제재적 의미와 k-means 군집화를 이용한 개념추출 검색

1. 서론

정보검색 시스템의 사용자는 자신의 원하는 정보를 얻기 위해서 정보 요구(information need)를 하게 된다. 정보검색 시스템은 수집된 정보 또는 정보자료의 내용을 분석한 뒤, 기존에 적합하게 높은 정보로부터 사용자가 원하는 정보를 제공해준다. 인터넷이 발달된 요즘 정보의 흐름이 심해지고 있는 사용자들이 필요로 하는 것은 많은 양의 정보가 아니라 정확한 양의 정보로, 이제는 비록 사용자가 자신의 원하는 요구를 갖추기 시스템에 제기하고자 한다 하더라도 그것이 제대로 자신의 정보 요구를 충분히 반영해준 것임을 볼 때 아니된다. 정보화 시대가 되면서 사용자들은 단순한 검색보다는 개념적 정보 요구를 필요로 하게 되었다. 그러나 사용자가 제시하는 개념의 개념적인 단어만으로는 개념적인 표현이나 문서의 의미기에 대해서 신뢰할 수가 없다[1]. 이런 정보 검색의 단계에서 사용하는 어휘의 통합 및 개념화에 대한 지원이 필요하다는 것을 알 수 있다.

이런 단점을 보완하기 위해서 많은 연구가 있었다. 대표적인 예로 시소리스(thesaurus)는 용어의 용어간의 개념적 구조를 나타내며[8] 동의어(synonym), 상위어(broader terms), 하위어(narrower terms) 및 관계어 등으로 정의와 검색을 돕는 용어사전이다. 시소리스는 이와 같이 구축된 사전에서 사용자의 정보요구를 확장하는 역할을 한다. 그러나 시소리스의 구축은 많은 비용과 시간이 드는 관계였기 때문에 많은 분야에서 적절한 시소리스를 구축하기는 어렵다고 있다. 또한 그렇게 구축한 시소리스도 시간이 흐르면 새로운 개념들이 추가되는 변화를 신속하게 반영해 개념화하기는 어렵다고 있는 실정이다. 이러한

이용으로 시소리스는 specific한 범위에 사용될 때에는 효과를 보여주지만 general한 범위 즉, 엄밀히 말하면 매우 크고 정복된 경우와 반대로 새로운 문서가 유입되는 공간에 있어서 사용될 때에는 오히려 검색효용이 저하된다[6].

또 다른 연구로 Latent Semantic Indexing(LSI)[11]의 장점은 인기 문서가 불법적인 검색어의 영향을 받지 않는 범위에서의 검색의 확률을 높이는 장점이 있다. 기존의 대부분의 검색모델들은 디렉터로 처리하되 반해 LSI는 문건의 상호관련성을 유도하여 검색효용을 개선시키는 장점을 가진다[7].

본 연구에서 LSI를 사용하여 장재적 의미로 표현한 새로운 의미의 범위의 공간을 클러스터링 기법을 적용하여 개념 공간을 구성하고 이를 이용한 개념 기반 검색을 다루었다.

본 논문의 구성은 다음과 같다. 2장에서는 장재적 의미 분석에 대한 LSI에 대한 연구와 클러스터링 기법 중 k-means algorithm을 설명하였다. 3장에서는 양의 두 단계로 구축된 공간에서 개념을 추출하는 과정을 설명한다. 4장에서는 이에 따른 실험 및 결과를 분석하였다.

2. 기존연구에 관한 고찰

본 장에서는 본 연구에서 사용한 두 방법론을 장재적 의미 분석과 k-means 군집화에 대한 방법론을 살펴보기로 하였다.

2.1 장재적 의미 분석(LSI)에 대한 연구

논문 1학년의 분과학습과 관련된 과제 번호 AA-00-2060-00의 제재적 지원을 받아 수행되었음.
LSI는 용어와 문서간의 상호 관련성을 측정하는 벡터 검색의 일종이며 모델링을 위해서 통계적 기법인 Singular Vector Decomposition (SVD)를 사용한다. SVD는 기대된 용어-문서의 행렬 F의 k-차원의 높이와 높이로 분해하여 이에 보통 k는 100-300사이의 값을 갖지만 [17] 각각의 용어와 문서는 SVD를 통해서 k-차원의 LSI 공간 안에 벡터로 표현되며 유사한 내용의 문서에 알려져어 이들 공간 안에 유사한 값으로 표현된다. SVD는 데이터 공간에서 major associateive patterns를 변형하도록 용어와 문서를 벡터공간에 배치시킨다. 이 과정을 통해서 smaller, less important influence를 무시하게 된다. LSI에서는 용어-문서간의 행렬이 주어지면 이를 수학적으로 고속화된 방법론(SVD로 적용시킨 후 결과를 식(2)의 벡터 유사도 계산법으로 얻을 수 있다. SVD를 수행하기 위해서는 SVDPACK [3]를 사용하였다. SVD는 벡터공간으로서 초기 용어-문서 행렬 X의 크기가 K로 표현하며 X는 다음과 같이 1개의 T, T1의 3개의 서로 다른 행렬로 분해된다. T, D, S의 각각은 열의 벡터이며, S, D는 차원으로 singular value가 각각 선행방향으로 항하는 대각선 행렬이다.

\[ X = T S D^T \]

과 같이 1. Reduced singular value decomposition

여기서 \( r \) 행렬의 행 개수이고 \( d \)는 열 개수이다. 행렬 \( X \)의 rank를 \( m = \text{rank}(X) \)이라고 하자면 \( k \)값은 \( m \)보다 작은 reduced model에서의 차원(dimensions)을 결정하는 수치이다. 즉 \( r \leq d \)이므로 \( m = \text{diag}(\sigma_1, \ldots, \sigma_k) \) 등은 특이치(singular value)로 각각 행렬 A의 차원행렬을 다룬다. SVD가 수행되고 나면 용어와 문서가 k-차원의 공간 안에 벡터형태로 표현되어 존재하게 된다. 이렇게 문서와 문서간에 차이가 있는 의미를 파악할 수 있는 모든 수치를 구하기 위해, 정의에 벡터값을 구하려고 한다. LSI에서는 접근을 했으나 하나의 가중 문서 (pseudo document)로 접근하여 태생으로 k-차원의 벡터형태로 표현한다. 이때 가장 중요한 용어 벡터들은 가중치들이 됩니다. 다음의 식(1)과 같이 표현된다. 

\[ D_k = X_k^T S_k^{-1} \]  \hspace{1cm} (1)

문서와 용어들의 유사성(similarity)을 구하기 위해서 두 벡터의 고유소수 값을 사용한다.

\[ \text{sim}(d_i, q_j) = \frac{d_i \cdot q_j}{||d_i|| \cdot ||q_j||} = \frac{\sum w_{d_i,j} \cdot w_{q_j,j}}{\sqrt{\sum w_{d_i,j}^2} \cdot \sqrt{\sum w_{q_j,j}^2}} \] \hspace{1cm} (2)

LSI는 순수한 벡터모델을 문서를 표현하는 차원을 용어의 개수만큼 고차원 벡터로 처리했던 어음들 SVD의 rank수만큼 줄이므로 계산의 효율을 높였으며 벡터모델은 용어의 관계를 독립적으로 처리한 것과 달리 LSI는 용어간의 상호관련성을 고려했다는 점에서 개념적인 공간을 구축하는 데에 의미가 있다고 할 수 있다. 그러나 시험에서 연관된 것처럼 실제로 실적에서 의미적 공간을 구축하기 위해서 사용자의 점차 개별에 있는 별도의 표를 보완하기 위한 방법은 고려되지 않았다. 따라서 새로운 의미적 공간에서 문서집합을 구축하기 위해서 글의 벡터를 보완하는 방법을 다음과에서 연구하였다.

2.2 k-means 군집화(clustering)에 대한 연구

클러스터링 기법 중 k-means algorithm은 Euclidean distance(거리)를 이용하여 가장 가까운 점들을 찾아 군집으로 묶어주는 기법으로 차원의 제한 없고 간단한 알고리즘으로 가장 둔기 간단하다.

LSI를 통해서 의미적(semantic)적으로 분해된 공간에서 서로 가까운 거리에 있는 문서의 군집화를 시도 시키기로 결정하였다. 용어-문서간의 상호관련성을 변형한 의미적 공간에서 군집화를 클러스터화는 문서의 가진 속성을 표현하는 역할을 한다. 사용된 k-means 알고리즘은 다음과 같다.

Input : list of m-dimension vector Output : k lists of m-dimension vector

Step 1: Select K seeds from the data set
Step 2: Allocate each record to one of K seeds, which is most similar to the record
Step 3: Compute the centroid of K groups of records
Step 4: Repeat the steps 2 and 3 until the centroids do not change

3. 잠재적 의미와 k-means 군집화를 통한 개념 추출

LSI를 거쳐 분해한 k-차원의 벡터들은 단순한 벡터의 의미뿐만 아니라 LSI의 특성인 유사한 내용의 문서의 유사성에 의해 군집화가 이루어진다. 따라서 이 벡터들의 군집화는 거쳐서 형성된 각 클러스터별로 중심값(centroid)을 문서집합을 표현하는 개념(concept)을 뜻할 수 있다. 따라서 초기 시에 설정한 사용자의 별도의 전환을 바탕으로 각 클러스터별 중심값을 표준화하여 정의하는 방법을 제시하고자 한다. 먼저 k-차원의 벡터 투영을 앞에서 제시한 k-means algorithm을 사용하여 가려히 생성한 벡터의 k개의 군집을 만든다. 각 군집의 중심값을 대표값으로 하여 각 군집의 벡터와 유사도 구해 가중값(static weight)을 정의하면 다음과 같다. 보정식은 다음과 같다. 

\[ D_{k+1} = \alpha D_k + \beta C \] \hspace{1cm} (3)

\[ \alpha, \beta \geq 0 \]의 값으로 결과를 얻었다. 이 경우는 클러스터링의 결과에 영향을 받지 않고 사용자의 전반적 초기 입력을 LSI의 결과로 반영할 방법이며, 반대로 \( \alpha = 0 \)이고 \( \beta \)가
1인 경우는 사용자의 질의와 클러스터의 중심값으로 정
원하여 검색하는 의미가 된다.

4. 실험 및 결과분석

데스트본 문서집합은 TREC에서 제공한 DOE 문서의
일부였으며 문서의 개수는 10,000개이고 포함하는 용어
의 수는 42,849개였다. TREC의 topic 중 96번,134번,
135번을 가지고 실험하였으며 각각의 topic은 문서
10,000개에 대해 relevant document를 29개,25개,40
개로 가지고 있다. 본 실험에서는 (3.3)에서 주어진 점의
어를 바탕으로 a와 b의 값을 1.0,0.0으로 조정하면서
recall, precision을 계산하여 아래와 같은 결과를 얻었
다. LSI 수행단계에서는 SVDPACKC(3) package를 사
용하였다. SVDPACKC의 특징은 large sparse matrices에 대하여 singular value decomposition을 수
행한다는 것이며 kappa가지에서는 n 개의 numerical
(iterative) 방법을 제공하는데 그 중에서 Single-Vector
Lanczos Methos를 사용하였다.

결과판에서는 여러 실험을 통해 k=50에 대해 적절한
결과를 얻었으며 원소 개수 5개 이상의 균일하게 결과에는
무시하였다. 실행 시 시스템은 Linux server(kernel
2.2.16-3kr2.마진)과 Ultra sparc 2었다.

위와 같이 topic 134와 135의 실험에서는 b값이 높
고 a값이 작을수록 좋은 결과를 나타내었다. 즉 클러스
터링의 정확도가 영향을 받을 때 적절한 결과가 높아졌음을
의미한다. 특히 topic 135의 경우, a=0.0이고 b=0.8일
경우, a=1.0으로 적용했을 때보다 적절도에 있어서
25%의 향상을 보여주었다. 이 경우에는 사용자의 질의의
불완전성과 문서마다 둔감한 클러스터링의 영향을 받
아 표현된 것이 개인적으로 잘 매칭된 경우라고 분석
할 수 있다. 그러나 반대로 topic 96에 대해서는 a=0.0이
높고 b값이 작을수록 좋은 결과를 보였다. 이 경우에는 사
용자의 질의보다 정확도가 높게 적절히 표현된 경우이
므로 b값을 높여 클러스터링의 영향을 받는 것이 오히려
적절한 점을 확인할 수 있었다. 그림 3은 a,b 벡율
에 따른 topic 별 검색 결과의 변화를 보여준다. 따라서
클러스터링 결과의 적절도는 질의어와 클러스터의 관계에
따라 다른 결과를 나타낼 수 있다. 초기 질의어와 각 클러스터의 유사도 및 질의어와 상위 relevant한
문서와의 유사도를 계산하여 그 상관관계를 조사한 뒤
세로운 질의어 생성에 대한 판단 기준으로 삼을 수 있다.

5. 결론 및 향후과제

용어-문서간의 상호관련성을 반영해주는 LSI 모델을
통하여 임계적 의미를 분석한 결과를 구축함과 동시에
의미들은 균일하게 함으로써 개념을 추출하는 연구를 수행
하였다. 문서집합에 클러스터가 잘 구축되어 있고 사용
자 질의가 정확하게 맞지 않을 때, 클러스터링
의 정확한 개념을 clear형태로 도출하게 되므로 질의어
표현 및 검색에 효과가 있다는 것을 알았다.

향후과제로서는 실험에서 사용자의 초기 질의에 따
다 다른 결과가 유도되므로 질의어와 문서 및 클러스터
와 어떤 상관관계가 있는지 연구할 필요가 있다. 또한
k-means 군집화에 있어서 클러스터 k값을 이용함으로
산출하는 방법을 연구하는 것과 군집화 기법에 있어서
문서 클러스터링에 많이 사용되는 계층적 클러스터링 방
법 등 다양한 방법을 적용해보아 가장 효율적인 군집화
기법을 찾아내고자 한다. 마지막으로 위 실험의 각각의
데이터 set를 TREC 전체에 적용해보자 한다.

참고문헌

application, 156-162, 1973
1.0), Computer Science Department, CS-93-194.
1993
[4] Warren Sarle., The number of clusters from the
SAS/STAT User's Guide (1990) and Sarle and
[5] Ian H.W., Data Mining practical Machine
Learning Tools and Techniques, Morgan
Kaufmann Publishers. 75, 210-211, 2000
[6] Ricardo B.Y., Modern Information Retrieval,
Addison Wesley, 19-34, 1999
No.8, 1782-1789, 1997.08
[8] 정영기, 정보검색론, 구미무역, 1993