자연어검색시스템을 위한 스태밍알고리즘의 설계 및 구현

A Stemming Algorithm for a Korean Language Free-text Retrieval System

이효숙(Hyo-Sook Lee)*

목 차

초 록

본 연구에서는 자연어검색시스템을 위한 스태밍알고리즘을 설계하고 이를 구현하였다. 알고리즘은 순환적으로 다음과 같은 세가지 과정으로 진행된다. : 불용어사전에 의한 불용어의 제거 ; 규칙 테이블1의 적용에 따른 기본 어미의 처리 ; 전단계에서 처리되고 남은 어절에 대해 규칙테이블 2를 적용하여 확장스태밍 및 다시쓰기루틴으로 진행된다. 알고리즘의 성능 평가를 위해 한글문헌집단을 사용하여 테스트한 결과 압축률 21.4%, 오류율 15.9%의 결과를 나타내었다.

ABSTRACT

A stemming algorithm for the Korean language free-text retrieval system has been designed and implemented. The algorithm contains three major parts and it operates iteratively ; firstly, stop-words are removed with a use of a stopword list ; secondly, a basic removing procedure proceeds with a rule table 1, which contains the suffixes, the postpositional particles, and the optionally adopted symbols specifying an each stemming action ; thirdly, an extended stemming and rewriting procedures continue with a rule table 2, which are composed of the suffixes and the optionally combined symbols representing various actions depending upon the context-sensitive rules. A test was carried out to obtain an indication of how successful the algorithm was and to identify any minor changes in the algorithm for an enhanced one. As a result of it, 21.4 % compression is achieved and an error rate is 15.9%.

* The University of Sheffield (Ph.D. scheme)
■ 논문 접수일 : 1997년 11월 17일

1. 서 론

자연어검색시스템에서 색인 및 탐색시 주요 문제 중의 하나는 데이타베이스에서 동일한 의미를 전달하는 단어가 다양한 이형태(allomorphic form)를 갖는다는 점이다. 이에 대한 해결을 위해 다양한 알고리즘들이 개발되었다(Frakes and Baeza-Yates, 1992). 정보검색에서 단어의 형태적 변이체를 다루기 위한 알고리즘들은 크게 다음 두 분야에서 서로 독립적으로 연구되어 왔다. 분석대상 언어를 중심으로 언어의 형태론적 특성을 분석하고 이를 기초로 단어의 변이형태에 따른 문제를 해결하는 스태밍알고리즘의 연구와, 분석대상 언어의 특성과는 독립적으로 단어의 변이체들을 다루는 스트링유사도알고리즘의 연구이다.

자연어검색시스템에서 어간이 동일한 단어가 문헌에서 여러가지 다양한 표층형태로 사용된다면, 탐색시 검색효과를 높이기 위해서는 검색시스템에 이러한 문제를 해결할 수 있는 알고리즘이 필요하다. 본 연구의 목적은 데이타베이스에서 단어의 굴절현상에 따른 문제를 해결하여 텍스트 데이타베이스의 압축과 정보검색의 재현율을 증가시킬 수 있는 스태밍알고리즘을 설계하고 이를 구현하는데 있다.

본 연구를 위해 사용된 연구방법은 다음과 같다 : 첫째, 정보검색분야에서 단어의 이형태 처리를 위해 연구된 다양한 알고리즘의 특성 및 기법들을 고찰하였다 ; 둘째, 자연어검색시스템을 위한 스태머(a stemmer)의 개발을 위해 알고리즘 설계에 필요한 실험을 하였다 ; 셋째, 어미사전과 문맥의존규칙(context-sensitive rules)들을 개발하여 스태밍알고리즘을 설계하고, C언어로 이를 구현하였다. 넷째, 한글문헌집단에 대해 테스트하여 그 결과를 분석하였다.

본 논문에서 사용된 용어인 어미에 대한 정의로서, 스태밍대상이 되는 어미 영역에는 한글단어에서 체언의 곡용에 의해 접미사와 조사가 사용되는 경우와, 용언 활용에 의해 생성되는 어미를 모두 포함하는 것으로 국어국문학에서 의미하는 용언의 활용어미보다 광범위하게 사용되었음을 밝힌다.

2. 자연어 검색에서 단어이형태의 처리

2.1 스태밍알고리즘

정보검색시스템에서 형태적 변이체(word variants)를 갖고 출현하는 단어들은 색인어 매칭 알고리즘으로는 해결되지 않는다. 대부분의 경우 동일 어간의 변이체들은 같은 의미를 전달하는 것으로 표층형태가 다른 이들 단어들을 처리할 수 있는 다양한 스태밍알고리즘들이 개발되었다. 이 연구들에서 기초로 하는 가정은 한 단어에서 의미적으로 유용한 정보는 어간 또는 어근에 존재하고, 어미의 변화는 단지 문법적인 목적을 위해 변화된다는 점이다. 따라서 스태밍알고리즘은 어간의 의미를 유지하고 다양한 어미들이 제거된 단어들이 정보검색시 사용될 수 있도록 하는데 그 목적이 있다(Hull, 1996).

자연어검색시스템에서 스태밍알고리즘이 갖는 주요 기능은 시스템에서 서로 다른 형태

로 출현한 단어의 수를 줄여 의미적으로 관련된 모든 용어의 탐색이 가능하도록 함으로써 재현율을 증가시키고, 부가적으로는 사전의 크기나 데이타베이스의 갱신에서 효율적으로 운용되도록 한다(Walker and Jones, 1987 ; Lennon et al., 1981 ; Popovic and Willett, 1992).

2.1.1 알고리즘기법

스태밍알고리즘은 어간사전 및 어미사전의 사용, 알고리즘의 목적, 사용 규칙 등에 따라 다양한 특성들을 가진 스태머가 개발되고 있다. 스태밍알고리즘의 개발은 형태론적 분석에 기초한 방법과 어미사전 및 적용 규칙들의 개발을 통해서 어간일치를 위한 방법들이 사용되고 있다

스태밍알고리즘의 설계를 위해서는 한 언어에서 나타나는 다양한 굴절현상에 촛점을 두고 단어를 분석하여 단어의 표층형태로부터 사전에 출현하는 어휘형태로 압축시킨다. 이를 위해 단어의 내부구조가 분석되고, 단어로부터 제거할 어미부분이 조사되며, 서로 다른 형태소가 만나는 위치, 즉 단어에서 절단하여야 할 위치를 결정한다. 또한 단어로부터 어미가 단순히 제거되는 작업외에 어간의 원형으로 복구하기 위한 작업이 포함된다. 언어분석에 기반을 둔 스태밍알고리즘은 필요에 따라 시스템에서 사전을 참조하는 루틴이 포함된다. 현재까지 개발된 알고리즘들에서 가장 널리 적용되고 있는 방법은 규칙과 함께 다양한 크기의 어미사전이 활용되고 있다(Lennon et al., 1981).

스태밍알고리즘의 실행을 위해서는 주로 다음 두가지 방법이 사용되고 있다.

순환적방법(iterative technique)으로 이 방법은 단어에서 어미가 일정한 순서에 따라 생성된다는 사실에 기초한다. 단어에서 어미생성의 순서로서 파생어미가 먼저 부가되고 그 다음에 굴절어미가 부가되는 점을 이용하여 스태머에서 이러한 순서를 정의하고 이를 자동적으로 처리할 수 있도록 설계된다. 순환알고리즘에서는 분석대상단어의 끝에서부터 한번에 하나씩 해당 순서에 있는 어미가 삭제되고, 각 순서 클래스에서는 하나 이상의 매칭이 이루어지지 않는다. 알고리즘에서는 어미를 적용할 순서 클래스와 각각에 해당하는 어미들이 사용되며, 일반적으로 최장일치 알고리즘보다 적은 어휘의 어미사전이 사용될 수 있는 반면에, 어미사전의 설계와 어미제거 순서의 결정, 프로그램 루틴간의 연결 등에서 복잡성이 요구된다.

최장일치방법(the longest-match technique)은 분석대상 단어와 일치하는 어미들로 구성된 어미클래스들이 사용되며, 하나이상의 어미와 매칭되었을 때, 해당 클래스에서 가장 긴 어미가 후보가 되고 어미길이의 내림차순으로 매칭된다. 최장일치알고리즘은 하나의 순서클래스만을 포함하며, 조합가능한 모든 어미가 포함되므로 광범위한 어미사전이 사용된다. 단어와 어미와의 매칭순서는 어미길이의 내림차순으로 가장 긴 길이의 어미가 삭제시 우선순위를 갖는다.

스태밍알고리즘은 앞에서 기술된 기본 기법의 적용만으로는 만족스러운 결과를 내기 어려운데 그 이유는 과잉스태밍(overstem-

ming)과 과소스태밍(understemming)의 결과를 내기 때문이다. 알고리즘에서는 이를 해결하기 위한 방법으로 문맥의존규칙이 적용되고 있는데, 어미사전의 어미 적용시 다양한 조건들을 규칙화하여 이를 실행하며, 어미삭제 과정 후 어간부분에 대해 다양한 조건에 의한 다시쓰기규칙을 적용한 후 최종결과를 낸다 (Lennon et al., 1981 ; Porter, 1980).

2.1.2 주요 알고리즘

서로 다른 접근방법으로 개발된 대표적인 스태밍알고리즘들에 대해 다음에서 살펴본다.

(1) 로빈의 최장일치알고리즘

로빈(Lovins, 1968)의 스태밍알고리즘은 영어텍스트 검색을 위해 최장일치알고리즘을 구현한 것으로 이제까지 연구된 스태머 중에 가장 먼저 개발된 것으로, 이 분야의 연구를 위해 중요한 기틀을 마련한 것으로 평가되고 있다(Frakes and Baeza-Yates, 1992).

이 알고리즘은 크게 2단계로 실행되어 첫단계에서는 11개 클래스에 속한 약 260개의 어미들로 구성된 어미사전을 사용하여 최장일치 방법에 의해 스태밍이 되고 둘째단계에서 스태머는 어미생성으로 인해 철자변경이 일어난 어간들을 처리하는 작업이 실행되는데 규칙적용시 부분매칭으로 시작하여 후보어간들을 대상으로 완전한 조건에 맞는 철자집합이 최종 어간으로 출력된다. 스태머는 어미들과 각 어미와 관련된 어간들에 대한 정보를 주는 조건코드들로 구성된 어미사전을 사용하고, 어미삭제 후 정확한 어간이 생성되도록 하기 위해 다시쓰기 규칙과 부분매칭이 사용된다.

(2) 포터(Porter)의 순환알고리즘

포터의 알고리즘은 5단계로 이루어져 어미처리를 위해 순환 루틴을 사용하는 알고리즘으로, 각 단계에서 서로 다른 사전이 조회될 수 있도록 설계되었다. 약 60개의 어미들로 구성된 어미사전이 사용되며, 알고리즘에서 처리될 모든 단어는 [c](vc)^{m}[v]의 철자순서를 가진 형태로 표현되며, c와 v는 자음과 모음, m은 한단어내에서 모음과 자음 순의 출현이 m번 나타남을 나타내고 이것은 어미삭제시 알고리즘에서 판단될 조건으로 사용된다.

어미 삭제규칙은 다음과 같이 적용된다 : 만일 한 단어가 어미 s1으로 끝나고, s1 앞의 어간이 조건에 만족하면, s1은 s2로 대체되는데, 이 때 조건은 m의 출현 수를 척도로 사용하고 다양한 조건들은 합, 적, 차집합의 결과로 표현되어 사용된다. 예를 들어 m〉 1의 조건을 만족하는 단어에서 어미 - tional은 -tion으로 대체될 수 있다. 그리고 어미의 과잉스태밍이 되지 않도록 하기 위해 m이 척도로 사용된다.

(3) 다양한 언어환경에서의 알고리즘

이제까지 살펴본 알고리즘은 영어텍스트를 처리하기 위한 알고리즘인데 비해, 포포빅(Popovic and Willett, 1990)은 영어와는 아주 다른 특성을 가진 슬로바키아 언어의 처리를 위해 알고리즘을 개발하였다

슬로바키아 언어에서는 단어의 굴절시 어간과 어미모두가 형태론적 변이가 일어나기 때문에 다수의 변이체를 갖는다. 예로서 하나의 어근이 94개의 굴절이 일어나는 경우가 있는

것으로 보고하고 있다. 이러한 성질의 언어를 처리할 수 있기 위해서 영어텍스트의 경우보다 훨씬 많은 어미와 보다 복잡한 규칙들이 설계되었고, 알고리즘의 구성은 다음과 같다 : 어미사전은 최소어간길이 코드와 스태밍 처리 코드를 가진 어미들로 구성되었다. 어미사전의 어미와 입력단어와 매칭되면 이 두가지 코드 조건에 맞는 각 루틴에 따라 처리되며, 어미삭제 후 최종 결과는 20개의 규칙으로 구성된 어간사전과 대조된 후에 어간이 출력된다.

그 밖에도 불어텍스트를 처리하기 위해 사전과 품사정보를 활용하여 굴절어미와 파생어미를 처리하기 위한 알고리즘이 개발되었는데 (Savoy, 1993), 이 스태머는 언어처리를 위해 불어가 갖는 대표적인 특성으로 철자규칙에서의 불규칙성이 분석되어 어미처리시에 언어사전과 단어의 변이체 화일을 사용한다. 굴절어미와 파생어미 처리시 다음과 같은 두 단계의 순환알고리즘이 실행된다 : 첫단계에서는 굴절어미가 삭제되고, 둘째단계에서는 단어의 품사에 따른 구별된 어미사전을 사용하여 파생어미가 제거된다. 제거작업은 어미 제거의 대상이 되는 어간과 어간의 품사 조건을 조사하여 실행되고, 파생어미가 제거된 후 철자규칙에 따라 수정된 어간이 최종 출력된다. 그 밖에 라틴어 텍스트에 출현하는 다양한 어미들을 처리할 수 있는 스태머가 개발되었다 (Schinke et al., 1996). 이 스태머는 기존 알고리즘들과 달리 단어의 품사에 따라 2가지로 구별된 어미사전과 각각의 규칙을 사용한다. 즉, 명사와 형용사 단어처리를 위한 어미사전과 규칙, 동사처리를 위한 어미사전과 규칙이 적용된다. 알고리즘에서는 최장일치방법을 사용하고, 과소스태밍의 해결을 위해 최소어간길이규칙이 적용되었다.

2.2 스트링유사도알고리즘

정보검색시스템에서 데이타베이스 및 질의어 작성에서 종종 일어나는 오류 중의 하나는 철자 및 타이핑오류의 발생이다. 이러한 문제를 해결하기 위한 방법의 하나로 근사스트링매칭(an approximate string matching)이 적용되고 있으며, 근사스트링매칭에 포함되는 기법으로 n그램 매칭(n-gram matching), 음소분석에 의한 코딩(phonetic coding), 유사키(similarity key)방법이 사용되고 있다 (Kukich, 1992).

스트링유사도기법은 한 단어에서 문자가 출현한 구조는 단어의 의미와 관련이 있고, 이것은 단어 쌍 간의 유사도를 산출하는데 유용한 단서를 제공한다는 점에 기초한다. 순서화된 문자열로 구성된 두 개의 스트링 간의 매칭수준을 비교하기 위해 유사도가 사용되며, 각 스트링은 길이, 스트링내에서 각 문자의 출현패턴, 출현 수 등에 따라 유사도가 달라지게 된다.

스트링유사도기법에서는 비교하려는 한 쌍의 스트링간에 산출되는 유사도는 0에서 1사이의 값으로 측정되며, 두 개의 스트링이 완전히 일치할 때 1의 값이 된다. 그리고 스트링유사도 산출시에는 세 가지 종류의 유사도가 사용될 수 있는데, 한 쌍의 스트링에서 완전히 일치하는 문자 수의 수준을 측정한 실질유사도(material similarity), 문자가 두 스트링에서 동일한 순서로 출현한 정도를 측정한 순서

유사도(ordinal similarity), 스트링에서 문자가 동일한 위치에 출현한 정도를 측정한 위치유사도(positional similartiy) 등이다(Angell et al., 1983). 위치유사도는 철자수정 알고리즘 구현을 위해서는 지나치게 엄밀성을 요구하는 것이만 철자오류의 수정을 위해서는 가장 적절한 것으로 평가되며, 순서유사도에 의한 방법은 Phonix나 SPEEDCOP과 같은 다수의 음소값 코딩 알고리즘의 개발시 적용되며(Gadd, 1990), 실질유사도는 철자수정 알고리즘의 구현을 위해서 가장 널리 적용되고 있다.

2.3 알고리즘의 효과

단어의 이형태를 처리하기 위해 개발된 알고리즘(word conflation algorithms)에 대한 평가로서 레논 등(Lennon et al. 1981)은 5가지 알고리즘들에 대해 검색효과와 도치화일 압축률 면에서 평가하였다. 분석결과, 사전의 압축률과 검색효과 면에서 알고리즘간에 차이는 없는 반면, 데이타베이스에 출현한 다양한 변이체를 처리하기 위해서 상세한 수작업설계 과정을 거쳐 개발된 알고리즘과 완전히 자동적 절차에 의해서 개발된 알고리즘 간에 차이가 없는 것으로 보고하였다.

프론드와 윌렛(1982)은 질의어와 색인어의 두단어가 기준치 이상의 유사도를 가질 때, 탐색어 확장을 위해 n 그램 매칭을 적용하여 본 결과, 다이그램(digram) 매칭시 다수의 부적합문헌들이 검색되는데 비해, 트라이그램(trigram) 매칭시 재현율 0.88, 정확률 0.53으로 비교적 효과적인 것으로 보고하였고, 연구자들은 결론적으로 정보검색시스템에서 n 그램매칭 기법이 탐색어의 부분스트링으로 사용하기에 효과적임을 밝혔다.

워커와 존스(1987)는 온라인열람목록에서 순환적 스태밍알고리즘을 실험평가한 결과, 강도가 높은 스태밍(strong stemming)보다 완만한 스태밍(weak stemming)방법이 텍스트압축률은 다소 낮은 반면, 정확률이 감소되지 않고 재현율은 증가하는 것으로 발표하였다. 반면에 하만(1991)의 연구에서는 세가지 스태머 즉, 포터, 로빈, 복수어미통제 알고리즘을 적용한 결과, 순위화된 출력결과를 내는 정보검색시스템에서는 검색효과를 크게 향상시키지는 않음을 보고하였다.

로버슨과 윌렛(1992)은 스트링유사도기법에 속한 다양한 알고리즘을 평가한 결과, 음소코딩 기법의 적용이 음소코딩 기법을 사용하지 않은 경우에 비해 검색성능이 훨씬 향상되었음을 밝혔고, 가장 효과적인 방법은 다이나믹 프로그래밍기법으로 분석되었으나 처리시간이 많이 소요되는데 비해, 다이그램 매칭이 재현율과 처리시간면에서 비교적 효과가 있음을 밝혔다. 프레익스과 예이츠(1992)는 스태밍알고리즘에 대한 비평적 리뷰 논문에서 정보검색에서 스태밍의 효과를 다음과 같이 평가하고 있다 : 스태밍은 수동적인 어미절단만큼 효과가 있으며 스태밍에 의한 결과는 색인화일 크기에 상당한 영향을 준다. 또한 정보검색에서 스태밍의 효과는 알고리즘간의 차이는 거의 없는 반면 데이타베이스에서 사용된 어휘 특성에 영향을 받는다.

포포빅과 윌렛(1992)은 스태밍된 질의어, 탐색전문가에 의한 절단 질의어, 어미통제되

지 않은 질의어 사용에 의한 검색효과의 분석에서 스태밍된 질의어와 절단 질의어 사용이 어미통제되지 않은 질의어를 사용한 경우보다 검색효과가 월등히 향상됨을 보고하였다. 이것은 하만의 연구결과와는 다른 평가인 점에 주목될 필요가 있으며, 이들 연구자들이 제시한 다음과 같은 연구결과가 향후 연구의 과제로 남아 있다 : 정보검색에서 스태밍알고리즘의 효과는 처리되는 언어의 형태론적 복잡성 수준에 따라 성능향상에 차이가 있다.

헐(1996)은 서로 다른 다섯유형의 알고리즘, 즉 로빈, 포터, 복수어미통제, 굴절어미 스태밍, 파생어미 스태밍이 정보검색에서 미치는 효과를 분석하였다. 평가척도로, 질의어 길이, 검색문헌 수, 재현율을 사용하였다. 분석결과, 스태밍에 의한 정보검색은 이를 적용하지 않은 경우보다 검색성능이 향상되나, 사용된 질의어에 따라 향상된 효과간에 차이(1~3%)를 보이며, 단순한 복수어미 통제는 보다 강도높은 스태밍알고리즘을 적용한 경우 보다는 덜 효과적인 반면, 알고리즘 기법 간에는 차이가 없는 것으로 보고하여 선행연구와 일치된 결과를 보이고 있다.

이제까지 평가연구들에 관하여 살펴본 결과, 정보검색에서 스태밍알고리즘은 검색효과를 개선하며, 검색효과를 개선하는 수준은 사용된 질의어, 테스트된 문헌의 길이, 사용된 언어의 형태론적 복잡성 정도 등과 밀접한 관련이 있다고 요약될 수 있다.

본 연구에서는 한글문헌의 자연어검색에서 사용할 스태밍알고리즘의 설계를 위해 다음과 같은 실험이 실시되었다.

3. 한글문헌의 실험

3.1 출현단어 및 빈도

문헌집단에서 단어의 분포특성은 단어의 출현빈도와 분리능력과의 관계로서 설명되어 왔고(Salton, 1988), 이 때 빈도분포는 지프곡선 상의 세 영역, 즉 불용어 영역, 주제어로서 이상적인 중간빈도영역, 저빈도의 매우 특정한 주제어 영역 등으로 구분되었다(Ashford and Willett, 1988). 또한 룬(Luhn, 1958)이 단어의 중요도 측정시 문헌에서의 출현빈도 정보를 사용할 수 있음을 제시한 이래 단어의 빈도데이타는 이러한 목적으로 널리 사용되어 왔다.

지프와 룬의 견해는 본 연구에서 알고리즘 설계를 위해 문헌집단에서의 의미를 가진 어간들을 구별하기 위한 첫 단계로서 그 이론적 기초를 제공한다. 대부분의 자동색인 실험에서 고빈도의 주제의미를 갖지 않은 기능어를 제거하기 위해 불용어 사전을 사용하고 있다.

정보검색시스템에서 불용어사전의 사용은 불필요한 무의미어를 제거하는 여과의 기능을 가지므로 정보검색 성능에 영향을 주는 반면, 불용어사전에 포함될 용어의 선택을 신중하게 할 것이 요구된다(Fox, 1992).

본 연구를 위해서는 국내에서 개발된 두 테스트 컬렉션으로 부터 문헌집단을 구성하여 이들을 사용하였다. 테스트 컬렉션은 KT test set(김성혁 외, 1994)와 KRIST test set(이준호 외, 1995)으로, 이들로부터 각각 250건으로 구성된 문헌집단이 실험문헌으로 사용되었다. 사용된 문헌집단에서는 〈표 1〉에서와 같

〈표 1〉 KT set와 KRIST set에서의 단어 유형 및 토큰 수

문헌집단	단어토큰		단어유형	
	no	%	no	%
KT set	15,476	100	6,088	39
KRIST set	39,284	100	10,021	26

이 54, 760의 단어 토큰들이 출현하였다.

본 실험을 위해 문헌집단에서의 단어유형에 따른 빈도 특성을 조사한 결과, 상위 그룹에 속한 단어들은 주로 기능어들이었고, 실제 여기에 속한 단어들과 그 분포는 〈표 2〉, 〈표 3〉과 같다.

〈표 2〉는 KT set에서 가장 높은 출현빈도를 가진 단어들로서, 소수의 예외(예, 알고리즘을, 시스템의)를 제외하고는 모두 무의미어들임을 보여주며, 이 두 용어와 같이 주제분야에

〈표 2〉 KT set에서 출현한 고빈도 단어유형과 빈도분포

순위	단어	출현빈도	비율(%)
1	본	243	1.6
2	수	238	1.5
3	논문에서는	147	0.9
4	이	127	0.8
5	있는	119	0.8
6	있다	114	0.7
7	및	111	0.7
8	위한	93	0.6
9	위하여	73	0.5
10	이용하여	69	0.5
11	논문은	64	0.4
12	대한	61	0.4
13	방법을	59	0.4
14	데이타	58	0.4
14	기존의	58	0.4
16	또한	55	0.4
16	이를	55	0.4
18	알고리즘을	54	0.4
19	시스템의	52	0.3
20	위해	51	0.3

〈표 3〉 KRIST set에서 출현한 고빈도 단어유형과 빈도분포

순위	단어	출현빈도	비율(%)
1	및	734	1.9
2	연구는	711	1.8
3	본	258	0.7
4	대한	184	0.5
5	위한	150	0.4
5	있는	150	0.4
7	있다	145	0.4
8	수	139	0.4
9	연구개발의	124	0.3
9	그	124	0.3
11	개발	122	0.3
12	조건을	111	0.3
13	내용	90	0.2
14	연구	82	0.2
15	또한	79	0.2
15	연구의	79	0.2
17	국제	73	0.2
17	범위	73	0.2
17	위하여	73	0.2
20	의한	72	0.2

따라 불용어후보들은 항상 기능어만 포함하지는 않음을 나타낸다. 〈표 3〉은 KRIST set에서 높은 출현빈도를 보이는 단어들로서, 이들 단어들 중에는 동일 의미를 전달하는 단어들이 여러가지로 굴절된 형태로 출현하고 있음을 나타낸다. '연구는', '연구', '연구의' ; '위한', '위하여' ; '있다', '있는' 등이 그러한 대표적인 예이다.

한글문헌집단에서 출현한 단어유형에 지프법칙, 즉 단어사용빈도와 단어사용순위와의 곱은 상수에 가깝다는 원리를 적용하여 본 결과, 〈표 4〉와 〈표 5〉의 결과를 보였고, 이로써 실험문헌집단에서 한글단어의 빈도분포상의 특성은 다음과 같이 요약된다 : 첫째, 단지 1회의 출현빈도만을 갖는 단어들이 다수 출현하고 있다 ; 둘째, 다수의 단어들은 문헌집단에서 다른 단어와 동일한 출현빈도를 가진다 ; 셋째, 단어빈도와 사용순위와의 곱이 상수값을 갖는다는 사실을 일반화하기는 어렵다.

3.2 불용어

스태밍알고리즘의 설계와 관련하여 불용어사전에 포함될 수 있는 후보 불용어의 수집을 위해 다음과 같은 5가지의 정보원을 사용하였다.

〈표 4〉 KT set에서 단어출현에 대한 지프현상의 테스트

단어	순위	출현빈도	순위 * 출현빈도
이용하여	10	69	690
위해	20	51	1020
이러한	30	37	1110
따라	40	31	1240
이용한	50	28	1400
하는	60	25	1500
때	70	22	1540
따라서	80	20	1600
중요한	90	18	1620
필요한	100	17	1700
문제	200	10	2000
빠른	300	8	2400
영역의	400	6	2400
규모의	500	5	2500
트리를	1000	3	3000

〈표 5〉 KRIST set에서 단어출현에 대한 지프현상의 테스트

단어	순위	출현빈도	순위 * 출현빈도
그	10	124	1240
의한	20	72	1440
현재	30	50	1500
검토	40	40	1600
그리고	50	36	1800
방법을	60	31	1860
개발을	70	28	1960
개발하여	80	25	2000
제작	90	23	2070
실험을	100	21	2100
촉매	200	13	2600
연구와	300	9	2700
사업의	400	7	2800
특성은	500	6	3000
필요하고	1000	4	4000
합금성분의	2000	2	4000

· 한글 문장의 부속성분에 속한 단어그룹(남기심, 1995)
· 우리말 역순 사전에 수록된 해당 품사그룹에 포함된 단어그룹(유재원, 1987)
· ETLARS (Electronics & Telecommunications Literature Analysis and Retrieval System)의 불용어사전
· ELIS (EWHA Library Information System)의 불용어사전
· 실험문헌 집단(KT set, KRIST set)으로부터 추출된 후보 불용어

상기의 정보원으로부터 각 화일로 수집된 후보 불용어들에서 중복 단어를 제외하고, 곡용 및 활용어미를 갖는 단어들에 대해서는 각 용어에 포함되는 굴절형태들을 모두 작성하여 최종적인 불용어사전을 구성하였다. 후보 불용어리스트로부터 최종 불용어로 선택하기 위해서 다음과 같은 기준을 사용하였다 :

· 한글문장의 부속성분에 속하는 부사, 관형사로서 무의미어 : 예) 어느, 저
· 수사, 지정사, 보조동사, 대명사에 속한 단어 : 예) 하나, 없다, 아니하다, 그것이,
· 한글문장의 주성분에 속한 품사영역의 단어이나, 주제어로서 유용성이 낮은 단어 : 예) 어르신, 여쭙다, 저쑵다, 조렇다

반면에 문헌집단에서의 낮은 출현빈도를 보이고 주제어로서의 의미가 적은 단어들로서 다음과 같은 단어들이 후보 불용어로 고려될 수도 있으나 일반적 이용목적의 불용어사전을 위해 다음과 같은 단어들은 불용어사전에 포함되지 않았다 :

· 의미의 섬세한 변조나 표현방식에 따라 결정되는 단어유형 : 뽀얗다, 보얗다, 하얗다, 새하얗다, 길다, 기다랗다
· 사물이나 사람의 움직임 강세에 따른 다양한 표현의 단어유형 : 끌어안다, 얼싸안다

한글문헌의 실험 결과, 우리말 문장에는 우리말표기에 의한 다수의 한자어들이 사용되고 있음을 나타내었고, 이들 중에는 다음 예에서 보이듯이 다수의 동음이의어들이 포함되어 있다. : 적(enemy : noun), 적(non content-bearing : suffix) ; 성(sex : noun), 성(non content-bearing : suffix) : 때(time : noun), 때(dirt : noun) ; 안(inner side : noun), 안(knew : verb). 이와 같은 유형의 단어들은 불용어사전에 포함되지 않았다.

결과적으로 총 2,469개의 단어들로 구성된 최종 불용어사전이 구성되었고, 이 사전은 스태밍알고리즘에서 분석대상 단어그룹으로부터 불용어를 제외하는 루틴에서 사용된다.

3.3 어미의 출현특성

알고리즘에서 사용될 어미사전의 개발을 위해 다음과 같은 두 가지 과정이 포함되었다.

첫째, 실험문헌으로 사용한 한글문헌집단에서 출현한 어미의 유형 및 출현특성의 분석과, 둘째, 문법이론에 기초한 표준형태의 어미들을 광범위하게 수집하기 위해 조사 및 활용어미에 관하여 기술된 이론서 및 우리말 사전을 사용하였다.

3.3.1 문헌집단에서 어미의 출현특성

KT set와 KRIST set로부터 수집된 각 250건의 문헌에 출현한 모든 단어들 중에 영어 및

〈표 6〉 각 문헌집단에 출현한 총 어미 수와 어미유형

문헌집단	출현 어미 수		어미유형	
	no	%	no	%
KT set	4,613	100	451	9.8
KRIST set	7,478	100	467	6.2

수식을 제외한 한글단어들이 모두 추출되었고, 이에 대해 자동적 처리에 의해 단어를 역순의 형태로 작성한 후, 수집된 단어들에 나타난 어미의 특성들을 알고리즘 설계의 관점에서 분석하였다. 두 문헌집단에서 출현한 총 어미 수와 어미의 유형은 〈표 6〉에서 보이는 바와 같이 어미의 전체 출현 중에 약 10% 미만의 어미들이 문헌에서 주로 출현하고 있음을 나타내고 있다.

수집된 어미들 가운데 출현빈도와 이들의 형태적 특성을 분석한 결과, 최상위 출현빈도

〈표 7〉 문헌집단에서 높은 출현빈도를 보이는 어미

	KRIST set			KT set		
어미	순위	출현빈도	비율(%)	순위	출현빈도	비율(%)
-의	1	826	11.0	1	526	11.4
-을	2	529	7.1	2	382	8.3
-에	3	489	6.5	3	349	7.6
-를	4	406	5.4	4	277	6.0
-는	5	247	3.3	5	190	4.1
-이	7	216	2.9	6	143	3.1
-으로	10	164	2.2	8	89	1.9
-여	11	154	2.1	13	75	1.6
-가	13	135	1.8	7	95	2.1
-고	14	125	1.7	12	79	1.7
-에서	15	114	1.5	11	80	1.7
-와	16	111	1.5	9	87	1.9
-성	17	98	1.3	25	32	0.7
-ㄴ	19	82	1.1	15	64	1.4
-ㄹ	20	71	0.9	10	84	1.8
-적	21	69	0.9	16	57	1.2
-화	22	68	0.9	36	21	0.5
-도	23	67	0.9	32	25	0.5
-인	24	66	0.9	22	37	0.8

를 가진 어미의 타입과 분포 특성은 〈표 7〉과 같다. 출현 순위에서 상위에 있는 어미들과 낮은 순위의 어미들간의 출현빈도는 급격히 감소하여 큰 차이를 보인 반면, 두 문헌집단 모두에서 높은 출현빈도를 보인 어미들이 상위 수준을 차지하고 있고, 잘 알려진 어미들이었으며, 소수의 예외를 제외하고 자주 출현하는 어미들은 문헌집단간에 공통적인 특성을 보여준다.

테스트된 두 문헌집단에서 가장 높은 빈도를 보인 상위 40위까지의 어미들을 비교분석한 결과, 각 문헌집단에서만 출현한 어미들과 이들이 차지하는 빈도는 〈표 8〉, 〈표 9〉와 같다.

매우 낮은 출현빈도(1회)를 나타낸 어미를 분석한 결과 〈표 10〉과 같이 출현한 전체 어미유형들 중에서 35%에서 42%를 차지하고 있음을 나타내었다. 이것은 알고리즘 설계를 위해서는 사용할 어미사전의 구성시 잘 알려진 어미유형 외에 가능한 모든 어미유형들이 광범위하게 포함되어야 알고리즘이 효과적일 수 있음을 나타낸다. 출현한 어미의 평균길이를 조사한 결과, 〈표 11〉에서 보인 바와 같이 6개 음절 이상의 길이를 가진 어미는 출현하지 않아 한글문헌에서 어미의 길이는 다른 언어(Popovic & Willett, 1990)에 비해 비교적 짧은 것으로 나타났다. 반면에, 알고리즘의 구현시 반영되어야 할 중요한 특성으로 한글어미들은 형태론적 환경에 따라 다양한 불규칙 형태가 생성되어 어미의 축약, 모음조화에 따른 표층형태의 변화, 다양한 이형태 출현 등의 특성을 보여 정확한 스태밍이 되기 위해서는 단어로부터 어미를 분리하는 지점의 결정과 스태밍이후의 처리가 중요함을 시사하였다.

3.4 어미사전

실험문헌집단의 분석을 통해 나타난 한글문

〈표 8〉 KRIST set에서만 높은 출현빈도를 보이는 어미

순위	어미	출현빈도	비율(%)
6	-과	225	0.6
8	-로	188	0.5
9	-은	181	0.5
12	-ㄴ	153	0.4
18	-용	83	0.2
25	-어	60	0.2
28	-다	56	0.1
32	-는	42	0.1
33	-었으며	40	0.1
35	-게	37	0.1
39	-중	31	0.1

〈표 9〉 KT set에서만 높은 출현빈도를 보이는 어미

순위	어미	출현빈도	비율(%)
14	-었다	71	0.5
17	-적인	50	0.4
24	-ㄹ	33	0.2
26	-ㅁ으로써	28	0.2
27	-들의	27	0.2
29	- 에서의	25	0.2
33	-어	24	0.2
34	-들을	23	0.1
34	-었으며	23	0.1
36	-에서는	21	0.1
37	-들이	20	0.1
37	-게	20	0.1

〈표 10〉 각 문헌집단에서 1회 출현빈도를 가진 어미의 비율

문헌집단	출현한 어미유형	1회 출현 어미 수	비율(%)
KRIST set	467	161	35
KT set	451	188	42

〈표 11〉 문헌집단에서 출현한 어미의 음절 수

음절 수	KRIST set		KT set	
자음 또는 모음형태	19	0.2(%)	16	0.3(%)
1	4772	63.8	2512	54.5
2	1957	26.2	1416	30.7
3	621	8.3	535	11.6
4	103	1.4	126	2.7
5	6	0.1	8	0.2
6+	0	0	0	0
total	7478	100.0	4613	100.0

헌에서 어미의 출현특성은 어미사전 개발의 관점에서 다음과 같은 중요한 사실들을 나타내었다 : 첫째, 소수의 어미들이 한글문헌에서는 자주 출현하는 경향을 가지며, 그 가운데는

순수한 한글어미가 대부분이지만, 한자어로부터 유래된 조사들이 포함되어 있으므로 이와 같은 어미의 유형들이 어미사전에 포함되어야 한다 ; 둘째, 중간빈도를 나타내는 어미 가운데는 두 실험문헌에서 서로 다른 유형의 어미들이 사용되고 있는 것으로 비추어 한글문헌에서 일반적으로 사용될 수 있는 알고리즘을 위해서는 광범위한 어미사전이 구성되어야 한다. 셋째, 사용되는 어미의 길이는 비교적 짧고, 어미의 형태 중에는 음절 형태가 아닌 자음 또는 모음 문자형태인 것이 포함된다.

실험결과에 의한 이러한 어미의 특성들이 알고리즘에 반영되기 위해서는 보다 정확한 이론적인 뒷받침이 필요하였으므로 한글단어에 대해 형태론과 전산언어학적인 관점에서 연구된 관련자료(남기심, 1995 ; 유재원, 1987 ; 김승곤, 1989 ; 서태룡, 1988 ; 강승식, 1993)들을 분석하였다. 그 결과 체언의 곡용과 용언의 활용에 의해 생성되는 다양한 어미의 정확한 형태들을 수집하였고, 복합조사의 생성조건과 유형, 용언의 규칙 및 불규칙에 의한 형태변이 조건들이 반영되어 1508개의 어미들로 구성된 어미사전화일을 작성하였다.

알고리즘의 기법과 규칙사용여부를 결정하기 위해 한글어미가 갖는 다음과 같은 성질들이 고려되어 문맥의존규칙을 사용하는 순환알고리즘으로 설계되었다 : 한글단어는 그 구성에서 교착어로서의 속성을 나타내어 하나의 어간에 접미사, 조사 또는 하나이상의 어미들이 병렬적으로 부가되는 특성을 가지며 그 유형과 결합형태가 매우 다양하다 ; 어간에 어미 부가시 축약현상이나 모음조화 등이 일어나 서로 다른 형태소간 경계의 구별이 용이하지 않다 ; 용언은 규칙활용과 불규칙활용을 하며, 각 경우에 예외적인 적용이 필요한 경우들이 다수 있다.

4. 스태밍알고리즘의 설계 및 구현

4.1 규칙테이블

본 연구에서 개발된 스태밍알고리즘은 크게 세가지 단계로 구성되며, 최소어간길이 규칙과 다시쓰기 규칙이 사용된다. 스태밍과정은 순환알고리즘으로 진행된다. 알고리즘의 실행시 불용어사전, 규칙테이블1, 규칙테이블2가 사용되고 그 내용은 다음과 같다.

첫째, 불용어사전을 사용하여 불용어를 일차적으로 제거하는 루틴으로 불용어는 앞으로 진행될 스태밍 단계에서 고려되지 않는다.

둘째, 기본적인 어미제거 루틴으로 규칙테이블 1에서 제거할 어미가 매칭되면, 문맥의존 규칙으로 최소어간 길이 2음절이 만족하는 조건하에 매칭된 어미가 있는 해당 규칙을 적용한다. 규칙테이블1을 구성하는 각 규칙은 역순형태의 어미, 제거대상 음절 수, 규칙테이블 2의 활성화, 규칙 적용의 순환적 단계 또는 작업 종료 등을 나타낸다.

셋째, 확장스태밍 단계와 다시쓰기 단계를 실행하는 루틴으로 규칙테이블 2의 활성화는 어미 제거 후 남는 어간이 다음 조건 중 하나에 해당되는 경우로 모두 98가지의 경우들이 규칙 테이블에 포함되었다.

- 모음으로 끝나는 어간과 모음으로 시작하는 어미가 결합하여 축약된 경우

- 용언의 어간이 불규칙활용을 하는 경우
- 어간의 말음절이 어미의 첫음절과 동일한 경우
- 용언 어간의 말음절과 체언 어간의 말음절이 같은 경우
- 두개 이상의 서로 다른 단어가 동일한 어간의 말음절을 갖는 경우

규칙테이블 1과 2는 어미의 마지막 음절에 따라 정렬된 섹션들로 구성되고, 이것은 현재 입력된 단어나 또는 전 단계의 처리를 거친 어절의 마지막 음절과 일치하는 섹션의 탐색시 비교된다. 탐색은 해당 섹션의 첫음절에 신속히 접근하기 위해 시스템에서는 색인화일을 사용하여 해당 섹션에 직접접근한다. 그리고 각 섹션내에서 규칙의 순서는 알고리즘 실행에서 중요한데, 그것은 과잉스태밍(over-stemming)이나 과소스태밍(understemming)이 되지 않도록 각 섹션에서 우선적으로 적용되어야 할 어미의 순서대로 정렬되어 규칙의 적용은 이 순서대로 순차적으로 적용된다.

규칙테이블 1과 규칙테이블 2의 구성은 다음과 같다.

규칙테이블 1 : 각 규칙은 다음 4가지 요소들로 구성되었다.

- 체언의 곡용 또는 용언의 활용시 어간에 부가되는 어미로서 어미사전에 수록된 음절의 역순형태의 어미 ;
- 선택적으로 채택된 심볼 '>>'으로서, 앞에서 설명된 98가지의 경우들에 해당하는 어미의 처리를 위해 규칙 테이블 2의 활성화로 진행 ;
- 제거할 어미의 음절 수를 나타내는 0에서 9 까지의 숫자 ;
- 규칙의 순환적 적용을 나타내는 심볼 '>' ;

규칙테이블 2 : 각 규칙은 다음 5가지의 요소들로 구성되었고, 해당 규칙이 매칭되면, 규칙에서 지시한 조건대로 각 프로시쥬어에서 진행된다.

- 역순형태의 어미 또는 어미와 어간의 축약형태 ;
- 제거할 어미의 음절 수를 나타내는 0에서 9까지의 숫자 ;
- 어미 또는 축약형 어미의 첫음절 바로 앞 음절 수를 나타내는 심볼 '~'와 '!' ;
- 규칙테이블의 다른 규칙 적용의 계속 또는 종료를 지시하는 심볼 '>', '>>',와 '.' ;
- 어미제거 후 대체해야 할 하나 이상의 음절

4.2 알고리즘

알고리즘에서 규칙테이블 1은 〈표 12〉에서와 같이 1차적인 어미제거시 제거대상 어미, 제거음절수, 프로시쥬어의 순환적 진행, 규칙테이블 2의 활성화를 위해 사용된다.

입력단어의 마지막음절과 매칭된 해당 섹션('도')에서 해당규칙을 탐색하여 다음과 같이 적용하는 것은 알고리즘의 1차 단계에서 진행되는 주요 특성들을 보여준다 : 입력된 세 단어 '전동기라도', '온도', '난로도'에 대해 '전동기라도' 는 후보어미 '-라도' 중 어미 '-도'

〈표 12〉 규칙테이블 1의 '도' 섹션 예

```
.
.
.
도라이지4>까>>
도라이터부5>로>>
도라이테한5>
도라2>이>>
도라지까서5>에>>
도라지3>까>>
도라지3>을>>
도라2>지>>
  라터부서5>에>>
도1>라>>
도러2>더>>
도1>러>>
도럽2>처>>
도려2>으>>
도1>려>>
도로2>대>>
도로2>으>>
도1>로>>
.
.
.
.
도터부서게5>에>>
도터부서테한6>
도터부3>로>>
도터2>부>>
도테한3>
도>>
```

를 적용하여 삭제되고 규칙 2가 활성화된 후 확장스태밍 단계에서 조건이 조사된 후 '-라'가 적용되어 어간으로 '전동기'가 출력된다. 입력단어 '온도'는 어미제거 루틴에서 규칙이 적용될 수 있는지 조건을 탐색하는 단계에서, 최소어간길이 조건이 만족되지 않으므로 입력단어의 형태변화 없이 '온도'로 출력된다. 단어 '난로도'에 적용가능한 후보어미는 '-로도'와 '-도'이다. 그러나 후보어미 '-로도'의 적용은 최소어간길이 조건에 만족되지 않으므로 어미 '-도'가 채택된다. 규칙테이블 1에서 삭제대상 어미 '-도'의 적용은 1차 단계가 아닌 규칙테이블 2의 활성화를 지시하므로 확장스태밍 단계에서 조건의 만족 여부가 조사된 후 단어 '난로도'에 대해 어미 '-도'가 삭제되어 어간으로 '난로'가 출력된다.

알고리즘의 2차 단계인 확장스태밍 및 다시쓰기 루틴은 규칙테이블 2의 적용으로 진행되며 분석대상 단어는 1차 단계에서 단어의 일부음절이 제거된 형태이거나, 형태를 유지한 입력단어가 된다. 확장스태밍 및 다시쓰기 루틴에 포함되는 세부 단계들은 다음과 같다.

- 규칙테이블 2에서 단어의 마지막 음절과 일치하여 매칭된 섹션에 직접접근 한 후, 후보 규칙을 탐색하기 위해 섹션내에서 순차적으로 분석대상 단어의 음절과 규칙의 음절과 비교하며, 완전일치한 규칙이 탐색될 때까지 진행된다.
- 규칙이 탐색되면, 규칙에서 현재의 음절 다음위치의 문자가 체크된다.
- 다음위치의 문자가 숫자인 경우는 규칙에서 지시한 수와 동일한 크기의 음절이 분석 대상 단어로부터 제거되고, 심볼인 경우는 매칭된 현재위치의 음절 바로 전 위치의 조건을 조사하여 조건이 만족되는 경우에만 규칙에서 지시된 수만큼의 음절이 삭제된다.
- 해당 어미의 삭제 후, 규칙에서 대체할 음절이 있는지 체크하여 현재의 위치에 대체할 음절을 삽입한다.
- 음절이 대체되고 난 후, 현재의 규칙에서 대체음절 다음위치를 읽어 작업의 종료, 또는 계속 여부가 결정되며 스태밍 루틴은

다음 중 해당하는 경우로 진행된다 : 1차 단계로 다시 돌아가 다음 단계의 규칙이 적용되어 과정이 진행되는 경우 ; 규칙 테이블 2의 다른 섹션의 탐색을 계속하여 확장스태밍이 진행되는 경우 ; 현재 분석중인 단어에 대해 스태밍 루틴이 종료되어 어미의 제거 및 다시쓰기 과정을 거친 후 현재 남은 음절이 어간으로 출력되는 경우 등이다.

확장스태밍 및 다시쓰기 단계를 거쳐 처리된 결과의 예는 〈표 13〉과 같다.

4.3 알고리즘의 평가

알고리즘은 C언어를 사용하여 구현되었고, 실험문헌집단에서 수집된 단어들을 데이타화일로 입력하여 알고리즘의 성능을 분석하였다. 알고리즘의 평가 목적은 설계된 알고리즘의 성능을 정확히 측정하고, 실패요인을 분석하여 실제 한글 자연어검색시스템에 보다 강력한 기능의 스태머를 구현하기 위한 것이다. 알고리즘의 평가는 텍스트 압축률과 스태밍의 정확률을 평가기준으로 사용하였고, 분석결과는 선행연구결과들과 비교하였다.

4.3.1 텍스트 압축률

알고리즘의 성능을 평가하기 위해 KRIST set와 KT set의 문헌 40건을 대상으로 수식 및 알파벳 문자를 제외한 2,013개 어절이 수집되었고, 이들은 실제 1,329개 단어타입이 출현한 것으로서 분석되었다.

알고리즘의 첫단계로서 불용어사전과의 매칭에 의해 불용어처리를 한 결과 분석대상단어는 1,110개의 단어들로 압축되었다. 그리고 총 1,110개의 단어들은 실제 스태밍단계에서

〈표 13〉 확장 스태밍 및 다시쓰기 단계 후 출력결과

규칙테이블2		적용 예	
규칙	전단계에서의 처리	다시쓰기규칙 적용 전	규칙 적용 후
웁 1 우>	웁니다 > 웁니 > 웁	웁	울 (웁 > 우 > 울)
우가 2 갑.	반가우니 > 반가우	반가우	반갑
우고 2 곱.	고우면 > 고우	고우	곱
우구 2 굽.	구우니까 > 구우니 > 구우	구우	굽
우두 2 둡.	어두우나마 > 어두우나 > 어두우	어두우	어둡
우벼 2 볍.	가벼우니까 > 가벼우니 > 가벼우	가벼우	가법
우매 2 맵.	매우므로 > 매우	매우	맵
우추 2 춥.	추우려고 > 추우려 > 추우	추우	춥
우기 1 울.	기우므로 > 기우	기우	기울
우~ 1울.	우느라고 > 우느라 > 우느 > 우	우	울

기본 어미처리단계와 확장스태밍 단계를 거쳐 873개의 체언 또는 용언의 어간형태로 분석되었고 스태밍알고리즘의 압축률은 21.4%를 나타내었다. 이것은 선행연구(Popovic, 1990)에서 보고된 스태머의 압축률 54.7%이나, 영어문헌의 처리환경에서 평가(Lennon et al., 1981)된 알고리즘들의 평균 압축률 26.2%보다는 낮은 결과를 보인 반면, 포터(1980) 알고리즘에 의한 결과 14.8%보다는 다소 높은 압축률을 나타내었다. 따라서 본 연구에서 개발된 스태밍알고리즘은 어미처리 강도가 중간수준의 완만한 스태머의 특성을 갖는 것으로 평가된다.

4.3.2 스태밍의 정확률 및 오류분석

스태밍된 압축어절 873개어가 정확한 어간의 형태로 처리되었는지 분석하고, 오류결과를 낸 단어타입들을 밝히기 위해 스태밍 과정을 거친 최종 출력단어들을 사전(유재원, 1987)에 수록된 체언 및 용언의 어간형태와 비교분석을 하였다. 그 결과, 정확한 어간의 형태로 출력된 단어 수 734개어, 부정확한 처리결과를 보인 단어들은 139개어로서 스태밍의 정확률은 85.1%의 결과를 보였다, 이와 같은 결과는 〈표 14〉에서 보이는 바와 같이 포터의 알고리즘이나 포포빅의 스태머보다는 다소 낮은 정확률을 나타내었다. 포터알고리즘에서는 압축률 14.8%에 대해 스태밍의 정확율은 91.3%를 보이고, 슬라빅언어의 강력스태머에 의해서는 54.7%의 압축률과 90.2%의 정확률이 보고된 데 비해, 본 연구에서 구현한 한글 스태머는 21.4%의 압축률과 85.1%의 정확률을 나타내는 것으로 분석되었다.

분석대상 단어중 정확한 어미처리에 실패한 단어타입과 실패 발생조건을 조사하기 위해 처리실패어절들을 조사한 결과, 과잉스태밍 또는 과소스태밍 처리된 단어타입들은 〈표 15〉와 같다.

앞에서 분석된 내용을 조사하여 본 결과, 스태밍 오류는 주로 다음 두 가지 경우에 의한 것으로 밝혀졌다. 즉 최소어간길이 규칙으로 사용된 두음절 조건이 과소스태밍의 결과를 내었다. 또한 어간이 체언으로서 외래어인 경우, 어간의 말음과 어미의 첫음절이 동일한 경우에는 과잉스태밍으로 처리되었다. 따라서 현재 개발된 알고리즘에서 스태밍의 오류율을 낮추기 위해서는 현 알고리즘에서 처리되는 어미의 형태에 따라 단음절을 갖는 어간을 식별하여 최소어간길이 규칙을 적용할 수 있는 조건이 추가되어야 할 것과, 어미사전에서 사

〈표 14〉 알고리즘의 압축률 및 오류율

분석대상언어	처리단어 수	스태밍된 어간 수	압축률(%)	오류율(%)
영어	1,250	1,065	14.8	8.7
슬로바키아어	2,616	1,184	54.7	9.2
한국어	1,110	873	21.4	15.9

〈표 15〉 과잉스태밍 및 과소스태밍된 단어의 예

처리유형	해당 어절	처리결과	정확한 어간형태
과잉스태밍	다중프로세서에서는	다중프로	다중프로세서
	디메틸디염화실란으로부터	디메틸디염화실	디메틸디염화실란
	리엑터로서	리액	리액터
	교화그래프상의	교화그래	교화그래프
	파라미터들에	파라미	파라미터
과소스태밍	고쳐서라도	고쳐	고치
	넘겨서	넘겨	넘기
	걸친지	걸친	걸치
	끝냈다	끝냈	끝내
	보였지만	보였	보이

용빈도가 높은 외래어 어간의 말음이 고려되어야 할 것으로 밝혀졌다.

5. 결 론

본 연구에서는 자연어검색시스템에서 검색효과를 높이기 위한 한글스태밍알고리즘을 설계하여 이를 구현하였고, 한글문헌을 대상으로 알고리즘의 성능을 분석하였다. 알고리즘의 구현과 평가결과를 요약하면 다음과 같다.

첫째, 스태밍알고리즘은 한글문헌에 출현한 한글단어를 대상으로 불용어사전, 어미사전, 2가지의 규칙테이블을 사용하여 단어의 다양한 이형태를 처리할 목적하에 순환적으로 실행된다.

둘째, 알고리즘의 단계는 크게 불용어 제거단계, 기본 어미처리단계, 확장스태밍 및 다시쓰기 단계로 진행된다. 불용어사전에 의해 불용어를 제거한 후, 어미사전을 사용하여 1차적으로 어미를 제거한다.

셋째, 알고리즘에서는 문맥의존규칙으로 최소어간길이 규칙과 다시쓰기 규칙이 사용되며, 문맥의존규칙은 규칙테이블 1을 사용한 기본 어미처리단계와 규칙테이블 2를 사용한 확장스태밍 및 다시쓰기 단계에서 적용되며, 분석대상 단어에 대해 적용 규칙이 더 이상 탐색되지 않을 때 알고리즘의 순환과정이 종료된다.

넷째, 알고리즘을 텍스트 압축률과 스태밍의 정확률 면에서 평가한 결과 분석대상 1,329개의 단어타입에 대해 21.4%의 압축률과 85.1%의 정확률을 나타내었다. 선행연구들과의 비교결과 알고리즘의 정확률에서는 유사한 성능을 보인 반면, 스태밍 강도는 낮은 스태머인 것으로 평가되었다. 끝으로 오류분석결과의 내용이 반영된 알고리즘을 한글 자연어검색시스템에서 구현하여 그 효과를 분석하는 것은 후속의 연구과제임을 밝힌다.

참고문헌

강승식(1992). 음절정보와 복수어 단위 정보를 이용한 한국어 형태소 분석. 박사학위논문 : 서울대학교.

김석득(1994). 우리말형태론. 서울 : 탑출판사.

김성혁 외(1994). 자동색인기 성능시험을 위한 Test set 개발. 한국정보관리학회지. 11(1) : 81-102.

김승곤(1989). 우리말 토씨에 관한 연구. 서울 : 건국대학교 출판부.

남기심(1995) . 표준국어문법론. 서울 : 탑출판사.

서태룡(1988). 국어 활용어미의 형태와 의미. 서울 : 탑출판사.

유재원(1987). 우리말 역순 사전. 서울 : 정음사.

이준호 외(1995). 정보검색 연구를 위한 KRIST 테스트 컬렉션의 개발. 한국정보관리학회지. 12(2) : 225-232.

Ashford, J. and Willett, P. (1988) Text Retrieval and Document Databases. Bromly : Chartwell-Bratt.

Angell, R. C., Freund, G. E. and Willett, P. (1983) Automatic spelling correction using a trigram similarity measure. *Information Processing and Management*, 19 : 255-261.

Fox, C. (1992) Lexical analysis and stoplists. In : Frakes, W.B. & Baeza-Yates, R.(eds.) Information Retrieval : Data Structures & Algorithms. Englewood Cliffs : Prentice Hall.

Frakes, W. B. (1992) Stemming algorithms. In : Frakes, W. B. and Baeza-Yates, R. (eds.) Information Retrieval : Data Structures and Algorithms. Englewood Cliffs : Prentice-Hall.

Freund, G. E. and Willett, P. (1982) Online identification of word variants and arbitrary truncation searching using a string similarity measure. *Information Technology : Research and Development*, 1 : 177-187.

Gadd, T. N. (1990) PHONIX : the algorithm. *Program*, 24 : 363-366.

Harman, D. (1991) How effective is suffixing? *Journal of the American Society for Information Science*, 42 : 321-331.

Hull, D. A. (1996) Stemming algorithms : a case study for detailed evaluation. *Journal of the American Society for Information Science*, 47 : 70-84.

Kukich, K. (1992) Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24 : 377-439.

Lennon, M., Pierce, D. S., Tarry, B. D. and Willett, P.(1981) An evalua-

tion of some conflation algorithms for information retrieval. *Journal of Information Science*, 3 : 177-183.

Lovins, J. B. (1968) Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11 : 22-31.

Luhn, H.P. (1958) The automatic creation of literature abstracts. *IBM Journal of Research and Development*. 2 : 159-165.

Popovic, M. and Willett, P. (1990) Processing of documents and queries in a Slovene language free text retrieval system. *Literary and Linguistic Computing*, 5 : 182-190.

Popovic, M. and Willett, P. (1992) The effectiveness of stemming for natural language access to Slovene textual data. *Journal of the American Society for Information Science*, 43 : 384-390.

Porter, M. F. (1980) An algorithm for suffix stripping. *Program*, 14 : 130-137.

Salton, G. & Buckley, C. (1988) Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24 : 513-523.

Savoy, J. (1993) Stemming of French words-based on grammatical categories. *Journal of the American Society for Information Science*, 44 : 1-9.

Schinke, R., Greengrass, M., Robertson, A.M., and Willett, P. (1996) A stemming algorithm for Latin text databases. *Journal of Documentation*, 52 : 172-187.

Walker, S. and Jones, R. M. (1987) Improving Subject Retrieval in Online Catalogues. London : British Library.