혼합 커널을 활용한 과학기술분야 용어간 관계 추출

(Extraction of Relationships between Scientific Terms based on Composite Kernels)

최 성 필 † 최 윤 수 †
(Sung-Pil Choi) (Yun-Soo Choi)

정 창 후 †† 맹 성 현 †††
(Chang-Hoo Jeong) (Sung-Hyon Myaeng)

요 약 본 논문에서는 합성곱 구문 트리 커널(convolution parse tree kernel)과, 한 문장에서 나타나는 두 개체 간의 관계를 가장 잘 설명하는 동사 상당어구에 대한 개념화를 통해 생성되는 워드넷 신셋 벡터(WordNet synsets vector) 커널을 활용하여 과학기술분야 전문용어 간의 관계 추출을 시도하였다. 본 논문에서 적용한 모델의 성능 평가를 위해서 세 가지 검증 컬렉션을 활용하였으며, 각각의 컬렉션 마다 기존의 접근 방법론 보다 우수한 성능을 보여주었다. 특히 KREC 2008 컬렉션을 대상으로 한 성능 실험에서는, 기존의 합성곱 구문 트리 커널과 동사 신셋 벡터(verb synsets vector)를 함께 적용한 합성 커널이 비교적 높은 성능 향상(8% F1)을 나타내고 있다. 이는 성능을 높이기 위해서 관계 추출에서 많이 활용하였던 개체 자질 정보와 더불어 개체 주변에 존재하는 주변 문맥 정보(동사 및 동사 상당어구)도 매우 유용한 정보임을 입증하고 있다.

키워드 : 관계 추출, 커널 기법, 혼합 커널, 합성곱 구문 트리 커널, 워드넷 신셋 커널, 기계 학습

Abstract In this paper, we attempted to extract binary relations between terminologies using composite kernels consisting of convolution parse tree kernels and WordNet verb synset vector kernels which explain the semantic relationships between two entities in a sentence. In order to evaluate the performance of our system, we used three domain specific test collections. The experimental results demonstrate the superiority of our system in all the targeted collection. Especially, the increase in the effectiveness on KREC 2008, 8% in F1, shows that the core contexts around the entities play an important role in boosting the entire performance of relation extraction.

Key words : Relation Extraction, Kernel Methods, Composite Kernel, Convolution Parse Tree Kernel, WordNet Synset Kernel, Machine Learning

1. 서 론

정보 추출(Information Extraction)은 자연어 처리 및 텍스트 마이닝 분야에서 핵심적인 요소기술로 인식되고 있으며 대용어 참조 해소, 개체명 인식, 관계 추출의 세 가지 요소 기술로 구성된다[1,2]. 위에서 설명한 정보 추출의 세 가지 요소 기술 중에서 관계 추출(Relation Extraction)은 현재까지도 가장 난이도가 높은 분야로 여겨지고 있다[1-3].

본 연구는 특히 과학기술분야에서 널리 활용되고 있는 기술용어를 개체(entity)로 정의하고, 이들 간의 연관관계를 어떻게 추출할 것인가에 대한 기초 연구를 시도하는 측면에서 기존 연구와 차별성이 있다. 또한 최근 관계추출 분야에서 활발하게 연구되고 있는 커널 기반 기계학습 모델을 기반으로 이를 다각적으로 적용하여 성능을 향상시키는 연구를 수행한다.

논문의 구성은 다음과 같다. 우선 2장에서는 관계추출 기술에 대한 선행 연구 사례에 대해서 소개한다. 이어 3장에서는 본 논문에서 적용한 모델인 합성곱 구문 트리 커널과 혼합 커널에 대해서 소개하고, 관계 인스턴스 가지치기(relation instance pruning) 기법에 대해서 설명한다. 관계추출 분야에서 널리 활용되고 있는 세 가지 테스트컬렉션에 대한 실험 방법 및 결과는 4장에서 제시한다. 마지막으로 5장에서 결론을 맺고 향후 연구 방향을 설명한다.

2. 관련 연구

지도 학습 기반 관계 추출(supervised relation extraction)은 처리 기법에 따라 규칙 기반 방법(rule-based methods), 자질 기반 방법(feature-based methods), 그리고 커널 기반 방법(kernel-based methods)으로 구분된다.

· 이 논문은 2009 한국컴퓨터종합학술대회에서 '혼합 커널을 활용한 과학기술분야 용어간 관계 추출'의 제목으로 발표된 논문을 확장한 것임

† 정 회 원 : 한국과학기술정보연구원 정보기술연구실 선임연구원
spchoi@kisti.re.kr
armian@kisti.re.kr
†† 정 회 원 : 한국과학기술정보연구원 정보기술연구실 연구원
chjeong@kisti.re.kr
††† 종신회원 : 한국과학기술원 전산학과 교수
myaeng@kaist.ac.kr

논문접수 : 2009년 8월 13일
심사완료 : 2009년 10월 14일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제12호(2009.12)

자질 기반 방법으로서 Kambhatla(2004)는 최초로 최대 엔트로피 모델(Maximum Entropy Model)을 기반으로 다양한 형태의 어휘적, 구문적, 의미적 자질들을 이용하여 관계 추출을 시도하였다[4]. 이를 기반으로 Guo-Dong et al.(2005)는 지지벡터기계(Support Vector Machines)를 활용하여 더 확장되고 세분화된 자질 정보를 관계 추출에 적용하였다[5]. 이와 유사하게 Zhao et al.(2005)는 모든 세부 자질을 종류별로 구분하고 이를 개별적인 선형 커널로 구성하여 최종적으로 혼합 커널로 결합하는 기법을 제안하였다[6].

커널 기반 기법의 단초는 Zelenko(2003)에서 제시하였다. 최초로 두 개의 구문 분석 트리에 대한 유사도를 재귀적으로 측정하는 연속 부분 트리 커널(contiguous subtree kernel)과 희소 부분 트리 커널(sparse subtree kernel)의 두 가지 구문 트리 커널을 고안하고, 이를 두 가지 이진 관계에 적용하여 매우 높은 성능을 보였다[7].

최근에는 Zhang et al.(2006)이 Collins and Duffy (2001)[8]에서 새롭게 고안한 합성곱 구문 트리 커널(convolution parse tree kernel)을 기반으로 다양한 구조적 자질 정보와 기존의 개체 자질 정보를 결합한 혼합 커널(composite kernel)을 개발하였다[9,10].

3. 커널기반 관계추출 모델

3.1 합성곱 구문트리커널(CPTK)

CPTK는 [8]에서 최초로 고안되었으며, [10]에서 다른 커널 함수와 복합적으로 활용되어 관계추출에 적용되었다. 두 개의 구문 분석 트리의 유사도를 측정하기 위한 커널 함수로서 이를 기반으로 다양한 향상된 기법들이 개발되기도 하였다. CPTK 내에서 특정 구문 트리 T는 다음과 같은 부분 트리의 발생 빈도 벡터로 표현된다.

$$\phi(T) = (\#st_1(T), ..., \#st_n(T)) \quad (1)$$

여기서 $\#st_i(T)$는 T 내에서의 특정 하부 트리 i의 발생 빈도를 나타낸다. 모든 구문 트리는 위와 같은 벡터로 표현되며, CPTK는 두 개의 입력 구문 트리에 대해서 모델 학습 시에 다음의 내적을 계산하게 된다.

$$K(T_1, T_2) = \langle \phi(T_1), \phi(T_2) \rangle \quad (2)$$

그림 1은 특정 구문 트리에 대한 모든 하부 트리를 보여준다. 총 9개의 하부 트리가 존재하며 이들 각각이 현재 구문 트리 벡터의 요소 항목이 된다. 만일 총 N개의 구문 트리가 존재하며 이들 전체를 대상으로 추출한 하부 트리의 종류가 M개라고 한다면, N개의 구문 트리 각각은 M 차원의 벡터로 표현된다. 아래 그림에서 보듯이, 특정 구문 트리의 하부 트리가 되기 위해서는 두 가지의 제약 조건이 있다. 첫째, 하부 트리의 노드는 무조건 2개 이상이어야 한다. 둘째, 하부 트리는 특정 생성규칙(Production Rule, CFP rule)에 부합되어야 한다. 예를 들어, 그림 1에서 [VP → VBD → got]은 하부 트리가 될 수 없다.

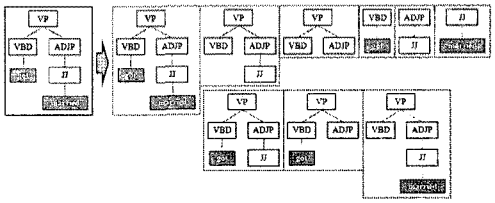

그림 1 구문 분석 트리와 그 하부 트리 집합

구문 트리 각각에 대한 벡터를 구성하기 위해서는 트리 T 내의 모든 하부 트리를 다 조사하여 빈도를 계산해야 하므로 비효율적이다. 커널 기반 방법에서는 두 구문 트리의 유사도만 계산하면 되므로, 각 구문 트리에 대해서 위와 같은 하부 트리 벡터를 구성할 필요가 없이 간접 유도 방식을 활용하여 효율적으로 유사도를 계산한다. 다음은 [8]에서 제시한 간접 유사도 계산 방법이다.

$$\begin{aligned} K(T_1,T_2) &= \langle \phi(T_1), \phi(T_2) \rangle \\ &= \sum_i \#subtree_i(T_1) \cdot \#subtree_i(T_2) \\ &= \sum_i \left(\sum_{n_1 \in N_1} I_{subtree_i}(n_1) \right) \cdot \left(\sum_{n_2 \in N_2} I_{subtree_i}(n_2) \right) \\ &= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \Delta(n_1, n_2) \end{aligned}$$

$N_1, N_2 \rightarrow$ the set of nodes in trees T_1 and T_2.

$$I_{subtree_i}(n) = \begin{cases} 1 & \text{if ROOT}(subtree_i) = n \\ 0 & \text{otherwise} \end{cases}$$

$$\Delta(n_1, n_2) = \sum_i I_{subtree_i}(n_1) \cdot I_{subtree_i}(n_2)$$

위 식에서 가장 시간이 많이 걸리는 부분은 $\Delta(n_1, n_2)$를 구하는 부분이다. 구문 분석 트리의 특성을 활용하여 이를 쉽게 구할 수 있는 방법은 다음과 같다.

1. 만약 n_1과 n_2에서의 CFP (문맥자유문법기반 생성) 규칙이 서로 다르면,
$$\Delta(n_1, n_2) = 0$$
2. 만약 n_1과 n_2 모두 다 pre-terminal (품사 태그) 이면,
$$\Delta(n_1, n_2) = 1 \times \lambda$$
3. 아니면,
$$\Delta(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} (1 + \Delta(ch(n_1, j), ch(n_2, j)))$$
$nc(n_1) \rightarrow$ the child number of n_1,
$ch(n, j) \rightarrow j^{th}$ child of node n
$\lambda \rightarrow$ decay factor

3.2 혼합 커널(Composite Kernel)

부가적으로 관계 추출을 위한 개별 학습 인스턴스마다 추출되는 주요 자질(개체속성, 개체 사이의 동사 의미 속성 등)을 기반으로 유사도를 계산하는 커널과, 앞에서 설명한 CPTK를 선형(linear) 관계로 결합하여 복

합 커널(composite kernel)을 구성하였다. 본 논문에서는 특정 문장에서 두 개체들을 의미적으로 연결하는 핵심 동사들을 워드넷(WordNet)에 사상시켜서 도출되는 동사 신셋(verb synset) 집합을 부가 자질로 활용하였다. 본 논문에서는 이 자질 집합을 신셋 벡터(synset vector)라고 부르기로 한다.

$$K'(R_1,R_2)=K_s(R_1,R_2)+\tau\times K(T_1,T_2) \quad (3)$$

위 식에서 $K_s(R_1, R_2)$는 인스턴스 R_1과 R_2 간의 신셋 벡터(synset vector) 기반 유사도를 계산하는 커널 함수이고, $K(T_1, T_2)$는 앞서 설명한 CPTK 커널을 나타낸다. 가중치 τ는 CPTK 커널의 기여도 조정 역할을 수행한다. 여기서는 CPTK를 위한 트리 유사도 계산 도구로서 [11]에서 개발한 Tree Kernel 도구를 활용하였고, SVM 학습을 위해서는 SVM^{light}[12]를 활용하였다.

3.3 관계 인스턴스 가지치기

커널 함수의 유사도 측정 성능을 향상시키고, 관계 추출을 위한 학습 시에 불필요한 문맥 정보들을 제외시키기 위해서 구문 트리의 일부분만을 남기고 나머지는 제거하는 작업을 수행하였다.

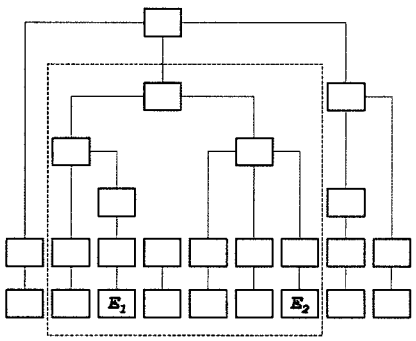

그림 2 최소 완전 트리 가지치기 기법

특정 문장 내에 두 개체 E_1, E_2가 존재할 때, 가지치기(Pruning)가 완료된 부분 트리는 두 개체를 포함하는 가장 작은 완전 트리가 된다.

그림 2에서 보는 바와 같이, 두 개체를 포함하고 있는 가장 작은 완전 트리만을 선별하여 관계추출 학습에 활용하게 된다 [10].

4. 실험 및 토의

4.1 실험 대상 시스템

실험에 사용되는 시스템은 그 특징과 기능에 따라 다음의 4가지로 나뉜다.

표 1에서 알 수 있듯이, CPTK는 구문 트리 유사도를 계산하는 트리 커널만을 활용한 시스템이고, CPTK-SVK는 CPTK에 개별 학습 인스턴스(학습 문장)마다 생성되는 후보 synset 정보를 벡터화한 정보에 대한 유사도를 측정할 수 있는 polynomial kernel을 결합한('+') 복합 커널을 활용한 시스템이다. CPTK-PR은 CPTK에 3.3절에서 설명한 인스턴스 가지치기 기법(최소 완전 트리 기법)을 적용한 시스템이고, CPTK-SVK-PR은 모든 기능을 다 결합한 시스템이다.

표 1 관계 분류 시스템 종류

시스템 종류	설 명
CPTK	· Convolution Parse Tree Kernel
CPTK-SVK	· Composite Kernel - Convolution Parse Tree Kernel + - WordNet Synset Vector Kernel (Polynomial Kernel, degree = 5)
CPTK-PR	· Convolution Parse Tree Kernel · Relation Instance Pruning
CPTK-SVK-PR	· Composite Kernel - Convolution Parse Tree Kernel + - WordNet Synset Vector Kernel (Polynomial Kernel, degree = 5) · Relation Instance Pruning

4.2 CoNLL 관계 식별 컬렉션

(CoNLL Relation Recognition Corpus)[1)]

3장에서 설명한 관계 탐지 및 분류 엔진은 기본적으로 영역 독립적인 시스템이므로 학습 데이터만 존재한다면 모든 분야에서 적용이 가능하다. 본 절에서는 구성된 관계 추출 모델의 보편적인 성능 측정을 위해서 D. Roth가 [13,14]에서 사용한, PLO(Person, Location, Organization) 개체명 간의 연관 관계를 포함하는 신문기사 문장으로 구성된 CoNLL 2004 관계식별 컬렉션을 기반으로 실험을 수행하였다. CoNLL 2004는 테스트 컬렉션 내부에 406개의 "located_in", 394개의 "work_for", 451개의 "orgBased_in", 521개의 "live_in", 268개의 "kill", 그리고 17,007개의 "관계없음" 연관관계가 설정되어 있다. 특정 문장에 여러 개의 연관관계 설정이 되어 있으며, 본 연구에서는 이를 처리하기 위해서 3.3절에서 설명한 Relation Instance Pruning 기법을 활용하여, 동일 문장에서도 해당 관계의 대상 개체 위치에 따라 각기 다른 형태의 학습 인스턴스를 생성할 수 있다.

성능 비교를 위해 [14]에서 제시된 성능 측정 결과를 함께 표시하였다. 학습 및 검증 컬렉션의 규모에 따라 결과가 약간씩 다르게 나올 수도 있겠으나, 개별 관계에 따른 성능 향상이 두드러짐을 볼 수 있다. 특히, "Omniscient" 모드는 [14]에서 다중 학습기를 적용하고, 활용 가능한 모든 자질 집합을 적용한 최적의 성능 결과임에도 불구하고 본 연구에서 개발된 시스템의 성능과 현격

1) http://l2r.cs.uiuc.edu/~cogcomp/Data/ER/ [13,14]

표 2 CoNLL 2004에 대한 성능 검증 비교

구분	work_for			orgBased_in			live_in			kill			평균 성능 (R/P/F)		
	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1
LP [14]	40.4	**72.9**	52.0	36.3	**90.1**	51.7	41.5	68.1	51.6	81.3	**82.2**	**81.7**	49.88	**78.33**	59.25
Omni.[14]	50.5	69.1	58.4	50.2	76.7	60.7	**57.0**	60.7	58.8	**82.1**	74.6	78.2	59.95	70.27	64.02
CPTK-PR	**97.2**	43.7	**60.3**	**68.7**	61.1	**64.7**	56.1	**83.6**	**67.2**	75.0	79.2	77.1	**74.28**	66.94	**67.32**

한 차이를 보이고 있다. 더불어 [14]에서는 "nothing" 관계에 대한 관계추정 성능을 제시하지 못하고 있다. 다시 말해서, 관계 분류 이전 단계에서 필수적인, 관계가 포함된 문장을 식별하는 관계 탐지(Relation Detection) 기능을 갖추고 있지 않다.

표 2에서 CoNLL 2004 버전에 대한 성능 실험 결과를 제시하였다. 이 실험을 위해서 전체 컬렉션의 80%에 해당하는 1,311개의 문장 집합이 무작위로 선정되어 학습에 활용되었고, 나머지 20%인 327개의 서로 다른 검증 컬렉션을 이용하여 성능을 측정하였다. 정확도와 재현율 측면에서 볼 때, 각 관계에 따라 편차가 있으나, 본 연구에서 개발된 CPTK 방법이 비교적 높은 성능을 보이고 있음을 알 수 있다. 특히 전체 대상 관계 집합에 대한 평균 F_1 값은 가장 높은 수치를 나타내고 있다.

4.3 BioText(Bioscience Text Analysis) 컬렉션

과학기술분야 데이터를 대상으로 관계추출 성능 검증을 하기 위해서, 본 연구에서는 버클리 대학에서 수행중인 BioText Project[2]의 일환으로 구축된 관계 분류 컬렉션을 활용하였다.

실질적으로 위의 트리플들은 문장 단위가 아니라, 초록이나 초록 내의 일부 텍스트를 기준으로 구성되었다. 따라서 [15]에서는 위의 데이터를 기반으로 생물학 전공자를 활용하여, 데이터베이스 내의 개별 문장에 대한 관계 태깅을 시도하였다. 본 연구에서는 그 구축 결과를 활용하여 CPTK 기반 관계추출기의 성능 측정을 수행하였다. 최대한 객관적인 성능 비교 검증을 위해서 [15]에서 실험한 것과 마찬가지로 상위 빈도 6개의 관계에 대해서 문장 단위 학습 및 관계 분류 성능 실험을 수행하였다. 참고로 [15]에서는 실험 대상 관계에 대한 상세한 정보는 제시되어 있지 않다.

본 연구에서는 CPTK 모델의 학습을 위해서 전체 컬렉션의 90%에 해당하는 총 1,307개의 관계 학습 인스턴스(Relation Instance)가 각 관계 클래스별로 무작위로 선정되어 활용되었으며, 나머지 10%에 해당하는 142개의 인스턴스를 기반으로 성능 검증을 수행하였다. [15]에서는 각 관계별 성능이 제시되지 않고 있으며, 6개 종류의 관계에 대한 평균 F_1 값만 나타내고 있다.

위 표 3에서도 알 수 있듯이 전체적인 성능이 기존 방법보다 높음을 알 수 있다. 그러나 개별 관계에 따른 성능 편차가 심하게 나타나고 있다. 특히 "upregulates" 관계에 대한 분류 성능이 매우 낮다. 또한 "binds"는 대상 학습 집합의 규모가 가장 큰데도 불구하고 그 분류 성능이 비교적 낮게 나온다. 이는 문장 내에서 활용되는 문맥 정보의 다양성과 구문 구조의 일반성에서 오는 관계별 특이성 약화에 기인한다고 볼 수 있다. 따라서 관계를 구성하는 개별 개체명(여기서는 단백질 및 세포 이름)에 대한 자질 정보를 학습에 반영하거나, 구문 구조 정보를 수정하여, 현재 컬렉션의 특성에 맞는 커널 함수를 재구성함으로써 성능 향상을 꾀할 수 있을 것이다.

4.4 KREC 2008

본 절에서는 과학기술분야 용어간 관계 추출 성능 평가를 위해서 자체적으로 구축한 후보연관관계 기반 관계추출 테스트 컬렉션(KREC 2008)을 활용하여 최상위 빈도를 가지는 4가지 관계에 대한 자동 추출 성능 실험을 수행하였다. 최상위 빈도 4가지 관계는 빈도순에 따라 "use(사용하다)", "change(변화하다)", "cause(야기하다)", "make(만들다)"로 구분된다. 아래 표 4는 대상 관계에 대한 상세 정보를 나타낸다.

실험은 학습 모드에 따라 총 4가지 시스템에 대해서 이루어진다. 앞서 3.2절에서 설명한 바와 같이, 서로 연관성을 가지는 기술용어 쌍과 그들 사이에 존재하는 동사의 개념화(워드넷 사상)를 통하여 후보 연관관계 집합이 도출될 수 있으므로 이 후보 연관관계를 벡터화하여 기존의 CPTK 커널과 복합 커널을 구성할 수 있다.

표 3 BioText에 대한 성능 실험 결과

구분	binds			requires			upregulates			inactivates			synergizes with			stimulates		
	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1
NN [15]	(Avg. F_1) 52.90																	
CPTK-PR	(Avg. F_1) 54.18																	
	43.9	80.6	56.9	60.0	33.3	42.9	34.8	38.1	36.4	92.3	80.0	85.7	75.0	33.3	46.2	66.7	50.0	57.1

2) http://biotext.berkeley.edu/

표 4 실험 대상 관계 정보

대상 관계(WordNet Synset)	관계의미	문장개수
change,alter,modify	변화하다	62
induce,stimulate,cause,have,get,make	야기하다	61
make,create	만들다	59
use,utilize,utilise,apply,employ	사용하다	121
합계		303

또한 연관 관계를 맺는 대상 개체명들의 위치에 따라서 구문 트리에서의 문맥 정보를 제약시켜 주는 관계 인스턴스 가지치기 기법도 적용할 수 있다. 본 실험에서는 이들 기법들의 효과를 검증하기 위해 표 1에서 제시한 바와 같이 다양한 형태의 시스템을 구성하고 각각에 대한 성능 검증을 수행하였다. 표 5는 위에서 제시된 4가지 시스템에 따른 성능 측정 결과를 보여준다.

표 5 KREC 2008 컬렉션에 대한 성능 실험 결과

구분	평균 성능(R/P/F)		
	재현율	정확도	F_1
CPTK	64.05	57.75	56.76
CPTK-SVK	79.86	63.77	61.00
CPTK-PR	77.63	65.90	66.52
CPTK-SVK-PR	**81.94**	**73.52**	**74.28**

표 5에서 나타나듯이 관계 인스턴스 가지치기 기법과 신셋 벡터(synset vector) 커널의 효과는 명확하게 드러난다. 특히 평균 성능 측면에서 살펴봤을 때, CPTK-PR에서 신셋 벡터(synset vector)의 적용은 비약적인 성능 향상(F-measure로 약 8%)을 가져온다. 비록 소규모 데이터를 기반으로 실험을 수행하였으나, 각 부가 기능별로 성능 향상 효과에 대한 직접적인 실증적 근거를 확보할 수 있었으며, 현재까지 구축된 테스트 컬렉션의 검증과 반자동 추가 구축에 중요한 역할을 수행할 수 있는 시스템으로서의 성능 평가를 할 수 있었다.

5. 결론 및 향후 연구 방향

본 논문에서는 합성곱 구문 트리 커널과, 한 문장에서 나타나는 두 개체 간의 관계를 가장 잘 설명하는 동사 상당어구에 대한 개념화를 통해 생성되는 워드넷 신셋 벡터 커널을 활용하여 과학기술분야 전문용어 간의 관계 추출을 시도하였다. 본 논문에서 적용한 모델의 성능 평가를 위해서 세 가지 검증 컬렉션을 활용하였으며, 각각의 컬렉션 마다 기존의 접근 방법론 보다 우수한 성능을 보여주었다.

향후 연구 과제로서 개체 자질 및 문맥 자질 등을 결합한 커널을 구성하고 이를 합성곱 구문 트리 커널과 결합한 확장 혼합 커널에 대한 성능 실험을 계획 중에 있다.

참 고 문 헌

[1] Bunescu, R. C., Mooney, R. J., "A Shortest Path Dependency Kernel for Relation Extraction," *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, B.C.*, pp.724-731, 2005.

[2] Culotta, A., Sorensen, J., "Dependency Tree Kernels for Relation Extraction," *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 2004.

[3] Bunescu, R. C., Mooney, R. J., Subsequence Kernels for Relation Extraction, *Advances in Neural Information Processing Systems*, 2006.

[4] Kambhatla N., "Combining lexical, syntactic and semantic features with Maximum Entropy models for extracting relations," *ACL'2004(Poster)*, pp.178- 181. 21-26 July 2004, Barcelona, Spain.

[5] GuoDong Z., Su J. Zhang J. and Zhang M., "Exploring various knowledge in relation extraction," *ACL'2005*, pp.427-434, 25-30 June, AnnArbor, Michgan, USA, 2005.

[6] Zhao, S. B., Grishman, R., "Extracting Relations with Integrated Information Using Kernel Methods," *ACL-2005*, 2005.

[7] Zelenko, D., Aone, C., Richardella, A., "Kernel Methods for Relation Extraction," *Journal of Machine Learning Research 3*, pp.1083-1106, 2003.

[8] Collins, M., Duffy, N., "Convolution Kernels for Natural Language," *NIPS-2001*, 2001.

[9] GuoDong Z., Min Z., Dong H. J., QiaoMing Z., "Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information," *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague*, pp.728-736, June 2007.

[10] Zhang, M., Zhang, J., Su, J., Zhou, G., "A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features," *21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pp.825-832, 2006.

[11] Moschitti A., "Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees," *Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany*, 2006.

[12] Joachims T., SVM Light, http://svmlight.joachims.org/, 2008.

[13] Roth D., Yih W., "Probabilistic Reasoning for Entity & Relation Recognition," *COLING'02*, Aug. 2002.

[14] D. Roth and W. Yih, "A Linear Programming Formulation for Global Inference in Natural Language Tasks," *CoNLL'04*, May. 2004.

[15] Rosario B., Hearst M., Multi-way Relation Classification: Application to Protein-Protein Interaction, *HLT/EMNLP'05, Vancouver*, 2005.