웹을 대상으로 한 지식지도의 작성

이상협

요 약

본 논문에서는 웹을 대상으로 동적인 지식지도를 작성하기 위한 모델을 제안한다. 제안한 모델은 웹에 존재하는 비정형의 한글 문서들을 대상으로 하여 문서의 주제어를 추출, 문서의 분류, 문서간의 유사도 측정 등의 텍스트마이닝 기법을 활용한 지식지도를 작성하는 방법이다.

1. 서 론

지식지도는 대량의 정보(information) 속에 숨겨진 특별한 형태(type)와 패턴(pattern)을 찾아 그 의미를 파악할 수 있도록 가시적인 형태의 결과를 보여주는 것으로, 작성에 활용되는 데이터베이스, 분석에 사용되는 기법, 결과물의 형태 등에 따라 그 유형이 구분될 수 있다.[1]

지금까지 진행된 대부분의 지식지도 관련 연구는 한정된 범위 안에서의 데이터를 기반으로 특정한 목적을 위한 정적인 지식지도를 작성하는 것을 목적으로 하였으나, 본 연구에서는 웹에 존재하는 모든 한글문서를 대상으로 지속적으로 변화하는 동적인 지식지도를 작성하는 것을 목적으로 한다. 다만, 짧은 기간에 웹상에 존재하는 모든 문서를 분야별로 분류하고 지식지도로 작성하기 위해서는 적지 않은 시간과 비용이 요구되므로, 본 연구에서는 국사 분야 하나의 분야만을 대상으로 하여 지식지도를 작성한다. 그러나 제안된 모델은 특정 분야에만 특화된 것이 아니기 때문에 타 분야에도 큰 무리 없이 적용할 수 있다.

본 논문의 구성은 다음과 같다. 제2장에서는 지식지도를 작성하기 위한 시스템의 구조를 설명한다. 제3장에서는 지식지도의 작성과정과 사용된 알고리즘에 대해 설명한다. 마지막으로 제4장에서는 결론 및 향후과제를 이야기한다.

2. 지식지도 작성 시스템의 구조

본 논문에서 제안하는 지식지도 작성 모델의 구조는 URL수집 에이전트, 웹문서 수집 에이전트, 지식관계 생성 에이전트와 URL서버, Document서버, URL DB, Document DB로 구성된다.

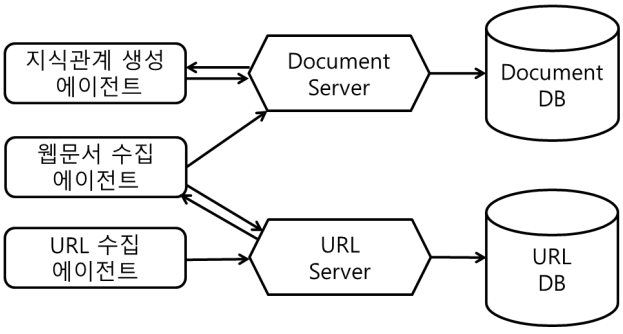

그림 1: 지식지도 작성 시스템의 구조

URL수집 에이전트는 웹문서 수집 에이전트에게 제

공할 URL을 수집하는 역활을 한다. 웹문서 수집 에이전트는 수집된 URL로 부터 해당 웹페이지를 방문하여 국사분야문서이면 주제어를 추출하여 문서정보와 함께 수집한다. 지식관계생성 에이전트는 수집된 웹문서간의 유사도를 측정하여 지식간의 관계를 생성한다. URL서버와 Document서버는 다수의 에이전트로 부터의 DB서버 접근을 조율하는 역할을 한다.

3. 지식지도 작성

3.1 지식지도 작성과정

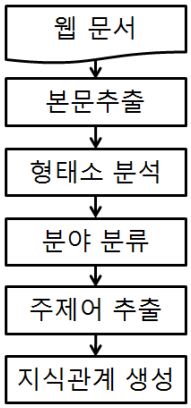

그림 2: 지식지도 작성과정

지식지도 작성의 첫 번째 과정은 소스코드 형태의 웹문서로부터 본문을 추출하는 것이다. 다음 과정으로는 추출된 본문에 대해서 형태소 분석을 실시한다. 다음 과정으로 형태소분석한 결과를 이용하여 문서의 분야를 분류하는데, Naive Bayesian Classifier를 사용하여 이진분류하였다. 분류를 통해서 국사분야에 해당하는 것으로 분류된 문서에 대해서는 주제어 추출을 실시한다. 주제어 추출에는 주성분 분석을 통하여 해당 문서의 주제어를 추출하였다. 이후에는 추출된 주제어로 해당 문서를 그룹화하여 DB에 저장한다. 이때 저장된 주제어가 지식이 되며, 지식관계의 생성은 특정 지식으로 그룹화된 문서들과 다른 지식으로 그룹화된 문서들간의 유사도를 비교하여 관계를 생성한다. 지식관계의 생성에는 Cosine Similarity를 사용하였다. 최종적으로 생성된 지식간의 관계를 시각적으로 표현하여 지식지도를 작성한다.

3.2 지식지도 작성과정의 알고리즘

3.2.1 문서의 분류

특정분야의 지식지도를 작성하기 위해서는 문서를 수집하는 과정에서 해당분야에 속하지 않는 문서들을 필터링할 필요가 있다. 본 논문에서는 국사 분야만을 대상으로 지식지도를 작성하므로, 문서에 대해서 국사분야인지 아닌지 여부를 이진 분류하여 문서를 필터링한다. Naive Bayesian Classifier는 지도학습(Supervised Learning)된 데이터를 기반으로 Bayes의 정리를 이용하여 새로운 문서가 등장하였을 때, 어떠한 분야에 속할 가능성이 높은가를 예측하는 확률적인 방법이다.

$$P(c_j|d) = \frac{P(C_j)P(d|C_j)}{P(d)}$$

여기에서 d는 임의의 웹문서를 의미하고, C_j는 문서의 분야를 의미하는 것으로, 국사분야와 그렇지 않은 분야의 2가지가 존재한다. P(d)는 모든 분야에 대해서 같은 값을 가지므로 학률을 계산하는데 있어 고려하지 않아도 된다. 따라서 $P(C_j)$와 $P(d|C_j)$만 알면 웹문서가 국사분야에 속할 확률과 그렇지 않은 확률을 구할 수 있다.

$$P(C_j) = \frac{n_{c_j}}{n}$$

$P(C_j)$는 모든 학습 문서들의 수(n)와 국사 분야에 속하는 학습 문서들의 수(n_{c_j})의 비율이다. 본 논문에서는 이 값을 0.5로 동등하게 적용하였다. Naive Bayesian Classifier는 $P(d|C_j)$의 계산을 단순하게 하기 위해, 문서 내에 존재하는 모든 키워드들이 서로 독립적인 관계

라고 가정한다. 이러한 가정을 따르면 $P(d|C_j)$는 다음과 같은 식으로 변형될 수 있다.

$$P(d|C_j) = \sum_{i=1}^{n} \log(\frac{w_i + 1}{C_j + 2})$$

여기에서 $\frac{w_i}{C_j}$는 국사분야의 학습문서 C_j에서 키워드 w_i를 포함하고 있는 문서의 비율이다. 만약, 학습데이터에 없는 새로운 키워드가 문서에서 등장하였을 경우에는 0이 곱해지게 된다. 이러한 문제를 해결하기 위해서 분모에 1을 더하고 분자에는 분류를 수행할 클래스의 개수인 2를 더하여 Laplace Smoothing을 적용한다. $P(d|C_j)$에 대한 확률을 구하고자 할 때에는 단순히 $\frac{w_i+1}{C_j+2}$의 값들을 곱하는 것으로 충분하지만, 매우 작은 수들이 연속적으로 곱해질 경우에 산술언더플로우(Arithmetic Underflow)가 발생할 수 있다. 이를 보완하기 위해서 log값을 취해 더함으로써 이 문제를 해결한다. 임의의 문서에 대해서 해당 문서가 국사분야의 문서일 때와 그렇지 않을때의 $P(d|C_j)$를 각각 구하여 그 값이 더 높은 분야로 문서를 분류한다.

3.2.2 주제어 추출

문서의 주제어추출에는 주성분 분석(Principal Component Analysis)을 사용하였다. 주성분 분석은 반응변수들을 선형 변환시켜, 주성분이라고 부르는 서로 독립적인 새로운 인공 변수들을 유도한다. 이때 각 주성분이 보유하는 변이의 크기를 기준으로 그 중요도 순서를 생각할 수 있는데, 그들 중 첫 소수 몇 개의 주성분이 원래자료에 내재하는 전체 변이 중 가능한 많은 부분을 보유하도록 변환시킴으로서 정보의 손실을 최소화하는 차원의 축약을 기할 수 있게 된다. 결국, 주성분 분석을 이용하여 문서의 키워드들을 축소하여 정보의 손실을 최소화하면서 소수의 몇 개 단어로 문서의 내용을 표현할 수 있다. 즉, 그 문서의 주제어를 추출할 수 있다.[2]

이번 절에서는 [2]에서 사용한 방법을 토대로 하여 "영일서 고대 분묘 1백10기 무더기 출토"라는 신문기사를 대상으로 한 주제어 추출 과정을 선보인다.

표 1: 주성분 분석에 이용한 변수 리스트의 예

변수	변수값
X1	철기류
X2	분묘
X3	토기류
X4	토광
X5	발굴
X6	삼국시대
X7	유물
X8	흥해읍
X9	박물관
X10	원삼국시대
X11	출토
X12	무더기
X13	자료
X14	평가

주성분 분석에서의 변수로는 전체 문서에서 두 번이상 등장한 키워드들이 이용된다. 문서에서 1번만 등장한 키워드들은 주성분 분석에서 제외되어 주제어로 추출되지 않는다. 주성분 분석을 실시한 결과에서는 가장 큰 고유값을 갖는 주성분과 가장 큰 상관도를 갖는 변수가 주제어로 추출된다. 주성분 분석결과인 표 2에서 가장 큰 고유값을 갖는 주성분은 'PRIN1'이다. 'PRIN1'의 요소들 중에서 절댓값을 취했을 때, 가장 큰 상관도를 갖는 변수는 'X5'이다. 따라서 표 1에서 변수 'X5'에 해당하는 키워드 '발굴'을 주제어로 선정한다.

표 2: 주성분 분석결과의 예

변수\주성분	PRIN1	PRIN2	PRIN3	PRIN4	PRIN5	PRIN6	PRIN7
X1	-0.192347	0.102856	0.351075	-0.205303	0.292597	-0.131236	-0.429585
X2	-0.403741	-0.220495	-0.090205	0.0854	-0.242671	0.099958	-0.051266
X3	-0.468141	0.0485	0.462948	-0.219512	-0.241632	-0.373062	0.063914
X4	0.134905	0.179357	0.031807	0.466664	-0.116246	-0.396816	-0.170693
X5	-0.486003	0.351422	-0.512956	0.167206	-0.15705	-0.22618	0.021415
X6	0.206662	-0.061093	-0.08798	-0.433705	-0.31622	-0.219451	0.41519
X7	-0.266268	-0.32738	0.24026	0.071497	-0.099462	-0.101211	0.122205
X8	0.079694	-0.285676	0.124093	0.398761	-0.345078	0.13049	-0.416845
X9	-0.041819	-0.028459	-0.479137	-0.284585	-0.173462	-0.090927	-0.38297
X10	0.222898	0.140205	0.175998	0.082291	-0.652399	0.117815	0.117596
X11	-0.38329	-0.196869	-0.063429	0.062587	0.019054	0.560971	0.217045
X12	-0.009183	-0.238228	-0.074713	0.430409	0.271539	-0.356666	0.426648
X13	0.063458	-0.486973	-0.139885	-0.117235	-0.008899	-0.206776	-0.119251
X14	0.063458	-0.486973	-0.139885	-0.117235	-0.008899	-0.206776	-0.119251
고유값	1.122028	068144	0.558183	0.460390	0.225416	0.213265	0.110746

변수\주성분	PRIN8	PRIN9	PRIN10	PRIN11	PRIN12	PRIN13	PRIN14
X1	0.439747	0.278604	0.24427	-0.049192	0.345225	-0.21766	0.0
X2	0.377065	-0.581822	0.097878	-0.449801	0.063023	0.069771	0.0
X3	-0.402692	0.110895	-0.19114	-0.18558	-0.038102	0.259718	0.0
X4	-0.395645	-0.262504	0.362215	-0.063916	0.242226	-0.333662	0.0
X5	0.178908	0.286411	0.221217	0.246824	-0.195894	0.103289	0.0
X6	0.12599	0.173685	0.26832	-0.298394	-0.153445	-0.45449	0.0
X7	0.12907	-0.204984	-0.181085	0.62333	-0.136149	-0.474837	0.0
X8	0.050506	0.436189	-0.093894	-0.234478	-0.395672	-0.120828	0.0
X9	-0.15317	0.008999	-0.540312	-0.043846	0.328766	-0.27635	0.0
X10	0.247245	0.104919	-0.02405	0.249202	0.49867	0.23683	0.0
X11	-0.371438	0.278604	0.24427	-0.049192	0.345225	-0.21766	0.0
X12	0.214681	0.254697	-0.339161	-0.232099	0.298893	0.012713	0.0
X13	-0.070749	0.04796	0.258131	0.146524	0.084952	0.25659	-0.707107
X14	-0.070749	0.04796	0.258131	0.146524	0.084952	0.25659	0.707107
고유값	0.070748	0	0	0	0	0	0

3.2.3 지식관계 생성

지식 관계의 생성에는 Cosine Similarity를 이용한다. Cosine Similarity는 내적 공간에서 두 벡터 간 각도의 코사인 값을 이용하여 벡터 간의 유사도를 계산한다. 이때 비교되는 벡터가 몇 차원이든지 상관없이 적용될 수 있기 때문에 정보 검색 및 텍스트 마이닝 분야에서 문서 간의 유사도를 측정하는 용도로 자주 사용된다. 본 논문에서는 다수의 웹 문서들로 구성된 지식 간의 관계를 생성하는 방법으로서, 두 가지 지식을 구성하는 웹 문서들의 유사도를 측정하여 관계를 생성한다.

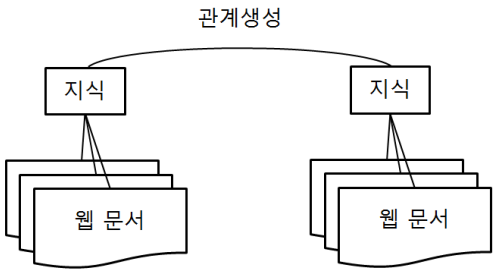

그림 3: 지식관계 생성

Cosine Similarity를 이용해 유사도 점수를 계산하기 위해서는 우선적으로 두 지식 간에 존재하는 웹문서들을 이용하여 벡터 공간 모델(Vector Space Model)을 생성한다. 벡터 공간 모델로부터 계산한 유사도 점수에 따라 지식 간의 관계를 생성한다.

벡터 공간 모델의 생성은 하나의 지식으로 그룹화된 문서들을 대상으로 하여 TF-IDF(Term Frequency Inverse Document Frequency)를 구해서 생성한다.

$$tf = f(t, d)$$

문서 d에서 단어 t의 총 빈도 f를 $f(t, d)$라 할 때, TF는 단순히 해당 문서에서 키워드의 총 빈도수이다.

$$idf = \log\left(\frac{D}{df}\right)$$

IDF는 역문서 빈도로서, 전체 문서의 수 D를 해당 단어를 포함한 문서의 빈도 df로 나눈 뒤 로그를 취한 것이다. 여기에서 정적인 지식지도와는 달리 동적으로 지식지도를 작성함에 따른 차이점이 발생한다. 한정된 개수의 문서들을 대상으로 하여 TF-IDF를 구할때는 전체 문서의 개수가 고정적이므로 큰 문제가 없다. 그러나 웹상의 문서들을 대상으로 동적인 지식지도를 작성할 때는 전체 문서의 개수가 고정적이지 않고 사실상 무한히 증가할 것이다. 그래서 문서의 개수가 증가할수록 DF(Document Frequency)를 구할때의 비용이 증가한다. 또한, TF-IDF를 사용하는 의도는 전체 문서에서 드물게 나타나고 특정 문서에서 빈번하게 나타나는 키워드에 대해서 특정 분야의 키워드로서 높은 가중치를 부여하기 위함이다. 그러나 본 논문에서는 Naive Bayesian Classifier에 의해 국사분야로 분류된 문서들만을 대상으로 지식지도를 작성하므로, 실제로 국사분야의 키워드라 할지라도 일반적인 키워드와 같이 낮은 TF-IDF 값이 부여될 수 있다. 이러한 문제를 해결하기 위해서 DF를 구할때는 국사분야의 문서를 대상으로 하지 않고, 국사분야가 아닌 별도의 문서들로 구성된 문서집합을 대상으로 하여 DF값을 구한다.

$$tfidf = tf \times idf$$

구해진 tf와 idf를 곱하여 $tfidf$를 계산한다. 이렇게 구해진 TF-IDF값은 유사도를 비교하려는 두 문서간의 길이에 대한 차이를 고려하지 않은 것이다. 이 부분을 보완하기 위해서 벡터를 길이가 1인 단위벡터로 만들어 정규화 한다.

$$||tfidf||_2 = \sqrt{\sum_{i=1}^{n} tfidf_i^2}$$

길이 정규화를 하기 위해서는 각각의 키워드들에 대한 TF-IDF를 제곱해서 하나로 합한다. 그리고 그 값에 루트를 씌운다. 이 값으로 전체 TF-IDF값을 개별적으

로 나누어서 길이를 정규화 한다. 마지막으로 생성된 벡터 공간 모델에 대해서 유사도를 계산한다.

$$\cos(\theta) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

여기에서 A_i는 문서 $d1$으로부터 만들어진 벡터 공간 모델에서 i번째 키워드에 대한 TF-IDF이다. B_i는 문서 $d2$로 부터 만들어진 벡터 공간 모델에서 i번째 키워드에 대한 TF-IDF이다. 따라서 $\sum_{i=1}^{n} A_i \times B_i$는 두 벡터간의 TF-IDF를 전부 곱해서 합산한 것으로 벡터의 내적을 의미한다. $\sqrt{\sum_{i=1}^{n}(A_i)^2}$와 $\sqrt{\sum_{i=1}^{n}(B_i)^2}$는 각각 문서 $d1$과 $d2$에 대한 단위 벡터들의 곱을 말하는 것으로, 해당 벡터에 존재하는 TF-IDF를 제곱해서 합산한 값에 루트를 씌운 것이다. 최종적으로 유사도 값은 벡터의 내적에 단위 벡터들의 곱을 나누어 계산한다.

4. 결론 및 향후과제

본 논문에서는 보편적인 텍스트마이닝 기법들을 활용한 지식지도의 작성 모델을 제안하였는데, 다음과 같은 몇 가지 과제가 남아있다.

1. 제안한 모델을 이용한 지식지도의 작성

   제안한 모델을 평가하기 위하여 웹에서 수집한 내용을 기반으로 소규모의 지식지도를 작성하였으나, 실제로 웹상에 존재하는 대량의 데이터를 기반으로 지식지도를 작성하기 위해서는 적지 않은 규모의 하드웨어 자원과 대량의 데이터를 처리할 수 있는 네트워크 환경이 요구된다. 이러한 요구 조건을 충족한 상태에서 보다 많은 양의 데이터를 기반으로 지식지도를 재작성할 필요가 있다.

2. 작성된 지식지도의 평가 및 검증

   작성된 지식지도에 대해서 해당 지식지도가 지식간의 관계를 올바르게 나타내는지, 또한 해당 지식지도가 어떠한 의미를 지니는지에 대해서 객관적으로 평가하고 검증할 수 있는 기법을 고안할 필요가 있다.

참고 문헌

[1] 이광희, "지식지도 작성을 위한 기초연구", 2007

[2] 이창범, 김민수, 이기호, 이귀상, 박혁로, "주성분 분석을 이용한 문서 주제어 추출", 정보과학회논문지 소프트웨어 및 응용 제29권 제10호, pp.747-754, 2002

[3] 김제욱, 김한준, 이상구 "Naive Bayes 문서 분류기를 위한 점진적 학습 모델 연구 (A Study Incremental Learning Model for Naive Bayes Text Classifier)", 정보기술과 데이타베이스 저널 제8권 제1호, pp.95-104, 2001