무형대용어 해결 기술을 이용한 백과사전 표제어 복원

황민국[O*], 김영태[*], 나동열[*], 임수종[+]
연세대학교 컴퓨터정보통신공학부[*], 한국전자통신연구원 지식마이닝연구실[+]
{peacsid, wolfnamu, dyra2246}@gmail.com[*], isj@etri.re.kr[+]

Restoring Encyclopedia Title Words Using a Zero Anaphora Resolution Technique

Min-Kook Hwang[O*], Young-Tae Kim[*], Dongyul Ra[*], Soojong Lim[+]
Yonsei Univ., Computer & Telecommunications Eng. Div.[*], ETRI, Knowledge Mining Research Team[+]
{peacsid, wolfnamu, dyra2246}@gmail.com[*], isj@etri.re.kr[+]

요 약

한국어 문장의 경우 문맥상 추론이 가능하다면 용언의 격이 생략되는 현상 즉 무형대용어 (zero anaphora) 현상이 흔히 발생한다. 무형대용어를 채울 수 있는 선행어 (명사구)를 찾는 문제는 대용어 해결 (anaphora resolution) 문제와 같은 성격의 문제이다. 이러한 생략현상은 백과사전이나 위키피디아 등 백과사전류 문서에서도 자주 발생한다. 특히 선행어로 표제어가 가능한 경우 무형대용어 현상이 빈번히 발생한다. 백과사전류 문서는 질의응답 (QA) 시스템의 정답 추출 정보원으로 많이 이용되는데 생략된 표제어의 복원이 없다면 유용한 정보를 제공하기 어렵다. 본 논문에서는 생략된 표제어 복원을 위해 무형대용어의 해결을 기반으로 하는 시스템을 제안한다.

주제어: 무형대용어, 대용어, 대용어해결, 표제어, 표제어복원

1. 서론

대용어(anaphor) 현상이란 문장의 용언의 격을 채우는 성분이 이미 앞에서 나타난 경우 이를 다시 반복하지 않고 대명사를 사용하는 현상을 말한다. 다음 텍스트를 보자 [1].

"철수는 학교에 갔다. 가는 도중 그는 영희를 만났다."

위 예에서 "그" 는 "철수"의 반복을 피하기 위한 대용어이다. 이때 "철수"를 대용어의 선행어(antecedent)라고 부른다. 대용어와 선행어는 동일한 개체를 지시하므로 상호참조 (coreferent) 관계에 있다고 한다 [1]. 대용어의 선행어를 찾는 문제를 대용어 해결(anaphora resolution)이라고 한다. 선행어가 될 후보 명사구는 문서가 긴 경우 수십 개가 될 수도 있다. 선행어가 텍스트 내에 존재하지 않는 경우도 있는데(비조응성; nonanaphoricity), 이러한 경우도 밝혀내야 하므로 문제가 더욱 어렵게 된다.

무형대용어 현상은(zero anaphora) 대용어가 생략되는 것을 말한다 [2]. 다음 예를 보자.

"철수는 학교에 갔다. 가는 도중 ϕ 영희를 만났다."

대용어 "그는" 이 생략되어 빈 자리가 되었으며(ϕ 로 표시된 곳), 여기에 무형대용어가 발생하였다고 한다. 무형대용어의 경우 형태가 없으므로 대용어에 비해 그 선행어를 찾는 작업이 더 어렵다.

위키피디아 등 백과사전류 문서에서는 표제어와 상호참조 관계를 가지는 명사구인 경우 특히 생략될 경우가 많다 [3]. 다음 예를 보자.

표제어: 블라디미르 푸틴(두산백과)

> 푸틴은 상트페테르부르크에서 출생하였다. 1975년 상트페테르부르크대학교 법학부 국제법과를 ϕ_1 졸업한 뒤 연방보안국(FSB)의 전신인 구 소련 국가안보위원회(KGB)에 ϕ_2 들어가 주로 동독에서 오랜 기간 첩보활동에 ϕ_3 종사하였다.
> 이어 1990년부터 1996년까지 상트페테르부르크 대표자회의 의장의 보좌관과 상트페테르부르크시 해외위원회 위원장 등을 ϕ_4 역임하였다. …

위 그림에서 ϕ_i 은 표제어가 생략된 무형대용어를 나타낸다. 푸틴에 대한 질의를 받은 질의응답 시스템은 답을 찾는데 위 문서의 내용을 이용해야 할 수 있다. 그러나 표제어가 생략되어 생긴 무형대용어가 많다면 이들의 복원 없이는 답을 추출하는데 실패할 수 있다 (다음 예 참조).

Query: 동독에서 첩보활동에 종사했던 대통령은 누구인가?
Answer: 블라디미르 푸틴

이와 같은 필요성에 따라 본 연구는 백과사전류 문서의 표제어 복원을 위한 기술을 개발하려는 것이다.

그러나 백과사전 문서에서 오직 표제어만이 생략되는 것은 아니다. 다른 무형대용어들도 존재할 수 있다. 이러한 무형대용어들에 대한 복원도 마찬가지로 정보추출에 매우 필요하다. 이러한 이유로 우리는 표제어 복원 시스템의 개발에 있어 무형대용어의 해결에 기반을 두는

접근을 추구하였다.

백과사전 문서에 나타나는 무형대용어는 다음 두 가지 성질에 따라 분류될 수 있다: 문서내에 선행어가 있나(a) 없나(~a)의 여부, 표제어로 채울 수 있나(b) 없나(~b)의 여부.

1) 타입 1: a & b
2) 타입 2: a & ~b
3) 타입 3: ~a & b
4) 타입 4: ~a & ~b

백과사전류 문서에서 위의 4 가지 형태의 모든 무형대용어에 대하여 선행어 또는 표제어에 의해 복원되도록 하는 것을 목표로 하는 시스템의 개발이 우리의 목표이다.

2. 관련 연구

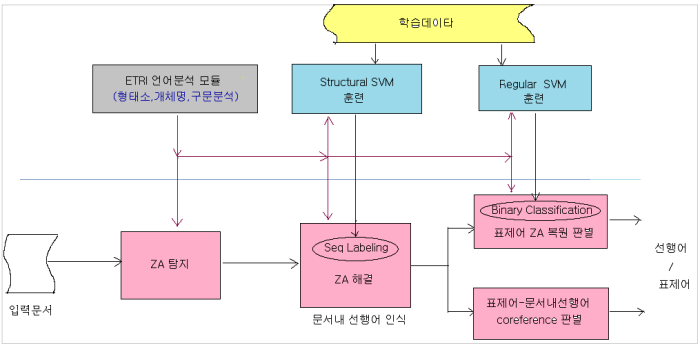
백과사전 문서에서 생략된 표제어를 복원하는 문제에 대한 연구로 [3]이 있다. 이 연구에서는 주어진 무형대용어에 대하여 문서내 선행어의 인식은 수행하지 않는다. 단지 이 무형대용어를 표제어로 복구할지, 복구하지 않을지 만을 결정하는 것을 목표로 한다. 결국 위의 타입 2의 경우에는 복원을 수행하지 않는다. 또한 주어진 무형대용어가 어느 타입인지의 구별도 수행하지 않는다.

이 연구의 특징은 규칙과 통계적인 방법을 하이브리드화 하여 사용하는 것이다. 예를 들면 특정 의미코드를 가지는 표제어의 문서의 경우 주어가 생략된 모든 무형대용어에 대하여 일괄적으로 표제어로 복원하는 것이다. 이러한 의미코드들을 선정하는 방법은 다음과 같다: 의미코드(s)에 속하는 대부분의 표제어들의 문서에서 주어가 생략된 무형대용어들 중 50% 이상이 표제어가 생략된 경우임이 관찰된다면 s를 선정대상으로 한다.

무형대용어 해결 문제는 일본에서 많이 연구되었다. 그 이유는 한국어와 유사한 종류의 언어인 일본어에서도 무형대용어 현상이 빈번히 발생하기 때문이다 [2,4,5]. 이 문제에 대한 최근의 대표적인 연구로는 Iida 의 연구를 들 수 있다 [6,7]. [6]의 연구에서는 선행어의 위치에 따라 문장 내 (intra-sentential), 문장 간 (inter-sentential) 로 나눈다. 전자는 선행어가 무형대용어와 동일한 문장에 나타나는 경우이며 후자는 앞의 다른 문장에 나타나는 경우이다. 결국 각 경우에 따라 따로 시스템을 구축하여 직렬로 시도하도록 한다. 또 다른 특징으로는 과거의 기법들에 비해 구문정보를 더욱 주요한 정보로 이용하는 것이다. 그들은 문장내 무형대용어만을 대상으로 할 때의 시스템 성능을 발표하였는데 F1=0.595를 가진다. 그러나 문장내 무형대용어는 전체 무형대용어의 일부이며 처리가 쉬운 경우라서 주어진 성능은 전체 무형대용어 시스템의 성능으로 간주하기 어렵다. 게다가 이들의 연구는 표제어의 복원 문제를 포함하지 않는다. 따라서 우리 시스템과의 직접적인 비교는 가능하지 않다. [7]의 연구에서는 Integer Linear Programming 이라는 보다 강력한 추론 기능을 이용하여 시스템의 성능을 향상하려는 시도가 소개되어 있다. 이 시스템의 성능은 전체적인 무형대용어 해결 작업에 대하여 F1 = 35 정도로 무형대용어 해결 작업이 상당히 어려운 문제임을 알 수 있다.

3. 시스템 구성

우리 시스템의 전체적인 구성은 그림 1과 같다. 표제어 복원 과정은 그림의 중간 선 아래 부분의 4 개의 모듈을 이용한다. 백과사전류 문서가 입력되면 먼저 문서내의 무형대용어(ZA)의 위치를 탐지한다. 그 다음 각각의 무형대용어에 대하여 "문서내 선행어 인식" 모듈을 거친다. 이 모듈에서는 문서내의 선행어 후보가 되는 명사구들 중에서 선행어가 있는지, 있다면 어느 것인지 인식하는 작업을 수행한다 (ZA 해결). 만약 문서 내에 선행어가 존재하지 않는다고 판단되면 "표제어 ZA 복원 판별" 모듈로 보낸다. 여기에서는 SVM 이진 분류기를 사용한다. 문서내에 무형대용어의 선행어가 존재하는 경우 "표제어 문서내 선행어 coreference 판별" 모듈을 거치도록 하는데 여기에서는 표제어와 이 선행어가 서로 상호참조 관계인지의 여부를 결정한다.



[그림 1] 전체 시스템 구성.

"문서내 선행어 인식" 모듈은 일반적인 무형대용어 해결 문제이다. 이의 기본 전략은 상호참조해결 문제에서 사용하는 기법에 기반하는데 지금까지 제안된 주요 방식은 "candidate-wise", "tournament" 두 가지가 제안되었다 [1,8,9,10,11,12,13,14]. 그러나 본 연구에서는 구조적 Support Vector Machine (SVM)을 이용하는 "시퀀스 레이블링 (sequence labeling)" 기법을 사용한다 [15].

결국 우리 시스템은 이진 분류를 위한 일반적인 SVM과 시퀀스 레이블링을 위한 구조적 SVM을 이용하는 방식으로서 기계학습 기법에 기반을 두고 있다.

4. 기계학습

4.1. 학습데이타

기계학습을 위하여 우리는 학습데이타를 구축하였다. 그림 2에서 각 무형대용어는 대괄호 쌍 "[...]" 안에 표시되어 있다. 예를 들어 [1,2/s] 는 주어(subj)가 생략

된 무형대용어이며 선행어는 1, 2 번으로 표시된 명사구 후보들로서 이 경우 문서내 선행어(2번 후보)와 표제어(1번 후보) 모두 선행어가 될 수 있음을 나타낸다.

표제어: <+1>현진건</1>

<+2>현진건</2>은 일제 강점기 조선의 소설가 겸 언론인이다. 「운수 좋은 날」, 「술 권하는 사회」 등 20편의 단편소설과 7편의 중·장편소설을 [1,2/s] 남겼다. 일제 지배하의 민족의 수난적 운명에 대한 객관적인 현실 묘사를 지향한 리얼리즘의 선구자로 [1,2/s] 꼽힌다.

[그림 2] 학습데이타

[표 1] 피쳐리스트.

(1) Pd 어절에서 맨 끝의 어미 앞의 형태소의 품사(동사/형용사 부분임)
(2) Pd 의 어미의 품사
(3) Pd 의 자동사/타동사/자타동사 여부: 0(자동사), 1(타동사), 2(자타동사)
(4) ZA 의 종류: s, o 중 하나일 것임
(5) NPi 와 ZA 가 동일 문장내에 있는지의 여부: 0(동일 문장), 1(다른 문장)
(6) NPi의 조사 이전의 품사(POS)
(7) NPi의 조사 이전의 형태소어휘(string or lexeme이라 부름)
(8) NPi의 조사의 품사(POS)
(9) NPi의 조사의 형태소어휘(string or lexeme이라 부름)
(10) NPi를 구성하는 명사의 NE type : NE 가 아닌 명사에 대해서는 notNE 라는 type 을 줄 것.
(11) NPi 의 지배소가 Pd 인지의 여부: 0(그렇지 않음), 1(그러함)
(12) Pd 가 NPi 의 조상인 경우 그 경로상의 의존관계 레이블 시퀀스
(13) NPi 와 Pd 사이에 존재하는 명사구의 수
(13') NPi 를 구성하는 형태소 중에 xsn 이 있는지 여부
(14) 이 NPi 가 표제어인지의 여부: 0(표제어가 아님), 1(표제어임)
(15) NPi 가 속한 문장과 Pd 가 속한 문장의 관계: 0(같은 문장), 1(바로 앞문장), 2(그 이외의 경우)
(16) NPi 가 문장에서 처음 나오는 명사구인가: 0 (아니다), 1 (그렇다).
(17) NPi 가 Pd 절 (Pd 가 이끄는 절)의 가장 좌측 형제 절의 가장 좌측 명사구 자식인가: 0(아님), 1(맞음).
(18) NPi 가 루트절의 맨 좌측 형제절의 맨 좌측 명사구인지의 여부: 0(아니다), 1(그렇다)
(19) NPi 노드와 Pd 노드를 연결하는 경로 상의 레이블 시퀀스

(주: Pd: 용언)

4.2. 피쳐

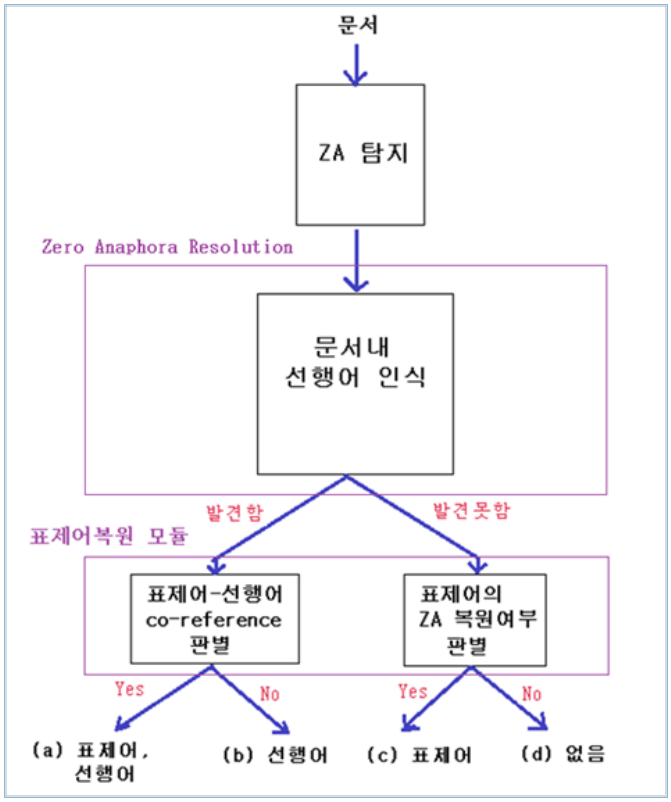
각 후보 명사구에 대하여 피쳐벡터가 준비된다. 후보 명사구는 무형대용어보다 이전에 나온 모든 명사구들이 대상이다. 물론 이 들 중에 일부만이 무형대용어의 선행어가 되므로 이들 모든 명사구들 중 일부만이 선행어 번호 태그를 가지게 된다.

우리는 여러 가지 다양한 정보를 피쳐로 준비한다: 어휘 정보, 위치정보, 구문구조 정보등을 포함한다. 우리가 사용하는 피쳐의 종류는 표 1과 같다.

5. 무형대용어 해결 기반 표제어 복원

우리 시스템은 백과사전류 문서내에 존재하는 모든 무형대용어를 처리 대상으로 한다. 각 무형대용어에 대하여 선행어가 있다면 이를 인식하고, 표제어도 선행어로 간주될 수 있다면 이를 인식한다. 결국 각 무형대용어는 1장에서 소개한 4 가지 타입 중 하나이며 해당 경우에 따라 선행어가 무엇인지, 표제어도 선행어로 볼 수 있는 여부를 결정한다.

우리 시스템의 작업 흐름은 그림 3과 같다. 입력은 백과사전류 문서이다. 문서는 표제어에 대한 설명을 담은 것으로서 표제어는 문서 텍스트의 일부가 아니라고 우리 연구에서는 가정한다. 그러나 문서마다 어느 표제어에 대한 문서인지는 주어진다고 가정한다. 따라서 표제어는 문서의 일부가 아니므로 선행어 후보 리스트에 들어 있지 않다. 주어진 문서에 대하여 시스템은 먼저 무형대용어 탐지를 수행한다. 무형대용어 탐지란 생략된 격을 찾는 작업이다. 우리의 경우 필수격 즉 주격과 목적격만을 탐지 대상으로 한다.



[그림 3] 표제어 복원 작업의 흐름도.

이를 위해 문장의 ETRI 구문분석 시스템이 제공하는 구문구조 분석 정보을 활용한다. 용언이 자동사나 형용사라면 필수격은 주격 하나이며, 타동사라면 주격 및 목적격 두 개가 필수격이다. 한국어의 경우 자동사 및 타동사 두 가지 경우로 이용될 수 있는 용언들이 많다 (이를 자타동사라고 부르자). 이들에 대해서는 현재로서는

주격만이 필수격이라고 본다. 결국 용언이 필요로 하는 필수격이 무엇인지를 파악하고 나서 구문분석 정보에서 필수격이 모두 존재하는 지를 파악하여 필수격의 생략여부를 판단한다.

탐지된 무형대용어에 대하여 문서내에 존재하는 선행어를 인식한다. 이를 위해 우리는 시퀀스 레이블링 기법을 사용한다. 이것은 후보 명사구 리스트 안의 각각의 명사구에 대하여 레이블 “Antec”, “NoAntec” 둘 중 하나의 레이블을 부여하는 작업이다. 이를 위해 우리는 Structural SVM을 사용한다 [15].

문서 내에서 선행어가 발견된 경우에는 (즉 Antec 레이블을 가진 명사구가 하나라도 존재하면) 이 선행어가 표제어와 상호참조 관계에 있는지 판단한다. 그렇다면 이 무형대용어는 타입 1 이며 그렇지 않다면 타입 2 이다. 타입 1의 경우에는 무형대용어의 위치에 선행어나 표제어로 복원해 줄 수 있다. 타입 2의 경우에는 선행어로만 복원할 수 있다. 이 상호참조 관계 판단은 현재로서는 선행어와 표제어의 스트링 매칭에 기반하고 있다.

문서내에서 선행어가 발견되지 않은 경우에는 이진 분류기를 이용하여 표제어가 탐지된 무형대용어의 선행어가 될 수 있는지를 결정한다. 이를 위해 SVM 기반의 이진분류 모델을 사용한다.

시퀀스 레이블링의 구조적 SVM, 표제어 복원 여부 판별을 위한 이진 분류 SVM 모두 4 장에서 소개한 피쳐셋을 이용한다.

6. 실험

우리 시스템의 성능은 Recall 과 Precision을 사용한다. 여기서 Recall이란 학습데이타에서 정답이라고 표시된 모든 무형대용어의 수에 대하여 시스템이 선행어를 제대로 찾아낸 무형대용어의 수의 비율이다. Precision 이란 시스템이 탐지한 무형대용어의 수에 대하여 시스템이 제대로 찾아낸 무형대용어의 수의 비율이다. 표 2는 전체 시스템의 성능을 보여 준다. 즉 모든 무형대용어에 대하여 타입 및 “선행어의 인식” 작업과 “표제어의 선행어 가능성 여부” 판단 작업을 정확히 알아내는 작업을 말한다.

[표 2] 전체 시스템 성능.

코퍼스	백과사전, 위키피디아
훈련데이타 크기	2,840 (ZA 발생 수)
테스트데이타 크기	1,093 (ZA 발생 수)
Recall	53.16
Precision	57.24
F1 score	55.12

무형대용어의 탐지 문제는 구문분석기의 성능에 큰 영향을 받는다. 이의 영향을 배제한 우리 시스템의 성능을 알아 보기 위해 탐지 모듈을 제외한 시스템의 성능을 표 3에 보여 주고 있다.

[표 3] 탐지 작업을 제외한 시스템 성능.

코퍼스	백과사전, 위키피디아
훈련데이타 크기	2,840 (ZA 발생수)
테스트데이타 크기	988 (ZA 발생수)
Recall	61.94
Precision	60.30
F1 score	61.11

7. 결론

본 논문에서는 백과사전이나 위키피디아 등 백과사전류 문서에서 나타나는 무형대용어의 복원 작업에 대한 시스템 개발을 다루었다. 이 목표를 위해 우리 시스템은 먼저 무형대용어 해결 작업을 수행한다. 그 다음 탐색된 선행어와 표제어의 상호참조 관계를 이용하여 표제어가 복원에 사용될 수 있는지를 판단한다. 우리 시스템의 핵심 모듈인 “무형대용어 해결 모듈”의 개발을 위해 우리는 시퀀스 레이블링 기법을 이용하였다. 실험 결과 우리 시스템의 성능은 F1 = 55 로서 가능한 선행어의 수가 크다는 점을 고려하면 양호한 것으로 생각된다. 하지만 실용적인 시스템이 되기 위해서는 상당한 성능향상이 필요하다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [10044577, (1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발]

참고문헌

[1] Soon, W. M., Ng, H. T., and Lim, D. C. Y. 2001. A machine learning approach to coreference resolution of noun phrases. Computa. Linguist. 27, 4, 521-544.

[2] Okumura, M. and Tamura, K. 1996. Zero pronoun resolution in Japanese discourse based on centering theory. In Proceedings of the 16th International Conference on Computational Linguistics (COLING). 871-876.

[3] 임수종, 이창기, 장명길, "백과사전 질의응답을 위한 생략된 표제어 복원에 관한 연구", 정보과학회 제32회 추계학술발표논문집, Vol.32, No. 2, 2005.

[4] Nariyama, S. 2002. Grammar for ellipsis resolution in Japanese. In Proceedings of the 9th Inter-national Conference on Theoretical and Methodological Issues in Machine Translation. 135-145.

[5] Seki, K., Fujii, A., and Ishikawa, T. 2002. A probabilistic method for analyzing Japanese anaphora integrating zero pronoun detection and resolution. In Proceedings of the 19th International Conference on Computational Linguistics (COLING). 911-917.
[6] Ryu Iida, Kentaro Inui, and Yuji Matsumoto, Zero-Anaphora Resolution by Learning Rich Syntactic Pattern Features, ACM Transactions on Asian Language Information Processing, Vol. 6, No. 4, Article 12, December 2007
[7] Iida, R. and Poesio, M., A Cross-lingual ILP Solution to Zero Anaphora Resolution, Proc. 49th Annual Meeting of the Association for Computational Linguistics, pp. 804-813, 2011.
[8] Iida, R., Inui, K., Takamura, H., and Matsumoto, Y. 2003. Incorporating contextual cues in trainable models for coreference resolution. In Proceedings of the 10th EACL Workshop on the Computational Treatment of Anaphora. 23-30.
[9] Ng, V. 2004. Learning noun phrase anaphoricity to improve coreference resolution: Issues in representation and optimization. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL). 152-159.
[10] Ng, V. and Cardie, C. 2002. Improving machine learning approaches to coreference resolution. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL). 104-111.
[11] Yang, X., Su, J., and Tan, C. L. 2005. Improving pronoun resolution using statistics-based semantic compatibility information. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL). 165-172.
[12] Yang, X., Su, J., and Tan, C. L. 2006. Kernel-based pronoun resolution with structured syntactic knowledge. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL). 41-48.
[13] Yang, X., Zhou, G., Su, J., and Tan, C. L. 2003. Coreference resolution using competition learning approach. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL). 176-183.
[14] Iida, R., Inui, K., and Matsumoto, Y. 2005. Anaphora resolution by antecedent identification followed by anaphoricity determination. ACM Trans. Asian Lang. Inf. Process. 4, 4, 417-434
[15] Soojong Lim, Changki Lee, and Donguul Ra, Dependency-based semantic role labeling using sequence labeling with a structural SVM, Pattern Recognition Letters, V. 34, April 2013.