Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Emerging topic identification from app reviews via adaptive online biterm topic modeling*

Wan ZHOU[1], Yong WANG[†‡1,2], Cuiyun GAO[3], Fei YANG[4]

[1]*School of Information and Computer, Anhui Polytechnic University, Wuhu 241000, China*

[2]*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210000, China*

[3]*School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518000, China*

[4]*Zhejiang Lab, Hangzhou 310000, China*

[†]E-mail: yongwang@ahpu.edu.cn

Received Sept. 30, 2021; Revision accepted Dec. 2, 2021; Crosschecked Mar. 1, 2022; Published online Apr. 11, 2022

Abstract: Emerging topics in app reviews highlight the topics (e.g., software bugs) with which users are concerned during certain periods. Identifying emerging topics accurately, and in a timely manner, could help developers more effectively update apps. Methods for identifying emerging topics in app reviews based on topic models or clustering methods have been proposed in the literature. However, the accuracy of emerging topic identification is reduced because reviews are short in length and offer limited information. To solve this problem, an improved emerging topic identification (IETI) approach is proposed in this work. Specifically, we adopt natural language processing techniques to reduce noisy data, and identify emerging topics in app reviews using the adaptive online biterm topic model. Then we interpret the implicature of emerging topics through relevant phrases and sentences. We adopt the official app changelogs as ground truth, and evaluate IETI in six common apps. The experimental results indicate that IETI is more accurate than the baseline in identifying emerging topics, with improvements in the F1 score of 0.126 for phrase labels and 0.061 for sentence labels. Finally, we release the codes of IETI on Github (https://github.com/wanizhou/IETI).

Key words: App reviews; Emerging topic identification; Topic model; Natural language processing
https://doi.org/10.1631/FITEE.2100465 **CLC number:** TP311.5

1 Introduction

App reviews, the most straightforward feedback of users' immediate experience, contain subjective and objective evaluations of software features (Nguyen et al., 2015), such as praise or criticism for a software function. Emerging topics in app reviews refer to topics related to app features with which users are concerned during certain periods, such as new bugs that affect user experience (e.g., software crash) or existing undesirable features (e.g., too many advertisements) (Gu and Kim, 2015).

Identifying emerging topics accurately and quickly can guide developers in updating their apps. According to statistics, there were no less

‡ Corresponding author

* Project supported by the Anhui Provincial Natural Science Foundation of China (No. 1908085MF183), the National Natural Science Foundation of China (Nos. 62002084 and 61976005), the Training Program for Young and Middle-Aged Top Talents of Anhui Polytechnic University, China (No. 201812), the Zhejiang Provincial Natural Science Foundation of China (No. LQ21F020004), the State Key Laboratory for Novel Software Technology (Nanjing University) Research Program, China (No. KFKT2019B23), the Open Research Fund of Anhui Key Laboratory of Detection Technology and Energy Saving Devices, Anhui Polytechnic University, China (No. DTESD2020B03), and the Stable Support Plan for Colleges and Universities in Shenzhen, China (No. GXWD20201230155427003-20200730101839009)

ORCID: Wan ZHOU, https://orcid.org/0000-0002-5024-958X; Yong WANG, https://orcid.org/0000-0002-2719-1017

than four million apps available in the Google Play and Apple App Store as of the fourth quarter of 2020 (https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/). To maintain the competitiveness and popularity of apps, it is crucial to continuously update rich features while maintaining user-friendly experience (McIlroy et al., 2016). Moreover, emerging topics can provide informative evidence for app developers to know what is going on with their apps, so they can effectively maintain and update apps by repairing software bugs and adding new functions to meet user needs (Sarro et al., 2015). For example, a popular game app in China named Gray Raven Punishing suffered from a lot of bad reviews on app stores in December 2019. The reason is that the probability of drawing card changed in releasing new versions and that developers failed to identify bugs quickly (https://www.bilibili.com/read/cv4175642/). This situation could have been mitigated if emerging issues had been identified in a timely manner from app reviews.

To the best of our knowledge, the most recent work on emerging topic identification is IDentify Emerging App (IDEA) issues (Gao et al., 2018), which can be directly applied to identify emerging topics from app reviews. Specifically, different versions of app reviews act as the input, and IDEA proposes adaptive online latent Dirichlet allocation (AOLDA) based on the topic model online latent Dirichlet allocation (OLDA) (AlSumait et al., 2008) to capture topic evolution and identify emerging topics.

However, there are two characteristics of app reviews that prior studies fail to consider, which limits their ability to accurately identify emerging topics. First, app reviews are generally short in length and provide limited context. According to Chen et al. (2014) and Genc-Nayebi and Abran (2017), the average length of an app review was 71 characters and only about 30% of reviews could provide valuable information for updating apps. The sampling algorithm often fails to converge due to the information sparsity in short text. Second, the impact of misspelled words and abbreviations in app reviews has rarely been considered in identifying emerging topic (Noei et al., 2021). Because application text is frequently dissimilar to standard language, topic models are likely to interpret it as a distinct cluster, resulting in an increase in the number of incorrectly identified topics.

In this study, we propose an improved emerging topic identification approach named IETI, to solve the above problems and accurately identify emerging topics in app reviews. Specifically, we adopt natural language processing techniques to reduce noisy data, including correcting misspelled words and extending abbreviations. To address the brevity characteristic of app reviews, we track topic evolutions using the adaptive online biterm topic model (AOBTM) (Hadi and Fard, 2020) and identify emerging topics by outlier detection methods.

To validate the effectiveness of IETI, we adopt the official app changelogs as our ground truth, and calculate semantic similarity between emerging topics and app changelogs to determine the validity of emerging topics. We conduct experiments on the same six popular apps, which are from Google Play and Apple App Store. Experimental results show that IETI can identify emerging topics more accurately than the baselines, with improvements in the F1 score of 0.126 in phrase labels and 0.061 in sentence labels.

The main contributions are as follows: (1) We suggest more sophisticated preprocessing methods for reducing noise in app reviews, including the correction of misspelled words and abbreviations, and more stringent filtering rules. (2) We apply the AOBTM topic model to track topic evolution in app reviews, and propose IETI to identify emerging topics in app reviews. (3) We evaluate the effectiveness of IETI by conducting experiments on six popular apps, and release the IETI codes on Github (https://github.com/wanizhou/IETI).

2 Preliminaries

2.1 App review emerging topic

According to Huang et al. (2017), the term "emerging topics" refers to topics that are diffusely discussed in the current time slice, but seldom mentioned in previous time slices. We consider the example in Gao et al. (2019). Fig. 1 shows the statistics for the number of reviews on the topic "sound" from June 29, 2017 to July 8, 2017. After releasing WeChat version X on July 5, the number of reviews related to "sound" rose abruptly. According to

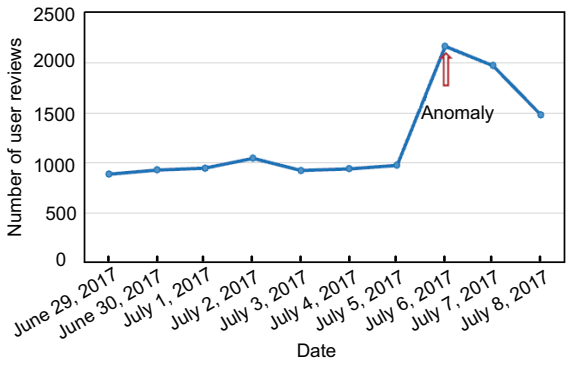

Fig. 1 The number of user reviews related to "sound" from June 29, 2017 to July 8, 2017

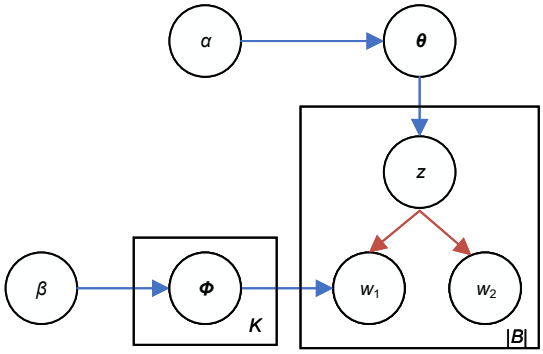

Fig. 2 Schematic of the biterm topic model (BTM)

Huang et al. (2017), the topic "sound" is an emerging topic. In reality, the version X has a functional bug when sharing sounds and pictures, leading to the rise of the topic "sound" in WeChat. The following definition of emerging topic is given:

Definition 1 (Emerging topic of app reviews) One topic that has rarely been discussed in reviews during previous time slices but is mentioned in many user reviews in the current time slice, is defined as an app review emerging topic. Notably, the time slices referred to in this paper are divided into different app versions, which means that each time slice corresponds to a specific app version.

2.2 Topic model

The benchmark topic models based on the Dirichlet hypothesis include mainly latent Dirichlet allocation (LDA), Dirichlet multinomial mixture (DMM), correlated topic model (CTM), and biterm topic model (BTM) (Jin et al., 2018). Due to the sparseness of short texts, the LDA sampling algorithm (Gibbs sampling) may not converge completely in computation, which ultimately reduces the accuracy of topic detection results. Details about the LDA model are available in Blei et al. (2003).

BTM solves the sparse text problem by constructing biterms to improve the accuracy of short text clustering (Li CL et al., 2017). Fig. 2 shows the schematic of BTM. Different from LDA, BTM is not concerned with whether the document belongs to one or more topics; it assumes that two words in each biterm belong to the same topic, and that each topic is a polynomial distribution of words. The generation process of each biterm is as follows:

1. Generate the topic distribution vector $\boldsymbol{\theta}$ from the Dirichlet prior distribution with parameter α : $\boldsymbol{\theta} \sim \mathrm{Dir}(\alpha)$.

2. For each topic k $(k = 1, 2, \ldots, K)$, the topic-word distribution vector $\boldsymbol{\Phi}_k$ is generated from the Dirichlet prior distribution with parameter β: $\boldsymbol{\Phi}_k \sim \mathrm{Dir}(\beta)$.

3. For each biterm $b = \{w_1, w_2\}$, the topic distribution of b is generated from $\boldsymbol{\theta}$: $z_b \sim \mathrm{Multi}(\boldsymbol{\theta})$. For the i^{th} position in b, the term $w_{b,i}$ is generated by the topic distribution z_b : $w_{b,i} \sim \mathrm{Multi}\left(\boldsymbol{\Phi}_{z_b}\right)$.

BTM obtains the hidden variables $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$ by maximizing the following joint probability:

$$\begin{aligned} L(B) &= \prod_{b \in B} \sum_{z} p(b, z, \boldsymbol{\theta}, \boldsymbol{\Phi} | \alpha, \beta) \\ &= \prod_{b \in B} \sum_{z} p(z, \boldsymbol{\theta}, \boldsymbol{\Phi} | b, \alpha, \beta), \end{aligned} \tag{1}$$

where B is the set of all biterms. Then BTM employs the collapsed Gibbs sampling algorithm to convert the conditional probability $p(z, \boldsymbol{\theta}, \boldsymbol{\Phi} | b, \alpha, \beta)$ into the conditional probability $p\left(z_b | z_{-b}, B, \alpha, \beta\right)$ of a topic probability distribution z_b on the remaining topics:

$$\begin{aligned} & p\left(z_b | z_{-(b)}, B, \alpha, \beta\right) \\ & \propto \left(n_k + \alpha\right) \frac{\left(n_{w_i|k} + \beta\right)\left(n_{w_j|k} + \beta\right)}{\left(\sum_w n_{w|z} + V\beta\right)^2}, \end{aligned} \tag{2}$$

where n_k is the number of times that biterm b is assigned to topic z, $n_{w_i|k}$ is the number of times that word w_i is assigned to topic k, and $V\beta$ is the number of unique words in the corpus.

2.3 App changelogs

App changelogs describe the improvements that developers made to the app in new versions. Although app changelogs may not cover all the changes to the previous version of the app, they represent the prescribed minimum and the most significant

parts of the change. For example, Fig. 3 shows the changelogs of the shopping app Ebay in version 6.12.1, and we can observe that Ebay fixed mainly a few bugs that cause functional errors in version 6.12.1.

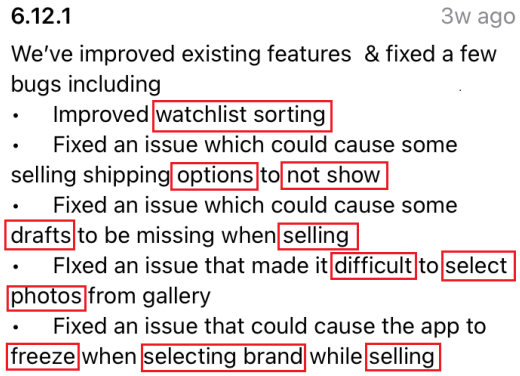

Fig. 3 Changelogs for Ebay in version 6.12.1
The rectangles highlight the main keywords in the updated content of version 6.12.1

If the app emerging topics identified in this study are semantically similar to the keywords described in app changelogs, then these topics can be considered to be covered in changelogs and have practical significance in guiding developers in updating and maintaining their apps. Therefore, we adopt app changelogs as the ground truth in verifying the accuracy of the identified emerging topics. Specific evaluation methods will be introduced in the following section.

3 Methodology

Fig. 4 exhibits the IETI framework. App reviews of diverse versions and the official changelogs serve as the input to the IETI. The outputs contain the evaluation score for identified emerging topics and the relevant phrases and sentences that are used to interpret the meaning of the identified emerging topics.

IETI consists of three phases. In the first phase, we preprocess app reviews to reduce noisy data, and the outputs include processed reviews and candidate topic labels. In the second phase, emerging topics are identified by AOBTM and anomaly detection methods. In the third stage, emerging topics are interpreted by labels, and the explanatory labels are divided into two types: one is the phrases most relevant to the topics, and the other is the sentences most relevant to the topics. Finally, we evaluate the effectiveness of emerging topics through the app changelogs.

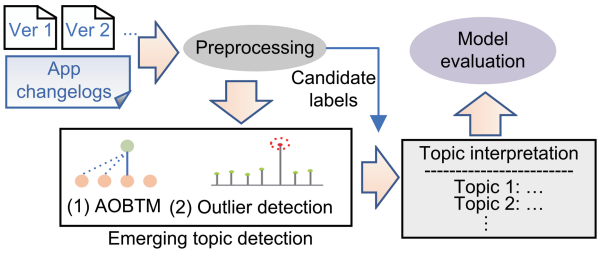

Fig. 4 Overview of the improved emerging topic identification (IETI)

3.1 Preprocessing

Because the submission of app reviews is limited to mobile devices and inconvenient keyboards, app reviews contain a mass of noisy data, such as misspelled words, abbreviations, and repetitive words. In this subsection, we preprocess initial app reviews to reduce noisy data and extract key phrases.

1. Word formatting

We cite some preprocessing steps in IDEA. First, we make all words lowercase. Then we use the natural language toolkit NLTK (http://www.nltk.org) for lemmatization and use "<digit>" to replace all numbers.

Because abbreviations and misspelled words may increase the number of misidentified topics, we need to restore abbreviations and correct misspelled words. Specifically, we restore abbreviations with the NLTK toolkit. Then we implement the fine-tuning task for correction of misspelled words using the open-source framework PyCorrector. Because the default model of PyCorrector adopts the Chinese pre-training model, we select the English pre-training model Bert-base-uncased (Devlin et al., 2019) from huggingface and introduce the custom dictionary Wiki Dictionary to enhance the effect of text error correction.

There are still some meaningless words in the reviews, such as articles (a, an, the) and personal pronouns (I, you, ...). These words are necessary elements of sentence formation, but their contribution to the identification of emerging topics has been neglected. Therefore, we combine the stop word list provided by NLTK with our stop word list to filter these words.

2. PMI phrase extraction

A phrase (in this study, a combination of two or more words) can commonly convey more semantic information than a single word. To better understand the actual meaning of emerging topics, we employ labeled data at the phrase or sentence level to interpret identified emerging topics.

We adopt the typical phrase extraction method pointwise mutual information (PMI) (https://en.wikipedia.org/wiki/Pointwise_mutual_information) based on the co-occurrence frequency to identify more meaningful combinations of words and link them together by underlining them. The PMI calculation formula is as follows:

$$\mathrm{PMI}(w_i, w_j) = \log_2 \frac{p(w_i w_j)}{p(w_i)\, p(w_j)}, \tag{3}$$

where $p(w_i w_j)$ represents the co-occurrence probability of w_i and w_j, and $p(w_i)$ represents the probability of w_i in the whole review collection. The value of PMI is proportional to the probability of the combination of w_i and w_j. By setting the threshold, we select the appropriate phrase as the candidate labeled data.

3.2 Emerging topic identification

In this subsection, we aim to identify emerging topics in the current app version by considering topics in previous versions. We adopt AOBTM (Hadi and Fard, 2020) to capture topic evolutions in different versions, and identify emerging topics from topic evolutions by the anomaly discovery method.

3.2.1 Topic evolutions

Whenever a new batch of text data is input, BTM requires retraining to capture potential topic distributions, which takes a lot of time. OBTM (Cheng et al., 2014) can fuse the features of topic distribution in the previous time slice to model the reviews in the current time slice. However, OBTM considers only the influence of the topic distribution in one time slice. In practice, developers might consider emerging topics from previous versions to ensure the effectiveness of version updates.

Similar to AOLDA, AOBTM allows users to customize the version window, that is, to control the number of time slices that affect the topic distribution characteristics in the current time slice. AOBTM can also control the influence weight of the distribution of topic features in historical time slices when calculating the topic distribution features of short texts in the current time slice. Different from AOLDA, AOBTM is more suitable for modeling of continuously input app reviews because the underlying model of AOBTM is BTM, which is more suitable for topic modeling of short texts. Fig. 5 shows the details of AOBTM.

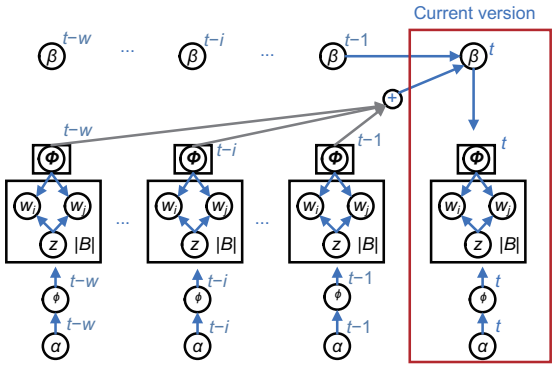

Fig. 5 Overview of the adaptive online biterm topic model (AOBTM)

The red rectangle highlights the probability distribution β_k^t in the current time slice t, which combines the topic distribution characteristics of the previous w time slices. References to color refer to the online version of this figure

In AOBTM, different versions of app reviews are expressed as $R = \{R^1, R^2, \ldots, R^t, \ldots\}$ (where t represents the app version). Enter reviews for each version in turn, and each review is treated as a separate corpus. We use α and β to represent the prior distribution of the corpus topics and topic words respectively, and α and β are defined initially. Set K to represent the number of topics. For the k^{th} topic, ϕ_k^t represents the probability of all input terms in time slice t. The parameter w defines the number of previous app versions that need to be considered when inferring the topic distribution of reviews in the current version. AOBTM adaptively integrates the distribution of topics from previous versions of w, denoted as $\{\phi^{t-1}, \ldots, \phi^{t-i}, \ldots, \phi^{t-w}\}$, for generating the prior probability distribution β^t of the t^{th} version. The adaptive integration is designed to summarize the topic distribution of different weights $\gamma^{t,i}$ of different versions. β^t is calculated as follows:

$$\beta_k^t = \sum_{i=1}^{w} \gamma_k^{t,i} \phi_k^{t-i} + n_{w|k}^t, \tag{4}$$

where i represents the i^{th} version before the current version t. $n_{w|k}^t$ represents the number of times that

w is assigned to topic k in version (time slice) t. The weight parameter $\gamma_k^{t,i}$ is determined by the similarity between the distributions of the k^{th} topic under the $(t-i)^{\text{th}}$ time slice. The calculation method is as follows:

$$\gamma_k^{t,i} = \frac{\exp\left(\phi_k^{t-i}\beta_k^{t-1}\right)}{\sum_{j=1}^{w}\exp\left(\phi_k^{t-j}\beta_k^{t-1}\right)}, \tag{5}$$

where the tensor dot product $\phi_k^{t-i}\beta_k^{t-1}$ calculates the similarity between the topic distribution ϕ_k^{t-i} and the prior probability distribution β_k^{t-1} of the $(t-1)^{\text{th}}$ version. This adaptive aggregation allows the topic distribution characteristics in the previous time slices to impact the topic distribution characteristics in the current time slice differently.

3.2.2 Outlier detection

The topic evolutions (i.e., β^t, β^{t-1}, $\cdots$, β^{t-i}) describe the topic distribution in different versions. There will be significant differences between the topic distributions in continuous versions when emerging topics exist. Therefore, we capture outliers (i.e., emerging topics) in topic evolution using an anomaly detection method.

First, we select classical Jensen–Shannon (JS) divergence (https://dit.readthedocs.io/en/latest/measures/divergences/jensen_shannon_divergence.html) to measure the difference in the k^{th} topic between two consecutive app versions (e.g., ϕ_k^t and ϕ_k^{t-1}). The JS divergence calculates the similarity between two probability distributions:

$$D_{\text{JS}}\left(\phi_k^t \| \phi_k^{t-1}\right) = \frac{1}{2}D_{\text{KL}}\left(\phi_k^t \| M\right) + \frac{1}{2}D_{\text{KL}}\left(\phi_k^{t-1} \| M\right), \tag{6}$$

where $M = \frac{1}{2}\left(\phi_k^t + \phi_k^{t-1}\right)$. The Kullback–Leibler (KL) divergence is employed to calculate discrimination between one probability distribution P and another probability distribution Q:

$$D_{\text{KL}}(P\|Q) = \sum_i P(i)\log_2\frac{P(i)}{Q(i)}, \tag{7}$$

where $P(i)$ is the i^{th} item in P. A higher JS divergence value represents a larger difference between the two distributions.

Second, we employ a typical outlier detection method (Rousseeuw and Hubert, 2011) to capture exceptional topics. The method assumes that the divergence obeys a Gaussian distribution with mean value μ and variance σ^2. The anomaly topic is then detected by setting a threshold δ. For version t of the app, threshold δ^t is defined dynamically in the following steps:

1. We calculate the previous w versions of D_{JS} for each topic and express it as a D_{JS} matrix of $w \times K$ (K is the number of topics).

2. We calculate the mean value μ and the variance σ^2 of all values in the D_{JS} matrix.

3. We set the threshold $\delta^t = \mu + 1.25\sigma$, and the coefficient 1.25 represents the acceptance of 10% of the topics as emerging topics.

For the t^{th} version, topics are regarded as emerging topics when the divergence D_{JS} value of topics exceeds the defined threshold δ^t.

3.3 Topic interpretation

Single words are not enough to understand the actual meaning of topics. For example, Table 1 shows the results of topics from a short text by BTM. We employ the five words with the highest probabilities under the first three topics to explain the topics. We observe that topics 1 and 2 use the same word "browser" to interpret topics. However, we cannot understand the actual semantic information of the topic "browser." It may be a browser crash or browser incompatibility. Furthermore, it is not clear whether the word "browser" in topics 1 and 2 contains the same semantic information.

Table 1 Output of the BTM

Topic 1	Topic 2	Topic 3
Browser	Compatibility	System
Exploit	Browser	Privacy
Web	Mac	Open
Cache	Versions	Connect
Html	Crash	Response

In this subsection, to better understand the actual meaning of emerging topics, we employ relevant phrases and sentences to interpret emerging topics.

3.3.1 Candidate labels

1. Phrases

In Section 3.1, we obtain the phrases extracted by PMI as the candidate labels. On this basis, we further constrain the phrase requirements: (1) the length of each word in the phrase should be no less than 4; (2) the phrases should contain at least one

noun or one verb, but no adverbs or determiners. The remaining phrases are regarded as our candidate phrase-level labels.

2. Sentences

We adopt the NLTK tool to segment the reviews and further filter the sentences as candidate sentence-level labels: (1) the sentence should include candidate phrase labels; (2) the length of the sentence must be no less than 5 words; (3) noisy data in sentences should be filtered out.

3.3.2 Similarity score

We employ the method in Mei et al. (2007) to calculate the similarity between the candidate label data (where the phrase is called a, the sentence is called s, and l is the label that is delegated to the phrase or sentence) and the emerging topic distribution ϕ_k^t. The similarity score calculation formula is as follows:

$$\text{Score}\left(l,\phi_k^t\right)=\text{Score}_{\text{sem}}\left(l,\phi_k^t\right)+\lambda\,\text{Score}_{\text{sen}}(l), \quad (8)$$

where $\text{Score}_{\text{sem}}\left(l,\phi_k^t\right)$ represents the semantic score, and the calculation method is as follows:

$$\text{Score}_{\text{sem}}\left(l,\phi_k^t\right)=\text{sim}\left(l,\phi_k^t\right)-\frac{\mu}{K-1}\sum_{j\neq k}\text{sim}\left(l,\phi_j^t\right). \quad (9)$$

Herein $\text{sim}\left(l,\phi_k^t\right)=-D_{\text{KL}}\left(l\|\phi_k^t\right)$, and it calculates the KL divergence between two probability distributions, mapped to calculate the vector similarity between label data l and topic distribution ϕ_k^t. In Eq. (9), the ideal situation is that a label has a high similarity with an emerging topic and a low similarity with other emerging topics. Therefore, the penalty term $\sum_{j\neq k}\text{sim}\left(l,\phi_j^t\right)$ is introduced to calculate the similarity between label l and other emerging topics other than the current emerging topic k. The coefficient $\frac{\mu}{K-1}$ represents the weight of the penalty item, where K is the number of topics. The higher the value of hyper-parameter μ, the lower the similarity score between candidate label l and the distribution of other emerging topics, indicating that l is weakly related to other emerging topics and strongly related to the current emerging topics.

$\text{Score}_{\text{sen}}(l)$ represents the sentiment score of label l. The star rating and text length of app reviews can reflect user' concerns: on one hand, a low star rating generally means that users are reporting issues for the app; on the other hand, long text reviews often provide more valuable information (Gao et al., 2018). $\text{Score}_{\text{sen}}(l)$ is calculated as follows:

$$\text{Score}_{\text{sen}}(l)=\exp\left(-\frac{r_l}{\log_2 h_l}\right), \quad (10)$$

where r_l and h_l represent the star rating and text length of the app review containing label l, respectively.

Based on $\text{Score}\left(l,\phi_k^t\right)$, the top three candidate phrases and sentences with the highest similarity scores are selected to interpret emerging topics.

4 Experiments

4.1 Dataset

We selected apps that meet the following four criteria: (1) these apps are popular ones in the app stores, which means that developers will constantly update them; (2) these apps come from different app categories and different platforms, to ensure the generalization of IETI; (3) each app has more than 5000 user reviews, which ensures the effective training of IETI; (4) most different versions of changelogs have detailed records to facilitate the evaluation of the model. We selected apps that meet these criteria in descending order from the rankings of different types of apps in the Apple App Store and Google Play. Overall, we obtained 16 4026 reviews from 89 app versions from six apps. Table 2 describes the dataset in more detail. All project codes run on the MacOS system, the programming language is Python3, and the CPU is M1.

Table 2 Subject apps

App name	Category	Platform	Number	Version
NOAA radar	Weather	Apple Store	8363	16
YouTube	Multimedia	Apple Store	37 718	33
Viber	Communication	Google Play	17 126	8
Clean master	Tool	Google Play	44 327	7
Ebay	Shopping	Google Play	35 483	9
SwiftKey	Productivity	Google Play	21 009	16

Number: the number of user reviews

4.2 Evaluation method

We employed the official app changelogs as the ground truth, and then manually extracted keywords from the changelogs as verification data. We adopted Word2Vec to compute the semantic similarity between label l (phrases and sentences) and the

keywords in the changelogs. We set a threshold to determine whether the emerging topic is covered in the changelogs, and we set the same similarity threshold 0.6. Then, we split each label into single words and calculated the similarity between each word and the keywords in the app changelogs. If the similarity score of a word is greater than the set threshold, we consider the label to be covered by the changelogs and mark it as a valid issue prediction.

In addition, we employed three performance metrics to evaluate the effectiveness of IETI. The first metric is employed to measure the accuracy of the emerging topics detected, and is defined as Precision. The second metric evaluates whether all topics detected by IETI (both emerging and non-emerging topics) are covered in the changelogs, and is defined as Recall. The third metric measures the balance between Precision and Recall, and is defined as F1 score.

$$\begin{cases} \text{Precision} = \dfrac{I(E \cap G)}{I(E)}, \\ \text{Recall} = \dfrac{I(L \cap G)}{I(G)}, \\ \text{F1 score} = 2 \times \dfrac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recal}}, \end{cases} \tag{11}$$

where E, G, and L represent the sets of detected emerging topics, keywords in changelogs, and all topics (including emerging and non-emerging ones) respectively, and $I(\cdot)$ counts the number elements in the set. During evaluation, we set the parameters as $w = 3$, $K = 10$, PMI=5, $\mu = 0.75$, and $\lambda = 0.75$, and initialized $\alpha = 0.1$ and $\beta = 0.01$.

4.3 Experimental results

Table 3 shows the evaluation results of emerging topic identification, and we analyzed the performance improvement of IETI in identifying emerging topics from the following three aspects:

1. For phrase labels, IETI achieved 0.628, 0.529, and 0.572 of Precision, Recall, and F1 score on average, respectively. Compared to IDEA, the average values of these three metrics of IETI were increased by 0.094, 0.107, and 0.126, respectively.

2. For sentence labels, IETI achieved 0.672, 0.628, and 0.647 of Precision, Recall, and F1 score on average, respectively. Compared to IDEA, the average values of these three metrics of IETI were increased by 0.068, 0.025, and 0.061, respectively.

3. Precision can accurately provide developers with the information on emerging topics, and Recall represents the ability of the model to identify emerging and common topics. These metrics help developers avoid missing some unexpected issues encountered by users. Precision and Recall collectively measure the ability of IETI to identify emerging topics, and IETI generally outperforms IDEA and OLDA in both metrics.

4.4 Analysis and discussion

4.4.1 Ablation study

We next investigated whether the improvement of the evaluation indexes created by IETI in Table 3 is attributable mainly to the unique preprocessing of the model, or AOBTM, or the combined effect of AOBTM and preprocessing. Therefore, we designed the following models for the ablation study:

1. IDEA$^+$: Replace the preprocessing of IDEA with the preprocessing of IETI, and the topic model is AOLDA.

2. IETI$^-$: Replace the preprocessing of IETI with the preprocessing of IDEA, and the topic model is AOBTM.

Then we applied IDEA, IDEA$^+$, IETI$^-$, and IETI to the same data sets. Table 4 shows the evaluation results of emerging topics identified by each model.

Compared with IDEA, Precision, Recall, and F1 score of IDEA$^+$ or IETI$^-$ were increased on both the phrase level and sentence level. Therefore, we confirmed that both the unique preprocessing and AOBTM of IETI improved the accuracy of emerging topic identification. We did not compare the contributions of the preprocessing and AOBTM to emerging topic identification, because we found that IDEA$^+$ and IETI$^-$ had some fluctuations in the comparison of evaluation indexes.

Compared with IDEA$^+$ and IETI$^-$, the Precision, Recall, and F1 score metrics of the IETI were increased on both the phrase level and sentence level. This shows that the improvement in evaluation indexes brought by IETI is attributable mainly to the combined effect of AOBTM and preprocessing.

4.4.2 Label diversity

Both the IETI and baselines adopt labels (phrases or sentences) to evaluate the effectiveness

Table 3 Comparison results of emerging topics identified by different models

App name	Method	Precision		Recall		F1 score	
		Phrase	Sentence	Phrase	Sentence	Phrase	Sentence
YouTube	OLDA	0.441	0.578	0.462	0.664	0.451	0.597
	IDEA	0.592	0.628	0.472	**0.666**	0.523	0.636
	IETI	**0.674**	**0.719**	**0.527**	0.625	**0.592**	**0.669**
Viber	OLDA	0.157	0.313	0.305	0.550	0.166	0.375
	IDEA	0.625	0.625	0.340	**0.651**	0.440	0.638
	IETI	**0.658**	**0.694**	**0.475**	0.642	**0.552**	**0.667**
SwiftKey	OLDA	0.100	0.367	0.567	0.617	0.148	0.458
	IDEA	0.517	0.583	**0.653**	**0.700**	0.523	0.587
	IETI	**0.565**	**0.622**	0.592	0.693	**0.578**	**0.656**
Clean master	OLDA	0.300	0.200	0.269	0.421	0.160	0.129
	IDEA	0.667	0.667	0.318	0.434	0.431	0.526
	IETI	**0.683**	**0.695**	**0.562**	**0.582**	**0.617**	**0.634**
Ebay	OLDA	0.167	0.500	0.238	0.488	0.196	0.494
	IDEA	0.229	0.646	0.251	0.527	0.227	0.580
	IETI	**0.574**	**0.693**	**0.434**	**0.572**	**0.494**	**0.627**
NOAA radar	OLDA	0.468	0.482	0.528	0.622	0.473	0.534
	IDEA	0.571	0.476	0.497	0.639	0.531	0.546
	IETI	**0.612**	**0.607**	**0.582**	**0.651**	**0.597**	**0.628**

Best results are in bold

Table 4 Ablation study results of different models

App name	Method	Precision		Recall		F1 score	
		Phrase	Sentence	Phrase	Sentence	Phrase	Sentence
YouTube	IDEA	0.592	0.628	0.472	**0.666**	0.523	0.636
	IDEA$^+$	0.621	0.684	0.481	0.634	0.542	0.658
	IETI$^-$	0.643	0.652	0.501	0.639	0.563	0.645
	IETI	**0.674**	**0.719**	**0.527**	0.625	**0.592**	**0.669**
Viber	IDEA	0.625	0.625	0.340	0.651	0.440	0.638
	IDEA$^+$	0.641	0.637	0.396	0.644	0.490	0.641
	IETI$^-$	0.637	0.652	0.425	**0.667**	0.501	0.659
	IETI	**0.658**	**0.694**	**0.475**	0.642	**0.552**	**0.667**
SwiftKey	IDEA	0.517	0.583	**0.653**	**0.700**	0.523	0.587
	IDEA$^+$	0.531	0.572	0.567	0.684	0.545	0.623
	IETI$^-$	0.557	0.617	0.574	0.677	0.566	0.646
	IETI	**0.565**	**0.622**	0.592	0.693	**0.578**	**0.656**
Clean master	IDEA	0.667	0.667	0.318	0.434	0.431	0.526
	IDEA$^+$	0.643	0.681	0.487	0.526	0.554	0.594
	IETI$^-$	0.671	0.680	0.523	0.547	0.588	0.601
	IETI	**0.683**	**0.695**	**0.562**	**0.582**	**0.617**	**0.634**
Ebay	IDEA	0.229	0.646	0.251	0.527	0.227	0.580
	IDEA$^+$	0.471	0.659	0.341	0.543	0.401	0.595
	IETI$^-$	0.525	0.668	0.382	0.552	0.442	0.604
	IETI	**0.574**	**0.693**	**0.434**	**0.572**	**0.494**	**0.627**
NOAA radar	IDEA	0.571	0.476	0.497	0.639	0.531	0.546
	IDEA$^+$	0.585	0.523	0.526	0.640	0.552	0.578
	IETI$^-$	0.571	0.574	0.554	0.642	0.562	0.606
	IETI	**0.612**	**0.607**	**0.582**	**0.651**	**0.597**	**0.628**

Best results are in bold

of emerging topics. However, the labels may be different, and label diversity is reflected in the following two aspects: on one hand, labels may be constructed differently and with different semantics; on the other hand, labels may be constructed differently but with similar semantic. Table 5 shows the emerging topics in NOAA radar. First of all, we need to declare that the order of labels of an emerging topic is random, which will change the order of emerging topics, but does not affect the evaluation results of emerging topics. In Table 5, we can observe that the phrase labels used by IETI were not completely the same as those used by IDEA. The label "lightning strike" appeared only in the experimental results of IDEA; although the phrases "waste money" and "pay money" were constructed differently, their semantics were similar.

Table 5 Emerging topic detection results of NOAA radar in version 1.7

IDEA	IETI
Topic 7	Topic 1
Lightning strike, 0.545	Extend forecast, 0.435
Extend forecast, 0.447	Loop speed, 0.419
Waste money, 0.353	Pay money, 0.365
Topic 9	Topic 8
Loop speed, 0.481	Extend forecast, 0.422
Extend forecast, 0.417	Loop speed, 0.412
Show nothing, 0.4123	Cloud cover, 0.385

The numbers represent the similarity scores of phrase labels

The difference between IETI and the baselines is embodied in the label differences. Our experimental results demonstrate that IETI adopts more accurate labels to interpret emerging topics or to reduce the number of falsely identified emerging topics, which means that IETI can obtain higher F1 score.

4.4.3 Labels

Based on the results in Table 3, we can have that the average Precision, Recall, and F1 score values used in IETI on these six apps were 0.628, 0.529, 0.572 respectively, for phrase labels, and 0.672, 0.628, 0.647 respectively, for sentence labels. Therefore, compared to phrase labels, adopting sentence labels to interpret the identified emerging topics can increase the average of Precision, Recall, and F1 score metrics by 0.044, 0.099, and 0.075, respectively. Furthermore, the superiority of adopting sentence-level labels includes obtaining more semantic details about identified emerging topics. For example, in NOAA radar's emerging topic identification results, emerging topics appear in version 3.1 and the label phrase with the highest probability is "weather apps." However, we cannot specifically understand the semantic information of "weather apps." The label sentence with the highest probability is "if Yahoo weather can then there be no reason others can not implement it as well," from which we can see that "weather apps" is likely to refer to "Yahoo weather," and the user response "Yahoo weather" is superior to "NOAA radar" in some app features.

We recommend employing labels at the sentence level to interpret the identified emerging topics. This approach can improve the accuracy of emerging topic recognition. More importantly, sentences can help developers better understand the practical significance of emerging topics.

4.4.4 Parameter influence

In this subsection, we demonstrate the impact of the two core parameters (the number of topics K and the window size w) on the performance of IETI. Other parameters in IETI have little influence on emerging topic identification, or are used only for algorithm initialization, which does not have practical guiding significance (Gao et al., 2018; Hadi and Fard, 2020).

1. Number of topics K

The topic model based on the Dirichlet hypothesis distribution demands that the topic number K is set in advance when modeling the text. Consequently, in benchmark models such as LDA and BTM, the results of topic modeling will be influenced by K. AOBTM is an improvement on BTM, and K likewise affects the results of emerging topic identification. Commonly, the larger the value of K, the more emerging topics are identified, and the higher the Precision. However, the number of misidentified emerging topics will also be increased, resulting in a lower Recall.

We attempt to understand the impact of K on the emerging topic identification results for six apps. Fig. 6 exhibits the effect of different numbers of topics with IETI on the F1 score. To compare IETI with IDEA under uniform conditions, we set K equal to 10. Applying different numbers of topics for six apps, we conjectured that a smaller number of topics seemed to yield higher F1 scores in minor scale text.

In major scale text, a larger number of topics can obtain a higher F1 score.

2. Window size w

When calculating the topic distribution of the current version, AOBTM will integrate some topic features of previous versions. Therefore, the number of previous versions will have a certain impact on the AOBTM calculation results. In this study, the number of previous versions is denoted as parameter window size w. Fig. 7 displays the impact of w on emerging topic recognition results. When we set different w values for six apps to identify emerging topics, we observed that the F1 score of the model was relatively high when w equaled 2 or 3. To maintain the consistency of the comparative experimental conditions, w was set to 3.

5 Threats to validity

1. Limitations of app changelogs

As mentioned, the emerging topics identified by IETI may not be covered in the app changelogs. However, because changelogs may not cover all the changes to the previous version of the app, these topics may represent unresolved issues in the updating procedures of multiple versions. It is difficult to formally define the practical meaning of these topics, but we can display these topics. Compared with reading each review, developers can reduce the time overhead by reviewing these topics.

2. Subjective emotions

We observed that some emerging topics are a result of the users' subjective emotions rather than specific software bugs or software features, such as the phrase label "waste money" in Table 5. These topics are rarely covered by app changelogs, which reduces the accuracy of emerging topic identification. However, these topics are valuable for app development, and analogously, we can also display these topics to developers.

3. Datasets and time slice

Our experimental data involved six types of apps, but we cannot predict whether our model is applicable to all types of apps. We mitigated this issue by selecting representative apps from different platforms and categories. In addition, fine-grained time

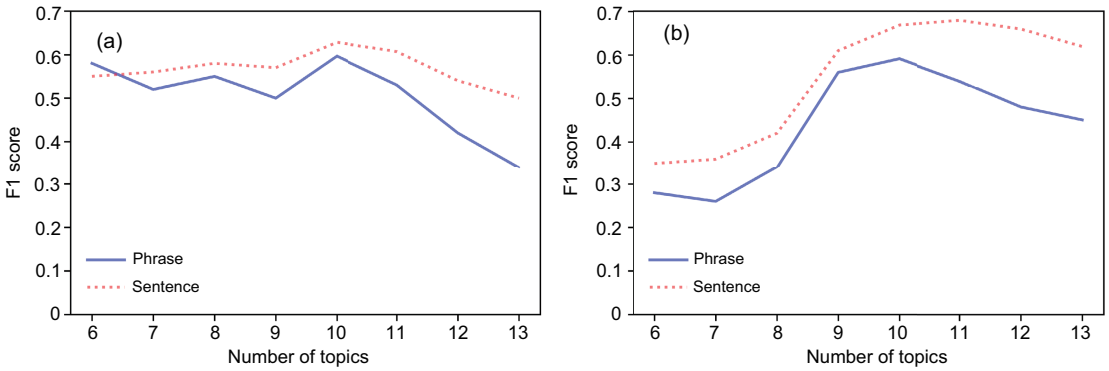

Fig. 6 Impact of the number of topics K on F1 score: (a) NOAA radar; (b) YouTube

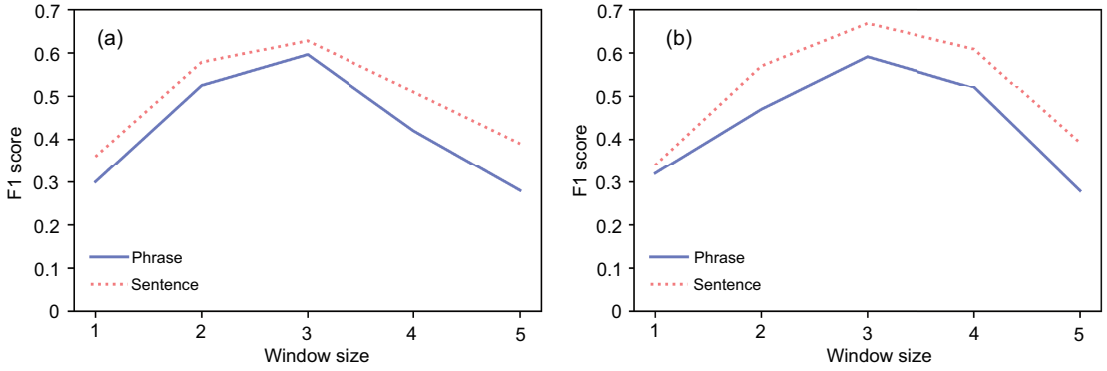

Fig. 7 Impact of the window size w on F1 score: (a) NOAA radar; (b) YouTube

slices can help the model quickly identify emerging topics when accepting continuous input. To alleviate this issue, we can divide time slices finely by treating them as new app versions.

6 Related works

This study applies the emerging topic identification method to app review analysis, so we divide the related works into two areas: app review analysis and emerging topic identification.

6.1 App review analysis

Because app reviews can provide valuable app related information from the perspective of immediate user experience, they are vital in app development. In recent years, more researchers have focused on automatically mining valuable information from app reviews (Liu YZ et al., 2019). The most familiar area of research in app review analysis is app review classification (Li YC et al., 2017; Jha and Mahmoud, 2019; Su et al., 2019). Some research has adopted machine learning techniques to classify app reviews and evaluate the effectiveness of models through manual tagging (Darbanibasmanj et al., 2019). Guzman et al. (2015) developed a set of machine learning classifiers that can automatically classify app user reviews into categories related to app development. The authors tested the effectiveness of the classifier on manually labeled data sets, and the final precision rate of the classifier reached 0.74 and the recall rate reached 0.59. Maalej and Nabil (2015) applied review metadata, text categorization, natural language processing, and sentiment analysis techniques to classify app review text into four categories: bug reports, feature requests, user experience, and ratings. Calefato et al. (2018) proposed a specially trained classifier called Senti4SD, which aims to classify original user reviews into reviews with various emotions. Aslam et al. (2020) adopted a convolutional neural network to classify app reviews, and the precision, recall, and F1 score of this method on a public data set were 0.95, 0.94, and 0.95, respectively.

In recent years, topic models have been widely used in app review analysis research. Some research concerned employing a topic model to model app reviews to complete tasks such as clustering or dimensionality reduction of app reviews. Park et al. (2015) proposed AppLDA, based on the LDA topic model, which models app reviews and app descriptions. The authors verified the effectiveness of AppLDA with more than one million reviews from 43 041 apps. Liu YD et al. (2016) proposed stratify app reviews (SAR), which is based on LDA. SAR classifies informative reviews into different levels and groups app reviews according to the context of users' attention. Wang et al. (2018) proposed group spamming latent Dirichlet allocation (GSLDA) to identify spam from reviews. Noei et al. (2021) adopted LDA to extract key topics from app reviews in different app categories.

6.2 Emerging topic identification

Some research has focused on identifying emerging topics from social media such as Twitter (Verasakulvong et al., 2018; Choi and Park, 2019). However, the characteristics of social media comments are different from those of app reviews. The main relevant research work for apps focuses on the changing trend of app topics over time and employs the traditional abnormal topic detection method to identify emerging topics (Gao et al., 2015). For example, Vu et al. (2016) proposed a phrase-based automatic method for extracting user opinions from app reviews, which adopts part-of-speech templates to extract phrases in reviews and monitors the outbreaks of phrase clusters with negative emotions over a period of time to extract user opinions. Zeng et al. (2018) employed a memory network for topic modeling and classification of short texts. The network adopts a novel topic memory mechanism to track the change in short text topics in various time slices. Fan and Ma (2014) summarized the application of LDA in the detection of emerging topics.

Identifying emerging topics in app reviews is a challenging task. IDEA was the first to apply online topic modeling methods to identify emerging topics from app reviews (Gao et al., 2018). In the framework of IDEA, the authors proposed a method called AOLDA to adaptively capture topic evolution in continuous app version reviews and identified emerging topics using anomaly detection. Based on the six most popular apps from the Apple App Store and Google Play, the evaluation scores of emerging topic identification by IDEA were as follows: the mean values of precision, recall, and F1 score were 0.604, 0.603, and 0.586, respectively.

7 Conclusions

Emerging topics in app reviews can guide developers in updating and maintaining their apps. In this study, we proposed IETI to identify emerging topics in continuous app versions. IETI reduced noisy data in reviews through richer preprocessing methods and applied AOBTM to emerging topic identification. We validated IETI's emerging topic identification effectiveness on six real-world apps. In the future, we will filter reviews related to app features or bugs to improve emerging topic identification accuracy and improve generalization of IETI by mixing reviews and fine-grained time slices.

Contributors

Wan ZHOU and Yong WANG designed the research. Wan ZHOU processed the data and drafted the paper. Yong WANG, Cuiyun GAO, and Fei YANG helped organize the paper. Wan ZHOU and Yong WANG revised and finalized the paper.

Compliance with ethics guidelines

Wan ZHOU, Yong WANG, Cuiyun GAO, and Fei YANG declare that they have no conflict of interest.

References

AlSumait L, Barbará D, Domeniconi C, 2008. On-line LDA: adaptive topic models for mining text streams with applications to topic detection and tracking. Proc 8th IEEE Int Conf on Data Mining, p.3-12. https://doi.org/10.1109/ICDM.2008.140

Aslam N, Ramay WY, Xia KW, et al., 2020. Convolutional neural network based classification of app reviews. *IEEE Access*, 8:185619-185628. https://doi.org/10.1109/ACCESS.2020.3029634

Blei DM, Ng AY, Jordan MI, 2003. Latent Dirichlet allocation. *J Mach Learn Res*, 3:993-1022.

Calefato F, Lanubile F, Maiorano F, et al., 2018. Sentiment polarity detection for software development. *Empir Softw Eng*, 23(3):1352-1382. https://doi.org/10.1007/s10664-017-9546-9

Chen N, Lin JL, Hoi SCH, et al., 2014. AR-miner: mining informative reviews for developers from mobile app marketplace. Proc 36th Int Conf on Software Engineering, p.767-778. https://doi.org/10.1145/2568225.2568263

Cheng XQ, Yan XH, Lan YY, et al., 2014. BTM: topic modeling over short texts. *IEEE Trans Knowl Data Eng*, 26(12):2928-2941. https://doi.org/10.1109/TKDE.2014.2313872

Choi HJ, Park CH, 2019. Emerging topic detection in Twitter stream based on high utility pattern mining. *Expert Syst Appl*, 115:27-36. https://doi.org/10.1016/j.eswa.2018.07.051

Darbanibasmanj AA, Persaud A, Ruhi U, 2019. Application of machine learning to mining customer reviews. Proc 25th Americas Conf on Information Systems, Article 21.

Devlin J, Chang MW, Lee K, et al., 2019. BERT: pre-training of deep bidirectional transformers for language understanding. Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.4171-4186. https://doi.org/10.18653/v1/N19-1423

Fan YM, Ma JX, 2014. Detection of emerging topics based on LDA and feature analysis of emerging topics. *J China Soc Sci Techn Inform*, 33(7):698-711 (in Chinese). https://doi.org/10.3772/j.issn.1000-0135.2014.07.003

Gao CY, Xu H, Hu JJ, et al., 2015. AR-Tracker: track the dynamics of mobile apps via user review mining. IEEE Symp on Service-Oriented System Engineering, p.284-290. https://doi.org/10.1109/SOSE.2015.13

Gao CY, Zeng JC, Lyu MR, et al., 2018. Online app review analysis for identifying emerging issues. Proc 40th Int Conf on Software Engineering, p.48-58. https://doi.org/10.1145/3180155.3180218

Gao CY, Zheng W, Deng Y, et al., 2019. Emerging app issue identification from user feedback: experience on WeChat. Proc 41st Int Conf on Software Engineering: Software Engineering in Practice, p.279-288. https://doi.org/10.1109/ICSE-SEIP.2019.00040

Genc-Nayebi N, Abran A, 2017. A systematic literature review: opinion mining studies from mobile app store user reviews. *J Syst Softw*, 125:207-219. https://doi.org/10.1016/j.jss.2016.11.027

Gu XD, Kim S, 2015. "What parts of your apps are loved by users?[1]". Proc 30th IEEE/ACM Int Conf on Automated Software Engineering, p.760-770. https://doi.org/10.1109/ASE.2015.57

Guzman E, El-Haliby M, Bruegge B, 2015. Ensemble methods for app review classification: an approach for software evolution. Proc 30th IEEE/ACM Int Conf on Automated Software Engineering, p.771-776. https://doi.org/10.1109/ASE.2015.88

Hadi MA, Fard FH, 2020. AOBTM: adaptive online biterm topic modeling for version sensitive short-texts analysis. IEEE Int Conf on Software Maintenance and Evolution, p.593-604. https://doi.org/10.1109/ICSME46990.2020.00062

Huang JJ, Peng M, Wang H, et al., 2017. A probabilistic method for emerging topic tracking in microblog stream. *World Wide Web*, 20(2):325-350. https://doi.org/10.1007/s11280-016-0390-4

Jha N, Mahmoud A, 2019. Mining non-functional requirements from app store reviews. *Empir Softw Eng*, 24(6):3659-3695. https://doi.org/10.1007/s10664-019-09716-7

Jin MM, Luo X, Zhu HL, et al., 2018. Combining deep learning and topic modeling for review understanding in context-aware recommendation. Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.1605-1614. https://doi.org/10.18653/v1/N18-1145

Li CL, Duan Y, Wang HR, et al., 2017. Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Trans Inform Syst*, 36(2):11. https://doi.org/10.1145/3091108

Li YC, Jia BX, Guo Y, et al., 2017. Mining user reviews for mobile app comparisons. *Proc ACM Interact Mob Wear Ubiquit Technol*, 1(3):75. https://doi.org/10.1145/3130935

Liu YD, Li YW, Guo YH, et al., 2016. Stratify mobile app reviews: E-LDA model based on hot "entity" discovery. Proc 12th Int Conf on Signal-Image Technology & Internet-Based Systems, p.581-588. https://doi.org/10.1109/SITIS.2016.97

Liu YZ, Liu L, Liu HX, et al., 2019. App store mining for iterative domain analysis: combine app descriptions with user reviews. *Softw Pract Exp*, 49(6):1013-1040. https://doi.org/10.1002/spe.2693

Maalej W, Nabil H, 2015. Bug report, feature request, or simply praise? On automatically classifying app reviews. Proc 23rd IEEE Int Requirements Engineering Conf, p.116-125. https://doi.org/10.1109/RE.2015.7320414

McIlroy S, Ali N, Hassan AE, 2016. Fresh apps: an empirical study of frequently-updated mobile apps in the Google Play Store. *Empir Softw Eng*, 21(3):1346-1370. https://doi.org/10.1007/s10664-015-9388-2

Mei QZ, Shen XH, Zhai CX, 2007. Automatic labeling of multinomial topic models. Proc 13th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.490-499. https://doi.org/10.1145/1281192.1281246

Nguyen TS, Lauw HW, Tsaparas P, 2015. Review synthesis for micro-review summarization. Proc 8th ACM Int Conf on Web Search and Data Mining, p.169-178. https://doi.org/10.1145/2684822.2685321

Noei E, Zhang F, Zou Y, 2021. Too many user-reviews! What should app developers look at first? *IEEE Trans Softw Eng*, 47(2):367-378. https://doi.org/10.1109/TSE.2019.2893171

Park DH, Liu MW, Zhai CX, et al., 2015. Leveraging user reviews to improve accuracy for mobile app retrieval. Proc 38th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.533-542. https://doi.org/10.1145/2766462.2767759

Rousseeuw PJ, Hubert M, 2011. Robust statistics for outlier detection. *WIREs Data Min Knowl Discov*, 1(1):73-79. https://doi.org/10.1002/widm.2

Sarro F, Al-Subaihin AA, Harman M, et al., 2015. Feature lifecycles as they spread, migrate, remain, and die in App Stores. Proc 23rd IEEE Int Requirements Engineering Conf, p.76-85. https://doi.org/10.1109/RE.2015.7320410

Su YQ, Wang YC, Yang WH, 2019. Mining and comparing user reviews across similar mobile apps. Proc 15th Int Conf on Mobile Ad-Hoc and Sensor Networks, p.338-342. https://doi.org/10.1109/MSN48538.2019.00070

Verasakulvong E, Vateekul P, Piyatumrong A, et al., 2018. Online emerging topic detection on Twitter using random forest with stock indicator features. Proc 15th Int Joint Conf on Computer Science and Software Engineering, p.1-6. https://doi.org/10.1109/JCSSE.2018.8457349

Vu PM, Pham HV, Nguyen TT, et al., 2016. Phrase-based extraction of user opinions in mobile app reviews. Proc 31st IEEE/ACM Int Conf on Automated Software Engineering, p.726-731. https://doi.org/10.1145/2970276.2970365

Wang Z, Gu SM, Xu XW, 2018. GSLDA: LDA-based group spamming detection in product reviews. *Appl Intell*, 48(9):3094-3107. https://doi.org/10.1007/s10489-018-1142-1

Zeng JC, Li J, Song Y, et al., 2018. Topic memory networks for short text classification. Conf on Empirical Methods in Natural Language Processing, p.3120-3131. https://doi.org/10.18653/v1/D18-1351