정보관리연구, vol.42, no.4 2011, pp.137-154
http://dx.doi.org/10.1633/JIM.2011.42.4.137

문서 확장을 이용한 표제어 검색시스템

Headword Finding System Using Document Expansion

김 재 훈* · 김 형 철**
Jae-Hoon Kim · Hyung-Chul Kim

차 례

초 록

표제어 검색시스템은 뜻풀이를 질의로 간주하는 정보검색 시스템이다. 이러한 시스템을 구축하기 위한 가장 간단한 방법으로 사전의 표제어 뜻풀이(사전 뜻풀이)를 문서로 간주하는 정보검색 시스템을 구축하는 것이다. 이 문서의 길이가 너무 짧아 사용자 질의(사용자 뜻풀이)에 대한 적절한 표제어를 검색하기 어렵다. 이 문제를 완화하기 위해서 본 논문에서는 정보검색에서 사용되는 질의 확장 개념을 문서 확장에 적용한다. 본 논문에서는 문서 확장 방법으로는 뜻풀이 확장과 유의어 확장을 사용한다. 뜻풀이 확장은 주어진 단어의 사전 뜻풀이에 속하는 단어의 뜻풀이를 문서에 포함시키는 방법이고, 유의어 확장은 무자질 군집화 알고리즘을 통해서 유의어를 찾고, 찾아진 유의어를 문서에 포함시키는 방법이다. 제안된 표제어 검색시스템은 사전 뜻풀이 그 자체를 입력으로 할 때, 16-포함률이 거의 100%에 달하였다. 또한 사용자 뜻풀이를 입력으로 할 때, 20-포함률이 66.9%였다. 사용자 뜻풀이가 단어의 의미를 충분히 전달할 수 없는 것으로 관찰되었으며 앞으로 정확하고 객관적인 평가를 위해서 평가 집합에 대한 연구가 추가적으로 필요한 실정이다.

키 워 드

무자질 군집화, 표제어 검색, 문서 확장, 정보검색

* 한국해양대학교 IT공학부 교수
(Professor, Division of IT Engineering, Korea Maritime University, jhoon@hhu.ac.kr)
** 삼성전자 DMC 연구소 연구원
(Researcher, DMC R&D Center, Samsung Electronics Co. Ltd, yhdosu@nate.com)
• 논문접수일자: 2011년 6월 17일
• 최종심사(수정)일자: 2011년 7월 26일
• 게재확정일자: 2011년 8월 30일

ABSTRACT

A headword finding system is defined as an information retrieval system using a word gloss as a query. We use the gloss as a document in order to implement such a system. Generally the gloss is very short in length and then makes very difficult to find the most proper headword for a given query. To alleviate this problem, we expand the document using the concept of query expansion in information retrieval. In this paper, we use 2 document expansion methods : gloss expansion and similar word expansion. The former is the process of inserting glosses of words, which include in the document, into a seed document. The latter is also the process of inserting similar words into a seed document. We use a featureless clustering algorithm for getting the similar words. The performance (r-inclusion rate) amounts to almost 100% when the queries are word glosses and r is 16, and to 66.9% when the queries are written in person by users. Through several experiments, we have observed that the document expansions are very useful for the headword finding system. In the future, new measures including the r-inclusion rate of our proposed measure are required for performance evaluation of headword finding systems and new evaluation sets are also needed for objective assessment.

KEYWORDS

Featureless Clustering, Headword Finding, Document Expansion, Information Finding

1. 서 론

단어 검색(word retrieval or word finding)이라는 용어는 의학 분야에서 기억언어상실증(dysnomia)이나 실어증(aphasia) 등의 연구에 널리 사용되어 왔으며 생각의 표현하거나 물체의 이름을 말하는 데 필요한 단어를 찾아내는 과정으로 정의하고 있다(German 2000; Wise et al. 1991). 그러나 정보기술 분야에서는 다소 생소한 용어이나 최근 정보검색 등의 기술이 발전하면서 정보검색 분야에서도 그 필요성이 대두되고 있다. 예를 들어 '낱말 맞추기 놀이'(crosswords puzzle)를 생각해보자. 이 놀이는 단어에 대한 설명을 통해서 원하는 단어를 찾아 지정된 칸을 채우는 놀이이다. 이 밖에도 여러 가지 형태의 창작 활동에 매우 다양한 형식으로 사용될 수 있을 것이다. 본 논문에서는 이와 같이 단어의 뜻이 주어졌을 때, 그 뜻에 해당하는 사전 표제어를 찾는 시스템을 제안하고자 한다. 즉, "자기의 잘못을 인정하고 용서를 빎"이라는 질의어를 입력하면 '사과'라는 표제어를 찾아주는 시스템이다. 본 논문에서는 다른 영역에서 사용되는 단어 검색(word retrieval)이라는 용어와 혼란을 피하기 위해 표제어 검색(headword retrieval)이라고 한다.

단어의 뜻풀이를 문서(document)로 간주

하면 표제어 검색시스템(headword retrieval system)은 문서검색시스템(document retrieval system)이다. 이와 같은 문서검색시스템은 몇 가지 문제를 가지고 있다. 첫째, 뜻풀이(문서)의 길이가 너무 짧다(박은진, 김재훈, 옥철영 2005). 사전의 뜻풀이는 일반적으로 10어절을 넘지 않을 정도로 매우 짧게 기술되어 있다. 둘째, 뜻풀이는 매우 함축적이다. 예를 들면 '개'(dog)에 대한 뜻풀이는 "〈동물〉갯과의 포유류"이며, '포유류'는 젖을 먹이는 동물이라는 의미를 내포하고 있다. 셋째, 모든 문서가 서로 독립이 아니다. 예를 들면 '여가수'의 뜻풀이는 "여자 가수"이고 '가수'의 뜻풀이는 "노래 부르는 것이 직업인 사람"이며, '사람'의 뜻풀이는 "생각을 하고 언어를 사용하며, 도구를 만들어 쓰고, 사회를 이루어 사는 동물"이다. 이처럼 '여가수' 〈 '가수' 〈 '사람' 〈 '동물'과 같은 계층을 가지고 있어 어떤 단어의 완전한 의미를 파악하기 위해서는 이 계층의 의미를 충분히 파악되어야 한다. 이는 일반적인 문서검색시스템에서 문서 간에는 서로 독립이라는 가정을 그대로 적용할 수 없다. 이와 같은 뜻풀이 문서의 특성은 사용자(일반인)의 단어 뜻풀이와는 아주 다르다. 예를 들면 '개'(dog)라는 단어에 대해서 사전의 뜻풀이는 '갯과의 포유류'인데 일반 사용자는 거의 이와 같은 질의를 하지 않을 것이다. 본 논문에서는 이와 같은 문제를 완화하기 위해서 정보검색에서 사용되는 질의 확장 개념(query expansion)(Baeza-Yates and Ribeiro-Neto 1999)을 이용한다. 문서 확장 방법을 이용하면 위에서 제기한 첫 번째 문제인 문서 길이 문제를 해결할 수 있다.

본 논문에서 문서 확장 방법으로 뜻풀이 확장(gloss expansion)과 유의어 확장(synonym expansion)을 이용한다. 뜻풀이 확장은 단어의 뜻풀이에 속하는 단어의 뜻풀이를 다시 문서에 포함시키는 방법이며 이를 통해서 위에서 제기한 의미 함축성과 뜻풀이 의존성 문제를 다소 완화할 수 있을 것으로 생각한다. 유의어 확장은 무자질 군집화(featureless clustering) 알고리즘(Wong, Liu, and Bennamoun 2009)을 이용한다. 이와 같은 방법으로 확장된 문서와 특정 단어에 대해서 길게 풀어서 설명한 정의나 의미로 주어지는 사용자의 질의와의 유사도를 계산하여 표제어를 검색한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 간략히 기술하고, 3장에서는 시스템의 전체 구조와 문서 확장 방법을 자세히 설명하고 4장에서 제안된 표제어 검색시스템에 대한 성능을 평가하고 분석한다. 마지막으로 6장에서는 결론을 맺고 향후 연구 방향에 대해서 기술한다.

2. 관련 연구

2.1 질의 확장

질의 확장은 검색 모델에 따라 다소 차이가 있지만 일반적으로 사용자 질의에 검색 성능

을 개선하기 위해 질의 용어를 추가하는 방법이다(Baeza-Yates and Ribeiro-Neto 1999). 구체적인 방법으로는 시소러스와 같은 정보를 이용해서 확장하는 전역적 질의 확장 방법과 적합성 피드백과 같은 기술을 이용하는 지역적 질의 확장 방법이 있다. 전역적 질의 확장 방법은 초기 사용자 질의의 검색 결과와는 관계없이 시소러스와 같은 범용 지식을 이용해서 유의어나 상위어 등과 같은 용어를 질의에 포함시키는 방법이다. 본 논문에서는 유의어 확장이 이 방법에 속한다고 볼 수 있다. 지역적 질의 확장 방법은 초기 검색 결과를 이용해서 적합성 피드백을 이용하는 방법이 주로 사용되며, 본 논문에서는 뜻풀이 확장이 이 개념을 이용한 것이다.

2.2 무자질 군집화

군집화(clustering)는 유사한 성향이나 패턴을 가지는 자료를 한 곳으로 모으는 것이다(Jain, Murty and Flynn 1999). 군집화를 위해서는 먼저 각 자료들 간의 유사도를 계산하여 유사도가 가까운 자료들을 하나의 군집으로 간주하는 것이다. 군집화에는 그 대상에 따라 문서 군집화(Andrews and Fox 2007), 단어 군집화(박은진, 김재훈, 옥철영 2005; Hodgel and Austin 2002) 등 다양한 방법들이 있다. 문서의 경우 일반적으로 각 문서를 구성하고 있는 단어들이 비슷하다면 문서가 유사하다고 말할 수 있으며 이 단어를 자질(feature)라고 한다. 단어의 경우에는 문서의 경우와 많은 차이를 보인다. 왜냐 하면 단어 그 자체에는 문자 이외에는 특별한 자질이 없으며 문자 그 자체만으로 어떤 두 단어의 의미 유사도를 측정할 수 없다. 본 절에서는 단어와 같이 그 자체에서 자질을 찾을 수 없어서 어떤 형식으로 자질을 수집하여 그 유사도를 측정하는 군집화를 무자질 군집화(featureless clustering)라고 한다(Wong, Liu and Bennamoun 2007; Wong, Liu and Bennamoun 2009).

2.2.1 무자질 군집화를 위한 유사도 측정

단어의 무자질 군집화를 위한 유사도 측정 방법에는 여러 가지가 있지만 이 절에서는 NGD(Normalised Google Distance)(Cilibrasi and Vitanyi 2007)와 n°W(n° of Wikipedia)(Wong, Liu and Bennamoun 2007)에 대해서 간략히 기술한다.

NGD는 구글에서 검색되는 페이지의 수를 이용해서 두 단어 사이의 유사도를 계산하며, 식 (1)과 같이 정의된다(Cilibrasi and Vitanyi 2007).

$$NGD(x,y)=\frac{\max\{\log f(x),\log f(y)\}-\log f(x,y)}{\log N-\min\{\log f(x),\log f(y)\}} \quad \text{---------(1)}$$

여기서 x와 y는 단어이고, $f(x)$는 x를 포함하는 구글 페이지의 수이고, $f(x,y)$는 x와 y를 모두 포함하는 구글 페이지의 수이다. N은 x 혹은 y가 포함된 구글 페이지 수이다.

n^oW은 그래프 구조를 가진 위키피디아를 이용하는 방법이다. 위키피디아는 사전과 비슷하게 각 단어를 설명하는 하나의 문서가 있다. 이 문서는 개념 구조에 해단하는 "Categories"를 포함하고 있으며 이를 통해서 각 단어는 서로 연결되어 있다. 따라서 위키피디아는 마치 Categories를 통해서 서로 연결된 거대한 그래프(graph)와 같다. 유사한 문서일수록 같은 Categories에 있을 가능성이 높다. 이러한 점을 이용하여 두 단어 사이의 유사도를 계산하며 식 (2)와 같이 정의된다(Wong, Liu and Bennamoun 2007).

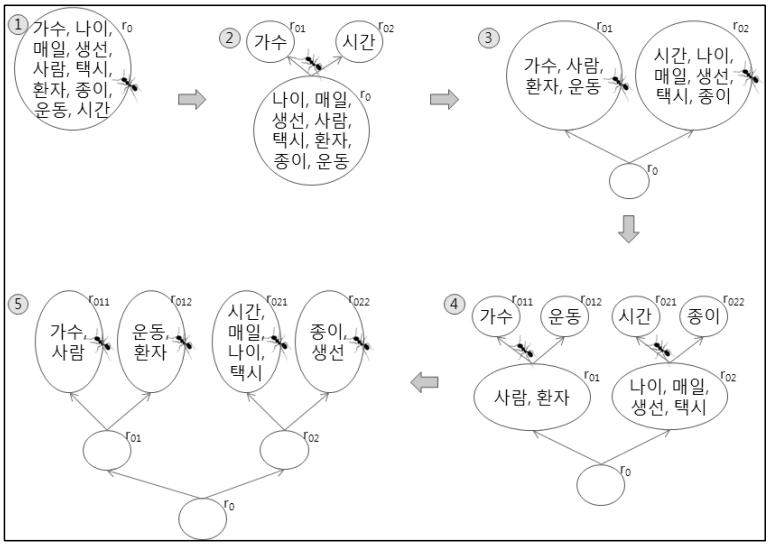
$$n^o W(d_x, d_y) = \sum_{k=1}^{|SP|} c_k \quad \text{----------(2)}$$

여기서 d_x는 단어 x를 설명하는 문서이고, d_y는 단어 y를 설명하는 문서이다. SP는 그래프 상에서 d_x와 d_y를 연결하는 최단경로이며 $|SP|$는 그 경로에 포함된 연결선(edge)의 수이고, c_k는 SP에 포함된 k번째 연결선의 가중치이다.

2.2.2 무자질 군집화 알고리즘: TTA(Tree-Traversing Ants)

본 논문에서는 무자질 군집화 알고리즘으로 3-Pass-TTA 알고리즘(Wong, Liu and Bennamoun 2007)을 수정하여 이용한다. 3-Pass-TTA 알고리즘은 TTA(Tree-Traversing Ants) 알고리즘(Handl, Knowles and Dorigo 2003)을 개선한 것이다. 먼저 TTA(Tree-Traversing Ants) 알고리즘에 대해서 살펴보자. TTA 알고리즘은 격자형태의 그래프 위에 군집하고자 하는 대상들을 임의로 올려놓고, 개미들이 지나다니며 유사한 것들끼리 모아가는 개념이다. 개미는 크게 세 종류의 일(pick-up, move, drop)을 수행한다. 〈그림 1〉은 TTA 알고리즘의 동작 원리를 보여주고 있다.

〈그림 1〉은 10개의 단어에 대한 군집화 과정으로 보이고 있으며, 여기서 사용된 유사도는 모두 임의로 가정된 것이다. 처음에 정점 r_0를 생성하고 군집 대상 단어 모두를 r_0에 할당하고 r_0에 개미 한 마리가 들어간다(①). 이 개미는 두 개의 정점 r_{01}과 r_{02}을 생성하고 r_0에 있는 단어들 중 유사도가 가장 낮은 두 단어를 선택하여 r_{01}과 r_{02}에 하나씩 옮긴다(②). 그림에서는 단어 '가수'와 '시간'의 유사도가 가장 낮은 것으로 가정하여 각각 r_{01}과 r_{02}으로 옮겨졌다. 그리고 나머지 남은 단어들을 '가수'와 '시간' 중 유사도가 높은 쪽으로 옮긴다(③). 모든 단어가 옮겨지면 각 정점(r_{01}과 r_{02})에 할당된 단어들의 유사도를 다시 계산해서 그 유사도가 사용자가 정한 일정한 수준보다 높게 되면 더 이상 노드를 분할하지 않는다. 그렇지 않으면 또 다시 개미를 그 정점에 보내어 분할하게 된다. 이 과정을 r_{01}과 r_{02}에 재귀적으로 적용한다. ④~⑤는 r_{01}과 r_{02} 모두 사용자가 정한 일정 값보다 유사도가 낮았으므로 다시 분할하는 모습을 보여준다. 자세한 TTA 알고리즘은 Handl, Knowles and Dorigo(2003)을 참조하기 바란다.

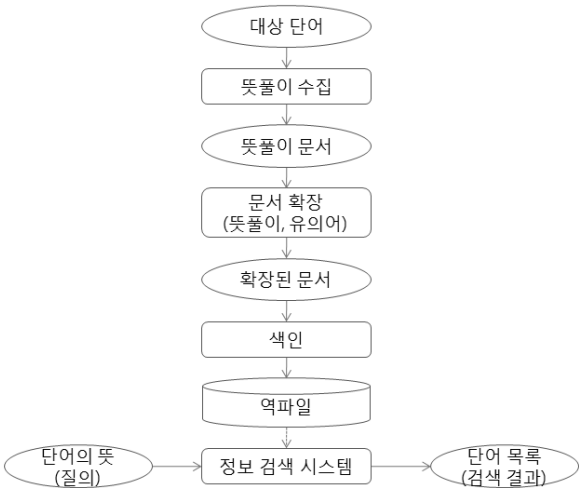


〈그림 1〉 예제(단어 군집)를 통한 TTA 알고리즘의 동작 원리

3. 문서 확장을 이용한 표제어 검색시스템

본 논문에서 제안된 표제어 검색시스템은 기본적으로 정보검색 시스템을 기초하고 있다. 문서는 일반 사전에 있는 단어 뜻풀이이며, 질의는 사용자 단어 뜻풀이이다. 앞에서 언급했듯이 사전 뜻풀이는 그 길이가 너무 짧고 함축적이어서 그대로 사용할 수 없다. 본 논문에서 질의 확장 개념을 문서에 적용하여 이 문제를 완화시키고자 한다. 본 논문에서 사용하는 문서 확장 방법으로 사전을 이용한 뜻풀이 확장(gloss expansion)과 단어 군집화를 이용한 유의어 확장(synonym expansion)을 사용한다. 뜻풀이 확장은 단어의 뜻풀이에 속하는 단어의 뜻풀이를 다시 문서에 포함시키는 방법이며 이를 통해서 앞에서 언급한 의미 함축성과 뜻풀이 의존성 문제를 다소 완화할 수 있을 것으로 판단한다. 유의어 확장은 무자질 군집화(featureless clustering) 알고리즘(Wong, Liu, and Bennamoun 2009)을 이용한다. 이와 같은 방법으로 확장된 문서와 특정 단어에 대해서 길게 풀어서 설명한 정의나 의미로 주어지는 사용자 질의와의 유사도를 계산하여 표제어를 검색한다. 〈그림 2〉는 전체 시스템 구조이다. 〈그림 2〉에서 대상 단어는 Naver와 Daum의 사전에 포함된 519,109개 단어를 선정하여 이들 단어의 뜻풀이를 기본 문서로 간주한다. 기본 문서는 여러 가지 문서 확장 방법을 통해서 문서를 확장하여 역파일(inverted file)을 구축하여 정보검색 시스템을 구축한다.

본 논문에서 정보검색 모델은 벡터 모델



〈그림 2〉 문서 확장을 통한 표제어 검색시스템의 구성도

(vector model)을 사용하며, 질의-문서 간 유사도(query-document similarity)는 가장 일반적으로 사용하는 코사인 유사도(cosine similarity)를 사용한다. 가중치는 TF-IDF 모형을 사용하며 TF(term frequency)는 일반적인 정보검색과 달리 경험적인 방법으로 결정되며 IDF(inverse document frequency)는 일반적인 정보검색에 사용하는 방법으로 그대로 사용한다. 색인 방법은 형태소 분석기를 이용하며 이 형태소 분석기는 세종 말뭉치(국립국어원 2007)와 CRF(Conditional Random Field)(CRFPP 2011)를 이용해서 구현되었다. 이 형태소 분석기는 동사의 복원이나 준말의 복원을 실행하지 않으며 실험 말뭉치에 대해서 98.1%의 성능을 보였다.

3.1 뜻풀이 수집

앞에서 언급했듯이 표제어 검색시스템의 기본 문서(basic document)는 사전 뜻풀이이다. 대상 단어의 뜻풀이는 인터넷에 공개된 사전(Naver와 Daum)을 이용하여 수집한다. 본 논문에서는 Naver와 Daum의 사전 openAPI를 이용하여 대상 단어의 뜻풀이를 수집하였다.

3.2 뜻풀이 문서 확장

이 절에서는 3.1절에서 수집된 기본 문서(뜻풀이)를 재귀적으로 확장하는 방법에 대해서 기술한다. 재귀적 확장은 3회로 제한한다. 예를 들면 대상 표제어 '사과'의 기본 문서는 "자기의 잘못을 인정하고 용서를 빎"이며 형태소 분석을 통해서 문서를 확장한다. 여기서 확장된 표제어와 가중치는 {(자기, 0.7), (잘못, 0.7), (인정하다, 0.6), (용서, 0.7), (빌다, 0.6)}이며 이를 1차 확장이라고 한다. 가중치 부여 방법은 〈표 1〉과 같다. 확장 차수가 높을수록 원문

서와는 거리가 멀어지므로 가중치를 낮게 설정하였고 동사보다는 명사가 정보검색에서 더 중요한 역할을 하므로(강현규, 박세영 1998) 높은 가중치를 부여하였다. 그러나 이 가중치들은 경험적으로 주어진 것이며 최적화를 위해서는 더 많은 연구가 필요한 실정이다.

〈표 1〉 색인어의 가중치 부여 방법

확장	1차	2차	3차
명사	0.7	0.5	0.3
동사	0.6	0.4	0.2

2차 확장은 1차 확장으로 기본 문서에 포함된 각 단어의 뜻풀이를 기본 문서에 다시 추가한다. 예를 들면 '자기'의 뜻풀이 문서는 "그 사람 자신"이므로 기본 문서에 {'사람', '자신'}이 추가되고, '잘못'의 뜻풀이 문서가 "잘하지 못하여 그릇되게 한 일"이므로 기본 문서에 {'잘하다', '못하다', '그릇되다', '하다', '일'}이 다시 추가된다. 이와 같은 과정으로 '인정하다', '용서', '빌다'에 대해서 적용하면 〈그림 3〉과 같은 확장 문서를 얻을 수 있다.

(간청하다, 0.4), (그렇다, 0.4), (그릇되다, 0.4), (꾸짖다, 0.4), (달라다, 0.4), (대하다, 0.4), (덮다, 0.4), (따위, 0.5), (못하다, 0.4), (바, 0.5), (바라다, 0.4), (벌하다, 0.4), *(빌다, 0.6)*, (사람, 1.0), (사물 0.5), (신, 0.5), (아니하다, 0.4), (여기다, 0.4), *(용서, 0.7)*, (이루다, 0.4), *(인정하다, 0.6)*, (일, 1.0), *(자기, 0.7)*, (자신, 0.5), *(잘못, 0.7)*, (잘못하다, 0.4), (잘하다, 0.4), (죄, 0.5), (하다, 0.4)

〈그림 3〉 단어 '사과'에 대한 2차 확장 문서

〈그림 3〉에서 기울임체(italic font)는 1차 확장에서 포함된 단어이고 밑줄(underline)로 표시된 단어는 2번 혹은 그 이상 출현되어 그 가중치를 합한 단어들이다. 이와 같은 방법으로 3차 확장을 통해서 최종 확장 문서가 결정된다.

3.3 무자질 군집화를 통한 유의어 추출

이 절에서는 검색 대상 단어에 대한 단어 군집화를 유의어 단어를 추출하고 추출된 유의어를 이용해서 기본 문서의 확장 방법을 기술한다. 본 논문에서는 기본적으로는 2.3절에서 설명한 3-Pass-TTA 알고리즘을 개선한 (Wong, Liu and Bennamoun 2007)을 그대로 사용하였다. 그러나 본 논문에서 많은 단어를 대상으로 군집화하므로 이 알고리즘을 그대로 구현할 수 없었다. 많은 단어를 대상으로 알고리즘으로 수행할 수 있도록 알고리즘을 일부 개선하였으며 그 결과는 〈알고리즘 1〉과 같다.

〈알고리즘 1〉의 알고리즘 antClustering( )는 먼저 3-Pass-TTA 알고리즘에서와 같이 antTraverse(〈알고리즘 2〉)를 호출하여 기본적인 단어 군집 G_1을 형성한다. 〈알고리즘 1〉의 5번에서 7번까지는 일차적으로 형성된 군집 G_1을 여러 가지 측도를 이용해서 재구성한다. 여기서 pickup_trail( )은 군집과정에서 형성된 트리의 단말노드를 수집하는 함수이고, merge( )는 두 군집을 병합하여 하나의 군집으로 만드는 함수이다.

Algorithm 1: antClustering($\{t_1, ..., t_n\}$)
입력 : 군집 대상의 단어 $T = \{t_1, ..., t_n\}$
출력 : 단어 군집
1: $r_0 = \{t_1, ..., t_n\}$
2: (θ_1, θ_2, θ_3)의 값을 설정한다.
3: ant = new_ant()
3: ant.antTraverse(r_0) // 〈알고리즘 2〉
4: G_1 = ant.pickup_trail( )
5: **foreach** $r_x \in G_1$:
5-1: **foreach** $r_y \in G_1$ **and** $r_x \neq r_y$: **if** $f^2_{TTA}(r_x, r_y) \rangle \theta_2$: merge(r_x, r_y)
6: G_2 = ant.picup_trail( )
7: **foreach** $r_x \in G_2$:
7-1: if $|r_x| \rangle 1$: **continue**
7-2: **foreach** $r_y \in G_2$ **and** $r_x \neq r_y$: if $f^3_{TTA}(r_x, r_y) \rangle \theta_3$: merge(r_x, r_y)
8: **return** ant.picup_trail( )

〈알고리즘 1〉 개선된 3-Pass-TTA 알고리즘: antClustering

Algorithm 2: antTraverse(r_m)
입력 : 단어 군집 r_m
출력 : 단어 군집들
1 : **if** $|r_m| == 1$ **then return**
2 : $\{a, b\} = r_m$에서 유사도가 가장 떨어지는 두 단어를 선택한다.
3 : **if** $f^1_{TTA}(a, b) \rangle \theta_1$ **then return**
4 : $\{r_{m1}, r_{m2}\} = r_m$의 부-노드(sub-node)로 생성한다.
5 : a와 b를 각각 r_{m1}과 r_{m2}로 이동한다.
6 : **foreach** $x \in r_m$:
7 : **if** NGD(a, x) $\rangle$ NGD(b, x) : x를 r_{m1}에 추가한다.
8 : **else** : x를 r_{m2}에 추가한다.
9 : ant1 = new_ant( )
10: ant1.antTraverse(r_{m1})
11: ant2 = new_ant()
12: ant2.antTraverse(r_{m2})

〈알고리즘 2〉 개선된 3Path-TTA 알고리즘의 antTraverse( )

〈알고리즘 2〉의 antTraverse( )는 하향식 방법(top-down approach)으로 하나의 군집을 두 군집으로 재귀적으로 분리하며 군집 내 유사도(intra-cluster similarity)가 θ_1보다 클 때까지 재귀적으로 반복해서 분리한다. 단어를 각 군집에 분배하는 방법은 k-mean 군집화

알고리즘(Hartigan and Wong 1979)과 같이 유사도(식 (3))가 가장 가까운 군집 씨앗에 할당한다. 군집 씨앗(cluster seed)은 k-means 알고리즘과 달리 군집 내에서 가장 유사하지 않은 두 단어를 선택하여 이를 군집 씨앗으로 간주한다.

$$f^1_{TTA}(x,y) = 1 - NGD(x,y) \quad \text{----------(3)}$$

여기서 x와 y는 단어이고 $NGD(x,y)$는 2장의 식 (1)에서 기술했던 두 단어 사이의 구글 거리이다. 〈알고리즘 1〉에서 사용된 $f^2_{TTA}(r_x, r_y)$와 $f^3_{TTA}(r_x, r_y)$는 각각 식 (4)와 (7)과 같이 정의된다(Wong, Liu, and Bennamoun 2009).

$$f^2_{TTA}(r_x, r_y) = \frac{1}{e^{(H[S] - S(r_x, r_y))} e^{V(r_x, r_y)}} \quad \text{---(4)}$$

여기서 r_x와 r_y는 단어 군집이고, $S(r_x, r_y)$는 두 군집 r_x, r_y 간의 유사도로 식 (5)와 같이 정의되고, $V(r_x, r_y)$는 두 군집 r_x와 r_y에 속하는 단어 쌍에 대한 유사도의 표준편차이다(식 (6)). 마지막으로 $H[S]$는 $S(r_x, r_y)$ 중에서 가장 큰 값을 나타낸다.

$$S(r_x, r_y) = \frac{\sum_{a \in r_x} \sum_{b \in r_y} f^1_{TTA}(a,b)}{|r_x||r_y|} \quad \text{----------(5)}$$

$$V(r_x, r_y) = \sqrt{\frac{\sum_{a \in r_x} \sum_{b \in r_y} (f^1_{TTA}(a,b) - S(r_x, r_y))^2}{|r_x||r_y|}} \quad \text{---------(6)}$$

$$f^3_{TTA}(r_x, r_y) = \frac{\sum_{a \in r_x} \sum_{b \in r_y} n^o W(a,b)}{|r_x||r_y|} \quad \text{--------(7)}$$

$NGD(x,y)$를 구하기 위해서 구글에서 제공하는 openAPI에서 검색된 전체 페이지 수를 제공하지 않으므로, 검색 페이지 수는 실제 검색 페이지를 내려받아서 HTML을 파싱하여 계산한다. $n^o W$를 계산하기 위해서 한글 위키피디아 전체를 파일로 다운로드 받아서 HTML을 파싱하여 미리 생성된 그래프로부터 구한다. 본 논문에서 사용된 임계값 θ_1과 θ_2 그리고 θ_3는 (Wong, Liu and Bennamoun 2009)에서 제공하는 것을 그대로 사용하였으며, 각각 0.69, 0.88, 0.91이다.

4. 실험 및 평가

정보검색 시스템을 위한 성능 평가 방법으로는 정확률(precision)과 재현율(recall)이 널리 사용되나(Baeza-Yates and Riberio-Neto 1999), 표제어 검색시스템에 대한 연구는 거의 이루어지지 않았으며 성능 평가 방법도 특별히 제안되지 않았다. 본 논문에서 표제어 검색 시스템의 성능을 평가하기 위해 새로운 평가 측도로 r-포함률(r-inclusion rate) rI을 제안한다(식 (9)).

$$rI = \frac{n_r}{N} \quad \text{----------(9)}$$

여기서 N은 검색 대상의 전체 표제어 수이고 n_r은 1위에서 r번째 순위 사이에 정답을 포함하는 표제어의 수를 의미한다. 즉, r-포함률은 표제어 검색시스템에서 검색한 r개의 표제어 중에 정답이 포함되어 있을 확률을 의미한다.

제안된 표제어 검색시스템을 평가하기 위해 두 가지 실험을 수행한다. 첫 번째 실험은 각 표제어에 대해서 사전의 뜻풀이를 표제어 검색시스템의 질의로 사용한 경우이고, 또 다른 실험은 사용자 5명에 의해서 직접 작성된 표제어의 뜻풀이를 질의로 사용한 경우이다.

4.1 사전 뜻풀이를 이용한 시스템 성능 평가

이 절에서는 표제어의 사전 뜻풀이를 입력하여 그 표제어가 얼마나 정확하게 검색되는지를 평가한다. 실험 대상 표제어는 수집된 모든 표제어를 그대로 사용하였으며 총 표제어 수 N은 519,109개이다. 〈표 2〉는 표제어 검색 시스템에서 r-포함률을 보이고 있다.

〈표 2〉에서 보는 바와 같이 1-포함률은 약 74%이다. 이는 단어의 뜻풀이를 질의로 사용할 경우에 약 74%가 1위로 검색됨을 의미한다. r이 16 이상일 때 거의 100%에 가까운 성능을 보였다. 일반적으로 정보검색 시스템이 첫 페이지를 10 ~ 20개의 페이지를 보여주는 것으로 감안할 때 표제어 검색시스템에서도 상위 20개 단어를 첫 페이지에 보여준다면 충분히 원하는 단어를 찾을 수 있다는 것으로 생각한다. 이와 같이 좋은 성능을 보인 이유는 실제 문서를 확장할 때 사전을 직접 사용했기 때문이다. 그러나 실제 사용자가 원하는 단어를 찾기 위해서는 사전의 뜻풀이 말을 이용할 수 없으며 사전의 뜻풀이와도 전혀 다른 형태의 질의를 사용할 것이다.

〈표 2〉 사전 뜻풀이를 이용한 시스템 성능 평가

r	r-포함률	r	r-포함률
1	74.34	11	91.81
2	78.01	12	93.00
3	80.98	13	96.48
4	82.47	14	97.00
5	83.24	15	97.24
6	84.56	16	99.45
7	86.10	17	99.68
8	88.34	18	99.68
9	89.25	19	99.79
10	90.94	20	99.79

4.2 사용자 뜻풀이에 대한 성능 평가

4.2.1 실험 대상 단어 선정 및 사용자 뜻풀이 구축

실험 대상 단어는 국립국어원이 2004년에 발표한 외국인을 위한 한국어 학습용 기본 어휘 6,000단어 중 A등급의 명사류 단어 200개를 무작위로 추출하여 단어를 선정하였다(〈표 3〉 참조). 선정된 200개의 단어에 대해서 5명의 연구원(대학생)에게 각 단어의 뜻풀이를 작성하여 1000개의 사용자 뜻풀이를 수집하였다. 〈표 4〉는 사용자1이 작성한 200개 단어 중에서 40개의 사용자 뜻풀이를 보여주고 있다.

〈표 3〉 사용자 뜻풀이에 선정된 200개의 단어

1. 가수	2. 나라	3. 마지막	4. 삼월	5. 아침	6. 은행	7. 책
8. 가족	9. 나이	10. 말씀	11. 색	12. 안	13. 음악	14. 처음
15. 갈비	16. 날	17. 매일	18. 샌드위치	19. 앞	20. 의자	21. 청소
22. 건물	23. 날짜	24. 머리	25. 생선	26. 약	27. 이때	28. 초등학교
29. 게임	30. 내년	31. 문제	32. 생활	33. 약속	34. 이번	35. 축구
36. 결혼	37. 냉면	38. 물건	39. 서점	40. 양복	41. 이월	42. 취미
43. 경찰	44. 넥타이	45. 바나나	46. 선물	47. 어깨	48. 인사	49. 치마
50. 계란	51. 노래	52. 바람	53. 선생님	54. 어린이	55. 일	56. 친구
57. 계획	58. 누나	59. 박물관	60. 설탕	61. 어제	62. 일요일	63. 칠판
64. 고등학교	65. 눈	66. 반	67. 세탁기	68. 얼굴	69. 일주일	70. 칫솔
71. 고양이	72. 뉴스	73. 발	74. 쇠고기	75. 엄마	76. 잎	77. 카메라
78. 곳	79. 다리	80. 밤	81. 수건	82. 연습	83. 자리	84. 커피
85. 공부	86. 다음	87. 방	88. 수업	89. 열쇠	90. 자전거	91. 컵
92. 공중전화	93. 달	94. 배	95. 수영장	96. 영화	97. 잔	98. 콜라
99. 과	100. 달력	101. 버스	102. 숙제	103. 옛날	104. 잠깐	105. 키
106. 과자	107. 닭고기	108. 별	109. 술	110. 오래간만	111. 장미	112. 택시
113. 교수	114. 대답	115. 병	116. 시간	117. 오렌지	118. 재미	119. 해
120. 구경	121. 대학	122. 병원	123. 시월	124. 오빠	125. 종이	126. 햄버거
127. 구름	128. 대학생	129. 보통	130. 시장	131. 오전	132. 주말	133. 형
134. 군인	135. 댁	136. 봄	137. 식당	138. 올해	139. 주스	140. 혼자
141. 그날	142. 도시	143. 부모님	144. 식탁	145. 외국	146. 준비	147. 화요일
148. 그때	149. 돈	150. 부엌	151. 신발	152. 외국인	153. 중학교	154. 환자
155. 그림	156. 동생	157. 북쪽	158. 십이월	159. 요리	160. 지갑	161. 회의
162. 근처	163. 동쪽	164. 불고기	165. 쓰레기	166. 우리나라	167. 지난달	168. 휴일
169. 급	170. 돼지고기	171. 비누	172. 아내	173. 우유	174. 지도	175. 휴지통
176. 기숙사	177. 등산	178. 비빔밥	179. 아래	180. 운동	181. 지하	182. 힘
183. 길	184. 딸기	185. 빨간색	186. 아빠	187. 운동화	188. 질문	
189. 김치	190. 떡	191. 사과	192. 아이스크림	193. 월요일	194. 찌개	
195. 꿈	196. 라면	197. 산	198. 아주머니	199. 위험	200. 차	

〈표 4〉 사용자 뜻풀이에 선정된 200개의 단어

번호	단어	사용자가 작성한 뜻풀이
1	가수	노래하는 사람
2	가족	집에서 같이 살고 있는 사이
3	갈비	고기 중에 비싼 것
4	건물	높게 지어 올리는 것. 빌딩
5	게임	하다보면 시간 가는 줄 모르고 재밌는 것
6	결혼	사랑하는 사람과 평생을 함께 살아감
7	경찰	민중의 지팡이
8	계란	반찬 없을 때 먹을 수 있음
9	계획	지키려고 세우지만 항상 끝까지 지켜지지는 않음
10	고등학교	중학교와 대학교 사이
11	고양이	쥐의 천적
12	곳	장소 ~~하는 곳
13	공부	큰 사람이 되려면 열심히 해야 하는 것
14	공중전화	길거리에 세워져 있는 전화기
15	과	대학교 전공 학과
16	과자	종류가 많고 맛있는 간식
17	교수	대학교의 선생님
18	구경	발생하는 상황을 지켜보고 있는 것
19	구름	하늘에 떠있는 하얀 연기
20	군인	젊은 남자들이 나라를 지키고 있음. 병역 의무
21	그날	과거를 회상할 때 그날은 ~했었다
22	그때	어느 한 시점을 강조
23	그림	여러 색으로 문자를 대신한 표현
24	근처	어느 장소의 주변
25	급	단계를 나눴을 때 특정한 영역 s급 a급
26	기숙사	학생들이 학교를 다니면서 생활하는 집
27	길	사람이나 자동차 등이 지나갈 수 있게 낸 공간
28	김치	한국 사람들의 가장 기본적인 반찬 배추, 무, 총각
29	꿈	미래에 되고 싶은 것이나 희망, 잠자면서 꾸는 것
30	나라	국민 전체가 살고 있는 곳
31	나이	매년 늘어가는 것
32	날	중요한 날이 있는 하루 ~~날
33	날짜	하루하루를 숫자로 표현
34	내년	올해를 지나고 다음 년도
35	냉면	얇고 잘 안 끊어지는 음식
36	넥타이	정장을 입고 목에 포인트로 매는것
37	노래	말에 음을 붙여 부르는 것
38	누나	남자가 자신보다 나이 많은 여자에게 부를 때
39	눈	사람이 살면서 앞을 볼 수 있는 중요한 기관
40	뉴스	매일 새로운 소식을 전하여 주는 방송. 새 소식

4.2.2 문서 확장 방법에 대한 성능 평가: r-포함률

〈표 3〉은 앞에서 구축된 1,000개의 사용자 뜻풀이에 대해서 평균 r-포함률을 보이고 있다. 평균 r-포함률은 1000개의 사용자 뜻풀이를 질의로 사용하여 얻은 r-포함률을 평균한 것이다. 본 논문에서 제안된 표제어 검색시스템의 문서 확장으로 뜻풀이 확장과 유의어 확장을 사용하였다(3장 참조). 〈표 5〉는 문서 확장 기법에 따른 성능을 나타내고 있다. 〈표 5〉를 보면 사전을 이용한 뜻풀이 확장(①)만으로는 52.1%의 20-포함률을 가지며, 단어 군집화를 이용한 유의어 확장(②)을 추가했을 경우, 20-포함률이 14.8% 증가한 66.9%를 보였다. 이는 사전 뜻풀이의 성능에 비하면 매우 낮은 결과이고 사전 뜻풀이의 1/1와 비슷한 수준이다. 이는 실제 사용자 뜻풀이가 사전 뜻풀이와 얼마나 다른지를 간접적으로 보여주고 있다.

제안된 표제어 검색시스템의 성능은 20-포함률이 약 67%이다. 이는 아직도 더 많은 연구가 진행되어야 함을 암시하고 있다. 먼저 객관적인 평가를 위한 평가 집합(evaluation set)이 절실히 필요하다. 본 연구에서 사용한 평가 집합은 많은 경우에 제3자는 이해할 수 없는 뜻풀

〈표 5〉 문서 확장에 따른 성능 평가(평균 r-포함률)

r	평균 r-포함률	
	사전 뜻풀이 확장(①)	① + 유의어 확장(②)
1	2.4	3.1
2	4.8	7.0
3	7.6	10.3
4	10.1	14.1
5	12.3	18.0
6	14.7	21.5
7	17.8	24.2
8	19.8	27.7
9	22.4	31.3
10	25.2	34.6
11	27.6	37.3
12	30.6	41.1
13	32.8	43.2
14	35.2	47.5
15	38.0	51.0
16	40.5	52.8
17	43.6	57.0
18	46.0	60.0
19	49.0	63.2
20	52.1	66.9

이들이 종종 보인다. 예를 들면 〈표 4〉에서 '건물'을 "높게 지어 올리는 것. 빌딩"이라는 뜻풀이를 사용한다. 이 뜻풀이로 표제어 '건물'을 바로 찾기는 쉽지 않다. 또한 초기 문서로 사전의 단어 뜻풀이를 사용하는 것이 올바른 방법인지도 한번쯤 생각해보아야 할 것이다.

4.2.3 문서 확장 방법에 대한 성능 평가: MRR

만약 사용자 뜻풀이를 질의/응답 시스템의 사용자 질의로 간주하면 질의/응답 시스템어서 가장 일반적으로 사용되는 평균역순위(mean reciprocal rank)(Voorhees 1999)를 사용할 수도 있을 것이다(식 (8)).

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_{q_i}} \quad \text{----------(8)}$$

여기서 Q는 전체 질의(사용자 뜻풀이) 집합이고, $rank_{q_i}$는 i번째 질의의 정답 순위이다. 평균역순위는 클수록 좋은 시스템이며 평균역순위가 1이면 모든 질의의 정답 순위가 1이라는 의미하며 1위에 가까우면 가까울수록 좋은 시스템이다. 〈표 6〉은 사용자 뜻풀이에 대한 MRR의 성능을 보이고 있다. 사용자 뜻풀이 확장의 경우 MRR이 0.11이고, 유의어 확장을 포함하면 0.13이다. 이 성능은 객관적으로 얼마나 좋은 결과인지를 비교하기 매우 어렵다. 왜냐하면 공개된 평가 집합을 찾을 수 없어서 객관적인 평가를 수행할 수 없었고 또 다른 표제어 검색시스템이 존재하지 않기 때문이다.

유의어 확장을 포함할 경우 대부분의 경우 성능이 개선되었으나 사용자4의 경우 MRR이 오히려 떨어지는 현상이 나타났다. 이는 사용자 뜻풀이가 사전 뜻풀이와 너무 큰 차이를 보이므로 이와 같은 현상이 나타났다. 〈표 4〉에서 '갈비'에 대한 사용자 뜻풀이가 "고기 중에 가장 비싼 것"이나 사전 뜻풀이는 "소나 돼지, 닭 따위의 가슴통을 이루는 좌우 열두 개의 굽은 뼈와 살을 식용으로 이르는 말"이다. 이 두 뜻풀이가 너무나 큰 차이를 보이고 있다. 그 밖에도 많은 단어에 대한 사용자 뜻풀이를 살펴보면 개인적이고 일반적이지 못한 문제가 있

〈표 6〉 문서 확장에 따른 성능 평가(MRR)

사용자	사전 뜻풀이 확장(①)		① + 유의어 확장(②)	
	MRR	가장 낮은 순위	MRR	가장 낮은 순위
사용자1	0.0996	35	0.1750	27
사용자2	0.1166	37	0.1212	30
사용자3	0.1270	32	0.1674	24
사용자4	0.0843	70	0.0710	65
사용자5	0.1073	33	0.1308	25
평균	0.1070		0.1331	

다. 향후에 이점을 개선하기 위해 좀 더 질 좋은 사용자 뜻풀이를 수집하여 할 것이며 표제어 검색시스템으로 보다 객관적인 평가를 위한 평가 집합을 구축하는 연구도 아울러 진행되어야 할 것으로 생각된다.

5. 결 론

본 논문은 정보검색 기술을 이용한 표제어 검색시스템을 제안하였다. 제안된 표제어 검색시스템은 일반적인 사전 시스템과는 정반대의 개념으로 단어의 정의나 의미가 입력으로 주어질 때, 그 정의나 의미에 가장 적절한 단어를 검색하는 시스템이다. 이러한 시스템을 구축하기 위한 가장 간단한 방법으로 사전의 뜻풀이를 문서로 간주하는 정보검색 시스템을 생각할 수 있다. 이 경우 문서(사전 뜻풀이)의 길이가 너무 짧아 사용자 질의에 대해 적절한 단어를 검색할 수 없다. 이 문제를 해결하기 위해서 본 논문에서는 정보검색에서 사용되는 질의 확장(query expansion) 개념을 이용한다. 즉 정보검색에서 질의 확장 개념을 표제어 검색시스템에서 문서 확장에 적용하였다. 본 논문에서는 문서 확장 방법으로는 사전을 이용한 뜻풀이 확장(glossary expansion)과 단어 군집화를 이용한 유의어 확장(similar word expansion)을 사용했다. 뜻풀이 확장은 주어진 단어의 뜻풀이에 속하는 단어의 뜻을 문서에 포함시키는 방법이며 본 논문에서는 3단계까지만 확장했다. 유의어 확장은 무자질 군집화 알고리즘을 통해서 유의어를 찾고, 찾아진 유의어를 문서에 포함시키는 방법이다. 이와 같이 다양한 방법으로 확장된 문서와 사용자의 질의(특정 단어에 대해서 길게 풀어서 설명한 정의나 의미)의 유사도를 계산하여 단어를 검색한다. 이와 같은 방법으로 구현된 시스템은 단어의 뜻풀이 그 자체를 입력으로 할 때, 16-포함률이 거의 100%에 달하였다. 또한 사용자들이 직접 작성한 뜻풀이에 대해서는 20-포함률이 71.4%였으나 사용자 본인들의 정말로 필요에 의해서 뜻풀이를 작성한다면 더 좋은 성능을 보일 수 있을 것으로 생각된다.

본 논문에서 제안된 표제어 검색시스템의 검색 시간은 매우 빠르다. 그러나 새로운 단어가 새로 추가하기 위해서는 많은 노력이 필요하다는 단점이 있다. 앞으로 이러한 문제를 개선하기 위한 연구가 더 필요할 것으로 생각되며 정확한 성능 평가를 위한 평가 측도와 객관적인 평가를 위한 평가 집합(evaluation set)에 대한 연구도 필요할 것이다. 또한 문서를 확장할 때 의미 분석을 통해서 유의어 외에 반의어도 확장된다면 조금 더 높은 성능을 보일 수 있을 것이다.

참고문헌

강현규, 박세영. 1988. 정보 검색.『정보처리』, 5(5) : 37-47.

국립국어원. 2007.『21세기 세종계획 최종 성과 발표회 자료집』. 문화관광부・국립국어원.

박은진, 김재훈, 옥철영. 2005. 자질 확장에 따른 용어 클러스터링의 성능 향상.『한국정보과학회 제32회 추계학술발표회 논문집』, 32(2): 529-531.

Andrews, N. and E. Fox. 2007. *Recent Developments in Document Clustering*, Technical Report TR-07-35, Computer Science, Virginia Tech.

Baeza-Yates, R. and B. Ribeiro-Neto. 1999. *Modern Information Retrieval*, Addison Wesley.

Bilotti, M. W. and E. Nyberg. 2008. "Improving Text Retrieval Precision and Answer Accuracy in Question Answering Systems." *Proceedings of the ACL 2nd Workshop on Information Retrieval for Question Answering*, pp.1-8.

Cilibrasi, R. L. and P. M. B. Vitanyi. 2007. "The Google Similarity Distance." *IEEE Transactions on Knowledge and Data Engineering*, 19(3): 370-383.

CRFPP. 2011. 〈http://crfpp.sourceforge.net〉.

German, D. J. 2000. "Basic Concepts in Child Word Finding." In *German, D. J. Test of Word Finding-Second Edition, Examiners Manual*. p.1-15. Austin.

Handl, J., J. Knowles, and M. Dorigo. 2003. *Ant-based Clustering: A Comparative Study of its Relative Performance with Respect to K-means, Average Link and 1D-som*, Technical Report TR/IRIDIA/ 2003-24. IRIDIA, Universite Libre de Bruxelles, Belgium.

Hartigan, J. A. and M. A. Wong. 1979. "Algorithm AS 136: A K-Means Clustering Algorithm." *Journal of the Royal Statistical Society*, 28(1): 100-108.

Hodgel, V. and J. Austin. 2002. "Hierarchical Word Clustering-Automatic Thesaurus Generation." *Neurocomputing*, 48: 819-846.

Jain, A., M. Murty, and P. Flynn. "Data Clustering: A Review." *ACM Computing Surveys*, 31(3): 264-323.

Manning, C. D. and H. Schutze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.

Voorhees, E. M. 1999. "The TREC-8 Question Answering Track Report." *Proceedings of the 8th Text Retrieval Conference*, 77-82.

Wise, R., F. Chollet, U. Hadar, K. Friston, E. Hoffner, and R. Frackowiak. 1991. "Distribution of Cortical Neural Networks Involved in Word Comprehension and Word Retrieval." *Brain*, 114(4): 1803-1817.

Wong, W., W. Liu, and M. Bennamoun. 2006. "Terms Clustering Using Tree-traversing Ants and Featureless Similarities." *Proceedings of the International Symposium on Practical Cognitive Agents and Robots*.

Wong, W., W. Liu, and M. Bennamoun. 2007. "Tree-Traversing Ant Algorithm for Term Clustering Based on Featureless Similarities." *Data Mining Knowledge Discovery*, 15: 349-381.

Wong, W., W. Liu, and M. Bennamoun. 2009. "Featureless Data Clustering." *Handbook of Research on Text and Web Mining Technologies*, 141-164.