# 품사 정보와 템플릿을 이용한 문장 축소 방법
## (A Sentence Reduction Method using Part-of-Speech Information and Templates)

이 승 수 † 염 기 원 †† 박 지 형 ††† 조 성 배 ††††
(Seung-Soo Lee) (Ki-Won Yeom) (Ji-Hyung Park) (Sung-Bae Cho)

**요 약** 문장 축소란 원본 문장의 기본적인 의미를 유지하면서 불필요한 단어나 구를 제거하는 일련의 정보 압축 과정을 의미한다. 기존의 문장 축소에 관한 연구들은 학습 과정에서 대량의 어휘나 구문적 자원을 필요로 하였으며, 복잡한 파싱 과정을 통해서 불필요한 문장의 구성원(예를 들어, 단어나 구, 절 등)들을 제거하여 문장을 요약하였다. 그러나 학습 데이타로부터 얻을 수 있는 어휘적 자원은 매우 한정적이며, 문장의 모호성과 예외적인 표현들 때문에 구문 분석 결과가 명료하게 제공되지 않은 언어에서는 문장 요약이 용이하지 않다.

이에 본 논문에서는 구문 분석을 대체하기 위한 방법으로 템플릿과 품사 정보를 이용한 문장 축소 방법을 제안한다. 제안하는 방법은 요약문의 구조적 형태를 결정하기 위한 문장 축소 템플릿(Sentence Reduction Templates)과 문법적으로 타당한 문장 구조를 구성하는 품사기반 축소규칙(Grammatical POS-based Reduction Rules)을 이용하여 요약 대상 문장의 구성을 분석하고 요약한다. 더불어, 문장 축소 템플릿 적용 시 발생하는 연산량 증가 문제를 은닉 마르코프 모델(HMM: Hidden Markov Model)의 비터비 알고리즘(Viterbi Algorithm)을 이용하여 효과적으로 처리한다. 마지막으로, 본 논문에서 제안한 문장 축소 방법의 결과와 기존 논문의 연구 결과를 비교 및 평가함으로써 제안하는 문장 축소 방법의 유용성을 확인한다.

**키워드** : 문서 요약, 문장 축소, 문장 축소 템플릿, 품사기반 축소규칙, 은닉 마르코프 모델, 비터비

***Abstract*** A sentence reduction is the information compression process which removes extraneous words and phrases and retains basic meaning of the original sentence. Most researches in the sentence reduction have required a large number of lexical and syntactic resources and focused on extracting or removing extraneous constituents such as words, phrases and clauses of the sentence via the complicated parsing process. However, these researches have some problems. First, the lexical resource which can be obtained in learning data is very limited. Second, it is difficult to reduce the sentence to languages that have no method for reliable syntactic parsing because of an ambiguity and exceptional expression of the sentence.

In order to solve these problems, we propose the sentence reduction method which uses templates and POS (part of speech) information without a parsing process. In our proposed method, we create a new sentence using both Sentence Reduction Templates that decide the reduction sentence form and Grammatical POS-based Reduction Rules that compose the grammatical sentence structure. In addition, We use Viterbi algorithms at HMM (Hidden Markov Models)

† 학생회원 : 삼성전자 DM연구소 연구원
seungsoo47.lee@samsung.com
†† 정 회 원 : 한국과학기술연구원 지능인터랙션연구센터 연구원
pragman@kist.re.kr
††† 정 회 원 : 한국과학기술연구원 지능인터랙션연구센터 센터장
jhpark@kist.re.kr
†††† 종신회원 : 연세대학교 컴퓨터과학과 교수
sbcho@cs.yonsei.ac.kr
논문접수 : 2007년 4월 30일
심사완료 : 2008년 4월 7일


정보과학회논문지: 소프트웨어 및 응용 제35권 제5호(2008.5)

to avoid the exponential calculation problem which occurs under applying to Sentence Reduction Templates. Finally, our experiments show that the proposed method achieves acceptable results in comparison to the previous sentence reduction methods.

**Key words** : Document Summary, Sentence Reduction, Sentence Reduction Templates, Grammatical POS-based Reduction Rules, HMM, Viterbi Algorithm

## 1. 서 론

오늘날 인터넷과 정보 통신 기술의 발달로 월드와이드웹(World-Wide Web)을 통해서 사용자가 얻을 수 있는 정보의 양은 하루가 다르게 방대해지고 있다. 이러한 방대한 양의 자료들로부터 사용자는 자신이 원하는 정보를 찾기 위해서 정보 검색 시스템을 이용하지만, 대부분의 검색 결과는 직접 읽어보면서 확인해야 하기 때문에 원하는 정보를 얻기 위해서는 많은 시간과 노력이 요구된다. 그로 인하여 보다 원활하고 신속한 정보 검색 및 정보 획득을 위한 새로운 방법들이 필요하게 되었으며, 최근에는 정보 검색과 더불어 문서 요약에 대한 연구들이 진행되고 있다.

문서 요약이란 원본 문서의 기본적 의미를 유지하면서 문서의 전체적인 요지를 쉽게 파악할 수 있도록 정보를 압축하는 일련의 과정을 의미한다[1]. 문서 요약 시스템을 구성하기 위해서는 자연어의 이해 및 요약문 생성과 같은 기술들이 필수적이며 이를 위한 보다 복잡한 처리 과정이 요구된다. 기존의 전통적인 문서 요약 시스템들은 문장의 길이 혹은 문서 내의 특정 위치(예를 들어, 문서나 단락의 첫 문장 등) 및 문서 전체에서 출현 빈도가 높게 나타나는 어휘들을 문서 요약의 대표적 특징(Features)으로 사용하여 문장을 추출하고 정렬함으로써 문서를 요약하였다[11]. 그러나 이러한 방법들은 단순히 문서에서 나타난 특징들의 통계적 수치에만 의존하기 때문에 요약 과정에서 주요 어휘 또는 중요 정보가 포함되지 않는 문장이 추출될 수 있으며, 추출된 문장들이 임의로 정렬되기 때문에 생성된 요약문서의 논리적 및 의미적인 일관성이 부족하다. 이러한 문제점을 극복하기 위하여 최근 들어 다양한 방법들이 제안되고 있으며, 특히 문장 단위로 요약 과정을 수행하여 문서를 압축하는 문장 축소 방법들이 활발하게 연구되고 있다.

기존의 문장 축소 연구들은 구문 분석을 활용하여 문장의 구성원(예를 들어, 단어나 구, 절 등)을 제거함으로써 문장을 요약하였다. Jing은 자동 요약 시스템으로부터 생성된 문장을 전문가의 요약문과 유사한 형태로 만들기 위해서 구문 분석과 같은 복잡한 파싱 과정을 수행한 후, 각 문장의 불필요한 구문을 식별하고 제거하는 문장 축소 방법을 제안하였다[2]. 그러나 이 방법은 구문 지식 및 문맥 정보, 그리고 전문가에 의해 작성된 예제 말뭉치(Corpus) 등 다양한 종류의 리소스로부터 계산된 통계적 정보를 이용하기 때문에 복잡한 자연어 처리 과정과 대량의 어휘 및 구문 정보를 필요로 하였다.

Knight와 Marcu는 문법적으로 적합한 요약문을 생성하기 위해 노이지 채널 모델(Noisy Channel Model)과 의사 결정 모델(Decision-based Model)을 제안하였다[3]. 노이지 채널 모델은 구문 분석을 통해 얻은 구문 트리로부터 부모와 자식 노드 간의 확률 값을 이용하여 문장을 축소하는 방법으로써, 문법성을 판단하기 위한 소스 모델(Source Model), 중요 정보를 유지하기 위한 채널 모델(Channel Model), 그리고 가장 적절한 요약문을 선택하기 위한 해석기(Decoder)를 이용하여 문장을 축소하였다. 의사 결정 모델은 4가지 종류의 함수(SHIFT, REDUCE, DROP, 그리고 ASSIGNTYPE 함수)를 이용한 문장 축소 방법으로써, 학습 데이타로부터 각 함수의 사용 시기와 순서를 IF-Then 규칙으로 구성하고, 새로운 문장이 입력되면 사전에 정의된 IF-Then 규칙에 따라서 문장을 축소하였다. 두 가지 방법은 전문가에 의해 작성된 문장을 분석함으로써 새로운 요약문을 구축하기 위한 가능성을 보였다. 그러나 학습 데이타로부터 얻을 수 있는 어휘적 자원은 매우 한정적이며, 문장의 모호성과 예외적인 표현들 때문에 구문 분석 결과가 명료하게 제공되지 않은 언어에서는 쉽게 적용하기 어렵다.

최근에는 구문 분석 결과를 이용하지 않고 문장을 축소하기 위한 연구들도 진행되고 있다. Riezler 등은 문장 요약 과정에서 구문 분석을 대체하기 위하여 모호성 묶음(Ambiguity- Packing)과 확률적 해소 방법(Stochastic Disambiguation Methods)을 이용한 문장 압축 방법을 제안하였으며[4], Withbrock과 Mittal은 <Headline, Document> 쌍으로부터 추출된 확률적 정보를 이용하여 문장을 요약하는 방법을 제안하였다[5]. 이러한 방법들은 주로 뉴스의 헤드라인 생성에서 활용되었다.

이외에도 Nguyen은 구문 분석 결과를 대체하기 위해 기계 번역(Machine Translation) 분야에서 제안된 번역 템플릿 학습 방법(TTL: Translation Template Learning)을 이용한 문장 축소 방법을 제안하였다[6,12]. 이 방법은 원문으로부터 논리적인 요약문을 생성하기 위해 뉴스나 신문 기사, 논문 등의 원문과 전문가에 의해 작

성된 요약문을 대상으로 템플릿을 정의한 후, 요약 대상 문장이 입력되면 각각의 템플릿과 비교하여 적절한 요약문을 생성하는 방법이다. 이러한 방법은 특정 분야의 전문 지식과 문장의 어휘 정보를 기반으로 문장의 중요한 정보나 어휘를 유지하며 요약문을 생성할 수 있다. 그러나 어휘 정보만을 이용하여 템플릿을 정의하기 때문에 학습 데이타로부터 많은 수의 어휘 조합을 구성하기 어려우며, 문법적으로 적합한 요약문을 생성하기 어렵다. 현재 옥스퍼드 영어 사전(http://www.oed.com)에 수록되어 있는 단어는 약 60만개 정도이며, 이것을 조합하여 어휘 정보를 구성할 경우 거의 무한대의 경우의 수가 발생하게 된다.

본 논문에서는 구문 분석을 이용하지 않고 적절한 요약문을 생성하기 위한 방법으로 템플릿과 품사 정보를 이용한 문장 축소 방법을 제안한다. 본 논문에서 사용한 문장 축소 템플릿(Sentence Reduction Templates)은 문장을 축소하는 과정에서 중요한 정보를 유지하고 요약문의 구조적 형태를 결정하기 위해 사용된다. 그러나 기존의 템플릿을 이용한 방법은 학습 데이타로부터 모든 어휘 조합을 구성하기 어려우며, 더욱이 문법적으로 적합한 요약문을 생성하기 어렵다. 또한, 문장 축소를 위해서 대량의 템플릿 및 축소 규칙을 참조할 경우 컴퓨팅 연산량이 크게 증가하는 문제점이 있다. 이러한 문제를 해결하기 위해서 템플릿 적용 시 발생하는 연산량 증가 문제를 은닉 마르코프 모델(HMM: Hidden Markov Model)의 비터비 알고리즘(Viterbi Algorithm)을 적용함으로써 각 구문들이 나타날 수 있는 가장 높은 확률의 상태열을 복잡한 과정 없이 효과적으로 처리한다. 더불어, 문법적으로 타당한 문장 구조를 구성하는 품사기반 축소규칙(Grammatical POS-based Reduction Rules)을 정의하여 요약 대상 문장의 구성을 분석한 후, 이를 요약한다. 품사기반 축소규칙은 문장 축소 템플릿 구축 시 정의되지 않은 단어나 구, 절 등을 문법적으로 적합하게 축소시킨다.

본 논문에서 제안하는 방법은 문법적으로 적합하고 의미전달이 가능한 축소된 문장을 생성함으로써 문서 요약 분야에 효과적으로 적용될 수 있으며, 다양한 주제로 논의된 회의록 및 보고서에 대한 요약 정보를 제공할 수 있다. 또한, 본 논문의 문서 요약 방법을 이용하여 비정형화된 문서를 정형화시킴으로써 문서의 색인화, 메타 정보화 및 지식 베이스 구성을 가능하게하며, 이를 통해 정보 검색이나 추천 시스템이 사용자에게 적절한 정보를 제공해 줄 수 있다.

본 논문의 구성은 다음과 같다. 다음 장에서는 본 논문의 기반이 되는 번역 템플릿 학습 방법과 HMM의 개요에 대해서 설명하며, 3장에서는 본 연구에서 제안한 문장 축소 방법의 구성도를 소개한다. 4장에서는 템플릿을 생성하기 위한 학습 방법이 기술되며, 품사기반 축소 규칙을 이용한 HMM 기반의 문장 축소 방법이 5장에서 설명된다. 그리고 6장에서는 실험 및 결과에 대해서 논의하며, 마지막으로 본 연구에 대한 결론 및 향후 연구 과제를 기술한다.

## 2. 배 경

### 2.1 번역 템플릿 학습 방법

본 논문에서는 예제 기반 기계 번역(EBMT: Example-based Machine Translation) 분야에서 제안된 번역 템플릿 학습 방법(TTL)을 이용한다. Nagao에 의해서 최초로 제안된 예제 기반 기계 번역 방법은 말뭉치 기반의 기계 번역 방법 중의 하나로서[7], 두 개의 언어로 구성된 문장 말뭉치로부터 원본 언어의 문장(예제)과 번역 언어의 문장 간에 서로 대응되는 구문의 구조적 유사성과 편차를 비교하고, 각 구문 간의 규칙을 생성하여 번역 과정을 수행한다. 그러나 이 연구는 두 언어 사이의 문장을 비교하고 번역 규칙을 생성하기 위해서 모든 학습 데이타를 수작업으로 처리하였다.

Cicekli는 이러한 작업을 자동으로 처리하기 위해서 템플릿 기반의 번역 시스템을 제안하였다[8,9]. 템플릿은 두 개의 언어로 구성된 문장들의 쌍으로 정의되는데, 유사한 부분(단어의 어간이나 형태소)은 그대로 유지하고 서로 다른 부분은 변수로 대체함으로써 번역 템플릿을 생성하였다. 이 연구에서 제안된 번역 템플릿 학습 방법은 원본 언어의 문장과 번역 언어의 문장 사이에서 패턴 간의 대응성을 추론하였다. 그림 1은 템플릿 학습 방법을 설명하기 위해서 영어와 터키어로 구성된 두 개의 번역 예제 쌍을 보여주고 있다.

그림 1은 하나의 언어(영어 혹은 터키어)에서 나타난 두 문장 간의 유사성을 밑줄로서 나타내고 있으며, 밑줄이 그어지지 않은 부분은 서로 다른 부분으로서 문장 간의 상이성을 나타낸다. 우선 그림 1에서와 같이 한 언어의 두 문장(예제)으로부터 유사성을 발견한 후, 유사한 부분은 그대로 유지하고 서로 상이한 부분은 변수로 대체하여 새로운 번역 템플릿을 얻는다. 즉, 영어 문장의 유사성 템플릿은 "I will drink $X^E$"으로 정의할 수 있으며, 이와 대응되는 터키어 문장의 유사성 템플릿은 "$X^T$ icecegim"로 정의할 수 있다. 여기서, $X^E$는 영어 문장에서 임의의 구조로 대체될 수 있는 문장구성요소(단어, 구, 절 등)를 의미하며, $X^T$는 터키어 문장에서

I will drink orange juice ↔ portakal suyu içeceğim
I will drink coffee ↔ kahve içeceğim

그림 1 영어와 터키어로 구성된 두 개의 번역 예제 쌍

적절한 구조로 대체될 수 있는 문장구성요소를 의미한다. 즉, 그림 1과 비교하면 $X^E$은 "orange juice"와 "coffee"로 대체될 수 있으며, $X^T$는 "portakal suyu"와 "kahve"로 대체될 수 있다. 여기서, "orange juice"와 "portakal suyu"은 서로 대응되며, "coffee"와 "kahve"는 대응됨을 추론할 수 있다.

다음으로 영어 문장에서는 두 개의 상이성 템플릿 "$X^E$ orange juice"와 "$X^E$ coffee"를 얻을 수 있으며, 또한 각각의 영어 상이성 템플릿들과 대응되는 터키어 상이성 템플릿 "portakal suyu $X^T$"와 "kahve $X^T$"을 얻을 수 있다. 즉, 영어의 첫 번째 상이성 템플릿은 터키어의 첫 번째 상이성 템플릿과 대응되며, 두 번째 템플릿 또한 서로 대응된다. 마지막으로 영어의 "I will drink"는 터키어의 "icecegim"과 대응됨을 추론할 수 있다. 이와 같이 두 개의 언어로 구성된 대량의 말뭉치가 존재하다면, 두 언어 간의 문장 사이에서 발견된 유사성과 상이성을 이용하여, 문장 번역 템플릿과 구문 간의 번역 규칙을 얻을 수 있다.

### 2.2 은닉 마르코프 모델

은닉 마르코프 모델은 직접적으로 관찰되지 않은 패턴(Hidden pattern)에 대해서 관찰이 가능한 패턴(Observation Pattern)을 이용하여 모델링을 하는 이중의 확률론적 과정으로서[10], 순서 정보를 모델링하기 위해 쓰이는 대표적인 방법이다. HMM은 상태(State)라고 불리는 N개의 노드와 상태 간의 전이를 표현하는 가지(Edge)로 구성된 그래프로 볼 수 있다. 각 상태 노드에는 초기 상태 분포와 해당 상태에서 M개의 관찰 가능한 기호(Symbol) 중 특정 기호를 관찰할 확률 분포가 저장되어 있으며, 각 가지에는 한 상태에서 다른 상태로 전이할 상태전이 확률 분포가 저장되어 있다. 그림 2는 상태수가 3인 우향 HMM을 보여주고 있다.

입력열 $O=O_1, O_2, ..., O_{T-1}, O_T$이 주어지면 HMM은 외부에서 그 상태의 전이 과정을 직접적으로 알 수는 없지만, 자체의 확률 매개변수를 이용하여 마르코프 과정의 확률함수로써 모델링할 수 있다. HMM($\lambda$)은 다음과 같이 3가지 요소($\Pi, A, B$)로 표현된다.

- 초기 상태 분포($\Pi=\{\pi_i\}$): 상태들의 초기 확률 값을 표현하는 벡터
- 상태전이 확률 분포($A=\{a_{ij}\}$): 이전의 상태에서 현재의 상태로 전이할 확률로써 모델 내부의 은닉 상태들 간의 전이 확률, 즉 상태 $S_i$에서 상태 $S_j$로 전이할 확률
- 관측기호 확률 분포($B=\{b_j(k)\}$): 특정 상태에서 관찰 가능한 각 기호들에 대한 확률, 즉 상태 $S_j$에서 기호 $v_k$을 관찰할 확률

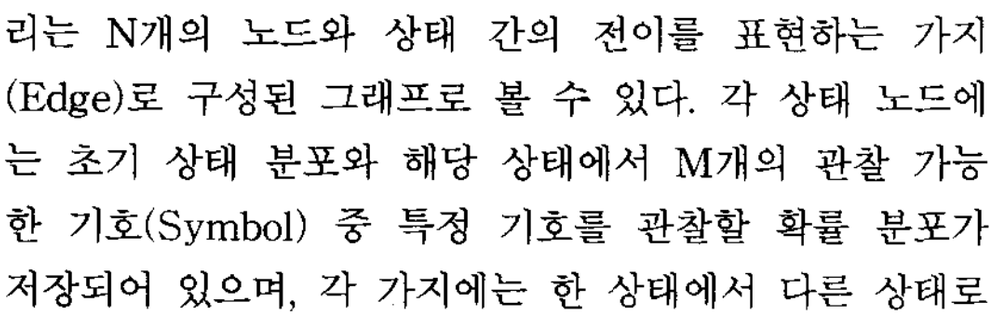


그림 2 우향(Left-to-Right) HMM의 예

이렇게 HMM을 정의하고 난 후, 모델로부터 정보를 추출하기 위해서 다음과 같은 3가지 문제를 해결해야 한다.

1. 확률 평가(Probability Estimation) 문제: 여러 개의 HMM 모델 중에서 관찰 상태의 변화를 가장 잘 표현하는 모델을 선택
2. 최적 상태 순서의 결정(Optimal Sequence) 문제: 관찰된 관측열 $O=(o_1, o_2, ..., o_{T-1}, o_T)$와 모델 $\lambda=(\Pi, A, B)$에 대하여 최적의 상태 순서를 발견
3. 매개변수 추정(Parameter Estimation) 문제: 관찰된 관측열 $O=(o_1, o_2, ..., o_{T-1}, o_T)$에 대하여 $P(O|\lambda)$을 최대로 하는 모델 $\lambda=(\Pi, A, B)$의 매개변수(parameter)를 결정

이와 같은 문제를 해결하기 위해 사용되는 대표적인 방법들이 있는데, 우선 확률 평가 문제를 해결하기 위해서는 전향 알고리즘(Forward Algorithm)과 후향 알고리즘(Backward Algorithm)을 이용하며, 최적의 상태 순서를 발견하기 위해서는 동적 프로그래밍 기법 중의 하나인 비터비 알고리즘을 이용한다. 마지막으로, 매개변수를 추정하기 위해서는 EM 알고리즘(Expectation Maximization Algorithm)으로 알려진 바움-웰치 알고리즘(Baum-Welch Algorithm)을 이용하여 처리한다. 본 논문에서는 요약 대상 문장으로부터 추출 가능한 품사 정보를 이용하여 가장 적절한 품사기반 축소규칙의 상태 순서를 발견하기 위해 비터비 알고리즘을 이용한다.

## 3. 문장 축소 방법의 구성도

기존 연구에서는 워드넷(Wordnet)이나 동의어사전(Thesaurus)과 같은 대량의 어휘나 구문 정보를 주로 이용하였으며, 복잡한 파싱 과정을 통해서 문장을 추출하고 요약하였다. 본 논문에서 제안하는 방법은 뉴스 사이트로부터 얻은 원본 문장과 축소된 문장, 그리고 전처리 과정을 통해서 얻은 문장의 품사 정보를 이용하여 템플릿을 구축한 후, 이를 입력된 문장과 비교하여 요약 문장을 생성한다. 그림 3은 본 논문에서 제안한 문장 축소 방법에 대한 구성도를 보여주고 있다.

본 논문의 문장 축소 방법은 크게 템플릿 학습부와 문장 축소부로 구성된다. 템플릿 학습부는 문장 축소 예

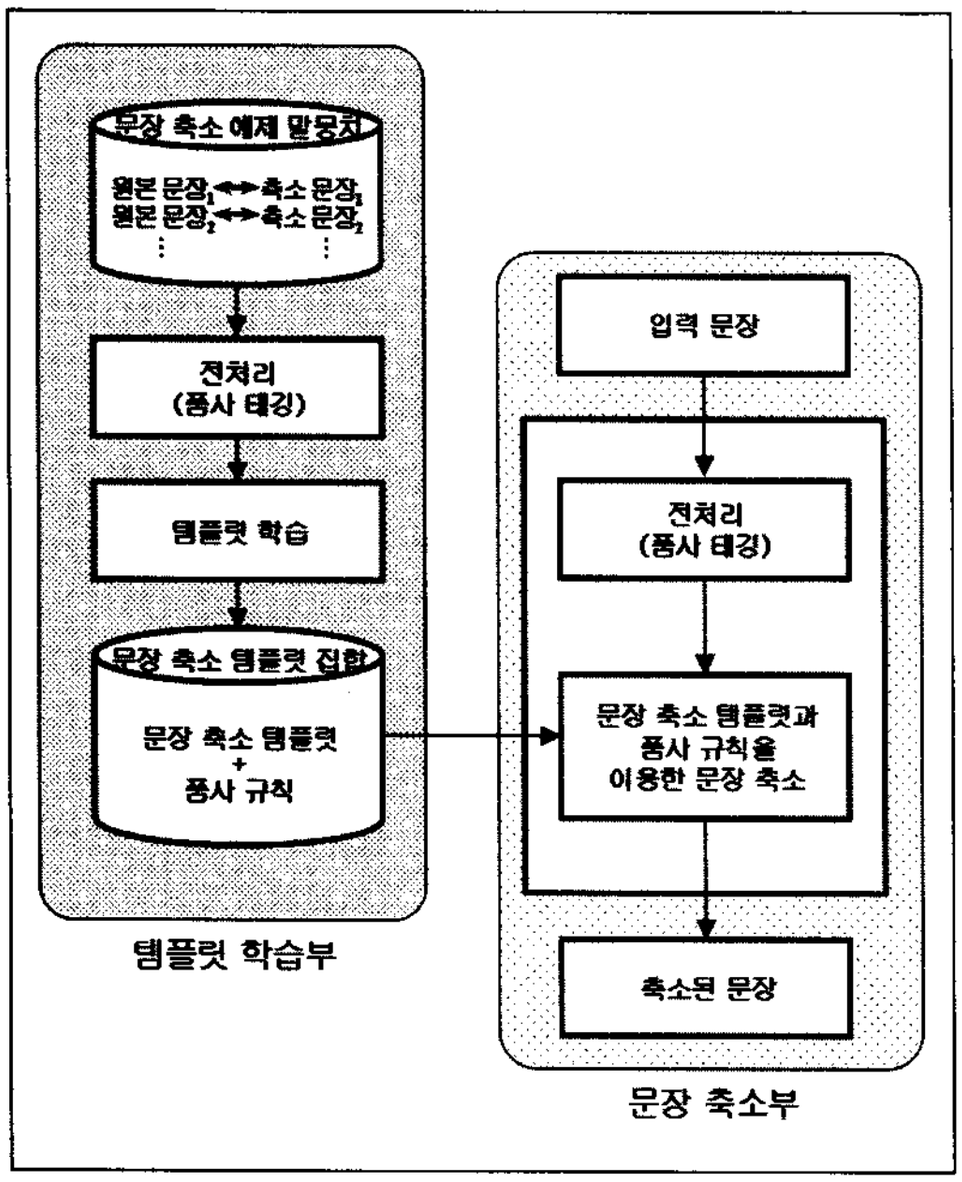


그림 3 제안한 방법의 구성도

**The documentation is concise and sufficient for normal use and maintenance** (원본 문장)

↔ **The documentation is concise** (축소된 문장)

그림 4 문장 축소 예제의 예

제 말뭉치 구축 과정, 전처리 과정, 그리고 템플릿 학습 과정으로 이루어지며, 문장 축소부는 입력 문장의 전처리 과정, 문장 축소 템플릿 및 품사기반 축소규칙을 이용한 문장 축소 과정으로 이루어진다. 우선, 학습 데이타 집합인 문장 축소 예제 말뭉치를 구축하기 위해서 뉴스 사이트로부터 원본 기사와 요약 기사를 얻은 후, 그림 4와 같이 원본/축소 문장의 쌍으로 구성된 문장 축소 예제들을 추출한다.

문장 축소 예제 말뭉치가 구축되면, 원본 문장과 축소 문장에 대해서 품사 태깅 과정을 수행하여 그림 5와 같은 품사 정보를 추출한다. 이렇게 얻은 어휘 및 품사 정보를 기반으로 템플릿 학습 방법을 이용하여 문장 축소 템플릿(Sentence Reduction Templates)과 어휘 규칙(Lexical Rules)을 생성한다. 문장 축소 템플릿은 두 개의 문장 축소 예제 사이에서 서로 공통적으로 나타난 단어들과 그렇지 않은 단어들로 세분화시킨 문장의 구조적 형태를 의미하며, 어휘 규칙은 템플릿 학습과정에서 추출된 연속적인 단어뭉치 간의 매핑 규칙을 의미한다. 마지막으로, 어휘 규칙에서 나타난 모든 단어들을 품사 정보로 대체함으로써, 품사뭉치 간의 매핑 관계를

**The/DT documentation/NN is/VBZ concise/JJ and/CC**
관사 일반명사(단수) 동사(현재) 형용사 접속사

**sufficient/JJ for/IN normal/JJ use/NN and/CC maintenance/NN**
형용사 전치사 형용사 일반명사 접속사 일반명사

↔ **The/DT documentation/NN is/VBZ concise/JJ**

그림 5 품사 정보가 태깅된 문장의 예

정의한 품사기반 축소규칙(Grammatical POS-based Reduction Rules)을 추출한다.

다음으로, 요약할 대상 문장이 입력되면 문장에 대한 품사 정보를 획득한 후, 문장 축소 템플릿과 비교하여 다수의 연속된 단어뭉치로 분리한다. 마지막으로 연속된 단어뭉치를 기반으로 적절한 품사기반 축소규칙을 선택하기 위해서 HMM 기법 중에 하나인 비터비 알고리즘을 적용하여 최선의 축소된 문장을 선택한다. 앞에서 언급한 각각의 처리 과정은 이어지는 4장과 5장에서 상세하게 기술된다.

## 4. 문장 축소를 위한 템플릿 학습 방법

본 논문에서는 기계 번역 분야에서 제안된 번역 템플릿 학습 방법(TTL)을 이용하여 문장 축소 템플릿(Sentence Reduction Templates)을 생성한다[8]. 문장 축소 방법에서 원본 문장과 축소 문장을 번역 템플릿 학습 방법의 원본 문장과 번역 문장으로 생각하면, 기계 번역 분야에서 사용하는 번역 방법을 문장 축소 방법에 적용시킬 수 있다. 본 논문에서 사용되는 문장 축소 템플릿은 요약 대상 문장을 다수의 연속된 단어뭉치로 분리함으로써, 요약문의 구조적 형태를 결정하고 구문 분석과 같은 복잡한 파싱 과정 없이 문장의 구조를 표현할 수 있다.

### 4.1 문장 축소 템플릿

문장 축소 템플릿은 아래와 같이 구문 간의 매핑 관계로 정의할 수 있다.

$$O^1\,O^2 \cdots O^k \cdots O^n \leftrightarrow R^1\,R^2 \cdots R^l \cdots R^m \quad (\text{단, } 1 \le n,m) \tag{1}$$

여기서, $O^k$는 원본 문장(Original Sentence)에서 나타난 변수나 상수를 의미하며, $R^l$는 축소 문장(Reduced Sentence)에서 나타난 변수나 상수를 의미한다. 여기서, 상수란 하나 이상으로 이루어진 연속적인 단어들의 집합(이하, 단어뭉치)을 의미하며, 변수는 임의의 상수를 취할 수 있다고 정의한다. 또한, 문장 축소 템플릿의 변수 $O^k$는 변수 $R^l$와 1:1 대응 관계를 갖는다고 정의한다.

그림 6은 문장 축소 템플릿에 대한 하나의 예를 보여주고 있다. 여기서, $O^1$과 $R^1$은 변수를 의미하며, 구문 "is concise and sufficient for normal use and maintenance"는 상수로써, 구문 "is concise"으로 축소됨을

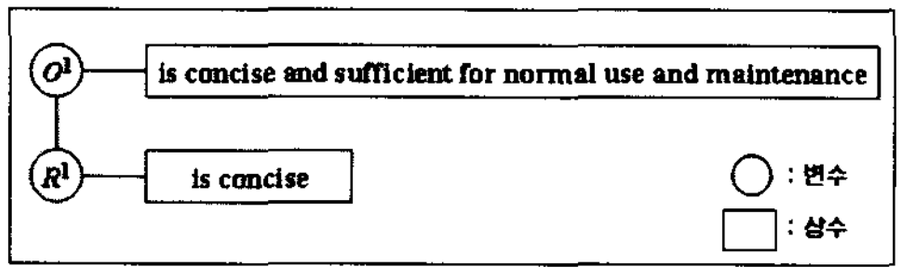


그림 6 문장 축소 템플릿의 예

보여준다. 만일, 새로운 문장 "The documentation is concise and sufficient for normal use and maintenance."가 입력된다면, 그림 6의 템플릿과 비교하여 축소된 문장 "The documentation is concise"를 생성할 수 있다.

### 4.2 문장 축소 템플릿의 생성

원본 문장과 축소 문장으로 구성된 하나의 쌍을 문장 축소 예제라고 하며, 말뭉치는 이들 예제들의 집합으로 구성된다[8]. 즉, 식 (2)에서와 같이 문장 축소 예제 $E_a$는 원본 문장 $O_a$와 축소 문장 $R_a$로 구성되며, $E_b$는 원본 문장 $O_b$와 축소 문장 $R_b$로 구성된다.

$$E_a : O_a \leftrightarrow R_a$$
$$E_b : O_b \leftrightarrow R_b \quad (2)$$

두 개의 문장 축소 예제 $E_a$와 $E_b$가 주어지면, 이들로부터 문장 축소 템플릿을 추출하기 위해서 단어 간의 유사성과 상이성을 분석하게 된다. 유사성이란 두 개의 원본 문장들($O_a$, $O_b$) 및 축소 문장들($R_a$, $R_b$) 사이에서 각각 공통으로 나타난 하나 이상의 연속된 단어들의 집합을 의미하며, 상이성은 공통으로 나타나지 않은 단어들의 집합을 의미한다. 즉, 각각의 문장($O_a$, $R_a$, $O_b$, $R_b$)을 식 (3)과 같이 단어($w$)들의 시퀀스로 생각할 경우, 유사성과 상이성은 두 개의 원본 문장과 두 개의 축소 문장 사이에서 각각 연속적으로 나타난 단어들의 집합으로 구성된다.

$$E_a : w^1_{O_a} w^2_{O_a} \cdots w^j_{O_a} \cdots w^s_{O_a} \leftrightarrow w^1_{R_a} w^2_{R_a} \cdots w^p_{R_a} \cdots w^u_{R_a}$$
$$E_b : w^1_{O_b} w^2_{O_b} \cdots w^j_{O_b} \cdots w^t_{O_b} \leftrightarrow w^1_{R_b} w^2_{R_b} \cdots w^q_{R_b} \cdots w^v_{R_b} \quad (3)$$

이 때, 원본 문장들($O_a$와 $O_b$)의 유사성 및 축소 문장들($R_a$와 $R_b$)의 유사성이 발견되지 않는다면, 문장 축소 템플릿은 말뭉치의 예제들로부터 학습될 수 없다. 반면에 유사성이 존재한다면, 식 (4)과 같이 Match Sequence $M_{a,b}$을 정의할 수 있다.

$$M_{a,b} = S^1_O D^1_O \cdots S^k_O D^k_O S^{k+1}_O \cdots S^m_O D^n_O \leftrightarrow S^1_R D^1_R \cdots S^l_R D^l_R S^{l+1}_R \cdots S^m_R D^m_R \quad (4)$$

(단, $1 \le n, m$이고, $S$: *similarity*, $D$: *difference*)

이전 절에서 정의한 바와 같이, $n, m \ge 1$ 일 때, $S^k_O$는 원본 문장($O_a$, $O_b$) 사이에서 공통으로 나타난 단어 뭉치를 가리키며 $S^l_R$는 두 개의 축소 문장($R_a$, $R_b$) 사

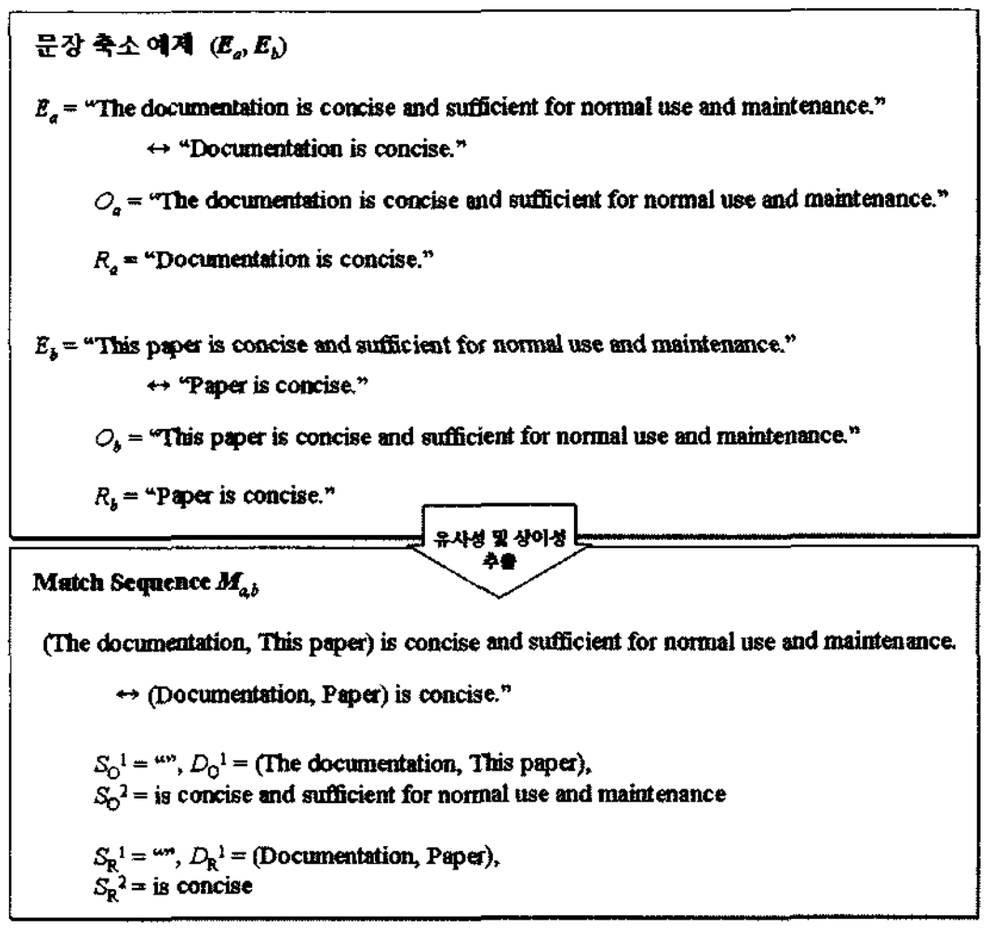


그림 7 템플릿 학습의 예

이에서 공통으로 나타난 단어뭉치를 가리킨다. 또한, $D^k_O = (D^k_{O_a}, D^k_{O_b})$와 $D^l_R = (D^l_{R_a}, D^l_{R_b})$은 $S^k_O$와 $S^{k+1}_O$, $S^l_R$와 $S^{l+1}_R$사이에서 각각 서로 상이하게 나타나는 연속적인 단어뭉치를 가리킨다. 여기서, $D^k_O$은 $D^k_{O_a}$와 $D^k_{O_b}$으로 구성되는데, $D^k_{O_a}$은 $O_a$에서 추출된 단어뭉치를 정의하며 $D^k_{O_b}$은 $O_b$에서 추출된 단어뭉치를 정의한다.

그림 7은 템플릿 학습에 대한 하나의 예를 보여주고 있다. 그림 7의 $M_{a,b}$을 살펴보면, 원본 문장($O_a$, $O_b$,) 간의 유사/상이 단어뭉치는 축소 문장($R_a$, $R_b$)간의 유사/상이 단어뭉치와 서로 대응되는 것을 볼 수 있다. 즉, 양쪽의 상이성 단어뭉치 $D^1_O$=("The documentation", "This paper")와 $D^1_R$=("Documentation", "Paper")는 서로 대응되며, 유사성 단어뭉치 $S^2_O$="is concise and sufficient for normal use and maintenance"와 $S^2_R$="is concise"은 서로 대응된다. 여기서, $M_{a,b}$의 상이성 부분인 $D^1_O$=("The documentation", "This paper")와 $D^1_R$=("Documentation", "Paper")를 각각 변수 $O^1$과 $R^1$로 대체함으로써 앞에서 보인 그림 6과 같은 문장 축소 템플릿을 생성할 수 있으며, 더불어 그림 8과 같은 세 개의 어휘 규칙을 얻을 수 있다.

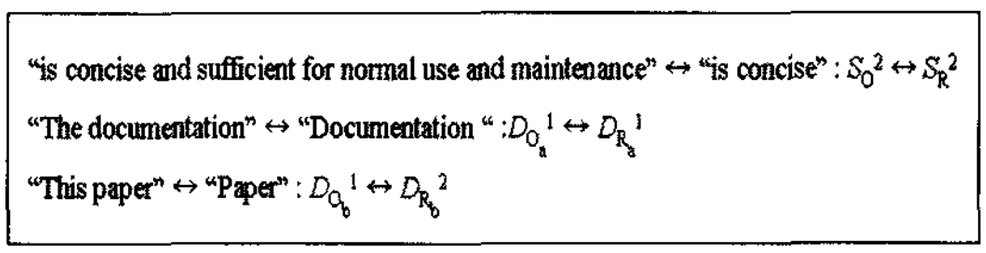


그림 8 어휘 규칙의 예

## 5. 품사기반 축소규칙을 이용한 문장 축소

이전 장에서는 템플릿 학습 방법을 이용하여 문장 축소 템플릿과 어휘 규칙을 생성하기 위한 방법에 대해서 설명하였다. 그러나 문장 축소 템플릿과 어휘 규칙을 이용하여 문장을 축소시킬 경우 문장의 구조나 중요한 구문을 유지하는 것은 용이하지만, 학습 데이타로부터 얻을 수 있는 어휘 정보는 한정적이기 때문에 학습 과정에서 나타나지 않은 어휘 정보를 문장 축소 과정에서 처리하는 것은 용이하지 않다. 더불어, 문장 축소 템플릿을 통하여 분리된 각각의 단어뭉치에서 적절한 규칙을 선택해야 할 경우, 지수승의 계량산 문제가 발생한다.

이번 장에서는 이러한 문제를 해결하기 위해서 어휘 규칙을 품사 정보로 대체하여 규칙의 일반화와 문법의 적합성을 유지시키는 품사 정보 기반의 문장 축소 방법을 설명한다. 더불어, 규칙을 선택하는 과정 중에서 발생하는 지수승의 계산량을 효과적으로 처리하기 위한 HMM 모델 기반의 문장 축소 방법에 대해서 기술한다.

### 5.1 어휘 규칙을 이용한 문장 축소의 문제점

템플릿과 어휘 규칙을 이용한 문장 축소 과정을 설명하기 위해서 그림 9와 같은 요약 대상 문장 "He said that both applications will operate on standard platforms such as Macintosh and Unix"를 이용한다.

그림 9에서와 같이 입력 문장에 대해서 두 개의 단어뭉치 "He said that"과 "will operate on"은 문장 축소 템플릿과 일치한다. 이러한 결과를 기반으로 변수 $O^2$와 $O^4$을 대체할 수 있는 가능한 모든 어휘 규칙을 찾는다. 즉, "both applications" 및 "standard platforms such as Macintosh and Unix"와 일치하는 모든 어휘 규칙을 발견한다. 그림 9는 $O^2$와 $O^4$에서 선택 가능한 5개의 어휘 규칙을 보여준다. 그러나 기존의 방법과 같이 문장 축소 템플릿과 어휘 규칙을 사용하여 문장을 축소할 경우, 다음과 같은 문제점들이 발생한다.

- 한정된 학습 데이타로 인해서 어휘 규칙 방법으로 축소되지 못하는 구문은 어떻게 처리를 해야 하나?
- 문장 축소 템플릿과 어휘 규칙을 통해서 단어뭉치가 축소될 경우, 축소된 단어뭉치 간의 관계는 문법적으로 타당한가?
- 하나의 문장 축소 템플릿이 $t$개의 단어뭉치를 가지고 있고, 각 단어뭉치가 $l$개의 규칙을 가지고 있다면, 하나의 문장을 축소하기 위해서는 $l^t$의 연산을 수행해야 한다. 이러한 지수적 연산을 어떻게 줄일 수 있을까?

이러한 문제를 해결하기 위해서, 다음 장에서는 품사기반 축소규칙을 이용한 HMM 기반의 방법을 제안한다.

### 5.2 품사기반 축소규칙을 이용한 HMM 기반의 문장 축소

#### 5.2.1 품사기반 축소규칙

어휘 규칙 기반의 문장 축소 방법은 수많은 단어 조합으로 구성된 어휘 규칙을 이용하여 문장을 축소시킨다. 그러나 학습 데이타로부터 얻은 어휘 규칙은 모든 어휘 정보를 고려할 수 없다. 예를 들어, 그림 9에서와 같이 두 개의 단어뭉치 "both applications"와 "standard platforms such as Macintosh and Unix"은 단순히 어휘 정보만을 고려하기 때문에, 한정된 학습 데이타로부터 적합한 어휘 규칙을 발견하는 것은 용이하지 않다. 더불어 기존의 문장 축소 방법은 각각의 단어뭉치 별로 축소 과정을 수행하기 때문에 단어뭉치 간의 문맥정보 및 요약문에 대한 문법성을 유지하기 어렵다.

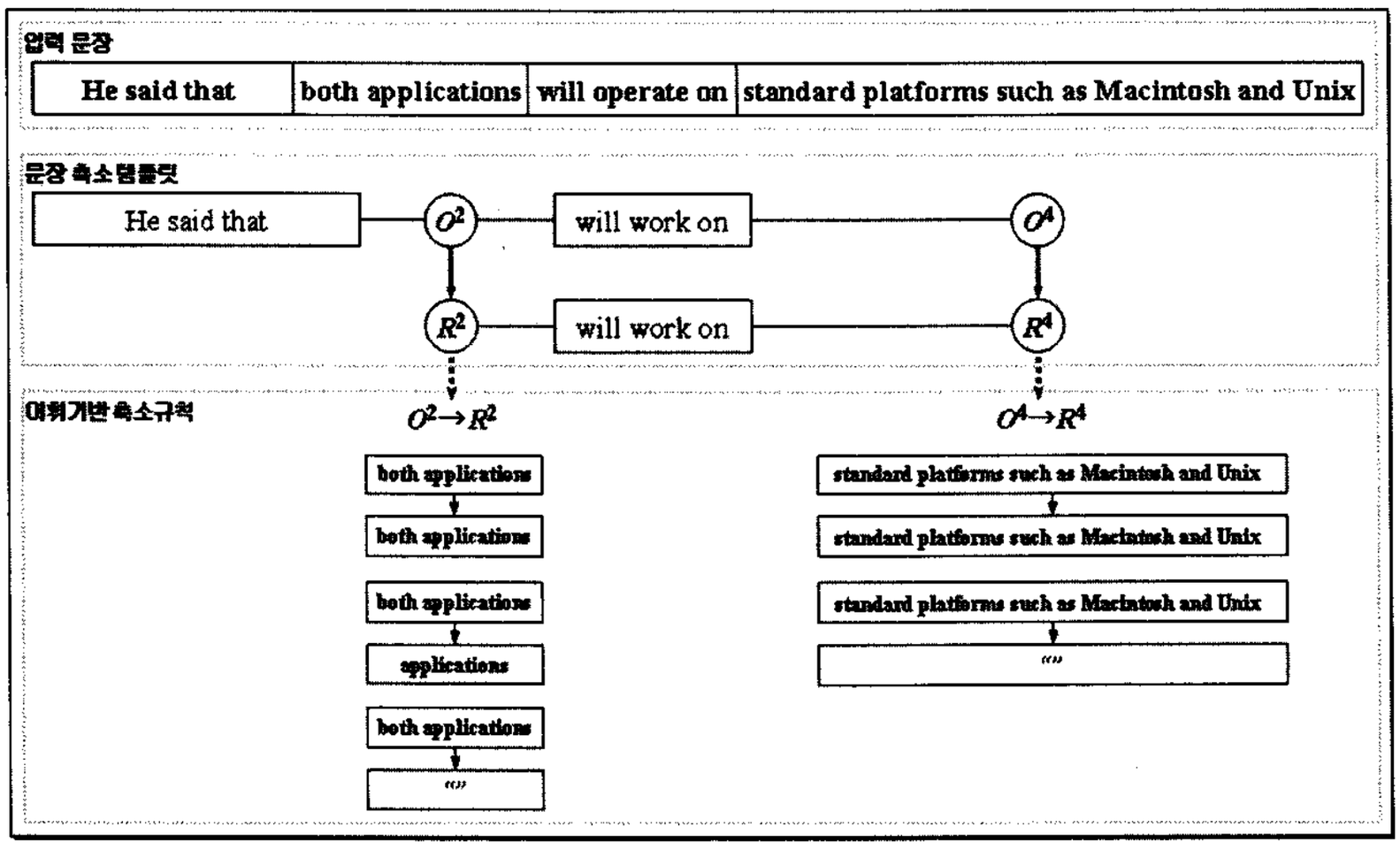


그림 9 어휘 규칙을 이용한 문장 축소의 예

이에 본 논문에서는 품사 정보를 이용하여 일반화된 축소 규칙을 유도하고 문법적으로 타당한 문장 구조를 구성하기 위해서 품사기반 축소규칙을 이용한 HMM 기반의 문장 축소 방법을 제안한다. 아래의 표 1은 단어뭉치 "The/DT PKlite/NNP compression/NN"이 나타낼 수 있는 모든 품사기반 축소규칙에 대한 정보를 설명하고 있다.

품사기반 축소규칙은 어휘 규칙을 대체한 방법으로써 학습 데이타에 포함되지 않은 어휘 정보를 용이하게 처리할 수 있으며, 문법 정보를 기반으로 문장을 축소함으로써 적절한 문법성을 보장할 수 있다. 이러한 품사기반 축소규칙은 학습 과정에서 생성된 어휘 규칙의 모든 단어 정보를 품사 정보로 대체하여 얻을 수 있다.

5.2.2 품사기반 축소규칙을 이용한 문장 축소

그림 9에서 설명된 어휘 규칙 기반의 문장 축소 방법은 수많은 단어뭉치를 구성하는데 한계가 있으며, 또한 각각의 단어뭉치 별로 축소 과정을 수행하기 때문에 요약문의 문법성을 파악하기 위해서는 추가적인 과정을 수행해야만 한다. 그러나 본 논문에서 제안한 방법은 품사정보를 이용하여 일반화된 품사뭉치를 구성하고, 각 품사뭉치 간의 전후 관계를 확률로 처리함으로써 문법적으로 타당한 요약문을 생성한다.

그림 10은 본 논문에서 제안한 문장 축소 방법을 보여주고 있다. 우선 요약할 대상 문장이 입력되면 전처리 과정을 수행하여 어휘에 대한 품사 정보를 추출한다.

다음으로, 그림 9와 같이 어휘 정보를 이용하여 문장 축소 템플릿과 비교한 후, 원본 문장($O^1$: "He said that", $O^2$: "both applications", $O^3$: "will operate on", $O^4$: "standard platforms such as Macintosh and Unix")을 다수의 단어뭉치로 분리하게 된다. 여기서, 문장 축소 템플릿은 기존의 방법과는 달리 입력대상 문장을 다수의 단어뭉치로 분리함으로써 문장의 구조적 형태를 파악하기 위해서만 사용된다. 이것은 이전 절에서도 설명한 것과 같이 단어 정보만을 활용하여 문장을 축소시킬 경우, 단어뭉치 간의 문맥 관계나 요약 문장의 문법성을 처리하는 것이 용이하지 않기 때문이다. 본 예제에서는 4개의 단어뭉치로 분리된 것을 보여주고 있다.

세 번째로, 각각 분리된 단어뭉치를 앞에서 추출한 품사 정보로 대체함으로써 4개의 품사뭉치를 얻게 된다. 이 때, 각각의 품사뭉치에서 선택할 수 있는 모든 품사

표 1 품사기반 축소규칙의 예

| 축소된 구문 | 품사기반 축소규칙 | 관측 확률 |
|---|---|---|
| The PKlite compression | P(DT NNP NN \| DT NNP NN) | 0.396113602 |
| The compression | P(DT NN \| DT NNP NN) | 0.267414051 |
| The PKlite | P(DT NNP \| DT NNP NN) | 0.206950673 |
| PKlite compression | P(NNP NN \| DT NNP NN) | 0.077204783 |
| compression | P(NN \| DT NNP NN) | 0.052300000 |

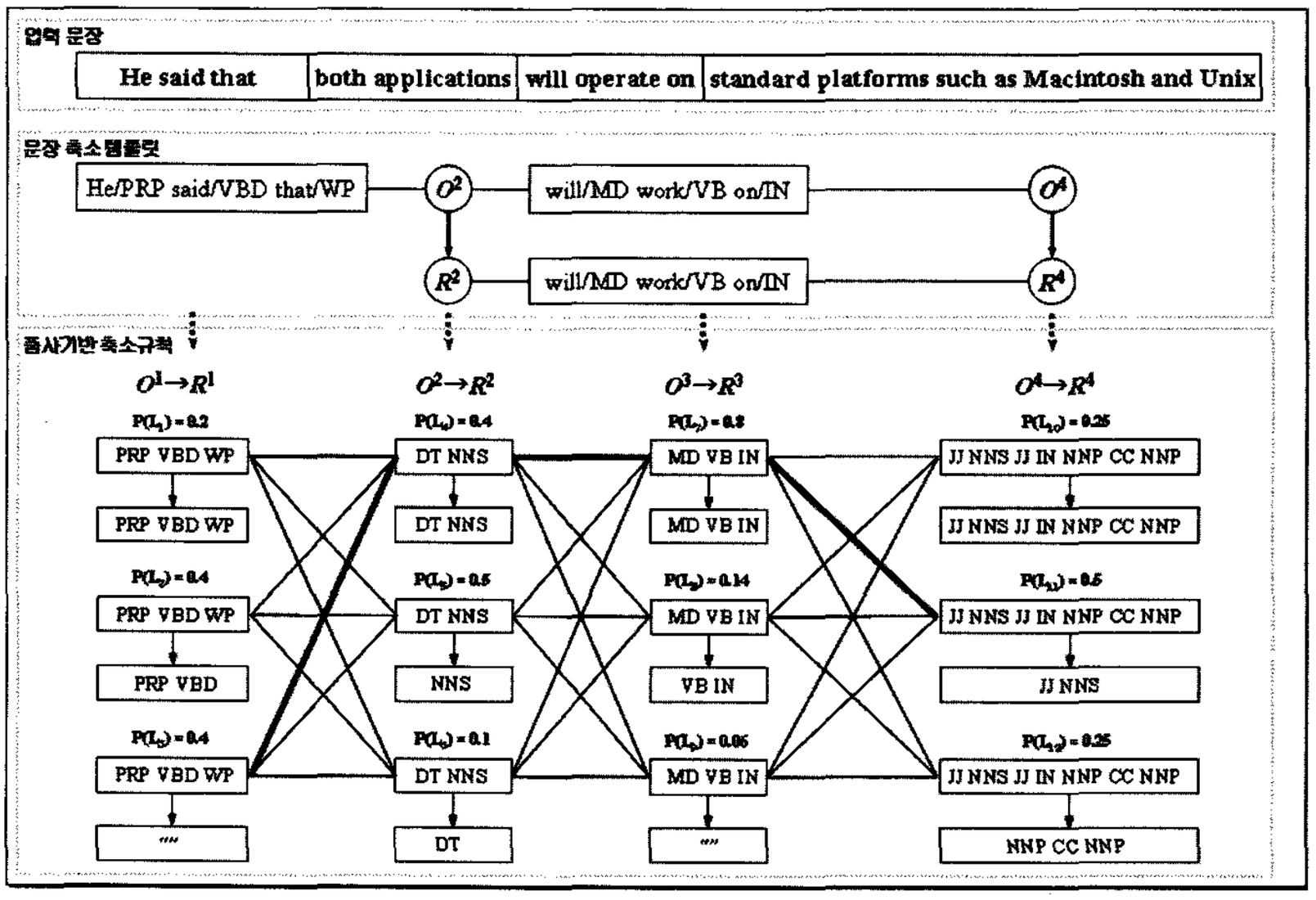


그림 10 품사기반 축소규칙을 이용한 HMM 기반의 문장 요약의 예

표 2 전이 확률의 예

| | $L_4$ | $L_5$ | $L_6$ |
|---|---|---|---|
| $L_1$ | **0.6** | 0.3 | 0.1 |
| $L_2$ | **0.7** | 0.2 | 0.1 |
| $L_3$ | **0.7** | 0.1 | 0.2 |

| | $L_7$ | $L_8$ | $L_9$ |
|---|---|---|---|
| $L_4$ | **0.7** | 0.2 | 0.1 |
| $L_5$ | **0.6** | 0.1 | 0.3 |
| $L_6$ | **0.7** | 0.3 | 0.0 |

| | $L_{10}$ | $L_{11}$ | $L_{12}$ |
|---|---|---|---|
| $L_7$ | 0.2 | **0.6** | 0.2 |
| $L_8$ | 0.4 | **0.5** | 0.1 |
| $L_9$ | 0.3 | **0.4** | 0.3 |

기반 축소규칙($L_1 \sim L_{12}$)을 발견한 후, 각 규칙이 나타날 확률 정보 및 각 규칙 간의 전이 확률 정보를 이용하여 최선의 품사기반 축소규칙 시퀀스를 발견하게 된다. 표 2는 그림 10에서 사용된 품사기반 축소규칙 간의 전이 확률 정보를 나타내고 있다. 이러한 전이 확률 정보와 관측 확률 정보는 HMM 모델의 비터비 알고리즘에 적용되어 품사기반 축소규칙 $L_3$, $L_4$, $L_7$, $L_{11}$을 선택하게 된다.

마지막으로, 선택된 품사기반 축소규칙으로부터 적절한 품사 결과를 추출한 후, 이를 원래의 어휘 정보로 대체함으로써 요약문을 생성하게 된다. 이러한 과정은 각 품사뭉치 간의 관계를 HMM 기반의 확률정보로 처리함으로써 요약문의 문법성을 추가적인 과정 없이 평가할 수 있으며, 더불어 기존의 어휘 규칙 기반의 방법보다 논리적이고 문법적으로 타당한 요약문장을 만든다.

5.2.3 품사기반 축소규칙 기반의 HMM 모델

품사기반 축소규칙은 어휘 규칙을 일반화시킨 형태의 축소 규칙으로써, 문법적으로 타당한 요약 문장을 생성하기 위해서 사용된다. 그러나 문장 축소 템플릿에서 나타난 품사뭉치와 각 뭉치에서 선택 가능한 품사기반 축소규칙이 많을 경우, 가장 적합한 품사기반 축소규칙의 시퀀스를 찾아내야 한다.

새로운 입력 문장 $e$에 대해서 품사 정보 $e_1, e_2, \cdots, e_m$와 품사기반 축소규칙 $L_1, L_2, \cdots, L_n$이 주어지면, 베이즈의 정리(Bayes' theorem)를 이용하여 다음과 같은 식 (5)을 얻을 수 있다.

$$P(L_1 L_2 \cdots L_n | e_1 e_2 \cdots e_m) = \frac{P(e_1 e_2 \cdots e_m | L_1 L_2 \cdots L_n)}{P(e_1 e_2 \cdots e_m)} \times P(L_1 L_2 \cdots L_n) \quad (5)$$

요약 대상 문장 $e(e_1, e_2, \cdots, e_m)$에 대한 확률 값 $P(e_1 e_2 \cdots e_m)$는 상수 값이 됨으로, 식 (5)는 식 (6)을 구하는 것으로 바꿀 수 있다.

$$P(e_1 e_2 \cdots e_m | L_1 L_2 \cdots L_n) \times P(L_1 L_2 \cdots L_n) \quad (6)$$

또한, 식 (6)의 $P(e_1 e_2 \cdots e_m | L_1 L_2 \cdots L_n)$은 Bigram 모델을 이용하여 식 (7)과 같은 확률 값을 구할 수 있다.

$$P(e_1 e_2 \cdots e_m | L_1 L_2 \cdots L_n) = \prod_{i=1}^{n} P(e_{j_i} \cdots e_{j_{i+l}} | L_i) \quad (7)$$

여기서, $j_i, j_{i+1}, \cdots, j_{i+l}$은 각 품사에 대한 시퀀스를 나타내며, $e_{j_i} \cdots e_{j_{i+l}}$은 품사기반 축소규칙 $L_i$의 부분과 일치한다. 즉, 식 (7)은 각 품사기반 축소규칙에 대한 관측 확률을 의미한다. 다음으로, $P(L_1 L_2 \cdots L_n)$은 식 (8)과 같이 나타낼 수 있으며, 이것은 각 품사기반 축소규칙 간의 전이 확률을 의미한다.

$$P(L_1 L_2 \cdots L_n) = \prod_{i=1}^{n-1} P(L_{i+1} | L_i) \quad (8)$$

마지막으로, 식 (7)과 (8)를 이용하여 식 (9)을 유도할 수 있다.

$$P(e_1 e_2 \cdots e_m | L_1 L_2 \cdots L_n) \times P(L_1 L_2 \cdots L_n) = \prod_{i=1}^{n} P(e_{j_i} \cdots e_{j_{i+l}} | L_i) \times \prod_{i=1}^{n-1} P(L_{i+1} | L_i) \quad (9)$$

본 논문에서는 식 (9)를 최대화시키는 품사기반 축소규칙의 시퀀스를 찾기 위해 HMM 모델의 비터비 알고리즘을 이용한다. 만일 문장 축소 템플릿이 $t$개의 품사뭉치를 가지고 있고, 각 품사뭉치가 $l$개의 품사기반 축소규칙과 대응된다면, 컴퓨팅 연산량은 $l^t$가 된다. 그러나 비터비 알고리즘을 통해서 연산할 경우, 연산량은 $l^2 t$가 되어 복잡한 처리과정 없이 효과적으로 성능을 향상시키며 정확하게 축소된 문장을 얻을 수 있다.

## 6. 실험 및 결과

### 6.1 실험 환경 및 방법

본 논문에서는 문장 축소 템플릿과 품사기반 축소규칙을 추출하기 위해서 Linguistic Data Consortium (LDC)에서 제공하는 Ziff-Davis 말뭉치를 사용하였다. 이 말뭉치는 컴퓨터 제품 판매를 위한 뉴스 기사를 다루며, 각 기사는 원문과 전문가에 의해 작성된 요약문을 함께 제공한다. 이 말뭉치로부터 문장 축소 템플릿을 생성하기 위한 프로그램을 구현하였으며, 총 1,360개의 원본 문장과 축소된 문장의 쌍을 획득하였다. 이 중에서 적합하지 않거나 오류가 발생한 문장을 제거함으로써 최종적으로 1,052개의 문장을 추출하였다. 학습 데이타는 800개의 문장 축소 예제를 사용하였으며, 나머지 예제로부터 무작위로 20개의 문장을 선택하여 실험을 수행하였다. 또한, 전처리 과정을 수행하기 위해 Stanford 대학(http://nlp.stanford.edu/)에서 제공하는 품사 태거를 이용하였으며, 문장 축소 템플릿 학습 방법을 수행함

으로써 1,316개의 문장 축소 템플릿과 3,480개의 품사기반 축소규칙을 획득하였다.

실험은 아래와 같이 4가지 방법을 통해서 얻은 축소된 문장을 이용하여 수행하였다.

- Baseline 방법: 가장 높은 word-bigram 점수를 가진 문장을 생성
- 템플릿 방법: 템플릿과 어휘 규칙을 이용하여 축소된 문장을 생성
- 제안한 방법: 본 논문에서 제안한 방법을 이용하여 축소된 문장을 생성
- 전문가: 기사로부터 전문가가 작성한 축소된 문장을 추출

본 논문의 실험에서는 각각의 축소 문장들을 비교하기 위해서 4명의 평가자들이 참여하였으며, 각 평가자에게 20개의 원본 문장에 대해서 4가지 방법으로 생성된 총 80개의 축소된 문장을 보여줌으로써 Knight와 같은 평가 방법을 수행하였다[3]. 본 논문에서는 문장의 중요 정보 유지도와 문장의 문법성을 평가하기 위해서 각 평가자들에게 두 가지 실험에 참여하도록 하였다. 우선 평가자들은 축소된 문장이 원본 문장에 포함된 중요 정보를 얼마나 유지하고 있는가에 대해 1부터 10까지 범위 내에서 평가하도록 하였으며, 다음으로 축소된 문장이 얼마나 문법적으로 타당한가에 대해 1부터 10까지 범위 내에서 평가하도록 하였다.

### 6.2 실험 결과

그림 11은 원문과 4가지 방법을 통해서 작성된 문장 축소 실험의 결과를 보여주고 있다. 왼쪽 부분은 각 원문과 매핑된 문장 축소 템플릿을 나타내며, 오른쪽 부분은 4가지 방법을 통해서 생성된 결과들을 보여주고 있다. 우선, 첫 번째 문장을 통해서 나타난 제안한 방법의 결과는 전문가의 요약문과 비교하여 압축률이 떨어지는 것을 볼 수 있다. 그러나 기존의 템플릿 방법과 동일한 축소된 문장을 얻었으며, Baseline 방법보다는 문법적으로 적합하고 원문을 이해하기 충분한 정보를 포함하고 있다. 두 번째 문장에서는 Baseline이나 기존의 템플릿 방법보다 문법적이고 압축률이 높은 것을 볼 수 있다. 더욱이 제안한 방법으로 생성된 문장은 전문가의 요약문과 같은 결과를 보여주고 있다. 기존의 템플릿을 이용한 방법은 제안한 방법보다 많은 정보를 제공하고 있지만, 이러한 결과는 잘된 요약이라기보다는 오히려 학습 구문에 포함되지 않은 단어뭉치로 구성되었기 때문에 요약되지 않은 결과로 볼 수 있다. Baseline의 결과는 문법적으로 적합하지 않은 결과를 보여주고 있다. 마지막으로, 세 번째 문장은 전문가의 요약문과 비슷한 압축과 정보력을 제공하지만, 서로 다른 의미의 결과를 보여주고 있다. 그러나 이것은 어떤 정보에 더 많은 중요도

| 문장 축소 템플릿 | 방 법 | 결 과 |
|---|---|---|
| <Original><br>( Intel will use ) $S_1$<br>( to schedule the multiple ) $S_2$<br>( , ) $S_3$ ( said )<br></ Original ><br><Reduction><br>( Intel will use ) $T_1$<br>( to schedule the multiple ) $T_2$<br></ Reduction > | 원본 문장 | Intel will use the Trace compiler to schedule the multiple instructions, Multiflow officials said. |
| | Baseline | The compiler. |
| | 템플릿 방법 | Intel will use the Trace compiler to schedule the multiple instructions. |
| | 제안한 방법 | Intel will use the Trace compiler to schedule the multiple instructions. |
| | 전문가 | The compiler will schedule the multiple instructions. |
| <Original><br>( He said that ) $S_1$<br>( will operate on ) $S_2$<br></ Original ><br><Reduction><br>$T_1$ ( will operate on ) $T_2$<br></ Reduction > | 원본 문장 | He said that both applications will operate on standard platforms such as Macintosh and Unix. |
| | Baseline | He said that applications operate on platforms such as Macintosh Unix. |
| | 템플릿 방법 | Both applications will operate on standard platforms such as Macintosh and Unix. |
| | 제안한 방법 | Both applications will operate on standard platforms. |
| | 전문가 | Both applications will operate on standard platforms. |
| <Original><br>( ArborScan is ) $S_1$<br>( in testing ) $S_2$<br>( but ) $S_3$<br></ Original ><br><Reduction><br>( ArborScan is ) $T_1$<br>( in testing ) $T_2$<br></ Reduction > | 원본 문장 | ArborScan is reliable and worked accurately in testing, but it produces very large DXF files. |
| | Baseline | ArborScan and worked in testing, but it very DXF files. |
| | 템플릿방법 | ArborScan is reliable and worked accurately in testing. |
| | 제안한 방법 | ArborScan is worked accurately in testing. |
| | 전문가 | ArborScan produces very large DXF files |

그림 11 문장 축소 실험에 대한 예제

를 두는지에 따라 달라질 수 있다.

표 5는 실험을 통하여 생성된 4가지 종류의 축소된 문장에 대해서 4명의 평가자들이 작성한 결과를 정리한 것이다. 각각의 평가자들은 "원본 문장과 비교하여 얼마만큼 중요한 정보를 유지하는가?"와 "생성된 문장이 문법적으로 적합한가?"에 대해서 평가하였으며, 생성된 각 문장의 길이를 비교하여 압축률을 계산하였다.

표 5의 첫 번째 열은 문장의 중요 정보 유지도에 대한 평가 결과이다. 우선, 제안한 방법의 결과는 다른 방법들과 비교하여 원본 문장의 중요 정보를 적절하게 유지하고 있음을 알 수 있으며, 더불어 전문가가 작성한 축소 문장과 유사한 결과를 보여주고 있다. 그러나 Baseline 방법을 통해서 나타난 결과는 다른 방법에 비해서 비교적 낮은 정보 유지도로 평가되었는데, 이것은 높은 압축률로 인해서 나타난 결과로 판단된다. 두 번째 열은 문법의 적합성에 대한 평가 결과이다. 본 논문에서 제안한 방법의 결과는 전문가의 방법과 비교하여 문법적으로 낮게 평가되었지만, Baseline 방법이나 템플릿과 어휘 규칙을 이용한 방법보다는 높게 평가되었다. 특히, Baseline 방법은 문법적 정보 없이 단어 간의 확률 정보만을 고려하기 때문에 문법적 측면에서 낮게 평가되었을 것이라고 판단된다. 마지막으로 표 5의 세 번째 열은 압축률을 나타내고 있으며, 압축률이 낮을수록 보다 짧게 축소된 문장이다. 압축률은 전문가에 의해서 생성된 문장이 가장 좋은 것으로 평가되었으며, 다음으로 Baseline방법, 제안한 방법, 그리고 템플릿을 이용한 방법 순으로 평가되었다.

이와 같은 실험 결과를 통해서 제안한 방법은 중요 정보 유지도 측면과 문법성 측면에서 기존의 문장 축소 방법인 Baseline 방법이나 템플릿 방법과 비교하여 우수한 성능을 나타낸다고 분석된다. 그러나 압축률 측면에서는 Baseline 방법이나 전문가의 방법보다는 낮은 결과를 나타내고 있다. 이것은 제안한 방법이 압축률보다는 문법적으로 적합하고 의미적으로 적절한 문장을 생성하는 것에 중점을 두었기 때문으로 분석된다.

본 논문에서는 평가자의 실험 결과에 대한 타당성을 입증하기 위해서 T-검정을 수행하였으며, 표 6은 이러한 결과를 나타내고 있다. 각각의 결과는 신뢰구간 99% 내에서 유의하지 않기 때문에 모든 평가자의 평가 결과와 각 평가자의 평가 결과는 차이가 없다고 분석할 수 있다. T-검정의 분석 결과에서 평가자 A는 문장의 문법적 관점에 중점을 두고 있으며, 평가자 B와 D는 정보의 유지 측면에 평가 관점을 두고 있다고 분석된다.

표 5 4가지 문장 축소 방법에 대한 평가 결과
(총 320개 실험 데이타의 평균/표준편차)

| | 중요 정보 유지 | 문법성 | 압축률 |
|---|---|---|---|
| Baseline | 6.61±2.67 | 7.53±2.58 | 65.5% |
| 템플릿 방법 | 7.39±2.35 | 8.35±2.23 | 71.4% |
| 제안한 방법 | 7.89±1.89 | 8.9±1.29 | 67.9% |
| 전문가 | 7.90±1.98 | 9.3±1.35 | 61.0% |

표 6 T-검정 결과
(유의 수준: $P < 0.01$)

| | 중요 정보 유지 | 문법성 |
|---|---|---|
| 평가자 A | 0.015247356 | 0.260119134 |
| 평가자 B | 0.355669019 | 0.089297267 |
| 평가자 C | 0.282165413 | 0.181685485 |
| 평가자 D | 0.441142894 | 0.014268745 |

## 7. 결론 및 향후 연구

본 논문에서는 구문 분석을 이용하지 않고 적절한 요약문을 생성하기 위해서 문장 축소 템플릿과 품사기반 축소규칙을 이용한 문장 축소 방법을 제안하였다. 문장 축소 템플릿은 기계 번역 분야에서 사용된 번역 템플릿 기반의 학습 알고리즘을 적용하였으며, 이를 이용하여 문장 내의 중요 정보를 유지하면서 불필요한 구나 절을 제거하고 요약문의 형태를 결정하였다. 또한, 문장 축소 템플릿을 생성하는 과정에서 정의되지 않은 문장 구성요소들을 문법적으로 적합하게 축소시키기 위해서 품사기반 축소규칙을 정의 및 활용하였으며, 그 결과 적절한 요약문을 얻을 수 있었다. 마지막으로 HMM의 비터비 알고리즘을 이용하여 문법 및 의미적으로 적합한 품사기반 축소규칙의 시퀀스를 효과적으로 발견할 수 있었다.

본 논문에서는 실험의 타당성을 증명하기 위해서 4명의 평가자들에게 서로 다른 방법으로 생성된 4개의 요약문을 평가하도록 하였다. 우선, 문장의 문법성 측면에서는 단어 간의 확률 정보를 이용한 방법이나 템플릿을 이용한 방법보다 문법적으로 적합하다고 분석되었으며, 중요 정보의 유지도 측면에서도 본 논문에서 제안한 방법이 기존의 방법과 비교하여 효과적으로 유지하고 있음을 확인하였다. 그러나 압축률 측면에서는 Baseline 방법이나 전문가에 의한 방법보다는 낮은 결과를 나타내고 있었다. 이러한 결과는 제안한 방법이 압축률보다는 문법적으로 적합하고 의미적으로 적절한 문장을 생성하는 것에 중점을 두었기 때문으로 분석되었으며, 이것은 T-검정을 통해서 타당성을 증명하였다.

마지막으로 본 논문의 방법을 활용한 문서 요약 시스템은 논리적으로 적합한 요약문을 생성함으로써 요약문서의 가독성과 응집성을 떨어뜨리지 않고 원본 문서의 유용한 정보를 손쉽게 전달할 수 있으며, 비정형화된 문서를 정형화시킴으로써 문서의 색인화 및 메타 정보화를 가능하게 하여 정보 검색이나 정보 추천 시 사용자

에게 적절한 정보를 제공해 줄 수 있다. 더불어 본 논문의 방법을 구현하기 위해서 수집된 대량의 학습 말뭉치는 오디오-스캐닝 서비스나 뉴스 요약 서비스와 같은 다양한 응용 분야에서도 유용하게 활용될 수 있다. 향후 연구로는 본 논문에서 제안한 방법을 여러 국가의 언어에 적용시키기 위한 연구가 필요하며, 또한 문서 요약의 성능을 향상시키기 위해서 문장 결합이나 문장 편집에 대한 연구 및 단락 단위의 정보를 압축하는 기술에 대한 연구가 필요하다.

## 참 고 문 헌


[1] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," *in Proceedings of ACM-SIGR*, pp. 68-73, 1995.

[2] H. Jing, "Using hidden markov modeling to decompose human-written summaries," *CL*, Vol.28, No.4, pp. 527-543, 2002.

[3] K. Knight and D. Marcu, "Summarization beyond sentence extraction: A probabilistic approach to sentence compression," *Artificial Intelligence*, Vol. 139, pp. 91-107, 2002.

[4] S. Riezler, T. H. King, R. Crouch and A. Zaenen, "Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar," *HCL-NAACL 2003*, pp. 197-204, 2003.

[5] J. M. Withbrock and O. V. Mittal, "Ultra-summarization: a statistical approach to generating highly condensed non-extractive summaries," *In Proceedings of the 22nddddddd International Conference on Research and Development in Information Retrieval (SIGR'99, Berkeley, CA). Poster session*, pp. 315-316, 1999.

[6] M. L. Nguyen, et al., "Probabilistic sentence reduction using support vector machines," *Proceedings of The 20th International Conference on Computational Linguistics*, pp. 23-27, 2004.

[7] M. Nagao, "Framework of a mechanical translation between Japanese and English by analogy principle," *Artif. Human Intell.*, pp. 173-180, North-Holland, Edinburgh, 1984.

[8] I. Cicekli and H. A. Guvenir, "Learning translation rules from a bilingual corpus," *In Proceedings of the Second International Conference on New Methods in Language Processing*, pp. 90-97, 1996.

[9] I. Cicekli and H. A. Guvenir, "Learning translation templates from bilingual translation examples," *Applied Intelligence*, Vol.15, pp. 57-76, 2001.

[10] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of IEEE*, Vol.77, No.2, February 1989.

[11] K. S. Han, D. H. Baek, and H. C. Rim, "Automatic text summarization using query expansion," *Proc. of the 27th Korean Information Science Society Spring Conference*, pp. 339-341, 2000.

[12] M. L. Nguyen, S. Horiguchi, A. Shimazu, and B.T. Ho, "Example-Based Sentence Reduction Using the Hidden Markov Model," *ACM Transactions on Asian Language Information Processing*, Vol.3, No.2, pp. 146-158, 2004.



이 승 수

2007년 연세대학교 컴퓨터과학과(석사) 현재 (주)삼성전자 DM연구소 연구원. 관심분야는 인공지능, 패턴인식

염 기 원

2007년 과학기술연합대학원대학교 HCI 및 로봇응용공학(박사). 현재 한국과학기술연구원 지능인터랙션 연구센터 연구원

박 지 형

1979년 서울대학교 기계설계학과(공학사) 1981년 서울대학교 기계설계학과(공학석사). 1993년 서울대학교 기계설계학과(공학박사). 1981년~현재 한국과학기술연구원 지능인터랙션연구센터 센터장. 2004년~현재 과학기술연합대학원대학교 HCI 및 로봇응용공학 교수. 관심분야는 Interactive Tabletop Computing, Cognitive Human-Robot Interaction, Reality Mining

조 성 배

정보과학회논문지 : 소프트웨어 및 응용 제 35 권 제 1 호 참조