트라이와 구간트리를 이용한 사전기반 전문용어 인식 속도 향상

김형철[O], 김재훈, 최윤수
한국 해양 대학교 컴퓨터 공학과, 한국과학기술정보연구원
yhdosu@nate.com jhoon@hhu.ac.kr, armian@kisti.re.kr

Improving Speed for Dictionary-Based Term Recognition Using Trie and Interval Tree

Hyung-Chul Kim[O], Jae-Hoon Kim, Yun-Soo Choi
Korea Maritime University, Korea Institute of Science and Technology Information

요 약

전문용어는 특정 분야의 문서들에서 그 분야 특징을 반영하는 용어를 지칭하는 말로 최근 이러한 전문용어를 자동으로 인식하는 연구들이 활발하게 이루어지고 있다. 본 논문에서는 전문용어 인식의 방법 중 규칙 기반 방법의 한 종류인 사전 기반 방법을 이용하여 전문용어를 인식한다. 사전 기반 방법의 보통 다음과 같은 문제점이 있다. 첫째 같은 의미를 가지지만 형태가 다른 전문용어의 인식이 어려우며, 둘째 정확한 경계를 인식하기 위해서는 모든 단어에 대해 사전에 존재하는 가장 긴 단어의 크기만큼 매칭을 시도해야하며, 셋째 인식된 경계가 겹칠 수 있다는 문제점이 있다. 본 논문에서는 사전 매칭시 정규표현을 이용하여 첫 번째 문제를 해결하며, 트라이를 이용하여 사전을 구축하고, 매칭시 스택을 이용한 병렬구조를 사용하여 두 번째 문제를 해결하였으며, 구간트리라는 자료구조를 이용하여 세 번째 문제를 해결하였다.

주제어: 트라이, 구간트리, 사전 기반 전문용어 인식, 속도 향상

1. 서론

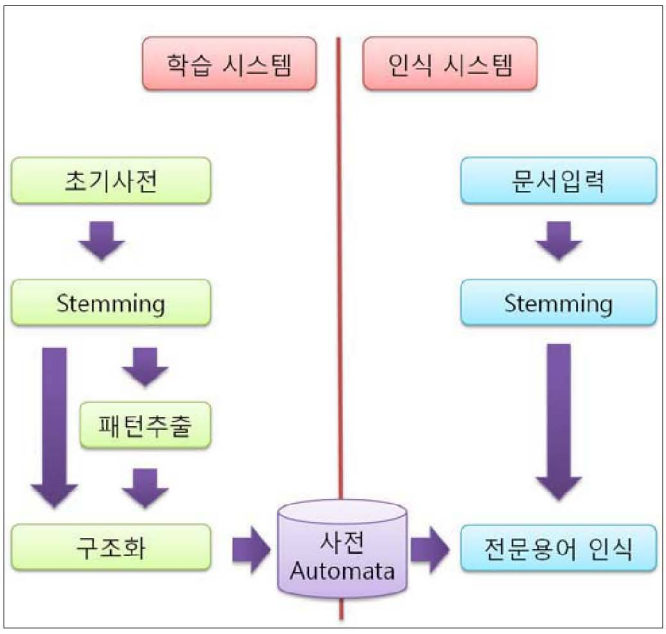
전문용어는 특정 분야의 문서들에서 그 분야의 특징을 반영하는 용어를 지칭하는 말로, 최근 이러한 전문용어를 자동으로 인식하는 연구들이 활발하게 이루어지고 있다. 이러한 연구의 경우 정보추출의 한 단계인 개체명인식과 비슷한 형태를 띠게 된다. 개체명 인식의 경우 그 역사가 길지만, 최근에 들어서는 정답이 존재하는 다량의 말뭉치를 이용한 기계학습 방법을 많이 사용한다. 하지만 전문 분야의 다양함과 각 분야별 전문용어를 태깅해 놓은 말뭉치 부족 등의 문제로 기계학습 방법을 이용하기가 힘이 든다. 때문에 초기에는 룰을 이용한 방법을 많이 이용하며 그 중에서도 가장 많이 이용되는 방법은 사전기반 전문용어 인식 방법이다. 하지만 사전 기반 방식의 경우 다음과 같은 문제점이 있다.

첫째 같은 의미를 가지지만 형태가 다른 전문용어의 인식이 어려우며, 둘째 정확한 경계를 인식 하기 위해서는 모든 단어에 대해 사전에 존재하는 가장 긴 단어의 크기만큼 매칭을 시도해야하며, 셋째 인식된 경계가 겹칠 수 있다는 문제점 등이 있다. 본 논문에서는 이러한 문제점들을 여러 가지 방법을 이용하여 해결하였으며, 그에 따른 성능 및 속도 향상을 꾀하였다.

2. 트라이와 구간트리를 이용한 사전기반 전문용어 인식 시스템

2.1 시스템 구성

전체 시스템은 크게 사전 모델(automata) 학습과 인식 시스템으로 이루어진다. (그림 1)에 전체 시스템의 논리적인 구성도를 나타내었다. 학습 시스템의 경우 초기 사전을 정제하여 패턴 추출 후 오토마타를 이용한 트라이 형태로 구조화한다. 인식 시스템은 입력된 문서를 정규화시키고 학습 시스템에서 만들어진 사전 오토마타를 이용하여 전문용어를 인식하고 출력하도록 구현된다. 2.2절과 2.3절에서 학습 시스템과 인식 시스템에 대해 각각 자세히 기술할 것이다.



(그림 1) 전체 시스템 구성

2.2 학습 시스템

학습 시스템의 경우 초기 사전을 정규화하는 작업을 가장 먼저 해야 한다. 사전 정규화 작업은 같은 의미를 가지지만 형태가 다른 전문용어들의 인식을 위해 진행된다. 이러한 형태 불일치는 크게 두 가지 형태로 나눌 수 있는데, 그 첫 번째는 수 불일치나 시제 불일치, 동명사 등 문법상의 형태 변이로 인한 것이고, 두 번째는 전문용어의 특성상 비슷한 형태를 띠는 것이다. 예를 들면 "P12"나 "P13" 모두 단백질을 나타내는 용어이지만, 두 용어 중 하나의 단어만 사전에 존재할 경우에도 나머지 비슷한 형태의 용어들을 인식할 수 있어야 한다는 것이다.

첫 번째 문제를 해결하기 위하여 전문용어에 맞는 스테머를 제작하여 사용하였다. 스테머는 기본적으로 포터-2 알고리즘에 기반을 두고 있으며, 포터-2 알고리즘[1] 마지막 처리부에 존재하는 접미사 리스트에 수집된 GT 관련 분야 사전의 접미사를 추출하여 추가하였다. 접미사 추출을 위해서는 "morfessor1.0"라는 오픈소스[2]를 이용하였다. 수집한 전문용어 사전에서 추출한 접미사에서 일반적인 단어들의 접미사를 제외하기 위하여 PenntreeBank의 단어들을 대상으로 접미사를 추출하여 중복되지 않은 접미사들만 최종적으로 선택하였으며, 그 중 빈도수가 높은 17개의 접미사를 추가하였다. 추가된 접미사는 <표 1>에 빈도수와 함께 표시 하였다.

<표 1> 추출된 접미사

접미사	빈도수	접미사	빈도수
aceae	171	ator	59
virus	130	osis	57
ine	127	mycin	57
ium	102	amide	57
itis	81	ates	56
ase	63	ceae	54
viridae	62	transferase	50
idae	61	ography	50
amine	61	-	-

두 번째 문제를 해결하기 위하여 정규 표현을 이용한 룰을 만들어 적용하였다. 전체 사전을 분석하여 몇 가지 룰을 만들었으며 그 규칙들은 <표 2>에 정리하였다.

<표 2> 사전 정규화를 위한 규칙

룰	구분	내 용
1	룰	숫자는 #으로 바꾸되 자리 수는 유지하도록 한다
	예제	P13 => P##
2	룰	이집트 숫자들도 #으로 바꾸되 자리 수는 유지하도록 한다
	예제	PⅣ => P#
3	룰	특수기호는 -으로 바꾸되 연속되는 특수기호는 하나로 통합한다
	예제	a$bc : a-b-c
4	룰	특별한 의미를 가지는 용어를 @로 바꾼다
	예제	GeanAlpha => Gena@

2.3 인식 시스템

인식 시스템은 입력된 문서에서 학습된 사전의 단어를 검색해 내는 시스템이다. 실제 학습 시스템은 사용자가 직접적으로 사용하는 시스템이 아니기 때문에 그 시간이 크게 문제가 되지 않지만, 인식 시스템의 경우는 다르다. 본 논문에서는 사전 기반 전문용어 인식 시스템의 속도 및 성능 향상을 위하여 본 논문이 제안하는 방법은 크게 세 가지 이다.

첫 번째 방법은 사전과 매칭할 때 정규표현을 이용하는 것이다. 위에 기술하였듯이 사전 구축 시에 미리 자주 발생하는 패턴에 대한 정규표현을 정의해 놓았으며, 실제 그 패턴을 이용하여 매칭하기 때문에 사전에 등록되어 있지 않지만, 패턴에 맞는 단어는 모두 매칭하게 된다.

두 번째 방법은 트라이를 이용하여 구축된 사전과 매칭할 때 스택을 이용하여 병렬로 동시에 여러 개의 단어를 매칭하도록 하는 것이다. 이 방법의 규칙은 간단하다. 공백이 발생되면 스택에 새로운 용어 인식 후보를 삽입하며, 입력 문서의 포인터가 한번 움직일 때 마다 스택 내부에 존재하는 후보를 검증해보고 사전에 없을 경우 삭제하고 사전에 있을 경우 출력하는 형태이다. 알고리즘을 <표 3>에 나타내었다.

1) http://snowball.tartarus.org/
2) http://www.cis.hut.fi/projects/morpho/

<표 3> 스택을 이용한 매칭 알고리즘

```
Dic := 사전 오토마타
Dic.CheckTerm(S) := 스택에 쌓인 단어가 용어인지 판별
Dic.next(S, P) := 스택에 P가 가르키는 글자를 추가했을 때 사전에 문제가 없는지 판별

Stack := 2중 스택
P := 입력 문서의 글자를 가르키는 포인터

P = 입력 문서의 첫 글자;
while(입력이 종료될때까지)
{
    if( *P == ' ' )
    {
        Stack.push(S);
    }
    foreach( S in Stack )
    {
        if( *P == ' ' ¦¦ Dic.CheckTerm(S) )
        {
            Print(S);
        }
        if ( Dic.next(S, P) == 1)
        {
            S.push(*P);
        }
        else { Stack.pop(S) }
    }
    if( Stack.Empty() ) P = 다음 공백으로;
    P++;
}
```

하지만 위 알고리즘을 통해 출력된 출력결과에서 용어들의 범위가 겹치는 문제가 발생 할 수도 있다. 이러한 문제를 해결하기 위해 세 번째 방법인 구간트리를 이용한다. 두 번째 단어와 세 번째 단어를 구간트리에 삽입하게 되면 범위가 겹치기 때문에 같은 노드의 자식으로 삽입 되게 되며 가장 길이가 긴 용어가 앞으로 오게 된다. 이 경우에 가장 긴 용어만을 선택하여 출력하게 된다.

3. 결론

본 논문에서는 사전기반 전문용어 인식 시스템의 문제점에 대해 논하였으며, 그 해결 방안을 제시 하였다.

스테밍을 이용한 사전 축소 및 사전 매칭시 정규표현을 이용함으로써 용어의 형태불이치 문제를 해결하였으며, 트라이를 이용하여 사전을 구축하고 매칭시 스택을 이용한 병렬구조를 사용하여 매칭 속도를 향상 시켰으며, 구간트리라는 자료구조를 이용하여 인식된 용어의 범위가 겹치는 문제를 해결하였다. 하지만 정규표현을 이용함으로써 매칭되지 말아야 할 용어들이 많이 매칭되는 등의 문제가 발생하였으며, 이러한 문제들을 해결하기 위한 다른 노력들이 필요할 것으로 보인다.

참고문헌

[1] Aoe, J. An Efficient Digital Search Algorithm by Using a Double-Array Structure. IEEE Transactions on Software Engineering. Vol. 15, pp. 1066-1077, 1989.

[2] M.F. Porter, An algorithm for suffix stripping, Program, 14(3) pp. 130–137, 1980.

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001.

[4] Ziqi Zhang, Jose Iria, C. B., and Ciravegna, F. A comparative evaluation of term recognition algorithms. In Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), 2008.