의 실질 형태소라는 명확한 증거가 없어 이러한 복합어에 빈도가 낮은 어미가 결합되면 분리에 실패하여 하나의 단어(즉, ε어미)로 취급될 것이다. 반면에 이 단어를 복합어로 인식할 수 있다면 분리가 보다 잘 이루어 질 것이다.

품사 정보가 있다면 또한 형태소 분리에 도움이 될 수 있다. 예를 들어 독립된 단어를 이루는 부사의 경우 비록 실질 형태소로 구분되지만 "매운 맛"에서 부사 "매우"와 어미 "ㄴ"으로 분리해서는 아니된다. 그 이유는 "매우"는 부사이며 "ㄴ"은 "앞서 간 발자국"에서의 "ㄴ"처럼 동사를 활용하게 하여 관형어를 이루기 때문이다. 그러나 현재는 단순히 실질 형태소에 대한 하위 범주가 존재하지 않기 때문에 이러한 오류를 막지 못한다. 이 문제는 실질 형태소가 말뭉치에서 어떠한 범주의 형식 형태소와 결합하는가에 대한 통계 정보로부터 클러스터링을 통하여 일종의 품사 범주를 생성함으로써 해결될 것으로 예상된다.

또한, 현재 말뭉치는 오류가 없는 것을 가정하고 있지만 대규모 말뭉치는 물론 소규모의 말뭉치조차도 오류가 없기를 기대하기는 어렵다. 특히 띄어쓰기의 오류는 매우 흔하며 따라서 이러한 오류를 찾아내어 통계에서 제외하거나 또는 오류에도 중요한 영향을 받지 않는 알고리즘에 대한 연구가 필요할 것으로 보인다.

마지막으로, 본 연구의 결과를 유사한 특징을 가진 외국어에 적용함으로써 언어에의 비의존성을 시험해보는 것도 향후의 중요한 연구 방향의 하나이다. 예를 들어 일본어의 경우 띄어쓰기가 없는 것을 제외하면 우리말과 매우 유사한 형태소 결합 구조를 가지고 있다. 이러한 경우 잘 알려진 분리 알고리즘과의 (예를 들어 [19]) 결합을 통한 일본어 형태소 분석/품사 태깅에의 적용을 생각할 수 있다.

참 고 문 헌

[1] 신상현, 이근배, 이종혁, "통계와 규칙에 기반한 2단계 한국어 품사 태깅 시스템", 정보과학회논문지(B) 제24권 제2호, pp.160-169, 1997.

[2] 남윤진, 옥철영, "말뭉치 분석에 기반한 명사파생접미사의 사전정보 구축", 정보과학회논문지(B), 제23권 제4호, pp.389-401, 1996.

[3] 강승식, "음절특성을 이용한 한국어 불규칙 용언의 형태소 분석", 정보과학회논문지(B) 제22권 제10호, pp.1480-1487, 1995.

[4] 최재형, 이상조, "양방향 최장 일치법에 의한 한국어 형태소 분석기에서의 사전 검색 횟수 감소 방안", 한국정보과학회논문지 Vol.20, No.10, pp.1497-1507, 1993.

[5] 김철수, 배우정, 이용식, 高江純一, "이중배열 트라이 구조를 이용한 한국어 전자 사전의 구축", 정보과학회논문지(B) 제23권 제1호, pp.85-94, 1996.

[6] 임희석, 윤보현, 임해창, "배제 정보를 이용한 효율적인 한국어 형태소 분석기", 한국정보과학회논문지, 제22권 제6호, pp.957-964, 1995.

[7] 심광섭, "음절간 상호정보를 이용한 한국어 자동 띄어쓰기", 정보과학회논문지 제23권 제9호, pp.991-1000, 1996.

[8] C. Manning and H. Schltze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.

[9] Lovins, J. B., "Development of stemming algorithms," in Machine Translation and Computational Linguistics, 11, 1968.

[10] Patrick Schone and Daniel Jurafsky, "Knowledge-free Induction of Morphology using Latent Semantic Analysis," in proceedings of the ACL99 workshop : Unsupervised learning in Natural Language Processing, University of Maryland.

[11] J. Goldsmith, "Unsupervised learning of the morphology of a natural language," University of Chicago, http : //humanities.uchicago.edu/faculty/goldsmith.

[12] LLuis Marquez, Lluis Padro, and Horacio Rodriguez, "A Machine Learning Approach to POS tagging," Machine Learning, Vol.39, pp.59-91, 2000.

[13] E. Gaussier, "Unsupervised learning of derivational morphology from inflectional lexicons," in proceedings of the ACL99 workshop : Unsupervised learning in Natural Language Processing, University of Maryland.

[14] Dejean, H., "Morphemes as necessary concepts for structures : Discovery from untagged corpora," University of Caen-Basse Normandie. http : //www.info.unicaen.fr/DeJean/travail/article/pg11.htm. 1998.

[15] 김홍규, 강범모, "한국어 형태소 및 어휘 사용 빈도의 분석", 고려대학교 민족문화연구원, 2000.

[16] M. F. Porter, "An algorithm for suffix stripping," Program, 14(3), pp.130-137, 1980.

[17] Zipf, G. K. *Human Behavior and the Principle of Least Effort*, Cambridge, MA : Addison-Wesley, 1949.

[18] W. Mendenhall and R.J.Beaver, *Introduction to Probability and Statistics*, Boston, MA, PWD-Kent publishing co. 1995.

[19] R. Ando and L. Lee, "Unsupervised Statistical Segmentation of Japanese Kanji Strings," Technical Report TR99-1756, Computer Science Department, Cornell University, 1999.

조 세 형

e-mail : shcho@mju.ac.kr

1981년 서울대학교 섬유공학과(학사)

1983년 서울대학교 대학원 계산통계학(석사)

1992년 Pennsylvania 주립대학 전산과(박사)

1984년~2000년 한국전자통신연구원 책임연구원

2000~현재 명지대학교 조교수

관심분야 : 자연언어 처리, 정보 검색 및 추출, 에이전트