# TextRank 알고리즘을 이용한 문서 범주화

## (Text Categorization Using TextRank Algorithm)

배 원 식 † 차 정 원 ††
(Won-Sik Bae) (Jeong-Won Cha)

**요 약** 본 논문에서는 TextRank 알고리즘을 이용한 문서 범주화 방법에 대해 기술한다. TextRank 알고리즘은 그래프 기반의 순위화 알고리즘이다. 문서에서 나타나는 각각의 단어를 노드로, 단어들 사이의 동시출현성을 이용하여 간선을 만들면 문서로부터 그래프를 생성할 수 있다. TextRank 알고리즘을 이용하여 생성된 그래프로부터 중요도가 높은 단어를 선택하고, 그 단어와 인접한 단어를 묶어 하나의 자질로 사용하여 문서 분류를 수행하였다. 동시출현 자질(인접한 단어 쌍)은 단어 하나가 갖는 의미를 보다 명확하게 만들어주므로 문서 분류에 좋은 자질로 사용될 수 있을 것이라 가정하였다. 문서 분류기로는 지지 벡터 기계, 베이지언 분류기, 최대 엔트로피 모델, k-NN 분류기 등을 사용하였다. 20 Newsgroups 문서 집합을 사용한 실험에서 모든 분류기에서 제안된 방법을 사용했을 때, 문서 분류 성능이 향상된 결과를 확인할 수 있었다.

**키워드** : TextRank 알고리즘, 문서 범주화, 동시출현 자질, 지지 벡터 기계, 베이지언 분류기, 최대 엔트로피 모델, k-NN 분류기, 20 Newsgroups 문서 집합

***Abstract*** We describe a new method for text categorization using TextRank algorithm. Text categorization is a problem that over one pre-defined categories are assigned to a text document. TextRank algorithm is a graph-based ranking algorithm. If we consider that each word is a vertex, and co-occurrence of two adjacent words is a edge, we can get a graph from a document. After that, we find important words using TextRank algorithm from the graph and make feature which are pairs of words which are each important word and a word adjacent to the important word. We use classifiers: SVM, Naïve Bayesian classifier, Maximum Entropy Model, and k-NN classifier. We use non-cross-posted version of 20 Newsgroups data set. In consequence, we had an improved performance in whole classifiers, and the result tells that is a possibility of TextRank algorithm in text categorization.

**Key words** : TextRank algorithm, Text Categorization, Co-occurrence Word, SVM, Naïve Bayesian classifier, Maximum Entropy Model, 20 Newsgroups data set

## 1. 서 론

문서 범주화는 텍스트 문서를 미리 정의한 범주 중 하나 이상의 범주로 자동으로 분류하는 문제를 다루는 분야이다. 문서 범주화 시스템은 자질 선택 방법과 문서 분류기에 의해 성능이 좌우된다. 자질 선택은 학습 문서에서 생성되는 많은 자질 중에서 범주 판단에 유용한 자질만 선택하는 것으로, 통계적인 방법을 사용한다. 대표적으로 TF-IDF, 상호정보(Mutual Information), 카이 제곱 통계량($\chi^2$ Statistics), 정보 획득량(Information Gain), Topic Signaure 등의 방법이 사용된다[1,2]. 문서 분류기는 자질 선택 과정을 통해 선별된 자질로부터 문서 범주화 모델을 생성하여, 문서의 범주를 판단하는데 사용한다. 주로 기계학습 방법인 베이지언 분류기(Naive Bayes Classifier)[3,4], 지지 벡터 기계(Support Vector Machine)[5], k-NN 분류기(k-Nearest Neighbor Classifier)[6], 최대 엔트로피 모델[7] 등이 사용된다. 자질 선택 방법은 카이 제곱 통계량, 문서 분류기는 지지 벡터 기계가 좋은 성능을 발휘하는 것으로 알려져 있다.

기존 문서 범주화 연구에서 주로 사용되는 자질은 문서에서 출현하는 단어 하나이다. 단어는 특정 개념을 추상화한 것이므로, 단어 하나는 포괄적인 의미를 담고 있는 경우가 많다. 예를 들어 "나무"라는 단어가 하나만 있을 때, 우리는 수많은 종류의 나무를 떠올릴 수 있다. 이 때, 만약 "나무"라는 단어 앞에 "사과"라는 단어가 있다면, "사과 나무"가 되어 의미를 명확하게 할 수 있다. 이와 같이 복합어나 단어 사이에 수식 관계가 존재하는 경우, 서로 인접한 단어를 함께 자질로 사용하는 것이 문서 범주화에 도움이 될 수 있을 것이라는 가설로부터 동시출현 자질을 사용하였다. 그러나 인접한 단어 쌍 전체를 자질로 사용하지는 않고, 자질 선택 방법으로

· 이 논문은 2009 한국컴퓨터종합학술대회에서 'TextRank 알고리즘을 이용한 문서 범주화'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 국립창원대학교 컴퓨터공학과
wonsigi529@changwon.ac.kr
†† 종신회원 : 국립창원대학교 컴퓨터공학과 교수
jcha@changwon.ac.kr
(Corresponding author임)

논문접수 : 2009년 8월 14일
심사완료 : 2009년 10월 20일

TextRank 알고리즘[8]을 사용하였다. 문서로부터 TextRank 알고리즘을 통해 중요도가 높은 단어를 추출하고, 그 단어와 인접한 단어의 쌍을 하나의 자질로 사용하였다. TextRank 알고리즘 외에 일반적으로 문서 범주화 시스템에서 사용하는 통계기반 자질 선택 방법은 사용하지 않았다. 제안된 시스템의 성능 평가에는 20 Newsgroups 문서 집합[9]과 지지 벡터 기계, 베이지언 분류기, 최대 엔트로피 모델, k-NN 분류기를 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 소개한다. 그리고 3장에서는 제안 시스템에 대해 자세히 설명하고, 4장에서는 실험을 통해 결과를 분석한다. 마지막으로 5장에서는 결론과 향후 과제에 대해서 다룬다.

## 2. 관련 연구

본 장에서는 동시출현 자질과 TextRank 알고리즘을 사용했던 관련 연구에 대해 설명한다.

### 2.1 동시출현 단어 쌍 자질

배원식 등은 단어 하나 대신 문서에서 빈번하게 동시출현하는 단어 쌍을 자질로 하는 문서 범주화 방법을 제안하였다[10]. [10]에서는 특정 단어와 그 단어 앞뒤 일정 크기의 윈도우 내에 존재하는 단어의 쌍을 자질로 사용하였다. 단어는 명사, 동사, 형용사로 제한하였고, 제목에 나타난 단어에 가중치를 부여하였다. 자질 선택 방법으로는 카이 제곱 통계량과 Topic Signature를 사용하였다. Reuters-21578 문서 집합[11]을 이용한 실험에서 단일단어 자질보다 동시출현 자질을 사용하는 경우에 성능이 크게 향상되는 결과를 보였다. 또한 Topic Signature를 사용한 자질 선택이 카이 제곱 통계량을 사용한 것에 비해 높은 성능을 보여, Topic Signature라는 자질 선택 방법에 대한 가능성도 보여주었다.

### 2.2 TextRank 알고리즘

Mihalcea와 Tarau는 PageRank 알고리즘[12]을 텍스트에 적용한 TextRank 알고리즘을 제안하였다[8]. PageRank 알고리즘은 그래프 기반의 순위화 알고리즘이다. 수집된 인터넷 문서 각각을 그래프의 노드, 문서 내부의 링크 정보를 간선으로 가정하여 방향성이 있는 그래프를 만들어 문서의 중요도를 계산한다. 이와 달리 TextRank 알고리즘은 한 문서에서 출현한 단어나 문장 등을 노드로 간주하고 방향성이 없는 그래프를 생성한다. 이 때, 한 문서 내에서는 가시적인 연결 고리가 없으므로 단어나 문장 사이를 연결할 가상의 연결 고리가 필요하다. [8]에서는 단어의 인접성을 가상의 연결 고리로 만들어 키워드 추출에 사용하였다. 또한 문장의 유사도를 연결 고리로 하여 중요문장 추출에 사용하였다. 실험을 통해 TextRank 알고리즘을 자연어 처리 분야에 적용할 수 있다는 가능성을 보여주고 있다.

## 3. 제안 시스템

제안 시스템에서 문서로부터 TextRank 알고리즘을 이용하여 중요한 단어를 추출하고, 그 단어와 인접한 단어 쌍을 결합하여 동시출현 자질을 생성하여 문서 분류에 사용한다. 카이 제곱 통계량과 같은 통계적 기법에 기반한 자질 선택 방법은 사용하지 않는다. 추출된 자질을 Bow Toolkit[13]에서 제공하는 지지 벡터 기계, 베이지언 분류기, 최대 엔트로피 모델, k-NN 분류기를 사용하여 문서를 분류하고 성능을 평가하였다. 지지 벡터 기계의 커널으로는 선형 커널(Linear kernel)을 사용하였다.

### 3.1 문서로부터 그래프 생성

문서로부터 그래프를 생성하는 방법은 다음과 같다. 먼저 그림 1과 같은 문서를 품사 태깅을 수행한다. 품사 태깅을 수행하면 그림 2의 괄호 안에 표기된 단어와 같이 고유명사를 제외한 복수 명사는 단수 형태로, 동사는 원형으로 복원된다. 그래프의 노드가 될 수 있는 단어는 품사가 명사(N), 동사(V), 형용사(J), 부사(R)인 단어 만으로 한정하였고, 단어의 어휘 및 품사를 결합하여 단어로 취급하였다. 예를 들어, 단어 "machine"의 품사가 명사면, "machine:N"의 형태로 단어로 사용한다. 그 결과 그림 2와 같은 단어 리스트를 얻을 수 있다.

BGI Drivers for SVGA

I require BGI drivers for Super VGA Displays and Super XVGA Displays.

Does anyone know where I could obtain the relevant drivers.

그림 1 원문

**BGI:N** Drivers(**driver:N**) **SVGA:N**
**require:V** **BGI:N** drivers(**driver:N**) **Super:N** **VGA:N** **Displays:N** **Super:N** **XVGA:N** **Displays:N**
Does(**do:N**) **anyone:N** **know:V** **obtain:V** **relevant:J** drivers(**driver:N**)

그림 2 품사 태깅 후 남은 단어

단어 리스트로부터 인접한 단어 사이에 간선을 연결하면 그림 3과 같은 방향성 없는 그래프를 생성할 수 있다. 단, 문장을 구분하지 않으며, 같은 단어에 대한 재귀 연결(Recursive link)은 허용하지 않는다. 또한 동일한 연결이 여러 번 가능하더라도 한 번만 그래프 생성에 쓰인다. 실제 시스템에서 그래프는 행렬로 구현하였다.

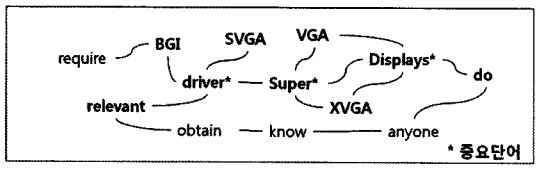


그림 3 문서로부터 생성한 그래프

### 3.2 TextRank 알고리즘을 이용한 자질 생성

3.1절의 과정에 의해 생성된 그래프에 식 (1)을 적용하여 단어의 중요도 WA($V_i$)값을 계산한다. TextRank 알고리즘은 단어의 중요도가 일정 값으로 수렴될 때까지 반복적으로 연산을 수행한다.

$$WS(V_i) = (1-d) + d \times \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(v_j)} w_{jk}} WS(V_j) \quad (1)$$

식 (1)에서 $V_i$는 그래프 상의 임의의 단어이고, $V_j$는 단어 i와 인접한 단어, $V_k$는 단어 j와 인접한 단어이다. $WS(V_i)$는 현재 단계에서 단어 i의 중요도이고, $WS(V_j)$는 이전 단계에서 단어 j의 중요도이다. 초기 단계에서 $WS(V_j)$의 값은 1이다. $In(V_i)$는 단어 i와 인접한 단어의 집합, $Out(V_j)$는 단어 j와 인접한 단어의 집합이다. 그래프의 방향성이 없으므로 서로 인접한 단어들은 들어오는 링크(Incoming link)와 나가는 링크(Outgoing link)를 하나씩 갖는다. 또한 그래프의 가중치가 없으므로, 가중치 $w_{ji}$, $w_{jk}$는 1의 값을 가지며, d는 제동 계수(Damping factor)로 0.85의 값을 갖는다.

단어의 중요도 계산이 끝나면, 중요도가 높은 단어들을 추출하고, 그 단어와 인접한 단어들을 결합하여 동시출현 자질을 생성한다. 표 1은 그림 3의 그래프에 TextRank 알고리즘을 수행한 결과 중에 일부를 정리한 표이다.

표 1 TextRank 알고리즘 수행 결과

| 순위 | 중요도 | 단어 | 인접한 단어 |
|---|---|---|---|
| 1 | 1.5262 | driver | bgi, relevant, super, svga |
| 2 | 1.4516 | displays | know, super, vga, xvga |
| 3 | 1.4309 | super | super, displays, driver, vga, xvga |
| 4 | 0.8660 | require | bgi, svga |
| 5 | 0.8555 | obtain | know, relevant |

## 4. 실험 및 결과

본 장에서는 본 논문의 제안한 시스템의 성능 평가를 위해 사용한 문서 집합과 성능 평가 방법에 대해 설명한다. 또한 실험을 통해 Baseline과의 성능을 비교한 결과를 분석하고, 나아가 기존 연구와의 성능을 비교한 결과를 정리한다.

### 4.1 실험 데이터

20 Newsgroups 문서 집합을 실험 데이터로 사용하였다. 20 Newsgroups 문서 집합은 총 20개의 범주, 19,997 문서로 구성되어 있다. 그 중에서 약 500개의 문서는 두 개 이상의 범주에 할당되어 있다. 본 논문에서는 이러한 교차 개제 문서(Cross-posted Document) 분류를 위한 방법을 고려하지 않았으므로, 교차 개제 문서를 제외하고 남은 18,846 문서만 실험에 사용하였다. 문서 집합을 임의로 4등분하여 그 중 하나(25%; 4,718 문서)를 실험 문서 집합으로 사용하였고, 나머지(75%; 14,128 문서)를 학습 문서 집합으로 사용하였으며, 4단 교차 검증 방법(4-fold cross-validation)으로 실험을 진행하였다. 각 문서에는 많은 헤더 정보가 삽입되어 있는데, 제목과 본문 내용을 제외한 나머지는 제거하였다.

### 4.2 성능 평가 방법

표 2는 성능 평가에 사용하는 분할표(Contingency Table)[14]이다. 다중 범주 분류 문제에서는 각각의 범주에 대하여 표 2와 같은 분할표를 만들어 성능을 평가한다. 표의 열 방향이 실제 정답(Gold Standard), 행 방향이 시스템이 제대로 분류를 했는지(Positive), 아닌지(Negative)를 의미한다.

표 2 성능 평가를 위한 분할표

| | True | False |
|---|---|---|
| Positive | a | b |
| Negative | c | d |

표 2와 식 (2)로부터 정확도(P; Precision)와 재현율(R; Recall)을 계산한다.

$$P = \frac{a}{a+b}, \quad R = \frac{a}{a+c} \quad (2)$$

그리고 정확도와 재현율을 하나의 값으로 표현하기 위하여 식 (3)의 F1-Measure나 손익분기점(BEP; Break Even Point)을 사용한다.

$$F_1 = \frac{2PR}{P+R}, \quad BEP = \frac{P+R}{2} \quad (3)$$

또한 각각의 범주가 아닌 전체 시스템의 성능 측정을 위하여 범주별 분할표로부터 하나로 통합된 분할표를 만든다. 통합된 분할표로부터 F1-Measure 나 손익분기점을 계산할 수 있는데, 이러한 방법으로 성능을 측정하는 것을 Micro-averaging 방법[15]이라고 한다.

### 4.3 실험 결과

표 3은 20 Newsgroup 문서 집합을 사용한 기존 시스템과 제안 시스템의 성능을 정리한 표이다. NB는 베이지언 분류기, ME는 최대 엔트로피 모델을 의미한다. Baseline 시스템은 제안 시스템과 동일한 실험 환경에

표 3 문서 범주화 시스템 성능표(Micro F1-Measure)

| 시스템 | NB | k-NN | ME | SVM |
|---|---|---|---|---|
| Baseline | 85.32 | 37.05 | 82.70 | 88.46 |
| 제안 시스템 | 88.26 | 67.46 | 85.17 | 90.03 |
| Gliozzo, '05[16] | - | - | - | 88.60 |
| Tan, '07[17]* | - | - | - | 87.88 |
| Bekkerman, '01[18]* | - | - | - | 89.50 (BEP) |
| Yoon, '03[19]* | - | - | - | 87.11 |
| Yoon, '07[20]* | - | - | - | 85.30 |

서 단일 단어 자질을 사용하여 문서를 분류하여 성능을 측정하였다. 그리고 *가 표기된 시스템은 교차 게재 문서를 포함하여 성능을 평가한 시스템이다. [19]는 손익 분기점으로 표기되어 있던 성능을 정확도와 재현율로부터 F1-Measure로 재측정한 성능이다.

기존 시스템들은 자질 선택 방법보다는 문서 분류 방법을 개선하거나 문서 집합의 계층 구조를 활용하여 문서 범주화의 성능을 높이고자 하였다[16-20]. 그러나 제안 시스템은 자질 선택 방법의 개선을 통해 성능 향상을 꾀하였다. 표 3의 결과를 통해 자질 선택 방법의 개선을 통해 문서 범주화의 성능을 향상시킬 수 있음을 확인할 수 있다. 단일 단어 자질을 사용한 Baseline 시스템보다 모든 분류기에서 1.5% 이상의 성능 향상이 이루어졌다. 그리고 본 논문과 같이 중복 범주 문서를 제외한 기존 시스템[16]과 비교했을 때보다 높은 성능을 보이고 있다. 또한 중복 범주 문서를 포함한 기존 시스템들에 비해서도 2% 이상 높은 성능을 보여주고 있다. 물론 중복 범주 문서의 포함 여부 때문에 직접적으로 정확하게 성능 비교를 할 수는 없으나, 충분히 제안 시스템이 가능성이 있다는 사실을 말해주는 결과이다.

## 5. 결론 및 향후 과제

본 논문에서는 TextRank 알고리즘을 통해 단어의 중요도를 계산하고, 중요한 단어와 인접한 단어의 쌍을 자질로 사용하는 동시출현 자질을 제안하였다. 실험을 통해 단일단어 자질에 비해 동시출현 자질이 문서 범주화의 성능 향상에 기여한다는 사실을 확인할 수 있었다. 또한 기존의 시스템들과 비교 결과를 통해서 문서 분류 방법의 개선도 좋지만, 제안 시스템처럼 자질 선택 방법의 개선을 통해서도 성능 향상을 할 수 있다는 사실도 확인할 수 있었다. 이 결과는 앞으로의 연구에도 기여할 수 있을 것이다.

동시출현 자질은 분류 성능을 높이는데 필요한 자질을 많이 생성해주는 장점이 있지만, 불필요한 자질까지 많이 생성될 수 있다는 단점을 가지고 있다. 자질의 수가 증가하면 시스템의 계산 비용 또한 증가하는 문제가 발생할 수 밖에 없다. 따라서 불필요한 자질을 제거하는 방법에 대한 연구가 필요할 것이다. 또한 영어 품사 태거 대신에 한국어 품사 태거를 적용하면 어렵지 않게 제안 시스템의 방법론을 한국어 문서 분류에 사용할 수 있다. 따라서 향후, 제안 시스템을 한국어 문서 분류에 적용해보고, 가능성을 알아보고자 한다.

## 참 고 문 헌


[1] Y. Yang and J. O. Pederson, "A comparative study on feature selection in text categorization," *Proc. of the 14th International Conference on Machine Learning*, pp.412-420, 1997.

[2] C. Y. Lin and E. Hovy, "The Automated Acquisition of Topic Signatures for Text Summarization," *Proc. of the 18th International Conference on Computation Linguistics*, pp.495-500, 2000.

[3] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," *Proc. of the 10th European Conference on Machine Learning*, pp.4-15, 1998.

[4] A. K. McCallum and K. Nigram, "A Comparison of Event Models for Naive Bayes Text Classification," *Proc. of the AAAI-98 Workshop on Learning for Text Categorization*, pp.41-48, 1998.

[5] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Proc. of the 10th European Conference on Machine Learning*, pp.137-142, 1998.

[6] Y. Yang, "Expert netword: Effective and efficient learning from human decisions in text categorization and retrieval," *Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.13-22, 1994.

[7] K. Nigam, J. Lafferty, and A. K. McCallum, "Using Maximum Entropy for Text Categorization," *Proc. of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, pp.61-67, 1999.

[8] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," *Proc. of the Conference on Empirical Methods in Natural Language Processing 2004*, pp.404-411, 2004.

[9] K. Lang, "The 20 Newsgroups data set," http://people.csail.mit.edu/~jrennie/20Newsgroups

[10] W. Bae, Y. Han, and J. Cha, "Text Categorization using Topic Signature and Co-occurrence Features," *Proc. of the KIISE Korea Computer Congress 2008*, vol.35, no.1, pp.262-267, 2008. (in Korean)

[11] D. D. Lewis, "The Reuters-21578 data set," http://www.daviddlewis.com/resources/testcollections/reuters21578

[12] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems*, vol.30, pp.107-117,

1998.

[13] A. K. McCallum, "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering," http://www.cs.cmu.edu/~mccallum/bow/, 1996.

[14] K. Pearson, "On the theory of contingency and its relation to association and normal correlation," In *Karl Pearson's early statistical papers*, Cambridge: Cambridge University Press, pp.443-475, 1904/1948.

[15] Y. Yang, "An evaluation of statistical approach to text categorization," *Information Retrieval*, vol.1, no.1-2, pp.69-90, 1996.

[16] A. Gliozzo and C. Strapparava, "Domain Kernels for Text Categorization," *Proc. of the 9th Conference on Computational Natural Language Learning*, pp.56-63, 2005.

[17] S. Tan, "Using Error-Correcting Output Codes with Model-Refinement to Boost Centroid Text Classifier," *Proc. of the ACL 2007 Demo and Poster Sessions*, pp.81-84, 2007.

[18] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter, "On feature distributional clustering for text categorization," *Proc. of 24th Annual International ACM SIGIR Conference*, pp.146-153, 2001.

[19] Y. Yoon, C. Lee, and G. G. Lee, "Hierarchical text categorization using support vector machine," *Proc. of the 15th Human and Cognitive Language Technology*, pp.1-8, 2003. (in Korean)

[20] Y. Yoon and G. G. Lee, "Efficient implementation of associative classifiers for document classification," *Information Processing and Management*, vol.43, pp.393-405, 2007.