구문 트리 가지치기 및 소멸 인자 조정을 통한 트리 커널 기반 단백질 간 상호작용 추출 성능 향상

(Performance Enhancement of Tree Kernel-based Protein-Protein Interaction Extraction by Parse Tree Pruning and Decay Factor Adjustment)

최 성 필 † 최 윤 수 † 정 창 후 † 맹 성 현 ††
(Sung-Pil Choi) (Yun-Soo Choi) (Chang-Hoo Jeong) (Sung-Hyon Myaeng)

요 약 본 논문에서는 기존의 연구에서 시도되었던 것과는 달리, 복잡하고 추출하기가 어려운 다양한 형태의 자질 및 단서 정보가 필요 없는 합성곱 구문 트리 커널 기반의 단백질 간 상호작용 추출 기법을 소개한다. 이 기법의 특징은 단백질 이름 쌍을 포함한 상호작용 포함 후보 문장에 대한 구문 트리만을 이용하여 추출을 시도한다는 것이며 부가적인 자질이나 커널 함수가 불필요하다는 장점이 있다. 이를 기반으로 본 논문의 연구 성과는 다음과 같다. 첫째, 단백질 간 상호작용 추출에 있어서 구문 트리 커널을 적용할 경우 불필요한 문맥 정보를 효과적으로 제거하는 구문 트리 가지치기 작업이 필수적임을 기존 연구 결과와의 성능 비교로써 증명한다. 둘째, 동일한 학습 조건에서 구문 트리 커널의 소멸 인자(decay factor)는 평활 인자(smoothing factor)로서 중요한 역할을 하며, 성능 변화의 핵심 요소임을 보인다. 특히 학습 집합의 규모에 따라서 소멸인자가 성능에 미치는 영향력이 상이한 패턴으로 나타남을 제시하였다. 결론적으로 기존의 최신 연구결과로서 주장한 "단일 커널보다 혼합 커널의 성능이 더 뛰어나다"라는 가설이 항상 성립하는 것은 아니라는 것을 합성곱 구문 트리 커널 단독으로 적용하여 높은 성능을 나타냄으로써 보여주었다. 동일한 조건으로 수행한 실험에서 기존의 두 연구 결과에 비해 19.8%, 14%의 성능 개선을 나타내었다.

키워드 : 단백질간 상호 작용 추출, 커널 기법, 합성곱 구문 트리 커널, 정보 추출, 관계 추출

Abstract This paper introduces a novel way to leverage convolution parse tree kernel to extract the interaction information between two proteins in a sentence without multiple features, clues and complicated kernels. Our approach needs only the parse tree alone of a candidate sentence including pairs of protein names which is potential to have interaction information. The main contribution of this paper is two folds. First, we show that for the PPI, it is imperative to execute parse tree pruning removing unnecessary context information in deciding whether the current sentence imposes interaction information between proteins by comparing with the latest existing approaches' performance. Secondly, this paper presents that tree kernel decay factor can play an pivotal role in improving the extraction performance with the identical learning conditions. Consequently, we could witness that it is not always the case that multiple kernels with multiple parsers perform better than each kernels alone for PPI extraction, which has been argued in the previous research by presenting our outperformed experimental results compared to the two existing methods by 19.8% and 14% respectively.

Key words : Protein-Protein Interaction Extraction, Kernel Methods, Convolution Parse Tree Kernel, Information Extraction, Relation Extraction

† 정 회 원 : 한국과학기술정보연구원 정보기술연구실
spchoi@kisti.re.kr
armian@kisti.re.k
chjeong@kisti.re.kr
(Corresponding author임)
†† 종신회원 : 한국과학기술원 전산학과 교수
myaeng@kaist.ac.kr
논문접수 : 2009년 10월 12일
심사완료 : 2009년 12월 11일

정보과학회논문지: 소프트웨어 및 응용 제37권 제2호(2010.2)

1. 서 론

최근 들어, 생의학 분야 전문 정보(논문, 특허, 보고서 등) 기반의 정보 추출(information extraction)에 대한 연구가 활발히 진행되고 있다. 이와 관련하여, 본문에 표현된 각종 전문용어, 단백질 및 유전자 이름 등을 포괄하는 개체명(named-entity)을 식별하고 이들 간의 의미적 연관 관계를 자동으로 추출하는 연구에 대한 중요성이 매우 강조되고 있는 실정이다[1-3]. 특히 본문 내에서 단백질 간 상호작용(Protein-Protein Interaction, PPI) 정보는 생물학적 프로세스를 이해하는데 가장 기본이 되는 정보이므로 이를 자동으로 추출하는 연구는 현재도 활발하게 진행 중이다[1].

생의학 분야 문헌의 본문 내에서 단백질 상호작용 정보를 추출하는 가장 일반적인 방법은 두 가지 이상의 단백질 이름을 포함하고 있는 문장을 분석하여 그 문장이 단백질 간의 의미적 상호 작용에 대한 내용을 표현하고 있는지를 파악하는 것이다. 이러한 연구를 위해서 다양한 말뭉치들이 구축되었으며, 그 중 가장 폭넓게 사용되고 있는 말뭉치들은 AImed[4], BioInfer[5], HPRD50 [6], IEPA[7], LLL[8] 등이 있다. 특히 이들 말뭉치들을 개별적으로 분석하여 단일 형식으로 정형화시킨 통합 말뭉치도 존재한다[9]. 그림 1은 BioInfer 말뭉치에서 추출한 단백질 포함 문장과 내부 단백질 간의 상호작용을 도식화한 그림이다.

단백질 상호작용 추출 연구의 초창기에는 문장 내에서 관계 표현에 해당하는 의미적 단서를 찾기 위해서 단어 동시 출현 정보(word co-occurrences) 등과 같은 비교적 단순한 기법을 사용하였다[10]. 최근에는 고수준의 자연어 처리 기법과 기계학습 모델을 이용한 복합적인 접근 방법이 많이 활용되고 있다. 특히 지도기반 기계학습에서 좋은 성능을 나타내고 있는 커널 모델이 관계추출 분야에서도 각광을 받고 있다[1,3]. 이미 신문기사에서 인명, 지명, 기관명 등과 같은 개체 간의 관계추출 연구 분야에서는 어휘적 자질기반의 커널뿐만 아니라 구문적 자질을 활용한 구문트리 커널 등과 같은 다양한 형태의 커널이 연구되어 우수한 성능을 보여주었다[11-14]. 이러한 커널 기반의 관계추출 기법이 단백질 간 상호작용 추출에 적용이 된 것은 최근이며, 현재까지도 비교적 초기 연구단계에 머무르고 있다.

본 논문에서는 기존의 연구에서 시도되었던 것과는 달리, 복잡하고 추출하기가 어려운 다양한 형태의 자질 및 단서 정보가 필요 없는 합성곱 구문트리커널 기반의 단백질 간 상호작용 추출 기법을 소개한다. 이 기법의 특징은 단백질 이름 쌍을 포함한 상호작용 포함 후보 문장에 대한 구문트리만을 이용하여 추출을 시도한다는 것이며 부가적인 자질이나 커널 함수가 불필요하다는 장점이 있다. 그럼에도 불구하고 실험에서도 알 수 있듯이 최근에 개발된 기법들에 비해서 우수한 성능 수준을 나타내고 있다.

이 논문의 구성은 다음과 같다. 우선 2장에서는 지금까지 연구되어 온 단백질 간 상호작용 추출 기법에 대해서 소개하고 각각의 장점 및 단점을 분석한다. 이어 3장에서는 합성곱 구문트리 커널 모델에 대해서 소개하고 4장에서는 이를 기반으로 한 관계추출 프레임워크인 SINDI-REX를 설명한다. 본 논문에서 개발된 시스템의

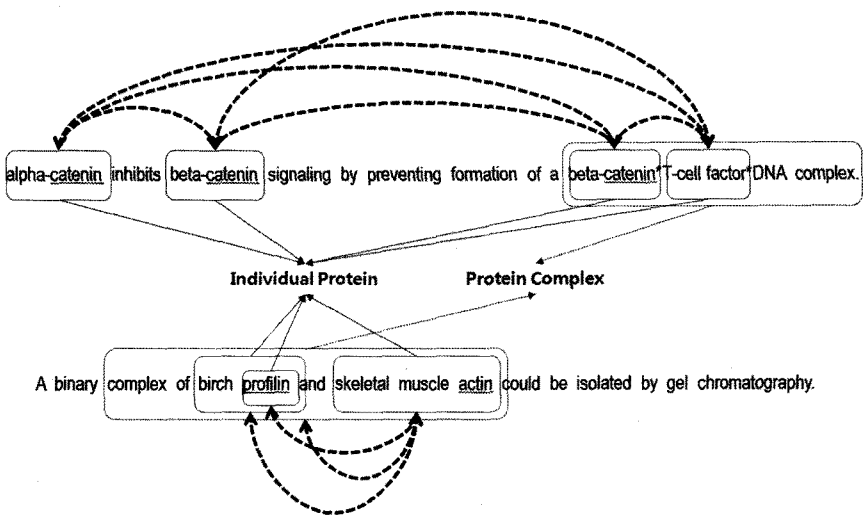

그림 1 BioInfer 말뭉치 내의 두 문장(문장번호 1, 2) 내에 포함된 단백질 및 그 상호 작용 정보

성능을 측정하기 위해서 5장에서는 위에서 언급한 5가지 컬렉션을 기반으로 수행된 실험 결과를 제시하고 분석한다. 마지막으로 6장에서 결론과 향후 연구 과제에 대해서 설명한다.

2. 관련 연구

단백질 간 상호작용 추출에 대한 연구는 그 중요성으로 인해 매우 활발하게 진행 중이며 다양한 기법들이 소개되었다. 따라서 본 논문에서는 기존의 연구 성과 중에서 가장 핵심적이고 성능 수준이 높은 것들을 중심으로 소개한다.

Blaschke et al.(1996)은 본격적으로 단백질 간 상호작용 추출에 대한 연구를 시도한 것으로 평가받고 있다[1]. 이 논문에서 저자들은 단서 동사 집합과 어휘 규칙에 의거하여 단백질 간 상호작용 포함 문장을 식별하는 방법을 채택하였다[10]. 다양한 형태의 관계표현 문장들에 대해서 심층 분석을 하고 관계표현 단서 동사 집합을 열거하였으나, 규칙 및 어휘 기반 접근 방법의 약점인 포괄성 및 규모 확장성에 대한 해결책은 제시하지 못하였다. 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[15]. 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[6].

최근 들어, 커널 기반의 단백질 간 관계추출에 대한 연구가 활발히 진행 중에 있다. 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 못 하였다[3].

한편, Miwa et al.(2009)는 단어자질 커널, 구문 트리 커널, 그래프 커널 등을 모두 적용한 혼합 커널을 구성하여 앞에서 소개한 총 5가지의 말뭉치를 대상으로 실험을 수행하였다[1]. 이 논문에서 사용한 구문 트리 축소 방법은 최소 경로 기법(Shortest Path Method)이다. 이 방법은 구문 트리를 하나의 그래프로 보고, 두 단백질 이름 노드들 간의 최소 경로에 위치하는 모든 구문 노드들을 추출하여 학습에 사용하는 방법이다. 두 단백질 사이의 관계를 직접적으로 나타내는 간략한 문맥 정보를 제공한다는 장점에도 불구하고 트리 가지치기 방법에서처럼 풍부한 구문 자질을 제공하기에는 역부족이었다. 따라서 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%).

그 외에도 이 분야에 대한 많은 연구가 이루어졌으나, 대부분 단일 말뭉치 혹은 자체 구축 말뭉치 등을 활용하여 성능 평가를 수행하였으므로, 완전한 성능 비교가 되지 않는다는 단점이 있다. 따라서 본 논문에서는 가장 최신에 발표된 연구 성과를 중심으로 객관적이고 포괄적인 성능 평가를 수행할 수 있는 두 개의 기존 연구[1,3]에 초점을 맞추었다.

문장의 구문 분석 정보를 직접적으로 활용할 수 있는 트리 커널의 효용성은 모두 인정하면서도 단백질 간 상호작용 자동추출에 있어서 이에 대한 심층적, 분석적, 실험적 연구는 현재까지 거의 수행되지 않았다. 특히 트리 커널의 가장 중요한 요소 중의 하나인 소멸 인자(decay factor)의 변화에 따른 성능 비교 연구는 전무하다. 부가적으로 대부분의 기존 연구에서 트리 커널을 사용할 때, 구문트리에 대한 가지치기를 거의 적용하지 않았다. 이는 커널이 두 구문트리를 비교함에 있어서 단백질 간 상호작용 표현 단서와는 상관없는 노이즈를 발생시키므로 성능 저하의 직접적인 원인이 된다.

본 논문에서는 Miwa et al.(2009)에서 활용한 다양하고 복잡한 다중 커널 혹은 다중 언어분석기 없이도 합성곱 구문 트리 커널을 효과적으로 적용함으로써 높은 성능의 단백질 간 상호작용 추출 모델을 구축할 수 있음을 보인다. 이를 위해서 이미 Miwa et al.(2009)를 포함한 기존의 연구에서 사용한 구문트리 커널의 적용에 있어서 가지치기와 소멸인자 기반 최적화 작업을 부가적으로 수행함으로써 성능이 극대화될 수 있음을 5개의 말뭉치를 이용한 실험으로 증명한다.

3. 단백질 간 상호작용 추출을 위한 구문 트리 커널 모델

3.1 구문 트리 커널 개요

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 [16]에서 제안한 부분 트리 커널(Sub-Tree Kernel, STK)과 Collins and Duffy(2001)가 [17]에서 고안한 부분 집합 트리 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, 부분 트리(Sub-Tree) 기법보다 훨씬 많은 구성 트리를 생성한다. Moschitti (2006)에 의하면, 부분 트리 커널의 성능은 부분 집합트리 커널에 비해서 성능이 매우 저조하게 나타났다[16-18]. 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.

3.2 합성곱 구문 트리 커널 함수

앞에서도 잠시 언급하였으나 트리 커널 계산 방법의 주된 아이디어는 비교 대상인 두 트리 T_1과 T_2의 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.

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

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.

$$\Delta(n_1, n_2) = \sum_{i=1}^{|F|} I_i(n_1) I_i(n_2) \quad (2)$$

식 (1)과 식 (2)를 기반으로 트리 커널 함수는 다음과 같이 정의된다.

$$K(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2) \quad (3)$$

N_{T_i}는 T_i의 모든 구성 노드 집합을 의미한다. 따라서 식 (3)은 두 트리 T_1, T_2의 모든 자식 노드 쌍에 대해서 $\Delta(n_1, n_2)$의 누적합을 계산한다. [17]에 기술된 내용을 바탕으로 식 (3)에서 $\Delta(n_1, n_2)$은 다음과 같은 알고리즘으로 계산될 수 있다.

```
1  FUNCTION delta(TreeNode n1, TreeNode n2, λ, σ)
2  n1 = one node of T1;                  // 구문 트리에서의 특정 노드
3  n2 = one node of T2;
4  λ = tree kernel decay factor;         // 트리 커널 소멸 인자
5  σ = substructure division methods;    // ST(0), SST(1) 선택 인자
6  BEGIN
7          nc1 = get_children_number(n1);       // 현재 노드의 자식노드 수
8          nc2 = get_children_number(n2);
9          // 두 노드가 모두 말단 노드일 경우,
10         IF nc1 EQUAL 0 AND nc2 EQUAL 0 THEN
11                 nv1 = get_node_value(n1); // 현재 노드의 노드 값을 가져옴
12                 nv2 = get_node_value(n2);
13                 // 두 노드 값(단어)이 같으면 1을 리턴
14                 IF nv1 EQUAL nv2 THEN
15                         RETURN 1;
16                 ENDIF
17         ENDIF
18         np1 = get_production_rule(n1);       // 노드의 문법생성규칙 가져옴
19         np2 = get_production_rule(n2);
20         IF np1 NOT EQUAL np2 THEN    // 만약 두 문법생성규칙이 서로 다르면
21                 RETURN 0;
22         END IF
23         // 만약 두 노드의 문법생성규칙이 같고,
24         // 두 노드 모두 품사태그 노드이면,
25         // 소멸인자 리턴
26         IF np1 EQUAL np2 AND nc1 EQUAL 1 AND nc2 EQUAL 1 THEN
27                 RETURN λ;
28         END IF
29
30         // 재귀적으로 현재 노드의 자식노드에 대한 delta 값 계산
31         // 현재 노드의 모든 첫 번째 자식 노드들에 대한 delta 값의 누승 계산
```

그림 2 식 (3)의 delta 함수 계산 알고리즘

```
32          mult_delta = 1;
33          FOR I = 1 TO nc1
34                  nch1 = I^th child of n1;
35                  nch2 = I^th child of n2;
36                  // 자식 노드들에 대한 delta 함수 재귀 호출
37                  mult_delta = mult_delta × (σ + delta(nch1, nch2, λ, σ));
38          END FOR
39
40          RETURN λ × mult_delta;
41  END
```

그림 2 식 (3)의 delta 함수 계산 알고리즘 (계속)

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 리턴한다. 또한 get_production_rule() 함수는 현재 노드와 자식노드 간의 구조를 파악함으로써 현재 위치에서의 문법생성규칙을 조사한다. 하부 트리 분리 방법 선택 인자 σ는 그림 2의 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.

4. 과학기술분야 관계추출 프레임워크, SINDI-REX

이 장에서는 본 논문에서 개발된 다중 모델 기반 관계 추출 프레임워크인 SINDI-REX(Scientific INtelligence DIscovery - Relation EXtraction)를 소개하고 세부 기능 및 처리 흐름에 대해서 간략하게 설명한다. 그림 3은 SINDI-REX의 전체적인 아키텍처를 보여준다.

아키텍처에 대한 세부 설명은 다음과 같다. 우선 본 논문의 시스템에서 사용한 언어 분석기는 구문분석기, 기저구 인식기(CRF Chunker), 품사 태거(CRF POS-tagger) 등이다. 구문분석기는 Charniak Parser[1]를 도입하여 시스템에 이식시켰다. 또한 다양한 형태의 언어 자질 추출을 위해서 기저구 인식기 및 품사 태거[2]도 독

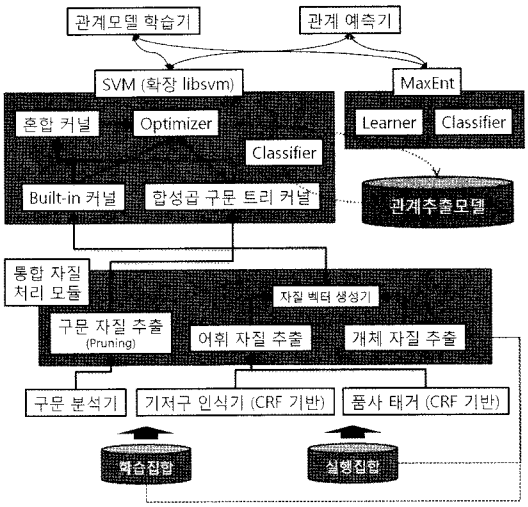

그림 3 관계추출 프레임워크 SINDI-REX 구조

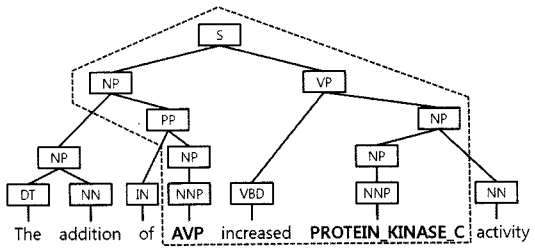

그림 4 경로포함 트리(Path-enclosed Tree) 가지치기 예

립적으로 개발하여 시스템에 결합하였다. 구문 분석 결과는 구문 자질 추출기로 입력되며 여기서 미리 표시된 단백질 명의 문장 내 위치와 주변 구문 정보를 이용하여 가지치기를 수행한다.

위 그림에서 보는 바와 같이 본 논문에서 기본적으로 제공되는 가지치기 기법은 [14]에서 고안한 다양한 방법 중에서 가장 성능이 좋은 것으로 평가받고 있는 경로포함 트리(Path-enclosed tree pruning) 기법을 채택하였다. 이 방법의 특징은 두 단백질 사이에 존재하는 상

1) http://www.cs.brown.edu/people/ec/#software
2) CRF 기반으로 개발된 이 시스템들은 내부적으로 개발되어 개체추출 및 관계추출에 특화된 시스템으로 사용 중이며 본 논문의 작성 기간 중에 학회에 발표 준비 중이다.

호작용을 표현하는 어휘자질과 이를 아우르는 구문자질을 동시에 집중적으로 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 SINDI-REX는 두 가지 기계학습 모델을 기반으로 구축되었다. 우선 SVM 기반 관계추출을 위해서는 *libsvm* 2.89[3]를 자체적으로 확장하여 여기에 구문 트리 커널을 이식시켰다. 결과적으로 *libsvm* 자체적으로 제공하는 네 가지 기본 커널(선형, 다항, RBF, Sigmoid)에 구문트리커널과 혼합 커널(기본 커널과 구문트리커널을 결합시킨 커널)을 추가적으로 제공하도록 하였다. 최대 엔트로피 모델을 활용하기 위해서 *Maximum Entropy Modeling Toolkit for Python and C++*[4]을 이용하였다. 전체 시스템은 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

5.1 실험 대상 말뭉치

실험은 [2]에서 구성한 5 가지의 PPI 말뭉치를 대상으로 수행하였다. 통상적으로 "*Five PPI Corpora*[5]"라고 불리는 이 말뭉치 집합은 AImed[4], BioInfer[5], HPRD50[6], IEPA[7] 그리고 LLL[8]을 하나의 정규화된 XML 형식으로 변환해 놓은 컬렉션으로서 단백질 간 상호작용 추출 기법의 준거 평가 컬렉션으로 활용되고 있다. 따라서 본 논문에서의 실험결과에 대한 비교 대상은 위의 말뭉치를 이용한 연구 결과로 제한한다. 이는 두 가지 이유에서 타당한 접근 방법이다. 첫째, 기존의 많은 연구에서 자신들의 접근 방법에 대한 실험을 수행함에 있어서 다양한 전처리 기법, 교차검증에서의 말뭉치 분리 방법, 말뭉치 선정 등에서 차이를 나타내므로 객관적인 성능비교가 힘들다. 그러나 [2]에서 구축한 말뭉치는 위의 5 가지 말뭉치를 하나로 통합하고, 실험에 필요한 많은 전처리 작업이 수행된 말뭉치이므로, 어느 정도 객관적인 성능 비교가 가능하다. 둘째, 기존의 많은 연구들은 대부분 특정 단일 말뭉치만을 기반으로 성능 측정을 수행하였으므로, 제안된 접근 방법의 일반화 강도(generalization power)를 평가하기가 힘들었으나 [2]의 통합 말뭉치를 사용하면 이러한 문제도 일정 수준 해결이 가능하다.

아래 표는 [2]에서 구성한 "*Five PPI Corpora*"에 포함된 개별 컬렉션의 규모와 상호작용 포함 문장 및 불포함 문장에 대한 통계치이다.

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

그림 5는 Five PPI Corpora에 포함된 BioInfer 말뭉치 내에 존재하는 첫 번째 인스턴스를 보여준다. 단백질 간의 알려진 상호작용의 비정상적 적용을 방지하기 위해서 문장 내의 모든 단백질 이름은 블라인드 처리가 되어 있음을 알 수 있다. 또한 총 4개의 단백질 명이 존재하며, 이들 간의 상호작용 쌍은 총 6가지이다. 결론적으로 위의 문장에서는 총 6개의 단백질간 상호작용 포함 문장이 구성될 수 있으며, 이들 각각은 동일한 문장을 공유하게 된다.

이를 기반으로 단백질 간 상호작용 추출은 개별 인스턴스(상호작용 포함/불포함 문장)에 대한 이진 분류 작업으로 규정할 수 있다. 본 논문에서의 모든 실험은 [1]과 [3]에서와 동일한 10겹 교차평가로 이루어졌다.

5.2 기계학습 매개변수 지정 및 최적화

본 논문의 실험에서는 성능 변화 측정 및 최적화를 위해서 SVM 모델의 정규화 인자(regularization parameter, C)와 트리 커널의 소멸 인자(λ)를 이용하였다.

표 1 *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

3) http://www.csie.ntu.edu.tw/~cjlin/libsvm/
4) http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html
5) http://mars.cs.utu.fi/PPICorpora/eval-standard.html

```
<sentence id="BioInfer.d0.s0" origId="2" text="_____ inhibits _____ signaling by
                    preventing formation of a _____*_____*DNA complex.">
        <entity charOffset="88-100" id="BioInfer.d0.s0.e0" origId="e.2.2" type="Individual_protein" />
        <entity charOffset="0-12" id="BioInfer.d0.s0.e1" origId="e.2.3" type="Individual_protein" />
        <entity charOffset="23-34" id="BioInfer.d0.s0.e2" origId="e.2.4" type="Individual_protein" />
        <entity charOffset="75-86" id="BioInfer.d0.s0.e3" origId="e.2.5" type="Individual_protein" />
        <pair e1="BioInfer.d0.s0.e0" e2="BioInfer.d0.s0.e1" id="BioInfer.d0.s0.p0" interaction="True" />
        <pair e1="BioInfer.d0.s0.e0" e2="BioInfer.d0.s0.e2" id="BioInfer.d0.s0.p1" interaction="True" />
        <pair e1="BioInfer.d0.s0.e0" e2="BioInfer.d0.s0.e3" id="BioInfer.d0.s0.p2" interaction="True" />
        <pair e1="BioInfer.d0.s0.e1" e2="BioInfer.d0.s0.e2" id="BioInfer.d0.s0.p3" interaction="True" />
        <pair e1="BioInfer.d0.s0.e1" e2="BioInfer.d0.s0.e3" id="BioInfer.d0.s0.p4" interaction="True" />
        <pair e1="BioInfer.d0.s0.e2" e2="BioInfer.d0.s0.e3" id="BioInfer.d0.s0.p5" interaction="True" />
</sentence>
```

그림 5 *Five PPI Corpora* 내에서의 BioInfer 말뭉치 첫 번째 문장

일반적으로 SVM을 이용한 실험에서는 대부분 n-겹 교차평가(n-fold cross validation)를 통해서 가장 높은 성능을 나타내는 정규화 인자를 선택한다. 본 논문에서도 이러한 최적화를 수행하였으나, 직접적인 성능 비교 대상인 [1]에서 이 인자를 기본 값 1로 지정하였으므로, 본 논문에서도 동일하게 지정한 결과를 부가적으로 나타낼 것이다.

트리 커널의 소멸 인자(λ)는 최소값 0.1에서 최대값 1.0까지 0.1 단위로 지정하면서 개별 설정에 대해서 10겹 교차평가를 수행하였다. 이에 대한 실험 결과도 아래에 제시하였다. 부가적으로 구문 트리 분리 방법은 모두 부분 집합 트리(SST)를 이용하였다.

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 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 성능 측정 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. [1,3]에서는 어떠한 방법을 활용하여 F-스코어를 계산하였는지에 대해서는 명확하게 설명하지 않았다.

따라서 본 논문에서는 위에서 나열한 두 가지 방법 모두를 사용한 측정 기준을 적용하고 이를 실험 결과로서 제시하였다. 이를 통하여 시스템의 성능을 다각적으로 분석할 수 있다.

5.4 실험 결과

아래 표는 각 말뭉치별로 가장 높은 성능(macro-averaged F1)을 나타내는 설정을 보여준다. 모든 말뭉치에 대해서 높은 성능을 보여주고 있다. 특히 LLL 말뭉치에 대해서는 88.53%의 매우 높은 F1 수치를 나타내고 있다. 또한 기존의 연구에서 성능 개선을 위해서 많은 시도를 하였던 AImed 말뭉치에 대해서도 80% 이상의 성능을 보여주고 있다.

표 2에서도 알 수 있듯이 정규화 매개변수 C는 4.0과 7.0 사이에서 최고 성능을 보이고 있다. 이 매개변수의 최소 제한값과 최대 제한값이 별도로 존재하지 않음을

표 2 두 가지 매개변수 설정에 따른 각 말뭉치별 최고 성능

	λ	C	micro-F1	정확도	재현율	macro-F1
AIMed	0.5	4.0	**89.31**	84.76	77.49	80.96
BioInfer	0.5	6.0	88.90	87.09	84.69	85.87
HPRD50	0.7	6.0	85.22	84.74	83.41	84.07
IEPA	0.4	7.0	79.17	78.51	78.30	78.41
LLL	0.3	4.0	88.48	**88.58**	**88.47**	**88.53**

고려할 때, 이 범위 값은 수렴 정도가 매우 높다고 볼 수 있다. 따라서 구문 트리 커널을 이용한 단백질 간 상호작용 추출에 적절한 매개변수를 이 범위로 지정할 수 있다고 사료된다. 부가적으로 AIMed 말뭉치를 이용한 동일한 조건에서 가지치기를 하지 않고 수행한 성능 실험 결과, 최고 성능은 각각 micro-F1 기준으로 82.67%, macro-F1 기준으로 49.26%의 저조한 성능을 보였다 (최적 소멸인자 : 0.4). 이러한 결과에서 알 수 있듯이 PPI 추출에서의 가지치기 적용은 필수적이라고 볼 수 있다.

트리 커널 소멸 인자에 대해서는 더욱 특이한 현상을 보이고 있다. 표 1에서 제시한 바와 같이 AIMed와 BioInfer 말뭉치는 비교적 규모가 큰 말뭉치이다. 이에 반해서 나머지 말뭉치들은 매우 적은 양의 학습 데이터를 포함하고 있다. 우선 규모가 큰 두 말뭉치의 최고 성능에 대한 소멸 인자는 둘 다 0.5로 동일하다. 그러나 규모가 적은 세 말뭉치의 최고 성능을 보여주는 소멸 인자의 변화는 0.3에서 0.7로 다양하다. 이러한 현상은 그림 6에서 더 명확하게 드러난다.

그래프에서도 알 수 있듯이, AIMed와 BioInfer 말뭉치에 대한 성능 궤적은 거의 포물선 형태로 일정하지만, 나머지 말뭉치의 성능 궤적은 매우 불규칙하게 나타난다. 이러한 현상은 자료 희귀성으로 인해서 트리 커널 소멸 인자의 역할이 두드러짐에서 기인한다고 볼 수 있다. 다시 말해서, 단백질 간 상호작용을 특징적으로 표현하는 구문구조의 다양성 및 포괄성이 부족한 경우에는 이 소멸 인자의 정규화 능력이 전체 성능에 매우 큰 영향을 미치고 있기 때문이다.

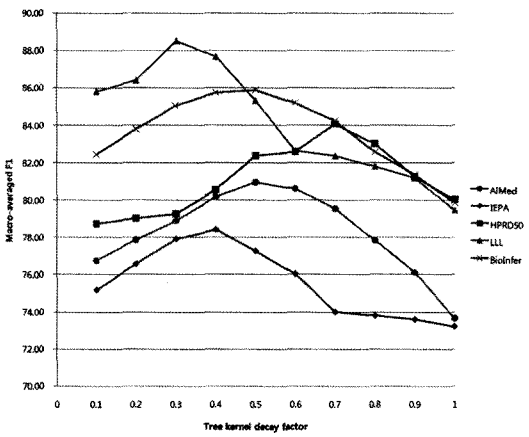

그림 6 트리커널 소멸인자에 따른 성능 변화 추이(개별 말뭉치에 대한 정규화 변수는 각각 최고 성능을 나타내는 값으로 지정)

다음으로 본 논문에서 구현한 접근 방법과 [1,3]에서의 접근 방법에 대한 성능 비교를 아래 표 3에 나타내었다.

비록 n-겹 교차평가의 세부적인 접근 방법[6]에 따라서 성능평가 결과의 차이는 있겠으나, 모든 말뭉치에서 SINDI-REX가 우수한 성능을 보이고 있다. 특히 BioInfer 말뭉치에서의 성능 향상이 두드러진다. 학습 매개변수 중의 하나인 정규화 인자는 1.0으로 [1]과의 객관적인 비교를 위해서 일치시켰다. 그 외에 트리 커널 소멸 인자는 모두 0.5로 동일하게 적용하였다.

성능 향상의 가장 큰 원인은 크게 두 가지로 볼 수 있는데 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_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 [1]에서는 구문트리 커널을 포함한 다양한 커널을 통합 적용했음에도 불구하고 성능 향상이 미비했던 이유도 여기에 있다.

부가적인 성능 개선 요소는 트리커널 소멸인자에 의한 최적화이다. 표 4는 AImed 말뭉치에서 소멸인자의 변화에 따른 성능 차이를 보여준다.

표에서 보는 바와 같이 소멸인자를 적용하지 않았을 때의(λ = 1.0) 성능과 최적의 성능을 보일 때의 성능 차

표 3 Macro-averaged F1 기준 성능 비교. SINDI-REX의 매개변수는 C = 1.0, λ = 0.5로 지정

	AIMed	BioInfer	HPRD50	IEPA	LLL	평균
Airola et al. (2008) [3]	56.4	61.3	63.4	75.1	76.8	66.60
Miwa et al. (2009) [1]	60.8	68.1	70.9	71.7	80.1	70.32
SINDI-REX	**76.8**	**82.0**	**78.2**	**76.0**	**86.2**	**79.84**

6) 폴드 분리 방법(fold division methods), F1 스코어 평균 방법(averaging methods for F1 scores) 등

표 4 AImed 말뭉치에서 트리커널 소멸인자에 따른 합성곱 구문트리 커널의 세부 성능 변화($C = 4.0$)

λ	micro-F1	정확도	재현율	macro-F1
0.1	87.32	81.90	72.18	76.73
0.2	87.84	82.64	73.61	77.87
0.3	88.33	83.46	74.80	78.89
0.4	88.93	84.22	76.51	80.18
0.5	**89.31**	84.76	**77.49**	**80.96**
0.6	89.18	**84.84**	76.80	80.62
0.7	88.69	84.46	75.15	79.54
0.8	87.88	83.29	73.04	77.83
0.9	87.10	82.05	70.98	76.11
1.0	**86.02**	**80.18**	**68.14**	**73.67**

이는 현저하다. F1 측도 기준으로 약 7.3% 정도의 향상을 보이고 있고, 정확도의 상승 정도(4.58%)보다 재현율의 상승(9.35%)이 더욱 돋보인다. 이러한 현상은 소멸인자가 두 구문 트리의 커널 계산에 있어서 평활 값(smoothing value) 역할을 하는 것임을 나타내는 것이다. 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.

지금까지 실험에서 제시한 바와 같이, 단백질 간 상호작용 추출에 있어서 합성곱 구문트리 커널의 성능은 매우 뛰어나다. 기존까지 가장 뛰어난 성능을 보여 왔던 시스템보다 약 14% 정도 향상된 성능을 보여주었다. 또한 부가적으로 트리함수 소멸인자를 조정함으로써 더 높은 성능을 나타내는 모델을 구성할 수 있음을 알 수 있었다.

6. 결론 및 향후 연구 과제

본 논문에서는 합성곱 구문트리 커널을 이용하여 생의학 분야 문헌에 표현된 단백질 간 상호작용 정보를 자동으로 인식하는 연구를 수행하였다. 텍스트 내의 두 개체 간의 의미적 연관 관계를 자동으로 추출할 수 있는 관계추출 프레임워크 SINDI-REX를 기반으로 구문트리커널을 적절히 최적화시킴으로써 높은 성능을 나타내는 시스템을 구성할 수 있음을 알 수 있었다.

본 논문의 중요한 연구 성과는 다음과 같다. 첫째, 단백질 간 상호작용 추출에 있어서 구문트리커널을 적용할 경우 불필요한 문맥정보를 효과적으로 제거하는 구문트리 가지치기 작업이 필수적임을 실질적인 성능 비교로써 보여주었다. 둘째, 동일한 학습 조건에서 구문트리커널의 소멸인자는 평활 인자(smoothing factor)로서 중요한 역할을 하며, 성능 변화의 핵심 요소임을 알 수 있었다. 특히 학습 집합의 규모에 따라서 소멸인자가 성능에 미치는 영향력이 상이한 패턴으로 나타남을 제시하였다. 마지막으로 [1]에서 연구의 결과로서 주장한 "*단일 커널보다 혼합 커널의 성능이 더 뛰어나다*"라는 가설이 항상 성립하는 것은 아니라는 것을 합성곱 구문트리 커널 단독으로 적용하여 높은 성능을 나타냄으로써 보여주었다. 이를 위한 객관적인 성능 평가를 위해 동일한 조건으로 수행한 실험에서 [1]의 성능 보다는 약 19.8%, [3] 보다는 약 14%의 성능 개선을 나타내었다.

향후 연구 과제로서 더 정교한 혼합 커널(composite kernel) 기반의 단백질 간 상호작용 추출 연구가 필요하다. 이미 [1]에서 비슷한 연구결과를 내 놓았으나, 다중 커널의 결합 구조가 매우 단순하다. 현재까지 개발된 커널들에 대해서 다양한 방식의 조합이 가능하며, 경우에 따라서는 단순한 커널 값 연산을 벗어나서 커널의 밀결합도 가능하다. 이와 같은 연구를 통해서 상호작용의 상세 종류까지도 추출할 수 있는 단백질 간 의미적 관계 추출도 가능하다.

참 고 문 헌

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

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

[8] Nedellec C., "Learning language in logic - genic interaction extraction challenge," *Proceedings of*

LLL'05, pp.31-37, 2005.
[9] Pyysalo S., Sætre R., Tsujii J., Salakoski T., "Why Biomedical Relation Extraction Results are Incomparable and What to do about it," *Proceedings of SMBM'08*, 2008.
[10] 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.
[11] Culotta A., Sorensen J., "Dependency tree kernels for relation extraction," *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics*, 2004.
[12] Bunescu R. C., Mooney R. J., "A shortest path dependency kernel for relation extraction," *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics*, pp.724-731, 2005.
[13] Bunescu R. C., Mooney R. J., "Subsequence Kernels for Relation Extraction," *NIPS-2005*, 2005.
[14] GuoDong Z., Zhang M., Ji D., 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 (EMNLP/CoNLL-2007)*, pp.728-736, 2007.
[15] 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.
[16] 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.
[17] Collins M., Duffy N., "Convolution Kernels for Natural Language," *NIPS-2001*, 2001.
[18] Moschitti A., "Making tree kernels practical for natural language learning," *Proceedings of EACL'06*, Trento, Italy, 2006.

최 성 필
1996년 부산대학교 전자계산학과 졸업(학사). 1998년 부산대학교 대학원 전자계산학과 졸업(석사). 2009년 한국과학기술원 대학원 정보통신공학과(박사 수료) 1998년~현재 한국과학기술정보연구원 정보기술연구실. 관심분야는 기계학습, 정보검색, 자연어처리, 정보추출, 텍스트마이닝

최 윤 수
1993년 충남대학교 컴퓨터공학과 졸업(학사). 1995년 충남대학교 대학원 컴퓨터공학과 졸업(석사). 1995년~현재 한국과학기술정보연구원 선임연구원. 관심분야는 데이터베이스, 정보검색

정 창 후
1999년 충남대학교 컴퓨터과학과 졸업(학사). 2002년 충남대학교 대학원 컴퓨터과학과 졸업(석사). 2003년~현재 한국과학기술정보연구원 정보기술연구실. 관심분야는 정보검색 및 추출, 분산 데이터 마이닝

맹 성 현
1983년 미국 캘리포니아 주립대학 학사 1985년 미국 Southern Methodist University(SMU) 석사. 1987년 미국 Southern Methodist University(SMU) 박사. 1987년~1988년 미국 Temple University 교수. 1988년~1994년 미국 Syracuse University 교수(tenured). 1994년~2003년 충남대학교 컴퓨터과학과 교수. 2003년~2009년 한국정보통신대학교 교수. 2009년~현재 한국과학기술원 교수. 관심분야는 정보검색, 텍스트 마이닝, HCI, 상황인지 컴퓨팅 등