토픽 모델을 이용한 모바일 앱 설명 노이즈 제거

윤희근O, 김솔, 박성배
경북대학교
{hkyoon,skim,sbpark}@sejong.knu.ac.kr

Noise Elimination in Mobile App Descriptions Based on Topic Model

Hee-Geun YoonO, Sol Kim, Seong-Bae Park
Kyungpook National University

요 약

스마트폰의 대중화로 인하여 앱 마켓 시장이 급속도로 성장하였다. 이로 인하여 하루에도 수십개의 새로운 앱들이 출시되고 있다. 이러한 앱 마켓 시장의 급격한 성장으로 인해 사용자들은 자신이 흥미를 가질만한 앱들을 선택하는데 큰 어려움을 겪고 있어 앱 추천 방법에 대한 연구에 많은 관심이 집중되고 있다. 기존 연구에서 협력 필터링 기반의 추천 방법들을 제안하였으나 이는 콜드 스타트 문제를 지니고 있다. 이와는 달리 컨텐츠 기반 필터링 방식은 콜드 스타트 문제를 효율적으로 해소할 수 있는 방법이지만 앱 설명에는 광고, 공지사항등 실질적으로 앱의 특징과는 무관한 노이즈들이 다수 존재하고 이들은 앱 사이의 유사관계를 파악하는데 방해가 된다. 본 논문에서는 이런 문제를 해결하기 위하여 앱 설명에서 노이즈에 해당하는 설명들을 자동으로 제거할 수 있는 모델을 제안한다. 제안하는 모델은 모바일 앱 설명을 구성하고 있는 각 문단을 LDA로 학습된 토픽들의 비율로 나타내고 이들을 분류문제에서 우수한 성능을 보이는 SVM을 이용하여 분류한다. 실험 결과에 따르면 본 논문에서 제안한 방법은 기존에 문서 분류에 많이 사용되는 Bag-of-Word 표현법에 기반한 문서 표현 방식보다 더 나은 분류 성능을 보였다.

주제어: 모바일 앱 추천, 노이즈 필터링, LDA

1. 서론

스마트폰의 대중적인 보급으로 인하여 수많은 모바일 앱이 출시되고 있다. 대표적인 스마트폰 앱 마켓인 애플의 앱스토어와 구글 플레이 스토어에는 현재 100만개 이상의 앱이 등록되어 있으며, 매일 수십여 개의 새로운 앱들이 등록되고 있다. 이러한 앱 마켓의 급진적인 성장은 사용자로 하여금 자신이 흥미를 가질만한 새로운 앱을 선택하는데 큰 어려움을 겪게 만들었다. 이로 인해 최근에는 사용자가 관심을 가질만한 앱을 추천해주는 서비스에 대한 연구가 증가하고 있다.

기존의 추천은 크게 협력 필터링(Collaborative filtering)과 컨텐츠 기반 필터링(Content-based filtering), 2가지 유형으로 구분된다. 협력 필터링 방법은 사용자들 사이의 상관관계에 기반하여 비슷한 취향의 사용자들의 정보에 기반하여 추천을 수행한다. 이 방법은 사용자들이 아이템에 대하여 평가한 이력이 충분하게 존재한다면 우수한 추천 성능을 보여주며, Amazon, CDnow 등 다양한 상업 사이트에 적용되었다. 하지만 평가 이력이 존재하지 않는 콜드 스타트 문제에 대해서는 협력 필터링 방법은 매우 취약하다. 특히 새롭게 출시되는 모바일 앱의 경우는 사용자들의 사용 및 평가 이력이 존재하지 않아 심각한 콜드 스타트 문제를 안고 있다. 그렇기 때문에 협력 필터링 방법에 기반한 모바일 앱 추천은 큰 한계를 지니고 있다.

컨텐츠 기반 필터링 방법은 사용자가 관심을 가졌던 과거 컨텐츠와 내용적으로 유사한 컨텐츠를 추천하는 방법이다. 컨텐츠 기반 필터링은 사용 이력이 존재하지 않더라도 컨텐츠의 유사성에 기반하여 추천을 수행하기 때문에 협력 필터링의 콜드 스타트 문제를 효과적으로 해소할 수 있다. 이와 같은 컨텐츠 기반 필터링을 적용하기 위해서는 앱의 내용을 잘 표현할 수 있는 컨텐츠에 대한 정의가 필요하다.

대표적인 모바일 앱 스토어인 구글 플레이 스토어와 애플의 앱스토어는 앱의 특징을 잘 표현할 수 있는 앱의 설명이 기술되어 있다. 이들 내용은 개발자가 직접 앱의 특징을 기술한 내용이므로 컨텐츠 기반 필터링 방법에 사용하기에 적합하다. 하지만 모바일 앱 설명의 모든 부분이 앱의 특징을 설명하고 있는 것은 아니다. 예를 들어 그림 1은 앱 설명에서 실질적으로 앱과 관계없는 노이즈에 대한 예를 보여주고 있다. 앱의 설명 부분은 앱의 직접적인 내용이나 특징을 설명하기도 하지만 때로는 앱의 내용과는 무관한 이벤트, 공지사항 등의 노이즈 내용이 포함되어 있다. 노이즈 설명들과 앱의 설명을 표현하고 있는 부분들은 서로 다른 문단에 포함되어 구성되어 있음을 볼 수 있다. 노이즈 문단들은 전체 앱 설명에 섞여 앱 설명들 사이의 유사도를 바탕으로 앱의 유사성을 측정하는데 방해가 된다,

본 논문에서는 모바일 앱 추천의 첫 단계로 앱 설명의 노이즈를 제거하기 위하여 Latent Dirichlet Allocation (LDA)와 Support Vector Machine (SVM) 모델에 기반한 앱 설명 문단 노이즈 제거 모델을 제안한다. 개발자가 작성한 앱 설명에는 앱의 특징을 설명하는 문단들과 앱의 특징과는 무관한 노이즈 문단들이 혼재되어 나타난

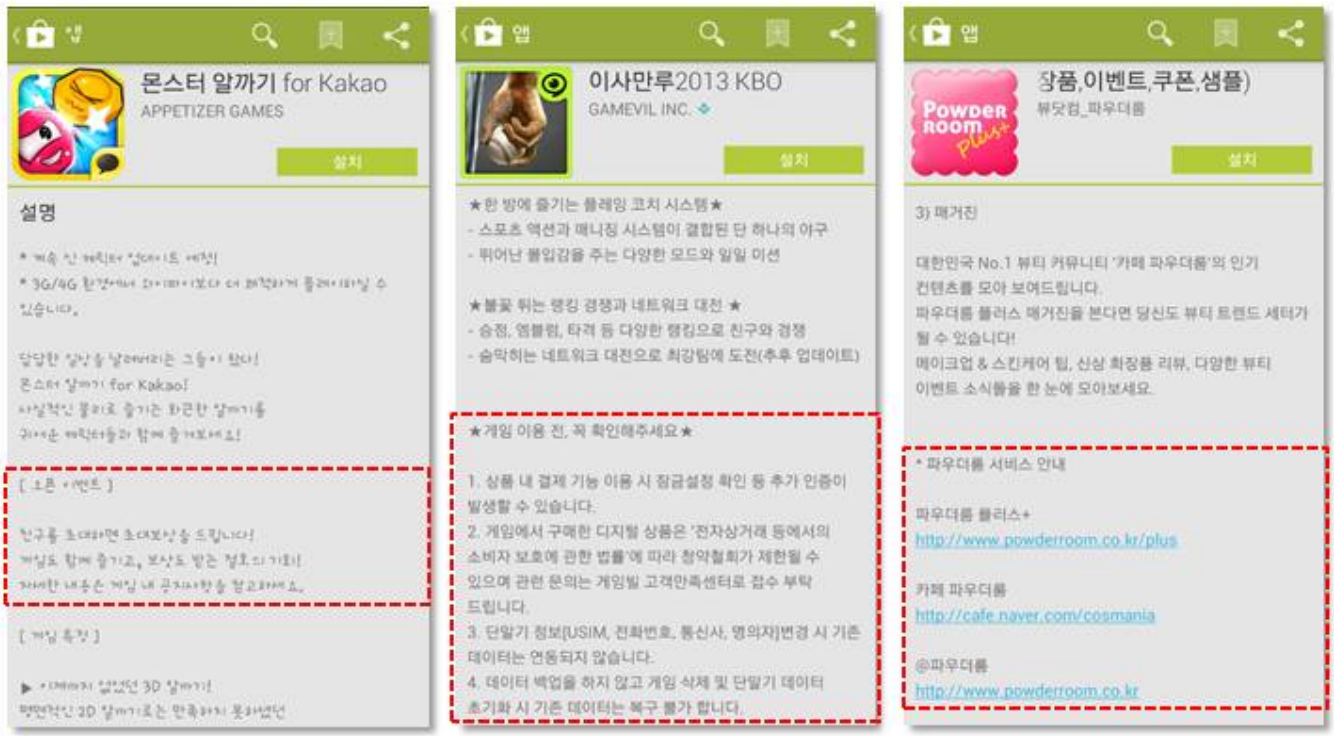

그림 1. 앱 설명에 포함되어 있는 노이즈 문단의 예

다. 일반적으로 앱 설명들은 일반적인 문서 분류 문제에서 사용되는 문서들에 비해 길이가 짧다는 특징이 있다. 특히 본 논문에서 분류의 대상으로 삼고 있는 단위는 문단이기 때문에 훨씬 적은 수의 단어로 구성되어 있다. 이런 이유로 문서 분류 문제에서 자주 적용되는 Bag-of-Word(BOW) 기반의 문서 표현법을 사용할 경우 데이터 부족 문제 (data sparseness problem)으로 인해 성능이 저하되는 문제가 발생할 수 있다. 이에 본 논문에서 제안하는 방법은 토픽 모델에 기반하여 앱 설명을 토픽들의 구성 비율로 표현하고 이를 바탕으로 SVM을 이용하여 분류하는 방법론을 제안한다. 실험 결과에 따르면 제안한 방법은 약 61%의 정확도를 보이는데, 이는 약 54%의 성능을 보이는 BOW 기반의 표현법을 사용하는 모델보다 훨씬 나은 성능을 보이며 제안한 모델이 앱 설명에서 노이즈를 제거하기에 적합함을 보여준다.

본 논문의 구성은 다음과 같다. 2장에서는 모바일 앱 추천 시스템과 토픽 모델링에 대한 기존 연구를 살펴보고, 3장에서는 LDA 모델에 기반한 문서 표현법과 이를 바탕으로 노이즈 문단을 분류하는 방법에 대하여 설명한다. 4장에서는 제안한 모델의 성능을 평가하기 위한 실험을 설명하고 성능을 평가한다. 5장에서는 결론을 다룬다.

2. 관련 연구

최근 모바일 앱 시장의 폭발적인 성장으로 인하여 모바일 앱과 관련한 다양한 연구들이 이루어지고 있다. 특히 모바일 앱 추천 또한 큰 관심을 받고 있다. 노우현[1] 등은 사용자의 앱 사용 이력과 상황을 고려하여 앱의 카테고리를 추천하는 모델을 제안하였다. 해당 모델에서는 각 상황별 각 카테고리의 적합도를 계산하기 위하여 베이지안 모델을 이용하였다.

많은 기존의 연구들은 협력적 필터링 기반 방법을 통한 추천을 제안하였다. Yan et al.[2] 은 협력적 필터링 기반 시스템 AppJoy를 소개하였다. AppJoy는 자체적으로 개발한 스마트폰 앱을 통해 사용자 사용 이력들을 수집하고, 협력적 필터링을 통하여 모바일 앱의 추천을 수행한다. 사용자의 앱 사용 이력의 누락으로 인한 콜드 스타트 문제를 풀기 위하여 새 사용자의 경우 초기에 사용자의 기기에 설치되어 있는 앱들을 바탕으로 추천을 수행하였다. Ozaki et al.[3]은 기존의 협력적 필터링과 사용자들 사이의 사회적인 관계까지 함께 고려하는 시스템을 제안하였다. 이 시스템에서는 일반적인 협력적 필터링 방법에 기반하여 추천 앱을 선정한 뒤, 다시 사용자들 사이의 사회적인 관계를 반영하여 앱 추천 스코어를 재평가하는 방법으로 적용하였다. 이 두 방법 모두 높은 성능을 보여주었으나, 협력적 필터링 방법의 한계로 인하여 새롭게 출시되어 평가 이력이 존재하지 않는 앱들에 대해서는 추천을 수행할 수 없는 문제가 존재하였다.

Lin et al.[4]은 협력적 필터링 방법의 콜드 스타트 문제를 해결하기 위하여 외부 자원을 활용하는 방법을 소개하였다. 이 시스템에서는 대표적인 SNS서비스인

twitter를 이용하여 사용자의 사용 이력과 앱의 평가 이력의 부족을 해소하고자 하였다. 이 논문은 앱의 추천을 위해서 외부자원을 효율적으로 활용하는 방안을 제시하였으나, 이 역시도 앱 추천을 위해서 앱 이외의 자원에 관한 이력이 존재해야 한다는 면에서 콜드 스타트 문제를 원천적으로 해결하지는 못하였다.

Kim et al.[5]은 모바일 앱 추천을 위하여 콘텐츠 기반 유사도 방법론을 소개하였다. 이 모델에서는 모바일 앱의 콘텐츠를 정의하기 위하여 모바일 앱을 위한 온톨로지를 설계하고 이를 바탕으로 각 앱의 콘텐츠를 정의하였다. 앱에 관하여 개발자, 이름, 다운로드 수 등 다양한 정보를 반영하였다. 하지만 앱 개발자가 직접적으로 앱 특징에 대하여 기술한 앱 설명 정보를 활용하지 않아 많은 정보 손실을 야기하였다.

문서 분류 문제에 토픽 모델을 도입하기 위하여 다양한 연구가 이루어졌다. Rubin et al.[6]은 멀티레이블 문서를 분류하기 위한 토픽 모델을 제안하였다. 기존 LDA를 멀티레이블 문서 분류에 사용하기 위하여 확장한 Flat-LDA, Prior-LDA, Dependency-LDA 모델을 제안하였다. 비록 판별 모델인 SVM에 비해 낮은 성능을 보여주긴 하였지만 그 성능 차이가 크지 않아 토픽 모델을 이용하여 멀티레이블 문서 분류를 구생할 수 있음을 보여주었다. Zhou et al.[7] 역시 LDA에 기반한 LDACLM을 제안하였다.

3. 앱 설명의 노이즈 제거

분류 모델을 이용하여 분류하기 위해서는 문서들을 컴퓨터로 처리할 수 있는 형태로 표현할 수 있는 방법이 필요하다. 본 장에서는 LDA에 기반하여 앱 설명 문단들을 벡터로 표현하는 방법과 이를 이용하여 노이즈 문단을 분류할 수 분류 방법을 설명한다.

3.1. Latent Dirichlet Allocation

LDA는 Blei[8]에 의해서 제안된 대표적인 토픽 모델 중 하나로, 하나의 문서는 다양한 토픽의 혼합으로 구성되어 있다고 가정하는 모델이다. LDA에서는 문서들이 자신이 가진 토픽들로부터 생성된 단어들로 표현된다고 본다. 여기서 토픽이란 문서를 구성하는 단어들 중 서로 연관성이 높은 단어들의 집합으로 볼 수 있다. 이러한 토픽들은 다항분포로서 정의되며 각 문서는 자신이 가진 토픽의 분포와 각 토픽들이 가진 단어들의 분포에 기반하여 추출된 단어들로 구성된다. 그리고 이렇게 추출된 단어들이 나열되어 해당 문서가 작성되는 것으로 본다. 그림 2는 LDA의 그래피컬 표현을 나타낸다. LDA는 다음과 같은 문서 생성 과정을 모델링한다.

- 문서 d가 가지고 있는 토픽들의 다항분포 θ_d를 Dirichlet 분포인 $\theta_d \sim Dir(\alpha)$로부터 추출한다.
- 각 토픽 k에 대한 단어들의 다항분포 Φ_k를 Dirichlet 분포인 $\Phi_k \sim Dir(\beta)$로부터 추출한다.
- 이를 바탕으로 i번째 단어 w_i는 다음과 같이 추출된다.

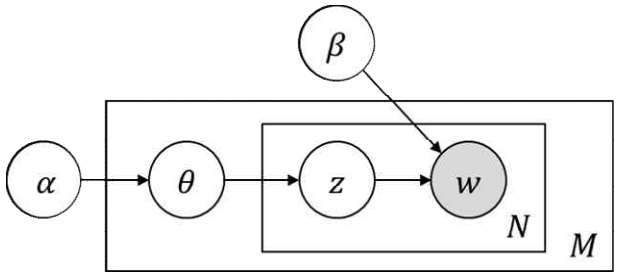

그림 2. LDA의 그래피컬 표현

- i번째 토픽 z_i를 다항분포 $z_i \sim Multinomial(\theta_d)$로부터 추출한다.
- i번째 단어 w_i를 다항분포 $w_i \sim Multinomial(\Phi_{z_i})$로부터 추출한다.

이 과정에서 모델이 추정해야 할 값은 하이퍼 파라메터인 α와 β이다. LDA는 주어진 학습 문서들을 이용하여 EM기반의 방법으로 로그 우도가 최대가 되는 α와 β를 찾는다. 이렇게 α와 β가 학습이 되면 새로운 문서의 단어 집합 $\mathbf{w}$가 주어질 때, 이 문서의 토픽들의 분포인 θ는 다음과 같이 구해질 수 있다.

$$p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{z}|\alpha, \beta)}$$

여기서 구해진 새로운 문서의 토픽 분포 θ는 해당 문서를 구성하고 있는 단어들을 대신하여 해당 문서를 벡터로 표현하는데 사용될 수 있다.

본 논문에서 제안하는 방법은 LDA 모델에 기반하여 앱 설명 문단들을 단어들이 아닌 각 토픽의 구성 비율로 표현한다. 기존에 문서 분류에서 자주 사용되는 BOW의 경우 문단의 길이가 짧고 데이터의 수가 많지 않을 경우에 데이터 부족 현상 때문에 성능이 저하되는 문제가 있다. 특히 본 논문에서 분류 대상으로 삼고 있는 단위는 앱 설명의 문단이기 때문에 그 길이가 매우 짧아 데이터 부족 현상에 의해 큰 영향을 받을 수 있다. 이에 본 논문에서는 BOW 모델 대신에 LDA 모델을 이용하여 문서를 어휘 수 보다 훨씬 적은 차원인 토픽으로 표현함으로써 데이터 부족 문제에 좀 더 강건한 데이터를 생성할 수 있도록 한다.

3.2. Support Vector Machine

SVM은 Vapnik[9]의 의해서 제안된 모델로 매우 우수한 성능을 보이는 이진 분류 모델 중 하나이다. SVM은 두 클래스의 데이터가 주어졌을 때, 두 클래스의 데이터를 잘 분류할 수 있는 초평면(hyperplane)을 찾는 모델이다. 경우에 따라서 두 클래스를 완전히 구분할 수 있는 초평면이 매우 많거나 또는 무한하게 존재할 수 있는데,

표 1. 실험 데이터 통계

속성	값
카테고리의 수	25
앱의 수	703
문단의 수	4,500
문단 별 평균 단어 수	24.33
노이즈 문단 수	2,104
비 노이즈 문단 수	2,396

이때 초평면에 가장 가까운 각 클래스의 데이터와 초평면 사이의 거리를 의미하는 마진(margin)이 가장 최대가 되는 초평면을 찾는다. 이렇게 찾은 초평면을 이용하여 새로운 데이터가 주어졌을 때, 해당 데이터의 클래스를 분류한다.

데이터 집합 $D=\left\{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in R^m, y_i \in \{-1,+1\}\right\}_{i=1}^{n}$가 주어지면, SVM의 초평면은 다음과 같이 정의될 수 있다.

$$H = \mathbf{w} \cdot \mathbf{x} + \mathrm{b}$$

SVM에서는 $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1$을 만족하면서 $\|\mathbf{w}\|$이 최소화되는 $\mathbf{w}$를 찾음으로써 최적의 초평면을 찾는다. 그리고 이렇게 학습된 초평면을 이용하여 새로운 데이터 x가 주어지면 다음과 같이 새로운 데이터의 클래스를 추정한다.

$$y = \begin{cases} 1 & \text{if } w \cdot x_i - b \geq 1 \\ -1 & \text{if } w \cdot x_i - b \leq -1 \end{cases}$$

4. 실험

본 논문에서 제안한 방법의 성능을 보이기 위하여 실제 모바일 앱의 설명들을 수집하여 실험을 수행하였다. 실험을 위하여 플레이 스토어[1)]에 등록되어 있는 모바일 앱들의 설명을 수집하였다. 표 2는 실험에 사용된 데이터의 통계를 보여준다. 플레이 스토어에 존재하는 전체 25개 카테고리의 703개의 앱에 대한 설명을 수집하였다. 이들 703개의 앱 설명을 html 태그에 기반하여 문단 단위로 분리하였다. 각 앱 설명에서 특수기호 및 URL은 제거하였다. 길이가 짧은 문단의 경우 앱 설명의 각 문단의 제목을 나타내는 경우가 대부분이기 때문에 문단 내에 포함된 음절의 길이가 15이하인 문단은 제외하였다. 최종적으로 남은 4,500개의 문단에 대하여 노이즈 여부를 수작업으로 판단하였다. 사용자가 모바일 앱의 카테고리와 문단을 함께 고려하여 각 문단의 노이즈 여부를 판단하였다. 실험에 사용된 문단들은 평균적으로 24.33개의 단어로 구성되어 있으며 전체 테스트 데이터 중 약 47%가 노이즈 문단으로 구성되어 있다. 전체 4,500개의 데이터 중 80%는 모델의 학습에 사용되었고 나머지 20%는 성능 측정하기 위한 테스트 데이터로 사용되었다.

1) http://play.google.co.kr

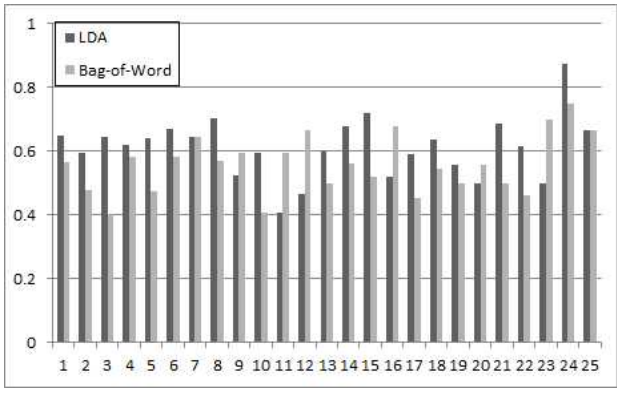

그림 3. 카테고리별 분류 정확도

표 2. 전체 카테고리에 대한 정확도

문단 표현 방법	정확도 (%)
Bag-of-Word	54.09
LDA	61.52

LDA 모델의 파라매터 중 토픽의 수는 100개로 지정하였다. 그리고 각 분산의 하이퍼 파라매터(Hyper Parameter) α와 β의 초기값은 0.1로 지정하였으며 학습 반복 횟수는 총 1,000회로 지정되어 학습되었다. SVM 분류모델에서 LDA 표현 기반 데이터는 다항 커널을 이용하였고 BOW 표현에 기반한 데이터는 선형 커널을 이용하여 수행하였다. 실험에 사용된 모든 파라매터는 실험적으로 각 모델에서 가장 우수한 성능을 보여준 값으로 선택되었다.

실험은 수집된 데이터 셋에 포함된 총 25개 카테고리에 대해서 독립적으로 수행되었다. 그림 3은 각 카테고리별 노이즈 문단의 분류 성능을 보여준다. 그래프의 가로축 카테고리 목록과 각 카테고리 별 데이터 비율은 표 4에 나타나 있다. 제안한 방법으로 표현된 문서 기반 분류 성능은 BOW로 표현된 데이터에 비하여 훨씬 많은 카테고리에서 더 높은 성능을 보여주었다. 특히 실험데이터에서 큰 비중을 차지하고 있는 카테고리들에서 비슷하거나 더 높은 분류 정확도를 보여주었다. 표 3은 전체 데이터 집합에 대한 분류 정확도를 보여준다. BOW 모델을 이용하여 표현된 문단들은 54.09%의 정확도를 보여주었다. 하지만 본 논문에서 제안하는 LDA에 기반한 문단들은 61.52%의 분류 정확도를 보여주었다. 비록 LDA에 기반한 모델이 BOW 표현에 기반한 모델보다 우수한 성능을 보여주긴 하나, 전반적으로 낮은 정확성을 보여주었다.

낮은 정확성의 원인을 파악하기 위하여 오류 데이터를 분석해보았다. 해당 데이터들을 분석해본 결과 많은 양의 앱들이 잘못된 카테고리로 분류되어 있는 것을 확인할 수 있었다. 앱의 카테고리는 개발자의 의해서 결정되는 것으로 명확한 규정이 존재하지 않아 오분류된 앱들

표 3. 실험에 사용된 앱 카테고리 및 데이터 수

번호	카테고리	데이터 수	번호	카테고리	데이터 수	번호	카테고리	데이터 수
1	게임	707	2	교육	478	3	라이프스타일	309
4	데코레이션	274	5	도서 및 참조자료	264	6	도구	234
7	음악 및 오디오	220	8	엔터테인먼트	216	9	여행 및 지역정보	208
10	커뮤니케이션	157	11	소셜 콘텐츠	154	12	미디어 및 동영상	148
13	건강 및 운동	146	14	비지니스	122	15	만화	122
16	사진	122	17	생산성	110	18	금융	108
19	교통	86	20	스포츠	86	21	의료	76
22	쇼핑	61	23	날씨	44	24	뉴스 및 잡지	35
25	라이브러리 및 데모	13						

이 매우 많이 존재한다. 앱의 설명에 기술되어 있는 문단들은 내용이 비슷해 보이더라도 카테고리에 따라 다르게 해석될 수 있다. 이에 본 논문에서는 노이즈 분류를 위한 모델을 카테고리별로 구축하여 실험을 수행하였는데, 잘못된 카테고리에 포함된 수많은 앱들이 영향으로 인해 분류 성능이 낮음을 확인할 수 있었다. 예를 들어 'ZLOTUS는 GO 실행기 테마를 사랑' 이라는 앱은 핸드폰을 꾸미기 위한 테마 앱으로 데코레이션 또는 도구 카테고리가 존재하지만, 실제로 이 앱은 만화 카테고리에 포함되어 있었다. 이런 앱들의 영향으로 인하여 BOW와 LDA에 기반한 모델 모두 전반적으로 낮은 분류 정확성을 보여주었다. 하지만 본 논문에서 제안한 모델은 동일한 환경 하에서 BOW에 비하여 더 우수한 분류 성능을 보여주어 앱 설명에서 노이즈를 제거하기에 적합함을 보여주었다.

5. 결론

본 논문에서는 앱 설명의 노이즈를 제거하기 위하여 앱 설명을 토픽으로 표현하여 분류하는 모델을 제안하였다. 제안한 방법은 앱 설명의 각 문단을 대표적인 토픽 모델인 LDA를 통해 표현하고 이들 데이터를 분류 문제에서 우수한 성능을 보여주는 SVM을 이용하여 분류한다. 제안한 방법은 앱 설명에 사용된 단어를 그대로 사용하지 않고 이를 토픽으로 표현함으로써 데이터 부족 문제에 대하여 강건한 노이즈 제거 방법이다. 이를 통해 BOW 모델에 비하여 우수한 성능을 보여주었다. 실험 결과에 의하면 BOW에 기반한 모델은 모든 카테고리에 대하여 54.09%의 정확도를 보여주는 반면 제안한 모델은 61.52%의 성능을 보여주었다. 비록 앱들의 카테고리 오분류 문제에 의해 전반적으로 낮은 분류 성능을 보여주었지만, 모바일 앱의 설명에서 노이즈 문단을 제거하기에 BOW에 기반한 모델보다 본 논문에서 제안한 LDA에 기반한 모델이 더욱더 적합함을 보여주었다.

현재 앱 마켓에는 이미 수백만 개의 앱들이 등록되어 있으며 이들을 설명하고 있는 문서의 수도 매우 방대하다. 이들 앱 설명을 수집하는 것은 어렵지 않으나 모델의 학습을 위하여 각 문단의 정답을 수작업으로 부착하는 것은 매우 큰 비용이 발생하는 문제이다. 향후 연구로 이미 존재하는 대량의 앱 설명을 큰 비용 없이 효율적으로 활용하기 위하여 반지도 또는 비지도 학습 방법에 기반한 분류 모델을 연구할 예정이다. 또한 본 논문의 실험에서 성능저하를 일으킨 앱의 카테고리 오분류에 의한 성능저하를 효율적으로 해결할 수 있는 모델에 대한 연구 또한 함께 진행할 예정이다.

감사의 글

본 논문은 지식경제부 산업원천기술개발사업(10035348, 모바일 플랫폼 기반 계획 및 학습 인지 모델 프레임워크 기술 개발)의 지원으로 수행되었음.

참고문헌

[1] 노우현, 조성배, "베이지안 네트워크를 이용한 상황별 모바일 앱 카테고리 추천 시스템", 한국정보과학회 2013 한국컴퓨터종합학술대회 논문집, pp.1408-1410, 2013.

[2] B. Yan and G. Chen, "AppJoy: Personalized Mobile Application Discovery", MobiSys '11 Proceedings of the 9th international conference on Mobile systems, applications, and services, pp.113-126, 2011.

[3] T. Ozaki and M. Ehoh, "Experimental Analysis of the Effects of Social Relations on Mobile Application Recommendation", Proceedings of the International MultiConference of Engineers and COmputer Scientists, 2012.

[4] J. Lin, K. Sugiyama, M. Kan and T. Chua, Addressing Cold-Start in App Recommendation: Latent User Models Constructed from Twitter Followers", Proceedings of SIGIR 2013, 2013.

[5] J. Kim, S. Kang, Y. Lim and H. Kim, "Recommendation algorithm of the app store by using semantic relations between apps", The Journal of Supercomputing, vol.65, pp.16-26, 2011.

[6] Timothy N. Rubin, America Chambers, Padhraic Smyth and Mark Steyvers, "Statisitcal topic

models for multi-label document classification", Journal of Machine Learning, vol.88, pp.157-208, 2012.
[7] Shibin Zhou, Kan Li and Yushu Liu, "Text Categoization Based on Topic Model", International Journal of Computational Intelligence Systems, vol.2, no.4, pp.398-409, 2009.
[8] David Blei, Andrew Ng, Michael Jordan, "Latent Dirichlet allocation", Journal of Machine Learning Research, vol.3, pp.993-1022, 2003.
[9] Corinna Cortes and Vladimir N. Vapnik, "Support-Vector Networks", Machine Learning, vol.20, 1995.