The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 14, No. 5, pp.189-196, Oct. 31, 2014. pISSN 2289-0238, eISSN 2289-0246

http://dx.doi.org/10.7236/JIIBC.2014.14.5.189
JIIBC 2014-5-26

한글 위키피디아를 이용한 트위터 문서의 주제별 클러스터링 기법

Topical Clustering Techniques of Twitter Documents Using Korean Wikipedia

장재영*

Jae-Young Chang*

요 약 최근 들어 트위터와 같은 SNS 환경에서 검색의 필요성이 증가하고 있다. 트위터 검색을 지원하기 위해서는 다량으로 검색된 문서를 주제별로 분류하는 클러스터링 기법이 필요하다. 하지만 트위터의 특성상 단순한 클러스터링 기술을 그대로 적용하기에는 많은 제약이 따른다. 본 논문에서는 이를 극복하기 위해 트위터 환경에 적합한 클러스터링 기법을 제안한다. 제안된 기법에서는 한글 위키피디아를 이용하여 각 트위터 문서에 대한 특징 벡터를 보강하고 각 특징들의 가중치를 재계산하는 방법을 이용하였다. 또한 한글 트위터 문서를 대상으로 실험을 실시하고 기존 기법과의 성능 비교를 통해서 제안된 기법의 유용성을 증명하였다.

Abstract Recently, the need for retrieving documents is growing in SNS environment such as twitter. For supporting the twitter search, a clustering technique classifying the massively retrieved documents in terms of topics is required. However, due to the nature of twitter, there is a limit in applying previous simple techniques to clustering the twitter documents. To overcome such problem, we propose in this paper a new clustering technique suitable to twitter environment. In proposed method, we augment new terms to feature vectors representing the twitter documents, and recalculate the weights of features using Korean Wikipedia. In addition, we performed the experiments with Korean twitter documents, and proved the usability of proposed method through performance comparison with the previous techniques.

KeyWords : SNS, Twitter, Clustering, Wikipedia, Feature Vector

Ⅰ. 서 론

2000년대 이후 웹 2.0과 인터넷 애플리케이션의 발전과 더불어 트위터(Twitter)로 대표되는 SNS(Social Network Service)의 등장은 인터넷 사용 환경에 큰 변화를 가져왔다. 다수의 사용자, 정보의 빠른 전파, 사용의 편리성으로 대표되는 SNS는 이제 기존 매체를 대체하는 수준에 이르렀다. SNS가 대중화되기 이전의 사용자는 주로 인터넷 포털 사이트를 이용하여 문서를 검색하거나 온라인 업무를 처리하는 등, 특정인들에 의해 제공되는 정보를 사용하는 수준에 머물렀다. 그러나 SNS를 사용함으로서 개방적인 웹 환경을 기반으로 한 네티즌들의

*정회원, 한성대학교 컴퓨터공학과
접수일자 : 2014년 7월 15일, 수정완료 : 2014년 9월 15일
게재확정일자 : 2014년 10월 10일

Received: 15 July, 2014 / Revised: 15 September, 2014
Accepted: 10 October, 2014
*Corresponding Author: jychang@hansung.ac.kr
Dept. of Computer Engineering, Hansung University, Korea

정보공유와 참여가 가능하게 된 것이다.

이러한 환경에서 SNS에서의 검색에 대한 관심이 대두되고 있으며, 최근 대부분의 포털에서는 트위터로 대표되는 SNS에 대한 검색 기능도 제공하고 있다. 그러나 기존의 포털에서 제공되는 검색 기능은 사용자의 의도와는 무관하게 단순히 검색어(query)에 대해 최근에 등록된 순서만으로 그 결과를 제공하고 있다. 그러나 이러한 방식은 게시글들(postings)의 중요도나 특성들을 반영하지 못하고 있어 사용자에게 큰 도움이 되지 못하고 있다. 이를 해결하기 위해서 최근에 SNS 중에서 특히 트위터 문서를 검색하는 다양한 방법이 제안되었다. 예를 들어 [1]에서는 트위터 상에서 작성자의 영향력을 정량적으로 평가하는 방법을 제안하였는데, 이 방법을 이용하면 작성자의 영향력 순으로 게시글을 정렬할 수 있다. 또한 [2]에서는 리트윗(rewteet) 빈도나 이외의 여러 가지 특징들을 적용하여 검색결과를 제공하는 방법을 제안하였다. 이외에도 검색어와의 관련성이나 중요도에 따라 검색된 게시글들을 랭킹하는 방법을 제안하였다 [3][4].

또 하나의 방법은 검색결과를 주제(topic)별로 클러스터링(clustering)하여 제공하는 것이다. 검색된 게시글들을 주제별로 분리하게 되면 검색결과를 조직화된 형태로 파악할 수 있어 전반적인 내용을 쉽게 파악할 수 있게 된다. 또한 요약(summarization), 추천(recommendation), 이벤트 검색(event detection) 등 문서 클러스터링의 응용분야에 다양하게 활용할 수도 있다.

본 논문에서는 트위터 환경에서 게시글들을 주제별로 클러스터링하는 기법을 제안한다. 기존에 문서 클러스터링을 위한 많은 기법들이 이미 제안되어 왔고 이러한 기법들이 현재 광범위하게 응용되고 있다. 그러나 이러한 기법들을 트위터에 직접적으로 적용하기에는 한계가 있다. 정보 제공자가 제한되지 않는 트위터는 그 특성상 기존의 웹문서와는 차별되는 몇 가지 특징을 갖고 있다[2]. 우선 문서들은 140bytes이하의 단문(short document)으로 구성된다. 따라서 클러스터링을 위한 특징 벡터(feature vector)가 지나치게 희박(sparse)하여 문서의 특징을 표현하기에는 정보가 많이 부족하다. 또한 많은 문서들이 객관적인(objective) 사실뿐만 아니라 주관적인 의견(subjective opinion)들이 포함되어 있는 등 다양한 종류의 문서들이 혼재한다[5]. 마지막으로 기존의 웹 문서와는 달리 문법적으로 정제되지 않은 문서들이 비율이 매우 높다. 이러한 특징들은 트위터 문서가 불특정 다수에 의해 대량으로 생성된다는 것에 기인한다. 현재 트위터는 전 세계적으로 매일 2억 개가 넘은 게시글이 등록되고 있다.

따라서 트위터 문서를 효율적으로 클러스터링하기 위해서는 트위터의 특성을 고려한 특별한 기법이 요구된다. 예를 들어 [6]에서는 트위터의 해시태그(hashtag)를 이용하여 동일 해시태그를 갖는 문서들을 하나의 가상 문서(virtual document)로 만들어 클러스터링을 수행하는 방법을 제안하였다. [7]에서는 위키피디아(Wikipedia)를 이용하여 개념(concept) 정보를 클러스터링에 활용하는 기법을 제안하였다.

본 논문에서는 [7]과 유사하게 위키피디아를 이용한 클러스터링 기법을 제안한다. 단, [7]에서는 위키피디아를 이용하여 트위터 문서에 나타나는 단어들의 개념을 파악하고 이를 문서간의 거리 계산에 이용한 반면, 본 논문에서는 위키피디아를 문서의 특징을 보강하는데 이용하였다. 이를 위해 첫째로 문서에 나타나는 단어에 해당하는 위키피디아 문서의 주요 단어들을 문서의 특징 벡터에 추가하여 클러스터링을 수행하였다. 또한 단순히 TF-IDF를 이용하여 단어(특징)들의 가중치(weight)를 계산하는 기존의 방법을 보강하였다. 즉, 각 단어가 문서에서 갖는 중요도 계산하기 위해 해당 위키피디아 문서의 크기나 위키피디아 문서간의 in-link 수를 이용하여 가중치를 계산하였다.

본 논문에서는 한글 트위터를 대상으로 하였으며, 또한 한글 위키피디아 문서를 활용하였다. 한글 위키피디아는 영문 위키피디아에 비해서 문서의 질이나 양적인 면에서 많이 부족한 것은 사실이나 2014년 4월 현재 27만 개 이상의 문서를 보유하고 있어 활용에는 큰 문제가 없을 것으로 판단된다.

본 논문에서 제안한 기법의 유용성을 증명하기 위해서 실험을 실시하였다. 실험은 6개의 주제로 나눈 트위터 문서에 대해서 클러스터링을 실시하여 위키피디아를 활용하기 않은 기존의 방법과 비교하여 정확도를 비교하였으며, 다양한 요소의 변화에 따른 정확도를 관찰하였다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 관련 연구에 대해 소개하고, 3장에서는 위키피디아를 이용하여 특징 가중치를 계산하는 방법과 특징 보강 방법을 소개한다. 4장에서는 실험결과를 제시하고, 마지막으로 5장에서는 결론을 맺는다.

II. 관련연구

문서 클러스터링은 텍스트 마이닝(text mining)이 등장한 초기부터 많은 연구가 진행되어 왔다. 그러나 트위터를 대상으로 한 클러스터링 연구는 비교적 최근에 시작되었다. 트위터 환경에서 클러스터링이 어려운 점은 문서가 매우 작기 때문이다. 따라서 문서 내에 단어의 빈도만으로 특징 벡터를 구성하는 전통적인 TF-IDF 방식[8]으로는 좋은 성능을 기대할 수 없다.

[6]에서는 해시태그를 이용한 트위터 문서의 클러스터링 기법을 제안하였다. 이 방법에서는 동일 해시태그를 갖는 게시글들을 하나의 가상 문서로 묶어서 이들 을 대상으로 클러스터링을 수행한다. 그리고 그 결과를 이용하여 새롭게 추가되는 문서를 가장 근접한 클러스터에 배정하는 방법을 제안하였다. 이외에도 [9][10][11] 등에서 해시태그를 이용한 트위터의 클러스터링 기법을 제안하였다.

[12]에서는 트위터에 대한 질의 결과 집합을 주제별로 분류하는 방법을 제안하였다. 여기서는 게시글의 구문(phase)중에서 확률적으로 일반적인 구문과 현저히 차이를 갖는 것을 탐색하여 이를 새로운 주제로 결정하였다. 그런 다음 주제별 유사도를 측정하여 이들을 병합(merge)하는 방법을 제안하였다. 그러나 이 연구에서는 자세한 실험결과를 제시하지 못하였다.

클러스터링에 있어서 단문으로 구성된 트위터는 많은 제약이 따를 수밖에 없다. 이를 극복하기 위해 많은 연구에서 외부의 정보를 클러스터링에 활용하는 노력이 있어 왔다. 대표적인 것이 위키피디아를 이용한 클러스터링 기술이다[7][13][14][15]. 예를 들어 [7]에서는 게시글의 각 단어에 대해서 의미를 파악하기 위해 위키피디어 문서 중에 동일한 단어가 존재하는 링크의 타겟 문서를 해당 단어의 개념으로 취급하였다. 또한 문서간의 거리를 계산하기 위해서 코사인 유사도(consine similarity)를 변형하여 위키피디아 문서간의 거리를 계산하는 새로운 방법을 제안하였다. 이 방법은 위키피디아를 이용했다는 점에서는 본 연구에서 유사하나, [7]에서는 유사도 계산에 이를 활용하였고, 본 연구에서는 특징 벡터를 보강하거나 가중치를 계산하는데 이용하였다는 점에서 차별성이 있다.

III. 트위터 클러스터링 기법

본 장에서는 본 논문이 제안하는 트위터 문서의 클러스터링 기법에 대해 소개한다. 본 논문이 제안하는 기법의 핵심은 위키피디아 문서를 이용하여 각 트위터 문서의 특징 벡터에 특징들을 새롭게 추가하고, 특징들의 가중치를 재계산하는 것이다. 단, 특징의 추가와 가중치 재계산은 독립적으로 이루어지는 것이 아니라 특징이 추가될 때 마다 가중치를 재계산하는 방식으로 이루어지므로 이들을 결합한 알고리즘의 개발도 필요하다. 따라서 본 장의 후반부에는 특징 벡터를 재구축하기 위한 알고리즘을 소개한다.

1. 특징 가중치 계산

특징 벡터에서 특징의 가중치는 각 특징이 해당 문서에서 어느 정도 중요도를 갖는가를 정량적으로 표현한 값이다. 일반적으로 문서 분류나 클러스터링을 위한 특징 벡터의 가중치는 문서에 나타난 특징, 즉 단어의 TF나 TF-IDF로 표현한다. 그러나 단문으로 구성된 트위터의 경우 각 특징의 출현 빈도가 1보다 큰 경우가 거의 없기 때문에 이를 보완하기 위한 방법이 필요하다.

본 논문에서는 각 특징이 갖는 개념(concept)의 중요도에 따라 가중치를 결정한다. 특징의 개념은 위키피디아를 활용하며, 특징의 중요도는 해당 특징이 위키피디어 문서에서 갖는 중요도를 이용하여 간접적으로 추정한다. 이를 위해서 본 논문에서는 두 가지 가정을 한다. 첫째, 위키피디아에서는 문서의 크기가 클수록 중요한 개념이라고 추정한다. 위키피디아를 보면 문서마다 크기가 다른데 자주 검색되고 중요한 키워드일수록 해당 문서의 크기가 큰 것을 알 수 있다. 둘째로 위키피디아에서 각 문서에 대한 in-link의 수가 많을수록 해당 문서가 중요한 개념이라고 추정한다. 여기서 in-link의 수란 다른 위키피디아 문서에서 해당 문서로 링크된 개수를 의미한다. 이 가정은 중요한 문서일수록 다른 문서에서 해당 문서를 자주 참조한다는 사실에 기초한다. 따라서 본 논문에서는 이 두 가지 가정을 이용하여 특징의 가중치를 계산한다.

우선 트위터 문서 d의 특징 벡터 중 하나인 특징 t에 대해서 t에 대한 위키피디아 문서를 $d_{wiki}(t)$라고 가정하자. 이때 위키피디아 문서 크기를 반영한 특징 t의 중요도 $size(t)$는 다음과 같이 정의할 수 있다.

$$size(t) = \frac{|d_{wiki}(t)|}{\text{최대문서크기}} \quad (1)$$

이 식에서 $|d_{wiki}(t)|$ 는 t에 대한 위키피디아 문서의 크기를 나타내며, 최대문서크기란 위키피디아 문서 중에 가장 큰 문서의 크기를 나타낸다. 이와 유사하게 in-link의 수를 반영한 t의 중요도 *in_link*(t)는 다음과 같이 정의된다.

$$in_link(t) = \frac{d_{wiki}(t)\text{에 대한}\, in-link\text{수}}{\text{최대}\, in-link\text{수}} \quad (2)$$

이 식에서 최대 in-link수란 위키피디아 문서 중에서 in-link의 수가 최대가 되는 문서의 in-link 수를 나타낸다.

식 (1)와 식 (2)를 이용하면 특징 t에 대해서 위키피디어 문서에서 t의 중요도 $wiki(t)$를 다음과 같이 계산할 수 있다.

$$wiki(t) = \alpha \times size(t) + (1-\alpha) \times in_link(t) \quad (3)$$

여기서 α는 0부터 1의 값을 가지며, $size(t)$와 $in_link(t)$의 비중을 결정하기 위한 요소로 사용된다. 참고로 4장의 실험에서는 α를 0.5로 고정하여 실시하였다.

마지막으로 식 (3)를 이용하면 t에 대한 최종적인 가중치를 계산할 수 있다. 위키피디아를 이용하기 전의 초기의 특징 가중치가 $w_0(t)$라고 가정하자. 앞서 설명한 대로 트위터 문서는 매우 짧은 문서이므로 TF를 사용한다면 이 값은 대부분 0이나 1의 값을 가질 것이다. 이때 t에 대한 특징 가중치는 이 값에 위키피디아에서 t의 중요도 $wiki(t)$를 반영하여 계산된다. 즉, t에 대한 특징 가중치 $w(t)$는 최종적으로 다음과 같이 계산된다.

$$w(t) = w_0(t) \times wiki(t) \quad (4)$$

2. 특징 보강

앞서 설명한 바와 같이 단문으로 구성된 트위터 문서는 정보의 양이 많지 않아 단독으로는 구성된 특징 벡터만으로는 클러스터링 결과를 신뢰할 수 없다. 본 논문에서는 이를 극복하기 위해 위키피디아의 주요 단어들을 특징 벡터에 추가하는 방법을 사용하여 이 문제를 해결하였다. 즉, 트위터 문서 d의 특징 벡터 $f = (t_1, ..., t_n)$에 존재하는 각 특징 $t_i (1 \le i \le n)$에 대해서, t_i에 대한 해당 위키피디아 문서의 주요 단어들을 d의 특징 벡터에 추가한다.

단어를 추가하기 위한 가장 좋은 방법은 단어 t_i에 대한 위키피디아 문서 $d_{wiki}(t_i)$에서 상위의 TF-IDF 값을 갖는 단어들을 추가하는 것이다. 여기서 위키피디아 문서의 각 단어들에 대한 TF 값을 계산하는 것은 큰 문제가 되지 않으나, 문서의 양이 너무 클 경우에는 IDF 값을 계산하는데 많은 시간이 소요될 수 있다. 따라서 이 경우에는 상위의 TF 값을 갖는 단어만을 추가하는 것도 하나의 대안이 될 수 있다.

또 하나 고려해야할 점은 문서 d의 모든 특징들에 대해서 동일한 개수의 단어들을 추가하는 것이 아니라 현재의 가중치 값에 비례하여 추가하는 것도 고려할 수 있다. 즉, d의 특징 벡터 $f = (t_1, ..., t_n)$에 대해서 k개의 추가한다면 각 $t_i (1 \le i \le n)$에 대해서 새롭게 추가되는 단어의 개수는 다음과 같이 계산될 수 있다.

$$\frac{w(t_i)}{\sum_{j=1}^{n} w(t_j)} \times k \quad (5)$$

이 때 새롭게 추가되는 특징들에 대해서 기본 가중치를 부여해야한다. 또한 기존의 특징들과 새롭게 추가된 특징들에 대한 가중치도 재조정할 필요성도 있다. 이 문제에 대해서는 다음절에서 다룬다.

3. 특징 재구성 알고리즘

1절과 2절서 각각 설명한 특징 가중치 계산 방법과 특징 추가 방법은 독립적으로 수행되는 것이 아니고 상호 유기적으로 병합되어야한다. 본 절은 이를 해결하기 위한 특징 재구성 알고리즘을 소개한다.

알고리즘은 그림 1과 같다. 이 그림에서 입력 값인 $f = (t_1, ..., t_n)$ 는 트위터 문서에 대한 최초의 특징 벡터를 나타내며, r은 2절에서 설명한 특징추가 방법에 의해서 한 번에 추가되는 특징들의 비율을 위미한다. 예를 들어 n이 100이고 r이 0.1이라면 한번에 10개의 특징들이 새롭게 추가된다는 것을 의미한다.

```
function Rebuilding_Feature(f, r)
input  : original feature vector f = (t_1, ..., t_n)
         augmentation ratio r
output :  new feature vector f'

 1: f' = f
 2:
 3: // feature weighting
 4: for all t_i in f'
 5:          w(t_i) = w_0(t_i) × wiki(t_i)
 6: end
 7:
 8: repeat
 9:    // feature augmentation
10:    for all t_i in f'
11:          find significant terms k_i = {k_i1, ..., k_im_i} in Wikipedia for t_i
                                          where m_i = w(t_i) / Σ_{j=1}^{n} w(t_j) × |f'| × r
12:          f' = f' ∪ k_i  with w_0(k_ij) = 1  (1 ≤ j ≤ m_i)
13:    end
14:    // feature weighting(reweighting)
15:    for all t_i in f'
16:          w(t_i) = w_0(t_i) × wiki(t_i)
17:    end
18: until |f'| stabilize or |f'|  is less than a threshold β
```

그림 1. 특징 재구성 알고리즘
Fig. 1. Feature reorganization algorithm

이 알고리즘의 출력 결과인 f'는 최종적으로 재구성된 특징 벡터를 나타낸다. 우선 알고리즘에서 라인 4 ~ 5에서는 초기 특징 벡터에 대해서 수식 (4)를 이용하여 특징들의 가중치를 재계산한다. 다음으로 라인 8 ~ 18까지 특징 가중치 계산과 특징 추가가 반복적으로 진행된다. 라인 10 ~ 12는 특징들을 추가하는 방법을 보여준다. 우선 수식 (5)를 이용하여 각 특징별로 그 중요도에 따라 추가될 새로운 특징들의 수를 계산하고, 위키피디아의 주요 단어들을 추출한다. 이렇게 구성된 새로운 특징 집합인 $k_i = \{k_{i1},...,k_{im_i}\}$는 기존의 특징 집합 f'에 추가된다. 이 때 라인 12에서 보는 바와 같이 새롭게 추가된 특징들에 대해서 가중치을 모두 1로 초기화 하였다. 여기서 1이라는 가정은 이 알고리즘의 입력 값인 최초의 특징 벡터 $f = (t_1, ..., t_n)$ 의 각 특징들이 초기 값으로 모두 1을 갖는 다는 가정 하에 결정된 것이다. 만약 다른 방법으로 초기 가중치를 결정한다면 그에 따라 달라질 수 있다.

다음으로 라인 15 ~ 16에서는 새롭게 구성된 특징 벡터에 대해서 가중치를 재계산한다. 이때 가중치의 재계산은 바로 이전에 재계산된 값을 기준으로 하는 것이 아니라 f'의 각 특징 t의 최초 가중치 값인 $w_0(t)$를 이용하여 재계산한다.

마지막으로 이 알고리즘은 f'의 전체 특징 수인 $|f'|$가 안정화되거나, 그렇지 않을 경우 특정 임계값인 β보다 작을 때까지 반복한다. $|f'|$ 의 안정화 여부는 r값에 따라서 달라질 수 있다. 라인 11을 보면 매 루프마다 특징들의 개수가 r의 비율만큼 증가되므로 무한히 증가되는 것처럼 보일 수 있다. 그러나 라인 11을 수행할 때 이전 루프에서 이미 존재하던 특징들에 대해서는 그 전에 이미 새로운 특징들이 추가되었으므로, 다음 루프 때는 이전 루프에서 추가된 특징들에 대해서만 위키피디아를 이용하여 새로운 특징들이 추가된다. 따라서 r이 작다면 매 루프마다 새롭게 추가되는 특징의 수는 점점 적어지

게 되어, 언젠가는 더 이상의 특징이 추가되지 않고 안정화된다. 예를 들어 각 특징들의 가중치가 모두 동일하게 계산된다면, 최종 특징 개수는 다음과 같다.

$$\sum_{i=0}^{\infty}(r^i \times |f|) \tag{6}$$

이 식을 보면 r이 1일 경우 매 루프마다 $|f|$ 씩 선형적으로 증가한다. 이 경우에는 특징들의 수가 무한히 증가할 수 있으므로 특정 임계값이 넘을 경우 반복을 중지해야 한다.

최종적으로 추가되는 특징의 수를 결정하는 것은 실험적으로 결정해야한다. 추가되는 특징의 수가 기존의 특징 수에 비해 작을 경우에는 클러스터링의 성능 향상을 기대하기 어려울 것이다. 반면에 특징 벡터가 지나치게 커질 경우에는 본래의 특징 벡터에 오히려 왜곡이 발생하여 클러스터링 성능이 더 떨어질 가능성이 있다. 이 문제에 대해서는 다음 장의 실험에 다시 다룬다.

Ⅳ. 성능 분석

본 논문에서 제안한 클러스터링 기법의 성능을 평가하기 위해서 실험을 수행하였다. 실험은 한글 트위터 문서를 대상으로 하였다. 2014년 5월 이후에 등록된 트위터 문서 중에서 정치, 경제, 스포츠, 연예, 게임에 관련된 게시글들을 각각 200개씩, 총 1,000개의 문서를 수집하였다. 문서간의 유사도는 코사인 유사도(Cosine similarity)를 이용하였으며, 클러스터링 알고리즘으로는 계층 클러스터링(hierarchical clustering) 알고리즘을 이용하였다.

우선 그림 2는 r의 값에 따라 증가되는 특징들의 수를 나타낸다. 이 표에서 r이 0일 때, 즉, 특징 추가가 이루어지지 않은 초기의 경우에는 문서 당 특징의 평균수는 9개 정도이다. r이 증가함에 따라 추가되는 특징들의 수도 증가되는데 r이 0.4까지는 그 수가 비교적 선형적으로 증가하는 반면 0.5보다 커질 경우 그 수가 급격히 증가하는 것을 관찰 할 수 있다. 그 이유는 식 (6)에서 보는 바와 같이 r이 커짐에 따라 그 수가 기하급수적으로 증가하기 때문이다. 본 실험에서는 r이 1 이상인 경우에서도 증가되는 특징수를 측정하였다. 그러나 1보다 큰 경우에는 그림 1의 알고리즘에서 증가되는 $|f'|$ 가 안정화되지 않고 무한히 증가하는 것으로 나타났다. 따라서 이 경우에는 적절한 임계값을 설정하여 그 이상 루프를 진행하지 않고 중단하는 방법을 고려해야한다.

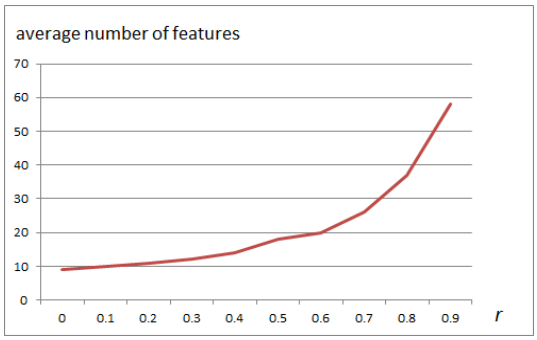

그림 2. 특징 수의 증가 추이
Fig. 2. Increasing trend of the number of features

클러스터링 결과의 평가 지표는 순도(purity)와 엔트로피(entropy)를 이용하였다[16]. 순도는 클러스터들의 순수성 정도를 나타내는 것으로 0에서 1의 값을 가지며 1에 가까울수록 성능이 좋다는 것을 의미한다. 실험 결과로 나온 클러스터 집합을 $C=\{C_1,...,C_n\}$이라 하고, 참조 집합을 $G=\{G_1,...,G_n\}$이라 하자. 그리고 $P_i(G_j)$를 C_i의 문서 중에서 G_j의 비율이라고 가정하자. 그러면 순도는 다음의 수식으로 계산된다.

$$purity(C) = \sum_{i=1}^{n}\frac{|C_i|}{|C|}max_j P_i(G_j) \tag{7}$$

여기서 $\max_j P_i(G_j)$는 C_i의 문서에서 G_j의 비율이 가장 큰 값을 나타내며, 모든 클러스터에 대해 이를 계산한 후 각 클러스터의 크기비율에 따라 평균을 구한다. 반면에 엔트로피는 불순도를 나타내는 것으로 순도와는 달리 0에 가까울수록 성능이 좋다는 것을 의미한다. 엔트로피는 다음의 수식으로 계산된다.

$$entropy(C_i) = -\sum_{j=1}^{n}P_i(G_j)\log_2 P_i(G_j) \tag{8}$$

$$entropy(C) = \sum_{i=1}^{n}\frac{|C_i|}{|C|}entropy(C_i) \tag{9}$$

식 (9)에서 보는 바와 같이 엔트로피도 식 (8)을 이용하여 각 클러스터에 대한 엔트로피를 먼저 구한 후에 각 클

러스터의 크기비율에 따라 평균을 구한다.

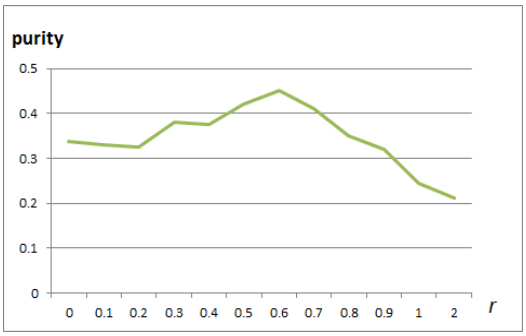

그림 3. 특징 수에 따른 순도의 변화
Fig. 3. Purity variation according to the number of features

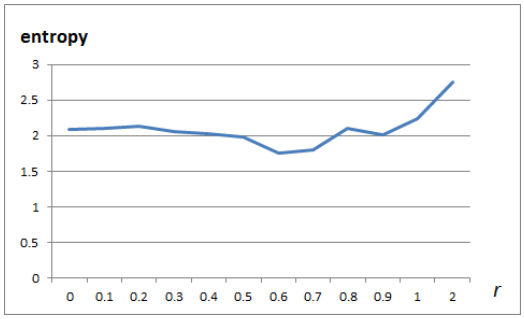

그림 4. 특징 수에 따른 엔트로피의 변화
Fig. 4. entropy variation according to the number of features

그림 3과 4는 각각 r의 변화에 따른 순도와 엔트로피의 결과를 보여준다. 이 실험에서 r이 1이상일 경우에는 보강되는 특징수가 무한히 증가하므로 그림 2의 증가추세를 고려하여, r이 1일 경우에는 100개, 2일 경우에는 200개로 임계값을 정하여 특징들을 보강하였다. 이 그림에서 보는 바와 같이 초기에는 r이 증가할수록 순도와 엔트로피가 모두 점차 좋은 성능을 보이고 있다. 그러나 0.6보다 커지게 되면 오히려 성능이 떨어지게 되며, r이 0.9 이상에서는 오히려 특징 보강을 하지 않는 최초의 클러스터링 결과보다 성능이 더 떨어지는 것을 관찰할 수 있다. 그 이유는 r이 작을 경우에는 클러스터의 성격을 규정짓는 관련 특징들이 점차 많이 보강되므로 성능이 좋아지게 된다. 반면에 점차 r이 일정 수준보다 클 경우에는 그림 2에서 보는 바와 같이 특징들이 기하급수적으로 보강되는데 이들은 대부분 클러스터의 성격과 맞지 않는 관련 없는 특징들이므로 오히려 성능이 떨어지는 것으로 해석된다. 따라서 특징은 일정 수준까지만 보강하는 것이 성능을 극대화할 수 있는 것으로 결론을 내릴 수 있다. 본 실험에서는 최적의 r값이 r이 0.6일 경우이며, 이 때 보강되는 특징의 수는 초기의 특징 수에 비해 대략 2배가 되는 경우이다.

V. 결 론

본 논문에서는 트위터 환경에 적합한 클러스터링 기법을 제안하였다. 트위터는 비교적 단문으로 구성되어 순수한 트위터 문서만으로는 정보의 양이 부족하여 좋은 클러스터링 성능을 기대할 수 없다. 이를 극복하기 위해서 제안된 기법에서는 한글 위키피디아를 이용하여 각 트위터 문서에 대한 특징 벡터를 보강하고 특징들의 가중치를 재계산하는 방법을 이용하였다. 실험 결과 제안된 방법으로 특징을 보강하는 기법이 순수한 특징만을 사용하는 것보다 더 좋은 성능을 나타내는 것을 확인할 수 있었다. 하지만 일정 수준 이상으로 지나치게 특징을 보강할 경우에는 오히려 각 클러스터의 성격을 훼손할 수 있으므로 최적의 보강 수준을 결정하는 것이 중요하다고 하겠다.

References

[1] J. Weng and Q. He, TwitterRank: Finding Topic-sensitive Influential Twitterers, Proceedings of ACM international conference on Web search and data mining conference, 2010.

[2] J.-Y. Chang, An Evaluation of Twitter Ranking Using the Retweet Information, Journal of Society for e-Business Studies, Vol. 17, No. 2, 2012.

[3] R. Nagmoti and M. D. Cock, Ranking Approach for Microblog Search, Proceedings of Web Intelligence-Intelligent Agent Technology conference, 2010.

[4] H. W. Lauw, A. Ntoulas and K. Kenthapadi, Estimating the Quality of Postings in the Real-time Web, Proceedings of WSDM 2010 Workshop on Search in Social Media, 2010.

[5] J.-Y. Chang, Automatic Retrieval of SNS Opinion Document Using Machine Learning Technique, The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 13, No. 5, October 2013.

[6] O. Tsur, A. Littman, and A. Rappoport, Efficient Clustering of Short Messages into General Domains, Proceedings of 7th International AAAI Conference on Weblogs and Social Media (ICWSM), 2013.

[7] T. Xu, and D. W. Orad, Wikipedia-based Topic Clustering for Microblogs, Proceedings of the American Society for Information Science and Technology, 2011.

[8] G. Salton, A. Wong, and C. S. Yang, A Vector Space Model for Automatic Indexing, Communications of the ACM, Vol. 18, No. 11, 1975.

[9] J. Yang, and J. Leskovec, Patterns od Temporal Variation in Online Media, Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 177-186, 2011.

[10] D. Romero, B. Meeder, and J. Kleinberg, Differences in the Mechanics of Information Diffusion Across Topics:Idioms, Political Hashtags, and Complex Contagion on Twitter, Proceedings of the 20th International Conference on World Wide Web, pp. 695-704, 2011.

[11] X. Zhao, and J. Jiang, An Empirical Comparision of Topics in Twitter and Traditional Media, Technical Paper Series, Singaapore Management University School of Information Systems, 2011.

[12] B. O'Connor, M. Krieger, and D. Ahn, TweetMotif: Exploratory Search and Topic Summarization for Twitter, Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media, 2010.

[13] M. Michelson, and S. A. Macskassy, Discovering Users Topics of Interest on Twitter: A First Look, Proceedings the Fourth Workshop on Analytics for Noisy Unstructured Text Data, pp. 73-80, 2010.

[14] Q. Chen, T Shipper, and L. Khan, Tweets mining using WIKIPEDIA and impurity cluster measurement, Proceedings of IEEE International Conference on Intelligence and Security Informatics, pp. 23-26, 2010.

[15] S. Ishikawa, Y. Arakawa, and S. Tagashira, Hot topic detection in local areas using Twitter and Wikipedia, Proceedings of International Conference on Architecture of Computing Systems, pp. 28-29, 2012.

[16] B. Liu, Web Data Mining: Exploring hyperlinks, contents, and usage data, Springer, 2006.

[17] J. Shim, H. C. Lee, The Development of Automatic Ontology Generation System Using Extended Search Keywords, Journal of the Korea Academia-Industrial cooperation Society(JKAIS), Vol. 11, no. 6, 2009.

저자 소개

장 재 영(정회원)

- 1992년 : 서울대학교 계산통계학과 (이학사)
- 1994년 : 서울대학교 계산통계학과 (이학석사)
- 1999년 : 서울대학교 계산통계학과 (이학박사)
- 2000년 ~ 현재 : 한성대학교 컴퓨터공학과 교수

<관심분야 : 데이터베이스, 정보검색, 데이터마이닝>

※ 본 연구는 한성대학교 교내학술연구비 지원과제임.