Link Grammar를 이용한 도메인 온톨로지 확장 방안

The Method of Domain Ontology Population Using Link Grammar

2010년 2월 9일

조선대학교 대학원

컴퓨터공학과

윤 병 수

Link Grammar를 이용한 도메인 온톨로지 확장 방안

지도교수 김 판 구

이 논문을 공학석사학위신청 논문으로 제출함.

2009년 10월

조선대학교 대학원

컴퓨터공학과

윤병수

윤병수의 석사학위논문을 인준함

위원장 조선대학교 교수 _____ (인)

위　원 조선대학교 교수 _____ (인)

위　원 조선대학교 교수 _____ (인)

2009년 11월 26일

조선대학교 대학원

목 차

표 목 차

그 림 목 차

ABSTRACT

The Method of Domain Ontology Population using Link Grammar

Byungsu Youn
Advisor : Prof. Pankoo Kim, Ph.D
Department of Computer Science
Graduate School of Chosun University

Ontology is constructed with concept, definition, and these relation. lately, much of data, ontology do not support reasoning information to users. so, there are so many necessary for ontology population. therefore many studies in ontology population. but most of study, they used manually extraction to concept, relation, properties. this method spend a lot of time and money to data mining. so, to solve this problem, automatic ontology population have studied. it save much of time, money. however it requires another knowledge-base or thesaurus. and they dependent on its source(thesaurus, knowledge-base, etc)

In my study, I deal with ontology population which are not using another famous dictionary, but using Link grammar and infobox of wikipedia. Link grammar is a syntactic parsing theory of English. I analyze link pattern to determine what link pattern will be candidate of relation-concept and extracted triples. then added weight value to each triples. in the result, i got relation and concepts, in order to weight value. weight values are purpose to extract good triples. then i apply infobox, navbox to classify relation, concepts.

First, I gather biology documents in wikipedia. then to get a body part of

wiki-document, apply stampling process. and extract terminologies, which make database to extract important sentences. i set the process for terminology extraction(tagging, tokenize, extraction). and verify terminologies. then i select important sentences to apply Link grammar. I define 7-patterns to get concept-relation triple. after i find good triple through the PMI value and TF value.

At last, i classify concept with infobox, navbox and classify relation with pre-defined classify table which defined Relation hierarchy. i make metadata with this properties, then visualization it.

There are some error in my study, it occurs in wrong Pos tagging, stopword interruption. anaphora relosolution. but visualization works well, and it's possible to extract more than 1 relations in 1 sentence. finally, I will study about named entity recognization to finding correct triples .

Ⅰ. 서론

A. 연구배경 및 목적

온톨로지는 특정 도메인에 대한 개념과 속성 ,관계, 추론규칙등의 정보를 제공함으로써 최근 주목받고 있는 의미적 정보처리의 지식베이스 역할을 가능하게 하고 있다. 온톨로지는 정형화된 표현을 통해 정확한 지식 처리와 추론관계를 명시해야 하기 때문에 온톨로지 확장에 대한 중요성 역시 강조되고 있다. 온톨로지 확장을 위한 기존의 방법들은 전문가를 통한 수작업 형태이거나 보편화된 사전이나 시소러스 집단의 분석을 통한 통계의 확률분포를 이용하는 반자동화된 방법들이 있다. 수작업으로 생성할 경우 컨셉 추출과 관계 생성에 대한 정확성은 뛰어나지만 많은 시간과 비용을 필요로 하며 이를 해결하기 위한 반자동화된 방법에서는 텍스트 분석시 태깅된 단어에 대한 해석의 차이점과 컨셉과 관계를 추출하기 위해 보편화된 사전이나 고차원적인 학습문서에 의존하는 경향이 존재한다. 따라서 참조한 문서가 수정되기 전까지 온톨로지 구축및 확장에는 제한적이다. 이를 해결하기 위해 본 연구에서는 대중의 지혜가 잘 반영되어 있는 위키피디아를 이용하였다. 위키피디아는 다양한 정보가 집약되어 있으며 목록 도표 등으로 잘 분류되어 있다. 그리고 가장 큰 특징으로 정보의 변화에 맞춰 빠르게 생성되어지고 수정되어 지는 장점을 가지고 있다. 따라서 이를 이용하여 개념-관계를 추출하거나 온톨로지 구축 및 확장을 시도하는 다양한 연구들이 수행되어 지고 있다[2].

본 연구에서는 대중에 의해 빠르게 정보를 획득할 수 있는 위키피디아를 대상으로 온톨로지 확장 방법을 제안한다. 가장 먼저 핵심어 기반의 중요문장들을 추출한다. 그리고 추출된 문장들을 Link gramma분석을 통해 문장을 구조화 하고 이로부터 다양하고 정확한 triple을 추출하는 온톨로지 확장방법을 제안하였다. Link gramma parser의 링크 패턴을 통한 문장분석은 내부의 사전을 기반으로 구성 단어들 간의 세부적인 연결 관계를 보임으로서 기존의 품사태깅에 의한 주어-서술어-목적어 Triple보다 다양한 문장 구조화가 가능하다.

제안된 논문의 실험을 위해 위키피디아의 생물학 관련문서로부터 핵심어 처리 과정을 통하여 핵심문장들을 추출하였고 Link grammar의 분석 패턴을 이용하여 컨셉과 관계를 추출하였다. 그리고 추출된 각 관계와 개념에 대해 가중치를 통해 순위화 하고 온톨로지 형식을 정의한 후 시각화하였다. 본 실험의 결과를 위해 위키피디아의 DNA문서로부터 추출된 400여개의 핵심문장들을 Link grammar를 이용하여 Triple을 추출하였고 그 결과를 평가하였다. 위키피디아에서는 전문적인 데이터가 아닌 대화식 어체로 인해 문장이 내포하고 있는 대명사, 관계어들이 noise로 작용하였지만 전체적으로 만족할 성능을 보였으며 기존 추출방법과 비교한 결과 Link grammar를 이용하여 추출한 Triple이 더 나은 정확성을 보였다. 그리고 추출된 Triple들의 시각화 역시 좋은 결과를 얻었다.

B. 연구내용 및 구성

본 논문에서는 온톨로지 확장을 위해 위키피디아의 도메인 문서들을 논문에서 이용가능한 형태로 처리한 후 문장내에서 핵심어를 추출하고 이를 바탕으로 하는 핵심문장을 추출하였다. 그리고 Link gramma의 Link Path의 패턴을 분석하여 핵심문장을 구조화 하고 상하위어 및 관계를 추출하였다. 추출한 상하위어 및 관계에 대하여 PMI(상호정보량)와 빈도수로 순위화 한 후 가중치가 높은 Triple들을 미리 정의한 온톨로지와 어휘매핑, 그리고 wikipedia의 infobox를 이용하여 속성과 관계의 계층을 정의하였고 이를 메타데이터로 기술하여 시각화하였다.

2장 관련연구에서는 기존의 온톨로지 확장과 Link grammar에 대해 살펴본다. 그리고 3장에서는 위키피디아에서 관련 도메인 문서를 읽어들이고 이중 관계와 컨셉 추출에 필요한 부분인 본문만을 처리하는 전처리과정과 문서의 의미를 포함하고 있는 핵심어 추출을 시행하고 전문용어 추출기와의 비교평가를 수행한다. 그리고 추출된 핵심어로부터 핵심문장만을 추출하는 방법을 제시한다. 4장에서는 Link grammar를 분석 한 후 링크패턴에 적합한 Triple을 추출한다. 그리고 이를 순위화 하고 관계와 개념들을 온톨로지 형식에 적용할 수 있도록 온톨로지 형식을 설계하였다. 5장에서는 온톨로지 형식에 맞게 정의된 상 하위어 및 관계들을 메타데이터로 기술하고 이를 시각화 한다. 마지막으로 6장에서는 추출된 상 하위어와 관계에 대한 신뢰도와 정확성을 평가, 기존의 품사태깅을 이용한 상하위어 및 관계추출 방법과의 비교평가를 실행한다.

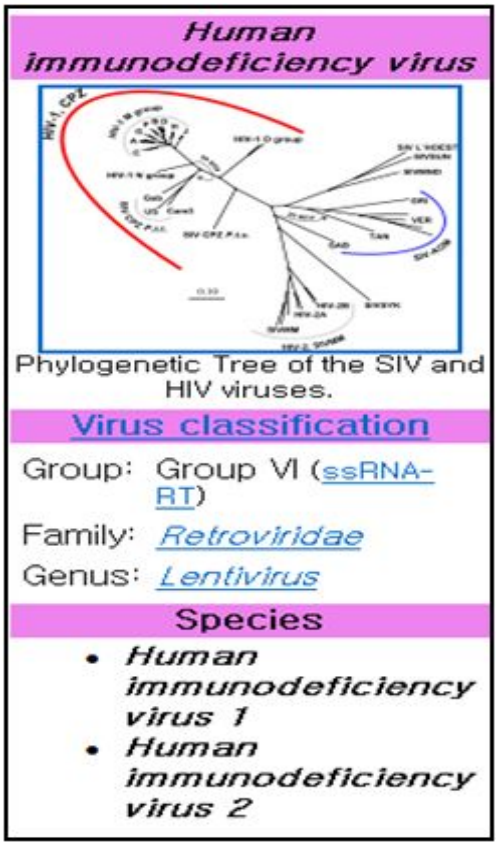
Ⅱ. 관련 연구

A. 위키피디아를 이용한 연구

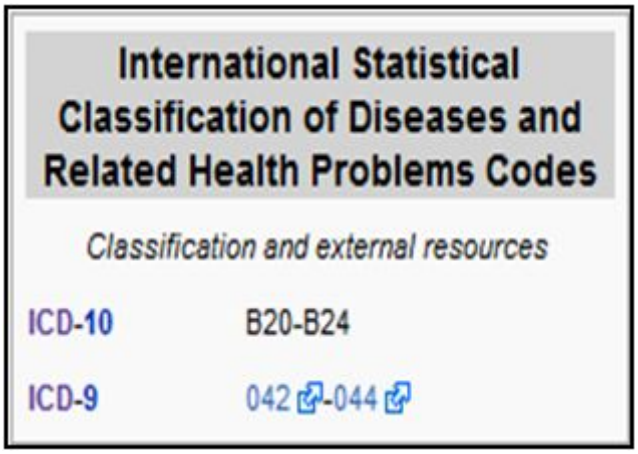
본 절에서는 Triple 추출을 위해 위키피디아를 이용하게 된 배경과 위키피디아의 장점 및 이를 이용하여 연구한 사례들에 대해 살펴본다.

위키피디아는 웹기반의 다국적 언어의 자유로운 컨텐츠를 지향하는 백과사전 프로젝트로부터 시작되어 오늘날 가장 많이 접속하는 웹사이트중의 하나이다. 위키피디아는 전 세계 모든 사용자들이 정보의 생산자 혹은 가공자로 참여하여 웹 2.0을 대표하는 대중의 지혜(The wisdom of crowds), 혹은 집단지성(The collective intelligence)이 가장 잘 반영된 곳으로, 특정 도메인의 개념과 관계를 추출하는데 있어서 훌륭한 대상이다. 위키피디아 문서의 구성은 주요 기사의 제목과 이를 설명하는 Text body와 그림, 표, 목차, reference, category들로 이루어져 있으며 중요한 특징으로 기사내용에 대한 분류항목을 지니고 있다. 목차에서는 기사내용을 순차적으로 기술하고 있고 문서 내부에 분류항목으로 infobox나 navbox와 같은 box형식의 표, 그림을 통해 세부 분류와 특징을 표현하고 있다. infobox와 navbox는 특정 양식에 따라 사용자들에 의해 작성되는 것으로 전문적인 사전이나 서적, 연구문헌, 뉴스등을 참고 하여 기재된다.

[그림 2-1]는 위키피디아 문서중 HIV(Human Immunodeficiency Virus)에 관한 infobox를 나타내고 있다. 사진이 있는 경우는 사진에 대한 설명과 (Group, Family, Genus)항목에 따른 분류, 종류등의 다양한 정보를 나타내고 있다. 또한 [그림 2-2]와 같이 외부의 리소스에 의한 분류 등의 다양한 정보를 포함하기도 한다. infobox 외에 위키피디아의 하단부의 navbox에서는 현재 기사의 토픽이나 관련기사, 종류 등의 다양한 정보를 표 형식으로 포함 하고 있다.



[그림 2-1] 위키피디아 infobox(1)



[그림 2-2] 위키피디아 infobox(2)

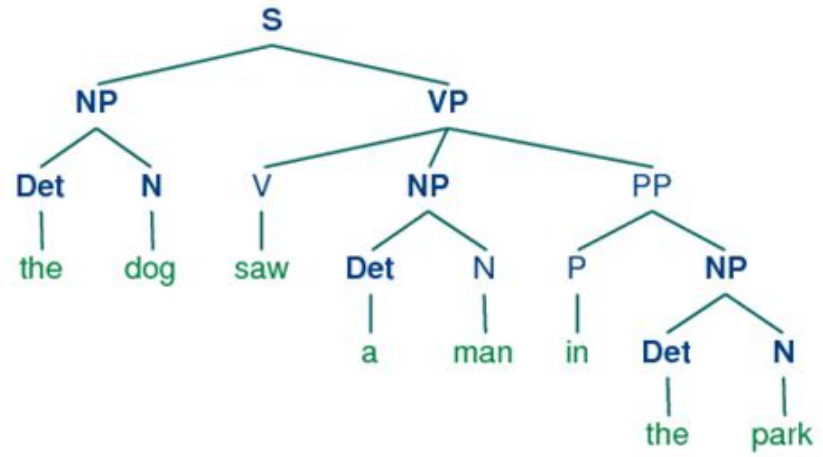
이처럼 위키피디아에서는 대중에 의한 빠른 생성과 수정, infobox와 같이 체계화된 정보를 포함하고 있기 때문에 최근에는 위키피디아를 이용하여 정보처리에 관한 다양한 연구들이 수행 되고 있다.

Harvesting Wiki Consensus[1]연구는 위키피디아 자체가 온톨로지 체계를 구성하고 있음을 파악하고 위키피디아 기반을 통해 온톨로지를 구성한다면 위키피디아의 개체들을 재사용할 수 있기 때문에 일반 온톨로지에서 사용되는 개체의 생성 및 유지가의 어려움을 줄일 수 있다고 주장하였다. 또한 위키피디아의 멀티미디어 개체들은 온톨로지에서의 의미명확화와 의미의 풍족화의 향상을 위해 사용 할 수 있다고 주장하였다.

Robust Minimal Recursion Semantics(RMRS)[2]시스템은 위키피디아의 biological내용으로부터 12,000개의 동물과 관련된 위키피디아 페이지로부터 개념들의 관계를 추출하고 이를 통해 온톨로지를 구축 하였으며. [3]의 연구에서는 위키피디아의 infobox의 class들을 SVM과 HMM등을 통해 IS-A관계등으로 추출하고 wordnet[4]과의 매핑을 통하여 온톨로지를 구축하였다. 본 논문에서는 이와 같은 위키피디아의 대중성과 발전성을 바탕으로 손쉽게 유지 보수가 가능한 온톨로지 확장 방법을 제안하였다.

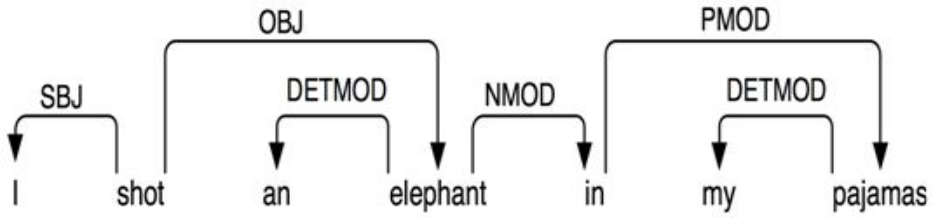
B. Link Grammar

현존하는 영어문법 파서는 대부분 품사태깅을 기초로 이루어진다. 대표적인 영어문법으로 구 구조 문법과 의존문법이 있다. 구 구조 문법 중에 하나인 context free grammar는 구에 대한 해석을 바탕으로 문맥의 흐름에 영향을 받지 않고 미리 정해진 규칙에 따라 문장의 해석트리를 구성한다. 그림[2-3]은 문장을 context free grammar에 따른 예를 보이고 있다. 문법적 형태를 구와 단어별로 정의한 다음 바꿔쓰기 규칙을 통해 문장을 해석해 나간다. 가장 단순한 문법으로 구를 통해 문장 요소에 대한 해석은 쉽지만 태거에 대한 의존성이 크며 중의적 어휘에 대한 해석이 어렵고 이를 위한 패턴이나 알고리즘을 필요로 한다.



[그림 2-3] context free grammar

의존문법은 구 구조 문법과 달리 구노드가 존재 하지 않는다. 대신 핵심어와 핵심어에 의존하는 용어들을 기반으로 문장을 분석한다. 일반적으로 의존 문법에서는 시제동사를 핵심어로 설정하고 다른 단어들을 의존어로 파악하여 관계를 형성한다.



[그림 2-4] 의존문법의 예

[그림2-4]는 "I shot an elephant in my pajamas"을 의존문법으로 분석한 그림이다. shot, elephant, in, pajamas가 head이다. 화살표를 받는 부분은 의존어들이며 head와 의존어의 관계가 표현되어 있다. 각각은 SBJ=주어, OBJ=목적어, DETMOD=한정어, NMOD=명사수식, PMOD=전시사 수식을 나타내고 있다. 구 구조문법과 의존 문법을 비교했을 때 주어, 서술어, 목적어를 직관적으로 파악할 수 있다는 장점이있다. 하지만 문장의 구성 성분은 핵심어와 관계하고 있기 때문에 핵심어를 제외한 단어들간의 상하위어 및 관계추출에 제약이 따른다.

link grammar는 의존문법과 유사하지만 핵심어와 의존어 관계가 존재하지 않으

며 방향성이 존재하기 때문에 다른 문법이라 할 수 있다. 핵심어-의존어가 존재 하지 않고 모든 단어들은 평등하게 1쌍 이상의 관계를 갖기 때문에 다양한 링크관계가 존재하며 이를 조합하여 다양한 관계추출을 고려 할 수 있다. link grammar는 대부분의 영어단어에 대해서 사전으로 단어의 링크 규칙을 정의하고 있다.

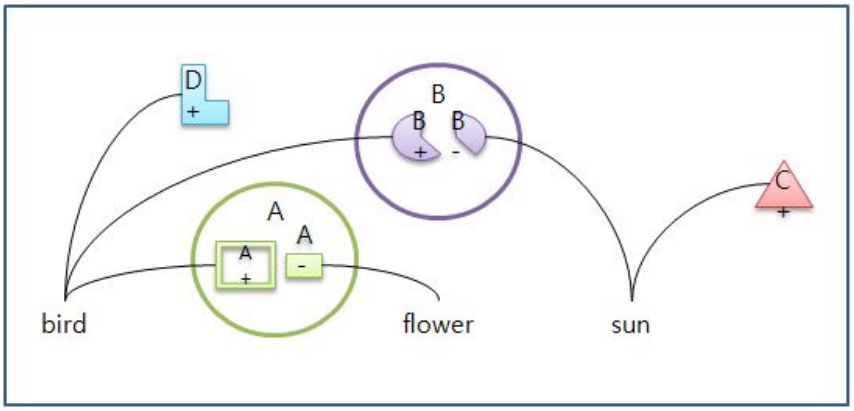
[표 2-1] link grammar에 정의된 단어의 링크 규칙

규칙	관계 규칙
1	blah: A+;
2	blah: A+ or (B- & C+);
3	blah: A+ & {B+};
4	blah: (A+ or B+) & {C- & (D+ or E-) } & {@F+ };

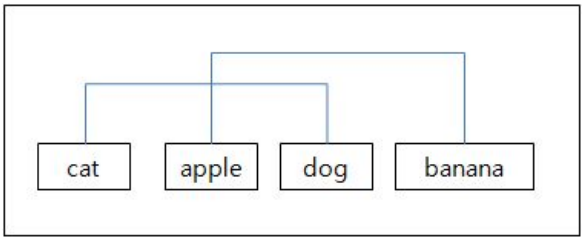
[표 2-1]은 link grammar 사전에 정의된 임의의 단어가 갖는 관계를 나타내고 있다. 1번 규칙에서 A는 Link grammar에서 사용되는 단어들 사이의 링크 관계를 나타낸다. 즉 A의 관계는 의존문법에서의 주어관계, 목적어 관계등을 나타내는 기호이다. 규칙 1의 단어(blah) 는 blah의 우측(+)에 위치한 단어들과 “A 관계를 가질 수 있다”.라고 표현된다. +는 -와 더불어 방향성을 가리키는 기호이다. 따라서 A관계를 맺고 있는 다른 단어는 blah와 A-라는 링크관계가 정의되어있다. 2번에서는 임의의 단어는 A의 관계를 갖거나 좌측(-)과 B의 관계이고 우측과 C의 관계를 갖는다고 표현된다. 3번 규칙의 중괄호는 선택사항으로서 A-관계 이외에 우측의 타 단어와 B-link path를 가질 수 있음을 의미한다. 그림[2-5]는 규칙성을 바탕으로 다른 단어들과 링크관계를 생성하는 방식을 보이고 있다. +링크와 -링크의 결합을 통해 관계가 생성된다.

또한 관계들간의 오류와 복잡성을 줄이기 위하여 전역규칙을 설정하고 있다.

(1). cross-linking: 단어들간의 교차 링크를 허용하지 않는다. 교차 관계를 생성할 경우 일관성이 결여된다. 즉 한문장에 다수의 주어가 존재하거나 주어를 수식하는 형용사가 목적어를 수식하는 오류를 범하게 된다. [그림2-6]은 cross-linking의 예를 보이고 있다.

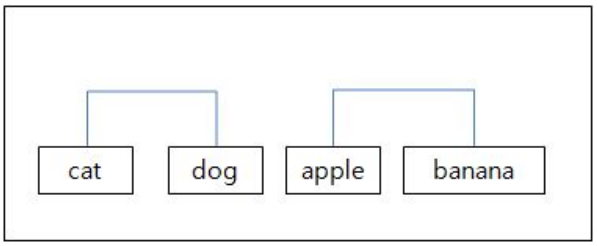


[그림 2-5] 링크생성 규칙



[그림 2-6] cross-linking

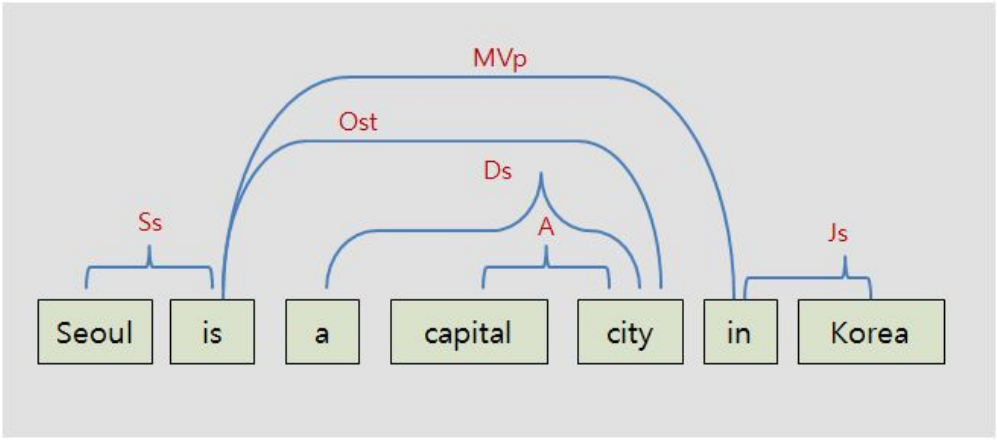
(2). connectivity: 모든 단어들은 직접 또는 간접적으로 연결되어 있어야 한다.



[그림 2-7] connectivity

[그림2-7]은 connectivity 규칙성을 설명하고 있다. 각 단어의 링크가 끊겨있으므로 적합한 여러 관계중 어떤 관계를 정의해야 할지 파악할 수 없다. 즉 단어간의 연결이 되지 않아 문장을 구조화 할 수 없다.

Link grammar에서 정의하고 있는 관계들은 주어, 목적어등 약 100여개의 관계를 기호로 나타내고 있다[5]. 따라서 문장을 세세하게 파싱할 수 있으며 관계의 조합을 통해 다양한 Triple들을 생성할 수 있다. [그림 2-8]은 Link grammar를 이용하여 문장을 파싱한 예를 보이고 있다.



[그림2-8] link grammar를 이용한 문장분석

그림에서 S(주어-동사), O(동사-목적어), D(한정사-명사), MV(동사-수식어구), A(형용사수식어-명사), J(전치사-목적어)를 각각 나타내며 [그림2-8]을 통해

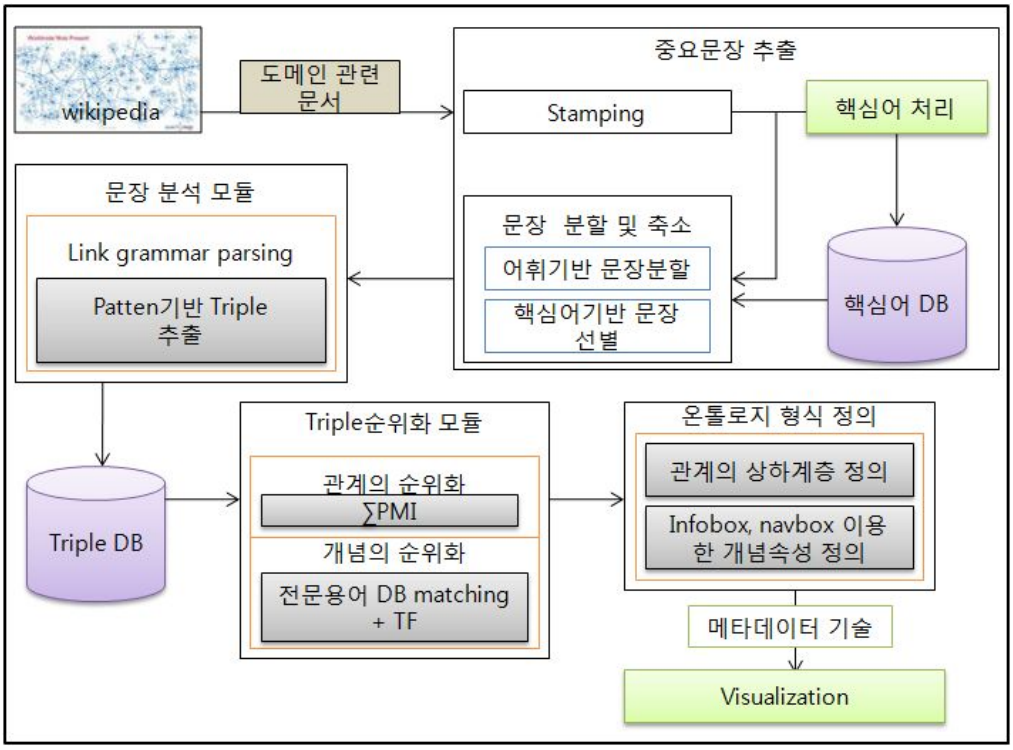
1. Seoul - is - a capital city
2. Seoul - is - in Korea 두 가지 형태의 Triple을 추출할 수 있다.

이와 같은 장점을 이용하여 본 논문에서는 위키피디아 문서를 핵심적인 패턴들을 이용하여 Link grammar로 구조화 한후 다양한 관계들을 추출하고자 한다.

다음 장에서는 논문의 전체 구성과 문장을 구조화하기 위하여 위키피디아로부터 핵심문장을 추출하는 과정에 대해서 다룬다.

Ⅲ. 핵심문장 추출

본 논문의 전체구성은 [그림 3-1]과 같다. 위키피디아에서 도메인 관련 문서들만을 수집한 후 중요문장 추출과정을 통해 해당 문서내에서 핵심어 처리 과정을 통해 핵심어를 파악하고 핵심어를 바탕으로 한 중요문장을 추출한다. 그리고 문장의 복잡성을 증가시키는 관계어나 접속사를 어휘매칭을 기반으로 분할한다. 그 후 문장 분석 모듈에서는 문장들을 Link grammar parser를 통해 구조화 하고 링크들을 분석한 패턴을 바탕으로 Triple을 추출한다. 추출된 Triple들을 순위화 모듈을 통해 각 개체들에 가중치를 부여한 후 높은 가중치를 가지는 Triple순으로 순위화 한다. 온톨로지 형식 정의단계에서는 미리 정의한 온톨로지와 Triple의 상하위어와 관계들에 어휘매핑, 그리고 위키피디아의 infobox와 navbox를 이용해 속성과 관계를 정의하고 최종적으로 메타데이터를 기술한 후에 이를 시각화 한다.



[그림 3-1] 전체 구성도

A. 문서처리

1. 위키피디아 본문추출

본 절에서는 위키피디아 도메인 관련 문서로부터 다양한 태그과 도표, 그림등을 배제하고 Text부분만을 수집하는 과정을 다룬다. 논문에서 사용될 자료들은 wikipedia중에서 medical관련 문서들만을 수집한 후 본문만을 추출한 data를 사용한다. 문서를 처리를 하기 위한 과정은 다음과 같다.

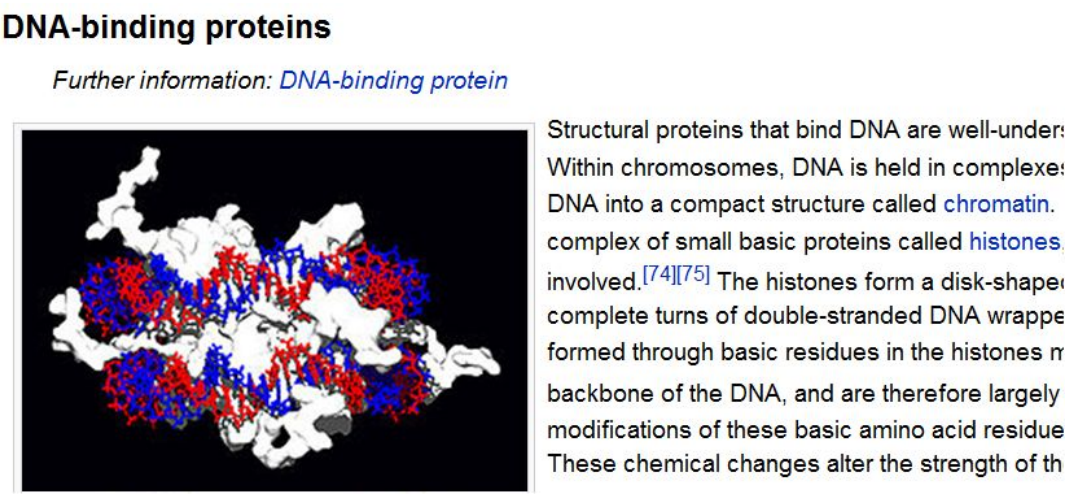
(1). html 문서를 읽어온다

(2). html문서에서 (p | table | from)tag를 포함하고 있는 문구를 읽어온다.

(3). (div | bt | tr)tag를 삭제 한 후 html tag들과 유니코드, 특수문자 처리한다.

(4). 각 단어들을 공백을 삭제한 후 줄바꿈 문자를 삽입한다.

(5). 구문단위로 배열에 저장한다.



[그림 3-2]위키피디아 본문

[그림 3-2]는 웹상에 나타난 위키피디아 본문을 나타내고 있으며 아래 [그림3-3]에서는 본문추출과정을 통해 변환된 문서를 보이고 있다.

Further information: DNA-binding protein
Interaction of DNA with histones (shown in white, top). These proteins' basic amino acids (below left, blue) bind to the acidic phosphate groups on DNA (below right, red).
Structural proteins that bind DNA are well-understood examples of non-specific DNA-protein interactions. Within chromosomes, DNA is held in complexes with structural proteins. These proteins organize the DNA into a compact structure called chromatin. In eukaryotes this structure involves DNA binding to a complex of small basic proteins called histones, while in prokaryotes multiple types of proteins are involved.[74][75] The histones form a disk-shaped complex called a nucleosome, which contains two complete turns of double-stranded DNA wrapped around its surface. These non-specific interactions are formed through basic residues in the histones making ionic bonds to the acidic sugar-phosphate backbone of the DNA, and are therefore largely independent of the base sequence.[76]

[그림 3-3] Tag를 삭제한 위키피디아 본문

현재까지 추출된 본문들은 본문에서 그림과 표, 목차들을 삭제한 결과이다. 이를 본 연구에 적용하기 위해 다음절의 전처리 과정을 거친다.

2. 문장 전처리 과정

전처리 과정을 통해 위키피디아로부터 추출된 본문들을 문장단위로 분류한다. 문장이라고 판단하는 기준은 각 paragraph의 끝인 줄바꿈 문자가 포함된 부분과 마침표를 기준으로 판단하였다. 하지만 소수점 단위의 수치를 나타내는 표현 등을 문장으로 취급하는 것을 배제하기 위해 마침표 이후에 다른 문자가 있는지를 파악하고 공백일 경우에만 문장으로 취급하였다. 먼저 본문을 정규식 표현 과정을 거침으로서 link grammar parser분석에 noise를 생성할 수 있는 문자열을 삭제한다. 즉 문장의 어미에 reperence를 나타내는 기호인 [숫자]와 ()나 물음표기호, 그리고 본문추출시 특수문자 처리 오류로 생성된 물음표 등의 기호를 모두 삭제하였다. 단 2-deoxyribose와 같이 '-'기호와 어퍼스트로피 문자는 문장의미 파악을 위해 삭제하지 않았다. [그림 3-4]는 위키피디아로부터 전처리 과정을 거친 최종적인 문장 추출결과를 보이고 있다.

```
Sentence 24: 'For instance, the largest human chromosome, chromosome number 1,
approximately 220 million base pairs long.'
Sentence 25: ' In living organisms, DNA does not usually exist as a single
molecule, but instead as a pair of molecules that are held tightly together.'
Sentence 26: ' These two long strands entwine like vines, in the shape of a
double helix.'
Sentence 27: 'The nucleotide repeats contain both the segment of the backbone
the molecule, which holds the chain together, and a base, which interacts with
the other DNA strand in the helix.'
Sentence 28: 'A base linked to a sugar is called a nucleoside and a base linke
to a sugar and one or more phosphate groups is called a nucleotide.'
Sentence 29: 'If multiple nucleotides are linked together, as in DNA, this
polymer is called a polynucleotide.'
```

[그림 3-4]위키피디아 문장 추출

분할된 문장들로부터 핵심문장만을 획득하기 위해서 다음 절에서 각 문장으로부터 핵심어를 파악하게 된다.

B. 핵심문장 추출

1. 핵심어 추출

핵심어를 추출하는 과정은 토큰화, 태깅, 핵심어 추출 과정으로 이루어진다. 핵심어란 주어진 도메인 안에서 의미를 가지고 있는 단어들의 집합으로 도메인 내에서 사용되는 개념을 표현하여 주제를 특성화해주는 어휘적 단위를 말한다. 이러한 핵심어는 하나의 도메인을 이해하는데 필요한 요소이기 때문에 특정 도메인에 대한 기계번역이나 정보검색을 보다 효율적이고, 정확히 수행하기 위해서 필요하다. 본 논문에서는 낮은 모호성(Ambiguity)과 높은 특정성(Specificity)을 지닌 핵심어를 이용하여 문서내에서 중요성을 지닌 문장을 추출한다. 핵심어를 바탕으로 문장을 추출하기 위해 가장 먼저 핵심어의 출현 형태를 분석하였다.

핵심어의 형태결합 방식은 매우 다양하다. 해당 도메인에 출현하는 대부분의 핵심어들은 약어이거나 복합명사 형태로 출현하며, 이를 분석한 결과 크게 두 가지의 결합 형태로 나눌 수 있다. 하나는 단일어절(Singleton Term) 즉, 띄어쓰기가 없는 한 어절로 나타나는 형태이고, 다른 하나는 다중어절(Multi-word Term)의 형태로 띄어쓰기가 나타나며 앞의 어절성분과 의미적으로 관련이 있는 두 어절이상으로 이루어진 복합명사이다. [표 3-1]은 이를 기반으로 핵심어의 출현 형태를 파악

하고 있다.

[표 3-1] 핵심어의 구조

Structure	example
1. Singleton Term(NN,NNS,NNP)	(chromosome, NN), (genes, NNS), (DNA, NNP), (strand, NN), (protein, NN)
2. multi-word Term(1+1, JJ + 1)	Ribonucleic acid, Nucleic Acids, Recombinant DNA, oxidative lesions

단일어절로 이루어진 전문용어들은 약어로 이루어진 경우와 일반 명사들로 이루어져있으며 약어일 경우 주로 고유명사인 NNP로 파악되는 경우와 단일명사(NN), 복수명사인 (NNS)로 파악되었으며 [표3-1]의 2번과 같이 명사와 명사 또는 수식어(JJ)와 명사의 결합으로 전문용어의 완전한 표현이나 복합어를 통한 새로운 용어들이 이에 해당한다. [표3-1]의 패턴형태에 따라 추출된 핵심어들은 데이터베이스화 한다. 다음 절에서는 핵심어를 추출하는 프로세스인 토큰화 과정과 태깅 그리고 핵심어 추출 단계에 대해서 기술한다.

a. 토큰화(Tokenizing)

핵심어 추출의 첫 단계는 문장 내 텍스트를 용어로 토큰화 하는 것이다. 공백(white space)을 기준으로 하는 토큰화 외에 몇가지 복잡한 경우를 고려한다. 일반적인 토큰화는 다음과 같다.

tagger.tokenize("Twin helical strands form the DNA backbone")

['Twin', 'helical', 'strands', 'from', 'the', 'DNA', 'backbone']

-특수문자가 용어내에 존재하는 경우에 특수문자와 결합된 용어를 한단어로 파악하였다.

tagger.tokenize("Parts-Of-Speech")

['Parts-Of-Speech']

tagger.tokenize("amazon.com")

['amazon.com']

-소유격이 존재하는 경우에는 소유격 표현을 분리한 후 태깅시에 활용하도록 하였다.

tagger.tokenize("my parents's car")

['my', 'parents', "'", 'car']

tagger.tokenize("my father's car")

['my', 'father', "'s", 'car']

-숫자표현과 날짜표현 역시 소수점과 -표현을 한 단어로 파악한다.

tagger.tokenize("12.4")

['12.4']

tagger.tokenize("-12.4")

['-12.4']

tagger.tokenize("09.06.1982")

['09.06.1982']

tagger.tokenize("09-06-1982")

['09-06-1982']

본 단계를 통해 문장을 토큰화 한 후 태깅과정을 거쳐 품사를 파악한다.

b. 태깅(Tagging)

토큰화(Tokenizing)된 각 단어들에 대해 품사태깅(Part of Speech tagging)을 실행한다. 품사태깅은 penn treebank project의 품사태거를 이용하였다.

[표 3-2] POS tag Label

	통화기호	JJR	Comparitive Adjective
,	comma	JJS	Superlative Adjective
.	period	MD	modal verb
:	colon, semi-colon, dash	NN	Singular Noun
POS	Possessive Ending	NNP	Singular Proper Noun
CC	Coordinating Conjunctions	NNSS	Plural Proper Noun
CD	cardinal Number	NNS	Plural Noun
DT	Determiner	RB	Adverb
IN	Preposition	To	to
JJ	Adjective	VB	Base Form Verb

[표3-2]는 품사태깅시 사용되는 tag들과 내용을 나타내고 있다. 본 논문에서는 전문용어가 주로 고유명사, 단일명사로 이루어진 Sing Term과 수식어 구나 복합명

사로 이루어진 Multi-Term을 추출하기 위하여 tag가 JJ로 시작되는 용어와 NN으로 시작하는 용어들을 이용한다.

태깅의 정확성 높이기 위해 명사(NN)와 같은 단수명사로 파악된 단어들이 복수명사(NNS)형태로 다시 출현 하더라도 같은 용어로 파악하기 위해 태깅후 본래 단어의 기본형을 함께 포함하여 배열에 저장한다.

['Twin', 'NNP', 'Twin'] ['helical', 'NN', 'helical'] ['strands', 'NNS', 'strand']
['form', 'NN', 'form'] ['the', 'DT', 'the'] ['DNA', 'NNP', 'DNA']
['backbone', 'NN', 'backbone']

c. 핵심어 추출

핵심어 추출에서는 품사태깅된 각 단어들을 정해진 패턴에 따라 핵심어 후보를 생성한다. 용어의 JJ(형용사)tag와 NN(명사)tag를 기준으로 하여 single Term(단일어절)과 multi-word Term(복합어절)을 생성한다.

1. tag가 NN 또는 JJ로 시작하는지를 파악한다.
2. NN으로 시작할 경우 핵심어 후보에 추가한 후 multi word탐색을 시작한다.
3. multi word탐색중이며 다음 tag가 NN일 경우는 현재 단어를 핵심어 후보에 추가한 후 현재까지의 용어를 저장하고 다음 단어의 tag를 파악한다.
4. 3의 과정 후 현재 단어의 tag가 NN이 아닐 경우 이전까지의 용어가 2어절 이상인지를 파악하고 핵심어 후보에 추가한다. NN일 경우는 3의 과정을 반복한다.
5. 1의 과정에서 JJ로 시작할 경우 현재 단어를 저장 후 multi-word를 탐색한다.
6. JJ이후 단어의 tag가 NN일 경우 현재까지 용어를 핵심어로 추가하고 저장한 후 다음 단어를 파악한다.

위와 같은 절차를 통해 핵심어 후보들을 추출하고 핵심어 추출과정중 multi-word Term이 생성된 경우 multi-word Term을 구성하는 single Term(1)과 multi-word Term(2,3) 각각을 모두 추출하도록 한다. 예를 들어 "Recombinant DNA"와 같은 경우는 다음과 같이 처리하도록 한다.

"Recombinant DNA" = [DNA, Recombinant DNA]

핵심어 추출 후 빈도수를 이용한 필터링을 거쳐 핵심어를 선정한다. [표 3-3]은 논문에서 사용할 핵심어 필터링에 대하여 나타내고 있다.

[표 3-3]핵심어 필터링(Term filtering)

Single Term	frequency(Single Term) ≥ 4
Multi Term	1<Multi Number <5

single Term인 경우에는 발생 빈도수가 4회 이상인 경우를 핵심어로 판단하였다. multi-word인 경우에는 1어절 이상이며 4어절 이하로 구성된 multi-word인 경우만을 핵심어로 판단하였다. multi-word term의 제한은 태깅의 오류와 본문추출 과정에서 wikipedia에 존재하는 표, 그림에 대한 주석들, 고유명사로 tagging된 목차, reperence의 고유명사들이 noise를 발생시켰다고 판단하여 다수의 어절을 가지는 multi-word는 제외하였다.

[표 3-4] single Term

Term	freq	Multi number
DNA	269	1
base	73	1
strand	69	1
sequence	69	1
protein	48	1
information	41	1
RNA	40	1
gene	32	1
structure	31	1
chromosome	29	1
enzyme	26	1
helix	25	1
transcription	25	1
cell	25	1

[표 3-4]는 위키피디아의 DNA문서 중 300개의 문장에 대한 single Term에 대해서 필터링을 적용한 후 총 850개의 핵심어중 빈도수 내림차순에 의한 상위 14개의 결과를 보이고 있다. [표 3-5]는 빈도수 내림차순으로 정리한 multi-word Term에 대한 결과이다.

[표 3-5] multi-word Term

Term	freq	Multi number
double helix	10	2
DNA replication	10	2
hydrogen bonds	9	2
genetic information	8	2
DNA strands	7	2
DNA sequence	7	2
base pairs	6	2
transcription factors	5	2
DNA nanotechnology	4	2
DNA-binding proteins	4	2

결과의 성능평가를 위해 위키피디아의 Term Extraction문서에 기술된 external link로 등록되어 있는 Translated LAB의 term extraction 도구와 비교평가를 수행하였다.

d. Term extraction의 결과비교

Translated lab의 term Finder와의 비교 평가를 위해 위키피디아의 "DNA 문서에서 추출한 문장으로부터 추출한 결과를 비교대상으로 선정하였다. Translated lab term Finder[6]는 대량의 문서집합으로부터 Glossary를 생성하기 위해 Translated lab에서 만든 도구로 문서상에서 키워드 측정평가를 통해 Google등의 검색엔진보다 향상된 검색 결과를 보였다. Term Finder는 전문용어 추출을 위해 Poisson 통계를 따르며 Maximum Likelihood Estimation과 약 100만 단어 이상을 포함하는 문서들의 Inverse Document frequence를 이용한다. 이를 통해 계산된 상위 20개의 전문용어를 보여준다. 전문용어와의 비교함으로써 추출된 핵심어가 얼마나 근접하였는지를 파악하기 위해 결과를 비교해 보았다. 결과는 [표 3-6]과 같다. 핵심어에 존재하는 단어는 O표시 하였고 존재하지 않으나 핵심어에 일부 단어가 존재하는 경우에는 해당 단어를 표기 하였다. Term Finder를 기준으로 비교 결과 약 70%의 정확율을 보였다.

[표 3-6] Term Finder와의 비교

Term Finder	핵심어
hydrogen peroxide produce	hydrogen peroxide
including dna replication	dna replication
lambda repressor helix-turn-helix transcription	O
repressor helix-turn-helix transcription factor	O
regulating gene expression	gene expression
pyrene diol epoxide	O
pentose sugar ribose	O
pentose five-carbon sugar	O
double helix	O
dna x-ray diffraction	O
high-energy electromagnetic radiation	O
dna supercoil dna	O
methylated cytosines	O
codons signifying	X
artificial nucleic acid	nucleic acid
imprinting transcriptional	X
cytosine methylation	O
bind single-stranded	O
ethidium bromide	O
single-stranded telomere dna	O

본 절에서는 핵심어를 추출하는 과정까지를 다루었다. 다음 절에서는 핵심어가 포함된 문장을 선별하고 긴 문장에 대해서는 link grammar에서 분석할 수 있는 형태로의 문장 분할에 대해서 다룬다.

2. 핵심어가 포함된 문장 추출

핵심어는 문장내의 정보를 다수 포함하고 있는 용어이다. 따라서 핵심어가 포함된 문장은 문서에서 표현하고자 하는 정보들을 많이 포함하고 있다고 판단 할 수 있다. 하지만 핵심어가 포함된 문장은 최종 목적인 Triple 추출에 복잡성을 생성한다고 판단하여 제거 하였다. 본 절에서는 핵심어를 이용한 문장 추출과 긴 문장으로 인해 의미의 복잡성과 의미파악의 어려움을 해소하기 위해 어휘매핑 기반의 문장 분할에 대해 다룬다.

a. 핵심어를 이용한 문장추출

본 절에서는 이전단계를 거쳐 생성된 핵심어들을 이용하여 핵심어가 포함된 문장을 추출하는 단계이다. 핵심어들을 기존에 전처리 과정을 거친 문장들과 어휘매핑을 통하여 문장들을 추출해낸다. 핵심문장 추출의 목적은 Link grammar Parser를 이용하여 상하위어, 관계 추출을 하기 위함이므로 여러 핵심어들로 인한 문장 중복을 피하도록 한다. 문장 추출을 위한 알고리즘은 다음과 같다.

1. 핵심어 데이터베이스로부터 핵심어를 선택한다.
2. 위키피디아 문서의 사전처리를 거친 문장을 차례로 선택한다.
3. 문장내에서 핵심어와 일치하는 단어가 존재하는지 파악한다.
4. 일치하는 단어가 존재할 경우 현재 문장의 번호를 이전에 파악된 핵심문장 번호가 저장된 배열과 비교하여 중복문장의 여부를 파악한다.
5. 일치하는 번호가 없는 경우 문장번호 배열에 추가하고 있는경우는 다음문장을 탐색한다.
6. 저장된 문장번호와 일치하는 핵심문장을 추출한다.

위의 알고리즘을 이용하여 추출된 문장들은 다음절인 문장 분할 과정을 통해 문장 길이를 제한하여 Link grammar Parser를 통한 분석에 이용할 수 있다.

b. 문장 분할

긴 문장 분석은 높은 복잡도로 인해 기계 번역에서 매우 어려운 문제이다. 특히 본 연구에서 사용하는 Link grammar의 특성은 각 단어마다 1개 이상의 link path를 가지기 때문에 문장의 길이가 길어질 수록 패턴기반의 관계, 인스턴스 추출이 어려워진다. 따라서 추출된 핵심문장의 길이를 조절함으로서 문장이 포함하고 있는 복잡성을 줄이는 과정이 필요하다.

본 연구에서는 문장분할의 기준이 되는 어휘들을 파악하고 이를 바탕으로 문장 내에서 분할 기준어휘 이전의 명사구를 새로운 문장의 주어로 판단한다. 주로 출현하는 어휘들을 파악한 결과 분할의 기준이 되는 단어는 대표적으로 관계대명사, 접속사 등이 있다. 하지만 이 용어들을 곧바로 이전 명사구로 바꾸는 것은 많은 오

차를 발생할 수 있다. 따라서 어휘가 발생하는 특정 패턴을 바탕으로 문장분할을 시도하였다. 본 논문에서는 대명사를 판단하기 위한 anaphora resolution과 같은 세부적인 방법까지는 다루지 않았다. 기존에 연구되었던 패턴들 중 일부만을 이용한다. 특정 어휘들이 발생했을 때 판단할 수 있는 패턴은 [표 3-7]과 같다.

[표 3-7] 문장분할 패턴

패턴	경우
(1)	명사구 + 관계대명사 +verb
(2)	'comma' + 관계대명사 or (while,so)
(3)	It + verb(verb ≠ be동사)

(1): <NN + 관계대명사+verb>

관계대명사 앞 어휘의 tag가 N으로 시작하는 명사 형태이고 관계대명사인 (who, which, it, that)등이 사용되며 동사(verb)가 출현하는 경우에는 관계대명사를 이전 어휘인 명사구로 대체한다. 문장내 어휘발생을 파악한 후 문장을 분할한다.

"protein sequence in a process called translation which depends on the same interaction between RNA" 와 같은 문장이 발생하였을 때, "protein sequence in a process called translation" 과 "translation depends on the same interaction between RNA" 인 2문장으로 분할할 수 있다.

(2): <'comma'+ 관계대명사 or while,so>

'comma' 다음에 오는 관계대명사의 경우 'comma' 이전의 명사구를 대용어로 판단한다.

"DNA helix is duplicated on each strand, which is vital in DNA replication."와 같은 문장이 발생하였을 때, "DNA helix is duplicated on each strand" 문장과 "each strand is vital in DNA replication"로 분할 할 수 있다.

'comma' 다음에 오는 while, so와 같은 경우에는 분할 기준 단어를 삭제하고 문장을 분할 한다.

"DNA binding to a complex of small basic proteins called histones, while in prokaryotes multiple types of proteins are involved."와 같은 문장이 주어졌을 때, "DNA binding to a complex of small basic proteins called histones"와 "in

prokaryotes multiple types of proteins are involved"로 분할된다.

(3): < It + verb(verb ≠ be동사) >

문장의 첫 어절에 It이 사용된 경우에는 It이 가리키는 주어가 이전문장의 주어라고 판단하고 It을 이전문장 주어절의 명사구로 대체하였다. it 이후의 동사가 is와 같은 경우에는 가주어와 같은 역할을 하는 경우를 고려하여 일반 동사인 경우만을 대체어로 사용하였다.

"HIV-1 is thought to have originated in southern Cameroon. It evolved from a Simian Immunodeficiency Virus SIVcpz." 위 문장에서 It은 이전 문장의 HIV-1으로 대체된다.

현재까지는 핵심문장 추출단계로 위키피디아 문서로부터 핵심어를 추출하고 이를 바탕으로 핵심문장을 추출하는 과정까지를 다루었다. 다음 장인 문장 분석 모듈에서는 추출된 핵심문장을 Link grammar Parser를 통해 문장을 구조화하고 이로부터 Triple을 추출하는 과정에 대해 기술한다.

Ⅳ. Link grammar를 이용한 관계 및 개념추출

본 장에서는 본 연구에서 제안하는 방법인 Link grammar의 Link Path의 패턴 분석을 통해 Triple를 생성하고 이를 분류하는 방법에 대해서 기술한다. Link grammar의 특징을 이용하여 기존의 주어-서술어-목적어의 정형화된 트리플 이외에 다양한 관계의 Triple들을 추출한다.

A. Patten기반 Triple 추출

본 절에서는 Link grammar parser를 이용하여 문장을 구조화 하고 Link grammar의 Link관계를 분석하여 Triple을 생성한다. 이를위해 Triple을 정의하고 Link grammar에서 주로 쓰이는 Link 관계에 대해 설명한다.

1. Triple 및 Link 관계정의

본 연구에서 다루는 Triple은 텍스트에서 IS-A 관계의 자동 추출 및 순위화에 기술된 특성을 보인다[13].

1. IS-A Triple (IS-A triple): IS-A 관계를 표현하는 [상위어, 관계표현, 하위어]의 세 개 구성요소를 가진 Triple를 말한다. 본 연구에서 구조화된 문장에서 IS-A 관계를 패턴을 이용하여 Triple을 추출한다. 예를 들어, 문장 “The DNA double helix is stabilized by hydrogen bonds”에서 IS-A 관계를 위한 Triple [be stabilized by, DNA double helix, hydrogen bonds]를 추출한다.

정의 2. 관계 표현 (relational expression): IS-A Triple에서 하위어와 상위어 사이의 의미적인 관계에 대한 언어적인 표현을 말한다. 위의 예에서 관계 표현 “be stabilized by”는 “DNA double helix”와 “hydrogen bonds”사이의 관계를 언어 적으로 표현하고 있다.

정의 3. 관계 인스턴스 (relation instance): IS-A 관계에서 [하위어, 상위어] 쌍을 말한다. 위의 예에서 [DNA double helix, hydrogen bonds]는 한 개의 관계 인스턴스이다. 관계 인스턴스는 IS-A Triple의 후보가 된다.

Link관계는 Link grammar분석에서 단어들 사이의 관계를 표현하는 기호이다. 본 연구에서는 이 Link관계의 순서 및 조합을 통해 Triple추출 규칙을 설정하였다.

[표 4-1] Link grammar의 Link관계

S	주어-동사	O	동사-목적어
M	명사-(전치사구, 분사구문)	MV	동사-전치사구
J	전치사-전치사의 목적어	OF	동사-of
P	be동사-보어	A	형용사-명사
N	조동사-부정어(not)	MX	명사,수식어 ‘,’로 연결된 관계

[표 4-1]는 본 논문에서 주로 이용하기 위한 Link관계들을 나타내고 있다. Link grammar에서 정의한 Link관계는 문장을 완벽히 구조화하기 위해서 대분류로 100여개의 Link관계를 정의하고 있다. 각각의 Link관계마다 세부적으로 3개 이상의 Lowcase Link관계를 가지고 있기 때문에 문장을 자세하게 구조화 할 수 있다.

2. 문장 내 Triple 생성패턴

본 연구에서는 문장에서 Triple를 추출하기 위해 전체 Link관계 중 몇 가지 관계만을 이용한다. Link관계들을 분석한 후 제안한 Link패턴은 [표4-2]과 같다. 주어와 동사가 포함된 S link이외에 형용사-명사관계의 A link와 전치사 구를 수반하는 M link 그리고 다시 전치사의 목적어를 수반하는 J link등의 관계를 통해 Triple을 파악하였다.

Link관계들은 2 단어로 구성되어 있다. 본 연구에서는 링크의 시작단어를 x라 파악하고 링크가 연결된 타 단어를 y라고 파악한다. 예를 들어 S(x,y)에서 x는 주어이고 y는 술어이다.

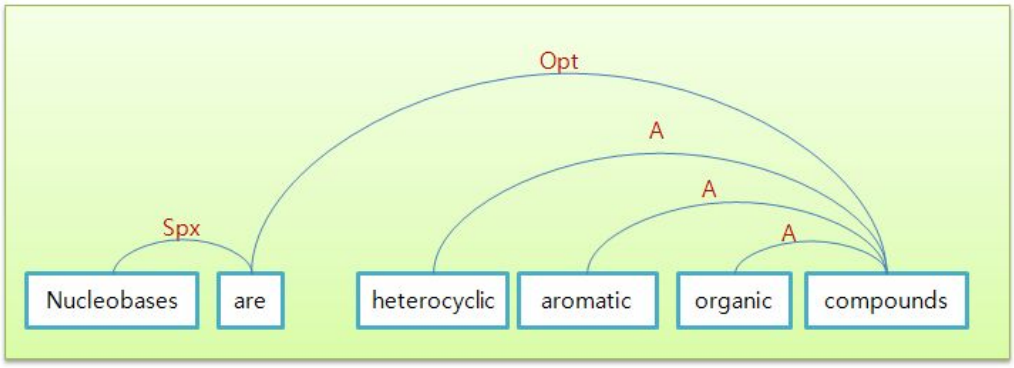
[표 4-2] Link 패턴

link-Path
① S -O (-Mp-J) ,(S-MV -J)
② S -OF-J
③ S -P -MV- O, (J), (Mp-J)
④ if x ⊂(S ∩ Mp) then select Mp-link
⑤ ifx ⊂(S ∩ MX)): 주어-S(x), MX(y)
⑥ A-Mp-J-(Mg)Mv-O(OF-J) , A-Mv-(MV)-J, A-(Mv)Mg-O(OF-J)
⑦ S(if y has I and N(y∈ S) -N -I -O(MV -J)

(1). S -O (-Mp-J) 패턴

S-O link는 가장 기본적인 형태로 주어 동사 목적어형태의 단순한 구조이다. O-link의 y(y∈O)에 해당하는 단어가 전치사 구를 수반하는 M Link를 가지고 있다면 다시 전치사구의 목적어인 J link 까지를 허용하며 O링크의 y부분이 목적어를 수식하는 A/AN링크를 가지고 있거나 고유명사를 나타내는 G링크, 수량을 나타내는 Dmcn 링크를 가지고 있다면 목적어의 인스턴스로 포함시킨다. 또한 목적어가 오지 않고 서술어에 전치사 구가 오는 경우도 파악하기 위하여 S-MV-J패턴도 고려하였다. "Nucleobases are heterocyclic aromatic organic compounds"라는 문장이 주어졌을 경우 Link grammar Parser를 이용한 문장 구조화는 [그림 4-1]과 같다.

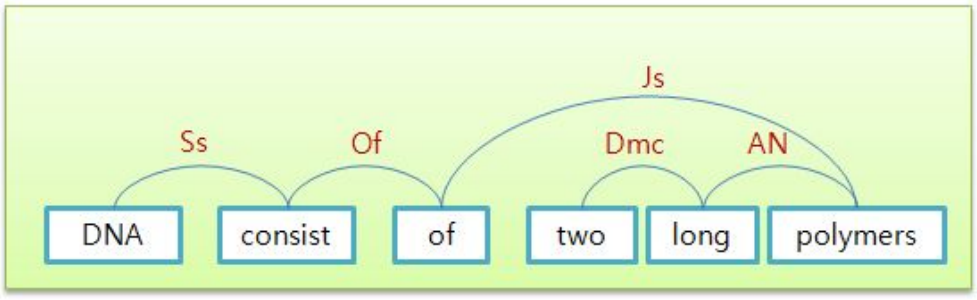
구조화된 문장의 링크 순서에 따라 Spx(Nucleobases, are), Opt(are, compound), A(heterocyclic, compound), A(aromatic, compound), A(organic, compound) link들이 생성되고 S-O 패턴을 통해 (Necleobases, are, heterocyclic aromatic organic compounds)의 Triple를 추출 할 수 있다.



[그림 4-1] S-O link 패턴

(2). S-OF-J 패턴

본 패턴은 구조화된 문장으로부터 of전치사를 필요로 하는 동사들로부터 전치사의 목적어를 파악하여 Triple을 추출하기 위해 설정하였다. of전치사를 취하는 동사인 consist of, made of, proud of 등의 관계와 인스턴스인 of와 연결된 J Link를 통해 파악할 수 있다. [그림 4-2]에서는 "DNA consists of two long polymers of simple units called nucleotides"라는 문장에 대한 구조화를 보이고 있다.

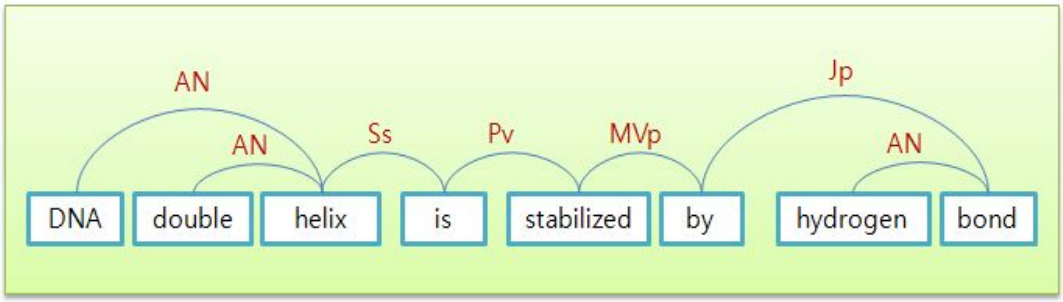


[그림 4-2] S-OF-J link 패턴

위 문장의 Link관계는 Ss(DNA, consists), OF(consists, of), J(of, polymers), A(long, polymers), Mp(polymers,of) 등으로 관계가 형성되며 패턴을 통해 (DNA, consists of, long polymers)의 Triple를 생성한다.

(3). S -P -MV -O, (J), (Mp-J) 패턴

수동태 문장이나 be동사와 보어와의 관계를 통해 Triple을 추출한다. [그림4-3]은 "The DNA double helix is stabilized by hydrogen bonds"문장에 대해 구조화된 결과를 나타내고 있으며 다음 Link관계를 얻을 수 있다.

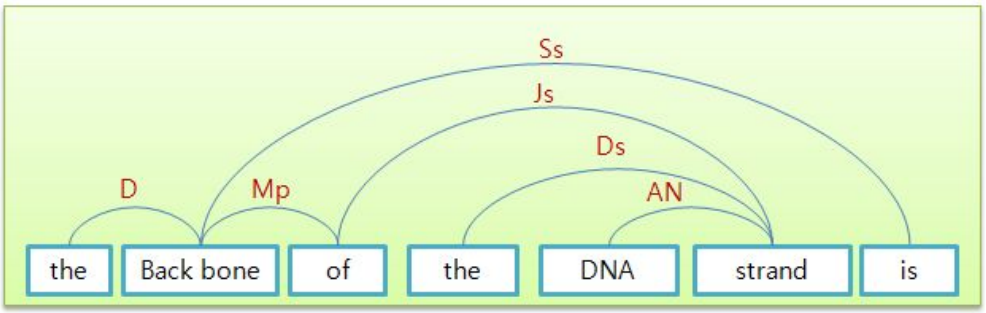


[그림 4-3] S-P-MV-J link패턴

S(DNA double helix, is) , Pv(is, stabilized), MV(stabilized, by), J(by, bond), AN(hydrogen, bond)를 얻을 수 있고 이를 통해 Triple인(DNA double helix, is stabilized by, hydrogen bond)를 추출 할 수 있다.

(4). if x ⊂(S ∩ Mp) then select Mp-link 패턴

Mp는 [표 4-1]의 M link의 한 종류로 명사와 이를 수식하는 전치사 구를 표현하는 Link관계이다. 1,2,3번의 link 관계들은 S link(주어-동사)를 파악한 후에 서술어를 바탕으로 한 목적어를 찾고 있지만 S link가 전치사구를 포함하는 형태로 이루어질 수 있다. 이러한 경우에는 S link의 주어가 가지는 Mp link를 따라서 주어를 생성한다. 전치사와 연결되어 있는 Mp는 주로 전치사구 끝까지가 의미가 형성되는 경우가 많다. 따라서 S link 내에서 link가 끝나는 단어까지를 주어로 한다. 아래에서는 strand가 이에 해당한다. [그림 4-4]는 "the backbone of the DNA strand is made from alternating phosphate and sugar residues" 문장의 일부를 Parsing한 관계를 나타내고 있다.

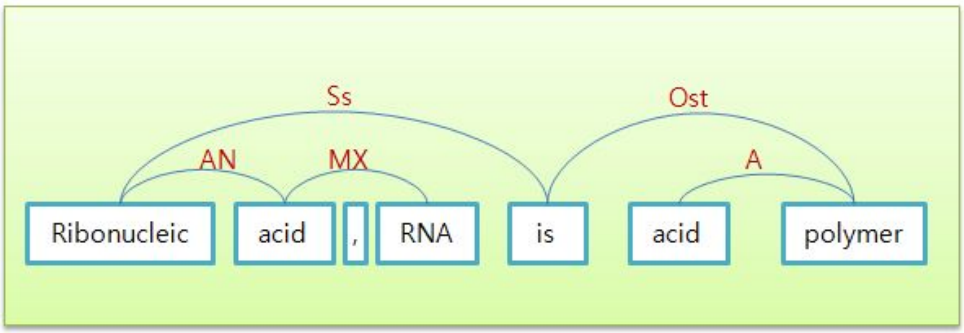


[그림 4-4] (S ∩ Mp) link패턴

위 문장에서 주어는 S link만을 따라갈 경우에는 주어는 backbone이다. 그러나 문장의 정확한 주어는 "the backbone of the DNA strand"이다. 따라서 S(x,y)의 x가 Mp link를 가지고 있는 경우에는 Mp Link를 선택하는것이 올바른 주어가 된다.

(5). if x ⊂(S ∩ MX)): 주어S(x), MX(y) 패턴

주어의 단어가 'comma'를 기준으로 다른 단어를 열거 하고 있다면 주어와 대등한 관계라 파악하고 각각의 주어에 대해 Triple을 추출하였다.

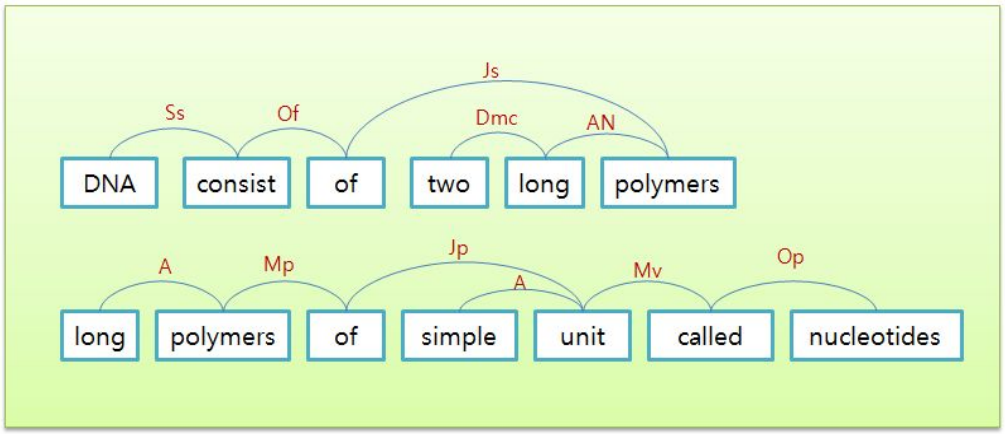


[그림 4-5] MX link를 포함하는 link패턴

[그림 4-5]에서는 "Ribonucleic acid, or RNA, is a nucleic acid polymer" 라는 문장의 구조화 결과를 대략적으로 나타내고 있다. 결과를 통해서 (Ribonucleic acid, is, acid polymer), (RNA, is, acid polymer) 두 개의 Triple을 추출한다.

(6). A-Mp-J-(Mg)Mv-O(OF-J) , A-Mv-(MV)-J, A-(Mv)Mg-O(OF-J) 패턴

5번까지의 각 패턴들이 S-link를 바탕으로 Triple을 시작하였다. 이번 패턴은 문장내의 타 Link를 바탕으로 Triple를 추출하기 위한 패턴이다. 전치사구나 분사를 수반하는 명사를 주어(인스턴스)로 추출하고 분사유형의 단어들을 서술어(관계)로 하는 Triple를 추출하는 조건들을 나타내고 있다. [그림 4-6]은 "DNA consists of two long polymers of simple units called nucleotides" 문장을 구조화 하고 있고 2번 패턴에 의해(DNA, consist of, long polymers)Triple을 추출할 수 있고, A[long polymer] Mp[polymers of] J[of units] A[simple units] Mv[units called] Op[called nucleotides]로부터 (long polymers of simple unit, called, nucleotides)인 Triple를 생성할 수 있다.



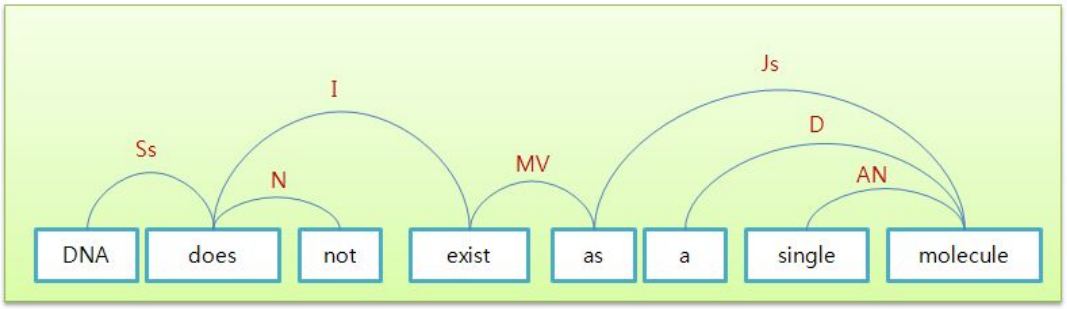
[그림 4-6] S link와 Triple생성 패턴

(7). S(if y(y∈ S) has I and N then S -N -I -O(MV -J) 패턴

본 패턴은 주어의 서술어가 부정어를 포함하고 있는 경우를 위한 조건으로 서술어 다음에 오는 부정어 관계 N, 부정사 관계 I를 통해서 본래 목적의 서술어를 찾기 위한 패턴을 나타낸다.

I : 부정사의 동사형태를 조동사 또는 "To"와 연결짓는 Link 관계이다. 예를 들어"you MUST DO it", "I want TO DO it"에서 대문자로 표기된 부분과 같다.

N : 조동사와 "not"의 연결을 뜻한다.



[그림 4-7] 부정어가 포함된 link패턴

[그림 4-7]은 “In living organisms DNA does not usually exist as a single molecule”에 대한 문장 구조화를 나타내고 있으며 [그림 4-7]의 link관계를 살펴볼 때, S의 동사는 does이지만 dose는 조동사로 Triple로는 부적합 하다. 의미를 갖는 관계는 I link로 연결된 exist이다. 따라서 부정어가 포함된 문장에서 Triple를 추출하기 위해 S(y)가 N-link를 가지는 경우 S(x)를 주어로 가지고 S(y)가 가지는 N(y)과 I(x) link를 관계로 파악하여 Triple를 생성한다.

[표 4-3]에서는 위 패턴을 통해 Triple를 추출한 결과를 보이고 있다.

[표 4-3] 패턴에 의해 생성된 Triple

패턴	Term(subject)	Relation(predicate)	Term(object)
(1)	Nucleobases	are	heterocyclic aromatic organic compounds
(2)	DNA	consist of	two long polymers
(3)	DNA double helix	is stabilized by	hydrogen bond
(4)	the backbone of the DNA strand	is made from	alternating phosphate
	the backbone of the DNA strand	is made from	sugar residues
(5)	Ribonucleic acid	is	acid polymer
	RNA	is	acid polymer
(6)	long polymer of simple unit	called	nucleotides
(7)	DNA	does not exist as	a single molecule

본 절에서는 Link grammar parser를 통해 구조화된 문장으로부터 Triple 추출을 위한 패턴을 정의하였고 문장으로부터 각 패턴에 따른 Triple들을 추출하였다. 다음 절인 Triple순위화 모듈에서는 Triple들에 가중치를 부여하여 중요도가 높은 순

으로 순위화한다.

B. Triple 순위화

Link grammar 패턴을 통해 추출한 Triple들은 검증되어 있지 않기 때문에 Triple의 속성들은 noise를 포함하고 있을 확률이 높다. 따라서 각 Triple들을 중요도가 높은 순으로 Triple을 순위화 할 필요가 있다.

1. 상호정보량 가중치 부여

본 절에서는 관계의 상호정보량을 이용하여 순위를 파악한다.

$$PMI(X,Y) = \log_2(\frac{p(X,Y)}{p(X)p(Y)})$$

Pointwise Mutual Information(PMI)는 정보 이론이나 통계학 분야에서 두 사건의 관련성을 측정을 위해 사용되는 방법으로서 X와 Y의 연관성이 높을수록 높은 값을 가지고 연관성이 적을 수록 낮은 값을 갖게 된다.[18] 본 논문에서는 추출된 Triple의 서술어가 다양한 상,하위어들과 함께 나타날수록 신뢰도가 높은 관계를 형성하고 있다고 판단하였다.

$$PMI(x,y)_{(y\in RList, x\in CList)} = \log(1+\sum_{x=1}^{n}(\frac{p(x\cap y)}{p(x)p(y)})) \quad (1)$$

(1)은 Triple의 각 관계(서술어)들에 MI가중치를 부여하기 위한 수식이다. 위 수식에서 x는 상,하위어 집합(CList)의 원소이며 y는 서술어집합(RList)의 원소를 나타낸다. 한 서술어와 다른 Triple의 모든 인스턴스간의 상호정보량을 취합한 값이 해당 Triple의 가중치로 이용된다. 1을 더한 값은 1번씩 발생하는 단어는 log0으로 취급되어 발생하는 계산상의 오류를 막기 위함이다.

PMI의 가장 큰 문제점은 전체 Triple에서 x,y가 드물게 발생 할수록 높은 값을 가지는 것이다. (2)에서는 이를 보완하기 위해 정보량을 측정할 각 단어의 빈도수

가 2이상일 경우만 값을 취하고 그렇지 않으면 0을 입력하였다.

$$freq(x) < 2 \text{ and} freq(y) < 2 : \ T = 0$$
$$T = (\frac{p(x \cap y)}{p(x)p(y)}) \qquad (2)$$

위키피디아의 DNA페이지에서 추출한 Triple을 PMI가중치를 통해 순위화한 후 I 관련 인스턴스와 함께 추출한는 것을 [표 4-4]에서 보이고 있다.

[표 4-4] PMI 가중치

Relation	PMI	Instance-Related	
bind	8.39231742278	DNA-binding proteins	single-stranded DNA
provid for	8.39231742278	double-stranded structure of DNA	DNA replication
curl in	8.39231742278	single-stranded DNA	long circle
is stabilize by	8.39231742278	DNA double helix	hydrogen bonds
read	8.13014150852	ribosome	RNA sequence by base-pairing
function in	7.80950019389	DNA polymerases	large complex
play in	7.61777354253	non-coding DNA sequences	chromosomes
read	7.49226760432	ribosome	messenger RNA
bind to	7.19147053172	transcription factor	particular of DNA sequences
use	6.9844184588	chromosome ends	enzyme telomerase

2. 인스턴스를 이용한 가중치 부여

PMI 를 통한 관계에 가중치를 부여한뒤 Triple에서 각 상,하위어들이 발생 확률이 높을 수록 중요 단어라고 판단하여 Term Frequency를 구하고 핵심어로 선정되었던 단어를 인스턴스가 포함하고 있을 경우 가중치를 부여한다.

$$Triple(x) = \sum_{x=1}^{N} \frac{freq(x)}{T}_{x \in Triple,\ T = CList} \qquad (1)$$

(1)은 각 Triple의 상,하위 인스턴스들의 Term Frequency를 구한다[9]. CList는 각 Triple의 모든 인스턴스(상,하위어)들의 집합이며, freq(x)는 해당 인스턴스의 빈도수를 나타낸다.

$$Triple(x) = \sum_{x=1}^{N} Md(x)_{x \in instance,\ Md(x) \in [0,1]} \qquad (2)$$

(2)는 각 인스턴스들과 핵심어 데이터베이스와 매칭하여 인스턴스 단어가 핵심어로 존재하는 경우에 가중치를 부여하였다. Md(x)는 인스턴스와 관련된 핵심어의 Term Frequency이다. 인스턴스가 핵심어를 내포하거나 핵심어로 존재하는 경우 모두 핵심어의 TF 가중치를 부여한다.

[표 3-4]에서는 Triple내 각 인스턴스들의 TF값과 Md(x)값을 관련된 각 Triple에 부여하여 순위화 한 결과를 보이고 있다. Related relation은 해당 Triple의 인스턴스들과 관련된 관계이다.

[표 4-5] TF 가중치

Related relation	TF	Md(x)
can bind to	0.0626865671642	0.00233918128655
organize	0.0582089552239	0.00233918128655
cut	0.0477611940299	0.00701754385965
copy into	0.0507462686567	0.00350877192982
consist of	0.0477611940299	0.0046783625731
is	0.0477611940299	0.0046783625731
organize	0.0477611940299	0.00350877192982
compact	0.0477611940299	0.00350877192982
is organ into	0.0462686567164	0.0046783625731

위의 가중치 측정결과를 총 도합하여 Triple을 순위화 한다. 순위화된 Triple들은 관계와 인스턴스를 온톨로지 형식에 맞게 정의한 후 시각화 한다.

본 장에서는 핵심문장들을 Link grammar Parser를 이용하여 구조화한 후 패턴을 바탕으로 Triple을 추출하였다. 그리고 중요한 Triple순으로 순위화 하기 위해서 PMI를 이용한 측정방법과 TF, 핵심어 데이터베이스를 이용한 가중치를 취합하여 순위화를 하였다. 다음 장에서는 온톨로지 확장을 위해 Triple들을 온톨로지 형식에 맞게 정의하고 이를 시각화 하는 방법에 대해 기술한다.

V. 온톨로지 설계 및 Visualization

학습을 통해 추출된 Triple을 온톨로지 확장에 이용하기 위해 온톨로지 형식에 맞게 정의할 필요성이 있다. 본 논문에서는 이를 위해 Triple의 인스턴스에 대한 클래스-속성관계를 미리 정의하고 관계에 대한 분류를 정의 하였다.

A. 인스턴스, 관계 정의

1. 인스턴스 정의

온톨로지 확장에 적용하기 위해서 인스턴스들에 대한 속성을 정의하고 이에 대한 분류를 하였다.

[표 5-1] 인스턴스 속성 정의

상위 클래스	하위 클래스	속성
Physical_entity	Source	
	Source-natural	
	organism	microorganism, Virus, Tissue, Cell component, Other Organism
	Substance	
	Substance-Compound	
	Amino_acid	Protein, peptide, Other Amino_acid
	Nucleic_acid	DNA, RNA, polynucleotide, Nucleotide, Other Nucleic_acid
	Lipid	steroid, Other lipid
	carbohydrate	type of carbohydrate
	Substance-Atom	type of Atom
psychology_ entity	Symptom	type of Symptoms
	Syndromes	type of Syndromes
Property_entity	Dynamics property	Activity type
		Expression type
	Location property	Location type
	Amount property	Amount type
	Function Property	Function type
		Signal type

본 연구에서는 [표 5-1]과 같이 인스턴스 분류 기준을 선정하였다. biology 도메인을 기준으로 분류 하였으며 각 클래스의 속성들과 매칭하여 인스턴스의 상, 하위간 위치를 파악할 수 있다. 최상의 클래스로부터 상위클래스인 physical, psychology, property entity로 나누고 각각의 하위클래스와 속성을 생성하였다. 인스턴스 DNA는 [표 5-1]에서 Nucleic_acid의 부류임을 알 수 있다. 그러나 표에 기술된 속성만으로는 인스턴스 전체에 대한 분류가 불가능하다. 이를 위해 위키피디아의 infobox와 navbox를 참조하여 개념에 대한 분류를 시행한다.

[표 5-2] HIV의 Infobox 속성

Human immunodeficiency virus	
Virus classification	Group: Group VI (SsRNA-RT virus)
	Family: Retroviridae
	Genus: Lentivirus
Species	Human immunodeficiency virus 1
	Human immunodeficiency virus 2

[표 5-2]는 Wikipedia 문서중 HIV의 infobox의 속성을 나타낸다. infobox중 개념의 속성을 파악하기 위해 infobox tag들 중 table head(th)와 head내 (td)tag의 text만을 이용한다. 위 [표 5-2]에서는 좌측이 table head이고 우측 부분이 td tag의 text이다. infobox의 정보를 통해 HIV는 virus에 속하는 분류임을 알 수 있다.

infobox table class가 존재하지 않는 단어에 대해서는 nav-box table class를 이용하여 속성을 판단하도록 한다.

navbox에서는 관련단어들이나 구성원, 발생지역등 해당용어와 관련되지 않는 tag들(title tag, 본문내용, link tag)이 다수 존재하여 noise로 작용하고 있기 때문에 파악이 쉽지 않다. 하지만 navbox내에서 해당 용어에 대한 표현은 strong class로 강조를 하고 있다. 따라서 본 연구에서는 strong class tag와 내부의 title tag들을 통해 찾고자 하는단어의 noise를 최소화 하는 방향으로 개념 파악을 수행한다.

[표 5-3]은 DNA의 navbox를 나타내고 있다. 표를 통해 DNA의 정확한 표현은 Deoxyribonucleic acid임을 파악할 수 있고 DNA는 nucleic_acid type임을 알 수 있

다.

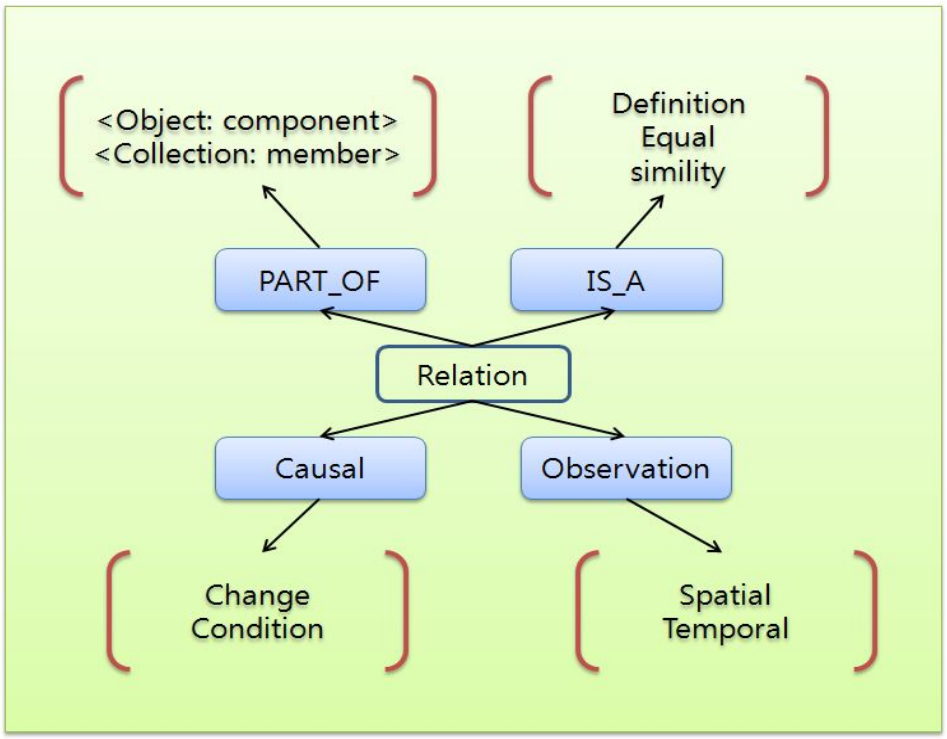
[표 5-3] DNA의 navbox 속성

<table class= "navbox">	<td class="navbox-group">	<td class= "navbox-list">
type of nucleic acids	Deoxyribonucleic acids	Complementary DNA
		CpDNA
		GDNA
		Multicopy single-stranded_DNA
		Mitochondrial DNA

본 절에서는 Triple의 인스턴스들의 도메인 온톨로지에 적합한 속성을 정의하기 위해 어휘매칭과 위키피디아의 infobox와 navbox를 이용하였다. 다음 절에서는 Triple의 관계들에 대한 분류를 행한다.

2. 관계 정의

추출된 관계들은 온톨로지 확장을 위해 온톨로지에서의 상하위 계층을 정의한다. 관계성 파악은 기존의 온톨로지[16][17]를 분석하여 미리 관계들의 계층을 파악하였다. [그림 5-1]은 미리 선정된 관계 분류를 보이고 있다.



[그림 5-1] 관계도

IS -A, PART_OF, Causal, Observation으로 분류하고 각 계층마다 세부분류를 정의하고 있다. Triple의 서술어 부분과 정의된 각 세부 분류 관계를 어휘 매칭하여 Triple의 관계를 정의한다.

[표 5-4]는 분류된 상하위 관계와 해당관계에 포함 가능한 용어들을 나타내고 있다.

[표 5-4] 관계 속성 정의

상위 클래스	하위 클래스	속성	
IS_A	define	is, called, known, define, identify	
	equal	equal, encode, corefer, compare	
	simility	similar, sqsimilar, stsimilar, fnsimilar	
PART_OF	Object:Component	involve, F-contain, substructure, contain, part, mutualcomplex	
	collection:member	member, kind, type, consist	
Causal	cause, participate		
	change	mutual-affect, affect, interact, provide, make, use	
	change-physical	depolymerize, cleave, disrupt, unbind, disassemble	
	change-physical	Modification	modify, add, acetylate, dephosphorylate, remove
	change-physical	Assemble	attach, cross-link, polymerize, assemble, bind
	change-dynamics	inactivate, halt, inhibit, downregulate, suppress, form	
	change-amount	increase, decrease	
	change-location	localize-to, localize, locate	
	condition	condition, trigger, control, modulate, read	
Observation	corelate, coregulate, transition		
	spatial	localize, coprecipitate, presence, absence, within	
	Temporal	coexpress, cooccur	

[표 5-4]에서 각 관계들은 최상위 클래스와 그 하위 클래스 그리고 속성으로 표현되어 있다. Triple을 통해 추출된 각 관계들을 위 표의 속성들과 매칭하여 일치하는 속성의 클래스를 따른다. 일치하지 않는 관계에 대해서는 중심단어에 대해 매칭한다. 즉 formed by와 같은 경우는 formed의 기본형인 form을 통해 causal -change -dynamic의 관계라고 판단할 수 있다.

본 절에서는 관계분류를 통해서 온톨로지 형식을 설정할 수 있으며 이를 바탕으로 다음절에서 메타데이터를 정의한다.

B. 메타데이터 생성

본 절에서는 이전 단계를 통해 얻어진 개념과 관계들을 이용하여 메타데이터를 형성하는 단계이다. 메타데이터를 작성하는 단계는 가장 먼저 온톨로지 형식과 관계 및 인스턴스를 정의한다. 미리 선정의한 관계 및 인스턴스 분류를 메타데이터 형식으로 표현하는 과정이다.

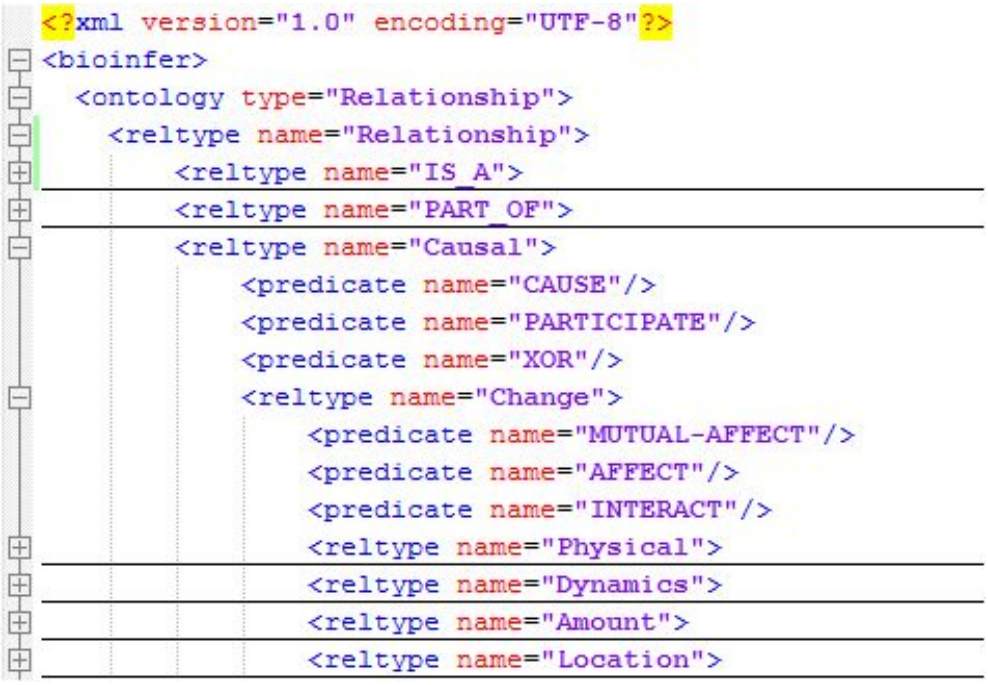
```
<?xml version="1.0" encoding="UTF-8"?>
<bioinfer>
  <ontology type="Relationship">
  <ontology type="Entity">
  <sentences>
</bioinfer>
```

[그림 5-2] 메타데이터 형식 정의

[그림 5-2]는 메타데이터의 형식과 관계, 인스턴스, 문장에 대해 나타내고 있다. 관계와 인스턴스는 각 클래스의 상하위 관계들과 속성들을 기술하였으며 위키피디아에서 추출된 핵심문장에서 추출된 Triple들의 관계를 시각화 하기 위해서 문장의 구조화와 생성된 Triple들의 관계를 기술한다.

1.관계, 인스턴스 생성

본 절에서는 이전까지 기술되었던 관계와 인스턴스들을 메타데이터상에 정의하는 부분이다. 관계분류는 [표 5-4]의 분류 형태로 메타데이터를 기술한다. [그림 5-3]은 관계에 대한 기술형태를 보이고 있다. 관계에는 하위 클래스로 IS_A, PART_OF, Causal, Observation이 있으며 각각의 하위 관계를 reltype으로 표현하고 속성을 기술할 때는 predicate로 표현하였다. 시각화 프로그램에서는 predicate name과 추출된 Triple의 서술어를 매칭 하여 인스턴스간의 속성을 파악하도록 한다.



[그림 5-3] 메타데이터(관계 정의)

인스턴스는 Triple내 서술어로 결합된 개체로 인스턴스 클래스의 상하위 관계를 공백을 기준으로 표현하였으며 "entitytype name"을 통해서 분류하였다. 인스턴스의 기술내용은[그림 5-4]같다. [표 5-1]의 순서에 맞게 인스턴스의 특성을 기술하며 entitytype name과 매칭하여 인스턴스의 속성을 파악한다.

```
<ontology type="Entity">
  <entitytype name="Entity">
      <entitytype name="Physical_entity">
          <entitytype name="Source">
              <entitytype name="Natural">
                  <entitytype name="Organism">
                      <entitytype name="microorganism">
                      </entitytype>
                      <entitytype name="Virus">
                      </entitytype>
                      <entitytype name="Tissue">
                      </entitytype>
                      <entitytype name="Cell_component">
                      </entitytype>
                      <entitytype name="Other_Organism">
                      </entitytype>
                  </entitytype>
              </entitytype>
          </entitytype>
          <entitytype name="Substance">
```

[그림 5-4] 메타데이터(인스턴스 정의)

다음은 문장에 대한 기술과 문장의 구조화를 바탕으로 시각화를 통해 보여지게 될 링크생성과 Triple의 인스턴스와 관련성 파악에 대해 살펴본다.

2. Triple 내 Link와 관계생성

본 절에서는 정의된 온톨로지 형식을 바탕으로 Triple이 추출된 문장에 대해서 시각화를 통해 분석내용을 보이기 위해 Triple이 추출된 문장과 Triple생성 링크, 그리고 시각화에서 사용할 관계와 속성을 파악하기 위한 부분에 대해 기술한다.

[그림 5-5]에서는 토큰화된 문장을 메타데이터에 기술하는 과정을 보이고 있다. 각 문장에서 Link grammar parser를 통해 생성되는 Triple을 보여주기 위하여 가장먼저 문장을 토큰화 한다. 단어별로 토큰과 단어의 시작 위치를 각각 기술한다. 기술된 단어를 바탕으로 Triple의 관계와 Link를 생성한다.

```
<sentence id ="4" origText="The double-stranded structure of DNA
    <token id="t.4.0" charOffset="0">
        <subtoken id="st.4.0.0" text="The"/>
    </token>
    <token id ="t.4.1" charOffset="4">
        <subtoken id="st.4.1.0" text="double-stranded"/>
    </token>
    <token id ="t.4.2" charOffset="20">
        <subtoken id="st.4.2.0" text="structure"/>
    </token>
    <token id ="t.4.3" charOffset="30">
        <subtoken id="st.4.3.0" text="of"/>
    </token>
    <token id ="t.4.4" charOffset="33">
        <subtoken id="st.4.4.0" text="DNA"/>
```

[그림 5-5] 메타데이터(문장 토큰화)

문장에 대한 파싱이 끝난 후 Link grammar parser를 통해 생성된 Triple의 속성 파악과 관계 파악을 수행한다. [그림 5-6]은 Triple의 관계와 속성을 기술하는 과정이다.

```
<entity id="e.1.0" type="Protein">
  <nestedsubtoken id="st.1.0.0"/>
  <nestedsubtoken id="st.1.1.0"/>
</entity>
<entity id="e.1.1" type="RELATIONSHIP_TEXTBINDING">
  <nestedsubtoken id="st.1.3.0"/>
</entity>
<entity id="e.1.2" type="DNA">
  <nestedsubtoken id="st.1.4.0"/>
  <nestedsubtoken id="st.1.5.0"/>
</entity>
```

[그림 5-6] 메타데이터(인스턴스간 관계정의)

Link grammar parser를 통해 판단된 Triple의 인스턴스의 속성을 기술하고 인스턴스, 이를 연결하는 관계를 개념화 하여 기술한다. 'nestedsubtoken'은 문장내에서 인스턴스와 관련된 subtoken을 나타내며 entity type은 인스턴스를 미리정의한 분

류표와 매칭하여 파악된 관계를 기술한다. 'RELATIONSHIP_TEXTBINDING'은 Triple의 인스턴스들을 연결하는 관계를 객체화 하고 있다. 관계는 다음 [그림 5-7]을 통해서 속성화 하고 기술한다.

```
<linkages>
  <linkage type="serial">
    <link type="AN" token1="t.1.0" token2="t.1.1"
    <link type="Sp" token1="t.1.1" token2="t.1.3"
    <link type="E" token1="t.1.2" token2="t.1.3"
    <link type="Os" token1="t.1.3" token2="t.1.5"
    <link type="Ah" token1="t.1.4" token2="t.1.5"
  </linkage>
</linkages>
<formulas>
  <formula>
    <relnode entity="e.1.1" predicate="ASSEMBLE">
      <entitynode entity="e.1.0"/>
      <entitynode entity="e.1.2"/>
```

[그림 5-7] 메타데이터(Link와 관계-속성 기술)

구조화 된 문장의 Link관계를 위해 'Linkages type'을 기술하고 'link type'을 통해 link관계의 종류와 이와 관계된 단어들을 기술한다. 'fomulars'에서는 'RELATIONSHIP_TEXTBINDING'으로 기술된 객체에 대해 미리 정의한 관계분류표와 매칭하여 속성을 파악하고 이를 기술 하였다. Triple이 생성된 각 문장에 대해서 메타데이터로 이를 기술하여 메타데이터를 완성한다. 완성된 메타데이터는 다음 절인 시각화를 통해 표현되어진다.

C. 시각화

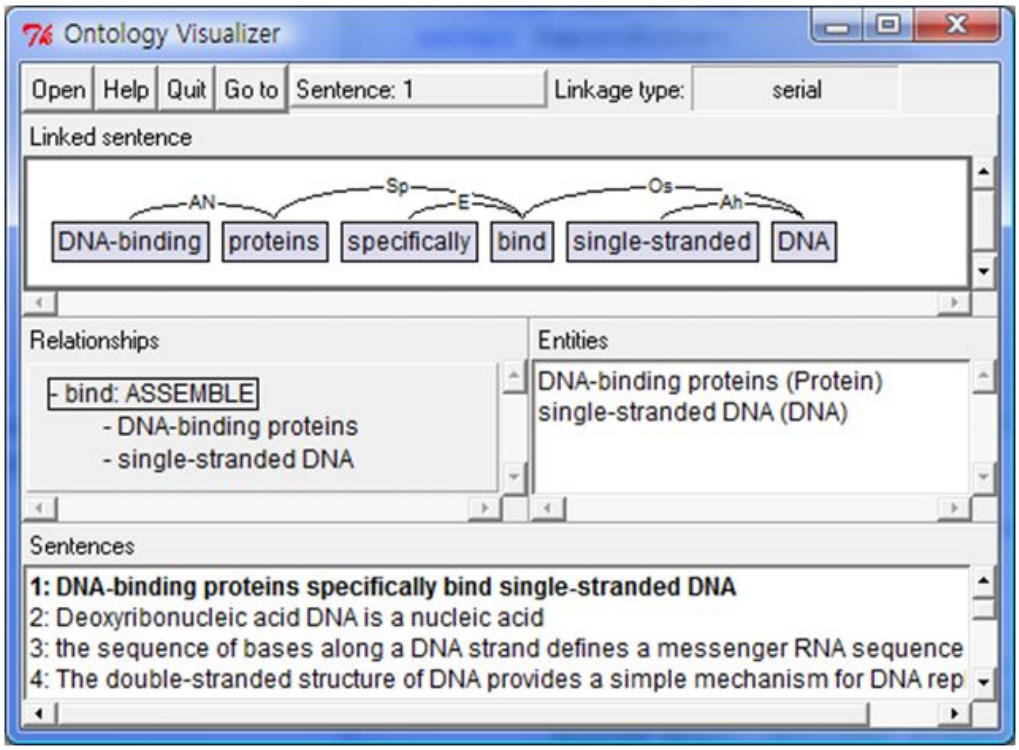
1. 인터페이스

시각화에서는 생성된 메타데이터를 분석하여 이해하기 쉽도록 나타내주기 위한

도구를 설명하는 과정으로 위 작성된 메타데이터를 이용하여 시각화하는 프로그램을 설명하도록 한다. 파이선 tool을 이용하여 생성하였으며 파이선의 그래픽 모듈인 Tkinter를 이용하였다.

시각화의 과정은 다음과 같다. 메타데이터를 파싱하여 관계와 인스턴스, 각 문장별로 토큰과 객체와 메타데이터의 클래스를 넘겨받아 인터페이스에 맞게 구성한다.

[그림 5-8]에서는 시각화 도구를 이용하여 메타데이터 표현하고 있다. 파싱된 각 문장들과 Link들이 'Linked sentence' Label에 나타나고 Triple의 관계와 인스턴스들이 각각 'Relationship'과 'Entities' Label에 표현되었다. 그리고 'sentence' Label을 통해 사용된 문장이 순차적으로 표현된다.



[그림 5-8] 온톨로지 Visualization tool

각 Label에 대한 설명으로 [그림 5-8]에서는 현재 1번 문장에 대한 표현을 나타내고 있다. 위 문장에서 생성되어진 Triple은(DNA-binding proteins, bind,

single-stranded DNA)이며 'Linked sentence'에서는 Link grammar Paser에서의 Link를 통한 Triple생성을 보이고 있다. 'Relationship'에서 관계인 bind와 미리 정의된 온톨로지 형식에 따른 속성을 보이고 있다. 그리고 bind와 관계된 각 인스턴스를 나타내고 있으며 인스턴스들은 우측의 'entities' label에서 각 정의된 온톨로지와 매칭된 속성을 나타낸다. 현재 표현하고 있는 문장은 'sentences'label에서 볼드체로 표현된다. 각 문장마다 1개 이상의 Triple이 생성 가능 하지만 순위화에 따른 Triple별로 생성하였기 때문에 각 문장별로 1개의 관계와 인스턴스들에 대해서 표현하였다.

본 과정들을 통해 도메인 문서로부터 핵심문장을 기반으로 Triple을 생성하고 시각화 하였다. 실험 및 평가에서는 문서에서 생성된 Triple의 정확도와 가중치 판단 알고리즘에 대해 기술하였다.

Ⅵ. 실험 및 평가

실험은 PC상에서 python[24]을 이용하여 구현 하였다. 입력데이터는 wikipedia의 DNA page를 통해 문장들을 추출하였고 Link grammar Parser를 통해 생성된 Triple을 순위화 하고 시각화 하였다.

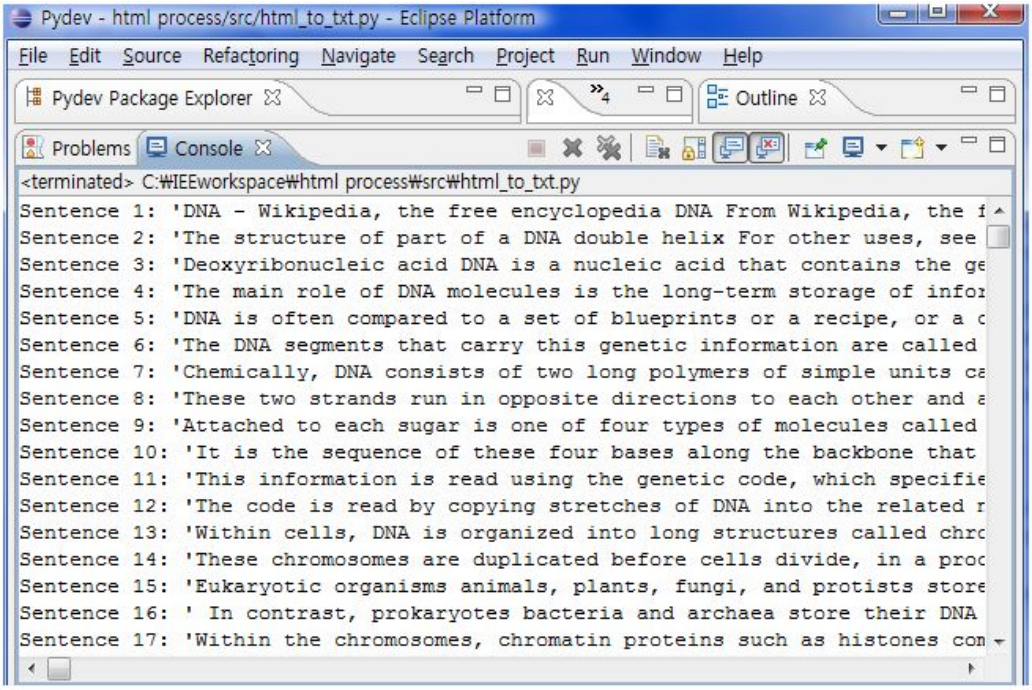
A. 실험 및 응용 방법

위키피디아의 html 문서로부터 tag를 떼어낸 text문서를 얻기 위해 python의 html parser와 Regular expression을 이용하였다. urlib모듈을 통해 html문서를 읽어들인뒤 tag를 제거하고 텍스트화 한다. 생성된 본문은 여러 가지 기호와 주석으로 구조화 되어 있지 않기 때문에 정규식 표현을 거쳐 [표 6-1]과 같이 문장 단위로 분류한다.

[표 6-1] 본문 정규화 알고리즘

```
sentences = nltk.sent_tokenize(data)
// 본문영역을 문장단위로 분할
for sentence in sentences:
    if sentence =='':
        continue
    else:
        Tsen =re.sub("(₩[+₩d+₩])", "", sentence)
        Tsen =re.sub("[^a-z',.₩"₩- A-Z',.₩- 0-9]+", " ", Tsen)
        Tsen = re.sub("[ ₩n]+", " ", Tsen)
        // reperence([0],[1])와 특수문자 (?!, '₩n')를 제거
        sentence_dict[index]=Tsen  //문장 단위로 배열화
```

[표 6-1]의 과정을 통해 생성된 문장들은 [그림 6-1]와 같다.



[그림 6-1] 핵심문장 추출

문장이 생성된 후에는 문장내 핵심어를 추출한다. 본 논문에서의 핵심어는 품사와 빈도수를 기준으로 판단하였다. 품사태깅된 각 단어들을 비교하여 명사로 시작하는 경우 저장하고 형용사로 시작하는 경우는 명사를 기다린다. 따라서 NN형태의 단일어휘와 JJ+ N(형용사+명사), N+N(명사+명사) 형태의 multi-word를 추출한다.

[표 6-2]는 핵심어를 추출하기 위한 알고리즘을 보이고 있다.

[표 6-2] 핵심어 추출 알고리즘

```
if state == SEARCH and tag.startswith('N'): //state가 search이고 tag가 N으로 시작할 경우
    state = NOUN                          // multi-word 용어를 탐색하기 위한 상태
    _add(term, norm, multiterm, terms)  //현재 단어, 정규품을 저장
elif state == SEARCH and tag == 'JJ':   //search이며 tag가 JJ인 경우
     state = NOUN                       // multi-word 용어를 탐색하기 위한 상태
    _add(term, norm, multiterm, terms)
elif state == NOUN and tag.startswith('N'):// 탐색 중이며 tag가 N으로 시작하는 경우
    _add(term, norm, multiterm, terms) // single-term과 multi-word를 각각 생성
elif state == NOUN and not tag.startswith('N'): //multi-word용어가 아닌경우
    state = SEARCH     //상태를 변화
    if len(multiterm) > 1  //현재까지의 용어가 multi-word였는지 판단후
        word = ' '.join([word for word, norm in multiterm]) // 각 배열에 저장
        terms.setdefault(word, 0)  //초기화
```

[그림 6-2]은 위 알고리즘을 통해 생성된 핵심어들을 나타내고 있다. Multi_number는 어휘 수 이며 freq는 발생빈도를 나타낸다.

Term	freq	Multi_number
genetic recombination	2	2
nucleoside triphosphat	2	2
DNA research	2	2
different chromosomes	2	2
Holliday junction	3	2
genetic information	8	2
genetic code	3	2
double-strand breaks	3	2
linear chromosomes	2	2
DNA polymerase	3	2
DNA bases	2	2
base pair	2	2

[그림 6-2] 핵심어 추출

핵심어 추출을 바탕으로 추출된 문장들 중 위키피디아의 reperence 이하 부분은 문장이라 보기 어려워 삭제하고 약 400여개의 핵심문장을 획득하였다. 핵심문장들을 Link grammar parser를 이용하여 Triple을 추출하고 pmi_value와 term_frequency value, Md(x) value를 구하였다. 각 관계가 모든 인스턴스들과 갖는 빈도수를 파악하기 위해 Triple을 재 분해하여 instance, relation, instance-relation배열을 구성하여 가중치를 측정하였다. [표 6-3]은 PMI 가중치를 측정하기 위한 알고리즘을 나타낸다.

[표 6-3] PMI 가중치 생성 알고리즘

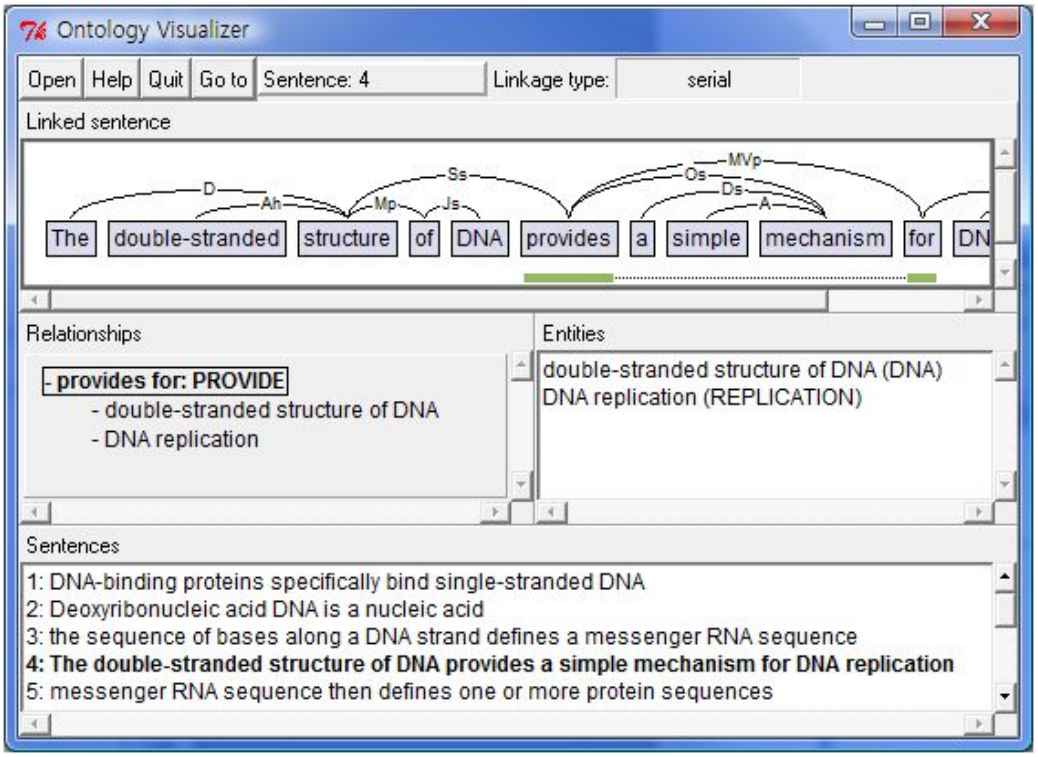
```
for entity in instance_relation:
    y = entity.strip().split(',')      //instance와 relation을 분해
    instance =y[0]; relation = y[1]  //각 단어를 파악
    freq1= fCdist.freq(instance)        // p(instance)
    freq2 = fRdist.freq(relation)      //  p(relation)
    freq3 = finstance_relation.freq(entity) // p(instance ∩ relation)
    if fCdist.get(instance)<2 and fRdist.get(relation)<2:  // 1번씩만 나온 단어일 경우
        freq_dict[entity]=0 // 0 입력
    else:
        freq_dict[entity] = (freq3/(freq1*freq2))
```

각 instance와 relation의 PMI를 측정 후 본래의 Triple과 비교하여 관계가 같은 단어들의 가중치를 합산 한 후 1을 더하여 log를 취한다. [그림 6-3]에서는 각 Term과 Md(x) value를 측정한 후 합산된 Triple의 순위화를 나타내고 있다.

instance	Relation	instance	PMI	TFreq	MD	total_value
DNA-binding pro	bind	single-stranded DN	8.392317423	0.00597015	0.012865497	8.411153069
sequence of base	defin	messenger RNA sec	8.392317423	0.00447761	0.012865497	8.409660532
double-stranded	provid for	DNA replication	8.392317423	0.00597015	0.010526316	8.408813888
messenger RNA s	defin	one or more protei	8.392317423	0.00447761	0.010526316	8.407321351
single-stranded D	curl in	long circle	8.392317423	0.00597015	0.008187135	8.406474707
DNA double helix	is stabil by	hydrogen bonds	8.392317423	0.00597015	0.008187135	8.406474707
Nucleases hydroly	are call	exonucleases	8.392317423	0.00298507	0.010526316	8.405828813
genetic recombin	can be involv in	DNA repair	8.392317423	0.00298507	0.010526316	8.405828813
guanine-rich sequ	may stabil	chromosome ends	8.392317423	0.00597015	0.007017544	8.405305116

[그림 6-3] Triple 순위화

최종적으로 순위화된 Triple들을 메타데이터로 구성하고 다음과 같이 시각화 하였다.



[그림 6-4] Triple 시각화

B. Precision Rate와 Recall Rate를 통한 평가

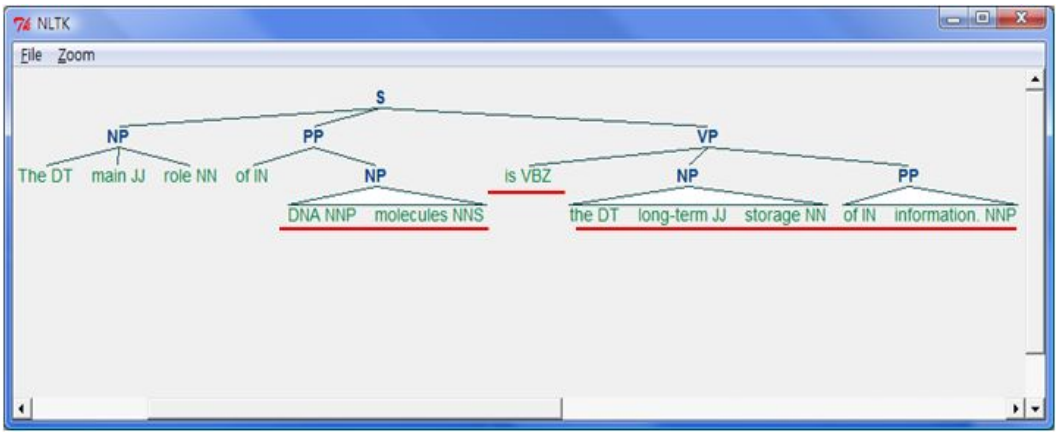
Precision Rate는 검색결과의 정확성을 평가하는 것이며, Recall Rate는 전체 내용중에 찾아야할 부분을 뜻하는 것으로 본 논문에서는 DNA문서에서 추출한 핵심문장들이 정보를 포함하고 있다고 가정하고 Precision Rate과 Recall Rate를 판단하였다. DNA문서에서 핵심문장으로 390개의 문장을 추출하였으며 문장들 중 Triple추출 패턴에 의해 Triple이 생성된 문장에 대해 Recall Rate를 판단하였고 추출된 Triple중 정확한 관계를 갖는 Triple을 통해 Precision Rate를 판단하였다.

390개의 문장을 Triple생성 패턴에 의해 추출한 결과 322개의 문장에서 Triple을 생성할 수 있었으며 Triple은 431개를 추출할 수 있었다.

추출된 Triple 중에서 관계가 잘못되거나 대명사 등을 포함하는 85개의 잘못된

Triple을 파악할 수 있었다.

본 연구의 실험을 비교하기 위해 동일 조건에서 Penn treebank Project의 품사태거로 태깅한뒤 context free grammar형식으로 문장을 구조화하고 품사와 구의 성분을 통해 실험하였다. [그림 6-5]에서는 분석된 문장의 Tree를 나타낸다.



[그림 6-5] 문장의 Tree구조

Triple의 상위어로는 동사 이전의 가장 하위의 명사구를 상위어로 선정하고 동사구를 이루는 동사를 관계어, 그리고 동사이하 명사구와 전치사구를 하위어로 선정하여 각 문장별로 Triple을 추출하였다. 각 구문의 분석 기준은 다음과 같다.

[표 6-4] Tree 생성 규칙

	S	문장
NP(명사구)	<DT|JJ|NN>	DT=관사,JJ=형용사,NN=명사
VP(동사구)	<VB><NP| PP| CLAUSE>	VB= 동사
PP(전치사구)	<IN><NP>	IN= 전치사
CLAUSE(절)	<NP><VP>	

구와 품사의 성분을 이용한 결과 총 390문장에서 316개의 문장에서 Tree를 생성할 수 있었으며 생성된 Tree의 수는 512개이며 이중 187개의 오류를 파악하였다.

위키피디아의 DNA page의 핵심문장을 이용하여 평가한 결과는 다음과 같다.

[표 6-5] 평가결과

	품사태깅을 이용한 경우	Link grammar를 이용한 경우
Precision Rate	0.6348	0.8027
Recall Rate	0.8103	0.8256

C. 실험 평가

비교평가한 결과 품사태거를 통한 구 성분에 따른 방법에 비해서 전체적으로 나은 결과를 얻을 수 있었다. 하지만 추출된 Triple을 비교해 보았을 때 본 연구의 방법이 다수의 문장들과 관계들이 특정 패턴에 의해 생성되지 않는 문제가 있었다. 이는 Triple 생성 패턴에 대한 고려가 더 필요하다고 본다. 또한 추출된 Triple의 평가 항목 중 가장 큰 오차 발생은 대명사와 접속사들로 인한 주어, 목적어 파악 문제와 긴 문장으로 인한 잘못된 Link설정이었다. 주어 인식은 it, that과 같은 대명사를 주어로 인식하여 의미파악이 불분명한 경우가 약 50여개 Triple에서 관측되었다. 긴 문장으로 인한 오류는 다양한 Triple을 추출하기 위해 설정한 패턴이 잘못된 Triple을 생성하는 경우로, Link grammar의 관계중 하나인 'MVp'는 동사와 전치사 구를 연결하는 Link를 생성하지만 긴 문장에서는 멀리 있는 전치사 구와의 연결로 인해 의미 파악이 잘못된 Triple이 발견되었다. 이외에 설정한 패턴 이외의 관계(B, R)가 중요정보를 획득한 경우, 정확한 Triple을 추출하지 못하였고 링크사전에 정의되지 않은 단어의 품사태깅 문제로 동사를 명사로 인식한 경우와 문장 Parsing으로 인한 error로 문서 내 소제목이 문장에 붙어 노이즈를 생성하는 경우들이 있었다. 이러한 점들을 극복하기 위해서는 향후연구로 개체명 인식이나 고유명사 판단에 대한 연구들이 선행되어야 할 것으로 판단된다.

Ⅶ. 결론 및 제언

최근 정보의 흐름에 따라 다양한 정보가 생성 또는 수정되고 있다. 따라서 의미적 정보처리의 핵심인 지식베이스 확장에 대한 중요성 역시 증가하고 있다. 본 논문에서는 온톨로지 확장을 위해서 정보가 풍부하고 변화가 빠른 위키피디아 문서를 대상으로 하여 Link grammar를 이용한 Triple 추출을 연구하였다.

전체과정을 위해 위키피디아 문서에서 본문영역만을 문장단위로 추출하고 핵심어 추출과정을 통해 위키피디아 문서내에서 중요도가 높다고 판단하는 핵심어휘를 추출하였다. 그리고 이를 바탕으로 핵심문장을 추출하였다. 추출된 핵심문장을 Link grammar로 구조화 할 때 분석의 정확성을 높이기 위해 긴 문장들에 대해서는 어휘매칭을 통해 문장을 분할 하였다. 그리고 Link grammar를 이용하여 핵심문장들을 구조화 하였고 Link grammar 관계분석을 통해 선정한 상하위어 및 관계추출 패턴을 바탕으로 구조화된 문장에서 이를 추출하였다. 추출된 상하위어와 관계를 기존의 품사태깅에 의한 상하위어 및 관계추출과 비교한 결과 더 나은 성능을 얻을 수 있었다. 추출된 상하위 관계들의 중요도를 파악하기 위해 가중치를 부여하여 순위를 판단하였다. 그리고 순위화된 상하위어 및 관계에 대하여 미리 정의한 온톨로지 형식에 맞춰 메타데이터를 생성하고 시각화 하였다. 이를 통해 전문적인 사전이나 시소러스를 통하지 않고 대중성을 지닌 위키피디아를 통해 온톨로지 확장이 가능함을 보였다.

본 논문에서 제안한 방법을 이용하여 위키피디아의 도메인 문서로부터 상하위어 및 관계를 추출할 수 있으나 이를 바로 적용하기에는 어려움이 있다. 대화체로 이루어진 위키피디아 문서는 많은 대명사와 고유명사들을 포함하고 있어 관계를 추출하지 못하는 문장들이 존재하였다. 따라서 정확성을 높이기 위해 Named Entity Recognition[25]에 대한 연구와 Anaphora Resolution[26]에 관한 연구가 필요하다. 그리고 Link grammar를 이용하여 풍부한 상하위어 및 관계 추출을 한 다양한 패턴들을 고려할 필요성이 있으며 다양한 분야의 문서에서의 정확도에 대한 평가가 요구된다. 이를 통해 향후 빠르게 변화하는 위키피디아의 장점을 활용함으로써 지식베이스 확장과 의미적 정보처리에 큰 효과를 가져올 것으로 기대하고 있다.

참고문헌

[1] M Hepp, K Siorpaes, D Bachlechner. "Harvesting WikiConsensus:Using Wikipedia Entries as Vocabulary for Knowledge Management", IEEE InternetComputing2007.

[2] A.Herbelotand A, Copestake. "Acquiring ontological relationships from Wikipedia using rmrs". Proceedings of Workshop on Web content Mining with Human Language Technologies, ISWC062006.

[3] F Wu, D Weld. "Automatically refining the Wikipedia infobox Ontology". portal.acm.org2008.

[4] http://wordnet.princeton.edu/.

[5] http://www.link.cs.cmu.edu/link/dict/index.html.

[6] http://labs.translated.net/terminology-extraction/.

[7] Gabrilovich and S. Markovitch. 2006. Overcomingthe brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge In Proceedings of AAI 2006.

[8] Daniel Sleator and Davy Temperley. 1991. Parsing English with a Link Grammar. Carnegie Mellon University Computer Science technical report CMU-CS-91-196, October 1991

[9] Soumen Chakrabarti저, 웹 마이닝(Ming the web)

[10] In Paul Buitelaar, P., Cimiano, P., Magnini B. (Eds.), Ontology Learning from Text: Methods, Applications and Evaluation (2005), pp. 3-12

[11] 황금하, 신지애, 최기선, "개념 및 관계 분류를 통한 분야 온톨로지 구축", 정보과학회논문지: 소프트웨어 및 응용 제 35 권 제 9호, 2008.9

[12] Feiyu Xu, Daniela Kurz, Jakub Piskorski, "Term Extraction and Mining of Term Relations from Unrestricted Texts in the Financial Domain, proceedings of BIS 2002, poznan, poland

[13] 류법모, 최기선, "텍스트에서 IS-A 관계의 자동 추출 및 순위화", 2007년도 제

[14] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, D. Mladenic. Triplet Extraction from Sentences. Ljubljana: 2007. Proceedings of the 10th International Multiconference "Information Society- IS 2007". Vol. A, pp. 218 - 222.

[15] K. Englmeier, F. Murtaghì, J. Mothe, Domain Ontology: Automatically Extracting and Structuring Community Language from Texts, IADIS Applied Computing, Spain, Espagne, 2007

[16] Joshi, R., Li, X.L., Ramachandaran, S., Leong, T.Y., " Automatic Model Structuring from Text using BioMedical Ontology", American Association for Artificial Intelligence (AAAI) Workshop on Adaptive Text Extraction and Mining., 2004

[17] MEDLINE : http://www.ncbi.nlm.nih.gov/entrez

[18] Peng, H.C., Long, F., and Ding, C., "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No.8, pp.1226-1238, 2005

[19] Erwan Moreau. 2004. Partial learning using link grammars data. In Proceedings of ICGI 2004

[20] Nguyen Bach & Sameer Badaskar, "A Survey on Relation Extraction"

[21] Erwan Moreau, LINA -FRE CNRS 2729, "From link grammars to categorial grammars"

[22] Zhihao Yang,Hongfei Lin and Baodong Wu, "BioPPIExtractor: A protein-protein interaction extraction system for biomedical literature", Expert Systems with Applications Volume 36, Issue 2, Part 1, March 2009, Pages 2228-2233

[23] Fabio Rinaldi, James Dowdall, Michael Hess, "The Role of Technical Terminology in Question Answering" Conference TIA-2003, Strasbourg,31 mars et 1 avril 2003

[24] http://www.nltk.org/

[25] Oliver Bender, Franz Josef Och and Hermann Ney, Maximum Entropy Models for Named Entity Recognition In: Proceedings of CoNLL-2003, Edmonton, Canada, 2003, pp.148-151

[26] http://plato.stanford.edu/entries/anaphora/

[27] 황명권, 윤병수, 정일용, 김판구 “Unknown Word Lexical Dictionary의 자동 생성 방법” 한국정보처리학회 춘계학술발표대회 논문집., 15권 1호(2008.5), pp.3-6

[28] Myunggwon Hwang, Chang Choi, Byungsu Youn, Pankoo Kim "Word Sense Disambiguation based on Relation Structure", International Conference on Advanced Language Processing and Web Information Technology., pp.15-20

[29] 황명권, 윤병수, 김판구, “지식베이스의 크기에 따른 WSD 알고리즘의 성능평가 방법“, 2008한국인터넷정보학회 추계학술발표대회., pp. 305-310

[30] 윤병수, 황명권, 김판구, “개념망 확장을 위한 명사구 추출방법”, 한국정보기술학회 하계종합학술대회 논문집.,

[31] 윤병수, 황명권, 김동철, 김판구, “도메인 용어의 시맨틱 네트워크 자동구축방안”, 한국정보과학회 2009 한국컴퓨터종합학술대회., pp.30-34

[32] 윤병수, 최준호, 김판구, “위키피디아 문서 분석을 통한 도메인 온톨로지 확장 방법” 2009 정보통신분야학회 합동학술대회., pp.611-615