TFIDF를 이용한 키워드 추출 시스템 설계

Design of Keyword Extraction System Using TFIDF

이 말 레* 배 환 국**
(Mal-Rey Lee) (Hwan-Kuk Bae)

요 약 본 논문에서는 먼저 Anchor Text의 단어들이 키워드로 적합한지 TFIDF를 이용하여 테스트하였다. 그 결과는 가중치가 높아서 키워드로 적합한 단어가 있었는가 하면, 아예 문서에 나오지도 않는 단어가 있어 키워드로 적합하지 않은 단어도 있었다. 이를 해결하기 위하여 새로운 키워드 추출 방법을 제시하였다. 본 논문에서는 적합하지 않은 키워드를 제거함으로써 새로운 키워드를 만들어 내고 TFIDF값을 각 키워드의 가중치로 이용하여 Ranking이 가능하게 하였다. 이렇게 추출된 키워드는 기존의 방법보다 정확도가 높아졌음 증명했다.

키워드 Anchor Text, TFIDF, 에이전트, 개념 그래프, 검색엔진

Abstract In this paper, a test was performed to determine whether words in Anchor Text were appropriate as key words. As a result of the test, there were proper words of high weighting factor, while some others did not even appear in the text, therefore, were not appropriate as key words. In order to resolve this problem, a new method was proposed to extract key words. Using the proposed method, inappropriate key words can be removed so that new key words be set, and then, ranking becomes possible with the TFIDF value as a weighting factor of the key word. It was verified that the new method has higher accuracy compared to the previous methods.

Keywords Anchor Text, TFIDF, Agent, Concept Graph, Search Engine

1. 서론

정보의 바다라 불리 우는 인터넷이 널리 퍼지면서 현대인들에게 있어서 제1의 정보원으로 되어가고 있다. 또한 인터넷 정보의 양이 너무 많아서 필요한 정보를 효과적으로 찾을 수 있는 방법으로 검색엔진이 많이 사용되고 있다. 그러나 현재 대부분의 검색엔진은 네티즌의 요구를 100% 만족시키지 못하고 있으며, 이를 해결하기 위하여 검색방법에 대한 연구가 진행되고 있다. 이러한 연구 중 하나가 개념 기반 검색엔진이다. 개념이란 특정 단어를 유사어 혹은 관계어로 확대시켜 나아가는 과정을 말하며, 개념 기반 검색은 단어의 철자에만 의존하지 않고 단어의 의미를 분석, 단어의 개념관계를 이용하여 검색을 확장하고, 단어의 유사어, 중의어, 계층표현을 가능하게 하는 검색방법이다[3,8].

* 여수대학교 멀티미디어학부
** (주)소프트캠프

웹 그래프는 개념 기반 검색 방법을 이용한 검색 엔진이다. 웹 그래프는 웹 문서의 키워드를 추출하기 위하여 하이퍼링크의 Anchor Text를 이용하였다[4]. 보통 어떤 웹 문서에서 다른 문서로 이동할 경우 Anchor Text를 참고하여 이동하므로 Anchor Text는 하이퍼링크로 연결된 웹 문서의 내용을 대표한다는 것을 묵시적으로 인지하고 있다[5].

실제로 Anchor Text는 어떤 웹 문서의 내용을 사람이 직접 요약해 놓은 것으로 그 문서의 키워드를 추출하는데 중요한 정보가 될 수 있다. 따라서 웹 그래프는 이를 이용하여 키워드를 추출하였고, 그러한 키워드를 이용하여 개념 관계를 나타내었으며, 이를 시각화하여 보여 주었다.

그러나, Anchor Text의 정보가 유용하다는 것은 개념적으로만 추측될 뿐이고, 수치적으로 증명하지는 못하였다. 더구나 Anchor Text는 웹 문서 자체에 있는 문자열들이 아니므로 Anchor Text 전체를 키워드로 추출하는 것은 문제가 있다.

본 논문에서는 실제로 Anchor Text가 문서를 대표할 수 있는지에 대해서 자동 인덱싱 기법인 TFIDF를 이용하여 이를 비교분석하며, 이를 토대로 지능형 정보 추출 에이전트 시스템을 제시하였다. 본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 기반연구로 TFIDF에 대해서 알아보고, 관련 연구로 웹 그래프에 대해서 설명한다. 3장에서는 실험에 필요한 시스템의 설계 및 구현에 대해서 설명하며, 4장에서는 실험 및 평가를 하여 5장에서 결론을 맺는다.

2. TFIDF와 웹 그래프

2.1 TFIDF

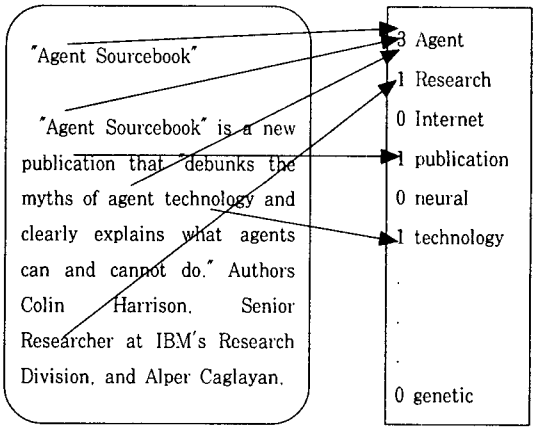
통계적 기법에 의한 자동 색인에서는 색인 어 선정을 위한 기준으로 주어진 문헌에서 특정 단어가 얼마나 자주 사용되었는가 하는 빈도 수 정보가 사용된다. 단어의 사용 빈도 수는 산출 방식에 따라 단순 빈도 수와 상대 빈도 수로 구분된다. 단순 빈도 수는 문헌의 크기나 분석 대상이 되는 텍스트의 길이 또는 단어의 출현 빈도 수를 전혀 고려하지 않은 것이므로, 이것을 기준으로 색인 어를 선정하기에는 부적절한 면이 있다. 상대 빈도 수는 이와 같은 문제점을 고려한 것으로써, 색인 어를 선정하는 기준으로 사용되기에 적합한 형태이다. 상대 빈도 수는 단어 빈도수를 문헌의 크기 등으로 나누어 줌으로써 빈도 수를 정규화(normalization)한 것이다. 단순 빈도수와 상대 빈도수의 산출 방식은 물론, 주제어 선택시 사용될 빈도수의 한계치는 모두 실험적으로 결정된다. 빈도 수가 지나치게 높거나 지나치게 낮은 단어는 주제어에서 제외된다. 즉, 문헌에서 빈번하게 나타나는 기능어를 수록한 용어 리스트를 사용하여 고 빈도어를 먼저 제거한 다음 나머지 단어들을 빈도수 순으로 배열하고, 임의로 정한 빈도 수의 최저 한계치를 초과하는 단어들을 색인어로 선택한다.

통계적 기법으로 대표적인 것이 TFIDF이다. TFIDF 방식이란 하나의 문서 d에서 단어 w에 대한 weight값을 산출하는 방식으로 다음의 수식으로 표현할 수 있다[2, 6].

$$TFIDF(w,d) = TF(w,d) * \log(\frac{n}{DF(w)})$$

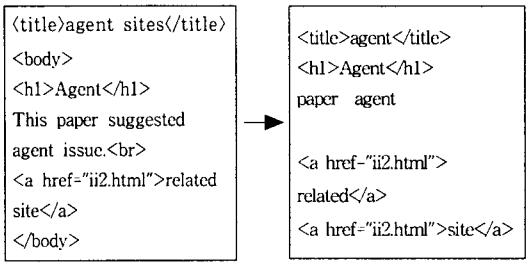
TF(w,d) : 문서 d 에 단어 w 가 나타나는 회수
DF(w) : 단어 w 가 들어가는 문서의 총 수
n : 전체 문서의 총수

즉, 어떤 단어에 대한 중요도는 그 단어가 문서에 나온 횟수(Term Frequency)에 비례하고, 그 단어가 있는 모든 문서의 총 수(Document Frequency)에 반비례한다는 것이다. log가 있는 까닭은 수집된 문서가 많아질 때, 즉 n의 값이 커지는 것에 대해 알맞게 조절하기 위함이다. TFIDF 방식을 이용하면 하나의 문서 중에서 가장 weight 값이 높은 단어가 그 문서에 키워드로 채택된다. 일반적으로 정보검색이나 텍스트 학습에서 하나의 문서를 표현하기 위해서는 TFIDF-vector 표현 방식을 많이 이용한다. TFIDF-vector 표현 방식은 하나의 문서에 대해서 단어의 순서나 구조에 관계없이 단순히 하나의 문서를 단어들의 모임(bag-of-words)으로 간주하는 방식이다. 이 방식을 이용하면 HTML과 같이 정형화가 잘 되어있는 문서나 일반 문서에서 유용하게 적용할 수 있다. <그림 1>은 TFIDF의 일반적인 방식을 보여준다.



<그림 1>일반적인 TFIDF 방식

본 논문에서는 이러한 방식을 기본으로 하여 문서의 구조를 검사하는 방식을 첨가하였다. 일반적으로 HTML 문서들은 일반적인 문서와는 달리 각 단어의 특징을 표현할 수 있는 태그들을 포함하고 있다. HTML 문서에서 사용하는 태그들은 "<" 로 시작하며 "/>" 로 종료하는 구조를 가지고 있다. 그러므로 웹 에이전트는 각 단어가 포함하고 있는 태그까지 하나의 단어로 분류한다. <그림 2>는 본 논문에서 이용된 문서표현 방식을 나타내고 있다.

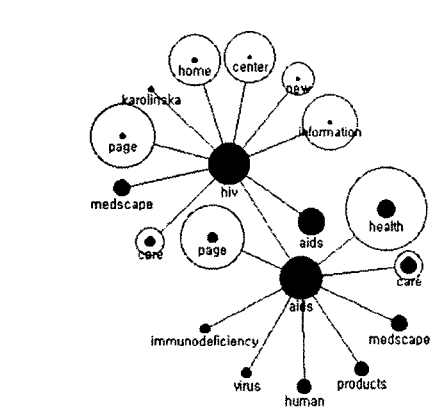


〈그림 2〉 문서표현

2.2 웹 그래프

웹 그래프란 웹 페이지간의 링크 정보를 이용하여, 사용자의 질의에 도식화된 개념 그래프를 보여줌으로써 원하는 정보를 쉽게 검색할 수 있도록 하는 검색엔진을 말한다.

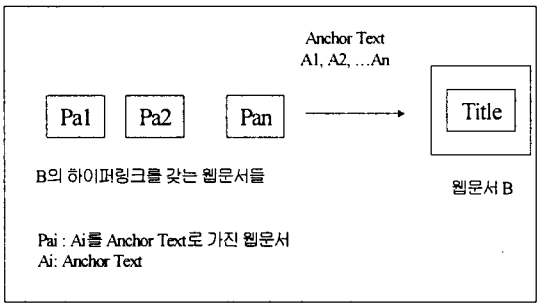
이 검색엔진은 개념 기반 검색 기법을 도입하였다. 개념기반 검색은 단 어의 의미를 분석 하여 단어의 개념관계를 이용해서 검색을 확장하게 해줄 수 있는 검색 방법이다. 따라서 사람의 사고 방식과 유사하며, 효과적이라 할 수 있다.



〈그림 3〉 웹 그래프 실행 예

또한 이 시스템은 키워드를 추출하는데 있어서 하이퍼링크 정보를 이용하였다. 이러한 하이퍼링크의 특징과 정보를 바탕으로, 요약 정보인 Anchor Text 나 Alt Text에서 추출된 핵심어와 관련 웹 문서를 지칭하는 특성을 이용하여 구축된 개념그래프를 사용하여 구축된 지식을 웹의 개념지식이라 한다. 〈그림 3〉은 웹 그래프의 실행 예이다. 아래 그림은 질의어를 aids로 했을 때의 결과이다. 이 상태에서 각 원을 클릭 하면 원에 있는 단어에 해당하는 URL들을 보여 주는 것이다. 이 그림에서 aids란 단어 외에 주위의 것들이 바로 하이퍼링크 정보를 이용하여 추출되어진 키워드들이고 질의어에 관련이 있는 키워드들이다.

웹 그래프는 웹 문서의 핵심어 추출을 위하여 하이퍼링크 정보중 하나인 Anchor Text와 웹 문서의 제목(title) 태그를 이용한다. 구체적인 핵심어 추출 방법은 다음과 같다.



〈그림 4〉 핵심어 추출을 위한 구성요소

웹문서 B의 핵심어 = 웹문서 A의 핵심어 + Anchor Text + B의 URL + B의 Title 이다.

키워드 추출시 고려해야 할 점으로는 full text와 anchor text 인데 Full Text에서 키워드를 추출하면, Anchor Text에서 키워드를 추출하는 것보다 더 정확성이 있을 수도 있다. 그러나 Full Text에서 추출한 키워드는 그 문서의 키워드는 될 수 있을지 몰라도 문서와 문서간, 혹은 키워드와 키워드간의 관계를 나타낸 웹의 개념지식을 나타낼 수 없다. 따라서 본 논문에서는 Full Text가 아닌 Anchor Text에서 키워드를 추출하고자 한다.

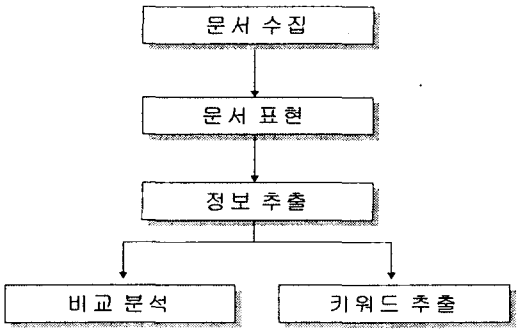
Anchor Text는 사람이 직접 요약한 것이고, 하이퍼링크를 포함하는 웹 문서에 반드시 존재하므로 그 하이퍼링크가 가리키는 곳의 문서의 키워드를 추출에 적합한 용도로 쓰인다[1]. 그러나 Anchor Text 자체가 본문의 내용이 아니고, Anchor Text를 작성한 사람에 따라 다르게 작성되며, 본문의 내용과 무관한 내용도 작성할 수 있다. 따라서 Anchor Text 자체를 어떠한 여과 없이 문서의 키워드로 받아들이긴 힘들다. 본 논문에서는 TFIDF를 통해 Anchor Text 내용중 키워드로 채택할 수 있는 단어들을 추출해냈다.

3. 지능형 정보추출 에이전트 시스템

본 논문에서 제안하고 있는 에이전트 시스템은 웹 문서의 키워드를 추출하는데 있어서 그 문서 자체가 아닌, 그 문서를 가리키고 있는 하이퍼링크의 Anchor Text에서 키워드를 효과적으로 추출하는 데 그 목적이 있다. 본 논문의 시스템은 크게 두 부분으로 나뉜다. 한 부분은 '키워드 추출'부분이고 다른 부분은 '비교 분석' 부분이다. 키워드 추출 부는 Anchor Text에서 TFIDF를 도입하여 키워드를 추출하는 부분이고, 비교 분석부는 Anchor Text에 있는 각각의 단어들이 문서의 키워드로써 얼마나 적합한가를 평가하는 부분이다. 비교 분석을 하는 이유는 Anchor Text에서 추출한 단어들의 가중치가 그 문서에 있는 단어들의 가중치와 비교해 볼 때 상대적으로 작은 가중치의 것들이라면, 키워드로 부적합할 수도 있기 때문이다. 즉, 비교 분석을 한 후 그 결과를 토대로 좀 더 나은 키워드 추출 방법을 제안하기 위해서 이다.

3.1 시스템 구성

본 논문의 시스템의 전체적 구성은 〈그림 5〉와 같다. 문서 수집 단계는 실험에 쓰이는 웹 문서들을 모으는 단계이다. 문서 표현 단계는 모아진 웹 문서들의 키워드를 TFIDF로 계산하기 위해 문서의 용어들을 최적화하는 단계이고, 정보 추출 단계는 모아진 웹 문서들의 용어들간의 TFIDF 값을 계산하여 내림 차순으로 정렬하는 단계이다. 비교 분석 단계는 TFIDF 값이 할당된 용어들과 웹 그래프의 하이퍼링크에서 추출된 키워드들과 비교 분석하는 단계이다. 마지막으로 키워드 추출단계에선 Anchor Text에서 키워드를 추출하는 단계이다.



〈그림 5〉 시스템 개요도

3.2 문서 수집

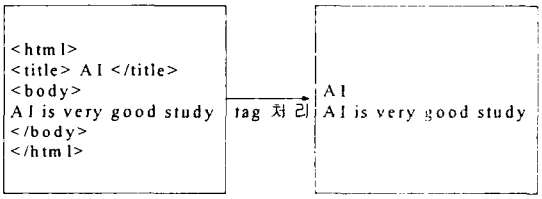
문서 수집 단계는 실험에 사용될 웹 문서를 모으는 단계이며, 본 실험에 사용된 웹 문서들은 html 문서로 한정했으며, 이 문서들은 웹 그래프에서 똑같이 사용될 문서들이므로 웹그래프에서 모은 웹 문서들과 같아야만 한다.

3.3 문서 표현

각 용어들마다 TFIDF값을 구하기 위해서는 용어들에 대해서 다음 오퍼레이션을 수행 해야만 한다.

1) HTML Tag 처리

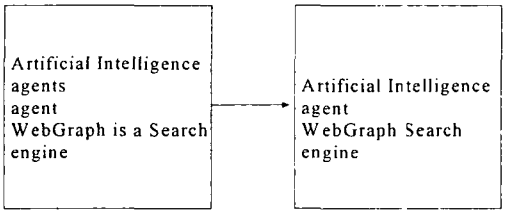
HTML 문서들은 tag로 이루어져 있다 예를 들어 html 문서는 〈html〉로 시작하여 〈/html〉로 끝나며, 〈title〉, 〈/title〉은 문서의 제목을 나타내며, 이러한 tag들은 문서의 실제적인 내용과 아무런 관련이 없으므로 이를 제거 한다.



〈그림 6〉 tag 처리

2) 불용어, 스테밍 처리

is, am, are와 같이 출연 빈도수는 많으나 키워드가 될 수 없는 용어들을 불용어로 간주하여 삭제시킨다. 또한 스테밍 처리를 하여 각 키워드의 어형론적인 변형을 찾는 방법을 제공한다. 즉, comes나 come은 같은 뜻인데도 불구하고, 주어가 무엇이냐에 따라 다르게 나온다. 그리고, nurse와 nurses도 간호사로 뜻은 같지만 단수, 복수의 차이가 있다. 이와 같은 것들을 같이 취급하는 것이다.



〈그림 7〉 불용어, 스태밍 처리

위의 두가지 방법을 이용하면 키워드 파일의 크기를 줄여, 색인에 요구되는 메모리 용량을 줄일 수 있고, 검색 효과를 높일 수 있다.

3.4 정보 추출

3.4.1 문서의 정규화

TFIDF는 Term frequency(어떤 문서에 해당 용어가 나온 횟수)와 Document frequendy(어떤 용어가 나온 문서 수)로 계산되어 진다. 그런데 Term frequency는 단순히 긴 문서라서 높은 값을 가질 수가 있다. 이러한 문제를 해결하기 위하여 문서의 정규화가 필요하다. 이러한 문서 정규화에 대한 방법으로는 여러 가지 방법이 있지만[7], 본 논문에서는 다음과 같은 방법을 사용하였다. 문서의 정규화를 위한 방법은 문서내의 최대 값을 이용한 방법을 이용하였다. 이 방법은 각 문서의 최대 키워드를 이용하여 문서내의 모든 키워드에 대한 가중치를 정규화 하는 것으로, 정규화는 식(1)과 같다.

$$K+(1-K)\times\frac{TF}{TFmax} \quad \text{where } 0 < K < 1 \qquad (1)$$

이때, 정규화에 대한 효율을 위하여 식내의 0.5, 0.5의 가중치는 조절한다. 또한 키워드의 최대수에 대한 정규화의 효율을 위하여 휴리스틱을 사용할 수 있다. 본 논문에서는 두 번째 방법을 사용하여 문서 길이에 따른 문제를 해결하였다.

3.4.2 TFIDF 식

본 논문에서 사용된 TFIDF 식은 2와 같다.

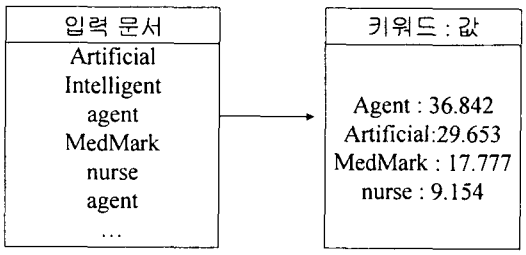
$$(K+(1-K)\times\frac{TF}{TFmax})\times\log(\frac{N}{n}) \qquad (2)$$

where N :총문서수, n: 특정단어가 있는 문서수

전체적으로 볼 때 어떤 키워드가 어떤 문서에서의 중요도는 문서에 키워드가 나타난 수, TF에 비례하며, 그 키워드를 가지고 있는 문서 수, DF에 반비례한다는 것을 따르며, 앞부분의 정규화 식으로 인해 문서 길이에 따른 문제도 해결하였다. n은 총 문서수를 의미한다.

3.4.3 적용

앞에서 언급한 식으로부터 각 문서의 키워드들에 대하여 중요도를 계산한다. 그리고 그 중요도에 따라 내림차순으로 정렬한다. 이는 웹 그래프의 키워드들이 어느 정도 중요도를 갖고 있는지를 알아내기 위함이다.



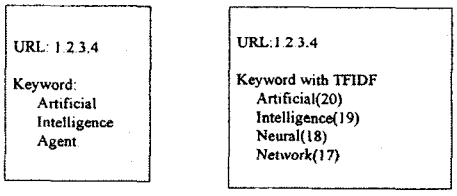
〈그림 8〉 TFIDF 적용

위 그림에서 왼쪽의 것이 주어진 웹 문서에서 단어들의 집합으로 나타낸 것이고 오른쪽이 각 단어에 대해서 TFIDF 식을 이용하여 나온 가중치를 표현한 것이다. 값이 가장 높은게 키워드로써의 적합성이 높다고 할 수 있다.

3.5 비교 분석

이 단계에서 웹 그래프에서 추출된 키워드와 비교를 하게된다. 현재 웹 그래프 시스템은 title, anchor text, 본문에서 키워드를 모두 추출하지만 주는 title과 anchor text이다. title은 제목 그 자체이므로, 그 자체로 키워드가 될 수 있다. 본 논문에서 실험 대상은 anchor text에서 추출된 키워드로 한정한다.

각각의 URL마다 여러개의 키워드가 있을 수 있고 그 키워드를 위에서 계산된 키워드들과 비교를 한다. 현재 TFIDF로 계산된 키워드들은 내림 차순으로 정리하였으므로, 상위의 키워드들과 매치될수록 문서를 대표할 만한 키워드가 웹 그래프에서도 추출되었음을 보여준다. 예를 들면 다음 〈그림 9〉와 같다.



하이퍼링크에서 추출된 키워드 실제 문서에서 TFIDF값을 갖는 단어들

〈그림 9〉 비교 분석의 예

위의 그림에서 왼쪽의 것이 URL 1.2.3.4를 가리키는 하이퍼링크에서 추출된 키워드를 나타내는 것이

고, 오른쪽의 것이 URL 1.2.3.4에 있는 실제 문서의 단어들(문서의 모든 단어들이며 TFIDF 값을 갖는다.)을 나타낸다. 위의 그림에서 볼 때, Artificial 이나 Intelligence 같은 경우, 실제 문서에서도 존재하며, 둘 다 가장 높은 값을 가지므로 그 두 단어는 그 문서에 대해 중요한 단어라고 할 수 있다. 그러나 Agent 같은 경우 아예 문서에 존재하지 않으므로 중요하지 않은 단어라고 할 수 있다. 실제로는 문서에 나타나지 않아도 그 문서를 대표할 수 있는 단어가 있을 수도 있지만 여기서는 그런 경우는 없다고 가정한다.

3.5.1 알고리즘

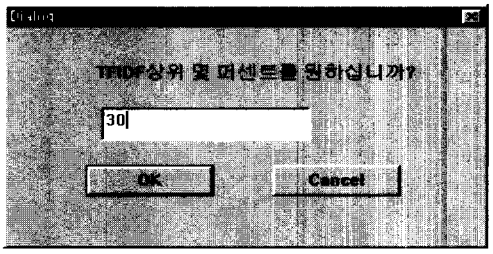
비교분석을 위한 알고리즘은 다음과 같다.

```
void Compare()
{
        do{
                Initialize();
                ReadTFIDF();
                ReadWebGraph();

                for(; percent;){
                        if(!stricmp(wkeyword, tkeyword))
                                        Hit++;
                }

                Hit_rate = Hit / wCount * 100;
        }while();

        total_rate = total_Hit / total_Count * 100;
}
```

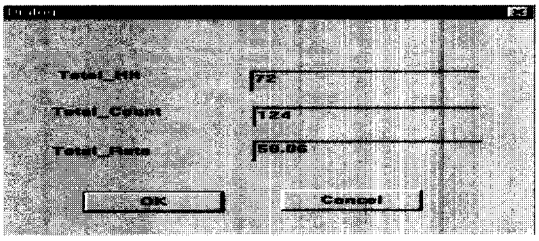
위 알고리즘에서 Initialize()는 각종 변수의 초기화 함수이고, ReadTFIDF()는 TFIDF값들이 구해진 각 문서의 정보를 읽어오는 것이고, ReadWebGraph()는 웹 그래프에서 추출한 키워드에 대한 정보를 읽어오는 것이다. 여기에서 percent란 웹 그래프에서 구해진 키워드들이 TFIDF값 상위 몇 퍼센트 안에 존재하는 가를 알아보는 것이다. 그래서 그 안에 해당 키워드가 있으면 Hit를 증가시켜 전체 웹 그래프에서 추출된 키워드의 개수(wCount)를 이용하여 Hit_rate를 구한다.

3.5.2 실행

이 프로그램은 웹 그래프에서 추출된 키워드가 TFIDF 값 상위 몇 퍼센트 안에 존재하는 가를 알아보는 프로그램으로써 그 몇 퍼센트를 입력하게 하여 결과로 Hit_rate를 보여주게 하였다.



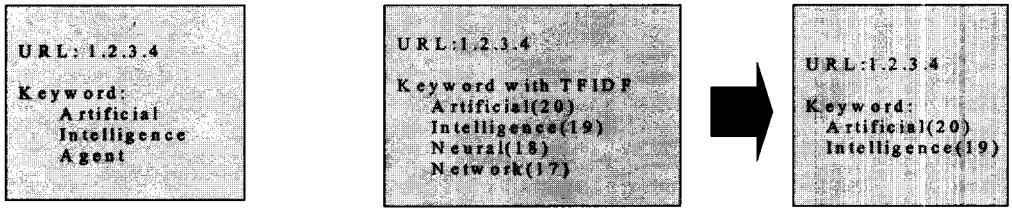
〈그림 10〉 비교 분석 실행 1



〈그림 11〉 비교 분석 실행 2

3.6 키워드 추출

Anchor Text에서 추출한 키워드가 반드시 본문에 나타난다는 보장이 없는 키워드는 삭제하였다. 즉, 위



하이퍼링크에서 추출된 키워드 | 실제 문서에서 TFIDF값을 갖는 단어들 | 새로 추출된 키워드

〈그림 12〉 키워드 추출 예

그림에서 하이퍼링크에서 추출한 키워드는 artificial, intelligence, agent이지만 agent는 실제적으로 문서에 존재하지도 않으므로, artificial 이나 intelligence만 키워드로 채택되는 것이다.

위의 그림에서 보다시피 새로운 키워드 추출이 단순히 agent가 없어진 것만 뜻하는 게 아니라 TFIDF값도 그대로 갖고 있어 Ranking화가 가능하다는 것도 의미한다.

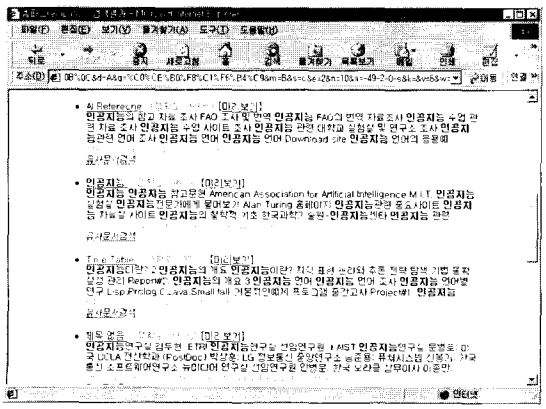
3.6.1 알고리즘

키워드 추출 알고리즘은 다음과 같다.

```
void KeywordExtract()
{

        do{
                Initialize();
                ReadTFIDF();
                ReadWebGraph();

                if(!stricmp(wkeyword, tkeyword))
                        SaveKeyword();

        }while();
        WriteKeyword();
}
```

위의 알고리즘에서 처음의 세 함수는 비교 분석 알고리즘과 같고, SaveKeyword()는 WebGraph의 키워드가 본문의 단어에 있을 때 그 것을 키워드로 채택하는 함수이고, WriteKeyword는 새로 추출된 정보를 파일에 기록하는 함수이다.

3.7 Ranking



〈그림 13〉 empas 검색엔진의 Rangking 예

검색엔진에서 질의어를 넣었을 때 그에 해당하는 URL들을 보여주는 것도 중요하지만, 찾아진 URL들을 어떻게 보여주는 가도 매우 중요한 문제가 된다. 왜냐하면, 질의어에 대해 찾아진 URL들이 수 천, 수 만개에 이를 수 있으므로 관련성이 거의 없는 것을 먼저 보여주면 사용자 입장에서 관련성이 높은 것을 찾고자 노력한다. 대부분 검색엔진는 이를 해결하기 위하여 스트링 매칭을 통해 얼마만큼 매칭되는가에 순서를 매겨 해결한다.

웹 그래프는 원래의 문서에서 얻어진 키워드가 아니므로 위와 같은 방법은 불가능하다. 그러나 본 논문에서는 실험한 중요도 값을 이용하여 그 순서를 매길 수 있다. 어떤 질의어에 해당하는 URL들을 보여주고자 할 때 중요도 값이 높은 것을 우선적으로 보여줌으로써 먼저 보여줌으로써 문제를 해결할 수 있다.

4. 실험 및 평가

4.1 실험 환경

자료를 저장하기 위한 DB로는 MS SQL 서버를 이용하였으며, 수집한 총 웹 문서는 536개이다. 그 중 파일 사이즈가 가장 큰 것은 62KB이었으며, 제일 작은 것은 1KB이다. 운영체제는 Windows NT를 사용하였으며, 본 프로그램을 구현한 컴파일러는 Visual C++6.0을 사용하였다.

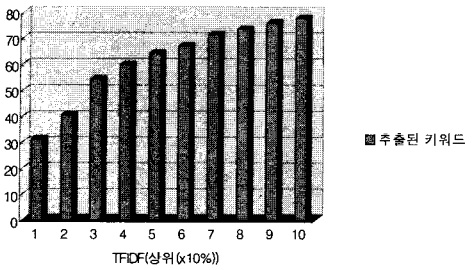
4.2 비교 분석

4.2.1 결과

어떤 웹 문서에 대해서, 그 것을 가리키고 있는 하이퍼링크의 Anchor Text에서의 키워드 추출의 적합성을 평가하는 비교 분석의 결과는 다음과 같다.

(표 1) 비교분석 결과

TFIDF (상위%)	10	20	30	40	50	60	70	80	90	100
추출된 키워드	31.2	40.3	54.2	59.8	64.3	66.9	71.2	73.4	75.9	77.6



〈그림 14〉 비교 분석 결과의 그래프

4.2.2 평가

결과에서 보면 TFIDF 값 상위 10%안에 Anchor Text에서 추출한 키워드는 30%이상임을 알 수 있다. 그 30%의 키워드들은 문서를 대표할 수 있을만한 높은 수치를 갖는 단어라 할 수 있다. 또한 하이퍼링크에서 추출된 키워드 50%가 실제 문서의 단어들 중 TFIDF 값 상위 25%안에 있음을 알수있다. 이 것 역시 Anchor Text의 키워드가 문서의 단어들 중 중요도가 높은 단어 쪽에 속한다는 것을 보여준다. 그러나 상위 100%(단어 전체)에서 하이퍼링크에서 추출된 키워드와 매칭되는 단어는 75%밖에 되지 않는 다는 것도 알 수 있다. 이 것은 하이퍼링크의 키워드가 문서에 존재하지 않을 수 도 있다는 것을 보여주며, 그 키워드들은 문서에 필요 없는 단어라 할 수 있다. 전체적으로 보면, Anchor Text에서 추출한 키워드가 문서상에 있으면, 대체적으로 높은 가중치를 가지고 있어서 키워드로써 적합한 반면에, 문서에 단 한번도 나오지 않는 키워드도 있어서 키워드를 추출할 때 이를 고려하여야 한다.

4.3 키워드 추출

새롭게 추출된 키워드에 대한 평가는 기존의 WebGraph에서 추출된 키워드와 비교해서 평가한다. 평가 기준은 정확율(Precision)과 재현율(Recall)로 하였다. 정확율과 재현율에 대한 식은 다음과 같다.

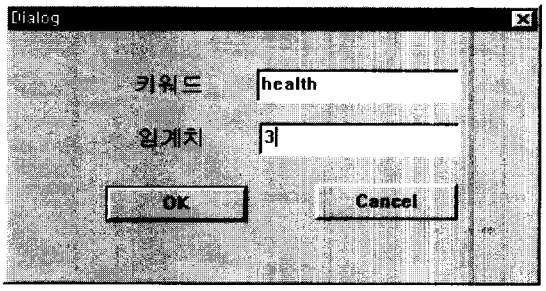
$$\text{정확율}(\text{Pr}ecision) = \frac{\text{나온 문서 중에 질의에 관련있는 문서 수}}{\text{질의에 대해 나온 문서 수}}$$

$$\text{재현율}(Recall) = \frac{\text{나온 문서중 질의에 관련 있는 문서 수}}{\text{질의에 관련 있는 문서의 총수}}$$

여기에서 '관련 있는'의 기준은 문서에 그 질의어가 나오는 횟수로 하였다. 즉, 질의어가 문서에 몇 번 이상 나오는가를 임계치로 두어 그 임계치를 넘어서면 '관련 있는' 문서로 하여 실험을 하였다.

〈표 2〉 추출된 키워드 평가

임계치	WebGraph		새로 제안한 모델	
	Precision	Recall	Precision	Recall
1	93.3	28.2	100	28.2
2	90.8	31.8	94.7	31.8
3	85.4	33.8	89.4	33.8
4	80.9	34.9	85.5	34.9
5	72.7	36.2	76.3	36.2
6	65.5	37.6	68.4	37.6
7	53.2	38.8	55.2	38.8
8	49.7	39.7	51.4	39.7
9	43.4	40.3	48.9	40.3
10	40.1	41.5	44.7	41.5



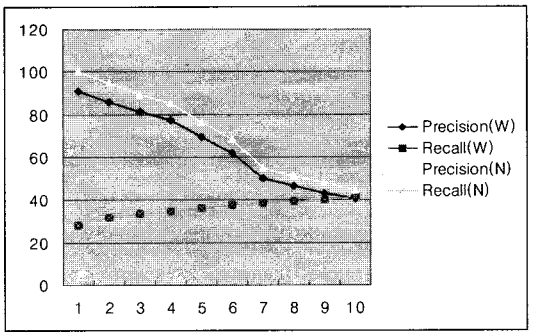
〈그림 15〉 질의 처리부 실행 예

이 실험을 위하여 '질의 처리부'란 프로그램을 따로 만들었다. 이 프로그램은 질의어와 임계치를 입력하면 그 질의어에 대한 Performance(정확율, 재현율)가 측정되는 프로그램이다. 그림 19는 질의 처리부의 실행 화면이다. 질의는 의료에 관련된 용어를 사용 하였으며, 각 질의에 대해 임계치를 1부터 10까지 하여 실험하였다.

4.3.1 결과 및 평가

임계치를 1로 정했을 경우 새로 제안한 모델의 경우 100%가 나왔다. 이 것은 키워드를 추출할때 문서상에 한번이라도 나와야 키워드로 채택되기 때문이다. 그에

반해 기존의 웹 그래프는 94%가 나왔다. 이는 문서상에 단 한번도 나오지 않은 단어가 키워드로 채택되었음을 보여준다. 즉 6%정도가 문서와 전혀 상관없는 단어가 키워드가 된 것으로 Anchor Text의 단어들이 모두 키워드가 될 수 없음을 보여준다. 임계치가 1인 경우는 패턴매치 검색엔진에서도 많이 쓰이는 방법으로, 보통 검색엔진에 많이 쓰이는 방법이다.

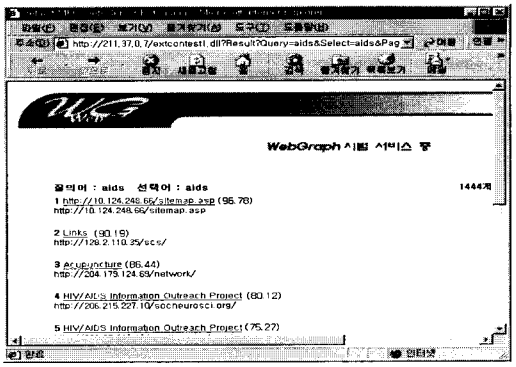


〈그림 16〉 추출된 키워드 평가의 그래프

전체적으로 보았을 때 기존의 웹 그래프 보다 정확율이 현전히 높아졌음을 보여준다. 임계치가 높아지면 질수록 차이가 점점 좁혀지긴 하지만 꾸준히 기존의 웹 그래프보다 높은 정확율을 가지고 있다. 이 것은 Anchor Text에서 추출한 단어 중 TFIDF로 걸러진 단어들만 키워드가 되기 때문에, 무작위로 Anchor Text 전체의 단어가 키워드로 된 방법보다는 좋은 정확성을 갖게 된 것이다. 이에 반하여 재현율은 변화가 없음을 보여주고 있다. 이 것은 정확율이 올라갔지만, 그 것은 정확성이 있는 문서를 더 보여준 것이 아니라 정확하지 않은 것을 제거 해준 것이기 때문에 정확율이 변화가 있다고 하더라도 재현율에는 영향을 안 주었음을 보여준다. 결론적으로 정확율은 올라가고, 재현율은 그대로이므로 시스템의 성능이 향상되었다고 할 수 있다.

4.4 Ranking

질의어에 대한 해당 문서를 찾아서 사용자에게 보여주고자 할 때 그 해당 문서들이 전부 관련이 있다고 하더라도, 관련 정도에 따라 보여주지 않으면, 해당문서가 너무 많아서 사용자가 전부 확인할 수 없을 시, 서로 관련정도가 없는 문서들만 볼 수 있다. 그렇기 때문에 관련 정도가 큰 문서 순으로 보여주는 것이 좋다. 본 논문에서는 관련정도를 TFIDF 값으로 하여 이를 웹 그래프에 적용 시켰다.



〈그림 17〉 웹그래프 Ranking

4.5 기존 방법과의 비교

기존 방법과의 비교는 패턴 매칭 방법의 대표적인 검색엔진이라 할 수 있는 Altavista와 개념 기반 검색엔진인 웹 그래프와 비교하였다.

(표 3) 기존 방법과의 비교

	Altavista	WebGraph	새로 제안한 모델
Automatic Indexing	패턴매칭		TFIDF
Ranking	가능	불가능	가능
웹의 개념화	불가능	가능	가능
visualization	불가능	가능	가능

기존의 웹 그래프는 자동 인덱싱 방법을 Anchor Text의 모든 단어들에 대해서 해당 문서를 연결 시켰으며, 새로운 모델은 TFIDF를 이용하여 자동인덱스화 하였으며, 이는 정확률을 높게 하였다. 또한 그 TFIDF 가중치를 이용하여, Ranking화가 가능하게 하였다.

본 논문의 가장 큰 장점으로는 시소러스가 없이 개념을 만든다는 것이다. 시소러스는 구축하기 위해서 많은 시간이 필요하며, 저장을 위한 방법과 공간이 필

(표 4) Anchor Text 내용과 비교

	CE Pro	SemioMap	본시스템
개념구축	. 시소러스 이용	. 시소러스 이용	. 하이퍼링크 이용
특징	. 질의 확장 . 영역지식제공 . 외부 검색엔진	. 영역 지식제공	. 질의 확장 . 영역 지식 제공 . 자체 검색 엔진
결과물	. 2D 그래프	. 3D 그래프 . 관련된 문서	. 2D 그래프 . 결과 웹문서 리스트
장점	. 특정분야의 개념만 구축가능 . 자유로운 질의어 확장	. 자유로운 그래프 탐색	. 시소러스 필요 없이 개념 구축 . 개념간의 관련정도 표시 가능 . 본문의 내용을 저장하지 않아도 핵심어 추출가능 . DB절약, 속도 빠름
단점	. 범용적 사용의 한계 . 시소러스 구축을 위한 프로세싱 필요	. 범용적 사용의 한계 . 시소러스 구축을 위한 프로세싱 필요	. 반드시 하이퍼링크가 존재해야함 . 중복의미의 단어의 개념으로의 추출 . 다단계의 그래프 탐색불가 . 두단계만의 질의어확장

요하다. 또한 인터넷과 같은 급속히 발전하는 분야에서 새로 생기는 모든 단어들을 신속히 적용하는데 많은 어려움이 있다. 그러나 시소러스를 가지면 오류가 없는 정교한 개념을 추출할 수 있다. 검색 시스템과 같은 대용량의 범용 서비스와 같은 경우 시소러스를 만드는 일은 막대한 시간과 노력이 요하게 되고, 만든 시소러스를 적용하기도 어렵다. 그 이유는 모든 분야의 단어에 대하여 시소러스를 만들어야 하기 때문이다. 따라서 범용 적인 검색시스템에서 개념을 적용하기 위한 시소러스의 구축은 자동적이거나 다른 방법으로 개념을 추출할 수 있어야 한다. 본 논문에서 제안하는 방법은 하이퍼링크가 없는 문서는 개념이나 핵심어를 추출할 수 없기 때문에 반드시 인터넷상의 웹 문서이어야 한다는 제약이 있으며 현재는 한글의 형태소 분석 문제로 한글 정보에 대한 처리를 못한다.

5. 결론

정보검색 결과의 정확도나 만족도를 높이기 위하여 새로운 검색방법들이 대두되고 있다. 단순한 패턴 매치에 의한 방법은 한계에 도달하였다. 웹 그래프는 Anchor Text에서 키워드를 추출하여 이를 개념그래프로 나타냄으로써 정확도나 만족도를 높이려고 한 시스템이다. 하이퍼링크를 이용한 키워드 추출방법은 사람이 작성한 요약정보를 이용함으로써 간단히 키워드를 추출할 수 있는 장점을 가지고 있으며, 직관적으로는 웹 문서의 키워드를 비교적 정확히 찾을 수 있다. 그러나 Anchor Text의 단어들이 그 Anchor Text를 갖고 있는 하이퍼링크가 가리키고 있는 문서의 키워드로써 적합한지에 대해서 검증된바 없으며, 또한 그 단어들은 실제 문서에 있는 단어들이 아니기 때문에 그대로 키워드로 채택이 되어지면 문제가 있다. 본 논문에서는 먼저 Anchor Text의 단어들이 키워드로 적합한지 TFIDF를 이용하여 테스트하였다. 결과는 가중치가 높아서 키워드로 적합한 단어가 있었는가 하면, 아예 문서에 나오지도 않는 단어도 있어서 키워드로 적합하지 않은 단어도 있었다. 이를 이용하여 새로운 키워드 추출 방법을 제시하였다. 위 실험에서 적합하지 않은 키워드를 제거함으로써 새로운 키워드를 만들어 내고 TFIDF값을 각 키워드의 가중치로 이용하여 Ranking이 가능하게 하였다. 이렇게 추출된 키워드는 기존의 방법보다 정확도가 높았다. 그러나 이 방법은 문서상에 단어의 빈도수에 따라 키워드를 추출하는 방법으로써 요약정보인 Anchor Text에는 적용하기가 어려울 때가 있다. Anchor Text는 문서의 요약정보이기 때문에 그 문서를 요약한 단어가 문서에 나오지 않을 수도 있기 때문이다. 향후 연구과제로는 Anchor Text에서 키워드 추출하는데 있어서 기존의 Full Text에서 키워드를 추출한 방법이 아니라 Anchor Text의 특성을 잘 이용한 키워드 추출 방법이 필요하다하겠다.

참고 문헌

[1] Sergy Brin, Lawrence Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine", In Proceeding of the 7th International World Wide Web Conference(WWW7), 1998

[2] Armstrong, R., Freitag, D., Joachims, T., Michell,T., "WebWatcher: A Learning Apprentice for the World Wide Web", AAAI 1195 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, Stanford, March 1995.

[3] Gerald Kowalski, *"Information Retrieval Systems Theory and Implementation"*, Kluwer Academic Publishers, 1997.

[4] Kurt D. Bollacker, Steve Lawence, and C. Lee Giles, "CiteSeer: An Automous Web Agent for Automatic Retrieval and Identification of Interesting Publications", In Agents '98, 1998.

[5] Salton G., A.Wong, and C.S. Yang, "A Vector Space Model for Automatic Indexing", Communications of the ACM, 18(11), 1975, pp. 613-620

[6] Salton, G., And Buckley, C. "Term weighting approaches in automatic text retrieval", Tech Report 87-881 Dept. of Computer Science, Cornell University, 1987.

[7] G. Salton. "Developments in automatic text retrieval." science, vol. 253, pp 974-979, 1991

[8] William. B. Frakes, Ricardo. Baeza/Yates *"Information Retrieval DataStructure and Algorithms"*, Prantice Hall PTR, Upper Saddle River, New Jersey 07458, 1992.