

## 하이퍼링크를 이용한 그래프 기반의 웹 문서 클러스터링

### Web Document Clustering based on Graph using Hyperlinks<sup>↓</sup>

이준, Joon Lee\*, 강진범, Jinbeom Kang\*\*, 최중민, Joongmin Choi\*\*\*

---

**요약** ~ 인터넷 상의 웹 문서의 수가 기하급수적으로 늘어남에 따라서, 정보검색에서의 웹 문서 클러스터링은 성능과 속도가 매우 중요하게 되었다. 웹 문서 클러스터링은 의미적으로 관계가 있는 웹 문서들을 같은 클러스터로 군집함으로써 정보 검색을 보다 빠르고, 정보를 정확하게 제공할 수 있다. 그물망 그래프 형태의 클러스터링은 모든 문서간의 유사도를 측정함으로써 재현율을 높일 수 있지만, 높은 계산 비용을 갖는다. 본 논문에서는 그물망 형태의 클러스터링의 재현율과 정확율을 유지하며 계산 비용을 줄이기 위하여, 웹 문서의 구조적 특징인 하이퍼링크(Hyperlinks)를 이용한 클러스터링 방법을 제안한다.

**Abstract** ~ With respect to the exponential increment of web documents on the internet, it is important how to improve performance of clustering method for web documents. Web document clustering techniques can offer accurate information and fast information retrieval by clustering web documents through semantic relationship. The clustering method based on mesh-graph provides high recall by calculating similarity for documents, but it requires high computation cost. This paper proposes a clustering method using hyperlinks which is structural feature of web documents in order to keep effectiveness and reduce computation cost.

**핵심어:** *Web document clustering, hyperlinks*

---

\*주저자 : 한양 대학교 컴퓨터공학과 석사 과정 e-mail: jlee@islab.hanyang.ac.kr

\*\*공동저자 : 한양 대학교 컴퓨터공학과 박사 과정 e-mail: midgetfx@hanyang.ac.kr

\*\*\*교신저자 : 한양 대학교 컴퓨터공학과 교수; e-mail: jmchoi@hanyang.ac.kr

## 1. 서론

인터넷 상의 웹 문서가 기하급수적으로 늘어남에 따라서, 사용자가 원하는 정보의 검색은 더 어렵게 되었다. 웹 문서 클러스터링은 의미적으로 관계가 있는 문서들을 같은 클러스터로 군집함으로써, 정보 검색을 보다 빠르고, 정확하게 제공할 수 있다.

문서 클러스터링은 분할적 클러스터링[1][2], 계층적 클러스터링[3][4], STC[5][6], 밀도기반 클러스터링[7][8] 등을 이용한 방법이 있다. 분할적 클러스터링으로 가장 많이 사용되는 K-Means 는 클러스터의 개수를 예측하기 어려운 단점이 있고, 계층적 클러스터링은 모든 문서 쌍의 유사도에 대한 사전 지식에 기반한 클러스터링으로써, 모든 문서 쌍의 유사도를 비교하므로 높은 계산 비용을 갖는다. STC 는 클러스터내의 요소의 중복을 허용함으로써 너무 많은 클러스터를 생성하는 단점이 있다. 이와 같이 기존의 클러스터링은 모든 요소들을 비교하기 때문에, 그물망 형태의 클러스터링 방법에 속한다. 그물망 형태의 클러스터링은 높은 계산 비용을 가지기 때문에 빠른 속도를 요구하는 웹에서는 적합하지 않다.

웹 문서의 구조적 특징인 하이퍼링크(Hyperlink)는 문서간의 의미적 연관성이 있음을 나타내는 표시이다. 하이퍼링크로 연결되어 있는 문서들은 서로 연관성이 있음을 나타내기 때문에 이러한 성질을 이용하여 클러스터링을 하면 문서간의 비교 횟수를 줄일 수 있다.

본 논문에서는 웹 문서에 포함 된 하이퍼링크를 이용하여 문서들간의 유사도 비교횟수를 줄여 클러스터링의 계산 비용을 줄인다. 또한 클러스터를 대표하는 단어를 추출하여 문서에 붙임으로써 문서의 분류를 명확하게 하여 성능을 개선시키고, 문서간의 유사유무를 판단하는 임계 값을 자동으로 식별하는 방법을 제안한다.

## 2. 관련 연구

클러스터링 알고리즘은 서론에서 설명한 것과 같이 분할적 클러스터링 (Partitioning Clustering), 계층적 클러스터링 (Hierarchical Clustering), STC (Suffix Tree Clustering), 밀도 기반 클러스터링 (Density-based Clustering) 알고리즘으로 나눌 수 있다.

분할적 클러스터링 알고리즘에 가장 많이 쓰이는 K-means 는 생성 될 클러스터의 수 K 를 입력 받아야 되는 단점이 있다. 이를 해결하기 위하여, Memarsadeghi, et al 는 [9]에서 ISODATA 클러스터링 알고리즘을 제안하였다. ISODATA 는 클러스터의 수 K 를 입력 받아서 실행되는 것이 아니라 클러스터의 크기에 따라 분할되고 병합되는 알고리즘이다. 이 알고리즘은 클러스터의 개수 K 를

입력받는 문제점은 해결하였지만, 초기 클러스터의 중심선택이 클러스터링의 결과에 영향을 주는 문제점은 해결하지 못 하였다.

계층적 클러스터링 알고리즘은 응집 (Agglomerative)과 분할 (Devisive) 방법으로 분류된다. 응집은 상향식 (Bottom-up) 접근 방법으로, 분리된 그룹으로 시작하여 종료 조건을 만족할 때 까지 반복하여 유사한 그룹을 클러스터링 한다. 응집과 반대로 분할은 하향식 (top-down) 접근 방식으로, 전체 요소를 포함하는 하나의 클러스터에서 종료 조건을 만족할 때 까지 반복하여 분할하는 방법이다. BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), FIHC (Frequent Itemset-based Hierarchical Clustering)등이 이에 속한다. 계층적 클러스터링 알고리즘은 모든 요소의 쌍을 비교하기 때문에 높은 계산비용을 갖는다.

STC 알고리즘은 사용자에게 검색어를 입력 받아 검색엔진으로부터 반환된 문서의 집합에 포함되어 있는 구(句)들을 이용하여 어미(語尾) 트리를 생성하고 클러스터를 생성한다. Zamir 는 [5]에서 검색엔진을 통해서 반환되는 웹 문서의 Full Text 를 이용하여 클러스터링을 하였다. 이는 클러스터의 중복을 허용하였고, 많은 계산비용과 너무 많은 클러스터를 생성하였다.

밀도 기반 클러스터링은 분포된 밀도가 주어진 임계값보다 크면 클러스터링을 하는 방법이다. 이 알고리즘은 밀도 기반 정의를 이용하므로 잡음(noise)을 효과적으로 처리할 수 있고, 불규칙한 모양과 크기의 클러스터를 생성할 수 있다. 밀도 기반 클러스터링에 대표적인 알고리즘으로는 DBSCAN(Density-Based Spatial Clustering of Application with Noise)와 OPTICS(Ordering Points To Identify the Clustering Structure)가 있다. DBSCAN 은 이웃에 위치한 데이터들의 밀도에 기반하여 클러스터링을 생성하며, OPTICS 데이터의 밀도 기반 클러스터링 구조에 기초하여 객체들을 순서화 한 후 그 정보에 의해서 가장 밀도가 높은 데이터들을 자동으로 클러스터링 하는 알고리즘이다.

이상의 기존 알고리즘들은 클러스터의 개수를 결정하여야 하는 문제점 그리고 잡음에 민감한 문제점을 갖는다. 또한 높은 계산 비용을 가지므로 웹 문서 클러스터링에는 적합하지 않다. 그래서 본 논문에서는 웹 문서의 구조적 정보인 하이퍼링크와 클러스터를 대표할 수 있는 단어의 집합을 이용하여 모든 요소를 비교하는 그물망 형태 클러스터링의 재현율을 유지하면서 적은 계산 비용을 들이는 알고리즘을 제안한다.

## 3. 결론

본문에서는 본 논문에서 제안하는 클러스터링을 위한 그래프 생성, 유사도 측정 및 클러스터링 과정에 대해서 기술한다.

1. 웹 문서와 하이퍼링크를 이용한 그래프 생성.
2. 생성된 그래프에서 임의의 노드를 선택.
3. 깊이 우선 탐색으로 선택한 노드와 에지로 연결된 노드들과의 유사도 계산.
4. 계산된 유사도가 임계값 보다 클 경우 클러스터링.
5. 생성된 클러스터에서 최하위 노드를 제외 한 노드들에서 클러스터를 대표하는 단어 N 개를 추출 후 최하위 노드에 해당하는 문서의 내용에 붙임.
6. 모든 클러스터가 생성 될 때까지 단계 4 반복.

표 1 제안하는 클러스터링 알고리즘의 절차

표 1 은 본 논문에서 제안하는 알고리즘을 간략하게 설명한 것이다. 웹 문서 집합에서 하이퍼링크를 이용하여 그래프를 생성하고, 클러스터링을 위해 임의의 노드를 선택한다. 선택한 노드에서 깊이 우선 탐색으로 웹 그래프를 탐색하며 클러스터링을 하는 단계를 거친다. 각 단계에 대한 자세한 설명은 다음과 같다.

### 3.1 웹의 구조적 정보를 이용한 그래프 생성

클러스터링을 하기 위하여 웹 문서들간의 하이퍼링크(Hyperlinks) 정보를 이용하여 방향성이 없는 그래프를 생성한다. 그래프의 노드(Node)는 각 문서를 표현하며, 인 링크(In-link) 및 아웃 링크(Out-link)는 그래프의 에지(Edge)로 표현된다. 클러스터링을 하기 위하여 하이퍼링크로 연결되어 있는 문서 집합을 그림 1 처럼 웹 그래프를 생성한다.

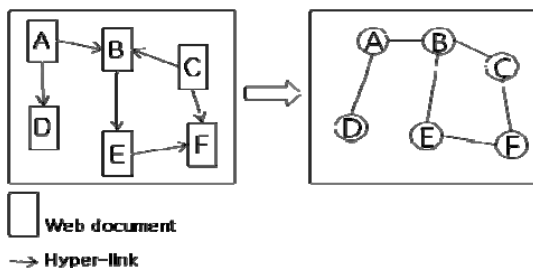


그림 1 웹 그래프 생성 예제

하이퍼링크는 문서와 문서간의 의미적으로 관계가 있음을 나타내는 표시이고, 문서간의 의미적 관계를 나타내는데 인 링크와 아웃 링크의 구분은 무의미 하기 때문에 방향성 없는 그래프를 생성한다.

### 3.2 그래프 기반의 클러스터링

본 논문에서는 시작 노드에서 에지를 따라서 깊이 우선 탐색을 하는 그래프 탐색 알고리즘을 클러스터링에 적용하였다. 본 논문에서 설명하는 하이퍼링크 기반의 클러스터링이란 시작 노드와 에지로 연결되어 있는 노드들의 유사도를 계산 후 임계값 보다 큰 경우 클러스터링 하는 방법이다. 그러나 이러한 방법은 깊이가 커질수록 시작 노드와의 연관성을 고려하지 않고, 상위 노드와의 연관성만을 고려하기 때문에, 시작노드와의 연관성이 적어져서 정확한 클러스터링을 할 수 없는 문제가 있다. 예를 들어 아래 그림 2 의 노드 B 와 같이 A,I 의 한 분야인 Semantic Web 관련 문서가 있을 때, 노드 B 와 노드 C 만의 유사도 비교 시 두 문서는 관련이 있지만 클러스터링이 되지 않을 수 있는 모호성 문제가 발생할 수 있다.

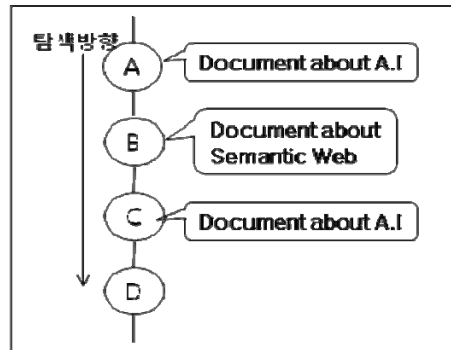


그림 2 클러스터링의 모호성 예제

위에 언급한 문제점을 해결하는 알고리즘은 다음과 같다. 3.1 에서 생성된 웹 그래프에서 임의의 노드를 선택하여 시작 노드로 정하고, 깊이 우선 탐색으로 시작 노드와 에지로 연결 되어 있는 노드들과의 유사도 측정을 한다. 유사도 측정은 Jaccard Coefficient 를 이용한다.

$$sim_{n_1, n_2 \in D} = \frac{|T(n_1) \cap T(n_2)|}{|T(n_1) \cup T(n_2)|} \quad (1)$$

$D$  는 전체 문서집합을 나타내고,  $T(n_i)$ 는 노드에 해당하는 문서에 포함된 단어의 집합이다. 유사도는 0 에서 1 사이의 값을 갖는다. Jaccard Coefficient 는 두 문서에 나타나는 전체 단어 중에 공통으로 포함된 단어의 비율을 계산하여 문서간의 유사도를 측정하는 방법이다.

수식 (1)을 이용하여 계산된 유사도가 임계값 보다 클 경우, 두 문서는 의미적으로 관계가 있다 판단 하여 클러스터를 생성한다. 같은 클러스터에 있는 상위 노드들에 해당하는 문서들에서 중요한 단어라고 판단되는

N 개의 단어를 추출하여 하위 노드에 해당하는 문서의 내용에 붙임으로써 위에서 언급한 문제를 해결한다. 클러스터에서 추출한 중요한 단어의 집합은 클러스터를 대표하는 단어가 될 수 있고, 이러한 단어의 집합을 하위 노드에 붙여 클러스터링을 하면 시작 노드와의 연관성이 적어지는 문제를 해결할 수 있다. 앞에서 언급한 그림 2 의 문제점에서 노드 B 에 노드 A 를 포함하는 클러스터를 대표하는 단어의 집합을 붙이면 노드 A 를 포함하는 클러스터의 주제가 노드 C 까지 이어지므로, 그래프에 모호한 노드가 연결되어 있어 발생하는 문제를 해결할 수 있다. 단어의 개수 N 은 실험을 통하여 정하였으며, N 의 값이 최하위 노드에 해당하는 문서 단어 수의 20%일때 가장 높은 성능을 보였다. 그리하여 본 논문의 실험에서 N 의 값은 하위 노드에 해당하는 문서 단어 수의 20%로 정하였다.

클러스터를 대표하는 단어를 추출하기 위해서 TF-IDF (Term Frequency-Inverse Document Frequency)를 사용한다. TF-IDF 는 문서집합에서, 특정 단어가 문서 내에서 얼마나 중요한 지를 나타내는 통계적 수치이다. 식 (2)는 단어의 가중치를 나타내는 수식이다.

$$w_{t,d} = f_{t,d} \times idf_t \quad (2)$$

$$f_{t,d} = a + (1-a) \frac{tf_{t,d}}{tf_{\max}(d)}, a = 0.5$$

$$idf_t = \log \frac{N}{df_t}$$

계산된 단어의 가중치  $w_{t,d}$  가 큰 단어를 하위 노드에 포함된 단어의 수의 20%만큼 추출하여 최하위 노드에 해당하는 문서의 내용에 붙이고 클러스터링을 반복한다.

## 4. 실험

실험을 하기 위하여 ACM(Association for Computing Machinery, portal.acm.org)의 논문 요약 페이지 1000 개를 수집하였다. 수집된 문서에 대해 불용어(Stopwords) 제거 및 어간처리(Stemming)을 적용하여 문서 별 어휘집(Bag-of-words)을 생성하였다.

실험은 Windows XP 에서, CPU 2.4GHz, Ram 1G 의 시스템 환경에서 실행되었다.

### 4.1 임계값 측정

식 (1) 을 이용하여 수집 된 전체 문서들간의 유사도를 측정 후, 빈도수가 가장 높은 유사도를 임계값으로 사용한다. 임계값은 0 에서 1 사이의 값을 가지며,

임계값이 작으면 문서간의 유사도가 작아도 같은 클러스터로 군집된다. 유사도가 작은 문서들의 집합으로 구성된 클러스터는 여러 분야의 문서 집합으로 구성되기 때문에 클러스터내의 문서의 수가 많아진다. 따라서 전체 클러스터의 수가 줄어든다. 반대로 임계값이 크면, 관련성이 큰 문서들의 집합만으로 클러스터가 생성되기 때문에 클러스터내의 문서의 수가 적어지고, 전체 클러스터의 수가 늘어난다.

가장 적합한 클러스터의 수와 의미적으로 관계가 있는 문서들의 집합으로 클러스터를 생성하기 위하여, 본 논문에서는 빈도수가 가장 높은 유사도를 임계값으로 정하였다. 임계값이 작을수록 클러스터는 일반적인 요소들로 구성되며, 반대로 임계값이 커질수록 클러스터는 상세한 값으로 구성되기 때문에, 유사도의 빈도수가 가장 높은 값은 가장 일반적이며 상세한 값이라 판단하여 임계값으로 설정한다.

불용어 및 어간처리를 적용한 모든 문서의 어휘집 간의 유사도를 측정 후 빈도수가 가장 많은 유사도를 임계값으로 설정하였다. 아래 그림 3 에서 알 수 있듯이, 임계값은 0.23 으로 측정되었다.

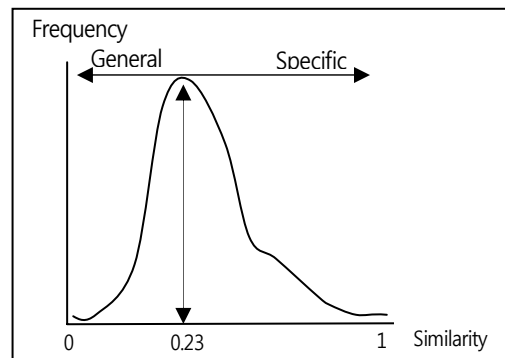


그림 3 실험을 통한 임계값 측정

임계값은 소수점 2 자리까지 계산한다. 소수점 2 자리로 임계값의 범위를 정한 것은, 유사도 빈도수의 분포가 너무 넓게 퍼지지 않고 일정 범위 내에 모이게 해 명확한 임계값을 측정하기 위해서다.

비교 실험에 앞서, 제안한 알고리즘의 가장 좋은 성능을 나타내는 클러스터링을 대표하는 단어의 개수 N 을 구하는 실험을 한다. N 은 실험 시간에 따라 가장 높은 F-Measure 값을 나타내는 값으로 한다.

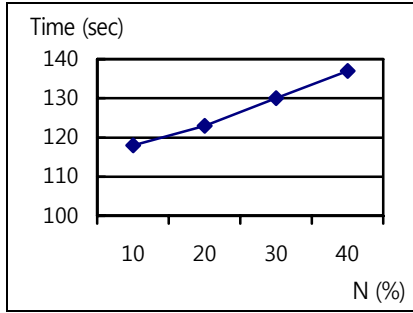


그림 4 N의 값에 따른 실험 시간의 변화

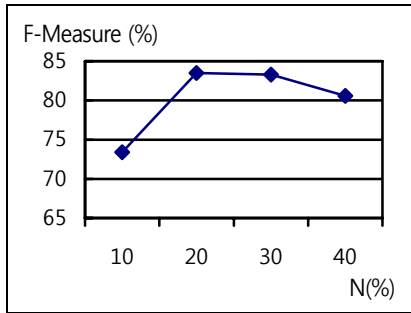


그림 5 N의 값에 따른 F-Measure 변화

클러스터링시 하위 문서에 해당하는 노드에 중요 단어라고 판단되는 단어를 몇 개 붙이는지에 따라서 실험 결과는 달라진다. 단어의 개수 N을 구하기 위하여, N에 따른 실험 시간과 F-Measure를, 그림 4와 그림 5를 통하여 알 수 있다.

그림 4에서 N이 증가할수록, 하위 문서에 더해지는 단어의 개수가 늘어나, 단어를 추출하는 계산 비용과 유사도 측정 등에 대한 계산 비용이 증가하게 되는 것을 알 수 있다. 그림 5에서는 N의 값이 20일 때 가장 높은 F-Measure를 보이는데, 이는 N의 값이 20보다 작을 경우, 클러스터를 대표하는 충분한 수의 단어들이 하위 문서에 더해지지 않아서 클러스터링이 정확히 되지 않았기 때문이다. 반대로 N의 값이 20보다 커질 경우, 의미 없는 단어들도 추출하여 하위 문서에 더해지기 때문에 정확한 클러스터링이 되지 않는다. 문서간의 유사도를 높이기 위해 충분하지 않은 수의 단어 또는 의미 없는 단어의 추가로 클러스터링의 재현율이 낮아지고, 결과적으로 F-Measure 값도 작아지게 된다. 그래서 본 논문의 실험에서는 F-Measure가 가장 높고 시간이 적게 걸리는 20%를 N 값으로 설정한다.

#### 4.2 성능 비교 및 분석

비교 실험을 위하여 전체 문서간의 유사도를 계산하는 그물망 그래프 기반 클러스터링 방법, 시작 노드와 에지로 연결되어 있는 노드들의 유사도를 계산 후 임계값 보다 큰 경우 클러스터링 하는 하이퍼링크를 이용한 깊이 우선 탐색 기반 클러스터링 방법과 본 논문에서 제안한 N 개의

단어를 붙인 그래프 기반 클러스터링 방법의 성능 및 시간을 평가한다. 성능의 평가는 재현율과 정확율을 이용한 F-Measure를 이용한다.

|                | 그물망<br>그래프 기반<br>방법 | 하이퍼링크<br>기반 방법 | 제안한<br>방법<br>(N=20) |
|----------------|---------------------|----------------|---------------------|
| 시간평가(ms)       | 186000              | 126000         | 132000              |
| F-Measure      | 86.2%               | 67.9%          | 84.4%               |
| 재현율(Recall)    | 87.0%               | 58.0%          | 83.8%               |
| 정확율(Precision) | 85.6%               | 82.2%          | 85.2%               |

표 2 성능 비교 평가

표 2 와 같이 제안한 클러스터링 방법은 하이퍼링크를 이용한 깊이 우선 탐색 기반 클러스터링 방법과 시간을 비교했을 때 약 4% 증가하였고, 그물망 그래프 기반 클러스터링 방법보다 29% 가량 감소하였다. F-Measure 를 비교하였을 때, 하이퍼링크 기반 클러스터링 방법보다 16.5% 증가하였고, 그물망 그래프 기반 클러스터링 방법과 유사한 성능을 보여주었다.

클러스터링을 대표하는 단어의 집합을 하위 노드에 붙임으로써, 하이퍼링크 기반 클러스터링 알고리즘 보다 노드 비교의 수가 증가하였고, 문서간의 유사도를 높였다. 그로 인해서 4%의 클러스터링 시간은 증가하였지만, 약 25.8%의 재현율을 높였다. 그리고 약 3%의 정확율의 증가를 보였는데, 이는 N 개의 단어를 붙임으로써 클러스터링 된 적합한 노드들의 수가 증가되었다는 것을 나타낸다. 결과적으로 하이퍼링크 기반 클러스터링 알고리즘과 비교하여 16.5%의 F-Measure 의 증가를 보였다.

그물망 그래프 기반 클러스터링 알고리즘과 비교하였을 때, 노드들간의 비교 기회가 상대적으로 적기 때문에 관련은 있지만 문서 비교를 하지 않아 클러스터링이 되지 않는 문서가 존재한다. 그래서 재현율이 3.2% 감소하였지만 정확율은 큰 변화가 없었다. 결과적으로 그물망 그래프 기반 클러스터링과 F-Measure 를 비교하였을 때, 차이가 크지 않아 성능을 유지함을 알 수 있다.

#### 5. 결론

본 논문에서는 기존의 웹 문서 클러스터링 방법의 성능을 유지하면서 속도 문제를 개선하고자 하였다. 속도 문제를 해결하기 위하여 웹의 구조적 정보인 하이퍼링크를 이용하여 방향성이 없는 그래프를 생성 및 탐색하였다. 하이퍼링크는 문서간의 연관성을 포함하고 있는 정보이므로, 이를 이용하여 문서간의 유사도 비교 횟수를 줄여 클러스터링 시간을 감소 시켰다. 성능을 유지하기 위하여 생성된 클러스터를 대표하는 단어를 추출하여 하위 노드에 붙이는

방법을 제안 하였다. 하위 노드에 클러스터를 대표하는 단어를 붙임으로써, 탐색 깊이가 깊어질수록 발생하는 시작 노드와의 연관성이 적어지는 문제와 적절하지 않는 노드가 있을 때의 모호성 문제를 해결 하였다.

제안한 알고리즘은 빠른 속도를 요구하는 웹 문서 클러스터링에서 기존의 그물망 그래프 기반 클러스터링의 성능을 유지하면서 계산비용을 감소하였다. 실험을 통하여 하이퍼링크 기반의 클러스터링 보다 4%의 클러스터링 시간이 증가하였지만, 17% F-Measure 의 증가를 보였고, 그물망 그래프 기반의 클러스터링과 비교하였을 때, 유사한 성능을 나타내며 29% 가량의 시간 소비를 감소하였다. 그렇기에 제안한 알고리즘은 웹 문서 클러스터링에 적합한 클러스터링 알고리즘이다.

## 참고문헌

- [1] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In KDD Workshop on Text Mining, 2000.
- [2] T. Kanungo, N. Netanyahu, A. Wu. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2002
- [3] Tian Zhang, Raghu Ramakrishnan, Miron Nivny. BIRCH: A New Data Clustering Algorithm and Its Applications. Data Mining and Knowledge Discovery, 1, 141-182 (1997)
- [4] C. Benjamin, M. Fung, Wang Ke, Ester Martin. Hierarchical document clustering using Frequent Itemsets. In Proceedings of the 3rd SIAM International Conference on Data Mining, 2003.
- [5] O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. In Research and Development in Information Retrieval, pages 46-54, 1998
- [6] D. Crabbtree, X. Gao, P. Andrae. Improving Web Clustering by Cluster Selection. International Conference on Web Intelligence (WI' 05)
- [7] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, 1996, pp. 226-231.
- [8] Ankerst M., Breunig M. M., Kriegel H.-P., Sander J. OPTICS: Ordering Points To Identify the Clustering Structure, Proc. ACM SIGMOD Int. Conf. on Management of Data, Philadelphia, PA, 1999.

[9] N. Memarsadeghi, David M. Mount, Nathan S. Netanyahu, Jacqueline Le Moigne. A Fast Implementation of the ISODATA Clustering Algorithm. International Journal of Computational Geometry & Applications