

# 머신러닝 기반 공지문 분류 및 추천 애플리케이션 개발

김혜민, 오지은, 정혜린, 이기용  
숙명여자대학교 소프트웨어학부  
e-mail: {1117hyemin, juneoh227, lync2846}@gmail.com  
kiyonglee@sookmyung.ac.kr

## Development of a Notice Classification and Recommendation Application Using Machine Learning Techniques

Hyemin Kim, Jiun Oh, Hyerin Chung, Ki Yong Lee  
Dept. of Computer Science, Sookmyung Women's University

### 요 약

본 논문에서는 웹 및 문자 공지문을 자동으로 분류하고 추천함으로써 사용자가 원하는 공지문만을 볼 수 있도록 하는 애플리케이션을 개발한다. 본 애플리케이션은 공지문을 여러 카테고리로 자동 분류하여 사용자가 원하는 카테고리에 속한 공지문만을 볼 수 있도록 하며, 사용자가 선호할 만한 공지문을 추천하는 기능을 제공한다. 공지문 분류를 위해 다층 신경망 모델과 Naive Bayes 분류기를 사용하였으며, 공지문 추천을 위해 키워드 기반 자체 알고리즘을 사용하였다. 그 밖에 Word2Vec 을 활용한 검색어 추천 등 부가 기능을 제공하여 사용자가 쉽게 공지문을 찾을 수 있도록 하였다. 본 애플리케이션을 통해 사용자는 수많은 공지문 중 관심 있는 공지문만을 효율적으로 확인할 수 있다.

### 1. 서론

많은 대학생이 필요한 학사, 취업 정보를 얻기 위해 웹서핑을 하는 등 정보를 찾는 데에 상당한 시간을 쓰고 있다. 그러한 학생을 지원하기 위하여 대학들도 알맞은 정보를 제공하고자 다양한 시도를 하고 있다. 예를 들어 숙명여자대학교에서는 학교 메인 홈페이지, SnoWe(학교 공지사항 홈페이지), 개별 학과 홈페이지, 문자 메시지 등 다양한 경로로 학생에게 유익한 정보를 전달한다. 그러나 학생들은 각각의 사이트에 일일이 접속해 공지문을 확인해야 하며, 공지사항이 카테고리별로 분류되어 있지 않거나, 학생 본인과 상관이 없는 문자 메시지를 지속적으로 수신하는 등의 불편을 겪고 있다.

그러므로 본 논문에서는 학교 전반의 웹, 문자 공지문을 모아 사용자가 자신에게 필요한 정보를 손쉽게 열람하고 보다 관심 있는 분야의 공지문을 찾을 수 있도록 하는 애플리케이션을 개발한다. 해당 애플리케이션은 제대로 분류되지 않고 한 게시판에 모여있는 웹 공지문을 알맞은 카테고리로 재분류하며, 사용자의 기호에 맞는 공지문을 추천하고, 사용자의 전공과 관련된 문자 메시지를 따로 모아 제공한다. 분류를 위해 다층 신경망(multilayer neural network)[1] 모델과 Naive Bayes 분류기[2]를 사용했고, 추천은 키워드 기반 자체 개발 알고리즘을 사용했다. 그밖에 검색어 추천과 유사 게시물 추천을 위해 각각 Word2Vec[3], N-gram 기법[4]을 이용했다.

어플리케이션은 'SSOK' 으로 부르며, 크게 다음과 같은 4 가지 기능을 제공한다.

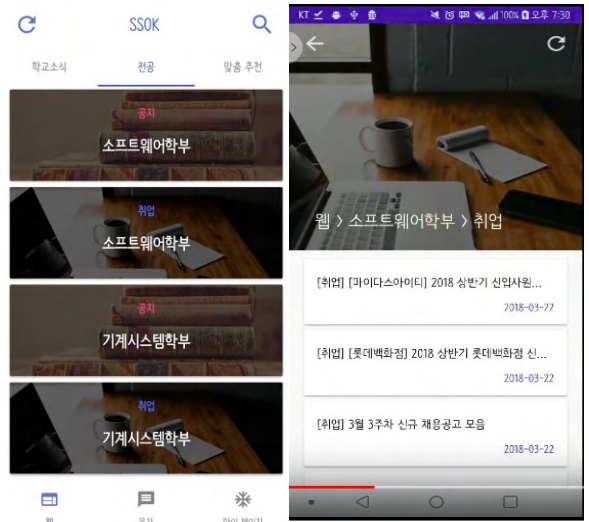
- 미분류 상태인 웹 공지문과 문자 공지를 알맞은 카테고리로 분류하는 기능
- 사용자가 입력한 개인 정보에 따라 맞춤형 웹 공지를 추천하는 기능
- 웹과 문자를 통합하여 검색하고, 사용자가 입력한 검색어와 유사한 검색어를 추천하는 기능
- 사용자가 유익하다고 판단한 공지를 따로 스크랩할 수 있는 보관함 기능

논문의 구성은 다음과 같다. 2 장에서는 애플리케이션의 세부 기능을 보인다. 3 장에서는 자세한 수행 사례를 보인다. 4 장에서는 기능 구현에 사용된 기술을 보이고 5 장에서 결론을 내린다.

### 2. 세부 기능

'SSOK' 의 핵심 기능은 분류되지 않은 웹 공지문들 내용에 맞는 카테고리로 분류하고, 사용자의 정보와 기호에 맞는 공지문과 검색어를 추천하는 것이다. 따라서 사용자는 일일이 여러 사이트를 방문해 정돈되지 않은 데이터를 찾아보는 대신, 찾고자 하는 정보를 보다 쉽게 열람하고 수집할 수 있다.

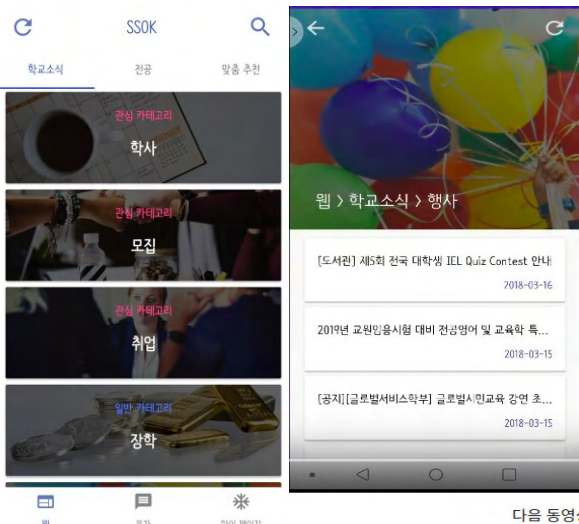
1. 학교/학과 공지문 분류 기능: 분류되지 않은 학교/학과 게시판의 글을 수집하고 내용에 따라 카테고리를 분류한다.
2. 추천 공지문 기능: 사용자의 기본정보와 공지문의 게시를 고려하여 사용자가 관심있어 할 만한 공지문을 10 개 추천한다.
3. 유사 공지문 추천 기능: 사용자가 현재 읽고 있는 공지문과 유사한 공지문 2 건을 추천해준다.
4. 검색 및 유사 검색어 기능: 서버에 저장된 웹 공지문과 사용자 휴대폰에 저장된 문자 공지문 중 검색어가 들어 있는 모든 공지문을 검색한다. 또한 사용자는 해당 검색어와 유사한 단어 4 건을 추천받을 수 있다.
5. 스크랩 기능: 사용자가 보관한 공지문을 모아볼 수 있도록 한다.



(그림 2) 웹 - 전공 탭

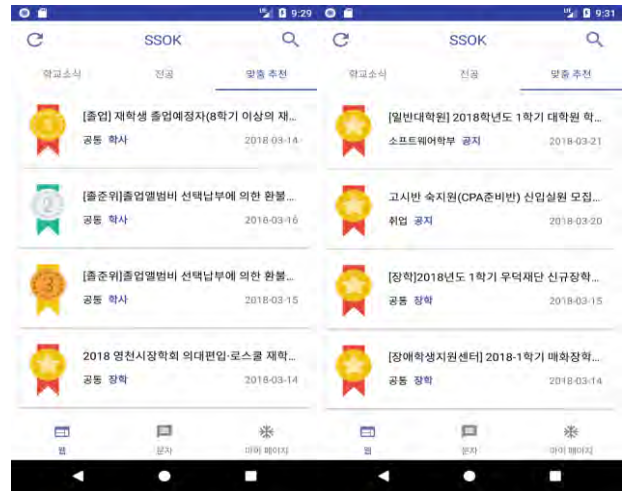
### 3. 수행 화면

그림 2 는 사용자가 속한 전공의 웹 공지문을 카테고리에 따라 보여주는 화면이다. 전공의 웹 공지문은 공지와 취업 카테고리로만 나뉜다. 그림 1 과 마찬가지로 카테고리를 눌러 해당 카테고리에 해당하는 웹 게시글을 열람할 수 있다. 학과 웹 공지문을 카테고리로 나누기 위해 Naïve Bayes 분류기를 사용하였다.



(그림 1) 웹 - 학교소식 탭

그림 1 은 학교 웹 공지문을 카테고리에 따라 보여주는 화면이다. 화면의 상단에는 사용자가 관심으로 지정한 카테고리가, 하단에는 사용자가 일반적으로 지정한 카테고리가 출력된다. 사용자가 무관심으로 지정한 카테고리는 보이지 않도록 한다. 사용자는 원하는 카테고리를 눌러 해당 카테고리에 속하는 웹 공지문들을 열람할 수 있다. 학교 웹 공지문을 카테고리로 분류하기 위해서 다층 신경망 모델을 사용하였다.



(그림 3) 웹 - 맞춤추천 탭

그림 3 은 소프트웨어학부 4 학년인 학생이 학사, 장학, 취업을 관심 카테고리로 지정한 경우, 해당 학생에 대한 맞춤 추천 공지문을 보여주는 화면이다. 추천 리스트에 있는 공지문을 클릭하면 해당 공지문 내용을 화면에 출력한다. 맞춤 공지문을 추천하기 위해서 키워드 기반 자체 알고리즘을 사용하였다.



(그림 4) 유사 게시글 추천 화면

그림 4는 사용자가 현재 '2018년도 재능인, 인천인, 응원 장학생 선발 공고'라는 제목의 공지문을 보고 있을 때 이와 가장 유사한 제목을 가진 웹 공지문 2개가 하단에 추천된 화면이다. 하단에 출력된 유사 공지문 제목을 클릭하면 해당 공지문 내용을 확인할 수 있다. 유사 공지문 제목을 찾기 위해 2-gram 기반 유사문 검색 알고리즘을 사용하였다.



(그림 5) 검색어 '토익'에 대한 추천 검색어 목록 화면

그림 5는 사용자가 입력한 검색어와 유사한 단어를 추천하는 화면이다. 단어를 누르면 관련된 데이터가 검색된다. 유사 단어 추천을 위해 Word2Vec을 사용하였다.

#### 4. 구현 기술

본 논문에서 개발한 안드로이드 앱은 다음과 같이 크게 2개의 모듈로 구성되며, 각 모듈의 세부 기능은 다음과 같다.

##### (1) 분류 모듈

학교 웹 공지문과 학과 웹 공지문을 카테고리별로 자동 분류하는 기능을 제공한다.

- 학습 데이터: 공지문 분류를 위해 학교 및 학과 웹을 크롤링(crawling)하여 공지문을 수집한 후, 학습 데이터로 공지문의 '제목'과 공지문이 속한 게시판 '카테고리'를 사용했다. 글 제목이 글 전체를 요약하여 충분히 글의 내용을 전달할 수 있다.
- 학교 공지문 분류 모듈: 다층 신경망 모델을 사용해 게시글의 제목을 분류한다. 구현을 위해 TensorFlow[5] 위에서 동작하는 딥러닝 라이브러리인 Keras[6]를 사용해서 모델을 생성했다. Word2vec 모델에서 생성한 wakat i 파일을 입력으로 하여 카테고리과 함께 모델을 학습시킨다. 사용된 트레이닝 데이터는 14552 건, 테스트 데이터는 4851 건이다. 모델의 평가 결과는 정확도(accuracy), 정밀도 (precision), 재현율(recall), F1-SCORE 이며, 결과는 아래와 같다.

정확도	정밀도	재현율	F1-score
0.95438	0.95	0.95	0.95

- 학과 공지문 분류 모듈: Naive Bayesian 분류기를 사용해 공지문을 분류한다. 구현을 위해 koNLPy[7]의 twitter 라이브러리[8]를 통해 단어의 출현 빈도 수를 계산하여 모델을 만들었다. 학과 공지문은 '공지'와 '취업'이라는 두 가지 카테고리로 나뉘며, 모델의 정확도는 0.9였다.

##### (2) 추천 모듈

사용자 개인정보에 따른 맞춤형 공지문 추천, 유사 공지문 추천, 유사 검색어 추천 등의 기능을 제공한다.

- 사용자 맞춤 추천 모듈: 공지문의 게시일이 현재 날짜로부터 7일 이내이며 다층 신경망으로 분류된 카테고리 중 사용자가 관심있는 카테고리에 속하거나 사용자 전공에 해당하는 웹 공지문을 추출한다. 그 후 1) 사용자의 관심 카테고리과 전공, 2) 사용자의 개인 정보로 판단했을 때 무관심 키워드(예: 1학년의 경우 '고학년', '수료생', '졸업', '학위복', '학위수여', '학석사', '신입사원', '졸준위', '대학원'), 3) 사용자의 개인 정보로 판단했을 때 관심 키워드(예: 1학년의 경우 '저학년', '신입생', '새내기', '입학식', '3학기', '1학년'), 4) 공지문 게시일과 현재 날짜의 차이를 각각 0~1 사이 값으로 변환해 공지문마다 점수를 매긴다. 수식은 다음과 같다.

score =  
 0.4 \* (사용자 관심 카테고리 또는 전공에  
 0 개 해당되는 경우 0,  
 1 개 해당되는 경우 0.5,  
 2 개 해당되는 경우 1)  
 + 0.3 \* (무관심 키워드가 0 개일 경우 1,  
 1 개 이상일 경우 0)  
 + 0.2 \* (관심 키워드가 0 개일 경우 0,  
 1 개일 경우 0.5,  
 2 개 이상일 경우 1) +  
 + 0.1 \* (1-게시일자와 현재 날짜의 차이/7)

- 유사 게시글 추천 모듈: 2-gram 을 사용해 사용자가 선택한 공지문의 제목과 가장 유사도가 높은 제목을 가진 두 개의 공지문을 찾은 뒤, 이를 유사도 순으로 출력한다.
- 검색어 추천 모듈: 사용자가 검색한 단어를 서버로 전송하고, 서버의 Word2Vec 모델을 이용해 검색어와 가장 벡터 거리가 가까운 단어 4 개를 얻어온다. Word2Vec 은 gensim 라이브러리[9]를 이용하였으며, 모델 학습을 위해 SnoWe 공지사항 (<https://snowe.sookmyung.ac.kr/bbs5/boards/notice>), SnoWe 취업경력개발원 (<https://snowe.sookmyung.ac.kr/bbs5/boards/jobcareer>), 숙명여대 국제협력팀 ([http://cms.sookmyung.ac.kr/wiz5/contents/board/board\\_action.php?home\\_id=exchange&command=list&listhome\\_id=exchange&listhandle=7&handle=7&board\\_id=7](http://cms.sookmyung.ac.kr/wiz5/contents/board/board_action.php?home_id=exchange&command=list&listhome_id=exchange&listhandle=7&handle=7&board_id=7)) 게시글 제목을 활용했다.

## 5. 결론

본 논문에서는 머신 러닝 기술을 활용하여, 사용자가 학교 내 여러 사이트를 번거롭게 살피지 않아도 하나의 애플리케이션에서 모든 공지문을 손쉽게 탐색할 수 있는 공지문 분류 및 추천 애플리케이션을 개발하였다. 해당 애플리케이션은 미분류 상태의 웹 공지문을 알맞게 분류하는 기능 외에도 사용자가 선호할 만한 공지문을 추천하고, 지금 읽은 글과 유사한 글 또는 지금 검색한 키워드와 유사한 검색어를 제안하고, 사용자가 유용하다고 판단한 공지를 따로 스크랩해 저장하는 기능을 제공한다. 이로써 사용자는 기존의 지나치게 복잡하고 필요한 정보를 찾기 힘든 공지 시스템 대신, 모바일 친화적인 환경에서 알맞게 분류된 공지를 효율적으로 쉽게 제공받을 수 있다. 그러므로 학생들이 꼭 필요한 정보를 보지 못하고 지나치거나 불필요하고 관계 없는 공지에 방해 받는 일을 개선할 수 있을 것으로 기대한다.

## Acknowledgment

이 논문은 2018 년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사

업입(No.NRF-2015R1C1A1A02037071).

## 참고문헌

- [1] Multilayer perceptron, [https://en.wikipedia.org/wiki/Multilayer\\_perceptron](https://en.wikipedia.org/wiki/Multilayer_perceptron)
- [2] Naïve Bayes Classifier, [https://en.wikipedia.org/wiki/Naive\\_Bayes\\_classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier)
- [3] Word2vec, <https://en.wikipedia.org/wiki/Word2vec>
- [4] N-gram, <https://en.wikipedia.org/wiki/N-gram>
- [5] TensorFlow, <https://www.tensorflow.org/>
- [6] Keras, <https://keras.io/>
- [7] KoNLPy, <http://konlpy.org/en/v0.4.4/>
- [8] KoNLPy tag Package, <http://konlpy.org/ko/v0.4.3/api/konlpy.tag/>
- [9] gensim, <https://radimrehurek.com/gensim/models/word2vec.html>