근접 문맥정보와 대규모 웹 데이터를 이용한 단어 의미 중의성 해소

강신재* 강인수**
(Sin-Jae Kang*, In-Su Kang**)

요 약 본 논문은 구글(Google), 워드넷(WordNet)과 같이 공개된 웹 자원과 리소스를 이용한 비교사학습(Unsupervised learning) 방법을 제안하여 단어 의미의 중의성 문제를 해결하고자 한다. 구글 검색 API를 이용하여 단어의 확장된 근접 문맥정보를 추출하고, 워드넷의 계층체계와 synset을 이용하여 단어 의미 구분정보를 자동 추출한 후, 추출된 정보 간 유사도 계산을 통해 중의성을 갖는 단어의 의미를 결정한다.

핵심주제어 : 비교사학습, 오픈 API, 워드넷, 단어의미중의성해소

Key Words : unsupervised learning, open API, WordNet, word sense disambiguation

Ⅰ. 서 론

단어 의미 중의성 해소(WSD, word sense disambiguation)는 자연어처리 분야에서 오랫동안 연구해 온 주제이면서도 그 성능의 향상이 두드러지지 않는 연구 분야이다.

단어 의미 중의성 해소란 단어가 가지고 있는 여러 의미 가운데 그 단어가 사용된 문장에서의 의미를 결정하는 문제인데, 단어의 의미는 주어진 문맥에서 정확히 결정될 수 있다. 따라서 대부분의 WSD 시스템은 접근 방법에 상관없이 WSD할 대상 단어(target word)의 문맥정보(contextual features)가 있어야 하고, 이 정보와 비교할 대상 단어의 의미 구분 정보(sense differentiation information)가 있어야 한다. 이러한 정보를 추출하기 위해 사용할 수 있는 정보를 정리해 보면 품사정보(parts-of-speech), 공기정보(collocation), 의미빈도, 선택 제약(selectional restriction)과 같은 정보가 필요하다.

WSD를 위한 접근법은 전자사전이나 시소러스를 사용하는 방법과 단어의 의미가 태깅된 말뭉치가 있는 경우 기계학습 방법을 활용한 방법, 그리고 이러한 리소스가 가용하지 않을 때 대규모의 말뭉치에서 통계정보를 추출하여 사용하는 방법 등으로 구분할 수 있다. 소규모로 정해진 단어를 대상으로 중의성을 해소할 때에는 의미 태깅된 말뭉치 등 학습 가능한 리소스를 어느 정도 이용할 수 있으나, 모든 단어를 대상으로 WSD를 수행할 때에는 자료 부족 현상이 발생하여 성능의 향상을 꾀하기 어려운 실정이다.

따라서 본 연구에서는 자료 부족 현상을 해결하기 위하여 오픈 리소스로 공개되어 있는 워드넷과 구글(Google) 검색 API를 이용하여 자동으로 단어 의미의 중의성을 해소하고 그 결과를 워드넷의 synset으로 표현하고자 한다.

Ⅱ. 오픈 리소스

* 대구대학교 정보통신대학 컴퓨터IT공학부 교수
** 경성대학교 멀티미디어대학 컴퓨터정보학부 교수

2-1. 워드넷(WordNet)

워드넷[1]은 널리 알려진 대규모의 영어 어휘 의미 목록 데이터베이스이다. 워드넷은 영어 단어를 'synset'이라는 유의어 집단으로 분류하여 간략하고 일반적인 정의를 제공하고, 이러한 어휘목록 사이의 다양한 의미 관계를 기록한다. 현재 영어를 대상으로 구현된 대부분의 자연어처리 응용 시스템에서는 단어 의미 구분 등의 처리를 하기 위해 워드넷을 활용하고 있으며, 이를 편리하게 하기 위해 워드넷 검색, 유사도 계산 API 등 많은 라이브러리와 소프트웨어 도구들이 개발되어 제공되고 있다.

2-2. 구글 검색 API

Google AJAX 검색 API[2]는 웹 페이지 및 다른 웹 응용프로그램에 Google 검색을 포함할 수 있게 해주는 자바스크립트 라이브러리이다. Flash 및 다른 비 자바스크립트 환경에서 API는 원시 RESTful 인터페이스를 표시한다. 이 인터페이스는 대부분의 언어 및 런타임에서 쉽게 처리할 수 있는 JSON 인코딩 검색결과를 반환한다.

이 API를 이용하면 한번에 4개씩 최대 28개까지 검색결과를 추출할 수 있다.

III. 연구 내용

워드넷은 이를 활용하기 위한 다양한 API(검색, 유사도계산 등)가 공개되어 있고, 구글은 PageRank 알고리즘을 개발/구현하여 현존 검색엔진 가운데 최상의 검색결과를 제공해주며, 사용자 프로그램에서 접근하여 결과를 사용할 수 있도록 API를 제공한다. 따라서 WSD를 위한 정보를 자동으로 추출하여 비교사학습 방법으로 WSD하기 위해서는 워드넷, 구글과 같은 공개 리소스/API를 활용하여 단어의미 중의성을 해소하는 방법을 고안하였다.

3-1. 근접 문맥정보 확장

입력문장에서 WSD의 대상이 되는 단어의 전후 세 단어를 근접 문맥정보로 정의하고 추출하였다. 기존 연구 가운데 Klapaftis[3]도 구글과 워드넷을 WSD에 사용하였지만, 이 연구에서는 WSD 대상 단어가 포함되어 있는 문장 전체에서 문맥정보를 일괄적으로 추출하였기 때문에, 섬세하게 정보를 추출하지 못한 면이 있다. 4장에서 본 연구의 결과와 비교하였다.

하지만 추출한 직접 문맥정보(direct contextual information)는 양이 매우 적으므로, 추가의 문맥정보를 획득하기 위해 직접 문맥정보를 질의어로 구글 사이트에서 검색하고, 검색된 결과의 요약부분에서 동일한 방법으로 간접 문맥정보(indirect contextual information)를 추출하였다(그림 1). 간접 문맥정보는 검색된 결과에서 HTML 태그와 불용어(stopword)를 제거하고 단어의 원형을 복원한 후, 워드넷에 등록되어 있는 단어만 선택하여 최종 단어 목록을 구성하게 된다. 문맥정보는 WSD 대상 단어별로 같이 공기(collocation)한 단어와 빈도수의 쌍으로 이루어진 목록이다.

이 과정의 구현을 위해서는 검색결과를 JSON 포맷으로 리턴하는 Google AJAX Search API를 사용하였고, 불용어 제거(stopping)와 스테밍(stemming)을 위해서는 Apache Lucene[4]에 포함되어 있는 모듈을 이용하였다.

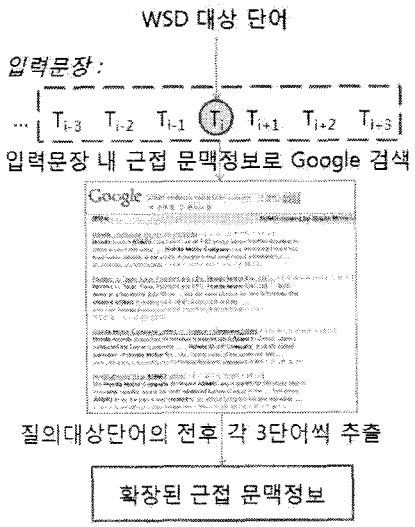

그림 1. 확장된 문맥정보 추출

3-2. 단어 의미 구분정보 추출

본 연구에서는 단어의 의미 표현을 위해서 워드넷의 synset ID를 이용하는데, 각 단어별 의미의 구분을 위해서는 의미에 해당하는 synset 뿐만 아니라 상위어, 하위어, 전체어, 부분어에 해당하는 synset들까지 모두 모아서 synset 목록을 만들어

사용한다(그림 2). 이 모듈의 구현을 위해서는 WordNet 3.0 데이터베이스[5]와 워드넷 검색 API 인 JWI 2.1.3 (MIT Java Wordnet Interface)[6]을 사용하였다.

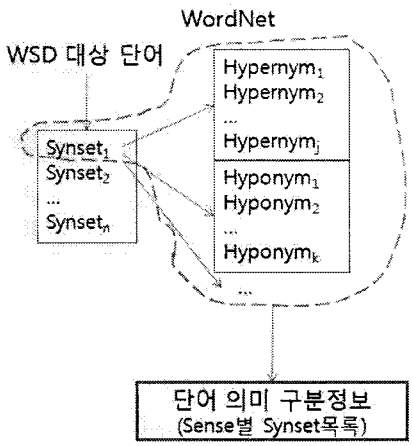

그림 2. 단어 의미 구분정보 추출

3-3. 유사도 계산

단어의 의미를 결정하기 위해서는 입력문장으로부터 유도된 문맥정보(단어목록)와 단어의미별 synset 목록(단어목록)을 비교하여 유사도 값이 가장 높은 단어의 의미(synset)를 선택하는 부분이 필요하다. 3-1절과 3-2절에서 구축한 목록들 간 유사도를 계산하기 위해서는 기존의 여러 척도를 사용할 수 있는데, 본 연구에서는 단순 단어 매칭에 의한 중복도 계산 실험을 하였고, 또한 각 목록에 있는 모든 단어쌍 간 Pirro & Seco[7], Resnik[8]이 제안한 유사도 계산을 한 후, 평균값의 비교를 통해 의미를 결정하였다.

워드넷에서 유사도를 계산하는 척도들은 대부분 정보 이론의 정보량(IC, Information Content)에 기반한 방법인데, c가 워드넷에 있는 개념이고, p(c)가 주어진 말뭉치에서 개념 c가 나타날 확률이라고 할 때, 정보량은 다음과 같이 정의할 수 있다.

$$IC(c) = -\log p(c)$$

Resnik[8]은 자주 등장하지 않는 단어가 자주 등장하는 단어보다 더 많은 정보를 가지고 있다는 전제하에, 각 개념들의 IC값을 안다면 주어진 두 개념 사이의 유사도를 계산할 수 있다고 주장하였다. 두 개념이 공통으로 가지는 정보의 양(MSCA, Most Specific Common Abstraction)에 따라 유사도가 결정되는 것인데, 다음과 같이 유사도 식을 정의하였다. S(c1, c2)는 개념 c1과 c2를 포함하는 개념들의 집합이다.

$$sim_{res}(c_1, c_2) = \max_{c \in S(c_1, c_2)} IC(c)$$

Pirro[7]는 말뭉치에 의존하는 IC 기반 유사도 계산 방법의 문제를 피하기 위해 특징 기반의 유사도 이론을 응용하여 다음과 같은 유사도 수식을 제안하였다.

$$sim_{tvr}(c_1, c_2) = 3IC(msca(c_1, c_2)) - IC(c_1) - IC(c_2)$$

$$sim_{PS}(c_1, c_2) = \begin{cases} sim_{tvr} & \text{if } c_1 \neq c_2 \\ 1 & \text{if } c_1 = c_2 \end{cases}$$

Ⅳ. 실 험

제안한 WSD 방법의 평가 및 기존 연구[3]와의 비교를 위하여 SemCor 3.0 말뭉치 가운데 처음 10개의 파일을 동일하게 선택하였다. SemCor[9]는 Brown 말뭉치의 일부를 워드넷의 synset을 이용하여 수작업으로 태깅한 말뭉치이다. 선택된 파일에는 총 5,463개의 명사를 포함하고 있다.

WSD 시스템의 성능을 평가하는 Senseval-1, 2, 3 워크샵이 열렸었는데, Senseval-3의 결과를 정리해 보면, 단어의 의미를 최다빈도의 의미(most-frequent sense)로 결정하는 방법을 베이스라인(Baseline)이라 할 때, 의미 태깅된 학습말뭉치 등을 사용한 교사학습(supervised learning)의 경우는 최고 성능이 베이스라인을 겨우 넘는 정도이고, 대부분의 비교사학습(unsupervised learning) 방법은 베이스라인에도 미치지 못하는 것을 알 수 있다.

WSD 대상 단어의 의미 결정을 위해 대상 단어의 문맥정보와 의미 구분 정보 간 유사도 계산 결과가 최단빈도의 의미가 아닌 경우, 유사도 최대값을 갖는 의미와 최단빈도 의미 간 비율을 계산하고, 임계값을 넘는 경우에는 최대값을 갖는 의미로 결정하고, 넘지 않는 경우에는 최다빈도의 의미를 대상 단어의 의미로 결정하였다. 실험결과 임계치가 0.4일 때 가장 좋은 성능을 보였다. 본 연구에서 한 실험과 기존 연구결과와의 비교는 아래의 표에 제시되었다.

표 1. 실험결과
(임계치 0.4, 문맥정보 추출은 전후 3단어씩)

실 험		Precision(%)
기존 방법	최다빈도 의미 선택	81.6
	Klapaftis et al.	65.9
제안하는 방법	단어 매칭	71.9
	Pirro & Seco 유사도식	75.8
	Resnik 유사도식	81.7

본 연구에서 제안한 방법은 비교사학습의 범주에 들어가지만 Resnik 유사도식을 사용한 경우, 베이스라인에 가까운 성능을 보이고 있으며, 워드넷과 구글을 사용한 기존 연구 Klapaftis[3]과 비교해 볼 때에도 상당한 차이를 보이고 있음을 확인할 수 있다. [3]은 WSD 대상 단어별 문맥정보를 구분하여 추출하지 않고 입력문장 전체를 사용하고 있어서, 문맥정보의 구분을 섬세하게 하지 못하고, 유사도 계산을 단순 단어 매칭에 의한 점수계산을 하기 때문에 본 연구보다 좋지 못한 결과를 보였다.

V. 결 론

본 논문은 단어 의미 중의성 해소를 공개된 API와 리소스를 이용하여 비교사학습 방법으로 해결하는 방법을 제시하였다.

단어 의미 중의성 해소 처리 시 학습데이터의 부족으로 인한 문제를 해결하기 위하여 오픈 소스(Wordnet, Apache Lucene)와 오픈 API(Google)를 최대한 활용하여 자동으로 단어 의미의 중의성을 해소하고 그 결과를 워드넷의 synset으로 표현하였다.

이는 WSD 기법을 적용한 온톨로지 개체(ontology instances) 일반화 도구의 개발이 가능케 됨으로써, 온톨로지 구축의 생산성을 높이고 온톨로지 학습의 핵심 기술을 확보하는 데에 기여할 수 있다.

본 논문에서 제안한 내용은 영어를 대상으로 하고 있으므로, 한국어에도 적용할 수 있는 모델로 개선할 계획이다.

참 고 문 헌

[1] C. Fellbaum, WordNet: An Electronic Lexical Database (Language, Speech, Communication), MIT Press, May 1998.

[2] Google AJAX 검색 API, http://code.google.com/intl/ko/apis/ajaxsearch/

[3] I. P. Klapaftis, and S. Manandhar, "Google & WordNet based word sense disambiguation," In Proceedings of ICML-2005 Workshop on Learning and Extending Ontologies by using Machine Learning Methods, 2005.

[4] Lucene Java 2.3.2, http://lucene.apache.org/java/docs/index.html

[5] WordNet 3.0 데이터베이스, http://wordnet.princeton.edu/obtain

[6] MIT Java Wordnet Interface, http://www.mit.edu/~markaf/prj/jwi/

[7] G. Pirrò, N. Seco, "Design, Implementation and Evaluation of a New Similarity Metric Combining Feature and Intrinsic Information Content". ODBASE 2008, LNCS, Springer Verlag, 2008.

[8] P. Resnik, "Information Content to Evaluate Semantic Similarity in a Taxonomy," In Proc. of IJCAI 1995, pp. 448-453, 1995.

[9] SemCor 3.0 말뭉치, http://www.cs.unt.edu/~rada/downloads.html#semcor