한국어 사전의 압축 구현
Implementation of Compressing a Korean Lexicon

임 한 규 · 박 상 호
(한서대학교 전산정보학과)
Hankyu Lim · Sang-Ho Park
(Department of Computer and Information Science, Hanseo University)

요 약

한국어 처리의 기본이 되는 형태소 분석을 위한 사전의 효율적인 구성을 위해 각 표제어의 반복 음절수에 의한 방식으로 이를 압축하고 복원하는 알고리즘을 보였다. 사전의 크기에 있어서 25% 줄일 수 있었으며 표제어를 검색할 때 횟수를 36 % 줄일 수 있었다. 아울러 빠른 검색을 위한 이진 사전을 오프셋에 의해 구성하였다.

Ⅰ. 서 론

자연 언어 처리의 어플리케이션에 따라 다르지만, 기계 번역의 경우, 보통 원시 언어의 분석을 위한 형태소 분석 사전 및 알고리즘, 변환을 위한 사전 및 알고리즘, 목적어 생성을 위한 생성 사전 및 생성 알고리즘으로 구성된다고 할 수 있다. 모든 경우에 있어서 이들은 사전을 이용하고 있으며 사전의 구성과 이에 수록된 정보는 자연 언어 처리 시스템의 성능을 크게 좌우하고 있다. 본 논문에서는 한국어 처리의 가장 기본이 되는 형태소 분석을 위한 사전의 효율적인 구성을 위한 방법을 제안하고 이를 구현한다. 형태소 분석에 있어서 그 특성상 사전의 검색을 상당히 빈번하게 수행하게 되므로 사전의 검색 시간을 단축하고 크기를 최소화할 수 있어야 한다. 따라서 2진 사전의 형태를 구축하여 이를 비교 검토하며, 사전 표제어의 반복되는 음절을 숫자로 대체하여 이를 압축한다.

Ⅱ. 사 전

자연 언어 처리의 기본 분석 단계를 얘기할 때 흔히 형태소 분석, 구문 분석, 의미

분석 등의 과정을 거친다. 각각의 분석 과정에 있어서 필요로 하는 정보의 형태와 깊이가 달라지기 때문에 이들의 각 단계에 필요한 사전을 따로 따로 구성할 수도 있고 아니면 통합된 사전을 사용할 수도 있을 것이다. 가령 형태소 분석인 경우, 이는 한국어 어절을 의미를 가지는 최소의 단위인 형태소로 구분해 내는 것을 말한다. 최소의 의미 단위라는 것은 의미로 보아 더 쪼갤 수 없는 소리 연결체를 말한다. 형태소는 어휘 형태소와 문법 형태소로 구성되므로 사전도 이 형태를 따른다. 따라서 형태소 분석을 위한 사전은 통상 표제어와 품사 정보 등을 가지며, 용언의 경우 불규칙 활용 정보 등을 가지고 있다. 물론 이는 형태소 분석 프로그램의 알고리즘과도 밀접한 연관 관계를 가진다. 즉 알고리즘에서 처리하지 않고 사전의 정보를 이용하려고 한다면, 당연히 사전에서 이러한 정보를 제공할 수 있어야 하며, 알고리즘에 의한 규칙 등으로 처리가 가능하다면, 따로 사전에 수록할 필요는 없다. 한국어의 경우 시중의 국어 대사전은 대략 30만 여개의 어휘를 가지고 있다. 그러나 상당수의 어휘는 실생활에서 거의 사용되고 있지 않으며, 실제 형태소 분석을 위한 사전의 표제어 수는 5만에서 10만 여개 정도만으로도 충분한 실정이다. 문법 형태소인 경우 그 수는 2000개 정도이다. 이 정도의 크기는 불과 수십 K 바이트에 불과하기 때문에 보통 주기억장치에 적재한다고 해도 아무 문제가 발생하지 않는다. 그러므로 형태소 분석에서 사전을 이야기할 때 주로 어휘 형태소를 위주로 한다. 어휘 형태소 사전에 수록된 품사 정보는 아래의 표 1과 같다.

<표 1> 어휘 형태소의 품사정보

체언	명사, 의존명사, 인칭 대명사, 지시 대명사, 수사
용언	자동사, 타동사, 불완전 자동사, 불완전 타동사, 피동사, 사동사, 보조 동사, 불완전 보조동사, 형용사, 보조 형용사
기타	관형사, 부사, 감탄사

사전의 구성시 주로 고려할 대상으로는 사전의 검색 속도, 암호화 및 크기의 경량화 등을 들 수 있으나 모든 조건을 최적의 상태로 만족하기는 실로 어려우므로 사용하고자 하는 어플리케이션의 성격 등을 고려하여 우선 고려할 대상을 선정해야한다.

III. 어휘 형태소 사전의 압축

약 10만 단어의 어휘 형태소 사전은 포함하는 정보의 형태나 종류에 따라 그 크기가 다양하게 달라질 수 있다. 사전의 용도는 어플리케이션에 따라 달라지며, 구성 형태 또한 변하게 마련이다. 또한 형태소 분석 프로그램이 실행되는 환경에 따라 사전의 검색이나 크기 등의 최적화에 있어서도 많은 영향이 있게 마련이다. 본 형태소 분석 프로그램은 PC 환경을 목적으로 하여 구성되었으므로 우선 사전의 크기를 최소화해야 하는 것이 주요 임무 중의 하나이다. 우선 한국어 어휘 형태소 사전의 구성 형태를 보면 다음의 그림 1과 같다.

가 *품사*
가가 *품사*
가가대소 *품사*
가가례 *품사*
가가문전 *품사*
가가호호 *품사*
.....
.....

<그림 1> 형태소 분석용 어휘 사전

사전의 레코드의 처음을 차지하는 표제어의 한글 코드는 조합형으로 이루어져 있으며 각 음절의 첫 번째 비트는 한글의 경우 항상 1로 설정되어져 있으며 그 다음 5비트가 초성, 다음 5비트가 중성, 마지막 5비트가 종성으로 구성되어 있다. 사전의 압축은 표제어의 성질, 목적 또는 사용 영역의 종류에 따라 다소 구성의 형태를 달리하고 있다. 영어나 독어 등의 경우, 문자의 발생 빈도수(letter frequency)에 근거한 압축이나 엔트로피가 높은 문자열, 즉, -s, -es, -ed, -ing 등을 압축하는 방식 등이 있다. 그러나 한국어의 경우 음절수가 11,172자에 이르고 있고 음절간의 엔트로피가 영어와 같이 구분이 뚜렷하지 않다. 물론 일반적인 텍스트의 압축에 관한 연구도 많이 있지만 형태소 분석용 사전의 경우, 사전의 검색 회수가 많고 실시간에 처리해야 하는 경우가 빈번하므로 복원 또한 빠른 시간 내에 이루어져야 한다. 따라서 본 절에서는 사전의 정렬 순

서를 이용해 보려고 한다. 우선 사전의 정렬에서 보듯이 각 표제어는 같은 음절이 반복되어 나타나는 경우가 빈번하다. 따라서 반복되는 음절들을 더 짧게 대체할 수단이 있으면 그만큼 줄어들 수 있는 여지가 있게 된다. 어휘 형태소 사전의 압축을 위해 제안된 알고리즘은 그림 2와 같다.

```
void main()
{
while(if it is not the end of file, read one
record of the lexicon){
find the number of the repeated characters compared with previous record

(if the repeated characters exist){
write the repeated number and the remaining characters
} if not {
write the lemma itself
create the index with the first
character and the record number
}
reserve the current record for comparison
}
}
```

<그림 2> 어휘 형태소 사전의 압축 알고리즘

위의 알고리즘은 반복되는 수만큼의 문자를 숫자로 대치하는 것을 말한다. 한글의 경우 **MSB(Most Significant Bit)**는 항상 1에 설정되어 있으므로 **MSB**가 0이라는 것은 그 문자는 숫자라는 것을 뜻한다. 따라서 MSB에 의해 한글과 숫자를 구분할 수 있다. 이에 의해 만들어진 사전의 형태는 그림 3과 같다.

압축된 사전을 복원하기 위해서 인덱스 파일이 필요한데 이는 압축 과정에서 만들어진다. 인덱스 파일은 첫 음절이 숫자로 대치되지 않은 표제어와 그 표제어의 레코드 번호로 이루어져 있다. 인덱스 파일의 예는 다음의 그림 4와 같다.

그림 5는 원하는 표제어를 찾기 위한 알고리즘을 기술한다.

가령 '가가례'를 찾으려고 할 때 우선 인덱스 파일에서 '가'를 찾아서 '가'의 레코드 번호를 찾는다. 즉 1은 '가'라는 첫 음절이 처음 나오는 레코드의 위치를 가리킨다. 그 다

가 *품사*

1가 *품사*

2대소 *품사*

2례 *품사*

2문전 *품사*

2호호 *품사*

.....

<그림 3> 사전의 압축된 형태

가 *1*

각 *897*

간 *1067*

.....

.....

힝 *98656*

<그림 4> 인덱스 파일

음은 순차적으로 레코드를 읽어서 원하는 것과 비교를 해서 찾으면 된다.

이런 방식에 의한 사전의 크기는 품사 정보를 제외했을 때, 원래의 747KB에서 559KB로 줄며, 인덱스 파일의 크기 15 KB를 고려하면 574 KB로 줄어든다. 이는 원래 크기의 75 %이다. 사전의 검색을 위해서는 인덱스 파일을 이진 탐색에 의해 찾아가면 된다. 이의 검색 횟수는 O(log2 n)이며 인덱스 파일의 레코드 수가 1486이므로 10.54가 된다. 만약 100,000 레코드를 이진 탐색에 의해 찾아가면 16.61번의 검색 횟수가 필요하다. 따라서 6.07회의 검색 횟수가 줄어드는 효과를 낳는다.

void main(void)

{

scanf(input of word for search in the lexicon);

find word length;

while(end-of-file){

(if the first syllable of the word for search is in the index file){

locate in the lexicon file by index number;

(if the word length for search is one DBCS long) print the record and quit;

read one record of the lexicon;

while(if the first character is number){

get the characters whose length is number, from the previous word;

(if the key matches the lemma in the lexicon)

{print and quit;}

reserve the current data in the variable for the previous data;

read the next record;

}

quit;

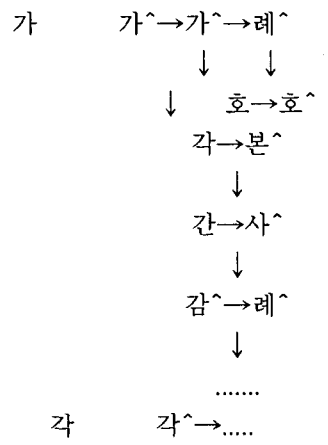
}

}

<그림 5> 복원 알고리즘

Ⅳ. 2진 사전

어휘 형태소 사전을 2진 트리로 구성하여 이를 살펴 본다.



※ ^ 표시 : 단어의 끝을 의미

<그림 6> 인덱스 내용

어휘 형태소사전을 그림 6의 형태대로 구성하였다. 2진 사전의 경우 크기는 750 K 바이트로서 크기에 있어서는 별 장점이 없다. 그러나 2진 탐색을 위한 2진 트리는 키 비교에 의한 탐색시 가장 적은 횟수의 비교만으로 검색할 수 있으므로 빠른 검색을 위할 때는 매우 효과적이라 할 수 있다.

<표 2> 인덱스 파일

코 드	인 덱 스
가	...
각	...
...	
...	....

<표 3> 인덱스 표

코 드	오 프 셋	단 어 끝
가	11(각)	end
가	4(각)	end
례	1(호)	end
호	0	
호	0	end
각	2(간)	
본	0	end
간	2(감)	
사	0	end
감	0	
례	0	end
....		
각		
....		

V. 결 론

사전 표제어의 반복되는 문자를 숫자로 대체하여 압축한 결과 이의 압축률은 1.33이었다. 아울러 사전의 검색 횟수에 있어서도 16.6에서 10.5로 줄여 많은 개선 효과를 가져올 수 있었다. 즉 사전의 경량화와 검색 회수의 감소로 인해 성능 향상에 절대적으로 기여할 수 있게 될 것으로 기대된다. 아울러 오프셋에 의한 2진 사전은 크기의 면에서는 효과가 없는 것으로 나타났으나, 검색 횟수가 많은 어플리케이션에는 빠른 검색 속도로 인해 적합하다고 할 수 있다. 앞으로의 연구로는 우선 사전에 수록된 표제어의 선정에 있어서, 각각의 발생 빈도의 가중치에 의한 선정을 함으로써 적정화된 사전의 구성을 일차적으로 할 수 있을 것으로 판단된다.

참 고 문 헌

[1] 남기심, 고영근, "표준 국어문법론", 탑출판사, 1994

[2] 강승식, "음절 정보와 복수어 단위 정보를 이용한 한국어 형태소 분석", 박사 학위 논문, 1993

[3] 이종연, 오상헌, "N-GRAM 한글 사전을 이용한 오인식 단어의 교정 알고리즘", 제 5회 한글 및 한국어 정보처리 학술대회 논문집, pp.271-283, 1993

[4] 심철민, 김민정, 이영식, 권혁철, "단어 간 지배 관계 및 연관 관계를 이용한 한국어 교열 시스템", 제 5회 한글 및 한국어 정보처리 학술대회 논문집, pp.303-316, 1993

[5] 장동수, 서영훈, "음절에 기반한 한국어 형태소 분석기", 제 5회 한글 및 한국어 정보처리 학술대회 논문집, pp.331-339, 1993

[6] 김덕봉, 최기선, "DDAG:효율적인 한국어 형태소 해석 방법", 제 5회 한글 및 한국어 정보처리 학술대회 논문집, pp.341-353, 1993

[7] 이병훈, 윤준태, 송만석, "말뭉치를 기반으로 한 한국어 철자 교정기의 구현", 제 5회 한글 및 한국어 정보처리 학술대회 논문집, pp.285-293, 1993

[8] 이재홍, 오상현, "한글 음절의 초성, 중성, 종성 단위의 발생 확률, 엔트로피 및 평균 상호 정보량", 대한전자공학회지, Vol. 26, No. 9, pp.1-9, 1989

[9] 변정용, "훈민정음 원리의 공학화에 기반한 한글 부호계의 발전 방향", 정보과학회지, 제 12권 제 8호, pp.72 - 88, 1994

[10] 박동순, "컴퓨터 한글코드의 사용과 표준화", 한국정보과학회지, 제6권, 제1호, 1988

[11] Hankyu Lim, Ungmo Kim, "Compressing the Korean Lexicon for a morphological analysis", Proceedings of the Fourteenth IASTED International Conference, pp.82-85, 1996

[12] Donald E. Knuth, "Sorting and Searching", Addison-Wesley Publishing, 1973