생물학 문헌으로부터 단백질 상호작용 정보 추출을 위한 자연어 처리 기법

노정호[O*], 차재혁*, 최용석**
한양대학교 정보통신대학원*, 컴퓨터교육과**
madgo1004@ihanyang.ac.kr, {chajh, cys}@hanyang.ac.kr

Full Parsing Approach to Extracting Protein-to-Protein Interactions from the Biological Literature

Jeong Ho Rho[O*], Jae Hyuk Cha*, Yong S. Choi**
Graduate School of Information and Communications*,
Dept. of Computer-Science Education**, Hanyang University

요 약

단백질 상호작용에 대한 연구는 생명현상의 전반적인 원리를 규명하는데 필수적이다. 생물학 문헌 데이타베이스로부터 단백질 상호작용 정보를 찾는 것은 많은 시간과 노력이 필요하기 때문에 컴퓨터로 자동화 시키는 방법이 요구된다. 문헌으로부터 단백질 상호작용 정보를 추출하는 작업은 단순 문자열 비교를 통한 정보검색으로는 한계가 있으므로 자연어 처리 기법을 적용해 문장의 문법 구조, 품사 정보 등을 이용하면 더 정확한 추출이 가능하다. 본 논문에서는 자연어 처리를 이용하여 문장을 트리로 표현한 뒤 가지치기, 병합 등을 통해 추상화된 트리를 패턴과 매칭하는 방법을 제안한다. 그리고 실제 데이타를 이용한 실험 결과를 통해 기존 방법에 비해 더 높아진 정확도를 확인하였다.

1. 서 론

인류의 미래를 주도할 첨단 산업 기술인 생명과학의 급격한 발전으로 인해 빠른 속도로 많은 양의 데이타가 생성되고 있다. 이런 수많은 데이타 안에서 의미 있는 정보를 찾는 것은 많은 시간과 노력이 필요하기 때문에 컴퓨터를 이용하여 자동으로 정보를 추출하는 방법이 필요하다.

본 연구의 목적은 생물학 문헌 초록으로부터 단백질 상호작용 정보를 찾는 것이다. 자연어로 기술된 초록으로부터 정보를 찾는 가장 일반적인 방법은 문자열 비교를 통해 관심 있는 단백질을 찾는 것이지만 문법 구조나 품사 정보를 고려하지 않은 단순 비교로는 정확한 정보를 추출할 수 없다. 이에 본 논문에서는 자연어 처리를 이용한 효과적인 단백질 상호작용 정보 추출 기법을 제시한다.

2. 관련 연구

문헌으로부터 단백질 상호작용 정보를 추출하는 가장 간단한 방법은 사용 빈도를 이용하는 것이다. 2개의 단백질이 동시에 쓰이면 서로 관계가 있다는 가정 하에 두 단어 간의 상관 계수를 계산하여 관련 있는 단백질을 추출하는 방법이다[1].

그러나 이 방법으로는 문맥의 파악과 상호작용 타입의 추출이 힘들기 때문에 패턴 매칭 (Simple Pattern Matching)이 필요하게 되었다. 문장 속에서 2개의 단백질과 1개의 상호작용 타입이 나타나는 부분에 대해서 *"ProteinA-Type-ProteinB"* 라는 패턴을 통해 *"ProteinA와 ProteinB는 Type의 관계"* 라는 의미를 추출하는 방법이지만 패턴 작성에 많은 시간이 필요하고 작성하기 힘든 패턴이 있다는 단점이 있다[2,3]. 더 나아가 패턴 매칭 방법에 품사 (Part Of Speech) 정보를 추가하여 이전까지 처리하지 못했던 부정문이나 접속사로 연결된 중문/복문까지 처리가 가능한 방법이 있지만 대명사나 단어의 의미적 차이까지는 고려되지 않았다[4]. 앞서 언급한 방법에서 나타난 문제점들을 개선해 보고자 본 논문에서는 자연어 처리를 이용한 정보 추출 기법을 제안한다.

3. 정보 추출 시스템

본 논문에서 제안하는 정보 추출 시스템은 그림 1에서 보이는 것처럼 크게 데이타베이스, 필터, 파서, 추출기의 4개 모듈로 구성되어 있다.

데이타베이스 모듈에서는 단백질 / 상호작용 타입, 검색할 대상인 문헌, 성능평가를 위한 코퍼스 데이타베이스를 각각 구축한다. 그런 다음 문헌 데이타베이스 내에서 불필요한 문장을 걸러내는 필터링 모듈을 거쳐 파서를 통해 문장을 트리 형태로 파싱한다. 생성된 트리는 불필요한 노드의 추상화와 정규표현식과의 매칭 부분으로 구성되어 있는 정보추출 모듈을 거치면서 단백질 상호작용 정보로 추출되게 된다. 각 모듈별 자세한 작동 과정은 다음과 같다.

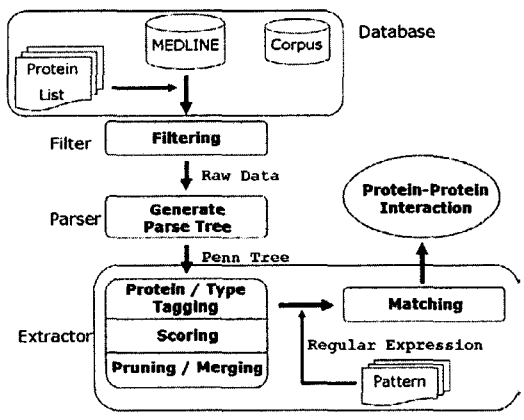

그림 1 정보 추출 시스템 아키텍처

본 논문은 과학기술부 프로테오믹스 이용기술 개발사업 (과제번호 : M102KM010014-04K1301-01411)의 지원을 받았음

3.1 데이타베이스

본 단계에서는 문헌 데이타베이스, 단백질 / 상호작용 타입 데이타베이스, 코퍼스 3가지 종류의 데이타베이스를 구축한다.

3.1.1 문헌 데이타베이스 구축

단백질 상호작용 정보를 추출할 대상으로 생물학 문헌 데이타베이스인 MEDLINE를 선택하였다. XML 형식으로 되어있는 12,481,060 건의 MEDLINE을 다운로드하여 정보 추출에 필요한 PMID, Title, AbstractText, Keyword 4가지 항목에 대하여 데이타베이스를 구축하였다.

3.1.2 단백질 / 상호작용 타입 데이타베이스 구축

ExPASy[5]의 단백질 데이타베이스와 HGNC[6]의 유전자 데이타베이스를 기반으로 Human(Homo Sapiens)과 관련된 단백질 목록을 수작업을 통해 정리하였다. 18,232개의 대표 이름과 그에 따른 동의어를 포함하여 총 39,492개의 단백질 이름과 주로 사용되는 27개의 단백질 상호작용 타입에 대해서 데이타베이스를 구축하였다.

3.1.3 코퍼스 (Corpus) 구축

정보 추출 시스템을 통해 추출된 정보가 정확한지를 검증하기 위해서는 코퍼스가 필요하다. 12,481,060 건의 MEDLINE 문헌 중 Angiogenesis 관련된 문헌에서 랜덤하게 600건을 추출한 뒤 수작업을 통하여 단백질 상호작용 정보를 찾아 기록하였다.

3.2 문장 필터링 (Filtering)

단백질 상호작용 정보는 2개 이상의 단백질과 1개 이상의 상호작용 타입으로 이루어지있기 때문에 전체 문헌 중에서 이러한 조건에 맞는 문장만 필터링 하여 정보 추출 대상 문장을 줄이는 단계가 필수적이다. 따라서 이 조건에 만족하지 않는 문장은 필터링 단계에서 걸러지게 되며 실제 600개의 문헌으로 실험 결과 약 70%의 문장의 필터링 효과가 있었다.

3.3 파스 트리 생성

필터링 단계를 거친 후보 문장을 Stanford Parser[7]를 이용하여 파스 트리로 표현한다. 파서에 의해 생성된 파스 트리는 oneline, penn 2가지 형식으로 출력이 가능한데 트리의 깊이, 노드간 포함 관계가 표시되는 penn 형식을 사용한다. 그림2는 "However, ECGF modulates the effect of HEPARIN on EC synthesis of FN." 라는 문장이 파서에 의해 트리로 생성된 모습을 보여주며 그림3은 penn 형식의 출력을 보여준다.

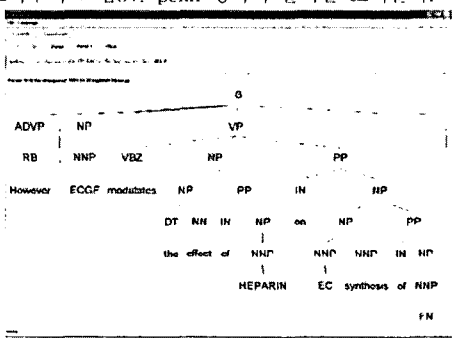

그림 2 파서에 의해 생성된 트리

3.4 정보 추출기

사전 구축된 단백질 이름을 이용하여 트리로 표현된 문장 속에서 정보를 추출하는 모듈이다.

```
(S
  (ADVP (RB However))
  (, ,)
  (NP (NNP ECGF))
  (VP (VBZ modulates)
    (NP
      (NP (DT the) (NN effect))
      (PP (IN of)
        (NP (NNP HEPARIN))))
    (PP (IN on)
      (NP
        (NP (NNP EC) (NNP synthesis))
        (PP (IN of)
          (NP (NNP FN))))))
  (. .))
```

그림 3 penn 형식의 출력

3.4.1 단백질 / 상호작용 타입 태깅 (Tagging)

penn 형식으로 출력된 결과에 단백질과 상호작용 타입에 해당하는 단어에 다음과 같은 형식으로 태깅을 한다.

○ 단백질 → ProName[단백질이름]
상호작용 타입 → IType[동사의 어근]

상호작용 타입의 경우 불규칙한 접미사를 처리하기 위해서 Porter Stemming Algorithm을 적용하는데 Apache Jakarta Project의 풀 텍스트 검색 엔진인 lucene package[8]를 사용한다.

3.4.2 트리의 깊이, 점수 부여 (Scoring)

트리의 깊이를 표시하고 단백질 / 상호작용 타입 노드로부터 루트 노드까지 모든 노드에 2점씩 점수를 부여한다.

```
6 : 0 : (S
0 : 1 : (ADVP (RB However))
0 : 1 : (, ,)
2 : 1 : (NP (NNP ProName[ECGF]))
4 : 1 : (VP (VBZ IType[modul])
2 : 2 : (NP
0 : 3 : (NP (DT the) (NN effect))
2 : 3 : (PP (IN of)
2 : 4 : (NP (NNP ProName[HEPARIN])))))
0 : 2 : (PP (IN on)
0 : 3 : (NP
0 : 4 : (NP (NNP EC) (NNP synthesis))
0 : 4 : (PP (IN of)
0 : 5 : (NP (NNP FN))))))
0 : 1 : (. .))
```

그림 4 트리의 깊이, 점수 부여 결과

3.4.3 노드의 가지치기와 병합

점수가 부여된 트리를 대상으로 불필요한 노드의 가지치기와 병합을 통한 트리의 추상화 작업을 한다. 점수가 0인 노드는 단백질과 무관한 노드이므로 삭제한 뒤 트리의 깊이를 다시 계산해준다. 하지만 가지치기 과정을 거치면서 부모-자식 노드가 중복되는 경우가 발생하기 때문에 부모 노드와 자식 노드의 품사와 점수가 모두 같으면 제거하는 병합과정을 거쳐야 한다. 가지치기와 병합하는 알고리즘은 그림5와 같고, 최종 결과는 그림6과 같다.

```
PennTree pruning(penntree)
  // 점수가 0인 노드 삭제
  for all lines in penntree
    if (score == 0) remove a line
  reCalculateDepth()

PennTree merging(penntree)
  // 부모-자식 노드간 품사와 점수가 같으면 삭제
  for all lines in penntree
    if ((POS-tag of parent == POS-tag) &&
      (score of parent == score)) remove a line
  reCalculateDepth()
```

그림 5 Pruning 과 Merging 알고리즘

```
6 : 0 : (S
2 : 1 : (NP (NNP ProName[ECGF]))
4 : 1 : (VP (VBZ [Type[modul])
2 : 2 : (NP
2 : 3 : (PP (IN of)
2 : 4 : (NP (NNP ProName[HEPARIN])))))
```

그림 6 최종 penn 결과

3.4.4 패턴과 매칭하기 (Matching)

여러 줄로 구성되어있는 penn 결과를 매칭시키기 위해서는 그림 7과 같은 형식으로 기술된 패턴이 필요하다.

```
id:6_pvp_1
(S
(NP AnyPOS ProteinName
(VP TypeVB
(NP
(PP (IN of)
(NP AnyPOS ProteinName
```

그림 7 패턴의 예

패턴마다 유일하게 부여된 id는 *패턴의전체줄수_패턴형태_일련번호* 형태로 구성되며 AnyPOS, ProteinName, TypeVB는 각각 임의의 품사, 단백질, 상호작용 타입을 매칭 시킬 수 있도록 템플릿으로 지정된 정규표현식이다. 이는 추후 사용자가 원하는 패턴을 추가, 수정이 용이하도록 한 설계이다.

```
boolean Matching(patternList, penntree)
  // 1개의 penntree를 patternList의 모든 패턴과 비교
  for all patterns in patternList
     if (pattern.size == penntree.size)
       wholeMatching(pattern, penntree)
     else if (pattern.size < penntree.size)
       partialMatching(pattern, penntree)
     // pattern.size > penntree.size는 매칭 불필요

boolean wholeMatching(pattern, penntree)
  for i = 0 to penntree.size
    if (pattern[i] != penntree[i])
    // 1줄이라도 다르면 매칭 실패
       return false
  return true

boolean partialMatching(pattern, penntree)
  idx = 0   // index of pattern
  for i = 0 to penntree.size
    if (pattern[idx] == penntree[i])
      idx++
      // 연속으로 pattern.size 만큼 같으면 매칭성공
      if (idx == pattern.size) return true
    else
      idx = 0
```

그림 8 Matching 알고리즘

penn결과와 패턴을 매칭하는 알고리즘은 그림8과 같이 구성되어 있다. penn 결과와 패턴의 크기가 같으면 전체 매칭을 시도하고 penn 결과의 크기가 더 크면 부분 매칭을 시도한다. 전체 매칭은 penn의 결과와 패턴의 모든 줄에 대해서 매칭을 시도하는 것이고 부분 매칭은 penn의 결과 안에서 패턴과 매칭되는 sub-penn을 찾는 것을 의미한다.

4. 성능 평가

앞에서 제안된 시스템을 구현하여 실제 데이타로 성능 평가를 하였다.

4.1 실험 환경

실험은 수작업으로 구축한 600건의 코퍼스와 Angiogenesis 관련 189개의 단백질, 27개의 상호작용 타입을 대상으로 단순 패턴 매칭(IE_PM)[3]과 제안하는 시스템에서 사용된 추출기(IE_NLP)의 성능을 비교하였다.

성능평가는 정보 추출기를 통해 추출된 상호작용 정보와 사람이 코퍼스에 기록한 상호작용 정보가 일치하는지 여부를 판단하여 측정하였다.

4.2 평가 결과

IE_PM과 IE_NLP의 성능 평가 결과는 표1과 같다.

표 1 IE와 IE_NLP 성능 평가 결과

	IE_PM	IE_NLP
검증된 정보 (개)	232	232
추출된 정보 (개)	257	114
공통 정보 (개)	103	80
Precision (%)	40.1	70.2
Recall (%)	44.4	34.5

문장의 문법 구조, 품사 정보를 이용하여 불필요한 노드를 삭제, 병합하는 방법으로 추상화함에 따라 자연어 처리를 이용한 IE_NLP가 패턴 매칭 방법에 비해 정보 추출의 정확도(precision)는 높았다. 하지만 37개의 패턴으로는 600개의 문헌을 수용하기가 다소 부족하여 재현율(recall)은 낮은 결과가 나왔다.

품사 정보와 패턴 매칭을 병행한 기존 기법[4]은 4개의 상호작용 타입과 보다 제한된 수의 단백질에 대하여 80% 이상의 재현율(recall)과 90% 이상의 정확도(precision)를 보여주었다. 그러나 본 연구에서 제안하는 기법은 특정 상호작용 타입이나 단백질에 국한된 것이 아니라 보다 일반적인 구성(189개의 단백질, 27개의 상호작용 타입)을 대상으로 실험하였으며 필요한 모든 단백질(및 동의어)과 상호작용 타입에 대하여 확장가능한 방식이다.

5. 결론 및 향후 연구계획

본 논문에서는 자연어 처리를 이용하여 문헌으로부터 단백질 상호작용 정보를 추출하는 기법을 제안하였다. 기존의 시스템에서 사용하지 않았던 문장의 문법 구조와 품사 정보를 이용함으로써 추출의 정확도를 향상시킬 수 있었다.

문헌으로부터 단백질 상호작용 정보를 추출하는 전 과정을 자동화하는 시스템을 구축하였으며 이는 사용자가 관심 있는 단백질과 상호작용 타입을 입력하면 상호작용 정보 추출이 가능한 일반적이고 유연성 있는 시스템이다.

또한 기계 학습 기법을 적용하여 코퍼스로부터 매칭에 사용되는 패턴을 자동으로 생성하는 연구가 진행 중이다.

참고 문헌

[1] Salton G., McGill M, Introduction to Modern Information Retrieval, McGraw-Hill, 1983

[2] Blaschke, et. al. , Automatic Extraction of Biological Information from Scientific Text : Protein-Protein Interactions. In Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology, 1999

[3] 최용석 외 10인, AngioDB : 혈관신생 인자에 대한 데이타베이스 구축 및 활용연구, 대한의료정보학회, 2002년 11월

[4] Toshihide Ono, et. al, Automated Extraction of Information on Protein-Protein Interactions from the Biological Literature, BIOINFORMATICS, 2001

[5] http://ca.expasy.org/

[6] http://www.gene.ucl.ac.uk/nomenclature/

[7] Dan Klein and Christopher D. Manning, Accurate Unlexicalized Parsing, Proceedings of the 41st Meeting of the Association for Computational Linguistics, 2003

[8] http://jakarta.apache.org/lucene/docs/index.html