KIPS Tr. Software and Data Eng.
Vol.5, No.11 pp.563~568 pISSN: 2287-5905

http://dx.doi.org/10.3745/KTSDE.2016.5.11.563

Word Sense Classification Using Support Vector Machines

Park Jun Hyeok† · Lee Songwook††

ABSTRACT

The word sense disambiguation problem is to find the correct sense of an ambiguous word having multiple senses in a dictionary in a sentence. We regard this problem as a multi-class classification problem and classify the ambiguous word by using Support Vector Machines. Context words of the ambiguous word, which are extracted from Sejong sense tagged corpus, are represented to two kinds of vector space. One vector space is composed of context words vectors having binary weights. The other vector space has vectors where the context words are mapped by word embedding model. After experiments, we acquired accuracy of 87.0% with context word vectors and 86.0% with word embedding model.

Keywords : Word Sense Disambiguation, Muliti-Class Classification, Word Embedding, Support Vector Machine

지지벡터기계를 이용한 단어 의미 분류

박 준 혁† · 이 성 욱††

요 약

단어 의미 분별 문제는 문장에서 어떤 단어가 사전에 가지고 있는 여러 가지 의미 중 정확한 의미를 파악하는 문제이다. 우리는 이 문제를 다중 클래스 분류 문제로 간주하고 지지벡터기계를 이용하여 분류한다. 세종 의미 부착 말뭉치에서 추출한 의미 중의성 단어의 문맥 단어를 두 가지 벡터 공간에 표현한다. 첫 번째는 문맥 단어들로 이뤄진 벡터 공간이고 이진 가중치를 사용한다. 두 번째는 문맥 단어의 윈도우 크기에 따라 문맥 단어를 단어 임베딩 모델로 사상한 벡터 공간이다. 실험결과, 문맥 단어 벡터를 사용하였을 때 약 87.0%, 단어 임베딩을 사용하였을 때 약 86.0%의 정확도를 얻었다.

키워드 : 단어 의미 분별, 다중 클래스 분류, 단어 임베딩, 지지벡터기계

1. 서 론

단어 의미 중의성이란 하나의 단어가 두 가지 이상의 의미(sense)를 갖는 현상이다. 예를 들어 '배'라는 단어는 쓰임에 따라 과일인 배(pear)를 의미하기도 하고, 신체의 일부인 배(abdomen)를 의미하기도 하고, 운송수단인 배(boat)를 의미하기도 한다. 이렇게 여러 의미를 갖는 단어의 정확한 의미를 분별하는 것은 정보 검색이나 기계 번역 등 자연 언어 처리의 여러 응용 시스템의 성능에 큰 영향을 끼친다.

단어 의미 중의성을 해소하는 것을 단어 의미 분별(Word Sense Disambiguation(WSD))이라고 하며 지식과 말뭉치 등을 이용하여 다양한 방법에 대한 연구가 이뤄져 왔다. 우리는 WSD 문제를 의미 분류 문제로 간주한다. 분류를 위한 자질로 단어 주위의 문맥 정보를 사용하며 문맥을 추출하기 위해 세종 의미 부착 말뭉치(sense tagged corpus)를 활용한다. 의미 중의성 단어의 문맥 단어들을 두 가지 벡터 공간에 표현한다. 첫 번째는 문맥 단어들로 이뤄진 벡터 공간이고 이진 가중치를 사용한다. 두 번째는 문맥 단어의 윈도우 크기에 따라 문맥 단어를 단어 임베딩 모델로 사상한 벡터 공간이다. 단어 임베딩 모델은 Word2Vec[1]을 이용한다. 두 가지 종류의 벡터로 지지벡터기계(Support Vector Machines(SVM))를 각각 학습하고 단어의 의미를 분류한 후 그 결과를 비교한다.

2장에서 관련 연구를 살펴보고, 3장에서 제안하는 의미 분류 방법을 단어 벡터 공간과 단어 임베딩 벡터 공간으로 나누어 설명한다. 4장에서 실험 결과를 보이며 5장에서 결론을 맺는다.

2. 관련 연구

WSD 방법은 활용 자원에 따라 크게 지식 기반(knowledge-based) 방법과 말뭉치 기반(corpus-based) 방법, 그리고 이 두 가지를 혼용한 방법 등으로 분류할 수 있다.

† 비 회 원 : 한국교통대학교 컴퓨터정보공학과 학사과정
†† 정 회 원 : 한국교통대학교 컴퓨터정보공학과 교수
Manuscript Received : October 4, 2016
Accepted : October 12, 2016
* Corresponding Author : Lee Songwook(leesw@ut.ac.kr)

2.1 지식 기반 방법

지식기반 방법은 사전(Machine Readable Dictionary)이나 WordNet, 한국어 어휘의미망(Korean Lexico-semantic Network(KorLex)) 등의 시소러스를 이용한 방법이다. 사전을 이용한 방법은 사전에 정의된 특정 단어의 여러 의미들의 뜻을 풀이한 문장들을 추출한 후 주어진 문장과 가장 유사한 문장들이 많이 존재하는 의미로 특정 단어의 의미로 결정하는 방법이다[2]. 이 방법은 구현이 간단하지만 사전에 기술된 문장의 길이와 수에 따라 성능차이가 발생한다. [3]에서는 연세 한국어 사전과 표준 국어 대사전을 이용하여 의미 중의성을 가진 단어들에 대한 문맥 정보를 추출하고 사전에서 얻은 문맥 정보와 주어진 문장 사이에 공통된 연어가 존재하면 이를 자동으로 태깅하는 연어 공간 기반 방법을 통해 학습 문맥을 구축하고 나이브 베이지언 분류기를 이용하여 의미를 분류하였다.

사전을 이용한 방법은 사전의 정의가 부실한 경우 자료 부족 문제가 발생하는데, 이를 보완하기 위해 [4]와 [5]는 시소러스를 이용하였다. [4]는 문장에 대하여 연어 관계에 있는 정보를 추출한 후 WordNet에서 의미 중의성을 지닌 단어에 대한 정보를 추출하여 의미 그래프를 구축하였다. 차수 중심성(Degree Centrality)을 이용하여 의미를 결정하였다. [5]는 WordNet과 위키피디어 사전을 연결하여 만든 어휘 의미망 사전 BabelNet을 사용하여 모든 중의성 단어에 대하여 의미 그래프를 구축하였다. Word2Vec를 이용하여 문장이나 문맥에서 나온 단어들 중 의미적으로 유사한 단어끼리 정합하여 의미 그래프를 구축하여 의미를 분별하였다.

2.2 말뭉치 기반(Corpus-based) 방법

[6]은 의미 부착 말뭉치로부터 효율적인 문맥을 추출하기 위해 휴리스틱을 이용한 가변 길이의 윈도우를 사용하였다. 추출된 문맥은 의미 단어별 사전으로 사용되고 주어진 테스트 문맥에서 의미 단어의 빈도와 문맥 단어의 빈도를 이용하여 의미를 선택하였다. [7]은 세종 의미 분석 말뭉치를 학습할 때 중의성을 가진 단어의 각 의미와 윈도우 크기 안에 공기한 단어의 빈도와 거리를 벡터 가중치로 사용하여 주어진 문장의 벡터와 학습된 의미들의 벡터와의 코사인 유사도를 계산하여 의미를 분별하였다. [8]은 [7]에서 생성된 벡터의 차원이 커서 수행속도가 느린 단점을 보완하기 위해 단어 임베딩을 하여 벡터의 차원을 축소한 후 의미를 결정하였는데 성능은 조금 떨어졌으나 실행 속도를 향상시켰다. 우리는 문맥 단어 벡터 공간의 형성에 [7]의 방법을 사용하였으나 [7]에서 사용한 가중치와 달리 이진가중치를 사용한다.

[9]에서는 세종 의미 부착 말뭉치와 세종 전자사전의 용언 하위범주화 정보와 KorLex를 이용하여 단어 의미 중의성 해소를 위한 규칙을 생성한 후 원시 말뭉치를 확장하는 준지도 학습 방법을 제안하였다.

3. SVM을 이용한 단어 의미 분류

먼저 세종 의미 부착 말뭉치를 이용하여 각 중의성 단어의 문맥 단어들을 추출한다. 이 문맥 단어들이 SVM 학습을 위한 자질이 된다. 문맥 단어 벡터 공간과 이를 단어 임베딩으로 사상한 단어 임베딩 벡터 공간을 살펴보자.

3.1 문맥 단어 벡터 공간

Fig. 1.은 SVM을 이용한 단어 의미 분류 과정을 나타낸다.

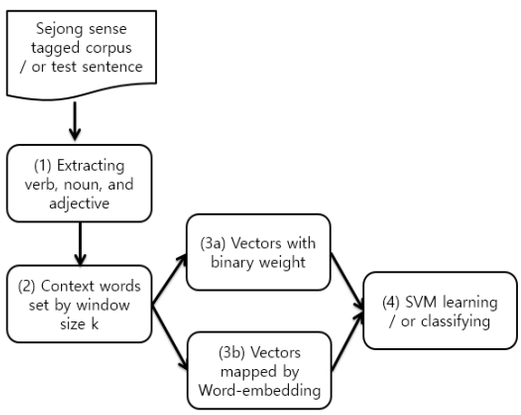

Fig. 1. Word Sense Classification Using SVM

Fig. 1의 각 단계는 다음과 같다.

(1) 먼저 세종 의미 부착 말뭉치에서 명사, 동사, 형용사만 추출한다.

(2) 말뭉치에서 i번째에 위치한 의미 중의성 단어 w_s가 의미 s를 가질 때 문맥 단어 집합 C_{w_s}을 Equation (1)과 같이 생성한다. C_{w_s}는 의미 부착 말뭉치에서 의미 태그 s를 가진 단어 w가 관찰될 때마다 w와 윈도우 크기 k이내에 있는 모든 단어 c를 문맥 단어 집합으로 구성한다(실험에서 k=5). Equation (2)와 같이 수집된 모든 문맥 단어 집합의 합집합 U가 문맥 단어 벡터 공간을 구성하게 되며, 결과적으로 |U|차원의 문맥 단어 벡터 공간이 된다.

$$C_{w_s} = \{c_{i-k}, \ldots, c_{i-1}, c_{i+1}, \ldots, c_{i+k}\} \quad (1)$$

$$U_w = \bigsqcup_{s=1}^{m} \{C_{w_s}\} = \{cw_1, cw_2, \ldots, cw_n\} \quad (2)$$

(3a) 문맥 단어 벡터 v는 이진가중치 t값을 가지는데 수 Equation (3)에 따라 결정한다. 즉 문맥 단어 집합 C_{w_s}에 존재하는 단어로 벡터의 가중치를 결정하는데, Equation (2)의 i번째 차원의 단어 cw_i가 문맥집합 C_{w_s}에 존재하면 해당 차원의 가중치를 1, 존재하지 않으면 0으로 결정한다.

$$v_{w_s} = < t_1, t_2, \ldots, t_n > \qquad t_i = \begin{cases} 1, & cw_i \in C_{w_s} \\ 0, & otherwise \end{cases} \quad (3)$$

(3b) 이 단계는 단어 임베딩을 이용한 벡터 생성 과정인데 3.2절에서 설명한다.

(4) (3)단계에서 생성된 벡터들이 SVM을 학습하는데, 의미 중의성을 가진 단어마다 독립적으로 SVM 학습이 이뤄진다. 학습된 SVM을 이용하면 의미 중의성 단어를 포함하는 새로운 문장에 대해 의미를 분별할 수 있다. 우리는 이진 가중치 벡터와 단어 임베딩 벡터의 의미 분류 학습에 선형 SVM을 사용하며 scikit-learn[10] 패키지를 이용하였다.

각 단계의 결과는 Fig. 2의 예문과 같다.

input: "사람들은 주머니에 돈이 생기면 기름진 음식으로 배를 채우고 그만큼 쌀을 멀리하고 만다."
(1) "사람/NNG 주머니/NNG 돈__01/NNG 생기/VV 기름지/VA 음식/NNG 배__01/NNG 채우__03/VV 쌀/NNG 멀리하/VV"
(2) context words set $C_{배_{01/NNG}}$= {주머니/NNG, 돈/NNG, 생기/VV, 기름지/VA, 음식/NNG, 채우/VV, 쌀/NNG, 멀리하/VV}
(3) context words vector v=
< foreach i
cw_i:0,
except 주머니/NNG:1, 돈/NNG:1, 생기/VV:1, 기름지/VA:1, 음식/NNG:1, 채우/VV:1, 쌀/NNG:1, 멀리하/VV:1 >

Fig. 2. An Example of Building the Context Word Vector

3.2 단어 임베딩 벡터 공간

단어 임베딩을 위해 우리는 Word2Vec[1]를 사용하였다. Word2Vec은 신경망을 이용하여 입력 말뭉치를 구성하고 있는 각각의 단어를 서로 다른 벡터로 표현해 준다. 학습 방법에 따라서 연속 단어 주머니(Continuous Bag Of Word (CBOW)) 모형과 스킵그램(Skip-gram) 모형으로 나뉜다. CBOW는 문맥을 학습하여 특정 단어를 예측하는 모형이고, 스킵그램은 특정 단어를 학습하여 주변에 올 수 있는 문맥을 예측하는 모형이다.

우리는 세종 형태소 부착 말뭉치, 세종 의미 부착 말뭉치, 네이트 뉴스 말뭉치를 이용하여 Word2Vec을 학습하였다. 말뭉치에서 명사, 동사, 형용사만 추출하고, 윈도우 크기 8인 CBOW 모델을 사용하여 문맥을 학습하였다. 예측 단어에 해당하는 고유벡터의 차원은 50차원으로 설정하였다.

Equation (1)의 문맥 단어 집합 C_{w_s}의 각 원소 c_k를 Word2Vec을 이용하면 고유벡터 $w2v(c_k)$를 얻는다. 이 때 고유벡터의 차원을 d라 하면 단어 임베딩 벡터 공간의 전체 차원은 $|C_{w_s}|\times d$이며 $2kd$와 같다. Equation (4)는 의미 중의성 단어 w_s을 위한 단어 임베딩 벡터 v_s^{w2v}를 나타내는데, 문맥 단어 집합 C_{w_s}를 고유벡터 $w2v(c_k)$를 이용하여 단어 임베딩 벡터 공간으로 매핑한 것이다.

$$v_{w_s}^{w2v} = < c_{-k}^{w2v},\ c_{-(k-1)}^{w2v},\ \cdots,\ c_{k-1}^{w2v},\ c_k^{w2v} >, \quad (4)$$
$$c_k^{w2v} = w2v(c_k)$$

Fig. 2의 예문의 경우, 윈도우 크기 k가 만약 2라면, 벡터 $v_{배_{01/NNG}}^{w2v}$ = <w2v(기름지/VA), w2v(음식/NNG, w2v(채우/VV), w2v(쌀/NNG)>를 얻게 되며, 200차원(2*2*50)의 벡터가 된다.

4. 실험 및 결과

우리는 시스템 평가를 위해 일반적으로 자주 쓰이면서 의미 중의성이 높은 단어인 '배'와 의미 중의성이 가장 낮은 '고개'를 선택하였으며, 그 외 평균적인 의미 중의성을 가진 '전기', '사기'를 선정하고 이들 네 단어에 대해 각각 의미 분류 실험을 수행하였다[11]. 세종 의미 부착 말뭉치에서 이들 의미 중의성 단어를 포함하는 문장의 10%를 무작위로 선택하여 평가집합으로 사용하였으며 나머지 90%의 문장과 의미 중의성 단어를 포함하지 않는 말뭉치 전체를 학습에 사용하였다.

Table 1은 학습집합과 평가집합의 구성을 나타낸다. 평가 문장은 805개이며 학습문장은 약 74만개이다.

Table 1. Training Set and Test Set

set	# of sentences	# of words	# of ambiguous words
Training	746,068	9,038,022	8,310
Test	805	12,105	859

Table 2는 중의성 단어의 빈도를 나타낸다. '배'와 '고개'가 전체 집합의 주를 이루고 있으며 '전기'와 '사기'는 두 단어에 비해 상대적으로 적은 수의 빈도를 보인다.

Table 2. Frequency of Ambiguous Words

ambiguous word	# of training	# of test
배	3,984	395
전기	863	80
고개	3,012	335
사기	451	49
합계	8,310	859

Table 3은 의미 중의성 단어의 의미별 빈도수를 나타낸다. 각 단어의 의미의 개수가 분류할 클래스 개수가 되며 분류 문제의 복잡도를 나타낸다고 할 수 있다. 대체로 학습 집합의 의미별 분포를 평가 집합도 따르는 것을 알 수 있다. 저빈도 의미의 경우 학습 집합에는 나타나지만 평가 집합에 나타나지 않는 경우가 많다. '배'의 경우 총 17개의 의미를 가지는데 평가 집합에는 그중 9개의 의미만 포함되었고, '전기'의 경우 8개 중 5개만 포함되었다. '사기'는 7개 중 4개만

평가집합에 포함되었다. '고개'는 의미가 2개 밖에 없고 상대적으로 저빈도 의미가 없기 때문에 2개 모두 평가집합에 포함되었다.

Table 3. Frequency of Senses

ambiguous word (# of senses)	sense	학습 빈도	평가빈도
배 (17)	09/NNG	1162	111
	01/NNG	1103	121
	02/NNG	1060	102
	01/VV	303	28
	03/NNG	162	16
	88/NNG	108	8
	02/VV	34	5
	06/NNG	19	1
	08/NNG	11	3
	05/NNG	5	0
	04/VV	5	0
	88/VV	5	0
	10/NNG	3	0
	00/NNG	1	0
	12/NNG	1	0
	03/VV	1	0
	99/VV	1	0
전기 (8)	15/NNG	635	58
	09/NNG	108	11
	12/NNG	54	7
	27/NNG	43	3
	11/NNG	17	0
	19/NNG	3	1
	07/NNG	2	0
	08/NNG	1	0
고개 (2)	01/NNG	2742	302
	02/NNG	270	33
사기 (7)	25/NNG	294	36
	01/NNG	106	8
	11/NNG	37	1
	88/NNG	10	4
	03/NNG	2	0
	04/NNG	1	0
	14/NNG	1	0

Table 4는 문맥 단어 벡터와 단어 임베딩 벡터를 SVM으로 분류한 실험 결과를 정확도로 나타내었다. 비교 시스템으로는 공기 빈도와 거리 가중치를 사용한 (Park & Lee 2012)[8]를 구현하여 제안 방법과 비교하였다. 3.1절에서 설명한 문맥 단어 벡터 모형은 이진가중치를 이용하였고 윈도우 크기는 5를 사용하였다. 3.2절에서 설명한 단어 임베딩 벡터 모형의 실험에서는 윈도우 크기(k)를 2로 사용하였다.

Table 4에서 먼저 의미별 성능을 살펴 보자. 두 가지 벡터 공간을 SVM으로 분류하는 제안 방법이 (Park & Lee 2012)보다 대부분의 의미 분류에 더 좋은 결과를 보였다. 낮은 성능을 보인 의미들은 대부분 자료부족 문제에 기인한 것으로 보인다. 예를 들어, '배'의 의미번호 '02/VV', '06/NNG', '08/NNG' 등의 학습 빈도가 Table 3에서와 같이 상대적으로 낮은 빈도를 갖기 때문에 다른 의미들에 비해 성능이 좋지 않다.

Table 4. The Accuracy of Systems

ambiguous word (# of senses)	sense	Park & Lee (2012)	context words vector (k=5)	word embedding vector (k=2)	rate(%)
배 (9)	09/NNG	85.6	**91**	87.4	28.1
	01/NNG	**83.5**	76.9	76.9	30.6
	02/NNG	80.4	**85.3**	72.5	25.8
	01/VV	14.3	50	**75**	7.1
	03/NNG	43.6	**62.5**	**62.5**	4.1
	88/NNG	75	**87.5**	**87.5**	2
	02/VV	0	20	**60**	1.3
	06/NNG	0	0	0	0.3
	08/NNG	0	0	**66.7**	0.8
	Overall	74.7	**79.2**	77.7	100
전기 (5)	15/NNG	98.3	**100**	96.6	72.5
	09/NNG	**72.7**	**72.7**	54.5	13.6
	12/NNG	**28.6**	14.3	**28.6**	8.8
	27/NNG	33.3	33.3	**100**	3.8
	19/NNG	0	0	0	1.3
	Overall	**85.0**	**85.0**	83.8	100
고개 (2)	01/NNG	**99.7**	99.0	98.3	90.1
	02/NNG	30.3	72.7	**78.8**	9.9
	Overall	92.8	**96.4**	**96.4**	100
사기 (4)	25/NNG	91.7	**97.2**	88.9	73.5
	01/NNG	37.5	50	62.5	16.3
	11/NNG	**100**	**100**	**100**	2
	88/NNG	75	75	**100**	8.2
	Overall	81.6	**87.8**	85.7	100
Overall	-	83.1	**87**	86	100

Table 4에서와 같이, 각 의미 중의성 단어별 의미 분류 성능에서도 제안 방법이 비교 시스템보다 '전기'를 제외한 3단어에서 비교적 좋은 성능을 보였다. 의미 개수가 가장 적고 학습데이터가 2번째로 많은 '고개'가 96.4%로 가장 좋은 성능을 보였고, 학습데이터는 가장 많았으나 의미 개수가 17개로 가장 많은 '배'가 79.2%로 가장 낮은 성능을 보였다. 따라서 분류의 복잡도와 학습 데이터의 양이 의미 분류 성능에 영향을 끼치는 요인이라 할 수 있다.

Fig. 3은 단어 임베딩 벡터를 이용한 실험에서 윈도우 크기 k에 따른 성능을 보인다.

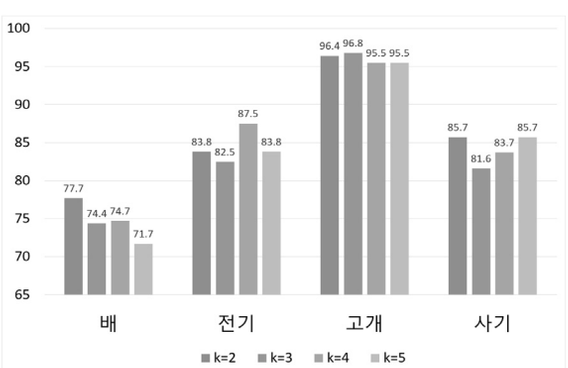

Fig. 3. The accuracy of the system using the word embedding vector with window size k

‘배’, ‘사기’의 경우 윈도우 크기가 작을 때 성능이 가장 좋았으나 ‘전기’, ‘고개’의 경우에는 윈도우 크기가 성능에 큰 영향을 끼치지 않았다. 이는 대체로 좋은 자질은 중의성 단어 가까이에 나타나지만 경우에 따라 멀리 떨어져 나타날 수도 있음을 의미한다. 문맥 벡터의 경우, 윈도우 사이즈가 커지면 자료부족 문제(sparse data problem)를 대처하는 효과가 있으나 단어 임베딩 벡터의 경우, 각 문맥 단어의 위치 정보가 사용되기 때문에 자료부족 문제를 심화시키는 효과가 있다. 왜냐하면 동일한 문맥 단어일지라도 중의성 단어와 떨어진 위치가 다르면 문맥 단어 벡터에서는 동일한 자질로 간주되나 단어 임베딩 벡터에서는 서로 다른 자질로 간주되어 서로 다른 벡터를 만들어 내기 때문에 학습 데이터의 양이 충분하지 않으면 자료부족 문제를 피할 수 없게 된다. 이 문제를 해결하기 위해서는 단순 거리 문맥 대신에 구문 분석기 등을 활용하여 올바른 문맥을 선택하는 방법과 단어 임베딩 벡터에서 서로 다른 위치 정보를 하나의 자질로 간주할 수 있는 방법에 대한 연구가 필요하다.

Table 5는 단어 ‘배’에 대한 단어 임베딩 벡터를 이용한 실험에서 윈도우 크기에 따른 성능을 각 의미별로 보인 것이며 의미를 명사와 동사로 구분하여 성능을 나타내었다.

Table 5. The Accuracy for ‘배(bae)’ Using the Word Embedding Vector with Window Size k

sense	$k=2$	$k=3$	$k=4$	$k=5$	rate(%)
09/NNG	**87.4**	85.6	84.7	82.0	28.1
01/NNG	**76.9**	70.2	72.7	62.0	30.6
02/NNG	72.5	72.5	72.5	**79.4**	25.8
01/VV	75	64.3	67.9	60.7	7.1
03/NNG	**62.5**	**62.5**	**62.5**	50.0	4.1
88/NNG	87.5	**100**	87.5	**100**	2
02/VV	**60.0**	40.0	40.0	40.0	1.3
06/NNG	0	0	0	0	0.3
08/NNG	**66.7**	**66.7**	33.3	33.3	0.8
Noun	**78.2**	75.7	75.7	72.9	91.6
Verb	**72.7**	60.6	63.6	57.6	8.4
Overall	**77.7**	74.4	74.7	71.6	100

‘배’의 동사 의미 6개 중에서 평가 데이터에 나타난 의미가 2개밖에 없었으나, Table 5에서와 같이 동사 의미의 경우에는 근거리 문맥만 사용하였을 때와 그렇지 않을 때의 성능의 편차가 명사 의미의 편차보다 더 크게 나타났다. 이는 동사 의미는 가까운 문맥 단어의 영향을 많이 받고, 명사 의미는 문맥 단어의 거리에 큰 영향을 받지 않았다고 볼 수 있다. 이는 술어의 논항이 술어 주위에 나타나는 일반적인 언어 현상에 부합한다. 또한 명사 의미의 의미 분별에 단순한 윈도우 크기로 문맥 단어를 선정하는 것의 한계를 나타내는 것으로 볼 수 있다. 구문 분석기 등을 활용하여 문맥 단어를 올바르게 선정한다면 더 좋은 성능을 얻을 수 있을 것이다.

Table 6은 실험에 사용된 각 벡터의 차원의 크기를 나타낸다.

Table 6. The Dimension of Vector Spaces

ambiguous word (# of senses)	context words vector (k=5)	word embedding vector			
		k=2	3	4	5
배	8970	200	300	400	500
전기	2181				
고개	6388				
사기	3269				

문맥 단어 벡터를 사용할 때의 차원이 단어 임베딩을 사용한 벡터보다 10배 이상 큰 것을 알 수 있다. 문맥 단어 벡터의 각 축은 기껏해야 문맥의 개수만큼만 1의 값을 가지고 나머지 모든 축은 0의 값을 갖는 희소(sparse) 벡터인 반면, 단어 임베딩 벡터는 Word2Vec의 고유벡터에 의해 모든 차원이 가중치 값을 갖는 벡터이다. 이러한 벡터 차원의 크기는 분류 속도에 영향을 끼친다. 다음 Table 7은 각 실험 시스템의 평가집합에 대한 지연 시간과 4개 중의성 단어 전체의 정확도를 나타낸 표이다. 시스템 지연 시간 측정에 사용한 컴퓨터의 사양은 Windows 7 Home Premium K, Intel(R) Core(TM) i5-3470 3.20GHZ, 4GB DDR3이었다.

Table 7. The Delay Time and Accuracy of Each System

system		loading time (sec)	execution time (sec)	accuracy (%)
Park & Lee 2012		9.14	0.11	83.1
context words vector		**299.90**	**13.99**	**87.0**
word embedding vector (k)	2	15.16	0.34	86.0
	3	17.68	0.48	84.3
	4	20.34	0.67	84.5
	5	23.30	0.86	82.9

단어 문맥 벡터를 이진가중치로 사용한 SVM 분류 방법이 가장 정확률이 높지만 지연시간이 가장 길었다. 반면 (Park & Lee 2012)가 실행 속도가 가장 빨랐지만 성능은 가장 낮았다. 제안한 방법 중에는 윈도우 크기 k=2인 단어 임베딩 방법이 성능은 단어 문맥 벡터보다 근소하게 떨어지지만 가장 빠른 성능을 보였다.

5. 결 론

우리는 단어 의미 분별을 위해 세종 의미 부착 말뭉치로부터 의미 중의성 단어의 문맥을 추출하였다. 추출된 문맥 단어들은 이진 가중치를 가진 문맥 단어 벡터와 단어 임베딩을 이용한 벡터로 각각 변환되었다. 각 벡터는 SVM을 이용해 학습과 분류에 사용되었으며 4가지 중의성 단어에 대해 실험하였다. 실험 결과, 이진 가중치를 사용한 문맥 단어

벡터를 사용한 SVM 분류의 성능이 가장 좋았으며 이전 연구보다 나은 성능을 보였다. 그러나 문맥 단어 벡터의 차원이 상대적으로 커서 다른 방법보다 실행속도가 느린 단점이 있었다. 반면 단어 임베딩 벡터를 사용한 SVM 분류는 문맥 단어 벡터보다 근소하게 성능이 떨어지나 훨씬 빠른 실행 속도를 보였다.

단어 의미 분류의 성능을 향상시키기 위해서는 문맥의 위치 정보에 영향을 받지 않도록 단어 임베딩 방법을 개선할 필요가 있으며, 구문 분석기 등을 이용하여 더 정확한 문맥 단어를 찾는 방법의 연구가 필요하다. 그 외, 문맥 단어 벡터와 단어 임베딩 벡터를 결합하는 방법, 의미망 등을 활용하여 자료 부족 문제를 해소하는 방법, 그리고 의미 부착 말뭉치를 자동으로 확장할 수 있는 방법 등에 대한 연구도 필요하다.

References

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space,” *arXiv:1301.3781*, 2013.

[2] Michael Lesk, “Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone,” in *Proceedings of the 5th Annual International Conference on Systems Documentation*, 1986.

[3] Yong-Gu Lee and Young-Mee Chung, “An Experimental Study on an Effective Word Sense Disambiguation Model Based on Automatic Sense Tagging Using Dictionary Information,” *Journal of the Korean Society for Information Management*, Vol.24, No.1, pp.321-342, 2007.

[4] Jung-Gil Cho and Kwang-Cheul Shin, “A Graph-based Word Sense Disambiguation Using Measures of Graph Connectivity,” *Journal of Korean Institute of Information Technology*, Vol.12, No.6, pp.143-151, 2014.

[5] Dongsuk O, Sangwoo Kang, and Jungyun Seo, “An Iterative Approach to Graph-based Word Sense Disambiguation Using Word2ec,” *Korean Journal of Cognitive Science*, Vol.2, No.1, pp.43-60, 2016.

[6] SangKeun Park, Jeeyeon Choi, and Key-Sun Choi, “Word Sense Disambiguation using Dynamic Sized Window and Frequency Weighting,” *Korea Information Science Society*, pp.441-443, 2014.

[7] Yong Min Park and Jae Sung Lee, “Word Sense Disambiguation using Korean Word Space Model,” *Journal of the Korea Contents Association,* Vol.12, No.6, pp.41-47, 2012.

[8] Myung Yun Kang, Bogyum Kim, and Jae Sung Lee, “Word Sense Disambiguation using Word2Vec,” in *Proceedings of the 27th Annual Conference on Human & Cognitive Language Technology*, pp.81-84, 2015.

[9] Sangwook Kang, Minho Kim, Hyuk-chul Kwon, and Jyhyun Oh, “Word Sense Disambiguation of Predicate using Semi-supervised Learning and Sejong Electronic Dictionary,” *KIISE Transactions on Computing Practices*, Vol.22, No.2, pp.107-112, 2016.

[10] Scikit Learn, 2016 [Internet], http://scikit-learn.org/stable/modules/svm.html.

[11] Yeohoon Yoon, “Word sense disambiguation through the acyclic semantic transition network,” Master thesis, Sogang University, 2003.

박 준 혁

e-mail : ghfkddlsktm@naver.com
2010년～현 재 한국교통대학교 컴퓨터정보공학과 학사과정
관심분야 : 자연언어처리, 기계학습, 의미분별

이 성 욱

e-mail : leesw@ut.ac.kr
1996년 서강대학교 전자계산학과(학사)
1998년 서강대학교 컴퓨터학과(석사)
2003년 서강대학교 컴퓨터학과(Ph.D.)
2003년～2004년 서강대학교 산업기술연구소 연구원
2003년～2005년 서강대학교 정보통신대학원 대우교수
2004년～2005년 LG전자 기술원 선임연구원
2005년～2007년 동서대학교 컴퓨터공학과 전임강사
2007년～현 재 한국교통대학교 컴퓨터정보공학과 교수
관심분야: 자연언어처리, 대화인터페이스, 기계학습, 인공지능