시맨틱 구문 트리 커널을 이용한 생명공학 분야 전문용어간 관계 식별 및 분류 연구

A Study on the Identification and Classification of Relation Between Biotechnology Terms Using Semantic Parse Tree Kernel

최성필(Sung-Pil Choi)*, 정창후(Chang-Hoo Jeong)**
전홍우(Hong-Woo Chun)***, 조현양(Hyun-Yang Cho)****

목 차

초 록

본 논문에서는 단백질 간 상호작용 자동 추출을 위해서 기존에 연구되어 높은 성능을 나타낸 구문 트리 커널을 확장한 시맨틱 구문 트리 커널을 제안한다. 기존 구문 트리 커널의 문제점은 구문 트리의 단말 노드를 구성하는 개별 어휘에 대한 단순 외형적 비교로 인해, 실제 의미적으로는 유사한 두 구문 트리의 커널 값이 상대적으로 낮아지는 현상이며 결국 상호작용 자동 추출의 전체 성능에 악영향을 줄 수 있다는 점이다. 본 논문에서는 두 구문 트리의 구문적 유사도(syntactic similarity)와 어휘 의미적 유사도(lexical semantic similarity)를 동시에 효과적으로 계산하여 이를 결합하는 새로운 커널을 고안하였다. 어휘 의미적 유사도 계산을 위해서 문맥 및 워드넷 기반의 어휘 중의성 해소 시스템과 이 시스템의 출력으로 도출되는 어휘 개념(WordNet synset)의 추상화를 통한 기존 커널의 확장을 시도하였다. 실험에서는 단백질 간 상호작용 추출(PPII, PPIC) 성능의 심층적 최적화를 위해서 기존의 SVM에서 지원되던 정규화 매개변수 외에 구문 트리 커널의 소멸인자와 시맨틱 구문 트리 커널의 어휘 추상화 인자를 새롭게 도입하였다. 이를 통해 구문 트리 커널을 적용함에 있어서 소멸인자 역할의 중요성을 확인할 수 있었고, 시맨틱 구문 트리 커널이 기존 시스템의 성능향상에 도움을 줄 수 있음을 실험적으로 보여주었다. 특히 단백질 간 상호작용 식별 문제보다도 비교적 난이도가 높은 상호작용 분류에 더욱 효과적임을 알 수 있었다.

ABSTRACT

In this paper, we propose a novel kernel called a semantic parse tree kernel that extends the parse tree kernel previously studied to extract protein-protein interactions(PPIs) and shown prominent results. Among the drawbacks of the existing parse tree kernel is that it could degenerate the overall performance of PPI extraction because the kernel function may produce lower kernel values of two sentences than the actual analogy between them due to the simple comparison mechanisms handling only the superficial aspects of the constituting words. The new kernel can compute the lexical semantic similarity as well as the syntactic analogy between two parse trees of target sentences. In order to calculate the lexical semantic similarity, it incorporates context-based word sense disambiguation producing synsets in WordNet as its outputs, which, in turn, can be transformed into more general ones. In experiments, we introduced two new parameters: tree kernel decay factors, and degrees of abstracting lexical concepts which can accelerate the optimization of PPI extraction performance in addition to the conventional SVM's regularization factor. Through these multi-strategic experiments, we confirmed the pivotal role of the newly applied parameters. Additionally, the experimental results showed that semantic parse tree kernel is superior to the conventional kernels especially in the PPI classification tasks.

키워드: 관계 추출, 커널 기반 방법, 구문 트리 커널, 시멘틱 구문 트리 커널, 어휘 중의성 해소
Relation Extraction, Kernel-based Approaches, Parse Tree Kernels, Semantic Parse Tree Kernels, Word Sense Disambiguation

---

* 한국과학기술정보연구원 정보기술연구실 선임연구원(spchoi@kisti.re.kr) (제1저자)
** 한국과학기술정보연구원 정보기술연구실 선임연구원(chjeong@kisti.re.kr) (교신저자)
*** 한국과학기술정보연구원 정보기술연구실 선임연구원(hw.chun@kisti.re.kr) (공동저자)
**** 경기대학교 문헌정보학과 교수(hycho@kyonggi.ac.kr) (공동저자)
논문접수일자: 2011년 4월 13일 최초심사일자: 2011년 4월 17일 게재확정일자: 2011년 5월 13일
한국문헌정보학회지, 45(2): 251-275, 2011. [DOI:10.4275/KSLIS.2011.45.2.251]

1. 개 요

단백질 간 상호 작용(Protein-Protein Interaction, PPI)은 인접한 두 단백질 분자들 상호간의 직접적인 관계뿐만 아니라, 수십 나노미터 떨어진 수용성 단백질(hydrated protein)들 사이에서 수용체(aqueous solution), 전해질(electorlyte) 등에 의해 이루어지는 간접적인 상호작용까지도 포괄한다(Wikipedia 2010). 이러한 PPI 정보는 무수한 생물학적 기능을 식별하고 분석하는데 중요한 역할을 할 수 있으며, 지금까지 많은 생의학 분야 연구자들이 생화학적 기법(biochemical methods), 생물 물리학적 기법(biophysical methods) 등을 기반으로 실험 혹은 이론적 분석에 의거하여 관련 연구를 수행해 왔다.

새로운 PPI의 발견은 대규모 연구결과의 일부로서 혹은 그 자체로 논문, 특허, 기술 보고서 등으로 발표되며, 대부분 일반 서술식 문장 형태로 기술된다. 따라서 언어처리 및 기계학습 기술 등을 적용하여 텍스트에서 이러한 정보를 자동으로 식별, 추출하는 작업은 다음의 두 가지 측면에서 매우 중요하다. 첫째, 과거에 인지된 PPI의 정형적이고 체계적인 수집이다. 기존에 발견된 PPI를 수집하고 체계적으로 정리한 데이터베이스는 신규 연구의 중요한 기반 자원이 될 수 있다. 관련 연구자들은 자신의 연구에 앞서 이 자원을 검색해 봄으로써 연구의 시발점 및 방향 등을 설정하는데 도움을 받을 수 있다. 이미 전 세계적으로 과거에 발견된 단백질, 유전자 혹은 분자 수준에서의 상호작용 정보를 분야별로 체계적으로 수집하여 정형화한 데이터베이스들이 상당수 존재한다(MIPS,[1] DIP,[2] MINT,[3] Pathways Knowledge Base,[4] HPRD[5] 등). 그러나 이들 대부분이 전문가를 이용한 수작업에 의해서 구축된 DB들이며, PPI 자동 추출은 매우 제한적으로 수행되고 있다. 따라서 기존에 발견된 PPI에 대한 보다 정확하고 포괄적인 정보 수집과 활용이라는 측면에서 자동 PPI 추출에 관한 연구는 필수적이라고 볼 수 있다. 두 번째로 신규로 발견되는 PPI에 대한 신속한 수집이다. 연구자들은 자신이 발견하거나 확증한 PPI에 대한 내용을 즉시 국제학술대회나 학술지에 투고한다. 그러나 큐레이터(curator)라고 불리는 전문가들에 의해서 수작업으로 이들 PPI를 수집한다면 최종적으로 데이터베이스에 등록되기까지는 상당한 시간이 소요되기 마련이다. 따라서 PPI 데이터베이스의 신규성을 유지하고 신속한 자료 확장을 위해서는 자동 PPI 추출 기술이 매우 중요하다.

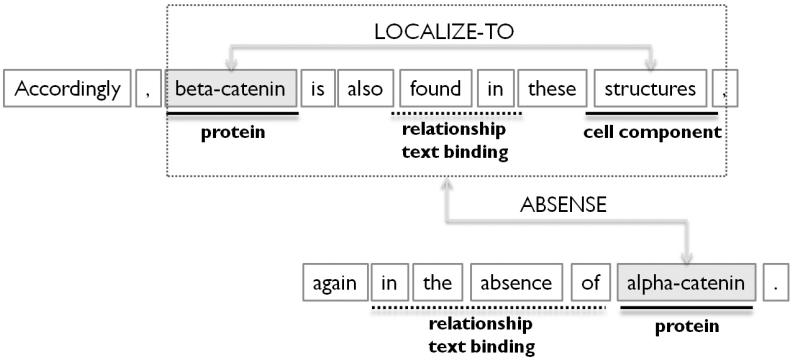
단백질 간 상호작용 자동 추출(Protein-Protein Interaction Extraction, PPIE)은 텍스트 내에 표현된 단서 어휘 및 구문 구조 혹은 기타 부가 자질들을 활용하여 그 텍스트 내에 출현한 다수의 단백질들 간의 상호작용에 관한 정보를 자동으로 추출하는 기술이며 PPI 식별(PPI Identification, PPII)과 PPI 분류(PPI Classification, PPIC)로 구성된다. PPII는 여러 종류

1) Munich Information Center for Protein Sequences, 〈http://www.helmholtz-muenchen.de/en/mips/〉.
2) Database of Interacting Proteins, 〈http://dip.doe-mbi.ucla.edu/dip/Main.cgi〉.
3) Molecular INTeraction database, 〈http://mint.bio.uniroma2.it/mint/Welcome.do〉.
4) 〈http://www.ingenuity.com/〉.
5) Human Protein Reference Database, 〈http://www.hprd.org/〉.

의 단백질 이름을 포함하는 문장 혹은 단락이 그 단백질들 간의 상호작용을 표현하고 있는지 여부를 판단하는 기술이다. 기계학습 관점에서 볼 때, PPII 문제는 이진 분류(binary classification) 모델로 표현할 수 있으며, 현재까지 많은 연구가 진행되어 왔다. 더불어 PPIC는 PPI를 포함한 문장 혹은 단락을 대상으로 보다 심층적인 분석을 통해서 구체적인 상호작용의 종류를 결정하는 작업이다. 상호작용의 종류가 3개 이상이므로 다중 분류 모델로 설명될 수 있으며 현재까지도 연구가 계속 진행 중이다. 또한 PPII와 PPIC는 하나로 결합되어 단일 다중 분류 모델로도 표현될 수 있다.[6] 〈그림 1〉은 단백질 및 그 상호작용을 표현하고 있는 문장과 그 내부 구조를 보여준다.

〈그림 1〉의 문장은 두 개의 단백질 이름을 포함하고 있다. 그림에서 보는 바와 같이 문장의 서두에 출현하는 암 유발 단백질의 일종인 "베타 카테닌(beta-catenin)"은 이 문장의 이전 문장에서 제시된 "these structures"라고 표현된 특정 세포 구성 요소(cell component)와 "LOCALIZE-TO" 관계로 연결되어 있다. 따라서 이는 PPI가 아니다.[7] 이에 반해서 문장의 말미에 출현하는 "알파 카테닌(alpha-catenin)"은 비록 직접적이지는 않으나 위의 세포 구성요소에 포함된 "beta-catenin"과 "ABSENSE"라는 관계로 맺어져 있다. 따라서 이 문장은 하나의 PPI를 포함하고 있음을 알 수 있다. 위 그림에서 "relationship text binding"은 PPI의 종류를 핵심적으로 설명하고 있는 단서 단락 정보이다.

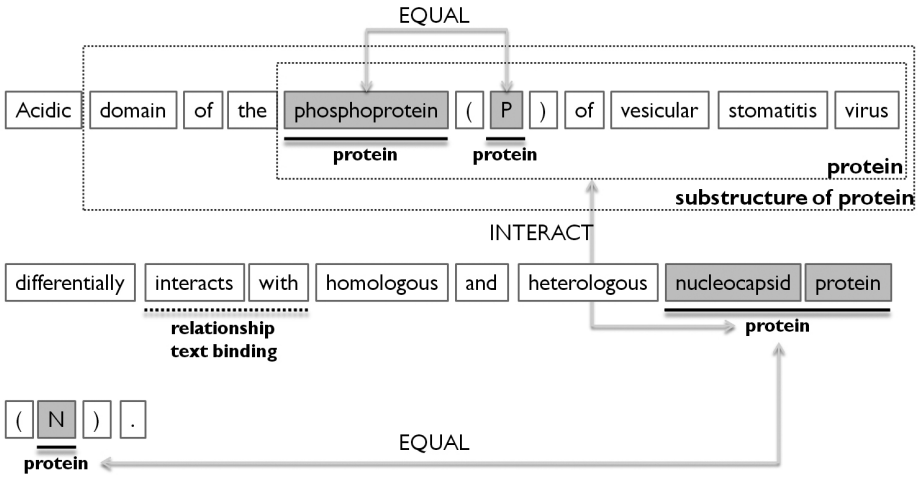
〈그림 2〉의 문장은 〈그림 1〉에서보다 더 복잡하고 다양한 형태의 PPI를 내포하고 있다. 우선 "인단백질(phosphoprotein)"과 직접적으로 인접해 있는 약어인 "P"는 서로 동일관계("EQUAL")로 연결되어 있다. 또한 "소수포성 구내염 바이러스(vesicular stomatitis virus)"의 인단백질(phosphoprotein)은 뉴클레오캡시드 단백질(nucleocapsid protein, NP)과



〈그림 1〉 BioInfer 말뭉치의 단일 PPI 포함 문장 예

6) 만일 상호작용의 종류가 N개라고 한다면, "상호작용 불포함" 클래스를 추가하여, N+1개의 클래스를 가지는 단일 다중 분류 모델로 설명될 수 있으나, 자료 희귀성 문제로 인해서 자주 활용되지는 않는다.

7) PPI는 단백질과 단백질 사이의 상호작용을 의미한다. 위의 관계는 단백질과 세포 구성요소와의 상호작용이므로 PPI라고 볼 수 없다.



〈그림 2〉 BioInfer 말뭉치의 다중 PPI 포함 문장 예

"INTERACT" 관계로 연관되어 있다. 일반적으로 "NP" 혹은 "NCP"로 축약되는 뉴클레오캡시드 단백질은 이 문장에서는 "N"으로 축약되어 원본 용어와 동일관계로 엮여 있다. 따라서 이 문장은 총 3개의 PPI를 포함하고 있다.

위에서 알 수 있듯이 문장 혹은 단락 내에서의 PPI 추출은 쉽지 않다. 특히 위의 〈그림 1〉, 〈그림 2〉와 같이 단일 문장에 두 개 이상의 상호작용 정보를 포함하고 있는 경우가 빈번하다. 이러한 상황에서 동일한 문장을 바탕으로 서로 다른 PPI를 식별하고, 이들의 유형을 분류하는 작업은 매우 어려운 작업이다. 앞에서 지적하였듯이, 만일 이러한 작업들을 수작업으로 수행한다면 고도의 언어지능과 분야지식을 소유하고 경험이 풍부한 전문가들이 동원되어야 한다.

본 논문에서는 이러한 PPI 자동 추출을 위해서 기존에 연구되어 높은 성능을 나타낸 구문 트리 커널을 확장한 시맨틱 구문 트리 커널을 제안한다. 기존 구문 트리 커널의 문제점은 구문 트리의 단말 노드를 구성하는 개별 어휘에 대한 단순 외형적 비교로 인해, 실제 의미적으로는 유사한 두 구문 트리의 커널 수치가 상대적으로 낮아지는 현상이며 결국 PPI 자동 추출의 전체적인 성능에 악영향을 줄 수 있다는 점이다. 이를 극복하기 위해서 본 논문에서는 두 구문 트리의 구문적 유사도(syntactic similarity)와 어휘 의미적 유사도(lexical semantic similarity)를 동시에 효과적으로 계산하여 이를 결합하는 새로운 커널을 고안하였다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 현재까지 PPI 자동 추출과 관련한 연구 성과를 분석하고 장단점을 제시한다. 이어 3장에서는 본 논문에서 새롭게 제시하는 시맨틱 구문 트리 커널에 대해서 상세하게 다룬다. PPI 인식 및 분류를 위해 본 논문에서 개발한 시스템의 전체 아키텍처와 특징을 4장에서 설명하고, 5장에서는 다양한 PPI 말뭉치를 기반으로 실험을 수행한 결과를 제시하며 결과에 대한 분석도 병행하였다. 마지막으로 6장에서는 본 논문의 결론과 향후 연구 방향을 제시한다.

2. 관련 연구

PPI 자동 추출에 관한 연구는 그 중요성으로 인해 매우 활발하게 진행되어 왔으며 다양한 기법들이 소개되었다. Zhou and He(2008)는 최근까지 연구된 PPI 추출 기법을 전산언어학 기반 방법, 규칙 기반 방법 그리고 기계학습 및 통계적 기법의 세 가지 종류로 분류하여 설명하고 있다.

우선 전산언어학 기반 기법에서는 PPI를 표현할 수 있는 대표적인 문장 구조를 분석하여 이를 문법으로 구성한다. 이러한 요소 문법들은 상호작용 추출을 위한 특화된 언어분석시스템(품사태거, 기저구인식기, 구문분석기 등)의 기반 문법으로 활용된다. PPI 자동 추출 기술의 특성에서 볼 때, 문장에 대한 심층 분석은 필수적이며 그 분석정도에 따라 shallow parsing을 활용한 기법과 full parsing 기반 기법으로 나눌 수 있다. Shallow parsing 기법은 문장을 요소 기저구(base chunk)들로 분리하고 이들 기저구 간의 지역적 의존관계를 파악함으로써 PPI 포함문장에 대한 식별을 가능하게 하였다(Sekimizu, Park, and Tsujii 1998; Gondy, Hsinchun, and Martinez 2003). 이에 반해서 full parsing 기법은 단백질, 유전자 혹은 세포 간의 상호작용을 식별할 수 있는 어휘 분석기와 확장된 문맥자유문법(CFG)을 구성하여 이를 기반으로 특화된 구문분석을 수행한다(Temkin and Gilder 2003; Nikolai et al. 2004). 이렇게 도출되는 구문구조의 패턴을 파악함으로써 PPI 포함문장을 식별하였다.

두 번째로 규칙 기반 기법은 상호작용 표현의 단서가 될 수 있는 어휘적 패턴 집합을 수작업으로 정의하고, 이를 기반으로 문장에서 이들 패턴과 일치하는 부분을 찾는 과정을 수반한다. 이 범주에 속하는 방법의 하나로서 Blaschke et al.(1999)은 상호작용 단서 어휘집합을 수집하고 이를 기반으로 어휘적 규칙(lexical rules)을 고안하여 PPI에 적용하였다. 이에 따라, 문장 내에서 발견한 어휘적 규칙에 대한 신뢰도를 자동으로 계산하여 이를 추출된 PPI의 신뢰도로써 활용하였다. Ono et al.(2001)은 부정 표현 구조까지도 포괄하는 어휘 및 구문 자질 기반의 상호작용 추출 패턴을 정의하고, 효모의 일종인 사카로미세스 세레비시아(Saccharomyces cerevisiae)와 대장균속 세균인 에스케리치아 콜리(Escherichia coli)에 관한 문서를 대상으로 실험한 결과 높은 성능을 보여주었다. 더불어 Fundel et al.(2007)은 의존 구문 트리 기반의 관계 추출 모델을 제안함으로써 고수준 자연어 처리 시스템을 적용한 관계 추출의 중대한 발판을 마련하였다. LLL과 HPRD50을 이용한 실험에서 각각 82%, 78%(F-score)의 높은 성능을 나타내었다.

마지막으로 기계학습 및 통계적 기법은 가장 최근에 도래한 기법으로서, 지도학습 혹은 반지도 학습 기반의 기계학습 모델을 적용하여, 미리 수작업으로 구성된 학습 집합을 기반으로 관계 및 상호작용을 표현하는 핵심 단서인 자질 집합을 자동으로 추출하고 이를 학습에 적용한다. 확장성 및 효율성 측면에서 가장 높은 성능을 나타내고 있으며, 현재도 활발하게 연구가 진행되고 있다(Andrade and Valencia 1998; Marcotte, Xenarios, and Eisenberg 2001; Craven and Kumlien 1999).

이 범주에 속하는 기법들 중에서 특히 커널

기반의 PPI 자동 추출에 관한 연구가 활발하다. Airola et al.(2008)은 기존 의존 구문 트리 커널의 단점을 극복하기 위해서 후보 문장들에 대한 의존 구문 트리를 그래프로 변형하고 이에 그래프 커널(graph kernel)을 이용하여 단백질 간 상호작용 추출 시도를 하였으나, 기존의 기법에 비해서 나은 성능을 나타내지는 못하였다. 한편, Miwa et al.(2009)은 단어자질 커널, 구문트리 커널, 그래프 커널 등을 모두 적용한 혼합 커널을 구성하여 앞에서 소개한 총 5가지의 말뭉치를 대상으로 실험을 수행하였다. 그러나 적용한 기법의 다양성이나 광범위한 단서 자질의 적용에도 불구하고 성능은 일반적인 수준이었다. 특히 Fundel et al.(2007)과 비교해서는 오히려 성능이 낮게 나타났다(LLL: 80.1%, HPRD50: 70.9%).

기존 연구와 관련하여 한 가지 주지할 사실은 대부분의 연구가 단백질 간 상호작용 포함 문장 혹은 단락 식별(PPII)에 국한되어 수행되었다는 점이다. 본 논문에서는 기존 구문 트리 커널의 단점을 개선한 시맨틱 구문 트리 커널(Semantic Parse Tree Kernel)을 제안하고 이를 PPII 뿐만 아니라 단백질 간 상호작용 분류(PPIC)에도 적용한다. 다양한 학습 및 실행 옵션을 적용하여 PPI 자동추출의 성능을 최적화시킬 수 있는 방안도 함께 살펴본다.

3. 시맨틱 구문 트리 커널

3.1 구문 트리 커널의 문제점

합성곱 구문 트리 커널의 기본적인 개념은 구문 트리를 요소 하부 트리로 분리하고 이들 하부 트리를 벡터 공간의 개별 축(axis)으로 전사시킴으로써 M개의 하부 트리에 대해서 M차원의 벡터 공간을 구성하는 것이다. 이 때 개별 구문 트리는 벡터공간의 특정 벡터로 전사된다. 벡터 공간으로 전사된 구문 트리 집합 쌍은 그들 간의 내적을 계산함으로써 유사도를 측정할 수 있으며, 이 내적 값이 바로 구문 트리 커널의 출력이다.

트리 커널은 하부 트리 분리 방법에 따라 Vishwanathan and Smola(2003)가 제안한 부분트리 커널(SubTree Kernel, STK)과 Collins and Duffy(2001)가 고안한 부분집합트리 커널(SubSet Tree Kernel, SSTK)로 나뉜다. 부분트리 기법은 트리 내에서 특정 노드의 모든 자식 노드로 구성된 부분 트리를 구성하는 것이다. 따라서 모든 부분 트리는 말단 자식 노드로서 전체 트리의 잎 노드(leaf node)를 가져야 하며, 구문 생성 규칙에 위배되지 말아야 한다. 이에 반해서 부분집합 트리 기법은 부분트리 기법보다 더 일반화된 방법으로서, 특정 부분 트리가 반드시 전체 트리의 잎 노드(leaf node)를 가질 필요는 없다. 다시 말해서, 구문 생성 규칙에 위배되지만 않는다면, 특정 노드에서 출발하여 그 노드의 자식 노드 중 일부분을 포함할 수 있으며, STK 기법보다 훨씬 많은 부분 트리를 생성한다. Moschitti(2006)에 의하면, 부분트리 커널(Subtree kernel, STK)은 부분집합트리 커널에 비해서 성능이 매우 저조하게 나타났다. 한편 Moschitti(2006)는 이들 두 가지 커널을 빠르게 계산할 수 있는 알고리즘을 개발하고, 이를 의미역 결정(semantic role labeling)에 활용하여 괄목할 만한 성능을 보여

주었다.

이러한 장점에도 불구하고, 구문 트리 커널은 두 개의 구문 트리를 비교함에 있어서 각 단말 노드를 구성하는 개별 어휘에 대한 단순 비교/매칭으로 인해 실질적으로는 매우 유사한 두 구문 트리의 유사도를 낮추는 현상을 발생시킨다. 비록 매우 단순한 예제이지만 〈그림 3〉에서 이와 같은 현상을 구체적으로 보여준다.

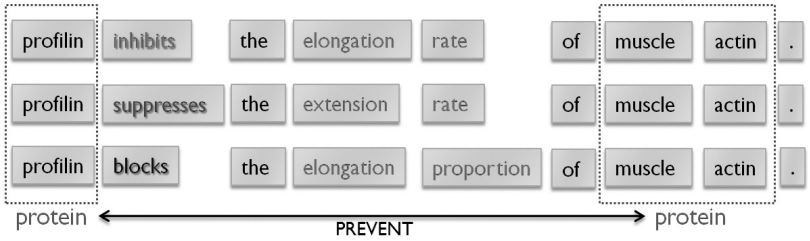
〈그림 3〉의 세 문장은 모두 동일한 의미를 표현하고 있으며, PPI의 관점에서도 모두 "profilin"과 "muscle actin"을 "PREVENT" 관계로 표현하고 있다. 각 문장의 구문 구조도 모두 동일하므로 개별 문장 쌍에 대한 커널 값(유사도)은 매우 높게 나타나야 한다. 만일 Moschitti(2006)가 개발한 알고리즘을 그대로 적용한다면, 단말 노드를 구성하는 개별 어휘들에 대한 단순 외형적 비교로 인해서 우리가 예상하는 수치보다 낮은 커널 값을 도출하게 된다. 따라서 위 그림에서 보듯이 "inhibit", "suppress", "block" 등의 동사들과 "elongation", "extension", "rate", "proportion"과 같은 명사들에 대한 의미 분석을 통해서 가능한 모든 어휘들에 대한 개념화(conceptualization)가 이루어지면 위와 같은 문제를 해결하고 더 나아가서 높은 성능의 PPI 자동 추출 시스템을 구성할 수 있다.

3.2 시맨틱 구문 트리 커널

본 논문에서 제안하는 확장된 구문 트리 커널의 기본적인 개념은 기존의 구문 트리 커널에서 강조되었던 두 문장 간의 구문적 유사도를 기반으로 여기에 개별 어휘들의 개념화를 통해 의미적 유사도를 더욱 확대 적용하는 것이다. 결론적으로 두 구문 트리의 구문적 유사성과 의미적 유사성을 동시에 계산하여 이를 결합하는 새로운 형태의 커널이 시맨틱 구문 트리 커널이다. 아래에서도 설명되겠지만 두 구문 트리의 의미적 유사도 계산은 (1) 개별 구문 트리를 구성하는 어휘들의 문맥 기반 개념 정보를 생성하고, (2) 이를 바탕으로 어휘들의 단순 형태적 비교가 아닌 개념적 비교를 수행함으로써, (3) 이를 기존의 합성곱 구문 트리 커널에 적용함으로써 수행된다. 문맥 기반 어휘 개념은 구문 트리의 단말 노드를 구성하는 어휘들에 대한 중의성 해소(Word Sense Disambiguation, WSD)를 통해 이루어지며 이를 위해서 워드넷 기반의 어휘 중의성 해소 알고리즘을 도입하였다.

3.2.1 워드넷 기반 어휘 중의성 해소

구문 트리를 구성하는 어휘들의 개념을 정확



〈그림 3〉 동일한 의미에 대한 상이한 표현에 따른 구문 트리 커널의 한계점

하게 식별하기 위해서 어휘 중의성 해소 단계는 매우 중요하다. 특정 단어가 그 의미에 따라 워드넷 내에서 복수 개의 synset에 사상될 수 있으므로, 현재 문맥에서 그 단어의 정확한 의미를 나타내는 synset을 선택하는 과정이 필수적이다. 대상 문헌이 생명공학 분야 논문이므로 이 분야에 해당하는 전문용어들은 중의성 해소가 필요없을 정도로 그 뜻이 명확하지만, 그 용어들을 아우르는 일반 어휘들에 대한 의미적 중의성 해소는 매우 중요하다.

Lesk(1986)는 사전 정의문을 이용한 어휘 중의성 해소 알고리즘을 최초로 제안하였다. 이 연구에서 그는 서로 이웃해 있는 문맥 단어들은 그들의 사전 정의문 측면에서 서로 공통된 단어들을 공유하는 현상을 보여주었으며, 이 가설에 근거하여 어휘 중의성 해소 알고리즘을 고안하였다. 이를 기반으로 Banerjee and Pedersen(2002)은 Lesk 알고리즘을 확장하여 사전 정의문 대신 워드넷(WordNet)을 활용한 알고리즘을 고안했으며, 워드넷의 구성요소인 synset, gloss 및 계층 정보가 어휘 중의성 해소의 성능 향상에 기여할 수 있음을 밝혔다.

다음 알고리즘에서 어휘 중의성 해소 함수의 입력은 대상 단어, 대상 단어에 대한 품사, 주변 문맥, 그리고 추상화 수준이다. 우선 처리 대상 단어를 워드넷에서 검색하면 다수의 synset 집합이 검색된다(line 7). 이는 대상 단어가 워드넷에서 여러 의미로 표현되어 있다는 것이고, 이

〈표 1〉 WordNet을 활용한 문맥 기반 어휘 중의성 해소 알고리즘

```
1   FUNCTION word_sense_disambiguation(word, POS, context, level)
2   word = target word to be disambiguated;
3   POS = Part-Of-Speech of the word;
4   context = neighboring words of word;
5   level = synset level to be considered in extracting synset words;
6   BEGIN
7       synsets = search_word_in_WordNet(word, POS);
8       IF (synsets IS EMPTY) THEN
9           RETURN NULL;
10      max_dups = 0;
11      max_synset = NULL;
12      FOR EACH synset IN synsets retrieved
13      BEGIN
14          sw = get_synset_words(synset, level);
15          dups = get_duplication_count(sw, context);
16          IF max_dups < dups THEN
17              max_dups = dups;
18              max_synset = synset;
19          END IF
20      END FOR
21
22      RETURN max_synset;
23  END
```

들 중에서 대상 단어의 문맥 정보와 가장 일치하는 synset을 선택하는 과정을 수행해야 한다. 이는 (line 12)에서 (line 20)까지의 반복문에서 이루어진다. 우선 특정 synset을 구성하는 구성 어휘를 수집한다(line 14). get_synset_words 함수는 입력 synset을 표현하거나 정의하고 있는 모든 구성 어휘를 반환한다. 여기에는 동의어 집합(synonym set)과 정의문(gloss)도 포함된다. 이 함수의 마지막 인자인 level은 synset 구성 어휘의 반환 범위를 지정한다. 만일 level이 0이면 현재 synset의 구성 어휘만을, level이 1이면 현재 synset의 구성 어휘에 그 부모 synset의 구성 어휘까지도 함께 반환하게 된다. get_duplication_count 함수는 대상 단어의 문맥 정보와 synset 구성 어휘간의 공통 어휘를 계산하는 기능을 수행한다. 가장 많은 공통 어휘를 가지는 synset이 최종 출력으로 반환된다. 결론적으로 본 논문에서 구현한 어휘 중의성 해소 알고리즘의 최종 출력 값은 워드넷의 특정 synset이 된다.

3.2.2 시맨틱 구문 트리 커널

두 문장의 유사도를 계산하는 시맨틱 구문 트리 커널은 다음과 같이 표현될 수 있다.

$$K_{sem}(T_1, T_2, \lambda, \sigma, \alpha) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta_{sem}(n_1, n_2, \lambda, \sigma, \alpha)$$

(식 1)

여기서 T_1, T_2는 두 개의 입력 구문 트리를 나타낸다. 이와 더불어 정밀한 커널 계산을 위해 3가지 매개변수가 수반되는데, 이에 대한 내용은 〈표 2〉에 정리되어 있다.

〈표 2〉에서 보는 바와 같이, 트리 커널 소멸 인자와 커널 계산 방법 선별 인자는 시맨틱 구문 트리 커널의 구문적 유사도 계산에 관여하고, 어휘 개념 추상화 수준 지정 인자는 어휘 의미적 유사도 계산에 관여하는 매개 변수이다. 마지막으로, Δ_{sem}은 특정 노드 n_1과 n_2를 최상위 노드로 가지는 트리의 공통 하부 트리 개수를 계산하며, 상세한 알고리즘은 〈표 3〉에서 나타내었다.

〈표 2〉 시맨틱 구문 트리 커널의 입력 매개 변수

매개변수	변수	설명
트리 커널 소멸 인자	λ	• 비교 대상이 되는 구문 트리들의 깊이(tree depth)가 서로 상이함에 따라 발생하는 커널 값의 불일치성을 해결하기 위해서 도입 • 두 개의 구문 트리가 비교되는 과정에서 단말 노드에 가까워짐에 따라 노드별 일치 여부가 커널 값에 대하여 기여하는 정도가 작아질 수 있도록 값을 지정
트리 커널 계산 방법 지정 인자	σ	• 유사도 측정을 위한 구문 트리 분리 방법 지정 • 부분트리(SubTree, ST) • 부분집합트리(SubSet Tree, SST)
어휘개념 추상화 수준 지정 인자	α	• 구문 트리의 구성 어휘에 대한 개념 생성 시에 워드넷에서의 추상화 수준 지정 • 0 : WSD에 의한 출력 synset 그대로 사용 • 1 : 출력 synset의 부모 synset 사용 • 2 : 출력 synset의 조부모 synset 사용

〈표 3〉 $\Delta_{sem}(n1, n2, \lambda, \sigma, \alpha)$ 계산 알고리즘

```
1   FUNCTION Semantic_Delta(TreeNode n1, TreeNode n2, λ, σ, α)
2
3   BEGIN
4       IF n1 and n2 are both terminal nodes THEN
5           concept1 = get_semantic_concpet(n1, α);
6           concept2 = get_semantic_concept(n2, α);
7           IF concept1 == concept2 THEN
8               RETURN 1;
9           RETURN 0;
10      END IF
11
12      IF n1 and n2 are from different productions THEN
13          RETURN 0;
14      END IF
15
16      IF the productions of n1 and n2 are the same THEN
17          IF n1 and n2 are pre-terminal nodes THEN
18              RETURN λ;
19
20          RETURN λ∏_{j=1}^{nc(n1)}(σ + Semantic_Delta(ch_{n1}^j, ch_{n2}^j, λ, σ, α))
21
22      END IF
23
24  END
```

위 알고리즘에서 가장 핵심적인 부분은 어휘 개념 생성 부분이다(line 5, 6). 이를 위해서 get_semantic_concept 함수는 〈표 1〉에서 제시한 워드넷 기반 어휘 의미 중의성 해소 모듈을 이용하여, 현재 단어에 가장 적정한 synset을 반환하며, 동시에 α 값에 따라 추상화 작업을 수행한다. 어휘 개념 생성 절차(get_semantic_concept 함수)를 세부적으로 살펴보면 다음과 같다.

① 현재 단말 노드 값(단어) 중심으로 각각 m개의 좌/우 단말 노드 값(문맥 어휘 집합)의 집합을 추출
② 현재 노드의 품사 정보(부모 노드 값)를 추출
③ 〈표 1〉의 word_sense_disambiguation (word, context, POS, level)을 호출하여 연관성이 가장 높은 워드넷 synset을 결정
④ 결정된 synset에 대해서 a 만큼 부모 synset으로 이동
⑤ 이동 완료된 synset offset을 어휘 개념으로 반환

이는 구문 트리에서 어휘를 나타내는 단말 노드에 국한되어 실행되며 구문 트리 소멸 인자와 상관없이 두 개념이 동일하면 1을 아니면 0을 반환하게 된다. 〈표 3〉에서 제시된 알고리즘의 나머지 부분은 Moschitti(2006)가 제안한 내용과 동일하다.

3.3 시맨틱 구문 트리 커널에 대한 분석적 접근

앞에서도 지적하였듯이, 시맨틱 구문 트리 커널은 두 문장의 구문적 유사성과 어휘 의미적 유사성을 동시에 측정하여 결합함으로써 계산된다. 따라서 (식 1)에서 제시한 커널 함수는 다음과 같이 표현될 수 있다.

$$K_{sem}(T_1,T_2,\lambda,\sigma,\alpha)=\sum_{n_1\in N_{T_1}}\sum_{n_2\in N_{T_2}}\Delta_{sem}(n_1,n_2,\lambda,\sigma,\alpha)$$
$$=sim_{lex}(T_1,T_2,\alpha)+sim_{syn}(T_1,T_2,\lambda,\sigma)$$

(식 2)

(식 1)을 기반으로 (식 2)는 시맨틱 구문 트리 커널이 (1) 어휘 의미적 유사도(lexical semantic similarity, $sim_{lex}(T_1,T_2,\alpha)$)와 (2) 구문적 유사도(syntactic similarity, $sim_{syn}(T_1,T_2,\lambda,\sigma)$)를 합산하여 커널 값을 계산하고 있음을 나타낸다. 어휘 의미적 유사도는 구문 트리의 단말 노드만을 대상으로 계산되며, 구문적 유사도 계산은 그 외의 노드를 대상으로 이루어진다. 우선 본 논문에서 제안하는 어휘 의미적 유사도에 대한 세부적인 계산 방법은 다음 (식 3)과 같다.

$$sim_{lex}(T_1,T_2,\alpha)\equiv sim(W_{T_1},W_{T_2},\alpha)$$

(식 3-1)

$$\approx\sum_{w_1\in W_{T_1},c_1\subset W_{T_1}^{w_1}}\left(\sum_{w_2\in W_{T_2},c_2\subset W_{T_2}^{w_2}}sim(w_1,w_2,c_1,c_2,\alpha)\right)$$

(식 3-2)

$$\approx\sum_{w_1\in W_{T_1},c_1\subset W_{T_1}^{w_1}}\left(\sum_{w_2\in W_{T_2},c_2\subset W_{T_2}^{w_2}}I(concept(w_1,c_1,\alpha),concept(w_2,c_2,\alpha))\right)$$

(식 3-3)

$$\approx\sum_{w_1\in W_{T_1},c_1\subset W_{T_1}^{w_1}}\left(\sum_{w_2\in W_{T_2},c_2\subset W_{T_2}^{w_2}}I(synset(w_1,c_1,\alpha),synset(w_2,c_2,\alpha))\right)$$

(식 3-4)

$$=\sum_{w_1\in W_{T_1},c_1\subset W_{T_1}^{w_1}}\left(\sum_{w_2\in W_{T_2},c_2\subset W_{T_2}^{w_2}}I(pos(synset(w_1,c_1,\alpha)),pos(synset(w_2,c_2,\alpha)))\right)$$

(식 3-5)

W_T는 구문 트리 T를 구성하는 단어 집합이며, $W_T^{w_i}$는 이 중에서 w_i 주변의 문맥 단어 집합이다. 본 논문에서는 앞에서 제시한 어휘 의미적 유사도를 다음과 같이 정의한다.

(정의 1) 두 문장의 어휘 의미적 유사도(lexical semantic similarity)는 이들 문장을 구성하는 모든 단어들에 대한 개념적 교차 비교(conceptual cross-comparison)를 통해서 도출되는 공통 어휘 개념(lexical concept)들의 개수이다.

(정의 1)은 (식 3-1)과 (식 3-2)에서 명확하게 표현된다. (식 3-2)에서 두 문장을 구성하는 모든 단어들에 대해서 문맥(c_1, c_2) 기반의 단어 유사도가 누산된다. 단어 간 유사도를 수치화하여 정확하게 계산하기는 어려우므로, (식 3-3)에서는 개별 단어를 개념화함으로써 두 단어의 유사도 계산을 단순화하였다. 이 식에서 $I(A, B)$는 A와 B가 동일하면 1을 반환하

고, 아니면 0을 반환한다. 결론적으로 이 식은 두 문장을 교차 비교하면서 동일한 개념을 나타내는 단어 쌍의 개수를 계산한다. 이를 위해서 본 논문에서는 특정 단어의 개념을 다음과 같이 정의하였다.

(정의 2) 특정 어휘의 의미적 개념(lexical concept)은 그 단어에 대한 워드넷(WordNet)에서의 synset 집합 중에서, 주변 문맥 단어들과 가장 일치하는 synset이다. 따라서 동일한 단어도 문맥에 따라서 다른 의미적 개념을 나타낼 수 있다.

(정의 2)를 기반으로 (식 3-4)에서는 앞의 〈표 1〉에서 제안한 어휘 중의성 해소 알고리즘을 통해서 도출된 개별 어휘들의 synset들이 비교된다. 워드넷에서 특정 synset은 전체 파일 내에서의 위치 정보로 식별될 수 있으므로, 이들 정보를 기반으로 실질적으로 비교 작업을 수행하였다. 마지막 식에서 *pos(S)*는 synset S의 파일 옵셋 정보를 반환한다. 모든 식에서 포함된 추상화 수준 지정 인자 α는 개별 어휘의 의미적 개념들의 추상화 수준을 결정하게 된다. 한편, 두 문장에 대한 구문적 유사도를 나타내는 $sim_{syn}(T_1, T_2, \lambda, \sigma)$은 Moschitti(2006)가 제안한 것과 동일하게 다음과 같이 표현될 수 있다.

$$sim_{syn}(T_1, T_2, \lambda, \sigma) \equiv \sum_{n_1 \in N_{T_1}, n_1 \notin L_{T_1}} \left(\sum_{n_2 \in N_{T_2}, n_2 \notin L_{T_2}} \Delta(n_1, n_2, \lambda, \sigma) \right)$$

(식 4)

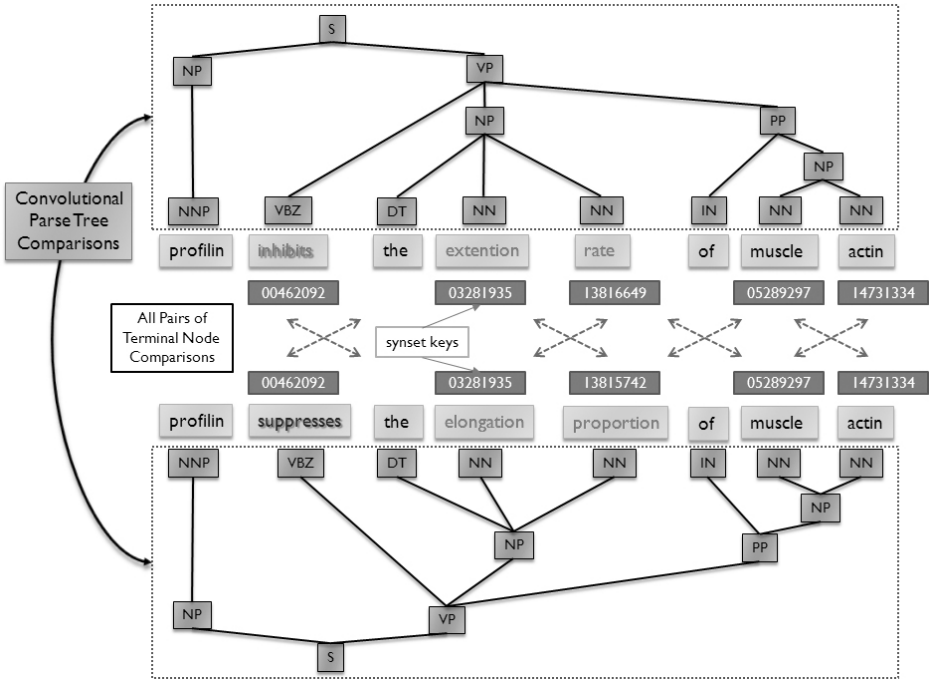
N_{T_i}는 구문 트리 T_i의 전체 노드 집합이고, L_{T_i}는 T_i의 모든 단말 노드 집합을 나타낸다. 그리고 $\Delta(n_1, n_2, \lambda, \sigma)$은 Moschitti(2006)가 제안한 특정 노드 n_1과 n_2를 최상위 노드로 가지는 트리의 공통 하부 트리 개수 계산 알고리즘이다. (식 4)에 대한 자세한 내용은 Collins and Duffy (2001)와 Moschitti(2006)에서 상세히 기술하고 있다. (식 3)과 (식 4)에 의거하여, 시맨틱 구문 트리 커널은 다음과 같이 표현될 수 있다.

$$K_{sem}(T_1, T_2, \lambda, \sigma, \alpha) \equiv \sum_{w_1 \in W_{T_1}, c_1 \subset W_{T_1}^{w_1}} \left(\sum_{w_2 \in W_{T_2}, c_2 \subset W_{T_2}^{w_2}} I(pos(synset(w_1, c_1, \alpha)), pos(synset(w_2, c_2, \alpha))) \right) + \sum_{n_1 \in N_{T_1}, n_1 \notin L_{T_1}} \left(\sum_{n_2 \in N_{T_2}, n_2 \notin L_{T_2}} \Delta(n_1, n_2, \lambda, \sigma) \right)$$

(식 5)

(식 5)는 시맨틱 구문 트리 커널의 세부적인 계산 방법을 나타낸다. 식에서도 알 수 있듯이 우변의 첫째 항은 어휘 의미적 유사도를, 둘째 항은 구문적 유사도를 나타내며, 최종 커널 값은 이 두 유사도를 더한 값이다.

〈그림 4〉는 두 문장에 대한 시맨틱 구문 트리 커널 수치를 계산하는 형태를 도식화한 것이다. 그림의 중앙 부근에서는 단말 노드 즉 구성 단어들에 대한 어휘 의미적 유사도를 계산하고 상단과 하단의 두 구문 구조를 기반으로 구문적 유사도를 계산하게 된다. 이 때, 중앙의 개별 단어들은 워드넷 기반 어휘 중의성 해소 알고리즘이 적용되어 개념화된다. 단말 노드의 상/하단에 나타나는 수치는 생성된 어휘 개념의 식별자 즉 워드넷에서의 파일 옵셋이다. "inhibit"와 "suppress"는 동일한 어휘 개념으로 사상되어서 개념 식별자가 같으나, "rate"와 "proportion"은 다른 개념으로 인식되어서 서로 다른 개념 식별자가 지정되었다.



〈그림 4〉 시맨틱 구문 트리 커널 계산 방법 도식화

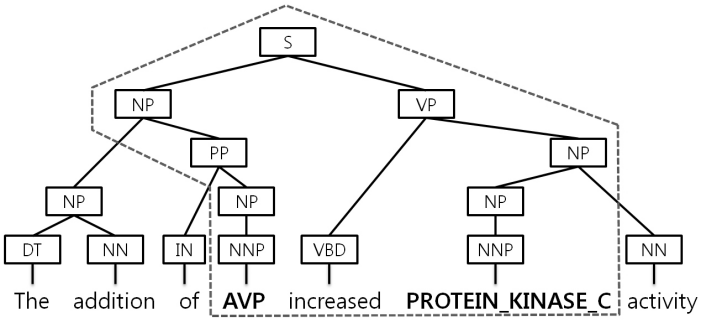
4. 시스템 구성

이 장에서는 본 논문에서 구성한 시스템을 세부적으로 설명하고 그 특징 및 확장성에 대해서 다룬다. 아키텍처에 대한 세부 설명은 다음과 같다. 우선 본 논문의 시스템에 포함된 언어 분석기는 구문분석기, 기저구 인식기(CRF Chunker), 품사 태거(CRF POS-tagger) 등이다. 구문분석기는 Charniak Parser[8]를 도입하여 시스템에 이식시켰다. 또한 논문에서는 활용되지 않았으나 부가적으로 장래에 다양한 형태의 언어자질 추출을 위해서 기저구 인식기 및 품사 태거도 독립적으로 개발하여 시스템에 결합하였다.

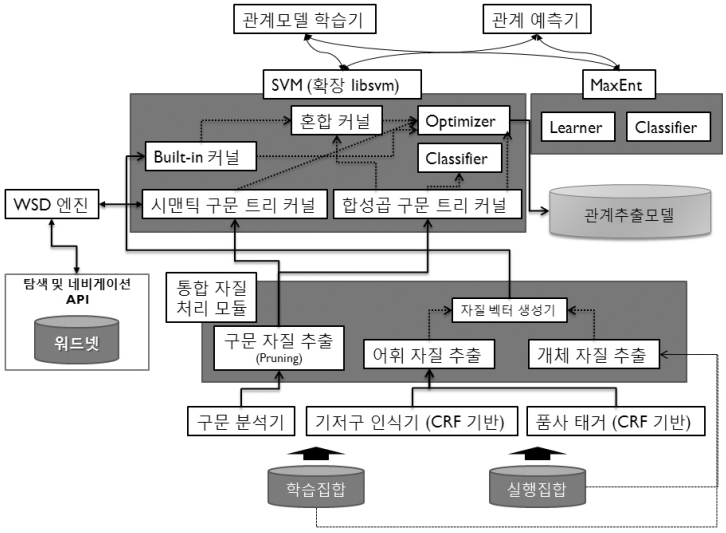
문장에 대한 구문 분석 결과는 구문 자질 추출기로 입력되며 여기서 미리 표시된 단백질의 문장 내 위치와 주변 구문 정보를 이용하여 가지치기(parse tree pruning)를 수행한다.

〈그림 5〉에서 보는 바와 같이 본 논문에서 기본적으로 제공되는 가지치기 기법은 Zhang et al.(2006)이 고안한 다양한 방법 중에서 가장 성능이 좋은 것으로 평가받고 있는 경로 포함 트리(Path-enclosed tree pruning) 기법을 채택하였다. 이 방법의 특징은 두 단백질 사이에 존재하는 상호작용을 표현하는 어휘자질과 이를 아우르는 구문자질을 동시에 집중적으로

8) 〈http://www.cs.brown.edu/people/ec/#software〉.



〈그림 5〉 경로포함 트리(Path-enclosed Tree) 가지치기 예



〈그림 6〉 시스템 구성도

적용할 수 있다는 것이다.

어휘 자질 추출기는 문장에 대한 품사 태깅이나 기저구 인식을 통해서 생성되는 품사정보 및 기저구 정보와 함께 문장 내에 발생한 단어 집합을 이용한 일반 자질 벡터를 구성하는데 사용된다. 개체 자질 추출기는 단백질의 고유한 특성 정보가 제공되면 이를 자질화하여 관계추출에 적용하기 위한 모듈이다.

본 논문에서 개발된 시스템은 두 가지 기계학습 모델을 기반으로 구축되었다(〈그림 6〉 참조). 우선 SVM 기반 관계추출을 위해서는 *libsvm* 2.89[9]를 자체적으로 확장하여 여기에 구문 트리 커널을 이식시켰다. 결과적으로 *libsvm* 자체적으로 제공하는 네 가지 기본 커널(선형, 다항, RBF,

9) 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm/〉.

Sigmoid)에 구문트리커널과 혼합커널(기본 커널과 구문트리커널을 결합시킨 커널)을 추가적으로 제공하도록 하였다. 본 논문에서 제안된 시맨틱 구문 트리 커널은 워드넷 기반의 어휘 중의성 해소 시스템을 수반하므로 이와 밀접하게 연결되어 있다. 어휘 중의성 해소 알고리즘에서 빈번하게 활용되는 워드넷 접근 API는 워드넷 배포판에서 기본적으로 제공하고 있는 엔진을 변형/확장하여 구성하였다. 최대 엔트로피 모델을 활용하기 위해서 *Maximum Entropy Modeling Toolkit for Python and C++*[10]을 이용하였다.

각각의 기능별 구성 모듈은 모두 소스 수준에서 통합되어 모듈화된 하나의 패키지 형태로 개발되었다. 따라서 설정 지정 및 모듈 추가/변경을 통해서 다양한 응용분야에 적용될 수 있다.

5. 실험 및 분석

이 장에서는 본 논문에서 제안한 시맨틱 구문 트리 커널의 성능을 파악하기 위해서 다양한 말뭉치 기반의 실험 결과를 제시한다. 특히 기존의 연구와는 달리, 단백질 상호 작용 식별(PPII)뿐만 아니라 다중 분류체계기반의 단백질 상호작용 분류(PPIC) 실험도 수행하였다. 시맨틱 구문 트리 커널의 객관적인 성능 수준을 알아보기 위해서, 기존의 대표적인 연구 결과에서 사용된 말뭉치를 활용하였으며, PPIC 실험을 위해서는 약 25 종류의 연관관계가 포함된 BioInfer 말뭉치를 이용하였다.

5.1 실험 대상 말뭉치

본 절에서는 논문에서 사용한 실험 말뭉치를 소개한다. 말뭉치는 크게 단백질 상호작용 식별 성능 측정용 말뭉치와 분류 성능 측정용 말뭉치로 나뉜다. 아래 절에서 본 논문의 실험에서 사용한 두 종류의 말뭉치에 대해서 소개한다.

5.1.1 단백질 상호작용 식별 말뭉치

단백질 상호작용 식별 실험은 Pyysalo et al. (2008)이 구성한 5가지의 PPI 말뭉치를 대상으로 수행하였다. 통상적으로 "*Five PPI Corpora*[11]"라고 불리는 이 말뭉치 집합은 AIMed, BioInfer, HPRD50, IEPA 그리고 LLL을 단일화된 XML 형식으로 변환해 놓은 컬렉션으로서, 현재 단백질 간 상호작용 추출 기법의 준거 평가 컬렉션으로 활용되고 있다(Bunescu et al. 2005; Pyysalo et al. 2007; Fundel, Kuffner, and Zimmer 2007; Ding et al. 2002; Nedellec 2005) (〈표 4〉 참조).

〈그림 1〉과 〈그림 2〉에서 보는 바와 같이, 특정 문장에 2개 이상의 단백질 이름이 출현하고, 그것들 간의 상호작용 관계가 설정되어 있으면 단일 문장에 대해서도 여러 개의 상호작용 포함 문장이 구성된다. 또한 문장 내에 단백질 이름이 존재하더라도 상호작용 관계가 설정되어 있지 않다면 상호작용 포함 문장도 불포함 문장으로 동시에 설정될 수 있다. 이를 기반으로

10) 〈http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html〉.
11) 〈http://mars.cs.utu.fi/PPICorpora/eval-standard.html〉.

〈표 4〉 Five PPI Corpora 규모 및 내용

말뭉치	AIMed	BioInfer	HPRD50	IEPA	LLL
문장 개수	1,955	1,100	145	486	77
단백질 간 상호작용 포함 문장 (Positive instance)	1,000	2,534	163	335	164
단백질 간 상호작용 불포함 문장 (Negative instance)	4,834	7,132	270	482	166

단백질 간 상호작용 추출은 개별 인스턴스(상호작용 포함/불포함 문장)에 대한 이진 분류 작업으로 규정할 수 있다.

5.1.2 단백질 상호작용 분류 말뭉치

단백질 상호작용 분류 실험을 위해서 본 논문에서는 Pyysalo et al.(2007)이 개발한 BioInfer[12] 말뭉치를 활용하였다. BioInfer는 생의학 분야 정보 추출을 위한 통합 말뭉치로서 이 분야 연구 논문의 초록에서 추출한 1,100개의 문장에 대해서 단일 단백질(individual protein), 복합 단백질(protein complex), 단백질 종(protein family or group), 기능 특성(function property) 등을 수동으로 식별하여 25가지의 관계 종류로 이들 간의 연관관계를 지정해 놓은 집합이다. 본 논문에서는 위의 여러 가지 개체들 중에서 단일 단백질(individual protein)에 대해서만 실험을 수행하였다. 〈표 5〉는 실험에 사용한 BioInfer 말뭉치의 상호작용 종류와 각 관계별 인스턴스 개수를 보여준다.

〈표 5〉에서 보듯이 논문에서 사용한 BioInfer 부분 집합 내에서의 개별 상호작용은 계층적으로 구성되어 있으며, 최상위 상호작용의 종류는 총 6가지이다.[13] 말단 상호작용 기준으로 "Assembly"가 가장 많은 788개의 인스턴스를 가지고 있고, "Full-Stop"이 겨우 1개의 인스턴스를 가지고 있다. 전체적으로 볼 때, 인스턴스의 편중현상이 심하고 고르지 않은 분포를 보이고 있다. 따라서 자동 분류 시에 자료 희귀성(data sparseness) 문제가 발생할 가능성이 매우 높다.

Pyysalo et al.(2007)이 제시한 것처럼 BioInfer의 관계 온톨로지(relationship ontology)를 구성하는 단말 노드들은 관계 술어(relation predicate)를 포함하고 있다. 이는 특정 관계를 표현하는 다양한 형태의 동사 혹은 동사구를 나타낸 것이다.[14] 따라서 만일 이 관계 온톨로지를 하나의 트리 형태로 본다면, 이 트리는 위 〈표 5〉에서 보인 관계 유형 클래스(relation type class)를 나타내는 노드와 관계 술어 노드로 구분될 수 있다. 〈표 6〉에서 이 두 종류의 노드들이 모두 포함된 BioInfer의 관계 온톨로지

12) 〈http://mars.cs.utu.fi/BioInfer〉.

13) BioInfer에서 정의한 관계(상호작용) 온톨로지는 "NOT", "REL-ENT"를 포함하여 총 8가지의 관계를 정의하고 있으나, 본 논문에서 분석한 결과 단백질 간의 상호작용에서는 두 관계가 나타나지 않았다.

14) 예를 들어, 〈표 5〉에서 "Negative" 관계는 "DOWNREGULATE, INHIBIT, SUPRESS"의 세 가지의 관계 술어를 포함하고 있으며, "Full-Stop" 관계는 "HALT, INACTIVATE" 술어를 포함한다. 자세한 내용은 [24]에서 찾아볼 수 있다.

〈표 5〉 BioInfer 말뭉치내의 단백질 상호작용 종류 및 인스턴스 개수

<table>
<tr><td colspan="6">Relations and #Instances</td></tr>
<tr><td rowspan="14">Causal(1,658)</td><td colspan="5">56</td></tr>
<tr><td rowspan="13">Change(1,599)</td><td colspan="4">205</td></tr>
<tr><td>Amount</td><td colspan="3">39</td></tr>
<tr><td rowspan="6">Dynamics (190)</td><td colspan="3">12</td></tr>
<tr><td>Full-Stop</td><td colspan="2">1</td></tr>
<tr><td>Negative</td><td colspan="2">48</td></tr>
<tr><td>Positive</td><td colspan="2">80</td></tr>
<tr><td>Start</td><td colspan="2">7</td></tr>
<tr><td>Unspecified</td><td colspan="2">42</td></tr>
<tr><td>Location</td><td colspan="3">255</td></tr>
<tr><td rowspan="5">Physical (910)</td><td colspan="3">8</td></tr>
<tr><td>Assembly</td><td colspan="2">788</td></tr>
<tr><td>Break-Down</td><td colspan="2">14</td></tr>
<tr><td rowspan="2">Modification (100)</td><td colspan="2">44</td></tr>
<tr><td>Addition</td><td>56</td></tr>
<tr><td>Condition(3)</td><td colspan="4">3</td></tr>
<tr><td>HUMANMADE(4)</td><td colspan="5">4</td></tr>
<tr><td rowspan="2">IS_A(125)</td><td colspan="5">95</td></tr>
<tr><td>Equality(30)</td><td colspan="4">30</td></tr>
<tr><td rowspan="3">Observation(55)</td><td colspan="5">27</td></tr>
<tr><td>Spatial</td><td colspan="4">7</td></tr>
<tr><td>Temporal</td><td colspan="4">21</td></tr>
<tr><td rowspan="2">PART_OF(318)</td><td>Collection:Member</td><td colspan="4">256</td></tr>
<tr><td>Object:Component</td><td colspan="4">62</td></tr>
<tr><td>RELATE(87)</td><td colspan="5">87</td></tr>
</table>

〈표 6〉 BioInfer Relationship Ontology에서의 수준별 노드 개수

Level	1	2	3	4	5	6	#Total
# Relation Type Classes	4	8	6	8	2	0	28
# Relation Predicates	4	6	20	7	26	5	68
# Total	8	14	26	15	28	5	96

에 대해서 수준별로 각각의 노드 개수를 나타내었다.

〈표 6〉에서 알 수 있듯이 BioInfer의 관계 온톨로지는 총 6개의 수준에 96개의 노드가 존재하며, 이중 관계 유형 클래스는 총 28개이고, 관계 술어는 68개이다. 실험에서는 〈표 5〉에서 나타낸 관계 집합을 기준으로, 전체 25가지의 관계 유형 클래스와 최상위 수준부터 시작하여 3수준까지의 관계 집합으로 구성된 3가지 말뭉치를 포함하여 도합 4가지 말뭉치를 이용하여

실험을 수행하였다. 비록 〈표 5〉에서는 나타내지 않았으나 실험에서는 관계 술어도 포함된 관계 집합을 사용하였다. 〈표 7〉은 이들 각각의 관계집합 및 개수를 보여준다.

관계 종류의 규모에 따른 성능 변화를 측정하기 위해서 본 논문에서는 위와 같이 서로 다른 정교성(granularity)을 가지는 관계 집합 4가지를 사용하였다. 이에 따른 성능 측정 결과는 5.5에서 살펴본다.

5.2 실험 대상 시스템

본 논문에서 수행한 실험은 총 세 가지의 매개변수가 수반되며 이 값의 지정 방법에 따라서 성능이 차이가 날 수 있다. 세부적인 내용은 〈표 8〉과 같다.

구문 트리 커널 소멸인자는 최소 0.1에서 최대 1.0까지 0.1 단위로 10개를 지정하였다. SVM의 정규화 인자는 1.0에서 7.0까지 7가지로 한정하였다. 마지막으로 〈표 2〉에서 설명한 어휘 개념 추상화 수준 지정 인자는 총 4가지로 설정하였다. 따라서 이들 세 가지 매개변수를 모두 적용하면 총 280가지의 설정이 도출된다. 본 논문에서는 이 설정 집합 각각에 대해서 10겹 교차평가를 수행하여 성능을 측정한다.

5.3 성능 측정 기준

본 논문에서 사용한 성능 측정 기준은 거시 평균 기반 F-스코어(macro-averaged F-score)와 미시 평균 기반 F-스코어(micro-averaged F-score)이다. 우선 거시 평균 기반 방법은 m개의 클래스에 대해서 개별적으로 정확도와 재현율이 합산된 F-스코어를 계산하고, 이를 m으로 나눈 평균을 계산하는 방법이다. 이에 반해 미시 평균 기반 방법은 전체 검증 데이터를 기

〈표 7〉 실험에 사용된 관계집합 종류 및 개수

관계 집합	설명	관계 수
RCP_L1	최상위 수준의 관계 집합	6
RCP_L2	두 번째 수준까지의 관계 집합	12
RCP_L3	세 번째 수준까지의 관계 집합	22
RCO	관계 클래스만으로 구성된 관계집합	25

〈표 8〉 실험 대상 시스템 설정 종류 및 개수

매개변수	설명(details)	범위(range)		설정개수
λ	구문트리커널 소멸인자	0.1 ~ 1.0(단위: 0.1)		10
C	SVM 정규화 매개변수	1.0 ~ 7.0(단위: 1.0)		7
a	시맨틱 구문트리 커널에서의 어휘 개념에 대한 추상화 수준 지정 인자 (generalization level)	0	Node concept 그대로 사용	4
		1	현재 node concept의 부모를 사용	
		2	현재 node concept의 조부모를 사용	
		N	기존 구문 트리 커널	
총 시스템 수				280

반으로 옳게 분류된 데이터와 그르게 분류된 데이터를 누산하고 이를 기반으로 *F*-스코어를 계산하는 방법이다. 전자는 학습 모델의 모든 클래스에 대한 분류 능력을 전체적으로 살펴볼 수 있는 장점이 있으나, 학습 집합의 클래스별 분포가 고르지 않을 경우 상대적으로 낮은 성능측정 결과를 가져온다. 미시 평균 기반 방법은 학습 모델의 특정 클래스에 대한 분류 능력이 상대적으로 낮을 경우, 이를 제대로 반영하지 못한다는 단점이 있다. 학습 집합의 클래스별 분포가 차이가 나는 경우나, 학습 모델의 특정 클래스 예측 성능이 낮게 나타날 경우에는 두 평가 방법의 수치 차이가 상당한 경우도 있다.

5.4 단백질 상호작용 식별(PPII) 실험

본 절에서는 단백질 간 상호작용 식별 실험의 결과를 보이고 이를 분석한다.

〈표 9〉는 총 280개의 시스템 중에서 최고 성능을 보이는 설정 및 상세 성능 수치를 5개의 말뭉치별로 보여준다. 총 5개의 말뭉치를 대상으로 한 실험에서 IEPA를 제외한 나머지 말뭉치에서 시맨틱 구문 트리 커널(SPTK)이 가장 높은 성능을 나타내고 있다. HPRD50은 일반 구문 트리 커널(PTK)과 시맨틱 구문 트리 커널의 성능이 동일하였다. 어휘 개념 추상화 수준은 0 혹은 1에서 가장 높은 성능을 나타내었고, 2일 때는 오히려 일반 구문 트리 커널보다 성능이 떨어지는 현상을 보여주었다. IEPA를 제외하고는 거시평균기반 F-점수 기준으로 대부분 80.0 이상을 나타내고 있다.

〈표 10〉은 거시평균기반 F-점수 기준으로 상위 20등까지의 시스템에 대해서 구문 트리 커널과 시맨틱 구문 트리 커널의 분포를 분석한 자료이다. IEPA를 제외한 나머지 자료에서 대부분 시맨틱 구문 트리 커널의 출현 횟수가 많았으며, BioInfer 말뭉치에 대해서는 20개의 시스템 중에서 총 17개가 시맨틱 구문 트리 커널이었다. 따라서 시맨틱 구문 트리 커널 기반 기법이 전반적으로 높은 성능을 유지하고 있음을 알 수 있다.

마지막으로 〈표 11〉은 본 논문에서 제안한 시스템의 성능과 기존 연구 결과와의 비교 분석 자료이다. 동일한 조건에서의 비교를 위해

〈표 9〉 각 말뭉치별 최고 성능을 나타내는 설정 값 및 성능 세부 정보

Collection	Tree Kernels	Abstraction Level	DF(λ)	Regularization Factor(C)	mi-F1	Precision	Recall	ma-F1
AImed	SPTK	1	0.5	7.0	**89.33**	84.86	77.45	80.99
BioInfer	SPTK	0	0.5	5.0	89.00	87.22	**84.81**	86.00
IEPA	PTK	-	0.4	7.0	79.17	78.51	78.30	78.41
HPRD50	SPTK/PTK	0/1/2	0.7	6.0	85.22	84.74	83.41	84.07
LLL	SPTK	1	0.4	4.0	88.48	**88.64**	88.47	**88.55**

ma-F1: macro-averaged F1
mi-F1: micro-averaged F1
SPTK: Semantic Parse Tree Kernel
PTK: Parse Tree Kernel

〈표 10〉 ma-F1 기준 상위 20개 시스템에서의 설정 분포

Collections	PTK	SPTK			SPTK(total)	Coverage rate
		α = 0	α = 1	α = 2		
AImed	7	4	5	4	13	65%
BioInfer	3	7	5	5	17	**85%**
IEPA	12	4	3	1	8	40%
HPRD50	6	4	4	6	14	70%
LLL	4	5	5	6	16	80%

서 Airola et al.(2008), Miwa et al.(2009)에서와 동일한 정규화 인자를 적용하였으며, 트리커널 소멸인자는 모두 0.4로 고정시켰다. 표에서 보는 바와 같이 본 논문에서의 최고 성능을 보이는 시스템이 기존 연구의 시스템보다 평균적으로 높게 나타나고 있다. 성능 향상의 원인은 크게 세 가지로 볼 수 있다. 우선 구문트리 가지치기의 적용을 들 수 있다. 앞에서 지적하였듯이 동일한 문장 내에 복수 개의 단백질 이름이 출현하였을 경우 각기 다른 단백질 이름 쌍들에 대한 문장에서의 상호작용 표현 자질은 달라질 수 있다. 예를 들어, "*PROT_A inhibits PROT_B that increases PROT_C's activity*"라는 문장에서 *PROT_A*와 *PROT_B*의 상호작용은 "*inhibits*"라는 동사로 표현되는 반면에 *PROT_B*와 *PROT_C* 사이의 상호작용은 "*increases*"이라는 동사로 나타난다. 만일 구문트리 가지치기를 하지 않았을 경우, 위 두 가지 단백질 쌍에 대한 구문적 자질은 모두 동일하게 적용되며 오히려 *PROT_A*와 *PROT_C*는 직접적인 연관이 없음에도 불구하고 두 단백질이 서로 상호작용을 한다는 판단을 내릴 수 있다. 따라서 구문트리커널을 적용함에 있어서 가지치기는 필수적이며, 선행 연구인 Miwa et al.(2009)에서는 구문트리 커널을 포함한 다양한 커널을 통합 적용했음에도 불구하고 성능 향상이 미비했던 이유도 이 가지치기 미적용 때문이라고 추정할 수 있다. 두 번째로 구문 트리 커널의 소멸인자에 의한 성능 최적화이다. 소멸인자는 기본적으로 구문 트리 커널 함수 내

〈표 11〉 타 시스템과의 성능 비교(동일한 정규화 인자 적용, C = 1.0)

대상 시스템	AImed	BioInfer	HPRD50	IEPA	LLL	평균
Airola et al.(2008) [13]	56.4	61.3	63.4	75.1	76.8	66.60
Miwa et al.(2009) [14]	60.8	68.1	70.9	71.7	80.1	70.32
Our system(PTK, λ = 0.4)	75.4	81.2	77.9	75.1	85.5	79.02
Our system (SPTK, α = 0, λ = 0.4)	75.5	81.4	77.9	75.6	85.2	**79.12**
Our system (SPTK, α = 1, λ = 0.4)	75.2	81.3	77.9	75.1	85.2	78.94
Our system (SPTK, α = 2, λ = 0.4)	74.8	81.2	77.9	75.1	85.5	78.90

에서 평활 값(smoothing value) 역할을 수행한다. 다시 말해서, 구문 트리의 유사도를 단말 노드까지 엄밀하게 계산하기 보다는 변형이 심하고 그에 따른 구문 오류도 많은 하부 트리의 커널 기여도를 일정 수준 통제하는 역할을 수행함으로써 학습된 모델의 포괄성을 증대시킨 결과이다. 마지막으로 시맨틱 구문 트리 커널의 적용이다. 이것의 효과는 3.2절에서 이미 설명하였으므로 생략하기로 한다.

비록 시맨틱 구문 트리 커널을 사용하고 어휘 개념을 추상화시키지 않은 시스템이 가장 좋은 성능을 보였으나 다른 시스템과 비교해서 성능향상이 확연하지 않았다. 이는 이진 분류 작업인 단백질 간 상호작용 식별(PPII)이 시맨틱 구문 트리 커널을 적용하여 두드러지는 성능을 보일 만큼 난이도가 높지 않다는 사실을 보여준다. 특이한 점은 어휘 개념에 대한 추상화 수준을 높일수록 성능은 오히려 감소한다는 것이다. 이는 추상화 수준이 높아질수록 개별 어휘들의 의미적 변별력이 낮아지는 현상과 관련이 있음을 보여준다.

5.5 단백질 상호작용 분류(PPIC) 실험

본 절에서는 앞의 5.1.2에서 소개한 BioInfer 말뭉치를 기반으로 수행한 단백질 간 상호작용 자동 분류에 대한 성능 측정 결과를 보인다. 우선 아래 표에서 5.4에서와 마찬가지로 280개의 시스템 중에서 최고 성능을 나타내는 설정을 제시한다. 여기서는 〈표 7〉에서 나타낸 서로 다른 관계 집합을 대상으로 개별 실험을 수행하였다.

다중 분류 작업인 단백질 간 상호작용 분류 실험에서는 〈표 12〉에서 보는 바와 같이 모든 관계 집합에서 시맨틱 구문 트리 커널의 성능이 가장 높게 나타난다. 전체적으로 소멸인자는 5.4절의 실험에서보다 약간 낮은 수치로 지정되고 있다. 관계 술어를 제외한 전체 25개의 관계집합(RCO)에 대한 성능은 거시평균기반 F 점수로 65.4이다. 표에서 보는 바와 같이 미시평균기반 F 점수와 거시평균기반 F 점수의 차이는 확연히 나타난다. 특히 6개의 최상위 분류집합으로 구성된 RCP_L1에서의 미시평균기반 F 점수는 91.63으로 매우 높은 수치를 나타내고 있으나 거시평균기반 F 점수는 68.51밖에 되지 않는다. 이는 〈표 5〉에서 지적한 것과 같이 각 구성관계별 인스턴스 편중현상이 그 원인이다.

시맨틱 구문 트리 커널의 성능 개선 정도를 보다 면밀히 살펴보기 위해서 5.4의 〈표 10〉에서 제시한 것과 같이 거시평균기반 F 점수 기준으로 상위 30등 이내의 시스템에 대한 분포를 분석하여 〈표 13〉에 나타내었다.

〈표 12〉 각 관계 집합별 최고 성능을 나타내는 설정 값 및 성능 세부 정보

Relation Set	Tree Kernels	Abstraction Level	DF(λ)	Regularization Factor(C)	mi-F1	Precision	Recall	ma-F1
RCP_L1	SPTK	2	0.3	5	**91.63**	75.05	**63.03**	**68.51**
RCP_L2	SPTK	0	0.2	7	90.52	**76.65**	60.27	67.48
RCP_L3	SPTK	1	0.1	4	78.06	71.86	52.65	60.77
RCO	SPTK	0	0.4	5	78.02	75.46	57.74	65.42

〈표 13〉 ma-F1 기준 상위 30개 시스템에서의 설정 분포

Collections	PTK	SPTK			SPTK(total)	Coverage rate
		α = 0	α = 1	α = 2		
RCP_L1	8	9	7	6	22	73.3%
RCP_L2	9	8	7	6	21	70.0%
RCP_L3	8	9	7	6	22	73.3%
RCO	5	9	9	7	25	83.3%

모든 관계 집합에 대해서 시맨틱 구문 트리 커널이 높은 점유율을 보이고 있다. 특히 RCO 집합에 대해서는 30개의 상위 시스템 중에서 25개가 시맨틱 구문 트리 커널로 83% 이상의 매우 높은 상위 분포 점유율을 나타내고 있다. 이러한 현상은 나머지 관계 집합이 관계 유형 클래스(relation type class)와 관계 술어(relation predicate)가 혼재된 다소 비정상적인 분류체계임에 반해 RCO는 하나의 완전한 분류체계라는 관점에서 매우 고무적인 현상이다.

마지막으로 일반적인 구문 트리 커널(PTK)과 시맨틱 구문 트리 커널(SPTK)과의 보다 정확한 성능 비교를 위해서 〈표 14〉에서는 PTK의 최고 성능을 나타내는 동일한 설정을 기준으로 두 커널의 성능 비교를 하였다.

〈표 14〉에서 알 수 있듯이, 그 차이는 현저하지 않으나 모든 관계 집합에서 SPTK가 더 나은 성능을 나타내고 있다. 성능 개선의 차이가

〈표 14〉 PTK의 최고성능설정과 동일한 설정에서의 SPTK와의 성능비교

관계집합(설정)	트리커널종류	mi-F1	Precision	Recall	ma-F1
RCP_L1 (λ=0.3, C=7.0)	PTK	**91.94**	**75.68**	62.23	68.30
	SPTK(α=0)	91.90	75.63	**62.37**	**68.36**
	SPTK(α=1)	91.72	75.55	62.29	68.28
	SPTK(α=2)	91.68	75.02	61.72	67.72
RCP_L2 (λ=0.2, C=5.0)	PTK	**89.99**	76.47	**58.91**	66.55
	SPTK(α=0)	89.90	76.43	58.80	66.47
	SPTK(α=1)	89.90	76.61	58.88	**66.58**
	SPTK(α=2)	89.81	**76.66**	58.03	66.06
RCP_L3 (λ=0.1, C=4.0)	PTK	78.02	71.84	51.81	60.20
	SPTK(α=0)	**78.10**	71.65	**52.73**	60.75
	SPTK(α=1)	78.06	**71.86**	52.65	**60.77**
	SPTK(α=2)	77.53	71.29	51.54	59.83
RCO (λ=0.4, C=5.0)	PTK	77.88	74.90	57.05	64.77
	SPTK(α=0)	**78.02**	75.46	**57.74**	**65.42**
	SPTK(α=1)	77.84	**75.50**	57.51	65.29
	SPTK(α=2)	77.44	74.96	57.09	64.82

예상과는 다르게 적게 나는 가장 핵심적인 원인은 앞에서 제시한 워드넷 기반의 어휘 중의성 해소 모듈의 비교적 낮은 성능 때문일 것으로 추측된다. 이는 향후 연구에서 좀 더 깊이 분석할 필요가 있다.

6. 결론 및 향후 연구 방향

본 논문에서는 기존 구문 트리 커널이 두 문장의 구문적 유사도에만 중점을 두어서 커널 수치를 계산하던 것을 확장하여 어휘 의미적 유사도를 동시에 적용한 시맨틱 구문 트리 커널을 고안하였다. 이를 위해서 문맥 및 워드넷 기반의 어휘 중의성 해소 시스템과 이 시스템의 출력으로 도출되는 어휘 개념(WordNet synset)의 추상화를 통해서 기존의 단백질 간 상호작용의 식별과 분류에 동시에 적용하였다. 커널 함수에서 문맥 기반 어휘 중의성 해소의 역할은 다음과 같다. 첫째, 대상이 되는 두 문장들의 구문적 유사성과 어휘 의미적 유사성을 동시에 활용할 수 있다. 둘째, 동형이의어 및 유사어로 인해 부적절한 커널 값이 계산되던 것을 일정 부분 방지하여 보다 정확한 유사도 계산이 가능하다. 마지막으로 워드넷을 이용하여 어휘 개념을 지정함으로써 워드넷의 계층구조를 활용한 어휘 개념 추상화 수준 인자를 추가적으로 도입하여 특정 영역에서의 전체 시스템 최적화를 지원할 수 있다.

마지막 역할과 관련하여 본 논문의 실험에서는 단백질 간 상호작용 추출(PPII, PPIC) 성능의 심층적 최적화를 위해서 기존의 SVM에서 지원되던 정규화 매개변수 외에 구문 트리 커널의 소멸인자와 시맨틱 구문 트리 커널의 어휘 추상화 인자를 새롭게 도입하였다. 이를 통해 구문 트리 커널을 적용함에 있어서 소멸인자의 역할의 중요성을 간파할 수 있었고, 시맨틱 구문 트리 커널이 기존 시스템의 성능향상에 도움을 줄 수 있음을 실험적으로 보여주었다. 특히 단백질 간 상호작용 식별 문제보다도 비교적 난이도가 높은 상호작용 분류에 더욱 효과적임을 알 수 있었다. 더불어 시맨틱 구문 트리 커널의 어휘 추상화 인자도 소멸인자와 더불어 성능향상에 중요한 요소임을 증명하였다.

향후 연구 방향으로 앞에서도 잠시 언급하였던 것처럼 시맨틱 구문 트리 커널에서 문맥기반 어휘 중의성 해소 시스템의 성능 향상이 가장 시급하다. 특정 문장 내에서 특정 어휘에 대한 의미를 파악함에 있어서 문맥 정보의 중요도는 매우 높다. 따라서 보다 심층적인 언어 분석을 통해서 대상 문맥을 확대시킴으로써 정확한 어휘 개념 추출이 이루어져야 한다. 이를 위해서 추가적인 언어자원이나 언어분석시스템의 도입이 필요하다. 이와 더불어 기존 커널과 본 논문의 시맨틱 구문 트리 커널을 결합한 혼합 커널(composite kernel) 개발을 생각해 볼 수 있다. Airola et al.(2008)가 제안한 의존 그래프 커널 등과 같은 기존 커널과의 밀결합을 통해서 새로운 커널을 구성할 수 있을 것이다. 마지막으로 본 논문에서 제안한 시맨틱 구문 트리 커널을 기반으로 범용 관계 추출(relation extraction) 및 의미역 부착(semantic role labeling) 분야에도 적용할 계획이다.

참 고 문 헌

[1] Airola, A., Pyysalo, S., Bjorne, J., Pahikkala, T., Ginter, F., & Salakoski, T. 2008. "All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning." *BMC Bioinformatics*, 9(S2).

[2] Andrade, Miguel A. & Valencia, A. 1998. "Automatic extraction of keywords from scientific text: Application to the knowledge domain of protein families." *Bioinformatics*, 14(7): 600-607.

[3] Banerjee, S., & Pedersen, T. 2002. "An adapted Lesk algorithm for word sense disambiguation using WordNet." *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics(CICLing-2002)*, 136-45.

[4] Blaschke, C., Andrade, M., Ouzounis, C., & Valencia, A. 1999. "Automatic extraction of biological information from scientific text: Protein-protein interactions." *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, 7: 60-67.

[5] Bunescu, R., Ge, R., Kate, R., Marcotte, E., Mooney, R., Ramani, A., & Wong, Y. 2005. "Comparative experiments on learning information extractors for proteins and their interactions." *Artiicial Inteligence in Medicine, Summarization and Information Extraction from Medical Documents*, 33: 139-155.

[6] Collins, M., & Duffy, N. 2001. "Convolution kernels for natural language." *NIPS-2001.*

[7] Craven, M., & Kumlien, J. 1999. "Constructing biological knowledge bases by extracting information from text sources." *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*, 77-86.

[8] Ding, J., Berleant, D., Nettleton, D., & Wurtele, E. 2002. "Mining MEDLINE: abstracts, sentences, or phrases?" *Proceedings of PSB'02*, 326-337.

[9] Fundel, K., Küffner, R., & Zimmer, R. 2007. "RelEx - Relation extraction using dependency parse trees." *Bioinformatics*, 23: 365-371.

[10] Gondy, L., Hsinchun, C., & Martinez, Jesse D. 2003. "A shallow parser based on closed-class words to capture relations in biomedical text." *Journal of Biomedical Informatics*, 36(3): 145-158.

[11] Lesk, M. 1986. "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone." *Proceedings of the 5th annual international conference on Systems documentation*, 24-26.

[12] Marcotte, Edward M., Xenarios, I., & Eisenberg, D. 2001. "Mining literature for protein-protein

interactions." *Bioinformatics*, 17(4): 359-363.

[13] Miwa, M., Sætre, R., Miyao, Y., & Tsujii, J. 2009. "Protein-protein interaction extraction by leveraging multiple kernels and parsers." *International Journal of Medical Informatics*.

[14] Moschitti, A. 2006. "Making tree kernels practical for natural language learning." *Proceedings of EACL*.

[15] Nedellec, C. 2005. "Learning language in logic - genic interaction extraction challenge." *Proceedings of LLL'05*, 31-37.

[16] Nikolai, D., Anton, Y., Sergei, E., Svetalana, N., Alexander, N., & llya, M. 2004. "Extracting human protein interactions from MEDLINE using a full-sentence parser." *Bioinformatics*, 20(5): 604-611.

[17] Ono, T., Hishigaki, H., Tanigam, A., & Takagi, T. 2001. "Automated extraction of information on protein-protein interactions from the biological literature." *Bioinformatics*, 17(2): 155-161.

[18] Pyysalo, S., Airola, A., Heimonen, J., Björne, J., Ginter, F., & Salakoski, T. 2008. "Comparative analysis of five protein-protein interaction corpora." *BMC Bioinformatics*, 9(S6).

[19] Pyysalo, S., Ginter, F., Heimonen, J., Bjorne, J., Boberg, J., Jarvinen, J., & Salakoski, T. 2007. "BioInfer: A corpus for information extraction in the biomedical domain." *BMC Bioinformatics*, 8(50).

[20] Sekimizu, T., Park, H. S., & Tsujii, J. 1998. "Identifying the interaction between genes and gene products based on frequently seen verbs in MEDLINE abstracts." *Workshop on genome informatics*, 9: 62-71.

[21] Temkin, Joshua M., & Gilder, Mark R. 2003. "Extraction of protein interaction information from unstructured text using a context-free grammar." *Bioinformatics*, 19(16): 2046-2053.

[22] Vishwanathan, S. V. N., & Smola, A. J. 2003. "Fast kernels for string and tree matching." *Advances in Neural Information Processing Systems, MIT Press*, 15: 569-576.

[23] Wikipedia. [online]. [cited 2010.11.1]. 〈http://en.wikipedia.org/wiki/Protein-protein_interaction〉.

[24] Zhang, M., Zhang, J., Su, J., & Zhou, G. 2006. "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*, 825-832.

[25] Zhou, D., & He, Y. 2008. "Extracting interactions between proteins from the literature." *Journal of Biomedical Informatics*, 41: 393-407.