작은 화면 기기에서의 출력을 위한 신문기사 헤드라인 형식의 문장 축약 시스템

이 공 주†

요 약

모바일 디바이스와 같이 작은 크기의 화면을 갖는 기기에서는 긴 문장의 내용을 한눈에 파악하기가 쉽지 않다. 본 논문에서는 신문기사와 그 헤드라인으로부터 추출한 정보로부터 문장을 자동으로 축약할 수 있는 시스템을 제안하고자 한다. 축약된 문장은 문장 내의 필수적이지 않은 요소들을 제거함으로써 그 기본 의미는 그대로 전달하되 문장의 길이를 축소시킨 것이다. 신문기사의 헤드라인으로부터 문장 축약 방법을 학습하였기 때문에 매우 간결한 형태로 문장을 축약할 수 있다. 예비 실험을 통해 본 논문에서 제안하고 있는 시스템이 생성해 내는 축약문장이 유용함을 보이고자 한다.

키워드 : 문장 축약, 문서 자동 요약, 모바일 디바이스, 신문기사 헤드라인

Sentence Compression of Headline-style Abstract for Displaying in Small Devices

Kong Joo Lee†

ABSTRACT

In this paper, we present a pilot system that can compress a Korean sentence automatically using knowledge extracted from news articles and their headlines. A set of compressed sentences can be presented as an abstraction of a document. As a compressed sentence is of headline-style, it could be easily displayed on small devices, such as mobile phones and other handheld devices. Our compressing system has shown to be promising through a preliminary experiment.

Key Words : Sentence Compression, Document Summarization, Mobile Device, News Headline

1. 서 론

무선 인터넷의 발전에 따라 휴대전화나 PDAs(Personal Digital Assistants)와 같은 모바일 디바이스들이 급속히 발전해 가고 있는 추세다. 그러나, 이러한 모바일 디바이스는 낮은 대역폭과 작은 크기의 화면, 적은 메모리 등으로 다양한 방식의 사용에 제약이 따른다. 특히 모바일 디바이스들의 작은 화면은 한번에 볼 수 있는 정보의 양에 큰 제한을 주어, 다소 긴 내용의 문서를 모바일 디바이스 화면에서 보는 것을 매우 어렵게 한다. 자동 문서 요약 기술은 이러한 문제를 해결해 줄 수 있는 핵심 기술 중 하나라 할 수 있다[14]. 대부분의 자동 문서 요약 시스템은 원형 문서로부터 몇 개의 기준을 이용하여 중요한 문장들을 추출하고 나열함으로써 문서를 요약하였다[6]. 그러나, 요약으로써 추출된 문장들조차도 길고 복잡하기 때문에 모바일 디바이스와 같은 작은 화면을 갖는 기기에서 표시하기에는 쉽지 않다[15]. 따라서 긴 문장을 다소 짧고 간결한 문장으로 축약해 주는 작업이 작은 화면을 갖는 디바이스에서의 출력에 매우 도움이 된다고 하겠다.

일반적으로 신문기사는 헤드라인과 본문으로 구성되어 있다. 헤드라인은 본문을 대표할 수 있는 가장 핵심적인 요약으로 간주할 수 있다. 또한 Wasson은 신문기사의 본문은 일반적으로 본문의 내용을 요약한 선두(leading) 문장으로 시작하는 경향이 있다는 사실을 밝혀냈다[13]. 대부분의 경우, 헤드라인은 본문의 선두문장보다 짧다. 그렇기 때문에, 신문기사의 헤드라인은 본문의 선두문장의 축약된 형태라고 생각할 수 있다. 본 논문은 이와 같은 기본 아이디어를 바탕으로 출발하였다. 본 논문에서는 신문기사의 첫 번째 문단의 첫 번째 문장을 '*선두문장(leading sentence)*'이라고 하고, 그 선두문장의 축약된 형태를 '*축약문장(compressed sentence)*'라고 언급하겠다. 다음의 예제를 살펴보도록 하자.

※ 이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음 (KRF-2004-003-D00328).
† 정 회 원 : 충남대학교 전기정보통신공학부 조교수
논문접수 : 2005년 5월 12일, 심사완료 : 2005년 7월 18일

(1a) ①**대구시는** 빠른 ②시일에 ③**경북도와** ④**시행정구역에** ⑤대한 ⑥공식적인 ⑦**협의를** ⑧갖기로_⑨했다.

(1b) 대구시 경북도와 시행정구역 협의

문장 (1a)는 신문기사로부터 추출한 선두문장이며, (1b)는 같은 신문기사의 헤드라인이다. 헤드라인 (1b)가 선두문장 (1a)로부터 추출되었다고 가정해 보자. 우선 상대적으로 덜 중요한 단어들이 생략된 후, ①, ④, ⑤, ⑧번 단어들만이 헤드라인에 포함되었다. 또한, 의미 전달에 문제만 없다면 조사나 어미와 같은 기능어도 헤드라인에서 생략된다. 선두문장 (1a)의 ①, ⑤, ⑧번 단어에서 각각 '는', '에', '를'이 생략되어 헤드라인 (1b)가 최종적으로 완성되었다.

대량의 신문기사를 관찰해 본 결과, 선두문장을 헤드라인으로 변환시킬 때 적용 가능한 다음과 같은 세 가지 기본 규칙을 발견할 수 있었다. 첫 번째는 선두문장으로부터 부사구와 같은 부가적인 단어들을 제거한다. 두 번째는 길고 복잡한 표현을 짧고 간단한 표현으로 바꾸는 것이다. 마지막 세 번째는 문장 의미의 변화만 없다면 조사와 같은 기능어도 생략한다.

본 논문에서는 헤드라인을 선두문장의 축약문장으로 간주하고, 선두문장과 그 헤드라인을 모아놓은 신문기사 코퍼스로부터 문장을 축약하는 방법을 자동으로 학습할 수 있는 시스템을 제안하고자 한다. 이와 같은 시스템에 의해 축약된 문장은 신문기사의 헤드라인의 형태를 띠고 있기 때문에 비교적 간결한 형태의 축약이 될 수 있으며, 동시에 작은 크기의 화면을 갖는 기기에서의 출력에 가장 적합할 것이다. 또한, 이와 같은 접근 방법은 신문기사 코퍼스만 있다면 다른 언어에도 쉽게 적용해 볼 수 있을 것으로 기대된다.

본 논문은 다음과 같이 구성되어 있다. 2장에서 관련연구를 간단히 살펴보고, 3장과 4장에서 시스템 모델링 및 구현에 대해서 살펴본다. 5장에서 시스템의 유용성을 실험을 통해 살펴보고, 마지막으로 결론을 맺고자 한다.

2. 관련 연구

자동 문서 요약에 대한 연구는 인터넷의 문서 양의 폭발적인 증가와 함께 급속한 발전을 하고 있다[11]. 자동 문서 요약에 대한 최근의 연구 경향은 원문서로부터 중요한 문장들을 추출하여 이를 재배치함으로써 요약을 생성해 내는 추세이다. 본 논문은 신문기사의 선두문장을 요약으로 간주하는 접근 방법[13]을 취한다. 선두문장은 대부분의 경우 매우 짧기 때문에 모바일 디바이스와 같은 기기에서의 출력에 적합하다.

[8]에서는 원문서의 가장 중요한 정보를 전달할 수 있는 문법적 요약을 만들어 내기 위해 노이지 채널(noisy-channel)과 결정 트리(decision tree)를 사용하였다. 기본 아이디어는 노이지 채널 모델을 이용하여 문장 대 문장을 번역하는 것으로 번역의 목표문장이 요약문이다. 즉, 길고 복잡한 원시문장을 짧고 간단한 목적문장으로 번역함으로써 요약을 만들어 내는 것이다. 이 방법은 길이별로 다양한 축약문장을 생성해 낼 수 있는 장점이 있는 반면, 추정해야 하는 매개변수의 값이 많으며 디코딩 과정에서 최대점수를 갖는 결과를 추출하는 작업이 용이하지 않다.

[2]에서는 수동으로 작성된 휴리스틱 규칙을 이용하여 신문기사의 헤드라인을 직접 생성하고자 하였다. 입력 문장을 구문 분석 한 후, "NP와 VP를 갖고 있는 가장 왼쪽 하위(lowest leftmost) S를 선택하라"와 같은 휴리스틱 규칙을 이용하여 구문 트리에서 불필요한 요소를 제거하였다. 이와 같은 방법은 매우 효율적이나 휴리스틱을 작성하는데 많은 시간이 소요되며 또한 다른 나라 언어로의 적용이 불가능해진다[1].

[15]에서는 일본어를 대상으로 문장 내지 문서를 한눈에 파악할 수 있도록 요약하는 시스템을 구축하였다. 입력 문장 내지 입력 문서에 대해 구문분석을 통해서 각 단어 사이의 구문관계를 파악하고, 그 중 가장 중요한 구문 관계들만을 모아서 불필요한 단어들을 제거한 후, 요약 문장을 생성하였다. 이 시스템의 경우, 요약 문장의 질은 구문관계 분석의 정확도에 따라 많은 차이가 나게 된다. 또한 구문관계의 중요성을 결정하는 점수부여 방법에 따라 시스템의 결과가 차이가 나게 된다.

3. 문장 축약 시스템 모델

선두문장은 우선 구문분석이 되어야 한다. 기본적으로 선두문장과 그 헤드라인에 동시에 포함된 단어들은 그렇지 못한 단어에 비해 중요 단어로 간주된다. 본 논문의 시스템은 선두문장의 구문 분석 결과로부터 축약에 포함될 단어와 그렇지 않은 단어들을 자동으로 학습함으로써 문장 축약을 수행하고자 한다.

3.1 학습 코퍼스

학습코퍼스는 신문기사의 선두문장과 헤드라인의 쌍으로 구성되었다. 학습 모델을 간단하게 하기 위해서 다음과 같

학습코퍼스 구성 조건

선두문장이 $S_l=(w_1,w_2,...,w_N)$ 이고, 그에 해당하는 헤드라인이 $S_c=(x_1,x_2,...,x_M)$ 일 때;

(C1) $N \geq M$

(C2) 다음과 같은 함수 $g: S_c \rightarrow S_l$가 존재한다

(C3) 헤드라인에서의 단어 사이의 위치가 $position(x_i) < position(x_j)$ 일 때, 그에 해당하는 선두문장에서의 위치 또한 $position(g(x_i)) < position(g(x_j))$ 이다.
여기서, 함수 $position(word)$는 $word$의 문장에서의 위치를 의미한다.

(C4) $g(x)=x$ 또는 $stem(g(x))=stem(x)$
여기서, 함수 $stem(x)$은 단어 x의 원형(stem)을 반환한다.

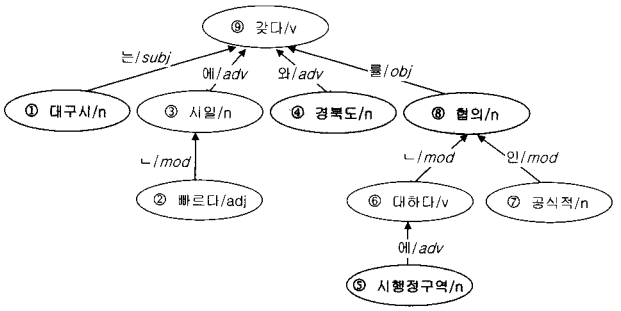
1) 한국어는 중심어 후위 언어이기 때문에, 예제에서 보인 휴리스틱의 경우, "가장 오른쪽 하위 S를 선택하라"로 규칙이 바뀌어야 한다.

은 4가지 조건을 동시에 만족하는 쌍으로만 학습 코퍼스를 구성하였다.

조건 (C1)은 헤드라인의 길이가 선두문장보다 짧거나 같아야 한다는 의미이며, 조건 (C2)는 헤드라인에 있는 모든 단어들은 선두문장에 있는 단어 중 하나로 대응되어야 함을 의미한다. 그러나 그 역은 성립하지 않는다. 조건 (C3)은 헤드라인과 선두문장에 동시에 나타나는 단어들은 문장에서의 그 순서가 동일해야 함을 의미한다. 마지막 (C4) 조건은 선두문장과 헤드라인에 동시에 나타나는 단어들은 동일한 원형을 갖고 있어야 함을 의미한다. 앞에서 살펴 본 예제문장 (1a)와 (1b)는 위의 4가지 조건을 모두 만족한다. 신문기사 코퍼스[1]로부터 위의 4가지 조건을 모두 만족하는 1,304 쌍의 헤드라인과 선두문장을 추출하였으며, 이를 이용하여 학습데이터로 사용하고자 한다.

3.2 표지트리(marked tree)

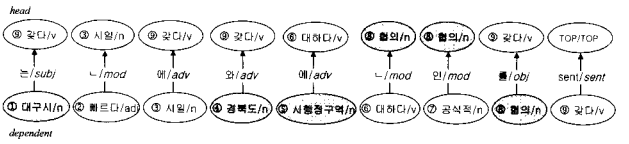
본 논문에서 '표지트리'라 함은 표지된(marked) 노드를 갖고 있는 구문 트리를 의미한다. 구문 트리 각각의 노드는 입력 단어와 해당 품사 그리고 표지 (0 또는 1) 심볼을 갖고 있으며, 각각의 에지는 노드 사이의 문법적 관계를 나타낸다. 선두문장을 구문 분석한 후, 표지트리로 변환하는데, 헤드라인과 선두문장에 모두 포함된 단어들에 해당하는 노드들이 표지트리에서 '1'로 표시되며, 그렇지 못한 노드들이 '0'으로 표시된다. (그림 1)은 문장 (1a)에 대한 표지트리를 보여준다. 노드 ①, ④, ⑤, ⑧이 헤드라인과 선두문장 모두에 포함되었기 때문에 '1'로 표지되었다. (그림 1)에서는 '1'로 표지된 노드를 회색 노드로 표시하였다. (그림 1)에서 각각의 에지는 각 노드들 사이의 문법적 관계를 나타내는 조사로써 표시되었다.



(그림 1) 선두문장 (1a)에 대한 표지트리

3.3 문장 축약 모델

본 논문에서 제안하는 시스템은 입력 문장을 신문기사의 헤드라인과 같은 스타일로 축약하는 것이다. 축약문장은 표지 정보를 갖고 있는 표지트리(marked tree)로부터 생성된다. 그렇기 때문에 구문 분석 결과, 구문 트리에서 어떤 노드들을 '1'로 표지할 것인지가 가장 중요한 문제라고 할 수 있다. 이와 같은 문제를 해결하기 위하여 가능한 모든 표지트리의 점수를 [3, 7]에서와 같이 계산한다. 가장 높은 점수를 받은 표지트리가 결정되면 이 트리의 표지된 노드들만을



(그림 2) 문장 축약 모델을 위한 표지트리의 재구성

〈표 1〉 그림 1에 대한 기호 표시 예제

i	1	2	3	4	5	6	7	8	9
w_i	대구시	빠르다	시일	경북도	시행정구역	대하다	공식적	협의	갖다
t_i	n	adj	n	n	n	v	n	n	v
m_i	1	0	0	1	1	0	0	1	0
$h(i)$	9	3	9	9	6	8	8	9	0
r_i	subj	mod	adv	adv	adv	mod	mod	obj	SENT

모아서 축약된 문장을 생성해 낼 수 있다.

(그림 2)는 문장 축약 시스템 모델링을 위해 (그림 1)의 표지트리를 단순화하여 재구성한 것이다. (그림 2)에서 각각의 노드는 한 개의 중심(head) 노드를 갖고 있으며, 마지막 ⑨번 노드의 경우 (TOP/TOP)을 중심 노드로 설정하였다. 노드 (TOP/TOP)은 문장의 최상위 가상 노드이다. 구문 트리에서 n_i는 i번째 단어에 해당하는 노드로써 단어 정보 w_i, 품사 정보 t_i, 그리고 표지 정보 m_i로써 구성된다. 즉, $n_i = (w_i, t_i, m_i)$이다. $h(i)$를 노드 n_i의 중심 노드 문장 내 위치라 하고, r_i를 노드 n_i와 $n_{h(i)}$의 문법적 관계라고 하면, 노드 n_i와 $nh(i)$를 잇는 에지는 $e_i = (n_i, r_i, n_{h(i)})$로 표시할 수 있다. 표지트리 T_m은 에지들의 리스트로 표현할 수 있으며, 즉, $T_m = (e_1, e_2, ...e_N)$이며, 표지트리 T_m의 점수는 다음의 식 (1)과 같이 계산할 수 있다.

$$Score(T_m) = \Pr(T_m) \cong \prod_{i=1}^{N} \Pr(m_i = 1 \mid e_i) \quad (1)$$

N은 표지트리의 노드 수이다. 표지트리의 점수 값이 높을수록 그 표지트리로부터 좋은 축약문장이 생성될 수 있다. <표 1>에서는 (그림 1)의 표지트리에서의 각 기호의 사용 예제를 보여주고 있다. 기호 표시의 일관성을 위해서 루트 노드 n_9의 중심 노드는 '0'로 표시하였으며, 그 문법적 관계 또한 'SENT'로 표시하였다.

수식 (1)의 확률 $\Pr(m_i = 1 \mid e_i) = \Pr(m_i = 1 \mid n_i, r_i, nh(i))$의 값은 최우추정법(maximum likelihood estimation)[10]으로 계산할 수 있다. 이후의 실험에서는 데이터 부족 문제를 완화시키기 위해 이 확률 값을 수식 (2)와 같이 평탄화한다.

$$\Pr(m_i = 1 \mid n_i, r_i, n_{h(i)}) \cong \Pr(m_i = 1 \mid t_i, r_i, t_{h(i)}, m_{h(i)}) = \frac{count(m_i = 1, t_i, r_i, t_{h(i)}, m_{h(i)})}{count(t_i, r_i, t_{h(i)}, m_{h(i)})} \quad (2)$$

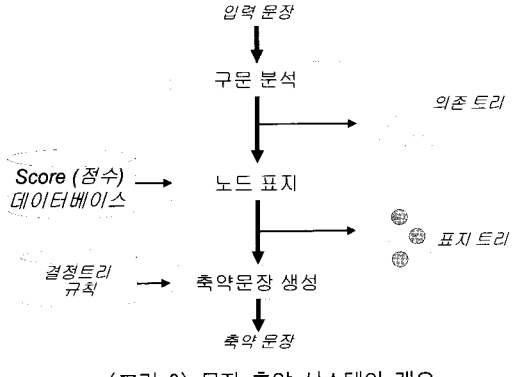
$count(x)$는 코퍼스에서 x의 출현 빈도를 의미한다. 수식 (1)의 계산을 간단히 하기 위하여 우선 입력 문장에 대한 구문 트리의 결과는 항상 한 개라고 가정한다. 입력 문장의 구문 트리가 N개의 노드를 갖고 있다고 가정할 때, 2^N개의 표지트리가 가능하다. 입력 문장이 길어짐에 따라 2^N의 수는 기하급수적으로 증가할 수 있으므로 수식 (1)의 계산을 효율적으로 수행하기 위해 Greedy 방법을 이용한다. 이 방법에서는 각 노드의 중요도(significance)를 정의하고 이 값이 가장 높은 노드부터 우선적으로 선택하여 표지한다. 각 노드의 중요도(significance)는 다음의 수식 (3)에 의해 계산되어지며, 높은 중요도 값을 가질수록 그 노드가 '1'로 표시될 확률이 높아진다고 간주한다. 각 노드의 중요도는 학습 코퍼스에서 그 노드가 헤드라인에 얼마나 자주 포함되었는가, 그 노드의 중심노드가 포함되어진 상태에서 그 노드가 헤드라인에 얼마나 포함되어졌는가와 그 노드의 구문 트리에서의 노드 깊이의 반비례로 계산되어진다. 각 노드의 중요도 $M(n_i)$는 수식 (3)과 같이 정의한다.

$$M(n_i)=\Pr(m_i=1|w_i,t_i,r_i)\times\Pr(m_i=1|m_{h(i)}=1,w_i,t_i,r_i,w_{h(i)},t_{h(i)})\times\frac{1}{d(n_i)+\alpha} \quad (3)$$

$d(n_i)$는 구문 트리 상에서 노드 n_i의 깊이이며, α는 상수 값이다. 수식 (3)의 우변의 두 번째 항은 자식 노드와 그 중심 노드 사이의 의존 관계를 고려한 것이다. 다시 말하면, 문장에서 수식어와 피수식어의 관계 사이의 의존성을 나타낸 것이다. 세 번째 항은 한국어 문장에서 일반적으로 주절이 종속적에 비해 더 중요함을 나타낸 것이다. 수식 (3)의 우변의 첫 번째, 두 번째 항은 수식 (2)와 같은 방법으로 평탄화될 수 있다.

4. 문장 축약 시스템

문장 축약 시스템의 개요가 (그림 3)에 제시되고 있다. 문장 축약의 첫 번째 단계는 입력 문장을 의존 트리로 분석하는 구문 분석 단계이다. 두 번째 단계는 의존 트리의 각각의 노드에 대해서 표지(marking)를 할 것인지를 결정하는 단계이며, 마지막 과정이 표지트리로부터 축약된 문장을 생성해 내는 것이다.



(그림 3) 문장 축약 시스템의 개요

4.1 구문 분석

한국어 확률 구문 분석기[9]을 이용하여 입력 문장을 분석한다. 본 시스템이 사용하는 구문 분석기는 의존트리를 출력으로 내지 않고, 구구조 형식의 트리를 출력으로 낸다. 두 구조는 서로 상호 호환적이며, 한 구조를 다른 구조로 변환시키는 작업이 용이하다고 알려져 있다[4, 5]. 원래의 구문 분석기를 조금 수정하여 (그림 1)과 같은 의존 트리를 출력하였다.

4.2 노드 표지

입력 의존 트리의 각각의 노드에 대해서 축약문장에 포함시킬지 아닐지를 노드 표지 과정을 통해서 결정한다. 앞서 언급한 바와 같이 Greedy 방법을 통해서 K개의 선택된 노드를 갖는 표지트리를 결정한다. 각 노드의 중요도는 수식 (3)과 같은 방법으로 결정하며, 수식 (3)에서 α 값은 2.0으로 하였다. 가장 높은 점수를 갖는 표지트리를 결정하기 위한 알고리즘이 (그림 4)에 제시되고 있다.

수식 (1)의 표지트리의 점수는 절대값이기 때문에, 선택된 노드 개수의 차이로 인한 보정이 필요하다. 그렇기 때문에 (그림 4)의 알고리즘에서는 표지된 노드 개수를 정규화시키기 위해, 점수의 값을 $\sqrt[k]{Score(Tm)}$로 보정하여 사용한다.

```
node_marking(T, N)
// 입력은 N개의 노드를 갖고 있는 구문트리 T
// 출력은 가장 높은 점수를 갖는 표지트리 tm_max
begin
  score_max = 0;          // 수식 (1)의 가장 높은 값을 저장
  tm_max = NULL;          // 수식 (1)의 가장 높은 값을 갖는 표지트리를 저장
  calculate M(n_i) for all nodes n_i in T using Equation (3);
  for k = 1 to N do
    choose the most proper marked tree T_m
          with the maximum Score(T_m) of Equation (1)
          for k marked nodes;
    if (k√Score(T_m) > score_max) then
        score_max = k√Score(T_m);
        tm_max = T_m;
    end if
  end for
  return tm_max;
end
```

(그림 4) 가장 좋은 점수를 갖는 표지트리를 결정하기 위한 알고리즘

4.3 축약문장 생성

표지트리를 입력으로 받아 형태소 문제를 해결한 후, 축약문장을 생성해 낸다. 본 논문의 시스템은 원문장의 어순과 동일한 축약문장을 생성해 내도록 되어 있다. 축약문장을 생성해 내는 방법은 '**그대로 생성하기**(as-is)', '**축약하기**(omission)', '**대치하기**(replacement)' 등 세 가지로 나누어 볼 수 있다. '**그대로 생성하기**'는 원문장의 단어에 아무런 변화도 가하지 않은 상태로 축약문장에 사용하는 것이며, '**축약하기**'는 원문장의 단어에서 조사와 같은 기능어를 생략하는 것이다. 예를 들어, 입력 단어가 '협의를'이었을 때, 조사 '를'을 생략하여 '협의' 만을 축약문장에서 사용하는 경우이다. '**대치하기**'는 원문장의 입력과 의미는 동일하면서 완전히 다른 형태의 단어로 바꿔치기 하는 경우이다. 한국어의 경우, 한자어로 바꿔 쓰는 경우가 대표적이다. 예를 들어

"늘고있다" 대신 한자어인 "증가(增加)"로 바꿔 쓰는 경우가 '**대치하기**'이다. 다음의 문장 (1b)는 문장 (1a)의 헤드라인이며, 문장 (1c)는 표지트리에서의 표지된 노드의 리스트이다. 축약문장 생성은 문장 (1c)를 입력으로 받아 문장 (1b)와 유사한 결과를 생성해 내는 것이다.

(1b) 대구시 경북도와 시행정구역 협의
(1c) 대구시**는**경북도와 시행정구역**에** 협의**를**

본 논문의 버전에서는 '**대치하기**'는 구현하지 않았으며, '**대치하기**'는 단순히 '**그대로 생성하기**'로 하였다. 결정 트리 C4.5를 이용하여 축약문장 생성하기를 구현하였다. C4.5를 학습하기 위해 사용한 학습 데이터가 <표 2>에 제시되고 있다. 입력 문장에서의 문장 위치, 실질어의 품사, 기능어의 품사가 학습 데이터의 속성정보로 사용되며, 학습 결과는 축약문장을 어떻게 생성할 것인지의 생성 방법이다. <표 2>의 4번째 열에 해당하는 생성 방법은 문장 (1b)와 (1c) 사이의 각각의 단어의 차이를 통해서 쉽게 얻어낼 수 있다. C4.5가 학습해 낸 결정 트리 규칙을 이용하여 문장 (1c)와 같은 입력으로부터 문장 (1b)와 같이 최종적인 축약문장을 생성할 수 있다.

〈표 2〉 C4.5를 학습하기 위해 문장 (1b)와 (1c)에서 추출한 학습 데이터

문장 위치	선두문장 (1c)로부터 추출한 단어들		생성 방법
	실질어의 품사	기능어의 품사	
1	proper_noun (대구시)	jxt (는)	생략하기
2	proper_noun (경북도)	jca (와)	그대로 생성하기
3	noun (시행정구역)	jca (에)	생략하기
4	verbal_noun (협의)	jco (를)	생략하기

5. 실험 결과

5.1 실험 코퍼스

실험 코퍼스로는 신문 기사 헤드라인과 그에 해당하는 선두문장으로 구성된 1,304 쌍의 데이터를 사용하였다. 이 데이터는 신문기사 코퍼스[1]로부터 추출한 기사의 헤드라인과 그 선두문장 중에서 3.1절에서 언급한 4가지 조건을 모두 만족하는 쌍으로만 구성되었다. 실험 코퍼스에 있는 선두문장의 평균 길이는 12.67 단어이며, 헤드라인의 경우 5.77 단어였다. 교차 검증(cross validation)을 위해 학습 코퍼스를 10등분하였으며, 9등분을 학습으로 나머지 한 등분을 실험 데이터로 사용하였다.

5.2 실험 결과

시스템 평가 기준은 정확률(precision)과 재현율(recall)을 사용한다. 정확률과 재현율은 아래와 같이 계산하였다.

$$\text{정확률} = \frac{\text{정확히 표지된 노드 개수}}{\text{시스템에 의해 표지된 노드 개수}}$$

$$\text{재현율} = \frac{\text{정확히 표지된 노드 개수}}{\text{학습코퍼스에서 표지된 노드 개수}}$$

본 논문에서 제안한 문장 축약 시스템과 비교하기 위한 기본 모델을 설정하고자 한다. 기본 모델은 마찬가지로 입력 문장에서 K개의 노드를 표지하여 축약문장을 생성해 내는데, K개의 노드는 학습 코퍼스의 헤드라인에서 가장 자주 사용된 단어로 선정한다. 또한 노드 개수 K는 (그림 4)의 알고리즘에 의해 결정하였다.

<표 3>에서 보는 바와 같이 재현율이 정확률에 비해 훨씬 높은 결과를 보였다. 이는 두 모델이 생성해 낸 축약문장의 길이가 코퍼스에 나타난 헤드라인의 길이보다 길었기 때문인 것으로 생각된다. 본 논문의 제안 모델의 정확률과 재현율은 다음과 같이 해석할 수 있을 것으로 보인다. 헤드라인에 있는 단어 중 한 단어 정도가 생성된 축약문장에 나타나지 못했으며, 생성된 축약문장 중 약 1.5 단어 정도가 잘못 나타난 것으로 해석할 수 있다. 마찬가지로 기본 모델의 정확률, 재현율도 해석할 수 있을 것이다. <표 3>의 4번째 열은 축약문장 생성 방법에 대한 결과이다. 이 결과는 4.3절에서 설명한 C4.5가 출력한 생성 방법-생략하기, 그대로 생성하기-에 대한 정확률만을 의미한다. 본 연구에서는 문장 생성의 규칙을 단순하게 적용했기 때문에, 각각 94.5%와 95.1%의 높은 결과를 얻을 수 있었다.

(그림 5)는 기본모델과 제안모델에서 실제 입력 문장을 축약한 결과이며 제일 아래 줄에는 해당 입력 문장의 실제 헤드라인이 제시되었다. 제안모델과 기본모델이 축약한 문장들에는 밑줄 친 4어절이 공통으로 사용되었음을 살펴볼 수 있다.

<표 3>의 결과만으로는 본 논문의 제안 모델이 기본 모델에 비해 월등히 뛰어난 성능을 발휘한다고 주장할 수 없다. 그렇기 때문에 축약된 문장의 성능을 평가자에 의해 평가해 보고자 한다. 평가자에게 본 논문이 제안한 시스템에 의해 생성된 축약문장과 기본 모델이 생성해 낸 축약문장을 원 입력 문장과 함께 제시하였다. 평가자는 두 개의 서로 다른 축약문장에 대해 좀더 나은 축약문장에는 1점을 나머지 문장에는 0점을 할당하였다. 할당할 수 있는 점수는 1점과 0점뿐이다. 이러한 방법은 대조적 평가 방법으로써 어떤 시스템이 다른 시스템에 비해 월등히 뛰어난지를 명확히 비교해 볼 수 있다. 평가의 공정성을 기하기 위해 3명의 평가

〈표 3〉 축약 시스템의 정확률과 재현율

	축약문장의 평균 단어수	정확률	재현율	축약문장 생성방법의 정확률
본 논문의 제안 모델	6.85	77.3%	91.95%	94.5%
기본 모델	7.02	75.2%	89.38%	95.1%

원입력문장	아남전자는 앰프의 출력기능과 음질조절기능이 대폭 향상된 고기능 컴포넌트오디오인 델타-1880GXE를 개발, 시판한다고 16일 밝혔다.
제안모델 결과	아남전자 앰프의 출력기능과 고기능 컴포넌트오디오인 델타-1880GXE 개발
기본모델 결과	아남전자 출력기능과 음질조절기능 향상된 컴포넌트오디오인 개발 밝혔다.
실제 헤드라인	아남전자, 고기능 컴포넌트오디오 델타-1880GXE 개발 시판

(그림 5) 기본모델과 제안모델의 축약문장 생성 예제

자가 동시에 평가를 수행하였다. 한 문장에 대한 3명의 평가자의 의견이 서로 상충될 경우, 다수의 의견에 따라 평가를 수행하였다. 1,304개의 평가 문장 중, 16개의 문장은 두 시스템이 똑같은 문장을 축약문장으로 생성해 냈기 때문에, 실질적인 평가는 1,288문장에 대해서 수행하였다. 실험 결과가 <표 4>에 제시되고 있다. 비록 두 모델 사이의 정확률과 재현율은 큰 차이를 보이지 않았지만, 평가자에 의한 축약문장의 비교 평가에서는 본 논문에서 제안한 모델이 훨씬 좋은 결과를 보임을 알 수 있었다. 본 논문에서 제안한 모델이 생성한 축약문장과 기본 모델이 생성한 축약문장에는 공통적으로 포함된 단어들이 많이 있다. 그럼에도 불구하고 본 논문에서 제안한 모델이 훨씬 좋은 결과를 보이고 있다. 이는 기본 모델의 경우, 축약문장 생성시 수식어와 피수식어 사이의 의존 관계를 고려하지 않기 때문이다. 즉, 기본 모델의 경우, 피수식어 없는 수식어만이 포함된 축약문장을 만들 수 있기 때문이다. 이렇게 생성된 문장은 매우 부자연스러운 문장이 되며, 원문장의 의미를 정확히 전달할 수 있는 축약문장이 아닌 것이다.

〈표 4〉 두 모델이 생성한 축약문장의 비교 평가 결과

	대조적 평가 결과
본 논문에서 제안한 모델	74.7%
기본 모델	25.3%

6. 결 론

본 논문에서는 신문기사의 헤드라인과 선두문장의 쌍으로부터 자동으로 문장을 축약할 수 있는 방법을 학습하는 시스템을 제안하였다. 이렇게 축약된 문장은 신문기사의 헤드라인 형식을 갖고 있기 때문에 가장 간결한 형태의 축약이라고 할 수 있다. 또한, 이와 같은 접근 방법은 신문기사의 헤드라인과 선두문장 쌍의 코퍼스만 존재한다면, 한국어뿐만 아니라 다른 언어에도 쉽게 적용해 볼 수 있을 것으로 기대된다.

축약문장의 표층 표현 생성 단계는 축약문장 생성을 위한 노드를 선택하는 단계 못지 않게 중요하다. 차후 연구과제 중 하나는 다소 길고 복잡한 표현을 좀더 간결한 표현으로 바꿔 쓸 수 있도록 시스템을 개선하는 것이다. 이렇게 함으로써 훨씬 효과적이며 자연스러운 문장 축약을 완성할 수 있을 것이다.

참 고 문 헌

[1] 맹성현, 장동현, 송사광, 김지영, 이석훈, 이응봉, 이준호, 서정현, (1999) "정보검색 테스트 컬렉션 구축 및 유효성 평가". *제 11회 한글 및 한국어 정보처리 학술대회 학술지.*

[2] Dorr, Bonnie, Zajic, D., and Schwartz R. (2003). "Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation." *Proc. of the HLT-NAACL 2003 Text Summarization Workshop.*

[3] Chung, H. and Rim, H.-C. (2003). "A New Probabilistic Dependency Parsing Model for Head-final, Free Word-order Languages." *IEICE Trans. on Information and Systems*, vol. E86-D, no. 11.

[4] Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*, Ph.D. Thesis. Department of Computer and Information Science, University of Pennsylvania.

[5] Gaifman, H. (1965). "Dependency Systems and Phrase-structure Systems." *Information and Control,* 8:304-307.

[6] Hovy, E. and Lin, C.-Y. (1999). "Automated Text Summarization in SUMMARIST system." Eds. I. Mani and M. T. Maybury, *Advances in Automatic Text Summarization.* MIT Press. pp. 81-94.

[7] Kato, Y., Matsubara, S., Toyama, K., and Inagaki, Y. (2001). "Efficient Incremental Dependency Parsing." Proceedings of *IWPT 2001.*

[8] Knight, K. and Marcu, D. (2002). "Summarization Beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression." *Artificial Intelligence*, 139:91-107.

[9] Lee, K. J., Kim, J.-H., Han, Y. S. and G. C. Kim (1997). "Restricted Representation of Phrase Structure Grammar for Building a Tree Annotated Corpus of Korean." *Natural Language Engineering*, vol. 3, part 2&3, pp. 215-230.

[10] Manning, C. D. and Schutze, Hinrich. (1999). *Foundations of Statistical Natural Language Processing*, The MIT Press.

[11] Mani, I. and Maybury, M. T. (1999). *Advances in Automatic Text Summarization,* The MIT Press.

[12] Vandeghinste, V. and Tjong Kim Sang, E. (2004). "Using a Parallel Transcript/Subtitle Corpus for Sentence Compression." *Proceedings of LREC2004.* ELRA. Paris.

[13] Wasson, M. (1998). "Using leading text for news summaries: Evaluation Results and Implications for Commercial Summarization Applications." *Proceedings of COLING-ACL 98,* pp. 1364-1368.

[14] Yang, C. C. and Wang, F. L. (2003). "Fractal Summarization: Summarization Based on Fractal Theory." *Proceedings of SIGIR 2003,* pp. 391-392.

[15] Yoshihiro, U., Mamiko, O., Takahiro, K., and Tadanobu, M. (2000). "Toward the at-a-glance Summary: Phrase-representation Summarization Method." *Proceedings of the International Conference on Computational Linguistics,* pp. 878-884.

이 공 주

e-mail : kjoolee@cnu.ac.kr
1992년 서강대학교 전자계산학과(학사)
1994년 한국과학기술원 전산학과(공학석사)
1998년 한국과학기술원 전산학과(공학박사)
1998년~2003년 (주)한국마이크로소프트 연구원
2003년 이화여자대학교 컴퓨터학과 대우전임강사
2004년 경인여자대학 전산정보과 전임강사
2005년~현재 충남대학교 전기정보통신공학부 조교수
관심분야 : 자연언어처리, 자연어인터페이스, 기계번역, 정보검색