LSI를 이용한 문서 클러스터링
(The Document Clustering using LSI of IR)

고 지 현*, 최 영 란*, 유 준 현* 박 순 철*
(Ji-Hyun Goh, Young-Ran Choi, Jun-Hyun Yoo, Soon-Cheol Park)

요 약 정보검색시스템에서 가장 중요한 것은 사용자의 요구에 부합하는 결과를 도출하는 것이다. 이를 위하여 사용자의 질의와 연관된 모든 문서들을 추출하게 되는데, 이 많은 결과 문서들 중에서 사용자가 원하는 문서는 소수이고, 원하는 문서를 찾는 것도 쉽지 않다. 따라서 적절한 결과 문서를 도출하기 위하여 연관된 문서들끼리 그룹화 시키는 클러스터링 방법이 많이 이용된다. 본 논문에서는 기존의 문서 내의 색인어 보다는 그 의미에 기반하여 클러스터링 하였다. 이를 위하여 LSI 모델을 적용하였고, 문서 클러스터링 방법으로 많이 사용하고 있는 K-Means 알고리즘을 이용한 클러스터링과의 차이점을 비교, 분석하였다.

Abstract The most critical issue in information retrieval system is to have adequate results corresponding to user requests. When all documents related with user inquiry retrieve, it is not easy not only to find correct document what user wants but is limited. Therefore, clustering method that grouped by corresponding documents has widely used so far. In this paper, we cluster on the basis of the meaning rather than the index term in the existing document and a LSI method is applied by this reason. Furthermore, we distinguish and analyze differences from the clustering using widely-used K-Means algorithm for the document clustering.

1. 서 론

최근 웹 문서 양이 기하급수적으로 증가하면서 정보검색엔진의 성능평가에 대한 논의가 대두되고 있다. 문서의 양 뿐만 아니라 작성된 문서의 종류 또한 다양하여 사용자의 요구에 적합한 결과 문서를 도출해 내기가 어렵다. 대부분의 정보검색엔진은 사용자의 질의를 분석하여 사용자의 의도와는 상관없이 질의에 따른 모든 문서를 찾아내는 방법을 사용한다. 많은 양의 결과 문서들을 사용자에게 보여주고, 사용자가 스스로 원하는 문서를 찾아내게 한다. 각 검색엔진 별로 문서의 순위화를 이용하고 있지만 적합한 문서를 찾아내는 것은 결국 사용자의 몫이다.

† 이 논문은 정보통신연구진흥원의 위탁으로 수행되고 있는 과제임 (과제명 : 음성 및 자연어 인터페이스 의미기반 정보검색 시스템 연구)

* 전북대학교 정보통신공학과

검색된 모든 문서들 중에서 사용자가 적절한 결과 문서를 추출하기란 쉽지 않다. 오히려 적합한 결과 문서를 찾지 못하는 경우가 더 많다. 그래서 사용자가 검색된 많은 문서들 중에서 결과 문서를 더 빨리 찾을 수 있도록 클러스터링 방법을 이용한다. 이 방법은 유사한 문서끼리 그룹화 시키는 방법으로 자동 문서분류나 데이터 마이닝 분야에서 많이 이용하고 있다. 이를 이용하여 결과 문서들을 클러스터링 하면 비슷한 문서별로 그룹화가 되기 때문에 사용자는 한 눈에 검색결과를 볼 수 있고, 자신이 생각하는 개념과의 일치 여부에 따라 원하는 결과 문서를 찾을 수 있다.

본 논문에서는 문서 내의 색인어의 의미를 기반으

로 하여 각 문서들을 클러스터링 한다. 이를 위해 LSI 이론을 도입하여 문서들을 저차원으로 사상시키고 각 문서들의 연관관계에 따라 그룹화 한다. 그리고 기존의 K-Means 알고리즘을 이용한 클러스터링 방법과 비교, 분석한다. 2장에서는 클러스터링의 개념과 비교를 위해 구현된 K-Means 알고리즘에 대해 설명한다. 3장에서는 본 연구를 위해 구현된 클러스터링의 기본 이론이 되는 LSI 모델에 대해 알아보고, 4장에서는 구현된 클러스터링의 구조에 대해 살펴본다. 5장에서는 각각 클러스터된 결과들을 비교, 분석하고, 끝으로 6장에서는 결론 및 향후 연구방향을 다룬다.

2. 클러스터링

문서 클러스터링은 사용자의 개입 없이 문서 집합에서 문서들 간의 유사도를 측정하여 유사한 문서들을 모아서 집단화하는 방법이다. 문서들 간의 유사도 측정은 각 문서들을 대표할 수 있는 색인어의 가중치에 따라 행해진다. 가중치 부여 방법은 여러 가지가 있는데 가장 흔히 쓰이는 방법은 빈도수에 따른 부여 방법이다.

클러스터링 기법은 계층적 기법과 비계층적 기법으로 구분된다. 일반적으로 클러스터링 성능에 있어서는 계층적 기법이 비계층적 기법에 비해 우수하지만 처리시간에 있어서는 비계층적 기법이 훨씬 효율적인 것으로 나타난다. 계층적 기법으로는 단일연결(single linkage), 완전연결(complete linkage), 그룹평균연결(group average linkage), 워드 기법(ward's mathod) 등이 있으며, 비계층적 기법으로는 싱글패스(single pass), K-Means 알고리즘, EM(expectation maximization) 알고리즘 등이 있다. 본 논문에서는 클러스터링 결과의 비교를 위하여 비계층적 기법 중 K-Means 알고리즘을 이용하였다. K-Means 알고리즘에 대한 설명은 다음과 같다.

---

1. K값(클러스터의 개수)을 정한다.

2. K개의 초기 중심값(proto-centroid)를 정한다.

3. 각 문서(d_i)들과 중심값(c_j)사이의 거리를 구한다.

[Euclidean Distance] :

$$dist(\vec{d_i}, \vec{c_j}) = \sum_{k=1}(d_{ki} - c_{kj})^2$$

($i = 1,2,\cdots,n$ n : 전체문서의 개수
$j = 1,2,\cdots,k$ k : 중심값의 개수
= 클러스터의 개수)

4. 가장 짧은 거리의 문서를 각 중심값의 클러스터에 할당한다.

$$\arg\min_{i,\ j} dist(\vec{d_i}, \vec{c_j})$$

$$d_i \in_i G_{c_j},$$

if $dist(\vec{d_i}, \vec{c_j}) < dist(\vec{d_i}, \vec{c_l})$
(for $all\ l = 1,2,\cdots,k \quad l \neq j$)

5. 새로운 중심값을 계산한다.

$$\vec{c_j} = \frac{1}{\left|\vec{c_j}\right|} \sum_{l=1}^{\left|\vec{c_j}\right|} \vec{d_l}$$

6. 이전의 중심값과 새로운 중심값을 비교하여 벡터간 차이가 거의 없을 때까지 반복한다.

$$If \max \delta(\vec{c}_j^{\,old}, \vec{c}_j^{\,new}) < \theta \ then \ return$$

else goto 3

---

3. LSI(Latent Semantic Indexing)

LSI는 문서의 내용이 서술된 색인어 보다는 그 안에 표현된 개념에 기반한다는 점에 착안하여 제안된 모델이다. 이것은 문서들이 같은 색인어로 구성되어 있지 않더라도 연관성을 나타낼 수 있다. 어떤 문서가 다른 문서와 개념을 공유한다면 유사한 문서라 할 수 있다. 이 모델의 요점은 문서들을 저차원 벡터 공간으로 사상시키는데 있다.

LSI를 위한 선재 조건은 색인어와 문서로 구성된 행렬을 적절하게 분해하는 것이다. 이를 위해서 SVD(Singular Value Decomposion)를 이용하는데 이에 대한 정의는 다음과 같다.

행렬 m×n으로 나타내는 전체 문서 집합(collection) A는 각 원소의 값으로 가중치를 갖는다고 하자. 여기서 m×n 행렬은 색인어×문서를 나타낸다. 이때 A를 SVD로 분해한다.

$$A = U\Sigma V^T$$

여기서 U는 색인어간 상관 행렬(association matrix)로부터 얻은 m×m 고유 벡터 행렬(orthgonal matrix) 이고, V는 문서간 상관 행렬로부터 얻은 n×n 고유 벡터 행렬이다. Σ는 단일 값을 갖는 m×n 대각 행렬(diagonal matrix)이다. U를 이용하여 단어들은 m차원, V를 이용하여 문서들은 n차원으로 사상 시킬 수 있다. 동일한 차원으로 단어 벡터와 문서 벡터를 사상시킨다면 단어와 단어의 관계, 단어와 문서간의 관계, 문서와 문서와의 관계를 알 수 있다.

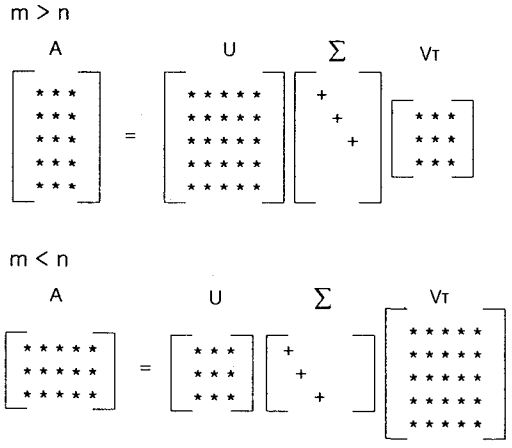

그림 1 SVD의 성분 행렬

예를 들어 이차원으로 사상시킨다면 그림 2와 같이 색인어의 관계를 나타내는 행렬 U를 m×2 행렬로 잘라주고, 문서들의 관계를 나타내는 행렬 V를 m×2 행렬로 잘라 준 다음 각 행렬의 첫 번째 열은 x 축에, 두 번째 열은 y 축으로 사상시킨다. 그러면 각 색인어의 벡터 값과 각 문서의 벡터 값이 같은 공간에 사상되기 때문에 벡터 값을 가진 각 원소들을 비교해 주면 서로의 연관 관계를 알 수 있다. 이때 연관 관계를 나타내기 위한 유사도 계산은 코사인 법칙을 이용한다.

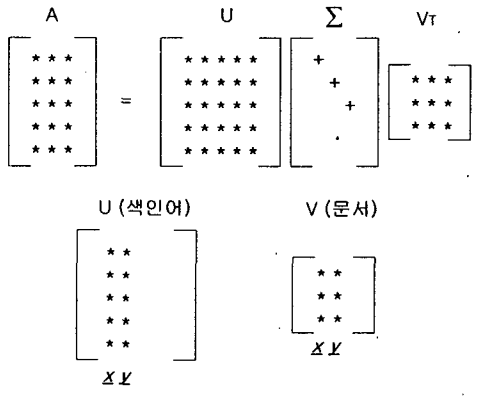

그림 2 이차원 행렬의 예

임의의 벡터 X, Y가 다음과 같을 경우,

$$X=(x_1, x_2, \cdots, x_n),\quad Y=(y_1, y_2, \cdots, y_n)$$

두 벡터의 코사인 값은 다음과 같다.

$$\cos\theta = \frac{x_1y_1 + x_2y_2 + \cdots + x_ny_n}{\sqrt{x_1^2+x_2^2+\cdots+x_n^2}\ \sqrt{y_1^2+y_2^2+\cdots+y_n^2}}$$

4. 클러스터링 구현

본 논문에서는 LSI를 이용한 클러스터링과 비교를 위한 K-Means 알고리즘을 이용한 클러스터링 부분을 구현하였다.

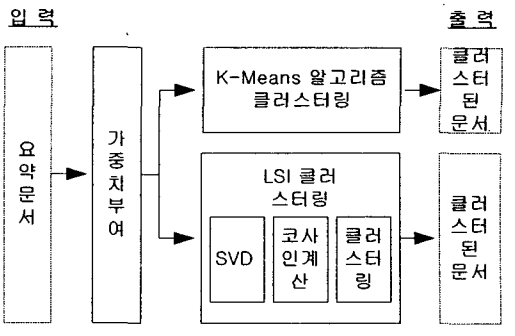

그림 3 클러스터링 구조

입력으로 사용한 실험 데이터는 요약된 문서들을 이용하였다. 이 데이터는 각 문서 별로 [색인어 | 문서 내의 빈도수(tf) | 전체 문서 내의 빈도수(df)]로 구성되어 있다. 102개의 문서와 7507개의 색인어로 구성되어 있으며 색인어는 각각 고유한 번호를 가지고 있다. 문서 내의 빈도수(tf)는 현재 색인어가 있는 문서에서 그 색인어가 존재하는 수이다. 전체 문서 내의 빈도수(df)는 전체 문서(collection)에서 그 색인어가 몇 번 나왔는지를 나타내어 준다.

입력 데이터를 이용하여 각 문서 당 색인어 별로 가중치 값을 계산한다. 이 때 문서 내의 빈도수에 역문헌빈도수를 곱한다. 여기서 역문헌빈도수는 $\log \frac{N}{df}$ 이다 (N : 전체 문서의 개수). 가중치를 부여하는 이유는 각 문서 당 색인어의 고유한 값을 부여하여 문서 고유성을 부가하기 위해서이다.

가중치를 부여한 다음 문서별 클러스터링을 한다. 이때 K-Means 알고리즘을 이용한 클러스터링과 LSI, 방법을 이용한 클러스터링 부분으로 각각 클러스터링 한다. LSI를 이용한 경우에는 7507×102 행렬(색인어×문서)을 SVD를 이용하여 분해한다. 문서간의 관계를 나타내 주는 행렬(102×102)를 가지고 기준 벡터 값에 따라 각각의 코사인 값을 계산한 다음 연관된 문서별로 그룹화 한다.

5. 결 과

본 논문에서는 이차원으로 사상시키기 위해서 문서 행렬을 102×2 행렬을 이용하였다. 이차원을 이용한 이유는 이차원이 평면으로 이루어져 있기 때문에 문서들의 위치를 한 눈에 알아 볼 수 있는 도식화에 유용하고, 서로의 연관관계를 나타내기가 쉽기 때문이다. 이차원 사상을 위한 문서간의 행렬(V)인 112×2 행렬은 다음 그림 4와 같이 나타났다.

문서들의 각 벡터의 위치에 따라 코사인 값을 이용하여 문서들을 클러스터링 할 수 있다. 그림에서와 같이 값이 일정한 구역 내에 존재하는 문서들은 연관된 문서들로 간주할 수 있다. 물론 클러스터의 개수에 따라 각 구역은 달라진다.

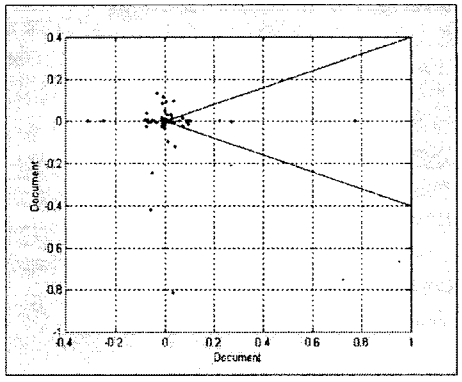

그림 4 이차원 사상

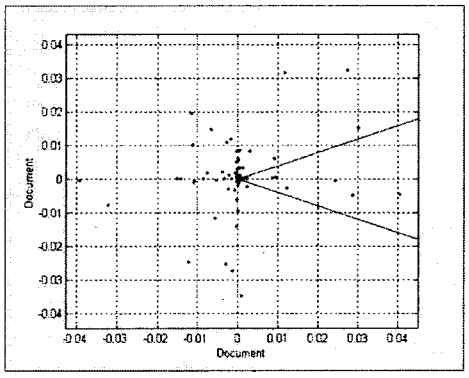

그림 5 이차원 사상 확대

그림 5는 그림 4를 확대한 그림이다. 여기서 같은 코사인 값을 나타내는 두 개의 직선이 x축을 기준으로 나뉘어 있다. 이 직선들을 기준으로 연관 문서들을 찾아낸다면 삼각형 내에 존재하는 약 6개 정도의 문서들이 유사한 문서로 정할 수가 있다. 이러한 방법을 이용하여 클러스터링 한 결과는 다음 그림과 같다.

Cluster:1 Number:22
25 50 56 57 58 59
61 62 73 74 75 76
79 84 86 88 90 93
94 98 99 100

Cluster:2 Number:4
29 51 52 71

Cluster:3 Number:36
2 3 4 5 6 9
10 12 13 14 15 16
17 18 19 20 21 26
27 28 30 33 34 35
37 38 39 40 41 42
43 44 46 78 80 81

Cluster:4 Number:7
11 22 24 36 45 53
67

Cluster:5 Number:33
1 7 8 23 31 32
47 48 49 54 55 60
63 64 65 66 68 69
70 72 77 82 83 85
87 89 91 92 95 96
97 101 102

그림 6 LSI 클러스터링 I

Cluster:1 Number:21
50 56 57 58 59 61 62
73 74 75 76 79 84 86
88 90 93 94 98 99 100

Cluster:2 Number:1
25

Cluster:3 Number:0

Cluster:4 Number:4
29 51 52 71

Cluster:5 Number:20
5 6 9 13 14 16 18
19 27 30 34 35 39 40
41 43 44 46 78 80

Cluster:6 Number:16
2 3 4 10 12 15 17
20 21 26 28 33 37 38
42 81

Cluster:7 Number:2
22 45

Cluster:8 Number:5
11 24 36 53 67

Cluster:9 Number:3
1 48 64

Cluster:10 Number:28
7 23 31 32 47 49 54
55 60 63 65 66 68 69
70 72 77 82 83 85 87
89 91 92 95 96 97 101

그림 7 LSI 클러스터링 II

좌측의 그림들은 LSI를 적용하여 클러스터링한 결과이다. 그림 6은 5개의 그룹으로 클러스터링 하였고, 그림 7은 10개의 그룹으로 클러스터링 하였다. 문서들의 벡터 값이 고정되어 있고, 이차원 평면 공간의 특성을 이용하여 x축을 기준으로 코사인 값을 구하고 각 문서 당 위치를 비교하였다..

Iteration : 6

Cluster : 1 *Number* : 1
10

Cluster : 2 Number : 1
17

Cluster : 3 Number : 1
29

Cluster : 4 Number : 16
1 6 11 18 22 28 34 40 46 56 59 62 65 71 90 93

Cluster : 5 Number : 1
45

Cluster : 6 Number : 1
52

Cluster : 7 Number : 62
2 3 4 5 7 8 9 12 13 14 15 16 19 20 21 23 24 25
26 27 30 32 33 36 37 39 41 42 43 48 49 50 51 53
54 57 58 61 64 67 68 70 72 75 76 78 79 80 82 83
85 86 88 89 92 96 97 98 99 100 101 102

Cluster : 8 Number : 1
74

Cluster : 9 Number : 17
31 35 38 44 47 55 60 63 66 69 73 77 81 84 87 91
94

Cluster : 10 Number : 1
95

그림 8 K-Means 클러스터링 I

그림 8과 그림 9는 K-Means 알고리즘을 이용하여 클러스터링을 한 결과이다. K-Means 알고리즘의 특성상 기준이 될 중심벡터를 임의로 정해주었다. 그 결과 초기 중심벡터에 따라 같은 알고리즘을 적용하였지만 다른 결과가 나타났다. 그림 9의 경우처럼 한 클러스터로 집중하는 경우도 있었다. K-Means 알고리즘의 이론상으로는 무한으로 중심값을 재조정할 경우 초기 중심벡터에 상관없이 같은 결과가 나온다. 하지만 실제로 무한 루프로 처리하여 조정 할 수 없기 때문에 어느 정도 기준을 정해주게 된다. 그림 8의 경우는 중심값을 5번 재조정하여 6번째에 나온 결과이고, 그림 9의 경우는 중심값의 재조정은 없었고 초기 중심값을 이용하여 클러스터 된 결과이다.

```
Iteration : 1

Cluster : 1          Number : 1
9

Cluster : 2          Number : 1
11

Cluster : 3          Number : 1
25

Cluster : 4          Number : 93
1 2 3 4 5 6 7 8 10 12 13 14 15 16 17 18 19 20
21 22 23 24 26 27 28 29 30 31 32 33 34 35 36 37
38 39 40 41 42 43 44 45 46 48 49 50 51 52 54 55
56 57 58 59 60 62 63 64 65 66 67 68 69 70 71 72
73 74 75 76 77 79 80 81 82 83 84 85 87 88 89 90
91 92 93 94 95 96 97 98 100 101 102

Cluster : 5          Number : 1
47

Cluster : 6          Number : 1
53

Cluster : 7          Number : 1
61

Cluster : 8          Number : 1
78

Cluster : 9          Number : 1
86

Cluster : 10         Number : 1
99
```

그림 9 K-Means 클러스터링 II

이에 반해 K-Means 알고리즘을 이용한 클러스터링 방법과 같은 클러스터의 개수로 LSI를 이용하여 문서들을 그룹화한 그림 7의 경우에는, 정해진 벡터 값을 기준으로 하여 각 문서들의 유사도 측정을 코사인 법칙을 적용하였기 때문에 클러스터링 되는 결과 문서들이 고정적이었다. 단지 클러스터의 개수에 따른 문서들의 차이만 있을 뿐 문서들의 유사도 결정에 있어서는 변화가 없었다.

6. 결 론

본 논문에서는 클러스터링에 있어서 LSI 모델을 적용하여 문서들을 클러스터링 하였다. 다른 클러스터링 방법과의 비교를 위하여 문서 클러스터링 부분에서 많이 사용되고 있는 K-Means 알고리즘을 이용하였다.

각 문서 당 색인어의 고유한 값을 부여하기 위하여 적절한 가중치 값을 부여하였다. 이 때 문서와 색인어의 관계와 전체 문서(collection)와 색인어의 관계를 고려하였다. 색인어와 문서로 이루어진 행렬로 구성하고 나서 SVD 이론을 적용하여 행렬을 분해하였다. 분해 결과 나온 행렬들 중 문서 간의 관계를 나타내는 행렬을 취한 후 이차원 사상을 위해 필요한 행렬로 재구성하였다. 문서간의 유사도를 측정하기 위하여 코사인 법칙을 적용하여 각 문서 당 벡터 값을 기준으로 정한 x축에 따른 코사인 값을 계산하였다. 코사인법칙의 특성을 이용하여 클러스터의 개수에 따른 범위에 따라 각 문서들을 클러스터링 하였다.

그 결과 LSI를 이용한 클러스터링이 K-Means 알고리즘을 적용한 클러스터링 보다 문서의 유사도 측정에 있어서 안정적이었다. K-Means 알고리즘이 임의적으로 정해주는 중심값에 따라서 클러스터의 결과에 많은 변화가 있었던 반면 LSI를 적용한 클러스터링 방법은 클러스터의 개수에 따른 문서의 변화만이 있었을 뿐 문서들의 연관관계에 있어서는 고정적이었다.

향후에는 벡터 계산에 있어서 속도가 많이 떨어지고 가상 메모리 공간을 많이 필요로 하는 경향이 있는데 이에 대한 연구도 요구된다.

7. 참고문헌

[1] Michael W. Berry, Murray Browne, "Understanding Search Engines", Univ. of Tennessee.

[2] Ricardo Baeza-Yates, Berthier Ribeiro-NetoRoger, "Modern Information Retrieval", Addison Wesley, 1999.

[3] Khaled Alsabti, Sanjay Ranka, Vineet Singh, "An Efficient K-Means Clustering Algorithm", IIPS 11th Intrnational Parallel Processing Symposium, 1998.

[4] Michael W. Berry, Susan T. Dumais, Todd A. Letsche, "Computational Methods for intelligent Information Access", ACM, 1995.

[5] William B. Frakes, Richard Baeza-Yates, "Information Retrieval", Prentice Hall, 1992.

[6] 정영미, 이재윤, "지식 분류의 자동화를 위한 클러스터링 모형 연구", 정보관리학회지, 18(2) : 203-230, 2001.

[7] David W. Lewis, "Matrix Theory", World Scientific, 1991.

[8] http://www.mathworks.com.