구문적 유사도와 의미적 유사도를 이용한 하이퍼텍스트 자동생성에 관한 연구*

°김문석, 남세진, 신동욱
충남대학교 컴퓨터공학과 정보검색 연구실

The Study of Automatic Hypertext Generation using the Syntactic and Semantic Similarity

°MunSeok Kim, SeJin Nam, DongWook Shin
Information Retrieval Lab, Dept. of Computer Engineering , Chungnam National University

요 약

본 논문에서는 일반문서를 대상으로 하여 그 문서를 하이퍼텍스트(hypertext)로 자동변환하는 기법을 제안하고자 한다. 자동변환의 과정은 대상 문서에서 키워드(keyword)의 인식, 문서를 노드(node) 단위로 분리, 키워드로부터 노드로의 링크(link) 생성의 3 단계로 이루어 진다. 기존의 연구에서는 문서에서 노드를 분리하는데 구문적 유사도만을 이용하는데, 본 논문에서는 양질의 하이퍼텍스트를 생성하기 위하여 구문적 유사도(syntactic similarity)뿐만 아니라 의미적 유사도(semantic similarity)를 사용한다. 구문적 유사도는 tf*idf 와 벡터 곱(vector product)을 이용하고, 의미적 유사도는 시소러스(thesaurus)와 부분부합(partial match)을 이용하여 계산되어 진다. 또 링크 생성시 잘못된 링크의 생성을 막기 위하여 시소러스를 이용하여 시소러스에 존재하는 용어에 한해서 링크를 생성한다.

1. 서론

최근들어 생성되는 문서들이 하이퍼텍스트의 형태로 만들어지기 시작하고, 이런 작업들이 단지 전문가들에의해 수행되기에는 매우 어렵기 때문에 하이퍼텍스트 자동생성에 대한 연구가 관심을 끌고 있다. 하이퍼텍스트 자동생성은 80 년대 말부터 G. Salton [7][8]등에 의해 연구되기 시작하였는데 이러한 연구는 대부분 기존에 정보검색 분야에서 널리 사용되는 벡터 공간 모델(Vector Space Model)을 하이퍼텍스트 생성에 적용하는 것들이다. 그러나 하이퍼텍스트를 생성하기 위해서는 주어진 문서의 구조및 의미를 잘 이해해야 하는데 벡터 공간 모델은 단순히 구문적 유사도만을 가지고 하이퍼텍스트를 생성하므로 좋은 하이퍼텍스트를 만드는데 한계가 있다.

즉 구조화가 잘된 하이퍼텍스트를 생성하기 위하여 문서내용에 대한 구문적 관련도뿐만 아니라 의미적 관련도도 연구되어야 한다. 이는 하이퍼텍스트를 생성하는 일이 매우 전문적인 작업을 포함하기 때문인데 이런 작업에는 다음과 같은 것들이 포함된다.

- 어느 수준이상의 문서의 내용에 대한 이해
- 문서내에서 중요한 키워드를 발견하는 일
- 문서를 노드단위로 나누는일
- 그 각각의 노드에 대해 주제를 파악
- 서로 관련된 키워드와 노드사이의 링크의 생성

본 논문은 양질의 하이퍼텍스트를 생성하는 방법과 효율적인 방법으로 구문적 유사도와 의미적 유사도의 개념을 조합하는

* 본 논문은 충남대학교 부설 소프트웨어 연구센터에서 수행하고 있는 '하이퍼텍스트 자동생성에 관한 연구'의 중간결과를 기술한 것입니다.

방법에 대해 기술한다. 구문적 유사도의 개념은 tf*idf 에 의한 가중치 부여방법과 벡터 곱(vector product)에 기반을 두는 반면 의미적 유사도의 개념은 시소러스와의 부분부합에 기반을 두고 있다. 본 논문에서는 처리 대상이 되는 문서에 대해 그 문서의 도메인(Domain)에 대한 충분한 지식을 표현하고 있는 시소러스가 있다고 가정한다.

2. 관련 연구

수년 전부터 G. Salton 및 J. Allan 등에 의해 하이퍼텍스트 자동 생성에 관한 연구가 시작되었고 지금까지 여러개의 결과가 발표되었다[1][2][3][7][8]. G. Salton 과 C. Buckley[7]는 유사한 내용을 가진 문서의 각 부분들을 서로 관련시키는 내용 링크(content link)를 인식하는 방법에 대해 제안하였다. 이 방법은 처음에는 문서들 사이에 전체 문서 분석(global text analysis)을 적용하고 문서들 사이의 높은 유사도를 나타내는 문서들을 찾기위해 각각의 문장을 사용하였다. 문서들의 각 부분들의 관련도는 문서간의 벡터 내적(inner vector product)에 의해 계산되었는데 이것은 각각의 벡터가 tf*idf 에 의한 계산으로 얻어진 용어 가중치(term weight)로 구성된다.

J. Allan[3]은 문서들 사이의 링크의 유형을 결정하는 기법을 제안했다. 이 기법은 벡터 공간 모델과 Salton 의 gloval and local text similarity[7]에 기반을 두며 여기에 노드들을 연결시키는 링크의 패턴에 따라 노드들 사이의 링크의 유형이 어떻게 분류되는가의 기준을 제시하였다. 이 접근 방법은 문서가 용어 가중치의 벡터로 표현되었을때, 그들의 의미적 유사도에 상관없이 벡터 내적을 사용하여 문서들 사이의 유사도를 사용함으로써 유사도 계산에 단지 구문적 유사도를 이용하였다.

M. Agosti[1][2]는 3 단계 하이퍼텍스트 검색 모델을 제안하였다. 각 단계는 다음과 같다.

- document level - 격자구조(lattice structure)로 서로 연결된 문서로 구성.
- index term level - 용어들 사이의 의미적 관련성을 나타냄.
- concept level - 개념들 사이의 관련성을 나타냄.

문서 집합이 주어졌을때 이 모델은 몇가지 기준에 따라서 여러개의 링크 유형을 만드는데, 링크의 유형은 다음과 같다.

- D-D link - 문서들 사이의 유사도
- T-T link - 색인 용어들 사이의 동의성과 연속성
- D-T link - 문서들과 색인 용어들 사이의 타당성
- T-C link - 색인 용어들과 개념들 사이의 의미

이 방법은 구문적 방법으로 문서들 사이의 유사도를 계산하고 의미적 방법으로 색인 용어들과 개념들 사이의 링크를 생성하면서 어느 정도는 구문적 유사도와 의미적 유사도의 개념을 조합하는 방식을 사용한다. 이 방법은 문서들 사이의 링크를 생성하는 과정에서 G. Salton 과 J. Allan 이 사용한 구문적 방법과는 약간의 차이를 보이고 있고, 의미적 접근이 용어들과 개념들 사이의 링크를 생성하는데 사용되고 있다.

3. 제안된 기법

앞에서 설명한 대로 구조화가 잘된 하이퍼텍스트를 생성하기 위해서 구문적 방법에 의한 유사도 만을 이용해서는 충분하지 않을 뿐만 아니라 전문가들이 하는 방법을 흉내내기도 힘들기 때문에 본 논문에서는 구문적 방법뿐만 아니라 의미적 방법을 함께 고려하는 방법을 제안한다. 의미에 대한 고려는 두가지 방법에 의해 이루어질 수 있다. 첫째는 자연어 처리 기술을 적용함으로써 문서 내용이 이해될 수 있으며, 둘째는 시소러스에 저장되어 있는 의미들 사이의 관련성을 이용하는 것이다. 첫번째 방법은 현재의 기술로는 매우 어렵기 때문에 본 논문에서는 의미를 처리하기 위하여 두번째 방법을 이용하고자 한다.

위의 내용과 더불어 본 논문에서는 문서를 하이퍼텍스트 형태로 만드는데 다음과 같은 기본 가정을 한다.

1. 문서의 기본 단위로서 문장(sentence)을 사용한다.
2. 하이퍼텍스트 생성에서 정확성(precision)은 매우 중요한 요소이기 때문에 잘못된 링크를 만드는 것 보다는 차라리 링크를 만들지 않는 것이 더 낫다.
3. 자금까지 링크의 유형에 대한 표준이 없기 때문에 World Wide Web 에서 널리 적용되고 있는 implicit 링크[3]만을 사용한다.

본 논문에서는 링크의 시작점으로 사용되는 키워드를 시소러스에 존재하는 용어만으로 제한한다. 그런데 시소러스를 기반으로 한 접근 방식에서는 시소러스의 용어가 한정되어 있으므로 자주 불철저성(non-exhaustivity)[4] 문제가 발생한다. 불철저성 문제란 문서내에 존재하는 용어들중에 일부분만이 시소러스 용어들과 부합(matching)되고 나머지 대부분의 용어들은 관계없는 것으로 취급되어 시소러스를 사용하는 의미가 없어지는 것을 의미한다. 이 문제를 해결하기 위해서 문서내

의 용어가 시소러스 내의 용어와 어느 정도 부합되는가를 계산하고, 만약 그 값이 주어진 경계값(threshold)보다 크다면 두 용어들을 같은 것으로 고려하는 부분부합(partial match)[11]을 적용한다.

문서를 노드단위로 분리할 때 본 논문에서는 TextTiling[6] 방법을 사용한다. TextTiling 이란 cosine 계산 방법에 의해 블럭들 사이의 유사도를 계산하고 그 값에 따라 두 블럭을 하나의 블럭으로 합칠것인지 그렇지 않은지를 결정하여 서로 관련된 블럭단위로 나누는 것이다. 블럭 단위로 나누는 과정에서 기존의 TextTiling 방법에서는 단지 구문적 유사도를 사용했지만 본 논문에서는 구문적 유사도와 의미적 유사도를 함께 사용한다.

노드와 키워드가 구별된 후에, 키워드에서 노드로의 implicit type 의 링크가 생성된다. 이때 키워드와 노드사이의 유사도는 노드내에 존재하는 키워드들의 가중치의 평균값으로 계산된 값 (구문적 유사도)과 키워드가 시소러스내에 있는 노드의 용어와 어느정도 인접해 있는가의 계산에 의한 값 (의미적 유사도)을 이용하여 구해진다. 이때 만약 두개 이상의 노드가 어떤 용어와 매우 밀접하게 관련이 있다면 그 용어로부터 대상노드 각각으로 링크가 생성될 것이다. 즉, 하나의 키워드에 여러개의 노드가 연결될 수 있는데 이럴 경우에 사용자는 하이퍼텍스트를 항해(navigation)할 때 키워드에 연결된 노드들의 리스트를 보고 그 중에서 원하는 노드로 갈 수 있다. 따라서 사용자는 좀더 자유롭게 하이퍼텍스트에서 자신이 원하는 부분으로 항해할 수 있을 것이다.

4. 색인 후보어(Indexing Candidates)와 키워드(Keywords)

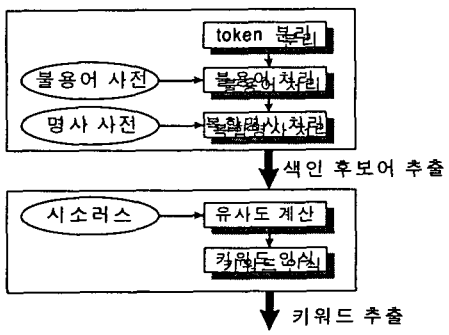

그림 1 색인 후보어의 추출과 키워드의 추출

본 논문에서 제안하는 하이퍼텍스트 자동생성 기법은 3 단계로 구성되어 있다. 첫번째는 링크의 시작점이 되는 문서내의 키워드를 추출해 내는 것이고, 두번째는 문서를 특정 주제를 다루는 노드단위로 나누는 것이고, 세번째는 키워드로부터 노드로의 링크를 생성하는 것이다. 이러한 과정들은 문서내에 존재하는 색인 후보어(indexing candidates)를 구별해내는 것으로부터 시작된다. 색인 후보어는 어떻게 문서를 노드단위로 분리하고 키워드를 노드로 연결하는데에 대한 기본 정보를 제공한다. 그림 1 은 색인 후보어를 추출하고 색인 후보어 중에서 키워드를 추출하는 과정을 나타내고 있다.

4.1 색인 후보어 추출(Extracting Indexing Candidates)

색인 후보어 추출은 문서의 내용을 나타내는 용어들을 검색해서 추출해 내는 단계이다. 본 논문에서는 색인 후보어로서 단순 명사와 복합명사를 포함하는 명사만을 대상으로 한다. 그 이유는 명사는 보다 정확하게 문서의 내용을 나타내고, 또한 복잡한 형태소 분석없이 명사를 추출하는 것이 용이하기 때문이다[10]. 복합명사의 경우 저자에 따라 명사 사이에 공백을 넣는 경우와 넣지 않는 경우가 있기 때문에 문서내에서 복합명사를 구별하는 것은 쉽지 않다. 예를 들어 "정보검색"과 "정보 검색"이 혼용되어 사용될 수 있는데 이 두가지 경우를 같게 취급하는 방법이 한글 처리에서 매우 중요하다. 특히 정보검색 분야에서 한글의 경우 복합명사가 많이 사용되고 대부분의 경우 중요한 의미를 내포하고 있기 때문에 이런 복합명사를 정확하게 처리하는 것이 정확성(precision) 향상에 많은 영향을 미칠 수 있다. 복합명사를 처리하는데 좋은 해결 방법중 한가지는 위 두가지 경우를 한가지 경우로 정규화(normalization)하는 것이다. 즉, 각각의 기본단어 사이에 존재하는 공백을 모두 제거하거나 공백을 삽입하는 것이다.

이 방법을 수행하기 위해서는 복합명사의 경계를 정확하게 구별해야 하는 문제가 있다. 본 논문에서는 형태학적 분석을 사용하지 않고 통계적 데이타 분석을 이용하여 이 문제를 처리한다.

즉, 이전의 연구에서[12] 한국어의 경우 복합명사의 96.4% 정도가 다음과 같은 형태를 취한다는 것이 발표되었다.

type1: 명사 + 명사 (58.4%)

type2: 명사 +'의'+ 명사 (22.6%)

type3: 명사 + 명사 + 명사 (7.0%)

type4: 명사 +'의'+ 명사 + 명사(3.1%)

type5: 명사 + 명사 +'의'+ 명사(5.1%)

본 논문에서는 복합명사를 처리하기 위하여 위의 연구에 기반을 둔다. 이 접근 방법은 한글의 경우 대부분의 복합명사 다음에 조사가 따라오기 때문에 복합명사의 끝을 알아내는데 보다 쉽다. 따라서 명사와 조사의 두 사전을 사용함으로써 복합명사의 끝을 정확하게 알아낼 수 있다. 복합명사의 끝을 알아낸 후에 그것으로부터 명사를 얻어내고 명사 사이의 공백을 제거하는 방식으로 복합명사를 얻게 된다.

즉 위의 5 가지 유형중에서 하나가 문서내에서 발견된다면 원래 문서에서 그들이 나타난 순서를 유지하면서 각각의 단순명사를 서로 연결하여 하나의 새로운 복합명사를 생성하게 된다. 즉, 문서내에서 혼자 쓰여진 단순명사 뿐만 아니라 새롭게 생성된 복합명사가 색인 후보어가 된다. 특히 세번째, 네번째, 다섯번째 유형의 경우 세개의 새로운 복합명사가 생성되는데 첫째는 처음과 두번째 명사를 연결하는 경우, 둘째는 두번째와 세번째 명사를 서로 연결하는 경우, 셋째는 세개를 서로 연결하는 경우이다. 예를 들어 "정보 검색 시스템"의 경우 "정보검색", "검색시스템", "정보검색시스템"의 3 개의 새로운 복합명사가 생성되어 원래 문서에서 존재하던 각각의 단순명사("정보", "검색", "시스템")까지 합해 모두 여섯개의 색인 후보어가 생성되게 된다.

4.2 키워드의 인식(Recognizing Keywords)

본 논문에서 키워드는 하이퍼텍스트에 존재하는 링크의 시작점을 의미하기 때문에 일반적인 정보검색 시스템에서보다 더 중요한 역할을 하게 된다. 앞에서 설명하였듯이 본 논문에서는 변형된 형태의 부분부합 기술[11]을 이용하여, 색인 후보어에서 키워드를 추출하기 위해서 시소러스를 사용한다. 부분부합 방법은 문서에서 사용된 용어를 시소러스에 존재하는 유사한 용어와 부합시킬수 있고 색인어의 불철저성(indexing non-exhaustivity) 문제[4]를 해결하는데 큰 도움을 준다.

색인 후보어와 시소러스 사이의 부분부합의 정도는 다음과 같이 계산되어 진다.

$$D = \frac{C_n^2}{T_n \times I_n}$$

- T_n : 시소러스에 존재하는 단일명사의 갯수.
- I_n : 색인 후보어에 존재하는 단일명사의 갯수.
- C_n : 양쪽 용어 모두에 존재하는 단일명사의 갯수.

이 수식에서 알 수 있듯이 두 용어들 사이에 계산된 값이 높을수록 두 용어는 서로 밀접한 관련이 있음을 나타낸다. 위의 식을 기반으로하여, 만약 D 값이 높으면 색인 후보어를 시소러스 용어 (이 이후부터는 키워드)와 관련자을 수 있다. 본 논문에서는 경계값을 0.6 으로 사용하였다. 만약 경계값보다 적으면 후보어는 시소러스 용어와 관련이 없는 것으로 판단하며, 의미적 유사도 계산에서 제외된다. 시소러스 용어와 연관된 색인 후보어만이 키워드와 링크의 시작점으로 취급된다. 이때 만약 경계값을 넘으면서 색인 후보어와 매치된 시소러스 용어가 두개 이상이 있다면 시소러스에 있는 중의어(하나의 용어가 두개 이상의 의미를 나타내는 용어)를 나타내는데 이때 문서내의 용어는 이 두개 이상의 시소러스 용어에 모두 부분 부합되며 부분 부합 정도가 같은 경우가 발생한다.

이런 중의어 문제가 발생했을 경우 본 논문에서는 다음과 같은 방법으로 중의어 문제를 해결하고자 한다. 중의어 발생이 확인되면 문서내의 중의어와 매치되는 용어 주변의 키워드를 추출한다. 추출된 키워드들은 시소러스에 매치되는 용어이므로 이 용어들과 중의어에 해당하는 시소러스 용어들과의 거리(Distance)가 계산되어 질 수 있다(거리 계산은 5 절에서 기술된 방법을 사용한다.) 이렇게 주변 키워드들과 중의어들과의 계산을 통한 거리의 합이 가장적은 중의어에 해당되는 용어들 중의 하나를 선택함으로써 중의어 문제를 해결하면서 문서내의 키워드를 추출하고자 한다.

5. 노드 생성(Node Creation)

노드 생성 단계에서는 문서를 특정 주제를 다루고 있는 노드 단위로 나누게 된다. 이 단계에서 사용하는 방법이 TextTiling 인데, TextTiling 이란 문서를 서로 연관된 여러개의 부분들로 나누는 방법이다. Hearst 는 문서의 기본단위(이후부터는 블럭)로 3-5 개의 연속된 문장을 사용하였으며 연속된 블럭들 사이의 관련도를 tf*idf 를 기반으로한 구문적 유사도를 이용하여 계산하였다. 모든 이웃한 블럭들 사이의 유사도를 계산한 후 그 결과를 이용하여 유사도가 적은 부분을 노드사이의 경계로 사용하여 노드를 생성하는 방법을 사용하였다. 본 논문

에서는 문서의 기본단위를 문장(sentence)으로 보고 이웃한 각 기본단위 사이의 유사도 계산에 구문적 유사도와 의미적 유사도를 이용한다.

이웃한 두 블럭 사이의 관련도는 각 블럭에 있는 키워드가 시소러스 상에서 서로 이웃하는 정도와 각 블럭의 색인 후보어들이 서로 중복되는 정도의 평균으로서 계산되어진다. 첫번째 부분은 두 블럭이 다른 용어를 사용하여 같은 내용을 나타낼때의 상호 관련 정도(의미적 유사도)를 나타내고 두번째는 두 블럭이 같은 용어를 사용하여 같은 내용를 나타낼때의 상호 관련 정도(구문적 유사도)를 나타낸다.

두 블럭간의 상호 관련도를 계산하기 위한 식은 다음과 같다.

$$SIM(b1,b2) = \frac{Sem(b1,b2) + Syn(b1,b2)}{2}$$

Sem(b1, b2)는 b1 과 b2 사이의 의미적 유사도이고 Syn(b1, b2)는 b1 과 b2 사이의 구문적 유사도이다. 이때 Sem(b1, b2)와 Syn(b1, b2)는 각각 다음과 같이 계산되어 진다.

$$Sem(b1,b2) = \sum_{t_i \in b1, t_j \in b2} \frac{D_{t_i} \times D_{t_j}}{\text{Distance} + 1}$$

t_i 와 t_j 는 b1 과 b2 에 존재하는 키워드이고 D_{t_i} 는 t_i 의 부분부합 정도이고, Distance 는 시소러스에 상에서의 t_i 와 t_j 사이의 거리이다.

$$Syn(b1,b2) = \sum_{t=1}^{n} W_{t,b1} \times W_{t,b2}$$

t 는 두 블럭 모두에 나타나는 용어이고, $W_{t,bi}$ 는 블럭 bi 에서 t 에 주어진 tf*idf 로 계산된 가중치이다.

두개의 용어 t_i 와 t_j 사이의 거리(Distance)는 t_i 와 t_j 가 시소러스 상에서 이루는 경로(path)상에 존재하는 노드(node)의 갯수에서 1 이 감소된 값으로 정의된다. 이때 t_i 와 t_j 로 이루어진 경로상에 시소러스의 최상위 노드(root node)가 존재하지 않아야 된다. 만약 존재하면 t_i 와 t_j 사이의 거리(Distance)는 무한대 값으로 정의된다. 왜냐하면 많은 경우에 있어서 루트노드를 포함하는 경로를 가진 용어들 간에는 거의 관계가 없기 때문이다.

6. 링크 생성(Link Generation)

링크는 키워드 추출단계에서 구해진 용어로부터 시작하며 그 용어와 밀접하게 관련된 노드에서 끝나게 된다. 노드 생성 과정에서처럼 링크 생성시에도 구문적 유사도와 의미적 유사도를 사용하고 있다. 링크 생성시에 링크 키워드와 노드사이의 관련도는 (1)링크 키워드와 시소러스 용어와의 부분부합 정도와 (2)링크 키워드와 대상 노드상의 용어와의 부분부합 정도(의미적 유사도)와 (3)대상노드에서의 각 용어들의 가중치(구문적 유사도)를 이용해서 구해진다. 용어 t 와 노드 N 의 관련도를 계산하는 식은 다음과 같이 정의된다.

$$Sim(t,N) = \sum_{t_i \in N} \left(\frac{D_t \times D_{t_i}}{\text{Distance} + 1} \right) + W_t$$

D_t 는 링크 용어의 부분부합 정도이고, D_{t_i} 는 노드내에 있는 키워드 t_i 의 부분부합 정도이고, Distance 는 5 절에 정의된 것과 같고 W_t 는 용어 t 와 노드 N 과의 구문적 유사도이다.

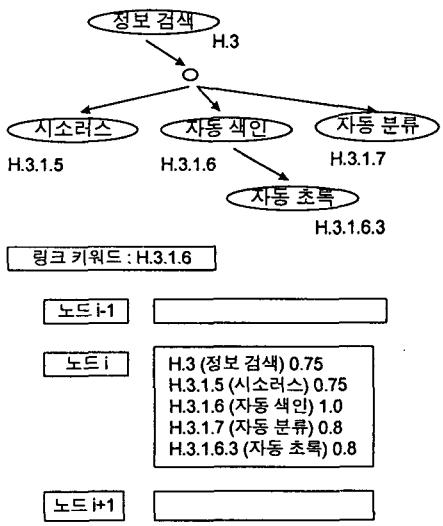

그림 2 노드와 시소러스 구조의 예

예를 들어 그림 2 에서와 같이 어떤 노드가 키워드 H.3, H.3.1.5, H.3.1.6, H.3.1.7, H.3.1.6.3 을 가지고 있으며, 그 각각의 키워드가 시소러스 용어와 0.75, 0.75, 1.0, 0.8, 0.8 의 관련도를 가지고 있고, 링크 키워드인 "자동 색인"의 부분부합 정도가 1 이라고 하자. 또한 "자동 색인" 용어의 가중치가 0.5 라고 하면 링크 키워드 "자동 색인"과 노드 i 와의 관련도는 다음과 같이 계산되어 진다.

$$sim(t,N) = \frac{1\times0.75}{3} + \frac{1\times0.75}{3} + \frac{1\times1}{1} + \frac{1\times0.8}{3} + \frac{1\times0.8}{2} + 0.5 = 2.67$$

만약 계산된 값이 주어진 경계값(threshold)보다 더 큰값을 가

지면 링크 키워드로부터 노드사이에 implicit 링크가 생성되고 그렇지 않으면 링크는 생성되지 않는다. 이때 관련도가 threshold 값을 넘으면 하나의 링크 키워드에서 여러개의 노드로 링키가 만들어 질 수도 있다. 이렇게 생성된 링크는 사용자가 보기를 원하는 노드를 선택할 수 있기 때문에 World Wide Web 에서 존재하는 현재 링크 스타일보다 사용자에게 더 많은 자유를 준다. 그리고 사용자 보기를 원하지 않는 노드는 방문할 필요가 없기 때문에 사용자는 좀더 효율적으로 하이퍼 텍스트를 브라우징(browsing)할 수 있다.

7. 결론

최근들어 새로 생성되는 문서들이 하이퍼텍스트의 형태로 만들어지기 시작하고, 이런 작업들이 단지 전문가들에 의해 수행되기에는 매우 어렵기 때문에 하이퍼텍스트 자동생성에 대한 연구가 많은 관심을 얻고 있다. 이에 따라 최근 하이퍼텍스트 자동생성에 관한 연구가 여러곳에서 진행되어왔다.

좋은 하이퍼 텍스트를 생성하기 위해 더 구체적이고 주의깊게 디자인된 이론들이 연구되어야 하는데 이는 정확율(Precision)이 재현율(Recall)보다 더 중요하고, 키워드들이링크 생성시 매우 중요한 역할을 차기 하기 때문이다.

본 논문에서는 양질의 하이퍼텍스트를 생성하기 위해서 벡터 모델(vector model)에서 사용하는 전통적인 유사도 계산 방법과 특정 도메인의 지식을 잘 표현하고 있는 시소러스를 사용하였다. 하이퍼텍스트 생성시 구문적 방법과 의미적 방법을 사용하여 내용들 간의 유사도를 평가 함으로써 본 논문에서 제안하는 방법이 전문가들이 수행하는 일을 상당히 흉내낼 수 있다고 생각된다. 본 논문에서는 벡터 내적(vector inner product) 계산시에 구문적 유사도를 반영하였고, 시소러스를 의미적 유사도를 계산하기 위해 사용하였다. 단, 시소러스는 특정 도메인에서 사용가능하기에 충분한 지식을 포함하고 있다는 가정을 하였다.

참고문헌

[1] M. Agosti, R.Colotti, and G. Gradenigo (1991), "A Two-Level Hypertext Retrieval Model for Legal Data", SIGIR '91, pp. 316-325.

[2] M. Agosti, M.Melucci, and F. Crestani (1995), "Automatic authoring and construction of hypermedia for information retrieval", Multimedia system, 3, pp. 15-24.

[3] J. Allan (1995), "Automatic Hypertext Construction", Ph.D. Dissertation, Department of Computer Science, Cornell University.

[4] M. Dillon and A.S. Gary (1983), "FASIT : A Fully Automatic Syntactically Based Indexing Systems", Journal of the American Society for Information Science, 34, pp. 99-108.

[5] M. Hearst and C. Plaunt (1993), "Subtopic structuring for Full-Length Document Access", SIGIR '93, pp. 59-68.

[6] M.Hearst (1993), "TextTiling : A quantitative approach to discourse segmentation", Technical Report 93/24, University of Berkeley.

[7] G. Salton, J.Allan, and C. Buckely (1989), "On the Automatic Generation of Content Links in Hypertext", TR pp. 89-993, Department of Computer Science Cornell University.

[8] G. Salton, J. Allan, and C. Buckely (1992), "Selective Use of Full-Text Database", TR 92-1300, Department of Computer Science, Cornell University.

[9] J. Sammet and A. Ralston (1982), "The new computing reviews classification system - Final version", Communication fo the ACM, 25(1), pp.13-25.

[10] 김문석, 남세진, 신동욱 (1996), "복합명사 통계자료를 이용한 한글 자동색인시스템 개발", 96 년 봄 정보과학회 학술발표 논문집, pp. 931-934

[11] 신동욱, 최기선 (1995), "The Development of an Automatic Indexing System based on a Thesaurus", NLPRS '95, pp. 486-491.

[12] 이창열 외 (1993), "자동 키워드 제작 시스템 설계", 93 년도 5 회 한글및 한국어 정보처리 학술발표 논문집, pp. 71-77.