ISBN 978-89-6211-581-9

의미 정보를 활용한 관계 추출 시스템 개발 및 성능 평가

정창후(KISTI)
최성필(KISTI)
최윤수(KISTI)
전홍우(KISTI)
정도헌(KISTI)
홍순찬(KISTI)
조민희(KISTI)
성원경(KISTI)

ISBN 978-89-6211-581-9

의미 정보를 활용한 관계 추출 시스템 개발 및 성능 평가

정창후(KISTI)
최성필(KISTI)
최윤수(KISTI)
전홍우(KISTI)
정도헌(KISTI)
홍순찬(KISTI)
조민희(KISTI)
성원경(KISTI)

한국과학기술정보연구원
Korea Institute of Science and Technology Information

목차

[검증평가의견서] ····· 1
[의미 정보를 활용한 관계 추출 시스템 개발 및 성능 평가] ····· 8
1. 개요 ····· 9
2. 관련 연구 ····· 11
3. 의미 정보를 활용한 관계 추출 시스템 개발 ····· 13
3.1 구문 트리 커널의 문제점 ····· 13
3.2 시맨틱 구문 트리 커널 ····· 19
3.3 시맨틱 구문 트리 커널에 대한 분석적 접근 ····· 23
3.4 PAS 패턴 유사도 커널 ····· 27
4. 시스템 구성 ····· 31
5. 실험 및 분석 ····· 35
5.1 실험 대상 말뭉치 ····· 35
5.2 실험 대상 시스템 ····· 43
5.3 성능 측정 기준 ····· 44
5.4 단백질 상호작용 식별(PPII) 실험 ····· 45
5.5 단백질 상호작용 분류(PPIC) 실험 ····· 48
5.6 PLOT 간 연관관계 추출 실험 ····· 50
6. 결론 및 향후 연구 ····· 52
[참고문헌] ····· 56

그림 목차

[그림 1] 부분트리 기법으로 분리된 구문 트리 ····· 14
[그림 2] 부분집합트리 기법으로 분리된 구문 트리 ····· 14
[그림 3] 동일한 의미에 대한 상이한 표현에 따른 구문 트리 커널의 한계점 ····· 18
[그림 4] 시맨틱 구문 트리 커널 계산 방법 도식화 ····· 26
[그림 5] Enju 파서의 문장 분석 결과 ····· 28
[그림 6] Predicate-Argument 관계 그래프 ····· 29
[그림 7] PAS 패턴 ····· 29
[그림 8] 시스템 구성도 ····· 31
[그림 9] 전체 구문 트리 및 PT 기반 가지치기된 부분 트리 ····· 33
[그림 10] 테스트컬렉션 구축 도구 ····· 43

표 목차

[표 1] (식 6)의 delta 함수 계산 알고리즘 ····· 16
[표 2] WordNet을 활용한 문맥 기반 어휘 중의성 해소 알고리즘 ····· 20
[표 3] 시맨틱 구문 트리 커널의 입력 매개 변수 ····· 21
[표 4] Δsem(n1, n2, λ, σ, α) 계산 알고리즘 ····· 22
[표 5] Enju 파서 분석 결과의 각 열에 대한 설명 ····· 28
[표 6] 구문 트리 가지치기 방법 ····· 32
[표 7] Five PPI Corpora 규모 및 내용 ····· 36
[표 8] BioInfer 말뭉치내의 단백질 상호작용 종류 및 인스턴스 개수 ····· 37
[표 9] BioInfer Relationship Ontology에서의 수준별 노드 개수 ·· 38
[표 10] 실험에 사용된 관계집합 종류 및 개수 ····· 39
[표 11] KREC DTD 정보 ····· 41
[표 12] KREC 연관관계 정보 ····· 42
[표 13] KREC 통계 ····· 43
[표 14] 실험 대상 시스템 설정 종류 및 개수 ····· 44
[표 15] 각 말뭉치별 최고 성능을 나타내는 설정 값 및 성능 세부 정보 ····· 45
[표 16] ma-F1 기준 상위 20개 시스템에서의 설정 분포 ····· 46
[표 17] 타 시스템과의 성능 비교 ····· 46
[표 18] 각 관계 집합별 최고 성능을 나타내는 설정 값 및 성능 세부 정보 ····· 48
[표 19] ma-F1 기준 상위 30개 시스템에서의 설정 분포 ····· 48
[표 20] PTK의 최고성능설정과 동일한 설정에서의 SPTK와의 성능비교 ····· 49
[표 21] PLOT 간 연관관계 추출 성능 ····· 50

[검증평가의견서]

임해창(고려대학교 교수)

조현양(경기대학교 교수)

오재일(중앙대학교 교수)

[검증평가의견서] - ①

● 검증평가자 : 임해창(고려대학교 교수, rim@nlp.korea.ac.kr)

정보 추출은 자연어에서 유용한 정보를 자동으로 수집하기 위해 비정형화된 텍스트 데이터 내에서 정형화된 데이터를 추출하는 태스크이다. 고유 명사 등을 판별하는 개체명 인식과 개체명 간의 관계를 판별하는 관계 추출이 가장 대표적인 정보 추출 문제이다. 본 연구는 자동 관계 추출을 하기 위한 자질과, 다양한 자질을 사용할 수 있는 커널 기반 방법에 대해 다루고 있다.

본 연구는 기존에 관계 추출 분야에서 높은 성능을 보인 구문 트리 커널을 확장한 시맨틱 구문 트리 커널을 제안하였다. 기존의 구문 트리 커널은 학습 집합과 실험 집합 각각의 구문 트리의 단말 노드인 어휘가 다를 경우 구문의 유사도도 낮아지는 문제를 보완하기 위해 각 문장에서 문맥으로 나타난 어휘와 워드넷을 이용하여 어휘의 의미적 유사도를 추가로 계산하는 방법으로 접근하였다. 또한 어휘를 추상화하는 수준을 세 단계로 지정할 수 있도록 하여 문제 영역과 활용 분야에 따라 중의성과 자료 부족 문제간에 적절한 균형을 맞출 수 있도록 하였다. 또한 본 연구에서는 술어-논항 구조 패턴의 유사도도 추가로 사용하였다. 이때 술어-논항 구조 패턴을 표현할 때 구문적 변형을 제거하여 최대한 일반화를 하였기 때문에 술어-논항 구조 패턴 유사도의 경우 적은 말뭉치에서도 최대한 많은 문장에 적용할 수 있도록 하였다.

관계 추출 태스크에서 제안하는 방법의 유용성을 보이기 위해 본 연구에서는 바이오 분야의 다섯 개의 말뭉치에 대해 이진 분류 기반의 단백질간 상호작용 식별, 다중 분류 체계 기반의 단백질간 상호작용 분류에 대한 실험을 각각 수행하였다. 또한 녹색기술 분야의 과학기술문헌에서 장소, 위치, 기관, 기술용어간의 연관 관계 자동 추출에 관한 실험도 수행하였다. 다양한 말뭉치에서 실험해본 결과 제안하는 방법이 유용함을 입증하였다.

기존 연구에서 어휘 유사성과 구문 구조 유사성만을 고려한 것에 비해 본 연구에서는 워드넷과 술어-논항 구조까지 고려하여 다양한 자질을 결합하고자 하였

다. 향후 연구를 위해서는 본 연구에서 제안한 각 자질이 얼마나 정확히 추출되었으며 전체 성능에 어느 정도 영향을 미치는지에 대한 분석이 필요할 것이다. 또한, 거시 평균 기반 F 점수와 미시 평균 기반 F 점수만 비교할 것이 아니라 각 클래스 별로 성능이 얼마나 차이가 났는지, 제안한 방법에서 발생한 오류 유형으로 어떤 것이 있는지에 대한 자세한 분석이 필요할 것이다.

[검증평가의견서] - ②

● 검증평가자 : 조현양(경기대학교 교수, hycho@kyonggi.ac.kr)

1. 본 연구의 의의

▷ 과학기술문헌의 문장 구성상 핵심적 혹은 주체가 되는 요소 즉, PLOT(Person, Location, Organization, Terminology) 간의 의미적 연관관계를 자동으로 추출하는 원천기술을 개발하고 이를 실용화하기 위한 다양한 성능개선노력이 엿보이는 보고서임.

▷ 기존의 구문트리 커널이 가지고 있는 문장 간 구문적 유사도 중심의 커널 수치 계산에 대한 문제점을 최소화할 수 있도록 어휘의 의미상의 유사도를 함께 고려할 수 있는 새로운 방안은 자동 추출 색인 어휘가 가지는 중의성 문제에 대한 해결책으로 제시될 수 있을 것임.

▷ 근본적으로는 현재까지 미해결 과제로 남아 있는 분야이므로 실용화를 위한 많은 노력이 필요할 것으로 사료됨.

▷ 개발된 시스템은 기술용어 간의 관계정보 추출을 통해서 미래 해당 분야의 기술발전 방향을 예측하거나, 해당 분야의 지식구조를 파악하는 데에도 활용될 수 있을 것임

2. 보고서의 구성

▷ 전체적으로 논문 형식으로 구성되어서 내용전개가 매우 간결하고 깔끔하나, 기술보고서라는 차원에서는 직관성이 조금 떨어지는 느낌이 있음.

▷ 목차나 표 및 그림, 그리고 내용의 기술이 매우 체계적으로 전개되어 있음.

▷ 보다 직관적으로 이해할 수 있는 다양한 예제가 추가되면 보고서의 내용이 보다 객관적이고 실질적으로 보완될 것으로 사료됨

3. 향후 발전방향 제시 및 보완 사항

▷ 현재까지는 실용화에 많은 문제가 있음이 보이지만 보다 다양한 분

야의 데이터를 대상으로 실험을 지속하여 그 결과에 대한 일관성을 보장함으로써 연구의 연속성을 확보하는 것이 가장 중요함.

▷ 계량 서지학에서 다루는 다양한 메타데이터 수준의 문헌 분석과의 연계 및 기술 결합이 가능하다면 본 보고서에서 개발된 연구결과물의 실질적인 활용이 쉽게 가시화될 수 있어 보임.

▷ 본 보고서에서 수행한 연구는 원천기술개발에 가까우므로 다양한 수준 높은 연구실적(논문, 특허 등)이 도출되어야 함이 마땅함. 이를 위해서, 연구 분야의 세계적 수준에 대한 면밀한 분석 및 파악이 선행되어야 함.

[검증평가의견서] - ③

● 검증평가자 : 오재일(중앙대학교 교수, ohjeill@cau.ac.kr)

1. 본 연구의 의의

▷ 전문용어 간의 상관관계를 자동으로 파악하는 기술에 대한 근원적인 접근을 한 원천기술연구라고 사료됨.

▷ 전문용어가 다양한 기술분야에서 독립적으로 만들어지고 있으며 이에 따라 그들간의 관계를 파악함에 있어서도 분야적 전문지식이 반드시 필요하다고 생각함.

▷ 본 연구는 이러한 분야적 전문지식을 적용하여 성능을 개선하기 이전 단계인 분야 독립적 원천기술 개발에 가까움.

2. 보고서의 구성

▷ 전체적인 구성이 학위 논문의 구성과 유사하게 전개되는 흐름이 보이는 것으로 판단됨. 연구 과제의 성격이 이러한 서술전개가 유리하다고 판단된다면 상관 없지만, 보다 기술보고서의 특성(성과물에 대한 보다 자세한 소개, 활용성의 부각 등등)을 지니는 형식으로 전개하는 것도 하나의 방법이라 판단됨.

▷ 1장 개요, 2장 관련 연구, 3장의 일부에서 모두 문헌조사의 내용이 언급되고 있는바 연구 배경을 설명하기 위한 문헌조사 소개 부분을 중복 관점에서 재정리할 필요가 있음.

▷ 3장은 전체적인 연구 논점의 접근 방법에 대한 설명이므로 이를 종합적으로 정리하는 내용을 3장 마지막 부분에 정리 차원에서 제시하였으면 함.

▷ 본문의 흐름 상 제시하지 못하는 추가적인 자료는 부록으로 표기하여 제시하여 보고서의 완성도를 높이는 것이 추후 연구자 및 후속 연구를 위해 바람직할 것으로 판단됨.

3. 향후 발전방향 제시 및 보완 사항

▷ 개발된 연구결과물의 세계적 수준에 대한 명확한 제시가 필요하며,

한계성 혹은 정량적이고 객관적인 평가 내용, 앞으로의 확장성에 대해서 보다 명확히 본문에 언급할 필요가 있음. 추후 연구의 개연성을 제시할 필요

▷ 분야 전문가들을 활용한 분야지식 적용을 통해 시스템의 성능을 개선하고 실용성을 확보하는 것이 시급함.

[의미 정보를 활용한 관계 추출 시스템 개발 및 성능 평가]

1. 개요

자연어 처리 및 텍스트 마이닝 분야에서 정보 추출(Information Extraction)은 핵심적인 영역으로 인식되고 있다. 정보 추출의 최종 목표는 비정형적인 텍스트 데이터 내에서 테이블화된 정형 데이터를 추출, 변환하기 위해서 텍스트 내에 존재하는 중요하고 연관성 있는 정보를 식별하는 것이다[30]. [31]에서는 이러한 정보 추출 기술을 구성하는 요소 기술로서 (1) 개체명 인식(Named-Entity Recognition), (2) 관계 추출(Relation Extraction), (3) 대용어 참조 해소(Coreference Resolution) 등을 언급하고 있다. 이 중 관계 추출은 현재까지도 가장 난이도가 높은 미해결 분야로 인식되고 있다[30, 31, 32].

현재까지 관계추출의 성능을 높이기 위해서 다양한 지도 학습(Supervised Learning) 기반의 관계 추출 기법이 소개되었다. 이들은 (1) 규칙기반 방법(Rule-based Methods), (2) 자질기반 방법(Feature-based Methods), (3) 커널기반 방법(Kernel-based Methods)의 세 가지 유형으로 분류될 수 있다. 이들 중 비교적 최근에 개발된 관계 추출에 특화된 커널 함수를 새롭게 구성하여 이를 기반으로 지지벡터기계(Support Vector Machines)에 적용하는 커널기반 방법의 효과가 주목 받고 있다. 관계추출 분야에서의 커널기반 방법의 특징은 한 문장에 존재하는 두 개체간의 관계를 가장 잘 표현하고, 이를 포함하는 두 관계 포함 문장들 간의 유사도를 가장 효과적으로 계산하는 커널을 구성하기만 하면, 그 성능이 매우 높게 나타난다는 것이다. 특히 개체 쌍 및 그들 간의 관계가 포함된 문장을 구문 분석하여 얻어지는 구문 트리 기반의 커널을 기반으로 한 합성곱 구문 트리 커널(Convolution Parse Tree Kernel)과 이것을 기반으로 추가 부가적인 자질을 결합한 혼합 커널(Composite Kernel) 기반 관계추출 기법의 성능이 매우 뛰어난 것으로 알려져 있다[20, 28].

본 연구에서는 관계 추출을 위해서 기존에 연구되어 높은 성능을 나타낸 구문 트리 커널을 확장한 시맨틱 구문 트리 커널을 제안한다. 기존 구

문 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 관계 추출의 전체적인 성능에 악영향을 줄 수 있다는 점이다. 이를 극복하기 위해서 본 연구에서는 두 구문 트리의 구문적 유사도(syntactic similarity)와 어휘 의미적 유사도(lexical semantic similarity)를 동시에 효과적으로 계산하여 이를 결합하는 새로운 커널을 고안하였다. 또한 문장 내에 존재하는 predicate와 argument 간의 의미적 연결 구조를 활용하는 PAS(Predicate-Argument Structure) 패턴 유사도 커널을 결합하여 좀 더 다채로운 혼합 커널을 구성해 보았다.

본 보고서의 구성은 다음과 같다. 우선 2장에서 관계추출과 관련한 선행 연구에 대해서 살펴본다. 이어서 3장에서는 본 연구에서 제안하는 의미 정보를 활용한 관계 추출 시스템의 개발에 대해서 살펴보고, 4장에서는 전체적인 시스템 아키텍처에 대해서 설명한다. 그리고 5장에서 본 연구에서 제시한 시스템의 성능을 평가하고, 실험 결과에 대한 분석을 제시한다. 마지막으로 6장에서 결론과 향후 연구 방향을 논의한다.

2. 관련 연구

지도학습기반 관계추출(supervised relation extraction)은 1997년도에 개최된 MUC-7(Message Understanding Conference 7)에서 처음으로 도입된 '템플릿 기반 관계 추출(Template Relation Extraction)' 태스크에서 본격적으로 기계학습 기반의 관계추출을 위한 학습 집합을 제공함으로써 이 분야 연구의 단초를 제공하였다.

그 이후로 많은 관계 추출 기법들이 개발되었으며, 이를 처리 기법에 따라 분류하면 크게 (1) 규칙 기반 방법(rule-based methods), (2) 자질 기반 방법(feature-based methods), 그리고 (3) 커널 기반 방법(kernel-based methods)으로 구분된다.

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[34]. 이를 기반으로 GuoDong et al. (2005)는 지지벡터기계(Support Vector Machines)를 활용하여 더 확장되고 세분화된 자질 정보를 관계 추출에 적용하였다[35]. 이와 유사하게 Zhao et al. (2005)는 모든 세부 자질을 종류별로 구분하고 이를 개별적인 선형 커널로 구성하여 최종적으로 혼합 커널로 결합하는 기법을 제안하였다[33]. 이 방법은 커널 함수를 직접 고안하고 적용하였다는 점에서 커널 기반 기법으로 분류될 수도 있으나, 커널의 구조가 단순하고 대부분 자질 벡터로 변환될 수 있는 점에 근거하여 자질 기반 방법으로 분류하였다. 기본적으로 위의 논문들 모두 관계 추출을 위한 자질 선정이나 구성 방법에 준거하여 자질 공학적 시도에 국한하여 접근하였으며, 관계 인스턴스의 구문 구조에 대한 적용은 매우 제한적으로 이루어졌다.

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[36]. 이 연구를 기반으로 Culotta et al. (2004)는 의존 구문 트리(dependency parse tree)의 유사도를 측정할 수 있는 커널을 개발하였으며, 최초로 ACE 컬렉션을 대상으로 실험하였으나 그 성능은 비교적 낮았다[30]. 또한 Bunescu et al. (2005)는 [30]의 결과를 확장하여 의존 구문 트리를 부분 트리로 분할하고, 문장 내의 의존 관계 경로를 대상으로 커널 함수를 구성하여 [30]에서보다 더 나은 결과를 얻었다[31].

최근에는 Zhang et al. (2006)이 Collins and Duffy (2001)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[20, 28]. 또한 GuoDong et al. (2007)은 [20]에서 제안한 구문 트리 커널의 가지치기 기법 및 커널 계산 기법을 확장하여 개체 쌍 주변 문맥까지도 포괄하는 새로운 트리 커널을 개발하였다[28]. 그러나 두 기법 모두 관계 인스턴스의 구문 구조 정보와 컬렉션에서 제공하는 개체 자질 정보만을 활용함으로써 일반적으로 자질 기반 기법에서 추구하고자 했던 관계 추출을 위한 인스턴스내 문맥 다양성 정보의 집약적 적용에 대한 연구는 수행되지 않았다.

본 연구에서는 이러한 기반 연구에 기초하여 커널 기반 기법의 구조와 특징에 대해서 살펴보고, 서로 다른 자질을 이용하는 다양한 커널의 조합을 활용하여 관계 추출 시스템의 성능을 향상시키고자 한다.

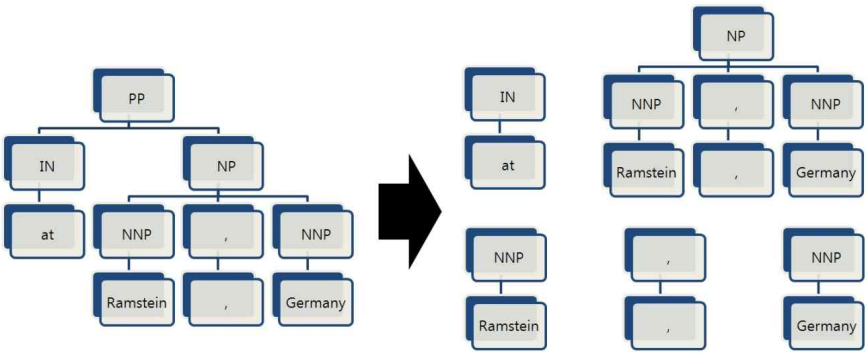
3. 의미 정보를 활용한 관계 추출 시스템 개발

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.

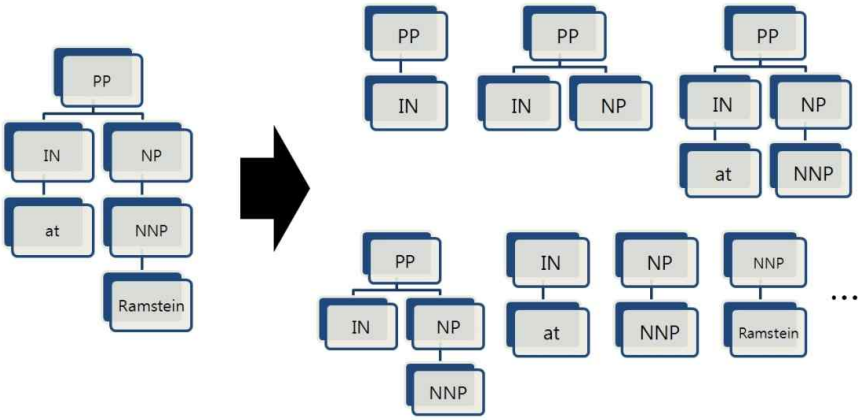
트리 커널은 하부 트리 분리 방법에 따라 Vishwanathan and Smola (2003)이 [15]에서 제안한 부분트리 커널(SubTree Kernel, STK)과 Collins and Duffy (2001)가 [16]에서 고안한 부분집합트리 커널(SubSet Tree Kernel, SSTK)로 나뉜다. 부분트리 기법은 트리 내에서 특정 노드의 모든 자식 노드로 구성된 부분 트리를 구성하는 것이다. 따라서 모든 부분 트리는 말단 자식 노드로서 전체 트리의 잎 노드(leaf node)를 가져야 하며, 구문 생성 규칙에 위배되지 말아야 한다. 이에 반해서 부분집합트리 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, 부분트리 커널 기법보다 훨씬 많은 부분 트리를 생성한다.

아래 그림은 두 가지 방법에 의해서 구문 트리 내의 부분 트리가 분리되는 모습을 보여준다.



[그림 1] 부분트리 기법으로 분리된 구문 트리

[그림 1]은 “at Ramstein, Germany”라는 전치사구에 대해서 부분트리 기법을 적용하여 분리된 부분트리 집합을 보여준다. [그림 2]는 “at Ramstein”이라는 전치사구가 부분집합트리 기법에 의해서 분리되는 모습을 보여준다.



[그림 2] 부분집합트리 기법으로 분리된 구문 트리

두 입력 구문 트리의 유사도를 효과적으로 측정하기 위해서, [16, 17]에서는 한 구문 트리를 구성하는 요소 트리 집합을 추출하고 이를 벡터 공간의 하나의 축으로 지정함으로써, 구문 트리가 공간 내의 특정 벡터로 사상(mapping)될 수 있도록 하였다. 따라서 입력 구문 트리 $\vec{x}_{pt}$ 는 다

음과 같이 함수 φ 에 의해서 새로운 자질 공간 Φ로전사된다.

$$\varphi : \vec{x}_{pt} \in \mathrm{X} \mapsto \varphi\left(\vec{x}_{pt}\right) \in \Phi \subseteq \square^{N} \qquad \text{(식 1)}$$

식 (1)에서 사상된 자질 공간은 N-차원의 유클리드 공간이며, 자질 공간 내에서의 $\varphi\left(\vec{x}_{pt}\right)$ 은 다음과 같이, 구성 요소 트리의 출현 빈도 벡터로 표현된다.

$$\begin{aligned} &\varphi\left(\vec{x}_{pt}\right) = \left(f_1\left(\vec{x}_{pt}\right), f_2\left(\vec{x}_{pt}\right), \ldots, f_N\left(\vec{x}_{pt}\right)\right) \\ &f_i\left(T\right) = \text{ the number of } subtree_i \in S, \text{ appearing in } T \\ &S = \text{a set of all the unique subtrees of the entire tree set.} \end{aligned} \qquad \text{(식 2)}$$

식 (2)에서 함수 $f_i(T)$는 구문 트리 T 내에 존재하는 i 번째 부분 트리의 출현 빈도를 계산한다. 따라서 $\varphi\left(\vec{x}_{pt}\right)$ 는 그 내부 구조에 따라 N-차원의 희소 벡터(sparse vector)로 표현될 수 있으며, 이들 간의 유사도, 즉 커널 값은 다음과 같이 내적을 통해서 계산할 수 있다.

$$\begin{aligned} K_{pt}\left(\vec{x}_{pt}, \vec{x}'_{pt}\right) &= \left\langle \varphi\left(\vec{x}_{pt}\right), \varphi\left(\vec{x}'_{pt}\right) \right\rangle \\ &= \sum_{i=1}^{N} \left[f_i\left(\vec{x}_{pt}\right) \square f_i\left(\vec{x}'_{pt}\right) \right] \end{aligned} \qquad \text{(식 3)}$$

그러나 특정 입력 구문 트리 집합 내에 존재하는 모든 부분 트리를 추출하고, 이를 이용하여 개별 입력 구문 트리에 대해서 벡터를 구성하는 작업은 상당히 비효율적이다. 따라서 [16]에서는 $K_{pt}\left(\vec{x}_{pt}, \vec{x}'_{pt}\right)$ 를 직접 계산하는 재귀적 방법을 고안해 냄으로써 처리 속도를 향상시켰다. 본 연구에서는 세부 알고리즘에 대한 내용은 생략하며, [16, 17]에 자세하게 기술되어 있다.

Moschitti (2006)에 의하면, 부분집합트리 커널의 성능이 부분트리 커널에 비해서 훨씬 좋게 나타났다[15, 16, 17]. 한편 Moschitti (2006)는 [17]에서 이들 두 가지 커널을 빠르게 계산할 수 있는 알고리즘을 개발하고, 이를 의미역 결정(semantic role labeling)에 활용하여 괄목할 만한 성능을

보여주었다.

합성곱 구문 트리 커널 계산 방법의 주된 아이디어는 비교 대상인 두 트리 T1과 T2의 공통 하부 요소 트리의 개수를 계산하는 것이다. 우선 특정 트리 T의 하부 요소 트리 집합을 F = {f_1, f_2, ...}라고 할 때, 지시함수 $I_i(n)$은 다음과 같이 정의된다.

$$I_i(n) = \begin{cases} 1 & f_i\text{의 최상위 노드가 } n\text{이면,} \\ 0 & \text{아니면.} \end{cases} \quad \text{(식 4)}$$

또한 특정 노드 n_1과 n_2를 최상위 노드로 가지는 트리의 공통 하부 트리 개수를 계산하기 위해서 다음과 같이 $\Delta(n_1, n_2)$를 정의한다.

$$\Delta(n_1, n_2) = \sum_{i=1}^{|F|} I_i(n_1) I_i(n_2) \quad \text{(식 5)}$$

(식 4)와 (식 5)를 기반으로 트리 커널 함수는 다음과 같이 정의된다.

$$K(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2) \quad \text{(식 6)}$$

N_{T_i}는 T_i의 모든 구성 노드 집합을 의미한다. 따라서 (식 6)은 두 트리 T_1, T_2의 모든 자식 노드 쌍에 대해서 $\Delta(n_1, n_2)$의 누적합을 계산한다. [16]에 기술된 내용을 바탕으로 (식 6)에서 $\Delta(n_1, n_2)$은 다음과 같은 알고리즘으로 계산될 수 있다.

[표 1] (식 6)의 delta 함수 계산 알고리즘

```
FUNCTION delta(TreeNode n1, TreeNode n2, λ, σ)
n1 = one node of T1;                // 구문 트리에서의 특정 노드
n2 = one node of T2;
λ = tree kernel decay factor;       // 트리 커널 소멸 인자
σ = substructure division methods;  // ST(0), SST(1) 선택 인자
BEGIN
        nc1 = get_children_number(n1);      // 현재 노드의 자식노드 수
        nc2 = get_children_number(n2);
        // 두 노드가 모두 말단 노드일 경우,
        IF nc1 EQUAL 0 AND nc2 EQUAL 0 THEN
                nv1 = get_node_value(n1); // 현재 노드의 노드 값을 가져
옴
                nv2 = get_node_value(n2);
                // 두 노드 값(단어)이 같으면 1을 리턴
```

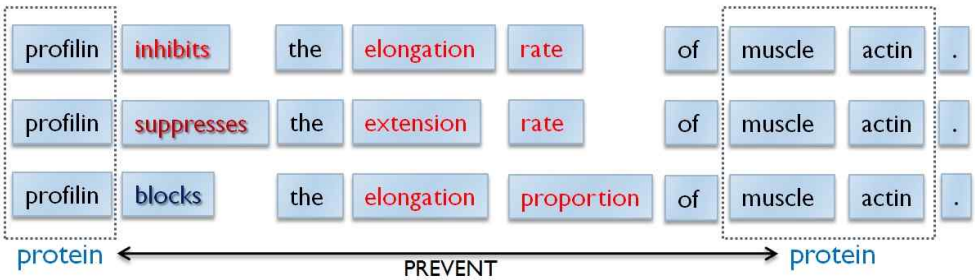
```
15                  IF nv1 EQUAL nv2 THEN
16                          RETURN 1;
17                  ENDIF
18          ENDIF
19          np1 = get_production_rule(n1);          // 노드의 문법생성규칙 가져
20 옴
21          np2 = get_production_rule(n2);
22          IF np1 NOT EQUAL np2 THEN     // 만약 두 문법생성규칙이 서
23 로 다르면
24                  RETURN 0;
25          END IF
26          // 만약 두 노드의 문법생성규칙이 같고,
27          // 두 노드 모두 품사태그 노드이면,
28          // 소멸인자 리턴
29          IF np1 EQUAL np2 AND nc1 EQUAL 1 AND nc2 EQUAL 1
30 THEN
31                  RETURN λ;
32          END IF
33
34          // 재귀적으로 현재 노드의 자식노드에 대한 delta 값 계산
35          // 현재 노드의 모든 첫 번째 자식 노드들에 대한 delta 값의 누
36 승 계산
37          mult_delta = 1;
38          FOR I = 1 TO nc1
39                  nch1 = Ith child of n1;
40                  nch2 = Ith child of n2;
41                  // 자식 노드들에 대한 delta 함수 재귀 호출
                    mult_delta = mult_delta × (σ + delta(nch1, nch2, λ, σ));
            END FOR

            RETURN λ × mult_delta;
    END
```

위 알고리즘에서 get_children_number() 함수는 현재 노드에 직접 연결된 자식 노드의 개수를 계산하고, get_node_value() 함수는 현재 노드의 값(품사태그, 구문태그, 단어)을 반환한다. 또한 get_production_rule() 함수는 현재 노드와 자식노드 간의 구조를 파악함으로써 현재 위치에서의 문법생성규칙을 조사한다. 하부 트리 분리 방법 선택 인자 σ는 [표 1]의 37 번째 라인에서 그 값에 따라 부분 트리커널 혹은 부분집합 트리커널

값을 계산하게 해 준다. 예를 들어, 소멸인자가 1이라고 가정할 때, 만일 σ가 0이면 현재 두 노드들의 모든 자식노드들의 문법생성규칙이 동일해야만 $\Delta(n_1, n_2)$가 1이 된다. 따라서 이는 부분 트리 커널(STK)을 의미한다.

마지막으로 트리 커널 소멸인자 λ는 비교 대상이 되는 구문 트리들의 깊이(tree depth)가 서로 상이함에 따라 발생하는 커널 값의 불일치성을 해결하기 위해서 도입되었다. 두 개의 구문 트리가 비교되는 과정에서 말단 노드에 가까워짐에 따라 커널 값에 대한 기여도가 작아질 수 있도록 값을 지정할 수 있다.

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 아래 그림에서 이와 같은 현상을 구체적으로 보여준다.



[그림 3] 동일한 의미에 대한 상이한 표현에 따른 구문 트리 커널의 한계점

[그림 3]의 세 문장은 모두 동일한 의미를 표현하고 있으며, PPI(Protein-Protein Interaction)의 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 [17]에서 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 연구에서 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

Lesk (1986)는 사전 정의문을 이용한 어휘 중의성 해소 알고리즘을 최초로 제안하였다[18]. 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 고안했

으며, 워드넷의 구성요소인 synset, gloss 및 계층 정보가 어휘 중의성 해소의 성능 향상에 기여할 수 있음을 밝혔다.

[표 2] WordNet을 활용한 문맥 기반 어휘 중의성 해소 알고리즘

```
    FUNCTION word_sense_disambiguation(word, POS, context,
 1  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
 6  synset words;
 7  BEGIN
 8       synsets = search_word_in_WordNet(word, POS);
 9       IF (synsets IS EMPTY) THEN
10           RETURN NULL;
11       max_dups = 0;
12       max_synset = NULL;
13       FOR EACH synset IN synsets retrieved
14       BEGIN
15           sw = get_synset_words(synset, level);
16           dups = get_duplication_count(sw, context);
17           IF max_dups < dups THEN
18                 max_dups = dups;
19                 max_synset = synset;
20           END IF
21       END FOR
22
23       RETURN max_synset;
    END
```

위 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, 이들 중에서 대상 단어의 문맥 정보와 가장 일치하는 synset을 선택하는 과정을 수

행해야 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 연구에서 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.

$$K_{sem}(T_1, T_2, \lambda, \sigma, \alpha) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta_{sem}(n_1, n_2, \lambda, \sigma, \alpha) \quad \text{(식 7)}$$

여기서 T_1, T_2는 두 개의 입력 구문 트리를 나타낸다. 이와 더불어 정밀한 커널 계산을 위해 3가지 매개변수가 수반되는데, 이에 대한 내용은 아래 표에 정리되어 있다.

[표 3] 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 대한 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

위 표에서 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, Δsem은 특정 노드 n_1과 n_2를 최상위 노드로 가지는 트리의 공통 하부 트리 개수를 계산하며, 상세한 알고리즘은 아래 표에서 나타내었다.

[표 4] $\Delta_{sem}(n_1, n_2, \lambda, \sigma, \alpha)$ 계산 알고리즘

```
FUNCTION Semantic_Delta(TreeNode n1, TreeNode n2, λ, σ,
α)

BEGIN
     IF n1 and n2 are both terminal nodes THEN
          concept1 = get_semantic_concpet(n1, α);
          concept2 = get_semantic_concept(n2, α);
          IF concept1 == concept2 THEN
               RETURN 1;
          RETURN 0;
     END IF

     IF n1 and n2 are from different productions THEN
          RETURN 0;
     END IF

```

```
17          IF the productions of n1 and n2 are the same THEN
18               IF n1 and n2 are pre-terminal nodes THEN
19                    RETURN λ;
20                         λ∏_{j=1}^{nc(n1)}(σ + Semantic_Delta(ch_{n1}^j, ch_{n2}^j, λ, σ, α))
21               RETURN
22
23          END IF
24
     END
```

위 알고리즘에서 가장 핵심적인 부분은 어휘 개념 생성 부분이다(line 5, 6). 이를 위해서 get_semantic_concept 함수는 [표 2]에서 제시한 워드넷 기반 어휘 의미 중의성 해소 모듈을 이용하여, 현재 단어에 가장 적절한 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 2]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 4]에서 제시된 알고리즘의 나머지 부분은 Moschitti (2006)가 [17]에서 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 7)에서 제시한 커널 함수는 다음과 같이 표현될 수 있다.

$$K_{sem}(T_1,T_2,\lambda,\sigma,\alpha)=\sum_{n_1\in N_{T_1}}\sum_{n_2\in N_{T_2}}\Delta_{sem}(n_1,n_2,\lambda,\sigma,\alpha)$$
$$=sim_{lex}(T_1,T_2,\alpha)+sim_{syn}(T_1,T_2,\lambda,\sigma) \qquad \text{(식 8)}$$

(식 7)을 기반으로 (식 8)은 시맨틱 구문 트리 커널이 (1) 어휘 의미적 유사도(lexical semantic similarity, simlex(T_1,T_2,α))와 (2) 구문적 유사도(syntactic similarity, simsyn(T_1,T_2,λ,σ))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 연구에서 제안하는 어휘 의미적 유사도에 대한 세부적인 계산 방법은 아래 (식 9)와 같다.

$$sim_{lex}(T_1,T_2,\alpha)\equiv sim(W_{T_1},W_{T_2},\alpha) \qquad \text{(식 9-1)}$$
$$\approx\sum_{w_1\in W_{T_1},c_1\subset W_{T_1}^{w_1}}\left(\sum_{w_2\in W_{T_2},c_2\subset W_{T_2}^{w_2}}sim(w_1,w_2,c_1,c_2,\alpha)\right) \qquad \text{(식 9-2)}$$
$$\approx\sum_{w_1\in W_{T_1},c_1\subset W_{T_1}^{w_1}}\left(\sum_{w_2\in W_{T_2},c_2\subset W_{T_2}^{w_2}}I(concept(w_1,c_1,\alpha),concept(w_2,c_2,\alpha))\right) \qquad \text{(식 9-3)}$$
$$\approx\sum_{w_1\in W_{T_1},c_1\subset W_{T_1}^{w_1}}\left(\sum_{w_2\in W_{T_2},c_2\subset W_{T_2}^{w_2}}I(synset(w_1,c_1,\alpha),synset(w_2,c_2,\alpha))\right) \qquad \text{(식 9-4)}$$
$$=\sum_{w_1\in W_{T_1},c_1\subset W_{T_1}^{w_1}}\left(\sum_{w_2\in W_{T_2},c_2\subset W_{T_2}^{w_2}}I(pos(synset(w_1,c_1,\alpha)),pos(synset(w_2,c_2,\alpha)))\right) \qquad \text{(식 9-5)}$$

W_T는 구문 트리 T를 구성하는 단어 집합이며, $W_T^{w_i}$는 이 중에서 w_i 주변의 문맥 단어 집합이다. 본 연구에서는 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 9-1)과 (식 9-2)에서 명확하게 표현된다. (식 9-2)에서 두 문장을 구성하는 모든 단어들에 대해서 문맥(c_1, c_2) 기반의 단어 유사도가 누산된다. 단어 간 유사도를 수치화하여 정확하게 계산하기는 어려우므로, (식 9-3)에서는 개별 단어를 개념화함으로써 두 단어의 유사도 계산을 단순화하였다. 이 식에서 I(A,B)는 A와 B가 동일하면 1을 반환하고, 아니면 0을 반환한다. 결론적으로 이 식은 두 문장을 교차 비교하면서 동일한 개념을 나타내는 단어 쌍의 개수를 계산한다. 이를 위해서 본 연구에서는 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 9-4)에서는 앞의 [표 2]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 식에 포함된 추상화 수준 지정 인자 α는 개별 어휘의 의미적 개념들의 추상화 수준을 결정하게 된다. 한편, 두 문장에 대한 구문적 유사도를 나타내는 sim_{syn}(T1,T2,λ,σ)은 Moschitti (2006)가 제안한 것과 동일하게 다음과 같이 표현될 수 있다.

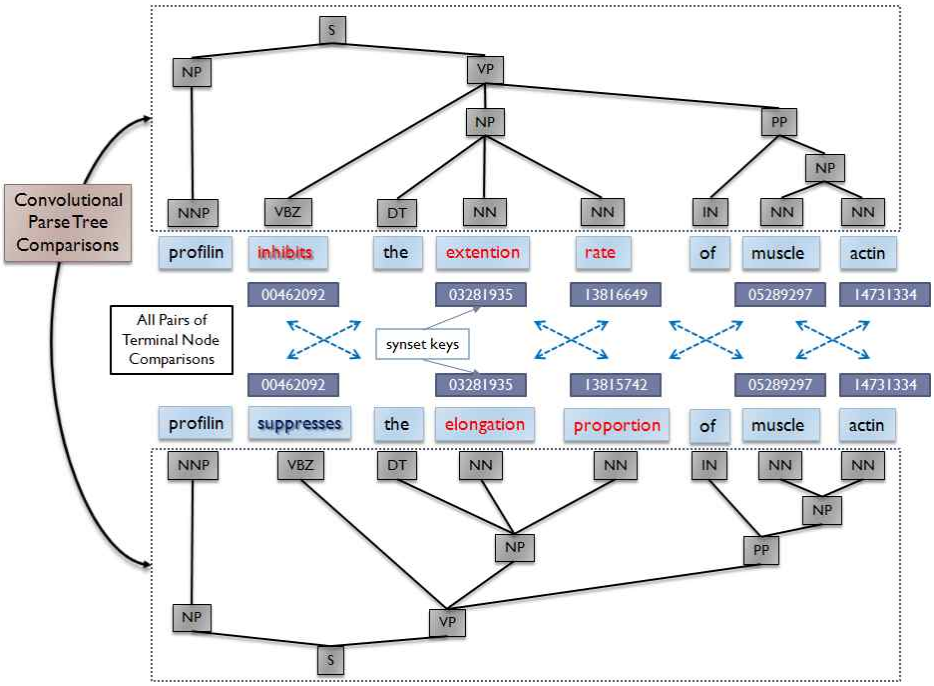
$$sim_{syn}(T_1, T_2, \lambda, \sigma) \equiv \sum_{n_1 \in N_{T_1}, n_1 \notin L_{T_1}} \left(\sum_{n_2 \in N_{T_2}, n_2 \notin L_{T_2}} \Delta(n_1, n_2, \lambda, \sigma) \right) \qquad \text{(식 10)}$$

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 10)에 대한 자세한 내용은 [16, 17]에서 상세히 기술하고 있다. (식 9)과 (식 10)에 의거하여, 시맨틱 구문 트리 커널은 다음과 같이 표현될 수 있다.

$$K_{sem}(T_1, T_2, \lambda, \sigma, \alpha) \equiv \sum_{w_1 \in W_{T_1}, c_1 \subset W_{T_1}^{w_1}} \left(\sum_{w_2 \in W_{T_2}, c_2 \subset W_{T_2}^{w_2}} I(pos(synset(w_1, c_1, \alpha)), pos(synset(w_2, c_2, \alpha))) \right) + \sum_{n_1 \in N_{T_1}, n_1 \notin L_{T_1}} \left(\sum_{n_2 \in N_{T_2}, n_2 \notin L_{T_2}} \Delta(n_1, n_2, \lambda, \sigma) \right) \quad \text{(식 11)}$$

(식 11)은 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] 시맨틱 구문 트리 커널 계산 방법 도식화

[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 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.

3.4 PAS 패턴 유사도 커널

PAS는 predicate-argument 관계를 이용하여 문장 내에 존재하는 각 단어 간의 유의미한 연관관계를 표현하는 구조이다. 그리고 PAS 패턴은 문장을 구성하는 모든 단어에 대한 predicate-argument 관계 그래프에서, 중요하게 지정된 개체와 개체를 연결하는 최소 집합의 predicate-argument로 구성된 순서 열을 의미한다. 이러한 특성 때문에 PAS 패턴은 문장 내에서 상호작용하는 두 개체 간의 연관관계를 표현해주는 중요한 자질 정보가 된다. 따라서 한 개체로부터 시작해서 다른 개체로까지의 의미적 연결고리를 제공해주는 PAS 패턴을 이용하여 관계 추출을 수행할 수 있다.

PAS는 문장에 존재하는 predicate의 argument를 결정함에 있어서 단순히 같은 절에서만 찾는 것이 아니라 풀 파싱을 통하여 지역적인 범위를 넘어서서 문장 전체에서 관련 대상을 찾는다. 따라서 predicate와 관계를 맺고 있는 실질적인 argument를 찾을 수 있다. 또한 PAS는 구문적 변형을 제거하여 동일한 의미의 PAS를 일반화된 형태로 표현하기 때문에 PAS로 구성된 패턴은 문장에서 표면적으로 드러나는 단어 열보다 훨씬 정규화 된 형태를 지니고 있다. 따라서 학습 집합의 규모가 작더라도 PAS 패턴은 좀 더 포괄적인 정보 추출 능력을 가진다. 다시 말해서 PAS

레벨에서는 표면적으로 드러나는 단어 패턴의 이질성이 감소되기 때문에 정보 추출 시스템의 재현율을 증가시킬 수 있다.

본 연구에서는 PAS 패턴을 추출하기 위해서 Enju 파서를 이용하였다. Enju 파서를 이용한 문장 분석 결과의 예는 [그림 5]와 같다.

```
stm@X8br87:~/SINDI/JNI-1.6/tool/enju-2.3.1
Radon_exposure is the second leading cause of lung_cancer in the general population.

ROOT     ROOT     ROOT     ROOT     -1       ROOT     ROOT     is       be       VBZ      VB       1
is       be       VBZ      VB       1        verb_arg12        ARG1     Radon_exposure    radon_exposure    NN       NN       0
is       be       VBZ      VB       1        verb_arg12        ARG2     cause    cause    NN       NN       5
the      the      DT       DT       2        det_arg1          ARG1     cause    cause    NN       NN       5
second   second   JJ       JJ       3        adj_arg1          ARG1     cause    cause    NN       NN       5
the      the      DT       DT       9        det_arg1          ARG1     population        population        NN       NN       11
leading  leading  JJ       JJ       4        adj_arg1          ARG1     cause    cause    NN       NN       5
general  general  JJ       JJ       10       adj_arg1          ARG1     population        population        NN       NN       11
of       of       IN       IN       6        prep_arg12        ARG1     cause    cause    NN       NN       5
of       of       IN       IN       6        prep_arg12        ARG2     lung_cancer       lung_cancer       NN       NN       7
in       in       IN       IN       8        prep_arg12        ARG1     cause    cause    NN       NN       5
in       in       IN       IN       8        prep_arg12        ARG2     population        population        NN       NN       11
```

[그림 5] Enju 파서의 문장 분석 결과

[그림 5]에서 보는 바와 같이 Enju 파서는 문장을 입력으로 받아서 문장을 구성하는 각 단어의 predicate-argument 관계를 분석하여 제공한다. 행으로 나열된 predicate-argument 분석 결과의 각 필드에 대한 설명은 [표 5]와 같다.

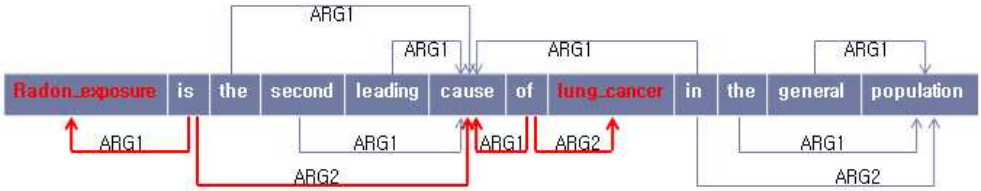
[표 5] Enju 파서 분석 결과의 각 열에 대한 설명

열 번호	상세 설명
1	predicate 단어
2	predicate 단어의 기본형
3	predicate 단어의 품사
4	predicate 단어의 기본형의 품사
5	문장에서 predicate 단어의 위치
6	predicate 종류
7	predicate와 argument 사이의 관계 레이블
8	argument 단어
9	argument 단어의 기본형
10	argument 단어의 품사
11	argument 단어의 기본형의 품사
12	문장에서 argument 단어의 위치

[표 5]에서 설명한 내용을 바탕으로 “Radon_exposure is the second

leading cause of lung_cancer in the general population." 문장에 대한 분석 결과인 [그림 5]의 2번째 행과 3번째 행을 설명하면, 우선 단어 'is'는 동사로서 argument 1과 2를 갖는데 그 중 argument 1은 명사인 단어 'radon_exposure'를 지칭하고 argument 2는 또 다른 명사인 단어 'cause'를 지칭한다는 사실을 나타낸다. 분석 결과의 1번째 행은 단순히 문장의 root predicate를 표현하는 것이고, 4번째 행부터는 2번째와 3번째 행을 해석한 것과 같은 방식으로 해석하면 된다.

Enju 파서에서 제공된 결과를 이용하여 각 단어의 predicate-argument 관계 그래프를 그리면 [그림 6]과 같이 표현된다.



[그림 6] Predicate-Argument 관계 그래프

[그림 6]에서 실제적으로 문장 내에 존재하는 두 개체 간의 유의미한 관계를 표현하는 PAS만을 추출하여 패턴을 구성하면 [그림 7]과 같다. 화살표의 연결은 한 개체로부터 상호작용하는 다른 개체로까지의 predicate-argument 관계를 추적할 수 있다는 것을 의미한다. 따라서 'radon_exposure'와 'lung_cancer' 사이의 관계를 추적해보면 'is cause of'와 같은 중요한 어휘적 패턴을 기반으로 관계가 형성되어 있음을 알 수 있다. 다시 한 번 말하지만, 이러한 패턴은 두 개체 간의 상호작용을 추출하는데 중요한 자질로 사용될 수 있다.

[그림 7] PAS 패턴

결과적으로 개체 1과 개체 2의 관계는 두 개체를 유의미한 관계로 연결

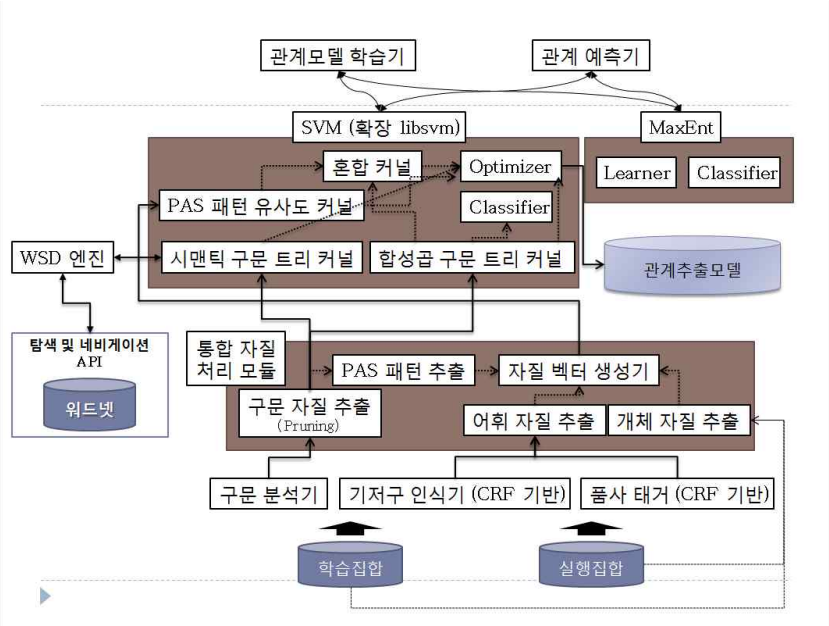
해주는 PAS 패턴에 의하여 식별될 수 있다. 따라서 관계 별로 나타나는 패턴의 집합을 구축하여 이 패턴 집합을 관계 예측의 근거 자질로 활용하면 개체 간의 관계를 추출하는 관계 추출 시스템에 활용할 수 있다.

PAS 패턴을 어휘 자질로 활용하기 위해서 predicate-argument 구조, 즉 [그림 7]에서 화살표로 연결되는 predicate 단어와 argument 단어, predicate의 종류, 그리고 predicate와 argument 사이의 관계 레이블을 이용하여 벡터 값을 생성하였다. 그리고 이 벡터 값을 SVM의 내장 커널 중 하나인 RBF 커널을 이용하여 관계 추출을 수행하였다.

이상으로 본 연구에서 활용하는 다양한 접근법에 대해서 설명하였다. 관계 추출 시스템을 개발함에 있어 어느 한 가지 방법론만을 가지고 접근하는 것이 아니라 대상으로 삼는 컬렉션에 따라서 다양한 자질의 조합이 필요하다. 따라서 본 연구에서는 구문 트리, 어휘 개념 정보, 그리고 PAS 패턴과 같은 다양한 정보를 사용할 수 있는 기반을 제공함으로써 처리 대상이 되는 컬렉션에 적합한 관계 추출 시스템을 구성할 수 있도록 하였다.

4. 시스템 구성

이 장에서는 본 연구에서 구성한 시스템을 세부적으로 설명하고 그 특징 및 확장성에 대해서 다룬다. 아키텍처에 대한 세부 설명은 다음과 같다. 우선 본 연구의 시스템에 포함된 언어 분석기는 구문분석기, 기저구 인식기(CRF Chunker), 품사 태거(CRF POS-tagger) 등이다. 구문분석기는 Charniak Parser[1], Stanford Parser[2], Enju Parser[3]를 도입하여 시스템에 이식시켰다. 또한 다양한 형태의 언어자질 추출을 위해서 기저구 인식기 및 품사 태거도 독립적으로 개발하여 시스템에 결합하였다.



[그림 8] 시스템 구성도

문장에 대한 구문 분석 결과는 구문 자질 추출기로 입력되며 여기서 미리 표시된 개체의 문장 내 위치와 주변 구문 정보를 이용하여 구문트리 가지치기(parse tree pruning)를 수행한다. 그리고 predicate-argument 관

1) http://www.cs.brown.edu/people/ec/#software
2) http://nlp.stanford.edu/software/lex-parser.shtml
3) http://www-tsujii.is.s.u-tokyo.ac.jp/enju

계를 분석하여 PAS 패턴을 생성한다. 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 개체의 고유한 특성 정보가 제공되면 이를 자질화하여 관계추출에 적용하기 위한 모듈이다.

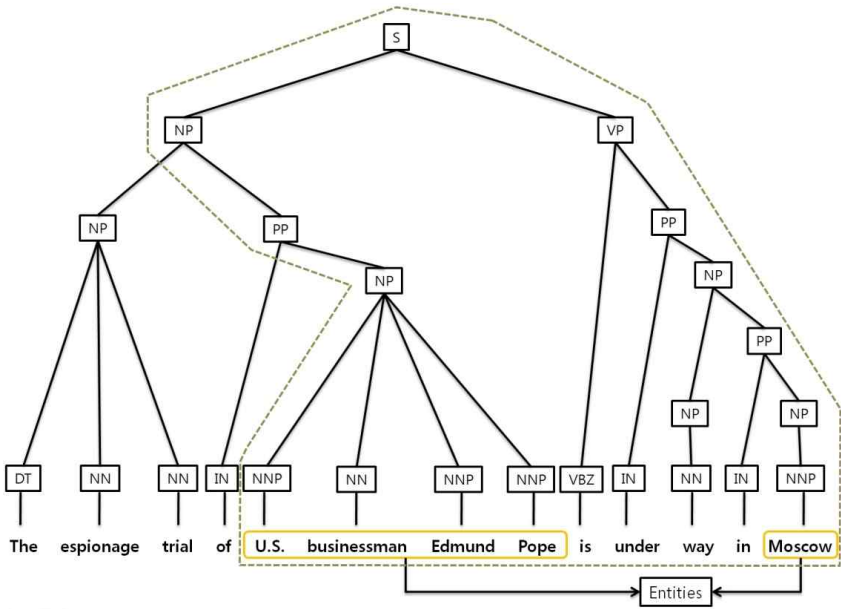
구문트리 가지치기에 대해서 좀 더 부연설명하자면, 이것은 [20]에서 소개된 개념으로서 커널 함수의 유사도 측정 성능을 향상시키고 관계 추출을 위한 학습 시에 불필요한 문맥 정보들을 제외시키기 위해서 구문 트리의 일부분만을 남기고 나머지는 제거하는 작업을 말한다. [20]에서는 구문 트리 내에서 제거되는 구문 정보 및 단말 노드 정보의 종류에 따라 5 가지의 가지치기 방법을 제시하였고, [29]에서는 이를 확장하여 총 7 가지의 가지치기 방법을 고안하였다. 다음 표에 총 7 종류의 가지치기 방법에 대한 세부 내용을 정리하였다.

[표 6] 구문 트리 가지치기 방법 [20, 28, 29]

Tree Pruning Methods	Details	F1 Ranking
Minimum Complete Tree(MCT)	구문 트리 내에서 두 개체를 포함하고 있는 최소 완전 부분 트리	7
Path-enclosed Tree(PT)	**두 개체를 연결하는 최소 경로 내에 포함된 부분 트리**	**1**
Chunking Tree(CT)	PT 에서 기저구(Base Phrase) 및 품사정보를 제외한 모든 내부 노드들을 제거한 트리	5
Context-sensitive PT(CPT)	PT 에서 좌측 개체의 좌측 노드 하나, 우측 개체의 우측 노드 하나를 추가한 트리	3
Context-sensitive CT(CCT)	CT 에서 좌측 개체의 좌측 노드 하나, 우측 개체의 우측 노드 하나를 추가한 트리	6
Flattened PT(FPT)	PT 에서 부모 노드 및 자식 노드가 각각 1개 뿐인 노드들을 제거(품사 노드 제외)	2
Flattened CPT(FCPT)	CT 에서 부모 노드 및 자식 노드가 각각 1개 뿐인 노드들을 제거(품사 노드 제외)	4

위 표에서는 각 가지치기에 대한 설명과 더불어 ACE 2003 검증 컬렉

션을 이용하여 합성곱 구문 트리 커널로 실험한 결과의 성능 순위를 함께 보여준다. 보는 바와 같이 PT 방법이 가장 좋은 성능을 보이고 있다. 따라서 전체 문장에서 좌, 우 개체를 포함하는 문장 일부의 구문 구조가 두 개체 간 관계를 추정하는데 있어 가장 좋은 단서가 될 수 있음을 알 수 있다.



[그림 9] 전체 구문 트리 및 PT 기반 가지치기된 부분 트리

위의 결과에 따라, 본 연구에서는 기저 시스템으로서 이들 방법 중에서 가장 성능이 높은 "경로 포함 트리(Path-enclosed Tree: PT)" 기법을 채택하였다. [그림 9]는 ACE 2003 컬렉션에 출현한 "TheespionagetrialofU.S.businessmanEdmundPopeisunderwayinMoscow" 라는 문장에 대한 구문 분석 트리와 PT 방법을 사용했을 때 생성되는 부분 트리를 점선으로 표시하였다. 그림에서 보는 바와 같이, PT는 두 개체와 그 개체 사이에 존재하는 단어들로 구성된 부분 문장들에 대한 구문적 구조 정보만을 남기고 나머지 정보들은 제거하게 된다. 따라서 위의 원본 문장이 "U.S.businessmanEdmundPopeisunderwayinMoscow"로 축소되므

로, 두 개체의 관계를 분류할 때, 불필요한 구조 자질들의 영향을 최소화하게 된다. 이 방법의 특징은 두 개체 사이에 존재하는 상호작용을 표현하는 어휘자질과 이를 아우르는 구문자질을 동시에 집중적으로 적용할 수 있다는 것이다.

본 연구에서 개발된 시스템은 두 가지 기계학습 모델을 기반으로 구축되었다. 우선 SVM 기반 관계추출을 위해서는 libsvm 2.89[4][21]를 자체적으로 확장하여 여기에 구문 트리 커널을 이식시켰다. 결과적으로 합성곱 구문 트리 커널, 시맨틱 구문 트리 커널, PAS 패턴 유사도 커널을 제공함으로써 각 커널의 조합을 이용한 혼합 커널을 함께 이용할 수 있도록 하였다. 시맨틱 구문 트리 커널은 워드넷 기반의 어휘 중의성 해소 시스템을 수반하므로 워드넷 자원을 추가적으로 이용한다. 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++[5]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.

---

4) http://www.csie.ntu.edu.tw/~cjlin/libsvm/
5) http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

5. 실험 및 분석

이 장에서는 본 연구에서 제안한 방법론의 성능을 파악하기 위해서 다양한 말뭉치 기반의 실험 결과를 제시한다. 특히 단백질 간 상호 작용 식별(Protein-Protein Interaction Identification)뿐만 아니라 다중 분류체계기반의 단백질 간 상호작용 분류(Protein-Protein Interaction Classification) 실험도 수행하였다. 시맨틱 구문 트리 커널의 객관적인 성능 수준을 알아보기 위해서, 기존의 대표적인 연구 결과에서 사용된 Five PPI Corpora를 이용하였으며, PPIC 실험을 위해서는 약 25 종류의 연관관계가 포함된 BioInfer 말뭉치를 이용하였다. 본 연구에서 파악된 바로는 이는 최초로 시도되는 실험이며 따라서 비교 대상이 되는 선행 연구가 없다. 또한 과학기술문헌에 존재하는 핵심개체 간의 연관관계 자동 추출(Place-Location-Organization-Technical_term Relation Extraction) 실험을 위해서 KREC(Kisti Relation Extraction Collection)를 자체적으로 구축하고 성능 평가를 수행하였다.

5.1 실험 대상 말뭉치

본 절에서는 연구에서 사용한 실험 말뭉치를 소개한다. 말뭉치는 크게 단백질 상호작용 식별 성능 측정용 말뭉치와 분류 성능 측정용 말뭉치, PLOT(Place-Location-Organization-Technical_term) 간의 연관관계 추출 말뭉치로 나뉜다. 아래 절에서 본 연구의 실험에서 사용한 세 종류의 말뭉치에 대해서 소개한다.

5.1.1 단백질 상호작용 식별 말뭉치

단백질 상호작용 식별 실험은 [22]에서 구성한 5 가지의 PPI 말뭉치를 대상으로 수행하였다. 통상적으로 “Five PPI Corpora[6)]”라고 불리는 이 말뭉치 집합은 AIMed[23], BioInfer[24], HPRD50[25], IEPA[26] 그리고

6) http://mars.cs.utu.fi/PPICorpora/eval-standard.html

LLL[27]을 단일화된 XML 형식으로 변환해 놓은 컬렉션으로서, 현재 단백질 간 상호작용 추출 기법의 준거 평가 컬렉션으로 활용되고 있다.

[표 7] 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

특정 문장에 2개 이상의 단백질 이름이 출현하고, 그 들 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 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 연구에서는 [24]에서 개발한 BioInfer[7] 말뭉치를 활용하였다. BioInfer는 생의학 분야 정보 추출을 위한 통합 말뭉치로서 이 분야 연구 연구의 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 연관 관계를 지정해 놓은 집합이다. 본 연구에서는 위의 여러 가지 개체들 중에서 단일 단백질(individual protein)에 대해서만 실험을 수행하였다. 아래 표는 실험에 사용한 BioInfer 말뭉치의 상호작용 종류와 각 관계별 인

7) http://mars.cs.utu.fi/BioInfer

스턴스 개수를 보여준다.

[표 8] BioInfer 말뭉치내의 단백질 상호작용 종류 및 인스턴스 개수

<table>
<tr><th colspan="6">Relations and #Instances</th></tr>
<tr><td rowspan="15">Causal (1,658)</td><td colspan="5">56</td></tr>
<tr><td rowspan="14">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="3">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 8]에서 보듯이 연구에서 사용한 BioInfer 부분 집합 내에서의 개별 상호작용은 계층적으로 구성되어 있으며, 최상위 상호작용의 종류는 총 6가지이다.[8] 말단 상호작용 기준으로 "Assembly"가 가장 많은 788개의 인스턴스를 가지고 있고, "Full-Stop"이 겨우 1개의 인스턴스를 가지고 있

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

다. 전체적으로 볼 때, 인스턴스의 편중현상이 심하고 고르지 않은 분포를 보이고 있다. 따라서 자동 분류 시에 자료 희귀성(data sparseness) 문제가 발생할 가능성이 매우 높다.

[24]에서 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.[9] 따라서 만일 이 관계 온톨로지를 하나의 트리 형태로 본다면, 이 트리는 위 [표 8]에서 보인 관계 유형 클래스(relation type class)를 나타내는 노드와 관계 술어 노드로 구분될 수 있다. [표 9]에서 이 두 종류의 노드들이 모두 포함된 BioInfer의 관계 온톨로지에 대해서 수준별로 각각의 노드 개수를 나타내었다.

[표 9] 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

위 표에서 알 수 있듯이 BioInfer의 관계 온톨로지는 총 6개의 수준에 96개의 노드가 존재하며, 이중 관계 유형 클래스는 총 28개이고, 관계 술어는 68개이다. 실험에서는 [표 8]에서 나타낸 관계 집합을 기준으로, 전체 25가지의 전체 관계 유형 클래스와 최상위 수준부터 시작하여 3 수준까지의 관계 집합으로 구성된 3가지 말뭉치를 포함하여 도합 4가지 말뭉치를 이용하여 실험을 수행하였다. 비록 [표 8]에서는 나타내지 않았으나 실험에서는 관계 술어도 포함된 관계 집합을 사용하였다. 아래 표는 이들 각각의 관계집합 및 개수를 보여준다.

9) 예를 들어, [표 8]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 10] 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 클래스 만으로 구성된 관계집합	25

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

5.1.3 PLOT 간 연관관계 추출 말뭉치

본 연구에서는 PLOT(Place-Location-Organization-Technical_term) 간 연관관계 추출 대상을 과학기술문헌 중에서 최근 들어 활발히 언급되고 있는 녹색기술 관련 분야로 한정하였다. 녹색기술 관련 문헌은 과학기술 뉴스와 NDSL에서 보유하고 있는 해외학술지로 선정하였다.

PLOT 간 연관관계 추출의 성능 평가를 위한 테스트컬렉션 KREC(Kisti Relation Extraction Collection)를 다음과 같이 구축하였다. 우선 과학기술 뉴스는 해외사이트[10]에서 녹색기술 분야 중에서 2000년도 이후의 문서에서 문서크기가 상위 80% 이상에 해당되는 문서들을 연도별로 임의로 선정하여, 전체 11,185건을 수집하였다. 해외학술지는 SCI급, 인용지수, 초록크기 등 여러 가지 요소들을 조합하여 수집하였다. 첫째, 해외학술지 중에서 인용지수(Impact Factor)를 기준으로 상위 50종을 우선 선별하였다. 둘째, 동일한 종에서 개별 초록의 크기가 평균초록 크기의 90%이상인 문서를 선정하였다. 셋째, 발행연도가 2000년 이후인 최신 문서를 선정하였다. 넷째, 선정된 종에서 종별로 각 25%의 문서를 선정하여 최종적으로 10,310건의 문서를 수집하였다. 선정된 문서는 [표 11]의 DTD를 갖는 XML문서 형식으로, [표 12]에서의 연관관계를 사용하여 태깅작업이 수행된다.

10) http://www.eurekalert.org/

[표 11] KREC DTD 정보

```
<?xml version="1.0"    encoding="EUC-KR"?>

<!-- Overview:
          DOC    := TEXT, NRLIST
                    [attrs: did]
          TEXT   := TI, AB
               TI    := S+
               AB   :=    S+
               S   := (PCDATA|NE)*
               NE   := PCDATA
                    [attrs: edi, co_ref, class,    nn]

          NRLIST := NR*
               NR := PCDATA
                    [attrs: rid, eid_1, eid_2,    rel, psv]
-->
<!-- DOC element consists of TEXT elements    and NRLIST elements that is
relations within TEXT elements. -->
<!ELEMENT DOC (TEXT, NRLIST)>
<!ATTLIST DOC
          did       CDATA    #REQUIRED>    <!-- document identifier -->

<!-- TEXT element consists of TI    element(title) and AB element(abstract).
<!ELEMENT TEXT (TI, AB)>

<!-- TI element consists of S    elements(sentence) -->
<!ELEMENT TI (S+)>

<!-- AB element also consists of S    elements -->
<!ELEMENT AB (S+)>

<!-- S element consists of NE    elements(Named Entity that is an science
```

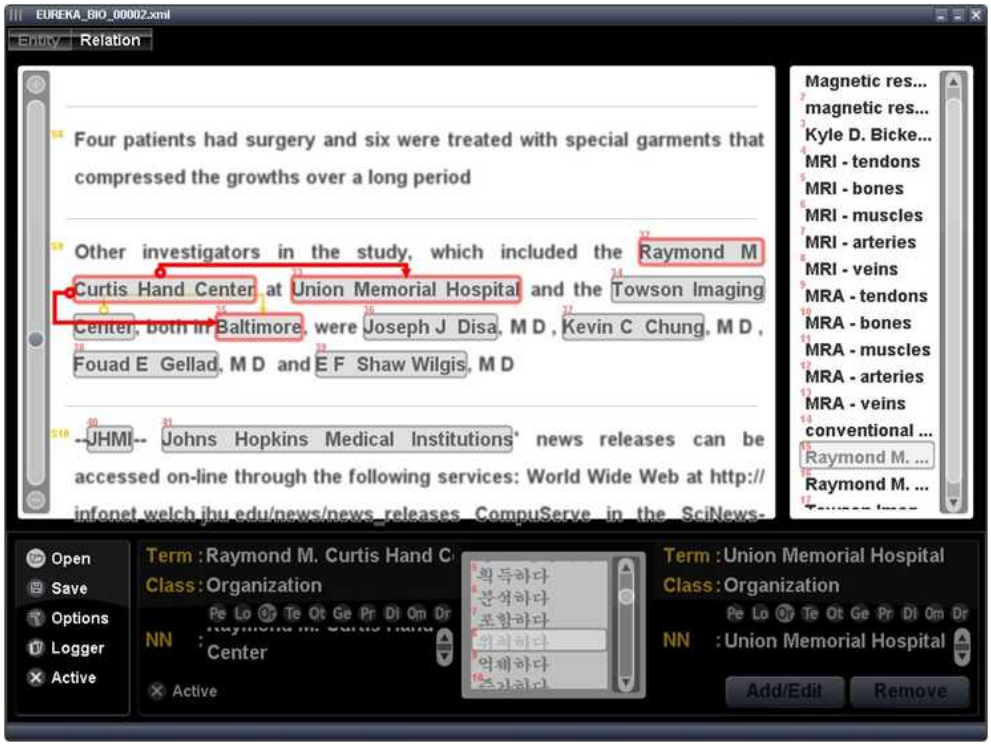
```
and technology core entity.
<!ELEMENT S (#PCDATA|NE)*>

<!-- NE element has information of real entity -->
<!ELEMENT NE (#PCDATA)>
<!ATTLIST NE
        eid     CDATA   #REQUIRED      <!-- an entity identifier -->
        co_ref  CDATA   #IMPLIED       <!-- a coreference identifier
-->
        class   CDATA   #REQUIRED      <!-- a class of an entity -->
        nn      CDATA   #REQUIRED>     <!-- a normalized name -->

<!-- NRLIST elements which is a collection of relations is consists of NR
elements. -->
<!ELEMENT NRLIST (NR*)>

<!-- NR element has relation information of between entities -->
<!ELEMENT NR EMPTY>
<!ATTLIST NRLIST
        rid     CDATA   #REQUIRED      <!-- a relation identifier -->
        eid_1   CDATA   #REQUIRED      <!-- the first entity id -->
        eid_2   CDATA   #REQUIRED      <!-- the second entity id -->
        rel     CDATA   #REQUIRED      <!-- a relation class -->
        psv     (0 | 1) #REQUIRED>     <!-- active (0) or passive (1) -->
```

[표 12] KREC 연관관계 정보

No	Relation Name	No	Relation Name
1	활용/적용하다(use, utilize, apply, employ)	21	도와주다(help, assist, aid)
2	변화/변경하다(change, alter, modify)	22	사업적 관계(business)
3	야기/유발/자극하다(induce, stimulate, cause, have, get,	23	혈연적 관계(family)

	make)		
4	창조/생산/생성하다(produce, make, create)	24	고용하다(hire, engage, employ)
5	획득하다(get, acquire)	25	관리하다(manage, deal, care, handle)
6	분석하다(analyze, analyse, study, examine, canvass, canvas)	26	일원/회원관계(membership)
7	포함하다(include)	27	설립하다(establish, set up, found, launch)
8	위치하다(locate, place, site)	28	근무하다(be on duty, work for, serve)
9	억제하다(suppress, stamp down, inhibit, subdue, conquer, curb)	29	연구하다(research)
10	증가시키다(increase)	30	전달하다(convey, transmit, communicate)
11	방지하다(prevent, keep)	31	해결하다(solve, work out, figure out, puzzle out, lick, work)
12	치료하다(remedy, curative, cure, therapeutic)	32	발견하다(discover)
13	활성화하다(activate)	33	결합연결(connection)
14	영향을 주다(affect, impact, bear upon, bear on, touch on, touch)	34	해체분해(disjoint)
15	의존하다(depend)	35	강화하다(strengthen)
16	응답하다(react, respond)	36	약화시키다(weaken)
17	동일하다(equal, be)	37	감소시키다(reduce)
18	부분(part, portion, component part, component)	38	결과/원인관계(cause/effect)
19	소유하다(own, have, possess)	39	조절/조정(regulate)
20	제공하다(yield, give, afford)		

테스트컬렉션 구축은 전문가 2인에 의해 수행되었고, 서로 교차하여 검토 비교하여 오류를 최소화 하였다. 테스트컬렉션 구축 시 발생되는 철자오류 및 태깅오류 등을 방지하고, 작업 속도를 높이기 위하여 [그림 10]과 같은 테스트컬렉션 구축지원도구를 사용하였다.



[그림 10] 테스트컬렉션 구축 도구

테스트컬렉션 구축도구는 문장 분리 및 합병, 핵심개체 지정 및 취소, 핵심개체 추천, 분류코드 지정, 오류 검증 등의 기능 등을 제공하고, 완성된 문서를 테스트컬렉션의 DTD에 맞는 XML문서로 저장한다. [표 13]은 본 연구에서 구축한 테스트컬렉션의 통계 정보이다.

[표 13] KREC 통계

문서	문장	핵심개체	연관관계
1,090	14,341	22,125	2,441

5.2 실험 대상 시스템

본 연구에서 수행한 실험은 총 세 가지의 매개변수가 수반되며 이 값의 지정 방법에 따라서 성능이 차이가 날 수 있다. 세부적인 내용은 아래 표와 같다.

[표 14] 실험 대상 시스템 설정 종류 및 개수

매개변수	설명(details)	범위 (range)		설정 개수
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**

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 3]에서 설명한 어휘 개념 추상화 수준 지정 인자는 총 4가지로 설정하였다. 따라서 이들 세 가지 매개변수를 모두 적용하면 총 280가지의 설정이 도출된다. 본 연구에서는 이 설정 집합 각각에 대해서 10겹 교차평가를 수행하여 성능을 측정한다.

5.3 성능 측정 기준

본 연구에서 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 기반으로 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 15] 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 15]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 나타내

고 있다.

[표 16] 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%

[표 16]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.

[표 17] 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

마지막으로 [표 17]은 본 연구에서 제안한 시스템의 성능과 기존 연구 결과와의 비교 분석 자료이다. 동일한 조건에서의 비교를 위해서 [13, 14]에서와 동일한 정규화 인자를 적용하였으며, 트리커널 소멸인자는 모두 0.4로 고정시켰다. 표에서 보는 바와 같이 본 연구에서의 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_A와 PROT_B의 상호작용은 ″inhibits″라는 동사로 표현되는 반면에 PROT_B와 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_A와 PROT_C는 직접적인 연관이 없음에도 불구하고 두 단백질이 서로 상호작용을 한다는 판단을 내릴 수 있다. 따라서 구문트리커널을 적용함에 있어서 가지치기는 필수적이며, 선행 연구인 [14]에서는 구문트리 커널을 포함한 다양한 커널을 통합 적용했음에도 불구하고 성능 향상이 미비했던 이유도 이 가지치기 미적용 때문이라고 추정할 수 있다. 두 번째로 구문트리 커널의 소멸인자에 의한 성능 최적화이다. 소멸인자는 기본적으로 구문 트리 커널 함수 내에서 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 10]에서 나타낸 서로 다른 관계 집합을 대상으로 개별 실험을 수행하였다.

[표 18] 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
Average	SPTK	-	-	-	84.55	74.75	58.42	65.54

다중 분류 작업인 단백질 간 상호작용 분류 실험에서는 [표 18]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 8]에서 지적한 것과 같이 각 구성 관계별 인스턴스 편중현상이 그 원인이다.

시맨틱 구문 트리 커널의 성능 개선 정도를 보다 면밀히 살펴보기 위해

[표 19] 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%

서 [표 16]에서 제시한 것과 같이 거시평균기반 F 점수 기준으로 상위 30등 이내의 시스템에 대한 분포를 분석하여 [표 19]에 나타내었다.

모든 관계 집합에 대해서 시맨틱 구문 트리 커널이 높은 점유율을 보이고 있다. 특히 RCO 집합에 대해서는 30개의 상위 시스템 중에서 25개가 시맨틱 구문 트리로 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 20]에서는 PTK의 최고 성능을 나타내는 동일한 설정을 기준으로 두 커널의 성능 비교를 하였다.

[표 20] 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

[표 20]에서 알 수 있듯이, 그 차이는 현저하지 않으나 대부분의 관계 집합에서 SPTK가 더 나은 성능을 나타내고 있다. 성능 개선의 차이가 예상과는 다르게 차이가 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.

5.6 PLOT 간 연관관계 추출 실험

본 절에서는 녹색 기술 관련 문서에 존재하는 PLOT 간 연관관계 추출 실험의 결과를 보이고 이를 분석한다. KREC에 존재하는 연관관계의 수는 39개이지만, 본 실험에서는 비슷한 성질의 연관관계를 통합하여 총 7개의 연관관계를 구성하여 실험을 진행하였다.

[표 21] PLOT 간 연관관계 추출 성능

커널 종류	mi-F1	Precision	Recall	ma-F1
PAS 패턴 유사도 커널	**64.60**	41.01	28.58	33.69
구문트리 커널	**68.78**	45.23	32.89	38.09
혼합 커널	**74.72**	47.27	38.33	42.33

[표 21]에서 보는 것처럼 구문트리 커널과 PAS 패턴 유사도 커널을 단독으로 수행했을 때는 구문트리 커널의 성능이 PAS 패턴 유사도 커널보다 더 좋은 것을 확인할 수 있다. 하지만 구문트리 커널 단독으로 사용하는 것보다는 PAS 패턴 유사도 커널을 결합하여 혼합 커널을 구성하였을 경우에 훨씬 더 나은 성능을 보이는 것을 확인할 수 있다. 따라서 PAS 패턴 유사도 커널은 기존의 다른 커널과 결합하여 사용되는 경우에는 상

당한 성능 향상을 일으킨다는 사실을 확인할 수 있다.

표에서 mi-F1보다 ma-F1의 성능이 떨어지는 것은 테스트컬렉션에 사용된 7가지 관계의 인스턴스의 분포 때문이다. 7개의 연관관계 중에서 인스턴스의 개수가 100개 미만인 연관관계가 3개 존재하는데, 이 3개의 연관관계는 전체 학습 인스턴스의 단지 6.3%(2,441 중에서 152개)만을 차지하고 있다. 다시 말해서 나머지 4개의 관계가 학습 집합의 93.7%를 차지한다는 사실로 그 편중현상이 매우 심함을 알 수 있다. 따라서 관계 추출의 정답과 오답만을 검사하는 mi-F1과는 달리 ma-F1은 각 관계 별 정답과 오답을 따로 검사한 후에 그것의 평균 값을 성능으로 취하기 때문에 관계별 인스턴스 개수의 분포가 고르지 못한 경우에는 성능이 낮게 나오기가 쉽다. 이는 관계 인스턴스가 적은 연관관계의 경우에 학습 집합의 부족 현상이 발생하기 때문이다. 따라서 향후에는 7가지 관계의 인스턴스 개수를 고르게 분포시켜서 실험을 다시 수행해볼 필요가 있다.

6. 결론 및 향후 연구

본 연구에서는 기존 구문 트리 커널이 두 문장의 구문적 유사도에만 중점을 두어서 커널 수치를 계산하던 것을 확장하여 어휘 의미적 유사도를 함께 고려하도록 설계한 시맨틱 구문 트리 커널을 제안하였다. 이를 위해서 문맥 및 워드넷 기반의 어휘 중의성 해소 시스템과 이 시스템의 출력으로 도출되는 어휘 개념(WordNet synset)의 추상화를 통해서 바이오 문헌의 단백질 간 상호작용 식별과 분류, 녹색기술 문헌의 PLOT 간 연관관계 추출 분야에 각각 적용하여 성능 향상을 입증하였다.

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 커널값이 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 연구의 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 비교적 난이도가 낮은 단백질 간 상호작용 식별 문제보다도 상호작용 분류에 더욱 효과적임을 알 수 있었다. 더불어 시맨틱 구문 트리 커널의 어휘 추상화 인자도 소멸인자와 더불어 성능향상에 중요한 요소임을 증명하였다.

끝으로 문장 내에 존재하는 predicate과 argument 간의 의미적 구조(semantic structure) 관계를 활용하는 PAS 패턴 유사도 커널 또한 기존

의 다른 접근법과 결합됨으로써 상당한 성능 개선 효과를 이끌어 내는 것을 확인하였다. 이로써 적용 도메인에 맞게 혼합 커널을 적절히 구성하는 것은 관계 추출의 성능 향상을 위하여 반드시 필요한 요소임을 확인할 수 있었다.

향후 연구 방향으로는 앞에서도 잠시 언급하였으나 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 언어분석시스템 도입이 필요하다. 이와 더불어 기존 커널과 본 연구에서 개발한 커널을 결합한 혼합 커널(composite kernel) 개발을 생각해 볼 수 있다. [13]에서 제안한 의존 그래프 커널 등과 같은 기존 커널과의 밀결합을 통해서 또 다시 새로운 커널을 구성할 수 있을 것이다.

[성과 달성 기술 요약표]

<table>
<tr><td>달성
목표</td><td>○ F1 기준으로 관계 추출률 80% 확보
- 단백질 간 상호작용 인식(Protein-Protein Interaction Identification), 단백질 간 상호작용 분류(Protein-Protein Interaction Classification), 과학기술 핵심개체 간 연관관계 추출(Place-Location-Organization-Technical_term Relation Extraction)과 같이 다양한 분야의 관계 추출 응용 도메인이 있음. 본 연구에서는 다양한 도메인에 적용할 수 있는 범용 관계추출 기술 개발을 목적으로 하기 때문에 전체적인 관계 추출률을 평균 80%로 설정함.</td></tr>
<tr><td>달성
결과</td><td>○ 목표 성능 달성
<table>
<tr><td rowspan="2">검증
컬렉션</td><td colspan="5">PPII</td><td>PPIC</td><td rowspan="2">KREC</td></tr>
<tr><td>AIMed</td><td>BioInfer</td><td>IEPA</td><td>HPRD50</td><td>LLL</td><td>BioInfer</td></tr>
<tr><td>성능 (F_1)</td><td>89.33</td><td>89.00</td><td>79.17</td><td>85.22</td><td>88.48</td><td>84.55</td><td>74.72</td></tr>
</table>
- 7개의 테스트컬렉션을 대상으로 실험한 결과 <u>평균 관계 추출률 84.35%를 달성</u>함으로써 목표 성능에 도달
- 최신 관계추출모델인 Convolution Parse Tree Kernel(CPTK)에 의미 정보를 결합한 Semantic Parse Tree Kernel(SPTK), 그리고 Predicate-Argument Structure(PAS) 패턴 유사도 커널을 결합한 혼합 커널 기술 개발 및 적용

○ 국제적으로 널리 통용되는 <u>다양한 분야의 테스트컬렉션을 활용</u>하여 성능 평가를 수행함으로써 개발된 관계추출엔진의 <u>객관적 기술 수준 파악</u>
- <u>바이오 분야</u>의 단백질 간 상호작용 인식 성능 평가를 위해서 Five PPI Corpora(AIMed, BioInfer, IEPA, HPRD50, LLL)를 활용
- <u>바이오 분야</u>의 단백질 간 상호작용 분류 성능 평가를 위해서 BioInfer Corpus를 활용</td></tr>
</table>

	- 녹색기술 분야의 PLOT(인명, 지명, 조직명, 기술명) 간 연관관계 자동 추출에 대한 성능 검증을 위해서 KREC를 활용

[참고문헌]

[1] http://en.wikipedia.org/wiki/Protein-protein_interaction, Wikipedia, 2009

[2] Zhou D., He Y., "Extracting interactions between proteins from the literature," Journal of Biomedical Informatics, v.41, pp.393-407, 2008

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

[4] Gondy L., Hsinchun C., Martinez Jesse D., "A shallow parser based on closed-class words to capture relations in biomedical text," J. Biomed. Informatics, vol.36, no.3, pp.145-158, 2003

[5] Temkin Joshua M., Gilder Mark R., "Extraction of protein interaction information from unstructured text using a context-free grammar," Bioinformatics, vol.19, no.16, pp.2046-2053, 2003

[6] Nikolai D., Anton Y., Sergei E., Svetalana N., Alexander N., llya M., "Extracting human protein interactions from MEDLINE using a full-sentence parser," Bioinformatics, vol.20, no.5, pp.604-611, 2004

[7] Blaschke C., Andrade M., Ouzounis C., Valencia A., "Automatic extraction of biological information from scientific text: protein-protein interactions," Proc. Int. Conf. Intell. Syst. Mol. Biol., pp.60-67, 1999

[8] Ono T., Hishigaki H., Tanigam A., Takagi T., "Automated extraction of information on protein-protein interactions from the biological literature," Bioinformatics, vol.17, no.2, pp.155-161, 2001

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

[10] Andrade Miguel A., Valencia A., "Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families," Bioinformatics, vol.14, no.7, pp.600-607, 1998

[11] Marcotte Edward M., Xenarios I., Eisenberg D., "Mining literature for protein-protein interactions," Bioinformatics, vol.17, no.4, pp.359-363, 2001

[12] Craven M., Kumlien J., "Constructing biological knowledge bases by extracting information from text sources," Proceedings of the 7th International conference on intelligent systems for molecular biology, Heidelberg, Germany, pp.77-86, 1999

[13] Airola A., Pyysalo S., Bjorne J., Pahikkala T., Ginter F., Salakoski T., "All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning," BMC Bioinformatics, vol.9, no.S2, 2008

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

[15] Vishwanathan S. V. N., Smola A. J., "Fast Kernels for String and Tree Matching," Advances in Neural Information Processing Systems, MIT Press, vol.15, pp.569-576, 2003

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

[17] Moschitti A., "Making tree kernels practical for natural language learning," Proceedings of EACL'06, Trento, Italy, 2006

[18] Lesk M., "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, pp.24-26, 1986

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

[20] 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

[21] Chang, C.-C. and Lin, C.-J., LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[22] Pyysalo S., Airola A., Heimonen J., Björne J., Ginter F., Salakoski T., "Comparative analysis of five protein-protein interaction corpora," BMC Bioinformatics, vol.9, no.S6, 2008

[23] Bunescu R., Ge R., Kate R., Marcotte E., Mooney R., Ramani, A., Wong, Y., "Comparative Experiments on Learning Information Extractors for Proteins and their Interactions," Artif. Intell. Med., Summarization and Information Extraction from Medical Documents, vol.33, pp.139-155, 2005

[24] Pyysalo S., Ginter F., Heimonen J., Bjorne J., Boberg J., Jarvinen J., Salakoski T., "BioInfer: a corpus for information extraction in the biomedical domain," BMC Bioinformatics, vol.8, no.50, 2007

[25] Fundel K., Küffner R., Zimmer R., "RelEx - Relation extraction using dependency parse trees," Bioinformatics, vol.23, pp.365-371, 2007

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

2002
[27] Nedellec C., "Learning language in logic - genic interaction extraction challenge," Proceedings of LLL'05, pp.31-37, 2005
[28] 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
[29] Zhang, M., GuoDong, Z., Aiti, A., "Exploring syntactic structured features over parse trees for relation extraction using kernel methods," Information Processing and Management, v.44, pp.687-701, 2008
[30] Culotta, A., Sorensen, J., "Dependency Tree Kernels for Relation Extraction," Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, 2004
[31] 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
[32] Bunescu, R. C., Mooney, R. J., "Subsequence Kernels for Relation Extraction," Advances in Neural Information Processing Systems, 2006
[33] Zhao, S. B., Grishman, R., "Extracting Relations with Integrated Information Using Kernel Methods," ACL-2005, 2005
[34] 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

[35] GuoDong Z., Su J. Zhang J. and Zhang M., “Exploring various knowledge in relation extraction,” ACL’2005, pp. 427-434, 25-30 June, Ann Arbor, Michgan, USA, 2005

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

[ISBN 978-89-6211-581-9]

정창후 • 최성필 • 최윤수 • 전홍우
정도헌 • 홍순찬 • 조민희 • 성원경

의미 정보를 활용한 관계 추출 시스템 개발 및 성능 평가

2010년 10월 27일 인쇄
2010년 10월 27일 발행

발 행 처

한국과학기술정보연구원

대전광역시 유성구 어은동 52-11
㊞ 305-806
전화 : 042-869-1004
등록 : 1991년 2월 12일 제 5-259호

발행인
박 영 서

인쇄처
㈜ 미래미디어