# 메타 속성을 융합한 기계 학습 기반 화재 뉴스 필터링 기법

김태준*, 김한준*
*서울시립대학교 전자전기컴퓨터공학부
e-mail: i2r.jun@gmail.com

# Machine Learning Based Fire News Filtering Technique Incorporating Meta-features

Tae-Jun Kim*, Han-joon Kim*
*School of Electrical and Computer Engineering, University of Seoul

**요 약**

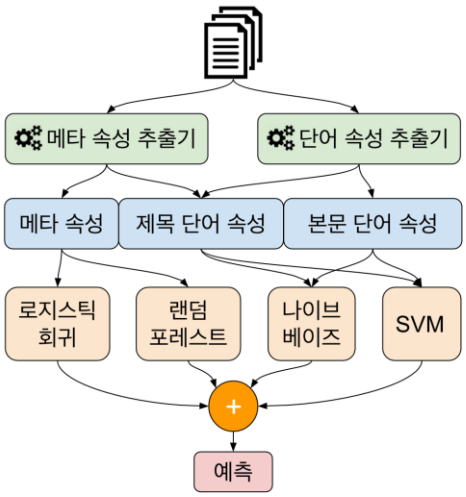
주제 기반 크롤링(Topical Crawling)으로 수집된 문서들은 서로 비슷한 단어들을 가지고 있기 때문에 정작 주어진 주제에 적합하지 않은 문서들을 포함할 수 있다. 이를 해결하기 위해 특정 주제에 해당하는 문서만을 필터링하는 작업이 필요하다. 본 논문은 화재 뉴스 기사에 대한 필터링을 위해 단어 기반 속성과 아울러 화재 뉴스 기사의 특성을 고려한 메타 데이터 속성을 추출하여 이에 특화된 기계학습 메커니즘을 제안하였다. 제안 기법의 F1-측정치는 92.1%로서, 현재 최고의 성능을 보이는 SVM, 나이브베이즈 알고리즘보다 2~3% 개선된 것이다.

## 1. 서론

인터넷은 방대한 양의 정보를 가지고 있어서 관심 있는 특정 주제의 문서만을 수집하는 주제 기반 크롤링이 자주 연구되어왔다[1]. 하지만, [2]와 같이 주제와 관련 있는 문서만을 수집하는 알고리즘을 적용하더라도 크롤링을 통해 얻은 문서 중에서 원하는 주제에서 벗어난 문서는 존재하기 마련이다. 따라서, 수집된 문서들을 데이터 분석에 사용하기 위해서는 주제와 관련 있는 문서만을 걸러내는 필터링 작업이 필수적이다. 그러나, 주제 기반 크롤러(topical crawler)가 수집한 문서들은 비슷한 단어들을 갖기 때문에 기존의 벡터 공간 모델(vector space model) [3]을 이용한 필터링의 성능에 한계가 있다. 본 논문은 화재 관련 뉴스 기사를 수집하는 경우, 문서의 메타 데이터를 활용하여 새로운 속성(feature)을 추출하고, 이를 기존의 단어 속성과 융합하고 이에 특화된 기계학습 메커니즘을 제안한다. 그림 1 은 제안하는 기계학습 메커니즘을 보여준다. 속성 유형에 적합한 기계학습 알고리즘을 채택하여 분류모델을 생성하고, 앙상블(ensemble) 기법으로 그들을 결합하여 필터링 성능을 개선하였다.

## 2. 속성 추출을 위한 메타 데이터 탐색

기존의 단어 속성 기반 필터링은 주제 기반 크롤링으로 수집한 문서들에 적용할 경우 일반적인 문서들에 비해 성능이 좋지 않다. 본 논문은 기존의 단어 속성만을 사용하는 기법을 개선하기 위해 문서의 메타 데이터 속성을 활용한다. 본 절에서는 메타 데이터에 대한 전반적인 탐색을 통해 메타 데이터가 왜 중요하고 문서 필터링에 도움이 되는지를 알아본다.

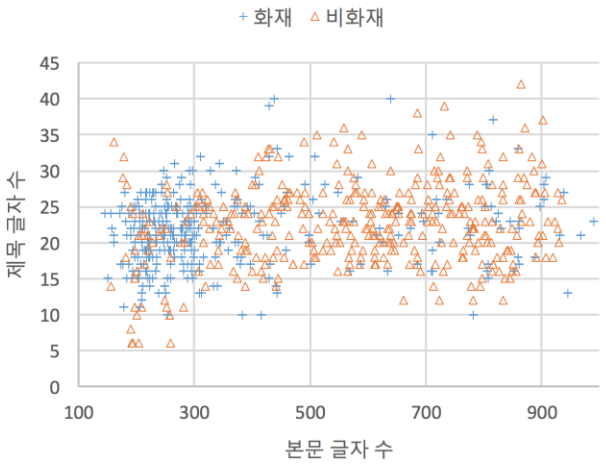



(그림 1) 기계학습 파이프라인

### 2.1 메타 데이터 탐색: 글자 수

먼저, 가장 기초적인 메타 데이터인 뉴스 기사들의 제목과 본문의 글자 수를 보자. 그림 2 는 이를 화재 기사와 비화재 기사를 분리하여 산점도로 나타낸 것이다. 그림 2 에서 비화재 기사의 표본들은 고르게 퍼져 있는 반면, 화재 기사의 경우 밀집해 있는 것을 알 수 있다. 특히, 화재 기사의 표본들은 본문의 글자

수가 적은 곳에 밀집해 있다. 이런 특성으로 인해 본문의 글자 수는 화재 뉴스 기사를 필터링하기 좋은 속성이라 할 수 있다.




(그림 2) 제목/본문 글자 수 산점도

### 2.2 메타 데이터 탐색: 단어 특성

<표 1> 단어의 특성과 위치에 따른 평균 출현 빈도

| 특성 | 위치 | 화재 | 비화재 | 비화재/화재 |
|---|---|---|---|---|
| 지명 | 제목 | 0.71 | 0.53 | 0.75 |
| | 본문 | 4.38 | 7.23 | 1.65 |
| 인명 | 제목 | 0.04 | 0.13 | 2.94 |
| | 본문 | 1.91 | 3.92 | 2.05 |

표 1 은 뉴스 기사에 나온 단어들의 특성과 출현 위치에 따른 평균 출현 빈도 수이다. 예를 들어, 비화재 기사 제목에 나타나는 평균 인명 단어 수는 0.13 개이고, 화재의 경우에는 0.04 개이다. 즉, 비화재 기사의 제목에서 인명 단어가 나올 확률은 화재에 비해 약 2.94 배 높다. 이와 같이 단어의 특성과 위치에 따른 빈도 수 또한 필터링에 좋은 속성이라 할 수 있다.

### 2.3 메타 데이터 탐색: 단어 품사

<표 2> 단어 품사 비율

| *위치* | *품사* | *화재* | *비화재* | *비화재/화재* |
|---|---|---|---|---|
| 제목 | 외국어 | 0.0014 | 0.0103 | 7.15 |
| 본문 | 한자 | 0.0001 | 0.0006 | 5.41 |
| 제목 | 긍정지정사 | 0.0002 | 0.0007 | 3.90 |
| 제목 | 어근 | 0.0011 | 0.0035 | 3.25 |
| 본문 | 어근 | 0.0009 | 0.0024 | 2.83 |

표 2 는 뉴스 기사의 단어들의 품사 비율 중 비화재 기사에서 더 많이 나오는 품사 상위 5 개를 나타낸 것이다. 제목의 외국어 비율은 비화재 기사가 화재 기사에 비해 약 7 배 정도 높다. 이처럼 화재 기사와 비화재 기사에서 많이 나오는 품사는 서로 다르기 때문에 품사 비율은 필터링 속성으로 쓰일 수 있다.

### 2.4 메타 데이터에 따른 단어 속성 탐색

기존의 단어 속성은 비슷한 단어를 갖는 문서들의 집합에서는 상대적으로 성능이 떨어지지만 여전히 유용한 속성이다. 따라서, 메타 데이터를 이용하여 벡터 공간 모델을 강화할 방법을 모색해본다.

<표 4> 단어 "공장"의 출현 빈도 순위의 변화

| 후보 범위 | 순위(±전체와의 차) | |
|---|---|---|
| | 화재 | 비화재 |
| 전체 | 14 | 182 |
| 제목 | 3(+11) | 562(-380) |
| 본문 | 19(-5) | 174(+8) |

다음 예를 통해 단어가 분류에 미치는 영향력이 메타 데이터에 따라 어떻게 변하는지 살펴보자. 표 4 는 뉴스 기사 내의 단어 "공장"의 출현 빈도 순위를 나타낸 것이다. 제목과 본문의 모든 단어들 중 "공장"은 화재 기사에서는 14 번째, 비화재 기사에서는 182 번째로 많이 나왔다. 하지만, 이를 제목 내에서, 그리고 본문 내에서의 순위로 나누어 보면 달라진다. 제목 내에서의 순위는 화재 기사의 경우 11 단계가 증가하여 3 위가 된 반면, 비화재 기사에서는 380 단계가 감소하여 562 위가 되었다. 본문 내에서는 각각 19 위와 174 위로 변하여 차이가 좁아졌다. 따라서, 어떤 뉴스 기사의 제목에서 "공장"이 나타났다면, 그 기사가 화재 기사일 확률은 매우 높다. 하지만, 본문에서 나타났다면 상대적으로 그리 높지 않다.

위의 예와 같이 단어가 출현한 위치(제목/본문)를 고려하여 속성을 설계한다면, 더욱더 정밀한 기계학습이 가능할 것이다.

## 3. 메타 데이터를 이용한 속성 설계

이전 장에서 다양한 메타 데이터의 형태와 가치를 살펴보았다. 이번 장에서는 이들을 기존의 단어 속성과 융합한다.

### 3.1 메타 속성 $V_{meta}$

<표 5> 속성 $V_{meta}$의 구성 요소

| 요소 | 벡터 길이 |
|---|---|
| 제목과 본문의 글자 수 | 2 |
| 제목과 본문의 지명 수 | 2 |
| 제목과 본문의 인명 수 | 2 |
| 제목의 단어들의 품사 별 구성 비율 | 16 |
| 본문의 단어들의 품사 별 구성 비율 | 16 |
| 총계 | 38 |

이전 장에서 살펴본 뉴스 기사들의 메타 데이터를

이용해 표 5 와 같은 구성 요소를 갖는 새로운 '메타 속성' $V_{meta}$를 정의한다.

### 3.2 메타 단어 속성 $V_{word}$

제목에서 출현한 단어들로 이루어진 단어 빈도 벡터를 $V_{title}$, 본문에서 출현한 단어들로 이루어진 것을 $V_{body}$라 하자. 이 두 벡터를 이어붙여 새로운 '메타 단어 속성' $V_{word}$를 정의한다. 이는 다음과 같이 구할 수 있다:

$$V_{word} = V_{title} \| V_{body}$$

반면, 기존의 단어 빈도 벡터 $V_{old}$는 다음과 같이 구해진다.

$$V_{old} = V_{title} + V_{body}$$

$V_{word}$는 $V_{old}$와 달리 같은 단어이더라도 제목 혹은 본문에서 나왔는지에 따라 차원을 다르게 할당함으로써 한 단어가 두 가지 의미를 갖게 된다.

또한, 새로 정의된 벡터 $V_{meta}$와 $V_{word}$ 모두를 고려한 기계 학습을 위해 두 벡터를 연결한 속성 $V_{all}$을 다음과 같이 정의한다:

$$V_{all} = V_{meta} \| V_{word}$$

## 4. 실험 방법

### 4.1 실험 데이터

뉴스 기사는 주제 기반 크롤러가 검색 엔진 '다음'에 "화재 -삼성화재 -동부화재 -메리츠화재"를 검색한 결과를 내려받아 수집되었다. 그 중 1326 건을 무작위 추출한 후, 전문가가 직접 화재/비화재 뉴스 기사로 분류한 것을 실험 데이터로 사용하였다. 학습에는 전체 데이터의 68.9%를, 검증에는 31.1%의 데이터를 사용했다. 자세한 실험 데이터 분포는 아래의 표 1 과 같다.

<표 6> 실험 데이터 분포

| | 화재 | 비화재 | 총합 | 비율(%) |
|---|---|---|---|---|
| 학습 | 380 | 534 | 914 | 68.9 |
| 검증 | 158 | 254 | 412 | 31.1 |
| 총합 | 538 | 788 | 1326 | |
| 비율(%) | 40.6 | 59.4 | | |

### 4.2 성능 평가

학습 모델들을 검증하기 위해 정확도(accuracy), 정밀도(precision), 재현율(recall), F1-measure(혹은 F1-score)를 사용한다. 모델 $C_i$의 예측 결과 분할표가 표 2 와 같이 주어졌을 때 $C_i$의 정확도($A_i$), 정밀도($P_i$), 재현율($R_i$), F1-measure($F_i$)는 다음과 같이 구할 수 있다:

$$A_i = \frac{tp + tn}{tp + tn + fp + fn}$$

$$P_i = \frac{tp}{tp + fp} \qquad R_i = \frac{tp}{tp + tn} \qquad F_i = 2 \cdot \frac{P_i \cdot R_i}{P_i + R_i}$$

<표 7> 모델 $C_i$의 예측 분할표

| *모델 $C_i$* | | *모델 예측* | |
|---|---|---|---|
| | | 화재 | 비화재 |
| 전문가 판단 | 화재 | tp(참 양성) | fn(거짓 음성) |
| | 비화재 | fp(거짓 양성) | tn(참 음성) |

본 논문의 목적은 뉴스 기사들 중 화재 뉴스 기사만을 걸러내는 것이다. 따라서, 표 2 의 두 가지 오류 중, 거짓 양성 오류가 거짓 음성 오류에 비해 상대적으로 심각하다. 그러므로, 모델은 정밀도를 극대화하여 최대한 거짓 양성을 줄여야 한다. 또한, 그와 동시에 재현율 또한 높여 가능한 많은 양의 화재 기사를 필터링 결과에 포함할 수 있어야 한다. 결과적으로, 본 실험에서는 이 두 가지 지표를 모두 아우를 수 있는 F1-measure 값을 극대화하는 것이 중요하다. 이를 위해 추가적인 평가 척도로 최대 F1-measure($F_{iMAX}$) 또한 사용한다. 이는 F1-measure 값을 극대화하기 위해 모델 $C_i$의 임계값(threshold) $T_i$를 조정한 후의 $F_i$이다. 이는 다음과 같이 구할 수 있다:

$$T_{iMAX} = \arg\max_t (F_i|_{T_i=t})$$

$$F_{iMAX} = F_i|_{T_i=T_{iMAX}}$$

### 4.3 모델 설계

각 속성을 다양한 학습 알고리즘에 적용했다. $V_{meta}$는 연속형 입력 변수에 성능이 좋기로 알려진 로지스틱 회귀(Logistic Regression)과 랜덤 포레스트(Random Forest)에 적용했으며, $V_{word}$는 텍스트 분류에 성능이 좋기로 알려진 나이브 베이즈(Naïve Bayes)와 서포트 벡터 머신(Support Vector Machine)에 적용했다. 또한, $V_{all}$을 사용한 모델과 대조군으로 $V_{old}$를 사용한 모델 또한 실험했다. 표 8 은 학습 알고리즘과 학습 속성에 따른 모델의 이름을 나타낸다.

<표 8> 모델 설계 상세

| 이름 | 학습 알고리즘 | 학습 속성 |
|---|---|---|
| $LR$ | 로지스틱 회귀 | $V_{meta}$ |
| $RF$ | 랜덤 포레스트 | $V_{meta}$ |
| $NB_{old}$ | 나이브 베이즈 | $V_{old}$ |
| $NB_{meta}$ | 나이브 베이즈 | $V_{word}$ |
| $NB_{all}$ | 나이브 베이즈 | $V_{all}$ |
| $SVM_{old}$ | 서포트 벡터 머신 | $V_{old}$ |
| $SVM_{meta}$ | 서포트 벡터 머신 | $V_{word}$ |
| $SVM_{all}$ | 서포트 벡터 머신 | $V_{all}$ |

### 4.4 모델 결합

$V_{meta}$와 $V_{word}$ 모두의 장점을 취하는 방법에는 $V_{all}$과 같이 속성을 연결하는 방법 외에 서로 다른 속성을 학습한 모델들을 결합하는 방법이 있다. 따라서, 본 논문은 이전에 설계한 모델들을 결합한 앙상블(ensemble) 모델 또한 실험했다.

사용된 앙상블 기법은 두 가지 방식의 다수결 투표

(Majority Voting)로 "클래스 투표"와 "확률 투표"이다. "클래스 투표"는 모델의 예측 값들 중 가장 많이 나온 값을 최종 예측 값($\hat{y}$)으로 출력하는 방식으로 다음과 같이 구한다:

$$d_{ij} = \begin{cases} 1, & \text{if } C_i(x) = j \\ 0, & \text{otherwise} \end{cases}$$

$$\hat{y} = \arg\max_j \sum_i d_{ij}$$

$d_{ij}$는 모델 $C_i$의 예측 값이 $j$와 같으면 1, 아니면 0 인 값이다. 반면, "확률 투표"는 모델들의 각 클래스에 대한 예측 확률에 대한 합계를 내어 확률이 더 높은 클래스를 택하는 것이다. 이 방식의 최종 예측 값($\hat{y}$)은 아래와 같이 계산된다:

$$\hat{y} = \arg\max_j \sum_i r_{ij}$$

여기서 $r_{ij}$는 클래스 $j$에 대한 모델 $C_i$의 예측 확률이다. "확률 투표"의 경우 앙상블 모델을 구성하는 모델들의 확률 측정(calibration) 능력이 좋은 경우 높은 성능을 보인다. 표 9 는 앙상블 모델의 구성원과 결합 방식에 따른 그들의 이름을 나타낸다.

<표 9> 앙상블 모델 설계 상세

| 이름 | 모델 구성원 | 결합 방식 |
|---|---|---|
| $M1$ | $LR$, $RF$, $SVM_{meta}$ | 클래스 투표 |
| $M2$ | $RF$, $NB_{meta}$, $SVM_{meta}$ | |
| $P1$ | $LR$, $RF$, $SVM_{meta}$ | 확률 투표 |
| $P2$ | $RF$, $NB_{meta}$, $SVM_{meta}$ | |

## 5. 실험 결과 및 분석

각 모델의 평가 척도에 따른 성능을 표 10 에 정리했다. 각 평가 척도 별로 가장 높은 값은 바탕색과 밑줄로 표시하고 2, 3 위는 바탕색으로만 표시했다. 또한, 기존의 방식을 사용한 대조군인 $NB_{old}$, $SVM_{old}$ 는 가장 위의 두 행으로 분리하여 표기하였다.

먼저 $V_{meta}$ 만을 사용한 $RF$ 를 살펴보자. 단어 빈도수를 전혀 고려하지 않았음에도 불구하고 정확도가 0.871, F1-measure 는 0.823 을 기록하고 있다. 게다가 정밀도는 $NB_{old}$ 보다 높은 수치인 0.872 이다. 하지만, 나이브 베이즈의 경우 새롭게 정의한 속성의 효과가 없었다. 그러나, 서포트 벡터 머신은 재현율을 제외한 모든 부문에서 본 논문이 제시한 속성을 학습한 경우가 더욱더 높은 수치를 보였다. 앙상블 모델 중에서는 확률 측정 능력이 좋은 것으로 알려진 모델만을 구성원으로 가진 $P1$이 3 가지 부문에서 높은 성능을 보였다. 또한, 최대 F1-measure 은 $P2$의 성능이 가장 좋았다.

$NB_{all}$, $SVM_{all}$ 그리고 다른 앙상블 모델들을 비교해 보면, 모두 $P1$ 혹은 $P2$의 성능이 더 높았다. 이를 통해 $V_{meta}$와 $V_{word}$ 모두의 장점을 취하기 위한 방법은 벡터를 이어 붙인 $V_{all}$을 학습하는 것보단 앙상블 기법을 통한 모델 결합이 더욱더 나은 방법임을 알 수 있었다.

<표 10> 각 모델의 성능

| 모델 ($C_i$) | 정확도 ($A_i$) | 정밀도 ($P_i$) | 재현율 ($R_i$) | F1 ($F_i$) | 최대 F1 ($F_{iMAX}$) |
|---|---|---|---|---|---|
| $NB_{old}$ | 0.922 | 0.858 | **0.956** | 0.904 | 0.904 |
| $SVM_{old}$ | 0.903 | 0.824 | 0.949 | 0.882 | 0.891 |
| $LR$ | 0.827 | 0.760 | 0.804 | 0.782 | 0.782 |
| $RF$ | 0.871 | 0.872 | 0.778 | 0.823 | 0.843 |
| $NB_{meta}$ | 0.922 | 0.856 | **0.955** | 0.903 | 0.903 |
| $NB_{all}$ | 0.917 | 0.851 | 0.949 | 0.897 | 0.897 |
| $SVM_{meta}$ | **0.932** | **0.886** | 0.942 | **0.913** | 0.916 |
| $SVM_{all}$ | 0.919 | **0.913** | 0.872 | 0.892 | **0.919** |
| $M1$ | 0.902 | 0.882 | 0.859 | 0.870 | 0.870 |
| $P1$ | **0.934** | **0.939** | 0.885 | **0.911** | 0.915 |
| $M2$ | **0.932** | **0.886** | 0.942 | **0.913** | 0.913 |
| $P2$ | 0.924 | 0.861 | **0.955** | 0.906 | **0.921** |

## 6. 결론

본 논문은 주제 기반 크롤링으로 얻은 뉴스 기사들 중 화재 뉴스 기사만을 필터링하기 위해 메타 속성과 메타 단어 속성을 정의했다. 그 결과, 기존의 단어 속성보다 높은 정확도, 정밀도, F1-measure 를 얻을 수 있었다. 향후 실험 계획은 이를 다른 분야의 뉴스 기사에도 적용해보는 것이다.

## 7. 감사의 글

본 연구는 국토교통부 도시건축연구사업의 연구비지원(**16AUDP-B100356-02**)에 의해 수행되었습니다.

## 참고문헌

[1] Pant, Gautam, and Filippo Menczer. "Topical crawling for business intelligence." International Conference on Theory and Practice of Digital Libraries. Springer Berlin Heidelberg, 2003.

[2] Wang, Can, et al. "On-line topical importance estimation: an effective focused crawling algorithm combining link and content analysis." Journal of Zhejiang University SCIENCE A 10.8 (2009): 1114-1124.

[3] Suzuki, Makoto, et al. "On a new model for automatic text categorization based on Vector Space Model." Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on. IEEE, 2010.