ISSN 2288-4866 (Print)
ISSN 2288-4882 (Online)
http://www.jiisonline.org
J Intell Inform Syst 2015 June: 21(2): 131~150 http://dx.doi.org/10.13088/jiis.2015.21.2.131

Structural SVM을 이용한 백과사전 문서 내 생략 문장성분 복원*

황민국
연세대학교 컴퓨터정보통신공학부
(peacsid@gmail.com)

김영태
연세대학교 컴퓨터정보통신공학부
(wolfnamu@gmail.com)

나동열
연세대학교 컴퓨터정보통신공학부
(dyra@yonsei.ac.kr)

임수종
한국전자통신연구원
(isj@etri.re.kr)

김현기
한국전자통신연구원
(hkk@etri.re.kr)

영어와 달리 한국어나 일본어 문장의 경우 용언의 필수격을 채우는 명사구가 생략되는 무형대용어 현상이 빈번하다. 특히 백과사전이나 위키피디아의 문서에서 표제어로 채울 수 있는 격의 경우 그 격이 문장에서 더 쉽게 생략된다. 정보검색, 질의응답 시스템 등 주요 지능형 응용시스템들은 백과사전류의 문서에서 주요한 정보를 추출하여 수집하여야 한다. 그러나 이러한 명사구 생략 현상으로 인해 양질의 정보추출이 어렵다. 본 논문에서는 백과사전 종류 문서에서 생략된 명사구 즉 무형대용어를 복원하는 시스템의 개발을 다루었다. 우리 시스템이 다루는 문제는 자연어처리의 무형대용어 해결 문제와 거의 유사하나, 우리 문제의 경우 문서의 일부가 아닌 표제어도 복원에 이용할 수 있다는 점이 다르다. 무형대용어 복원을 위해서는 먼저 무형대용어의 탐지 즉 문서 내에서 명사구 생략이 일어난 곳을 찾는 작업을 수행한다. 그 다음 무형대용어의 선행어 탐색 즉 무형대용어의 복원에 사용될 명사구를 문서 내에서 찾는 작업을 수행한다. 문서 내에서 선행어를 발견하지 못하면 표제어를 이용한 복원을 시도해 본다. 우리 방법의 특징은 복원에 사용된 문장성분을 찾기 위해 Structural SVM을 사용하는 것이다. 문서 내에서 생략이 일어난 위치보다 앞에 나온 명사구들에 대해 Structural SVM에 의한 시퀀스 레이블링(sequence labeling) 작업을 시행하여 복원에 이용 가능한 명사구인 선행어를 찾아내어 이를 이용하여 복원 작업을 수행한다. 우리 시스템의 성능은 F1 = 68.58로 측정되었으며 이는 의미정보의 이용 없이 달성한 점을 감안하면 높은 수준으로 평가된다.

주제어 : 백과사전, 명사구 생략, 무형대용어 해결, Structural SVM, 시퀀스 레이블링

논문접수일 : 2015년 3월 6일 논문수정일 : 2015년 5월 24일 게재확정일 : 2015년 6월 2일
투고유형 : 국문급행 교신저자 : 나동열

1. 서론

대용어 현상이란 문맥(context)으로부터 유추가 가능한 문장 성분을 대용어(anaphor)로 대치하는 현상을 말한다. 대치된 원래의 문장 성분을 선행어(antecedent)라 부른다. <Figure 1>의 (a) 에서 대용어 “그는” 의 선행어는 앞 문장의 “철수는”이다. 대용어의 선행어를 찾아 내는 작업을 “대용어 해결(anaphora resolution)”이라 부른다. 영어와 달리 한국어나 일본어의 경우 대용어가

* 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (No.R0101-15-0062, 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발)

생략될 수도 있는데, 이를 무형대용어(zero anaphor; ZA)라 부른다(Nariyama, 2002; Iida et al., 2007). <Figure 1> 의 (b) 에서 ∅로 표시된 곳에서 대용어 “그는”이 생략되어 무형대용어가 발생하였다. “무형대용어 해결”이란 생략된 원래의 문장 성분을 알아 내어 복원하는 작업으로서 ∅의 경우 이에 대한 선행어인 “철수는”을 찾아야 한다.

(a) 철수는 학교에 갔다. 가는 도중 그는 영희를 만났다	(b) 철수는 학교에 갔다. 가는 도중 ∅ 영희를 만났다.

〈Figure 1〉 Anaphora and Zero-anaphora in Texts

그런데 백과사전이나 위키피디아 문서에서는 표제어로 채울 수 있는 격(case)의 경우 더 자주 생략된다(Lim et al., 2005). 예로서 표제어 “블라디미르 푸틴”에 대한 위키피디아 문서인 <Figure 2>를 보자. 여기에서 $\varnothing_i$들은 표제어가 생략되어 발생한 무형대용어(ZA)들을 나타낸다.

표제어: 블라디미르 푸틴
러시아 대통령인 푸틴은 상트페테르부르크에서 출생하였다. 1975 년 상트페테르부르크대학교 법학부 국제법과를 $\varnothing_1$ 졸업한 뒤 연방보안국(FSB)의 전신인 구 소련 국가안보위원회(KGB)에 $\varnothing_2$ 들어가 주로 동독에서 오랜 기간 첩보활동에 $\varnothing_3$ 종사하였다. .....

〈Figure 2〉 Noun Phrase Omission in a Wikipedia Document

질의응답(Question Answering) 시스템 등 주요 응용시스템의 경우 질문의 정답을 찾기 위해 백과사전 문서를 많이 이용한다. 이러한 시스템의 성능을 높이기 위해서는 생략된 명사구의 복원이 매우 중요하다. 따라서 본 논문에서는 이를 위한 기술을 개발하고자 한다.

무형대용어(ZA) 해결 즉 복원을 위해서는 무형대용어와 상호참조(co-reference) 관계에 있는 명사구 즉 선행어를 찾아야 한다. 두 명사구가 세상의 동일한 객체를 지칭하는 표현이라면 이 둘은 상호참조 관계를 가진다고 한다(Soon et al., 2001). ZA의 선행어는 ZA가 발생한 문서 내에 1번 이상 나타나는 경우가 대부분이지만 (anaphoric ZA), 나타나지 않는 경우도 간혹 있을 수 있다(nonanaphoric ZA). 이중 어느 경우이든 백과사전 문서에서는 표제어로도 ZA를 복원하는 것이 가능한 경우가 많다.

선행어의 탐색 범위는 문서 내에서 무형대용어보다 앞에 나타난 명사구들로서 이들이 선행어 후보 리스트이다. 무형대용어 해결 문제는 일본에서 많이 연구되어 왔으며(Seki et al., 2002; Iida et al., 2007; Iida and Poesio, 2011), 한국어에서는 최근의 Kim et al.(2014)에 의한 연구가 있다. 이들 기존 연구는 우리 연구와 달리 일반 문서를 대상으로 한 것이어서 표제어에 의한 복원은 이들 연구에서는 다루지 않았다. 기존 연구에서는 각 후보 명사구에 대하여 선행어가 될 수 있을지 여부를 결정하는 이진 분류 기계학습 모델에 기반을 두고 있다는 점이 우리 연구와 다르다. Lim et al.(2005)의 연구는 백과사전에서 생략된 표제어의 복원 문제를 다루었다는 점에서 우리 연구와 목표가 같다. 그러나 이 연구는 명사 의미분류 및 격틀을 이용하는 규칙기반 모델을 큰 비중으로 사용하므로 확장성이 부족한 문제점을 가지고 있다. 우리의 연구는 기존 연구에서 사용한 모델보다 더 진보된 기계학습 모델을 이용한다는 점에서 새로운 시도이다.

우리는 문서 내의 선행어 탐색을 위해 시퀀스 레이블링(SL; sequence labeling) 메커니즘을 이용할 것을 제안한다. 기존 연구에서는 후보 명사구 리스트의 각 명사구마다 독립적으로 선행어 여부에 대한 이진 분류 결과에 기반한다(Iida et al., 2005). 이를 후보 별(candidate-wise) 기법이라 칭하자. 이와 달리 우리는 전체 후보 명사구들에 대하여 동시에 선행어 여부를 결정하는 전역적(global) 기법을 사용한다. SL 작업은 입력으로 여러 개체로 구성된 리스트를 받으며 출력으로 레이블 리스트를 생성한다. SL은 입력과 출력을 각각 하나의 개체가 아니라 여러 개체로 구성된 구조체로 취급하는 특징을 가진다. 단일 개체에 대한 이진분류에 적합한 일반 SVM은 SL 작업에 이용될 수 없다. 따라서 구조체를 입력과 출력으로 수용할 수 있는 Structural SVM을 우리 시스템의 모델로 채택한다(Nguyen and Guo, 2007).

본 논문에서 제안하는 접근 방법이 효과적임을 보이기 위해 시스템을 개발하고 실험을 통하여 성능을 확인하였다. Ng그 결과 우리 시스템의 성능은 F1 = 68.58로 측정되었다. 이는 기존의 연구 결과보다 향상된 성능이다.

본 논문의 구성은 다음과 같다. 2 장에서는 본 연구와 관련이 있는 기존의 연구들에 대해 살펴본다. 우리가 제안하는 기술에 대하여 3 장에서 설명하고 4 장에서는 이를 구현하여 실험한 결과를 보여 주고 시스템 성능에 대하여 평가한다. 마지막으로 5 장에서 결론을 언급한다.

2. 관련 연구

한국어 백과사전 문서에서 생략된 표제어의 복원에 대한 연구로 (Lim et al., 2005)이 있다. 이 연구에서는 규칙기반과 통계기반의 방법을 복합적(hybrid)으로 사용하는 2단계 기법을 제안하였다. 먼저 규칙기반 방법의 단계 1에서는 표제어 의미코드와 용언의 격틀 정보를 이용한다. 표제어 의미코드가 특정 부류의 의미클래스에 속하면 표제어에 의한 무형대용어의 복원이 가능하다고 판단한다. 또 다른 경우로, 무형대용어를 표제어로 복원한 상태에서의 문장의 구문구조가 해당 용언의 격(case) 틀(frame)에 어느 정도 이상으로 매칭되면 표제어에 의한 복원이 가능하다고 본다. 복원 가능으로 판단된 경우, 단계 2에서 Maximun Entropy(ME) 확률 모델에 의하여 복원의 가능성에 대한 확률을 구한다. 이 확률이 특정 임계치를 넘으면 복원을 수행한다. 이 연구의 실험 결과, F1 = 68.07의 최종성능을 얻었다. 이 연구의 문제점은 표제어만을 복원의 후보로 고려하는 점과 규칙기반 기법이 확장성이 부족하다는 데 있다.

백과사전 생략 성분 복원 문제는 무형대용어 해결 문제와 거의 같다. 무형대용어 해결 과정은 첫 단계에서 무형대용어의 발생을 탐지하고, 다음 단계에서 선행어를 탐색한다. 둘째 단계는 대용어 해결에서의 선행어 탐색 작업과 동일한 작업이다. 대용어의 선행어를 문서 내에서 탐색하는 문제와 관련하여 지금까지 많은 연구가 있었다(Soon, et al., 2001; Ng and Cardie, 2002; Ng, 2004; Iida et al., 2003; Iida et al., 2005; Yang et al., 2003). 이들 연구에서 사용하는 공통적인 방법론은 선행어 후보 명사구 리스트 내의 각 후보마다 대용어와 상호참조 관계 여부를 판단하는 분류 작업 후에 가장 가능성이 높은 명사구를 선택하는 것이다. 결국 이들 연구에서 제안된 방법은 넓은 관점에서 보면 이진분류 기계학습 모델을 이용하는 방법론이다. 그러나 이는 후보 별

(candidate-wise) 기법에 속하며, 문맥정보의 활용이 부족할 수 있다는 문제가 있다.

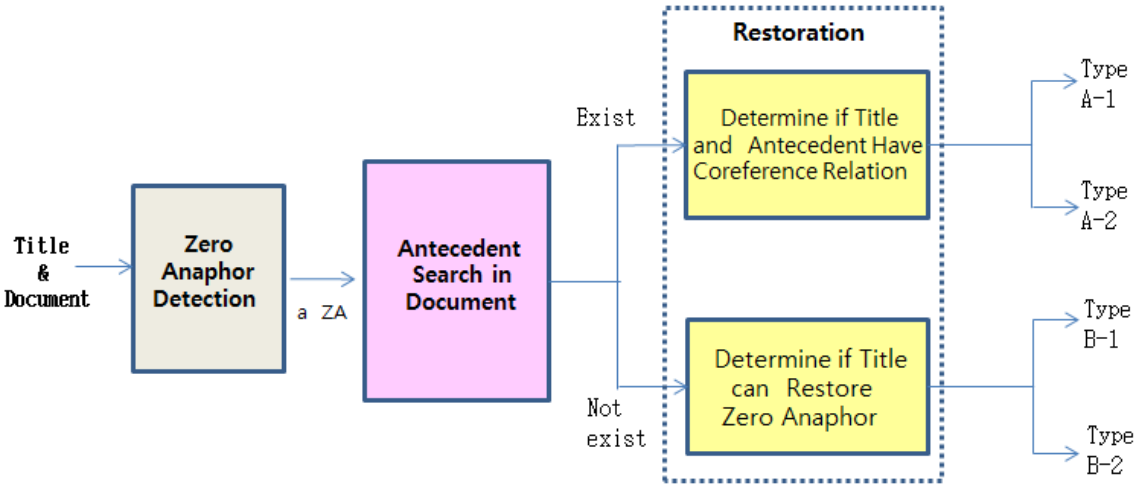
시퀀스 레이블링(SL)은 일반적으로 후보 별 기법보다 문맥정보를 더 잘 활용할 수 있는 전역적(global) 기법으로 알려져 있다. 품사 태깅, 의미역 결정과 같은 문제에서 SL을 적용한 결과 매우 효과적임이 밝혀졌다(Nguyen and Guo, 2007). 우리는 본 논문에서 무형대용어의 선행어 탐색에 SL을 적용하는 새로운 시도를 하였다. SL작업에서는 먼저 하나의 개체(object)에 부여할 수 있는 가능한 모든 레이블(label)의 집합 $L = \{l_1, \cdots, l_h\}$이 주어진다. SL에의 입력은 여러 개체를 가진 리스트로서 $\mathbf{x} = (x_1, \cdots, x_n)$로 나타낼 수 있다. SL 작업이란 $\mathbf{x}$내의 모든 개체에 대하여 동시에 L의 적절한 레이블을 부여하여 출력 레이블 리스트 $\mathbf{y} = (y_1, \cdots, y_n)$을 결정하는 작업이다. 여기서 y_i는 개체 x_i에 부여된 레이블이다. SL의 핵심적인 특징은 개체 별로 독립적으로 레이블을 결정하는 것이 아니라 다른 개체들에 부여되는 레이블을 참고하면서 전체 개체들의 레이블을 동시에 결정하는 것이다. 즉 입력 $\mathbf{x}$ 및 출력 $\mathbf{y}$를 각각 구조체(structure)로 간주하여 작업하는 것이다.

SL의 구현에 이용할 수 있는 기계학습 모델로 HMM, CRF, Structural SVM 등이 있다(Nguyen and Guo, 2007). 우리는 이 중에서 가장 최근에 소개된 모델인 Structural SVM을 이용하였다. Structural SVM은 일반 SVM을 확장하여 일반화한 모델로서 리스트나 트리 등 복잡한 구조의 출력까지도 가능하도록 한 것이다(Tsochantaridis et al., 2004, 2005). Structural SVM 모델은 차별화 함수 $\mathbf{F}(\mathbf{x}, \mathbf{y}; \mathbf{w})$를 이용하는데 여기서 벡터 $\mathbf{w}$는 시스템 파라메타이다. 이 함수의 값은 입출력 쌍 $(\mathbf{x}, \mathbf{y})$에서 얻는 자질(feature) 벡터와 $\mathbf{w}$의 내적(inner product)으로 구한다(3.3절 참조). Structural SVM 의 훈련이란 최적의 파라메터 벡터 $\mathbf{w}$를 구하는 작업이다. 그런데 그 계산 양이 매우 크다는 문제점이 있다. Structural SVM의 실용성을 위해서는 이 계산 양을 줄이는 기술이 필요하다(Sarawagi and Gupta, 2008; Joachims et al., 2009). Shalev-Shwartz et al.(2007, 2011)는 일반 SVM의 훈련에 SGD(stochastic gradient descent) 기법을 적용하여 속도 향상을 이룩할 수 있는 기법인 Pegasos 알고리즘을 소개하였다. Lee and Jang (2010)은 Pegasos알고리즘을 수정 및 확장하여 Structural SVM 의 훈련 속도를 향상시키도록 할 수 있는 Modified Pegasos 알고리즘을 제안하였다. 우리의 Structural SVM 훈련에는 Modified Pegasos 알고리즘이 이용되었다.

3. Structural SVM을 이용한 생략 복원

3.1. 전체 시스템 구성

우리 시스템의 구성은 <Figure 3>과 같다. 본 논문에서는 복원 대상인 생략된 명사구를 간단히 ZA(zero anaphor)로 표시한다. 복원 시스템으로의 입력은 하나의 표제어와 이에 대응하는 하나의 문서이다. 입력에 대해 시스템은 첫 단계에서 문서의 모든 문장들에서 필수 격이 생략된 곳들을 모두 찾아낸다(ZA Detection 모듈). 이를 위해 우리는 ETRI구문분석 시스템에 의한 구문분석 결과를 이용한다(Lim et al., 2014). 이 구문분석 시스템은 천이(transition) 시스템에 지배소 후위 원리를 적용하는 기법을 사용하였다. 어절 간의 의존관계 레이블까지도 출력해 주며, 레이블



〈Figure 3〉 Configuration of the System

부착 정확도(LAS)가 90.82%로서 현재까지 알려진 한국어 구문분석 시스템 중에서는 가장 높은 성능을 가진다. ZA 발생 탐지를 위해 각 문장에 대하여 구문분석을 수행하고 문장 내의 용언마다 이의 필수 격(주격, 목적격)을 채우는 문장성분이 생략되어 있는지 조사한다. 결국 ZA 탐지 성능은 ETRI 구문분석 시스템의 성능에 좌우된다.

탐지된 각 ZA 는 명사구가 생략되었음을 나타내며 이의 복원에 이용될 수 있는 명사구 즉 선행어를 문서 내에서 찾는 ZA 탐색작업을 수행한다(Antecedent Search 모듈). 그 다음 단계인 복원(Restoration) 모듈에서는 ZA 탐색 결과에 따라 다음 두 가지의 경우로 처리한다. (i) 선행어가 찾아졌다면 이것이 표제어와 상호참조 관계인지 판단한다. 상호참조 관계라면 답은 타입 A-1 이고, 그렇지 않다면 타입 A-2 이다. (ii) 선행어를 찾지 못한 경우에는 표제어로 복원할 수 있는지 판단한다. 표제어로 복원할 수 있으면 타입 B-1, 그렇지 않으면 타입 B-2로 결정한다. 타입 A-1 또는 B-1 이면 표제어로 복원해 주며, 타입 A-2 이면 찾은 선행어로 복원한다. 타입 B-2의 경우에는 복원하지 않는다.

3.2. 시퀀스 레이블링 기반 선행어 인식

다른 연구에서와 다르게 본 논문에서 제안하는 새로운 기법은 무형대용어를 복원할 수 있는 선행어를 문서 내에서 찾기 위하여 시퀀스 레이블링 기법을 적용하는 것이다. 이에 대하여 <Figure 4>의 문서를 예로 들어 보자. 이 문서에서 ∅로 표시된 위치에 “당선되었다”의 주격이 생략되어 있어 ZA가 탐지된다. 문서 내에서 ZA 보다 앞에 나타난 10 개의 명사구들이 선행어 후보가 된다: $(NP_1, \cdots, NP_n)$. 기존 연구들의 선행어 탐색 방법은 후보 리스트 내의 각 명사구마다 독립적으로 선행어로 가능한 지에 대하여 판단하는 방식에 기반을 두고 있다. 즉 NP_1 에 대하여 선행어 여부를 판단하고, NP_2에 대하여 선행어 여부를 판단하고, ..., NP_{10}에 대하여 선행어

표제어: 아시프 알리 자르다리

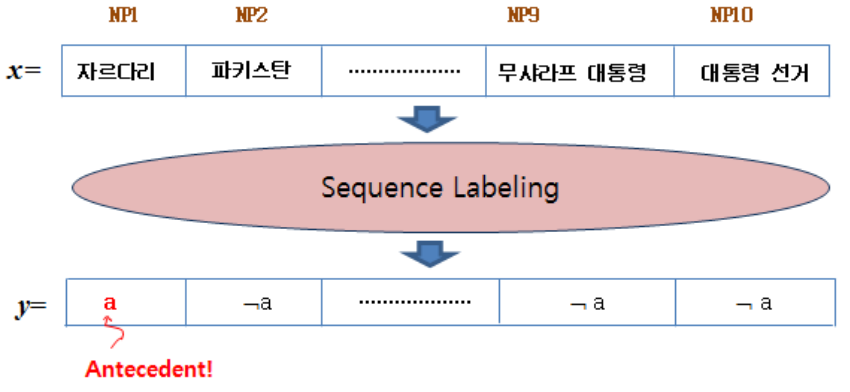
자르다리는 파키스탄의현직대통령이다. 총리를지낸부토의 남편이며, 부토가 2007년 테러로사망한이후유력한정치인으로떠올랐다. 무샤라프 대통령이사임한후, 2008년 9월 치러진 대통령 선거에서 ∅ **당선되었다.**

〈Figure 4〉 Antecedent Candidates in Text

여부를 판단하는 과정을 거친다. 여기서 각 후보에 대한 판단은 a(선행어), $\neg a$(비선행어)의 두 레이블 중 하나를 부여하는 이진분류로 볼 수 있다. 모든 후보의 레이블을 결정한 후, 레이블 a로 분류된 후보가 오직 하나 있으면 이를 선행어로 결정한다. 만약 이러한 후보가 두 개 이상이면 이 중에서 분류 확실성(confidence)이 가장 높은 후보로 결정한다. 이러한 방식을 후보 별(candidate-wise) 기법이라 부르자(Ng and Cardie, 2002; Iida et al., 2005).

그러나 우리는 후보 리스트 $(NP_1, \cdots, NP_n)$ 전체를 하나의 시퀀스 즉 한 구조체로 보고 이에 대하여 SL을 수행하여 전체 후보들에 대한 분류 결과를 나타내는 출력 $\mathbf{y} = (y_1, \cdots, y_n)$을 생성한다 ($= a$ or $\neg a$, $1 \leq i \leq n$). SL 기법의 특징은 입력 시퀀스 내의 각 원소의 레이블을 결정하는데 있어서 자신의 주변의 원소들이 가지는 레이블도 참조하여 결정하므로 문맥 정보의 이용 정도가 높다는 점이다. SL 수행 기법들은 특히 통계적인 추론(statistical inference) 등을 사용하여 전역적으로 가장 좋은 (globally best) 출력 시퀀스를 생성하게 할 수 있어 전역적(global) 기법이라 불린다. 이의 대표적인 예로는 품사태깅, 음성인식 등에 적용된 HMM(hidden Markov model)의 활용을 들 수 있다(Jelinek, 1998). 과거 연구에 따르면 후보 별 기법에 비해 SL 에 의한 전역적 기법이 향상된 성능을 얻을 수 있는 것으로 알려져 있다(Lafferty et al., 2001).

<Figure 4>에 주어진 문제의 경우 SL에의 입력은 이다. 이에 대한 **SL** 의 수행 예시가 <Figure 5>에 주어져 있다. 수행 결과 출력된 레이블 시퀀스에 의하면 "자르디리"는 레이블 a가



〈Figure 5〉 Antecedent Search Using Sequence Labeling

부여되었으므로 선행어로 가능하다고 판단된 것이다. 반면에 "대통령 선거"는 레이블 $\neg a$이 부여되어 선행어가 될 수 없다고 판단된 것이다. SL의 출력 **y** 내에 a 가 두 개 이상일 수도 있다. 이 경우에는 이 중 확실성이 가장 높은 하나만을 선택한다. 출력 **y** 내에 a 가 없다면 결국 문서 내에 선행어가 존재하지 않는 것으로 판단된 것이다.

3.3. Structural SVM 의 원리

Structural SVM은 지도학습에 기반한 기계학습 모델로서 리스트, 트리 등 구조적인 개체의 출력이 가능하도록 SVM 을 확장한 한 모델이다 (Tsochantaridis et al., 2004, 2005). Structural SVM은 다음과 같은 차별화 함수 **F**를 이용한다:

$$\mathbf{F}: \mathbf{X} \times \mathbf{Y} \rightarrow \mathcal{R} \tag{1}$$

이 수식에서 $\mathcal{R}$은 실수 집합, **X**는 모든 가능한 입력 개체의 집합, **Y**는 출력으로 가능한 모든 개체의 집합을 나타낸다. 이 함수는 주어진 입력 **x**에 대하여 이에 대한 정답 출력에 가까운 **y**일 수록 함수 값이 더 크도록 설계된다. 함수 **F**는 식 (2)로 구현된다:

$$\mathbf{F}(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \boldsymbol{w}^{\boldsymbol{T}} \Psi(\mathbf{x}, \mathbf{y}) \tag{2}$$

이 식에서 **w**는 시스템을 대변하는 파라메타 벡터이다($\boldsymbol{w}^{\boldsymbol{T}}$는 벡터 **w**의 transpose를 나타냄). 는 입출력 쌍 을 대변하는 자질(feature) 벡터이다. 즉 **F**는 **w**와 $\Psi(\mathbf{x}, \mathbf{y})$의 내적(inner product)으로 계산된다. 시스템에게 입력 **x**가 주어질 때 이에 대한 출력을 구하는 디코딩 원리는 식 (3)과 같다.

$$\hat{\boldsymbol{y}} = \boldsymbol{ARGMAX}_{\boldsymbol{y}}\ \mathbf{F}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \tag{3}$$

모델 구축에서 먼저 해결되어야 할 문제는 최적의 시스템 파라메타 **w**를 구하는 문제로서 이를 훈련(training)이라 부른다. 이 문제는 지도학습 기반의 기계학습(machine learning) 방법론을 사용하여 접근한다. 지도학습 방식으로 모델을 훈련시키기 위해서는 학습데이터가 필요한데 이것은 식 (4)과 같이 훈련예제(training example)들의 집합이다.

$$\mathrm{S} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_M, \mathbf{y}_M)\} \tag{4}$$

각 훈련예제 (x_i, y_i)는 y_i가 입력 x_i에 대한 출력 정답임을 말한다.

Structural SVM 의 훈련을 위해 우리는 Lee and Jang(2010)이 소개한 Modified Pegasos 알고리즘을 이용한다. Shalev-Shwartz et al.(2007, 2011)이 제안한 Pegasos 알고리즘은 일반 SVM의 빠른 훈련을 위해 subgradient 기법을 이용하는 방법이다. 이 기법을 Structural SVM에도 적용이 가능하도록 확장한 것이 Modified Pegasos 알고리즘이다. Modified Pegasos 알고리즘의 핵심은 다음과 같이 정의한 목적함수(objective function) f를 이용하는 것이다. 이 함수 f 는 **w**에 따른 시스템의 손실(loss)을 나타내며 결국 f를 최소화하는 가 최적의 시스템 파라메타가 된다. 일반적으로 SVM 훈련은 제한 조건하의 최적화 문제를 해결하는 것이다. 그러나 식 (5) 와 같이 f를 정의하면 제한 조건이 없는 최적화

(unconstrained optimization) 문제로 변경된다.

$$f(\mathbf{w}; A_t) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{k}\sum_{(\mathbf{x}_i,\mathbf{y}_i)\in A_t} l(\mathbf{w}; (\mathbf{x}_i, \mathbf{y}_i)) \quad (5)$$

이 수식에서 λ는 정규화 상수, A_t는 S로 부터 무작위로 k 개의 원소를 추출하여 만든 부분집합이다. 훈련예제 (x_i, y_i) 하나에 의한 손실(loss)을 구하는 함수 l은 다음 식 (6) 과 같이 정의된다.

$$l\big(\mathbf{w}; (\mathbf{x}_i, \mathbf{y}_i)\big) = \max\{0, \max_{\mathbf{y}}\{L(\mathbf{y}_i, \mathbf{y}) - \mathbf{w}^T \delta\Psi_i(\mathbf{x}_i, \mathbf{y}_i)\}\} \quad (6)$$

이 수식에서 $L(\mathbf{y}_i, \mathbf{y})$는 해밍 거리(Hamming distance)라 불리는데 $\mathbf{y}_i$와 $\mathbf{y}$을 원소 위치별로 원소 값을 비교하여 값이 다른 위치의 갯수를 말한다. 그리고 $\delta\Psi_i\big(\mathbf{w}; (\mathbf{x}_i, \mathbf{y}_i)\big) = (\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \mathbf{y})$로 정의한다. 결국 위 식에서 $w^T\delta\Psi_i(\ldots) = \mathrm{F}(\mathbf{x}_i, \mathbf{y}_i) - \mathrm{F}(\mathbf{x}_i, \mathbf{y})$이다. $\mathbf{y}_i$가 정답 출력이고 $\mathbf{y}$는 하나의 가능한 출력이므로 $\mathrm{F}(\mathbf{x}_i, \mathbf{y}_i) \geq \mathrm{F}(\mathbf{x}_i, \mathbf{y})$이 성립한다. 더 나아가 두 함수 값의 차이 $\mathrm{F}(\mathbf{x}_i, \mathbf{y}_i) - \mathrm{F}(\mathbf{x}_i, \mathbf{y})$는 $\mathrm{L}(\mathbf{y}_i, \mathbf{y})$ 만큼은 되어야 시스템 파라메타 $\mathbf{w}$가 최적의 값을 가진 것으로 볼 수 있다. 따라서 $\mathrm{F}(\mathbf{x}_i, \mathbf{y}_i) - \mathrm{F}(\mathbf{x}_i, \mathbf{y})$의 값이 $\mathrm{L}(\mathbf{y}_i, \mathbf{y})$ 값보다 작을 수록 손실이 많이 생긴 것이다. 결국 전자를 후자에서 뺀 값을 $\mathbf{y}$가 출력일 경우의 손실로 한다. 식 (6) 에서 $\max_{\mathbf{y}}(\ldots)$는 모든 가능한 $\mathbf{y}$ 중 가장 큰 손실을 주는 $\mathbf{y}$ 에 의한 손실을 구하는 것이다. 만약 이 손실이 음수이면 이득(gain)으로 해석해야 하는데 이 훈련에서 이득은 고려하지 않는 것으로 하므로, 손실이 음수이면 손실을 0으로 놓는다. 훈련예제 에 의한 손실 값 은 모든 가능한 출력 중에서 가장 큰 손실을 생성하는 에 의한 손실로 놓는다(아래 식 (7)에서는 이 $\mathbf{y}$를 $\mathbf{y}_i^*$로 표시함).

목적함수 f는 A_t에 속하는 훈련예제들의 손실의 총합으로 구성되며 가능하면 $\mathbf{W}$의 길이가 작게 되도록 $\|\boldsymbol{w}\|^2$도 포함시킨다. 손실이란 작을 수록 좋으므로 결국 f를 최소화하는 $\mathbf{w}$가 최적의 $\mathbf{w}$이다. 최적의 $\mathbf{w}$를 구하는 기본적인 방법은 $\mathbf{w}$를 $\mathbf{w} + (-\eta)\nabla\mathrm{f}(\mathbf{w}; A_t)$로 갱신하는 작업을 반복하는 subgradient 기법이다(학습율 η는 양의 상수). ∇f는 f의 gradient로서 식 (7)과 같다.

$$\nabla\mathrm{f}(w, A_t) = \lambda\mathbf{w} - \frac{1}{|A_t|}\sum_{(\mathbf{x}_i,\mathbf{y}_i)\in A_t^+} \delta\Psi_{i,j}(\mathbf{x}_i, \mathbf{y}_i^*) \quad (7)$$

$\boldsymbol{A_t^+}$는 A_t 의 원소 중 l 값이 양인 원소들의 집합이다.

3.4. 복원 결정 과정

입력 $\mathbf{x} = (NP_1, NP_2, \cdots, NP_n)$에 대하여 Structural SVM에 의한 SL 작업의 출력으로 레이블 시퀀스 $\mathbf{y} = (y_1, \cdots, y_n)$을 얻었다 하자. $\mathbf{y}$의 원소들 중에 오직 1 개만이 레이블 a 를 가지면 이 원소에 대응하는 NP 를 선행어로 결정한다. 그러나 만약 2 개 이상이 레이블a 를 가진다면 이 경우에는 Structural SVM이 각 원소마다 붙여 놓은 확실성 정도가 가장 높은 레이블 a 를 가진 원소를 선택한다. 이렇게 선택된 원소에 대응하는 NP를 선행어로 결정한다. 만약 모든 원

소에 레이블 $\neg a$ 가 부여되었다면 문서 내에 선행어가 존재하지 않는 것으로 판단한다.

문서 내에서 선행어가 탐지되었다면 이것으로 복원해도 된다. 그러나 표제어로도 복원이 가능한지 알기 위해 표제어와 선행어가 상호참조 관계인지 조사한다. 이것은 표제어 스트링(string)과 찾은 선행어의 스트링이 서로 유사한지 스트링 매칭(matching)을 통해 결정한다. 두 스트링이 서로 유사하다고 판단되면 결국 타입 A-1의 경우로 간주하여 표제어로 복원한다. 서로 유사하지 않다고 판단되면 타입 A-2의 경우로 간주하며 문서 내 선행어로 복원한다. 현재는 다음과 같은 매우 간단한 매칭 기준을 이용한다 (유사하다고 결정되면 변수 D에 1을 아니라면 0 을 넣는다):

- 표제어와 선행어 명사구를 분해하여 단위명사 리스트 ***tl***, ***sl*** 를 얻는다(주: 조사는 제외함),
- ***tl***, ***sl*** 의 맨 우측 명사를 각각의 헤드 명사로 보아 ***tn***, ***sn*** 에 넣는다,
- 표제어 헤드 ***tn*** 이 고유명사인 경우에는 ***tl***, ***sl*** 이 하나의 단위명사라도 공유한다면 D = 1 로 결정하고 하나도 공유하지 않는다면 D = 0.
- 반면에 표제어 헤드 ***tn*** 이 일반명사인 경우에는 두 헤드 ***tn*** 과 ***sn*** 이 같으면 D = 1 로 아니라면 D = 0.

· 예 1: 표제어 = “엘비스 프레슬리”, 선행어 = “엘비스”인 경우.

표제어 헤드 “프레슬리”가 고유명사이고 ***tl*** = {엘비스, 프레슬리}, ***sl*** = {엘비스} 이어서 명사 “엘비스”를 공유하므로 D = 1. (만약, 선행어=”존 덴버” 라면 D = 0.)

· 예 2: 표제어 = “포츠담 협정”, 선행어 = ”종전 협정”인 경우.

tn = ”협정”이 일반명사이다. 두 헤드 ***tn***, ***sn*** 이 모두 “협정”으로 같으므로 D = 1.

만약, 선행어후보=“포츠담”인 경우라면 ***sn*** = “포츠담”이어서 ***tn*** 과 다르므로 D = 0.

Structural SVM 의 SL 수행으로 문서 내에 선행어를 찾지 못한 경우에는 표제어가 복원에 사용될 수 있는지를 판단하여야 한다. 이것은 이진 분류 문제로서 일반 SVM 을 사용한다. 그 결과에 따라 표제어로 복원하거나 (타입 B-1), 복원하지 못할 수도 있다(타입 B-2).

4. 실험 및 평가

4.1. 실험데이터

앞의 3.3 절에서 설명하였듯이 기계학습 기반으로 시스템을 개발하기 위해서는 훈련 및 테스트를 위한 학습데이터 **S**가 필요하다. 우리는 실험을 위해 백과사전류인 한국어 위키피디아로부터 표제어 및 문서들을 수집하였다. 지도학습(supervised learning) 모델인 경우 학습데이터의 생성을 위해서는 정답이 붙여진 문서집단인 “정답 태깅 코퍼스”의 구축이 필요하다. 정답 태깅(tagging) 작업은 수작업으로 진행된다. 따라서 많은 노력과 시간이 소요되므로 대량의 데이터 구축은 어렵다. <Figure 6>은 정답 태깅된 문서의 모습이다.

우리는 정답을 표시해 주는 태그(tag)로 2 가지를 사용한다: 무형대용어(ZA) 태그, 선행어 태그. 문서 내에 발생한 ZA는 ZA태그로 표시되는데, [*n*/*case*] 형태로 구성된다. ZA 태그를 넣는 위치는 관련된 용언의 바로 앞이다. 이 태그 안

표제어: 넬슨 만델라

<+1>넬슨 롤리랄라 만델라<-1>는 남아프리카 공화국에서 평등 선거 실시 후 뽑힌 세계 최초의 흑인 대통령이다. 대통령으로 [1/s] **당선되기** 전에 <+1>그<-1>는, 아프리카 민족회의(ANC)의 지도자로서 반아파르트헤이트운동 즉, 남아공 옛 백인정권의 인종차별에 맞선 투쟁을 [1/s] 지도했다.
반역죄로 [1/s] 체포되어 종신형을 선고 [1/s] 받았으나, 26년 만인 1990년 2월 11일에 [1/s] 출소했다. 1994년 4월 27일 실시된 선거에서 ANC는 62%를 득표하여 ANC의 지도자인 <+1>넬슨 만델라<-1>는 1994년 5월 27일 남아프리카 공화국 최초의 흑인 대통령으로 취임하였고 <+2>진실과 화해위원회<-2>(TRC)를 [1/s] **결성하여** 용서와 화해를 [1/ms] 강조하는 과거사 청산을 [1/s] 실시했다.
<+2>TRC<-2>는 성공회 주교인 데스몬드 투투 주교가 참여하였으며, 수많은 과거사 관련 <+3>자료<-3>들을 [2/s] 수집하여 [2/s][3/o] 조사하였다.

〈Figure 6〉 An Answer-tagged Document

의 *n*은 번호로서 이 ZA에 대한 선행어가 어느 명사구인지를 가리키는 번호이다(선행어가 없으면 0). 슬래쉬(/) 다음의 *case* 는 어느 격(case)이 생략되어 ZA가 발생하였는지를 나타낸다(s: 주격, o: 목적격). 예를 들면 <Figure 6>에서 "**당선되기**" 바로 앞에 있는 ZA태그 [1/s]를 보자. 번호 1은 이 ZA의 선행어가 선행어 태그 번호가 1 인 명사구 즉 "넬슨 롤리랄라 만델라" (또는 표제어)임을 나타낸다. s는 주격이 생략되어 발생한 ZA임을 나타낸다.

문서 내에 발생한 ZA에 대한 선행어가 되는 명사구들을 선행어 태그(tag)로 표시해 준다. 이 태그는 명사구를 양측에서 <+*n*>, <-*n*> 로 둘러싸고 있다. 여기서 *n* 는 문서 내의 여러 선행어들을 구별하기 위한 선행어 번호이다. 같은 선행어 번호로 태그된 명사구들은 서로 상호참조 관계이어서 동일한 객체를 지칭한다. 그리고 특별히 표제어와 상호참조 관계인 선행어들은 선행어 번호를 무조건 1로 부여하기로 하였다.

<Figure 6>에서 "결성하여" 바로 앞의 ZA 태그 [1/s] 를 보자. 이것의 선행어 번호가 1이므로 이 ZA 태그 앞에 나타난 명사구들 중에 번호가 1 로 붙여진 3 개의 명사구들이 선행어가 될 수 있다는 의미이다. 그리고 선행어 번호 1은 특별히 표제어도 포함한다. 결국 다음 4 가지 중 어느 것이든지 이 ZA의 복원에 이용될 수 있음을 나타낸다(단, 대명사는 복원에서 제외):

<+1>넬슨 롤리랄라 만델라<-1>,
<+1>그<-1>,
<+1>넬슨 만델라<-1>

<Table 1> Experimental Data

Source of documents	Korean Wikipedia				
# Documents	508				
# Sentences	3,088				
# Zero anaphors	Total	Case		Anaphoricity	
	2,895	Subject	Object	Nonanaphoric	Others
		2,814	81	560	2,335

표제어(넬슨 만델라)

훈련 및 테스트를 위해 정답 태깅된 실험데이터로부터 학습데이터를 준비하여 이용한다. 우리 실험데이터의 크기 및 특성은 <Table 1>과 같다. 실험데이터에 나타난 무형대용어의 총 수는 2,895개인데, 이 중 2,814개가 주격의 생략에 의해 발생한 것이다. 이는 한국어 백과사전에서 목적격보다는 주격이 더 많이 생략됨을 나타낸다. <Table 1>에서 2,895개의 전체 무형대용어(ZA) 중 19.3%인 560개는 표제어나 문서 내의 선행어로 복원할 수 없는 경우이다(nonanaphoric ZA).

4.2. 학습데이터 생성

• Structural SVM의 학습데이터 생성

시스템 모델의 훈련과 테스트 모두 학습데이터가 필요하다. 두가지 학습데이터의 생성을 위해 전체 실험데이터를 훈련용 및 테스트용의 두 부분으로 나눈다. 그리고 각각으로부터 학습 데이터 $\{(\mathbf{x_1}, \mathbf{y_1}), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\}$을 생성한다. 학습데이터는 앞에서도 언급하였듯이 훈련예제의 집합이다. 정답 태깅 코퍼스의 각 문서 안의 각 ZA 마다 하나의 훈련예제 $(\mathbf{x}_i, \mathbf{y}_i)$가 생성된다. 문서 내에서 ZA 보다 앞에 나타난 명사구가 n 개라면 입력벡터 $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n})$가 생성된다. 여기서 $x_{i,j}$는 j 번째 명사구를 나타낸다. 이 입력에 대한 출력 레이블 벡터 $\mathbf{y}_i = (y_{i,1}, \ldots, y_{i,n})$의 생성은 다음과 같다:

$$y_{i,j} = \begin{cases} a, & if\ Num(tag\ of\ ZA) = Num(tag\ of\ x_{i,j}) \\ \neg a, & Otherwise \end{cases} \quad (8)$$

$for\ each\ j, 1 \le j \le n.$ 여기서 $Num(T)$은 태그 T 안의 선행어 번호를 나타낸다. 만약 명사구 $x_{i,j}$가 선행어 태그가 붙어 있지 않은 것이라면 $y_{i,j} = \neg a$로 한다.

• 일반 SVM의 학습데이터 생성

선행어를 찾지 못한 경우에 표제어로 ZA를 복원할 수 있는지 판단해야 한다. 우리는 이 문제를 이진 분류로 접근한다. 이를 위해 일반 SVM을 사용하며 이의 훈련을 위한 각 훈련예제는 정답 태깅 코퍼스 내의 문서 내의 각 ZA마다 하나씩 생성된다. 이 경우 훈련예제 하나를 $(\mathbf{x}_i, y_i)$로 표시할 수 있는데, 여기서 입력 $\mathbf{x}_i$는 표제어, ZA, ZA 가 나타난 문장 등 주변 문맥 정보를 대변한다고 볼 수 있다. y_i는 이 입력에 대한 정답 출력 레이블을 나타내는데 Y 또는 N 중의 하나이다. Y는 표제어로 ZA의 복원이 가능함을, N은 그렇지 않음을 나타낸다. 정답 태깅에서 이 ZA 태그의 선행어 번호가 1 이면 표제어로도 채울 수 있음을 나타낸다. 따라서 출력 레이블 y_i의 값은 다음과 같이 결정한다:

$$y_i = \begin{cases} Y, & if\ Num(tag\ of\ ZA) = 1 \\ N, & Otherwise \end{cases} \quad (9)$$

4.3. 자질 벡터 생성

SVM의 훈련 및 테스트의 수행을 위해서는 훈련예제로부터 생성된 자질(feature) 벡터가 필요하다. 자질 벡터는 함수 에 의해 생성된다. 현재 한국어의 경우 대규모 의미 정보원의 구축이 미

흡한 실정이다. 이로 인해 우리 시스템에서는 의미정보와 관련된 자질은 이용하지 못하고 있다. 이러한 문제점을 최소화 하기 위하여 <Table 2>에서 볼 수 있는 것처럼 우리는 구문구조 관련 자질을 강화하였다.

• Structural SVM 의 자질 벡터 생성:

훈련예제 (**x, y**) 에 대한 자질 벡터 $\Psi(\mathbf{x}, \mathbf{y})$는 자질 값들의 리스트이다. $\mathbf{x} = (x_1, \ldots, x_n)$의 각 원소 x_i는 명사구(NP)를 나타낸다. 훈련예제 (**x, y**)은 특정 ZA에 대하여 생성된 것으로 이 ZA가 나타난 문서 내의 주변 문맥 상황도 자질의 생성에 이용된다. 예를 들면, ZA가 일어난 용언 *Vb*(verb)도 자질 생성에 이용된다. 각 명사구 마다 <Table 2>에 열거된 모든 자질들에 대하여 자질 값을 생성한다. 이들은 기본 자질이라 불리는데 그대로 이용되지 않고 출력 레이블과 결합한 확장 자질이 되어 이용된다. 예를 들면, 의 헤드의 품사(POS; part of speech)가 nn 이라면 기본 자질 값 "**nn**" 이 생성된다. 그리고 $y_i = a$ 하자. 그러면 이들을 서로 결합하여 확장 자질 값 "**nn.a**" 를 만든다. 결국 x_i 의 모든 기본 자질들의 값들로 확장 자질 값 리스트를 만든다. **x** 내의 모든 x_i 들의 확장 자질 값 리스트들을 모아서 최종적인 자질 벡터 $\Psi(\mathbf{x}, \boldsymbol{y})$를 구성한다. $\Psi(\mathbf{x}, \boldsymbol{y})$는 확장 자질들 이외에 트랜지션 (transition) 자질들도 포함한다. 트랜지션 자질은 두 인접한 레이블의 결합으로 만들어 진다:

$$y_i . y_{i+1}, for\ each\ i, 1 \le i \le n-1. \quad (10)$$

• 일반 SVM 의 자질 벡터 생성:

일반 SVM은 복원에 표제어를 이용할 수 있는지를 결정하는 이진분류에 이용한다. 이 경우의 훈련예제는 $(\mathbf{x}, \boldsymbol{y})$로서 **x**는 표제어 및 ZA 가 발생한 주변 문맥정보를 대표한다. 결국 **x**에 포함되는 명사구는 표제어 하나이다. 이 표제어를 NP로 하여 <Table 2>의 자질들의 자질 값을 생성하고 이들로 자질 벡터를 만든다. 표제어는 ZA가 속한 문서를 구성하는 한 문장으로 간주하

〈Table 2〉 Basic Features

종류	자질(feature)
Lexical	NP: POS of head; Lexeme of head; POS of josa; Lexeme of josa; Named entity type; Vb: POS of stem; Lexeme of stem; POS of eomi; Lexeme of eomi; Is it transitive or intransitive?; The kind of the omitted case;
Position	Distance between NP and ZA: Are they in same sentence?; Number of NPs between them; Number of eojeols between them;
Syntactic structure	· Is NP a child of Vb? · Dependency label sequence from NP to Vb · Is NP the 1st noun phrase of the sentence containing it? · Is NP the 1st noun phrase of the leftmost clause which is sibling of the clause headed by Vb? · Is NP the leftmost noun phrase of the leftmost sibling clause of the main clause of the sentence containing ZA? · Is NP the subject of the first clause of the sentence containing it? · Is NP the subject of any clause?

지 않는다. 결국 <Table 2>에서 위치 및 구문구조에 속하는 자질들의 값을 구할 수 없다. 이 경우 자질 값은 널(null)을 가지는 것으로 가정한다.

표제어로부터 추출되는 자질 중 가장 시스템 성능에 가장 큰 영향을 미치는 자질은 의미적인 정보를 대략이나마 제공해 줄 수 있는 개체명(named entity) 타입 자질이다. 예를 들면 인물을 개체명 타입으로 가지는 표제어는 복원 성분으로 이용될 수 있는 경우가 많다. 표제어 하나로부터 개체명을 유추하는 것은 정확도가 나쁘다. 따라서 우리는 문서 안에 표제어를 포함하는 문장이 있는 경우 이 문장에서 표제어의 개체명을 인식하는 휴리스틱을 이용한다.

일반 SVM 의 경우는 기본 자질들을 이용해 자질 벡터 $\Psi(\mathbf{x}, \boldsymbol{y})$를 구성한다. 여기서는 확장 자질과 트랜지션 자질은 이용되지 않는다.

4.4. 성능 측정 및 평가

시스템의 성능평가를 위해 정보검색 및 자연어처리에서 많이 이용하는 성능평가 척도인 R(재현율), P(정확율), F1 score를 이용한다.

$$\mathrm{R} = \frac{\text{테스트데이터 내의 ZA 중 시스템이 올바로 맞춘 것들의 갯수}}{\text{시스템이 탐지한 모든 ZA의 갯수}} \tag{11}$$

$$\mathrm{P} = \frac{\text{테스트데이터 내의 ZA 중 시스템이 올바로 맞춘 것들의 갯수}}{\text{테스트데이터 내의 총 ZA 갯수}} \tag{12}$$

F1 score 는 위 두 척도를 종합하여 하나의 척도로 만든 것으로 P와 R의 조화평균(harmonic mean)으로 계산된다.

$$\mathrm{F1} = \frac{2PR}{P+R} \tag{13}$$

시스템의 성능을 테스트하기 위해서도 정답 태깅 정보가 필요하다. 그 이유는 시스템의 출력이 올바른지 판단하기 위해서는 입력에 대한 정답이 무엇인지를 알아야 하기 때문이다.

테스트용 정답 태깅 코퍼스 안에 태그로 표시된 각 ZA 마다 다음과 같은 과정을 수행하여 R을 계산한다. 먼저 이 ZA 를 시스템이 탐지하였는지 알아 본다(격 종류도 맞아야 함). 만약 탐지하지 못했으면 R의 분모를 1만큼 증가시키지만 분자는 증가시키지 않고 다음 ZA 로 간다. 탐지하였다면, 이 ZA를 나타내는 훈련예제 $(\mathbf{x}_i, \mathbf{y}_i)$를 준비한다(즉 i 번째ZA임). 문서에서 ZA 발생 위치보다 앞에 나온 명사구들에 의해 입력 $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n})$가 구성된다. 정답 태깅 정보를 이용하여 정답출력 $\mathbf{y}_i = (y_{i,1}, \ldots, y_{i,n})$를 준비한다. 이 ZA 에 대하여 우리 시스템의 처리 결과를 구한다. 앞 3.1 절에서 설명한대로 시스템의 출력은 4 가지 중 하나이며 각 경우에 대하여 다음과 같이 시스템의 출력 결과를 평가한다. (여기서 $Num(\mathrm{ZA})$은 이 ZA의 정답태그에 표시된 선행어 태그 번호이다.)

(a) 타입 A-1, 찾은 선행어 = $x_{i,j}$:
If $y_{i,j} = a$ & $Num(\mathrm{ZA}) = 1$ then pr ← 1 else pr ← 0;

(b) 타입 A-2, 찾은 선행어 = $x_{i,j}$:
If $y_{i,j} = a$ & $Num(\mathrm{ZA}) \neq 1$ then pr ← 1 else pr ← 0;

(c) 타입 B-1:
If $Num(\mathrm{ZA}) = 1$ & $\mathbf{y}_i$ has no a then pr ← 1 else pr ← 0;

(d) 타입 B-2:

〈Table 3〉 Measured Performance

	Our System						Lim et al.(2005)
	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Avg.	
P	66.12	64.53	67.73	70.37	67.46	67.24	60.25
R	72.73	68.95	68.87	71.70	67.82	70.01	78.23
F1	69.27	66.67	68.30	71.03	67.64	68.58	68.07

If $Num(\mathrm{ZA}) = 0$ & $\mathbf{y}_i$ has no a then pr ← 1 else pr ← 0;

위의 처리 결과 pr = 1이라면 이 ZA를 시스템이 잘 처리한 것이므로 R의 분모와 분자 각각을 1씩 증가시키지만, 그렇지 않다면 R의 분모만 1 증가시킨다.

P를 구하기 위해서 테스트용 정답 태깅 코퍼스에서 모든 정답태그를 제거한 문서들을 준비하여 시스템에게 제공한다. 시스템으로 하여금 이 문서들에 대하여 모든 ZA의 탐지 및 복원 작업을 수행하게 한다. 그 다음 시스템이 탐지한 ZA 마다 다음과 같이 처리한다. 이 ZA가 정답 태깅에서는 표시되지 않은 것이라면 P의 분모만 1만큼 증가시키고, 탐지된 다음 ZA 로 간다. 정답 태깅에서 표시된 것이라면 앞의 R의 계산에서 설명한 과정대로 수행한다. 그 결과 pr = 1 이라면 P의 분모와 분자를 모두 1 씩 증가하고 그렇지 않으면 분모만 1 증가시킨다. 그리고 탐지된 다음 ZA의 처리로 간다.

우리 시스템의 일반적인 성능을 측정하기 위하여 5-fold 교차검증(cross validation)을 실행하였다. 이를 위해 전체 실험데이터를 동일한 크기의 5개의 부분데이터로 나눈다. 실험 1에서는 첫번 째 부분데이터를 테스트용으로 나머지 4개를 훈련용으로 사용한다. 실험2 에서는 두번 째 부분데이터를 테스트용으로 나머지 4 개를 훈련용으로 사용한다. 이러한 방식으로 5 번의 실험을 실시하였다. <Table 3>은 우리 시스템의 성능 측정 결과를 보여 주며, 비교를 위해 마지막 열에 Lim et al.(2005)의 성능도 나타내었다.

실험 결과 우리 시스템의 F1 score 의 평균은 68.58로서 이는 (Lim et al., 2005)의 68.07에 비해 근소하게 높은 것으로 나타났다. 교차검증에서 F1 score 의 표준편차는 1.49로서 실험간 변동폭은 크지 않은 편이다. 이 비교에서 다음과 같은 점을 주목하여야 한다. 먼저, 우리 시스템의 경우 의미적인 정보는 전혀 이용하지 못했다. 반면 Lim et al.(2005) 시스템의 경우 명사들을 의미코드 별로 분류한 정보, 용언의 격틀 정보 등 의미적인 정보를 많이 이용하였다. 만약 우리 시스템에 이러한 의미적인 정보를 추가한다면 큰 폭의 성능 향상을 이룰 수 있을 것이라 예상된다. 둘째, 표제어와 상호참조 관계가 없는 문서 내 선행어에 의하여 복원되어야 하는 ZA 문제의 경우는 Lim et al.(2005) 시스템으로는 처리할 수 없다.

<Table 4>는 여러 관점에서 우리 시스템의 성능을 살펴 본 것이다. "Only ZA detection" 실험은 ZA 발생의 탐지 기능에 대한 성능을 측정한 것이다. ZA 발생 위치와 생략된 격(case)의 종류까지는 맞춰야 하지만 선행어 탐색은 문제에 포

함하지 않는 것이다. 이 결과를 보면 한국어 의존구조 구문분석의 현재 성능 수준인 90% 보다는 약간 못하다. 그 이유는 우리 시스템의 경우 임의의 문서에 대하여 구문분석을 실행한 것과 같으므로 구문분석 평가를 위한 실험데이터에서 측정한 구문분석 성능 측정치보다는 낮을 수 밖에 없으며, 품사태깅 단계의 성능이 100%가 아니므로 이 단계에서 약간의 성능저하를 일으키게 된다.

"Only ZA resolution" 실험에서는 ZA 가 잘 탐지되었다는 가정하에 (즉 위치 및 생략된 격의 종류는 알려진 상황에서) 시스템이 복원에 이용할 성분을 얼마나 잘 찾아내는지를 알아 본 것이다. 이 경우에는 P 와 R 이 같은 값을 가지는 데 그 이유는 찾아야 할 ZA 수(R 의 분모)와 시스템이 찾은 ZA의 수(P의 분모)가 동일하기 때문이다. "Only for people" 은 실험 대상을 (의인화가 가능한 "기관"을 포함한) 인물 관련 표제어의 문서들로만 국한 시킨 경우이다. 이에 속하지 않는 부류에 대한 실험은 "Only for non-people"로 주어져 있다. 이 실험 결과를 보면 인물의 경우보다 비인물의 경우가 더 어려운 문제임을 알 수 있다.

〈Table 4〉 Performance for Various Aspects

	Only ZA detection	Only ZA resolution	Only for people	Only for non-people
P	84.56	76.41	73.52	57.75
R	92.82	76.41	78.53	65.15
F1	88.49	76.41	75.94	61.23

우리 시스템의 ZA 발생 탐지 모듈은 문장에 대한 구문분석 결과를 이용한다. 시스템 출력에 대한 오류 분석 결과에 따르면, 우리 시스템의 성능은 구문분석의 정확성에 크게 의존함을 알 수 있었다. 구문분석의 오류는 ZA 탐지 오류로 이어지며 이는 시스템 성능의 하락으로 이어진다. 이 문제는 시스템 구조의 설계적인 면에서 우리 시스템이 가지는 한계점이다. 시스템 오류 분석으로 명확해진 또다른 한계점은 의미 정보를 이용하지 못하여 발생하는 성능하락 현상이다. 즉 명사의 의미분류, 용언의 격틀정보 및 하위범주화 정보 등 의미정보는 시스템 성능향상에 매우 중요하나 현재는 한국어의 경우 이러한 의미정보를 구하기 어려운 실정이다.

5. 결론

본 논문에서는 백과사전이나 위키피디아 등 백과사전류 문서에서 나타나는 명사구 생략의 복원을 수행하는 시스템의 개발을 다루었다. 우리 시스템의 특징은 시퀀스 레이블링 메커니즘을 사용하여 복원에 이용될 명사구를 탐색한다는 점이다. 이 작업의 실제 구현을 위한 장치로 우리는 Structural SVM 을 사용한다. 이는 대용어 또는 무형대용어 해결과 관련해서는 우리가 처음으로 시도한 방법이다. 문서 내에서 선행어를 찾은 경우에는 표제어와의 상호참조 여부 판단을 통하여 표제어로도 복원이 가능한지 결정한다. 문서 내에 선행어가 없는 경우에는 표제어로 생략 부분을 복원할 수 있는지의 여부를 이진분류 SVM를 이용하여 결정한다. 이와 같은 방법을 통하여 우리는 F1 = 68.58 인 성능을 얻게 되었는데, 이는 과거 다른 연구보다 향상된 결과이다. 특히 우리 시스템의 경우 의미정보를 전혀 이용하지 않고 얻은 성능임을 감안하면 높은 수

준으로 평가된다.

시스템의 성능 향상을 위하여 다음과 같은 향후 연구가 필요하다. 먼저, 현재 시스템의 설계 구조로 볼 때 무형대용어의 발생 탐지를 위하여 사용하는 구문분석기의 결과에 대한 우리 시스템의 종속성이 너무 크다. 이의 개선을 위해 적절한 휴리스틱을 사용하는 후처리 단계의 도입, ZA 탐지를 위한 기계학습 모델의 도입 등 여러 방향으로의 연구가 필요하다. 둘째, 의미정보를 이용하지 못하는 현 상황의 개선이다. 의미정보를 수작업으로 구축하는 것은 현실적으로 어렵기 때문에 이를 극복할 수 있는 대안을 찾는 것이 필요하다. 셋째, 선행어의 문서 내 존재 여부(anaphoricity)를 결정하는데 있어 현재보다 더욱 지능적인 접근이 필요한 것으로 생각된다.

참고문헌(References)

Iida, R., K. Inui, and Y. Matsumoto, "Anaphora resolution by antecedent identification followed by anaphoricity determination," *ACM Transactions on Asian Language Information Processing*, Vol. 4, No. 4(2005), 417~434.

Iida, R., K. Inui, H. Takamura, and Y. Matsumoto, "Incorporating contextual cues in trainable models for coreference resolution," *Proceedings of the 10th EACL Workshop on the Computational Treatment of Anaphora*, (2003), 23~30.

Iida, R., I. Kentaro, and Y. Matsumoto, "Zero-Anaphora Resolution by Learning Rich Syntactic Pattern Features," *ACM Transactions on Asian Language Information Processing*, Vol. 6, No. 4, Article 12(2007), 1~22.

Iida, R. and M. Poesio, "A Cross-lingual ILP Solution to Zero Anaphora Resolution," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, (2011), 804~813.

Jelinek, F., *Statistical Methods for Speech Recognition*, MIT Press, 1998.

Joachims, T., T. Finley, and C.-N. Yu, "Cutting-plane training of structural SVMs," *Machine Learning*, Vol. 77 No. 1 (2009), 27~59.

Kim, K.-S., S.-B. Park, and S.-J. Lee, "Identification and Application of Non-anaphoric Zero Pronouns in Intra-sentential Contexts based on the Cohesion between Clauses," *Journal of Korean Institute of Information Scientists and Engineers: Software and Applications*, Vol. 41, No. 3(2014), 233~240. (In Korean)

Lafferty, J., A. McCallum, F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *Proceedings of 18th International Conference on MachineLearning*, (2001), 282~289.

Lee, C. and M.-G. Jang, "A Modified Fixed-threshold SMO for 1-Slack Structural SVM," *ETRI Journal,* Vol.32, No.1, (2010), 120~128.

Lim J., Y. Yun, Y. Bae, S. Lim, H. Kim, and K. Lee, "Korean Dependency Parsing Model based on Transition System using Head Final Constraint," *Proceedings of 26th Annual Conference on Humanand Cognitive Language Technology*, (2014), 81~86. (In Korean)

Lim, S., C. Lee and M. Jang, "Restoring an Elided Entry Word in a Sentence for Encyclopedia QA System," *Proceedings of Second International Joint Conference on Natural Language Processing(IJCNLP-2005)*, (2005), 215~219.

Nariyama, S., "Grammar for ellipsis resolution in Japanese," *Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation*, (2002), 135~145.

Ng, V., "Learning noun phrase anaphoricity to improve coreference resolution: Issues in representation and optimization," *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, (2004), 152~159.

Ng, V. and C. Cardie, "Improving machine learning approaches to coreference resolution," *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, (2002), 104~111.

Nguyen, N. and Y. Guo, "Comparisons of Sequence Labeling Algorithms and Extensions," *Proceedings of 24th International Conference on Machine Learning*, (2007), 681~688.

Sarawagi, S. and R. Gupta, "Accurate Max-margin Training for Structured Output Spaces," *Proceedings of 25th International Conference on Machine Learning*, (2008), 888~895.

Seki, K., A. Fujii, and T. Ishikawa, "A probabilistic method for analyzing Japanese anaphora integrating zero pronoun detection and resolution," *Proceedings of 19th International Conference on Computational Linguistics*, (2002), 911~917.

Shalev-Shwartz, S., Y. Singer, N. Srebro, "Pegasos: Primal estimated subgradient solver for SVM," *Proceedings of 24th International Conference on Machine Learning*, (2007), 807~814.

Shalev-Shwartz, S., Y.Singer, N. Srebro, A. Cotter, "Pegasos: Primal estimated subgradient solver for SVM." *Mathematical Programming*, Vol. 127, No. 1, (2011), 3~30.

Soon, W. M., H. T. Ng, and D. C. Y. Lim, "A machine learning approach to coreference resolution of noun phrases," *Computational Linguistics*, Vol. 27, No. 4, (2001), 521~544.

Tsochantaridis, I., T. Hofmann, T. Joachims, Y. Altun, "Support vector machine learning for interdependent and structured output space," *Proceedngs of 21st International Conference on Machine Learaning*, (2004), 104~111.

Tsochantaridis, I., T. Joachims, T. Hofmann, Y. Altun, "Large margin methods for structured and interdependent and structured output spaces," *Journal of Machine Learning Research*, Vol. 6, (2005), 1453~1484.

Yang, X., G. Zhou, J. Su, and C. L. Tan, "Coreference resolution using competition learning approach," *Proceedings of the 41st AnnualMeetingoftheAssociationforComputationalLinguistics*, (2003), 176~183.

Abstract

Restoring Omitted Sentence Constituents in Encyclopedia Documents Using Structural SVM*

Min-Kook Hwang* · Youngtae Kim* · Dongyul Ra** · Soojong Lim*** · Hyunki Kim***

Omission of noun phrases for obligatory cases is a common phenomenon in sentences of Korean and Japanese, which is not observed in English. When an argument of a predicate can be filled with a noun phrase co-referential with the title, the argument is more easily omitted in Encyclopedia texts. The omitted noun phrase is called a zero anaphor or zero pronoun. Encyclopedias like Wikipedia are major source for information extraction by intelligent application systems such as information retrieval and question answering systems. However, omission of noun phrases makes the quality of information extraction poor. This paper deals with the problem of developing a system that can restore omitted noun phrases in encyclopedia documents. The problem that our system deals with is almost similar to zero anaphora resolution which is one of the important problems in natural language processing. A noun phrase existing in the text that can be used for restoration is called an antecedent. An antecedent must be co-referential with the zero anaphor. While the candidates for the antecedent are only noun phrases in the same text in case of zero anaphora resolution, the title is also a candidate in our problem.

In our system, the first stage is in charge of detecting the zero anaphor. In the second stage, antecedent search is carried out by considering the candidates. If antecedent search fails, an attempt made, in the third stage, to use the title as the antecedent. The main characteristic of our system is to make use of a structural SVM for finding the antecedent. The noun phrases in the text that appear before the position of zero anaphor comprise the search space. The main technique used in the methods proposed in previous research works is to perform binary classification for all the noun phrases in the search space. The noun phrase classified to be an antecedent with highest confidence is selected as the antecedent. However, we

* Computer & Telecommunications Engineering Division, Yonsei University
** Corresponding author: Dongyul Ra
Computer & Telecommunications Engineering Division, Yonsei University
1 Yonsei-Dae-Gil, Wonju, Kangwon, Korea
Tel:+82-033-760-2246, Fax:+82-033-763-4323, Email: dyra@yonsei.ac.kr
*** SW · Content Research Lab., Electronics and Telecommunications Research Institute

propose in this paper that antecedent search is viewed as the problem of assigning the antecedent indicator labels to a sequence of noun phrases. In other words, sequence labeling is employed in antecedent search in the text. We are the first to suggest this idea. To perform sequence labeling, we suggest to use a structural SVM which receives a sequence of noun phrases as input and returns the sequence of labels as output. An output label takes one of two values: one indicating that the corresponding noun phrase is the antecedent and the other indicating that it is not. The structural SVM we used is based on the modified Pegasos algorithm which exploits a subgradient descent methodology used for optimization problems.

To train and test our system we selected a set of Wikipedia texts and constructed the annotated corpus in which gold-standard answers are provided such as zero anaphors and their possible antecedents. Training examples are prepared using the annotated corpus and used to train the SVMs and test the system. For zero anaphor detection, sentences are parsed by a syntactic analyzer and subject or object cases omitted are identified. Thus performance of our system is dependent on that of the syntactic analyzer, which is a limitation of our system. When an antecedent is not found in the text, our system tries to use the title to restore the zero anaphor. This is based on binary classification using the regular SVM. The experiment showed that our system's performance is F1 = 68.58%. This means that state-of-the-art system can be developed with our technique. It is expected that future work that enables the system to utilize semantic information can lead to a significant performance improvement.

Key Words : Encyclopedia, Noun phrase omission, Zero anaphora resolution, Structural SVM, Sequence labeling

Received : March 6, 2015 Revised : May 24, 2015 Accepted : June 2, 2015
Type of Submission : Fast Track Corresponding Author : Dongyul Ra

저 자 소 개

황 민 국

연세대학교 컴퓨터정보통신공학부 학사를 졸업하였고, 현재는 연세대학교 컴퓨터정보통신공학부 석사 과정 중이다. 관심분야는 자연어처리, 인공지능, 빅데이타 처리, 기계학습 등이다.

김 영 태

한라대학교 컴퓨터공학과 학사 졸업 후 토마토 시스템에 근무하였고, 현재는 연세대학교 컴퓨터정보통신공학부 박사 과정 중이다. 연구 관심 분야는 한국어분석, 정보검색, 기계학습, 데이터마이닝, 딥러닝 등이다.

나 동 열

현재 연세대학교 컴퓨터정보통신공학부 교수로 재직 중이다. 서울학교 전자공학과 학사, 한국과학기술원 전산학과 석사를 졸업하였고, 미시간주립대학교 컴퓨터과학과에서 박사학위를 취득하였다. 관심 연구분야는 자연어처리, 정보검색, 기계학습 등이다.

임 수 종

현재 한국전자통신연구원 지식마이닝연구실 책임 연구원으로 재직 중이다. 연세대학교 수학과 학사 졸업하여 동대학 컴퓨터과학 석사, 전산학 박사 학위를 취득했다. 관심 분야는 자연어처리, 기계학습, 지식 처리 등이다.

김 현 기

현재 한국전자통신연구원 지식마이닝연구실 실장으로 재직 중이다. 전북대학교 컴퓨터공학과 학사 졸업하여 동대학 컴퓨터공학 석사 학위 후 플로리다 대학교 전산학 박사 학위를 취득했다. 관심 분야는 자연어 이해, 텍스트 마이닝, 자연어 질의응답 등이다.