Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6668344, 12 pages
https://doi.org/10.1155/2022/6668344

Research Article

A BTM-Based Adaptive Objectionable Short Text Filtering Framework

Dong Cui[1], **Qiaoyan Wen**[1], **Hua Zhang**[1], **Wenmin Li**[1], **Yijie Shi**[1], **Yingyu Zhou**[1] and **Lei Zhang**[2]

[1]*The State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Information Technology Department, Minsheng Bank, Beijing, China 101300*

Correspondence should be addressed to Hua Zhang; zhanghua_288@bupt.edu.cn and Wenmin Li; liwenmin02@outlook.com

Received 1 December 2020; Revised 20 October 2021; Accepted 3 December 2021; Published 22 January 2022

Academic Editor: Javier Prieto

Many methods are available for objectionable text filtering, such as URL-based filtering, keyword-based filtering, and intelligence-based analysis filtering approaches. URL-based filtering cannot filter the contents of objectionable short text. Keyword-based filtering faces the overblocking issue. Intelligence-based analysis filtering is inefficient and ineffective when filtering objectionable short text. In this paper, a biterm topic modelling- (BTM-) based adaptive objectionable short text filtering framework is proposed. We propose a feature extraction algorithm for objectionable short text and establish a sensitive word feature dataset using the descriptions of applications on the Internet. Then, we construct a judgment standard to automatically select the K value of the BTM topic model that can induce self-adaptation. The feature dataset constructed in this paper can effectively reflect the characteristics of objectionable short text. The proposed filtering framework can effectively identify objectionable short text and has a higher filtering rate than other approaches.

1. Introduction

Objectionable short text contains information such as pornography, violence, and murder. Moreover, such texts are abundant in web applications and mobile apps. This information is harmful to the health and growth of teenagers. Since the passage of the Communications Decency Act (CDA) [1] in 1995, a consortium including the Microsoft Corporation, Netscape Communications, and Progressive Networks has established standards that empower parents to block inappropriate web content. Some countries and international organisations, such as the United Kingdom, the European Commission, the Netherlands, China, and the Family Online Safety Institute (FOSI, have performed many similar actions to protect children and the general public from harmful content. An important part of the FOSI's work is the classification of applications and user-generated content.

There are currently four main types of research on objectionable text filtering: studies based on uniform resource locator (URL) filtering, keyword-based filtering, intelligence-based filtering, and topic-based filtering.

URL-based filtering methods [2–4] provide excellent manual control modes to filter unwanted websites with the correct metadata settings. The URL, as a filter unit, filters out standard text.

Keyword-based filtering [5–8] applies to fine-grained text filtering especially at the word level and such a method requires a list of keywords. A keyword filtering method easily misses objectionable text [8]. If filtering is performed by augmenting sensitive words, overblocking problems occur and the accuracy is significantly reduced [9].

Intelligence-based analysis filtering [10, 11] has difficulty fine-tuning sentences or paragraphs because such methods use web pages as the basic units to infer a filtering model, and the detection performance depends mostly on the quality of the given training set [12].

Topic-based filtering [13–15] enables the filtering of sentences or paragraphs by analysing the possible topics

contained in the input text. Traditional topic models, such as latent Dirichlet allocation (LDA), do not work well for shot text because these models implicitly capture document-level cooccurrence patterns to reveal text themes; thus, such approaches encounter data sparsity problems in short texts [16].

In summary, the four above types of filtering methods are not suitable for the filtering of objectionable short text for these reasons. To better solve the data sparsity problem of objectionable short text filtering and improve the accuracy of filtering, this paper proposes a new filtering framework.

1.1. Contributions. To solve the problem of semantically filtering objectionable short text on the Internet and to realize automatic and rapid filtering, this paper proposes a biterm topic modelling- (BTM-) based adaptive objectionable short text filtering framework. The main contributions of this paper are as follows:

An extraction method for sets of text features is proposed based on the term frequency-inverse document frequency (TF-IDF) algorithm. The extraction method divides the observed text features into general feature words and sensitive feature words and constructs a word weight table based on word frequency and word cooccurrence relationships.

An adaptive topic model is proposed based on a BTM. By using the topic model to capture the potential relationships between words, a filtering model for objectionable short text is assembled. Unlike traditional BTMs that do not consider choosing the number of topics, we propose a method to evaluate the number of topics in the BTM to achieve model adaptation.

An experimental model efficiency verification is provided. By collecting the short text information of objectionable applications in mobile application stores, such as wood ant, application treasure, and the Baidu mobile assistant application store, we establish a feature dataset, extract feature words, and train the developed model. The obtained results show that the combination of objectionable text feature selection and BTM-based adaptive topic model construction is superior to previously proposed methods in terms of detecting objectionable short text.

We produce an objectionable short text dataset and a standard short text dataset.

2. Related Work

To effectively filter objectionable text, researchers have done much work, which mainly includes studies on URL-based filtering, keyword-based filtering, intelligence-based analysis filtering, and topic-based filtering. URL-based filtering cannot filter objectionable short text, so we do not discuss it here. Furthermore, at the end of this section, we introduce the BTM-based topic model used in this paper.

2.1. Keyword-Based Filtering. Keyword-based filtering [5–7] methods find suspicious objectionable text based on the words appearing in the input text content. Each word is compared to a word in a keyword dictionary which consists of disallowed words and phrases. Once the number of matches reaches a predefined threshold, the text is determined to be objectionable text. The resource consumption of this method is low so it is widely used to filter web content. However, this method is prone to a well-known "overblocking" phenomenon. Therefore, a method based on intelligent analysis filtering is later used to solve this problem.

2.2. Intelligence-Based Analysis Filtering. Filtering objectionable text can be considered a two-category problem dividing the input text into standard text and objectionable text. K-nearest neighbors (KNN) algorithms, logistic regression algorithms, neural networks, naive Bayes classifiers, and support vector machines (SVMs) are all used in this case.

An SVM-based classifier can classify a web page as a node in a sensitive content category. Du et al. [17] used a classifier obtained by SVM training to filter sensitive text on the web and tested their approach on an adult text dataset collected from Yahoo's adult category pages. Jin et al. [18] used a text pattern similar to the concept of a "regular expression" in the Perl language as one of their model features. A classifier typically also uses several additional functions to improve its classification accuracy. For example, URLs, stitching features, and some structural features can be integrated into a learning algorithm to improve its classification performance [19, 20]. Later, Lee et al. [21] proposed a new detection framework to find target web pages by mining user search behaviours. The intelligent aspect of this framework involves utilizing a user intent model to learn how to capture new and objectionable web content. Ali et al. [22] used a supervised learning method with a naive Bayes classification algorithm and an SVM and found that the best model for detecting pornographic content on Twitter is an SVM with unigram and bigram combinations using the TF-IDF algorithm and the most common words; this approach achieved an F1-score of 91.14%.

2.3. Topic-Based Filtering. Filtering frameworks based on the topic models [9, 12, 13, 16] exhibit improved objectionable text detection ability by analysing the semantic content of the input text. To achieve fine-grained detection, such a framework uses sentences as the basic units for identification and filtering; that is, when filtering web content, each sentence is detected and judged to determine whether it constitutes objectionable text. A topic model analyses the semantics of the given sentence, and by building an internal semantic space and calculating relevant probabilities, the sensitivity of the sentence is measured. To avoid the use of complex model parameters, Blei et al. [13] used a potential Dirichlet distribution (via LDA). This distribution provides a better generation mechanism than other approaches.

2.4. BTM-Based Topic Models. Finding hidden topics from short texts such as tweets or instant messages has become an essential task for many content analysis applications. However, applying traditional topic models (such as LDA and probabilistic latent semantic analysis (PLSA)) directly to such short texts does not achieve the desired results.

The underlying reason for this is that traditional topic models implicitly capture document-level cooccurrence patterns to reveal topics, thus encountering the problem of severe data sparsity in short documents.

To solve the problems encountered by the traditional thematic model, Yan et al. proposed a BTM, which utilizes a disordered cooccurrence word pair in a short text window [16] to learn topics by directly modelling the generation of word cooccurrence patterns in the whole input corpus. A BTM explicitly models word cooccurrence patterns to enhance the topic of learning and uses the aggregated patterns in the whole corpus to learn topics and solve the problem of word cooccurrence pattern scarcity at the document level.

A BTM is a topic model that is ideal for short text scenes. Different from the traditional LDA-based topic model, a BTM not only maintains the correlations between words but can also infer the topic probability of a document, and compared with the individual LDA approach, it can better reveal the topics in text. A BTM uses biterms, which consist of two words each, to enhance the learning of the topic model. It uses the entire input corpus to sample topics and infers the global distribution of topics across the corpus. In a BTM, the given document is considered a random mix of possible topics. Each topic is thought to be a probability distribution. In a BTM, the *biterm-topic* distribution, the *document-biterm* distribution, and the *document-topic* distribution in the corpus can be represented by Formulas (1), (2), and (3), respectively [16]. The word vector space of a short text can be mapped to the topic vector space of the short text.

However, the traditional BTM does not provide a way to determine the number of topics K, which can only be resolved with specific data. Although the value of K can sometimes be estimated, this method does not meet the requirements of adaptively detecting objectionable text and is not suitable for this purpose. Moreover, traditional BTMs cannot be used to directly filter objectionable short texts.

$$P(z \mid b) = \frac{p(z)p(w_i \mid z)p(w_j \mid z)}{\sum_z p(z)p(w_i \mid z)p(w_j \mid z)}, \tag{1}$$

$$P(b \mid d) = \frac{n_d(b)}{\sum_b n_d(b)}, \tag{2}$$

$$P(z \mid d) = \sum_b P(z \mid b)P(b \mid d). \tag{3}$$

3. BTM-Based Filtering Framework for Objectionable Short Text

This paper proposes a new filtering framework with an adaptive topic model based on a BTM. We build a sensitive feature dataset by quantifying the importance and sensitivity levels of feature words. We build an adaptive BTM detection model (aBTMd) based on a sensitivity feature dataset. The BTM-based adaptive filtering framework can adapt to different datasets without requiring the K value of the topic model to be set in advance.

The objectionable short text filtering framework proposed in this paper is shown in Figure 1, and it includes two parts: a training phase and a detection phase. The training phase is divided into three steps: extracting feature words from the corpus (step I), calculating the weight table (step II), and building an adaptive topic model (step III). The detection phase is divided into two steps: converting the target short text to a biterm (step IV) and conducting detection with the aBTMd (step V).

(i) First, we extract general feature words from corpus D to obtain $W_{gf} = \{w_1, w_2, \cdots, w_n\}$. Then, we extract words from the list of sensitive feature words S to obtain $W_{sf} = \{w_1, w_2, \cdots, w_n\}$. Finally, we can obtain the biterm list through W_{gf} and W_{sf}

(ii) To obtain the feature weight table, we need to first calculate the Imp and Obj of each feature word and then combine the seed word list S and biterm list to perform calculations through an algorithm

(iii) In this step, we build an aBTMd through the adaptive algorithm proposed in this paper

(iv) To detect the target short text, we need to convert it into biterms; this process includes the extraction of feature words and the calculation of biterms to obtain $Tb = \{b_1, b_2, \cdots, b_n\}$

(v) We input Tb into the trained aBTMd for detection to identify whether the text is objectionable short text through the score obtained by the detector

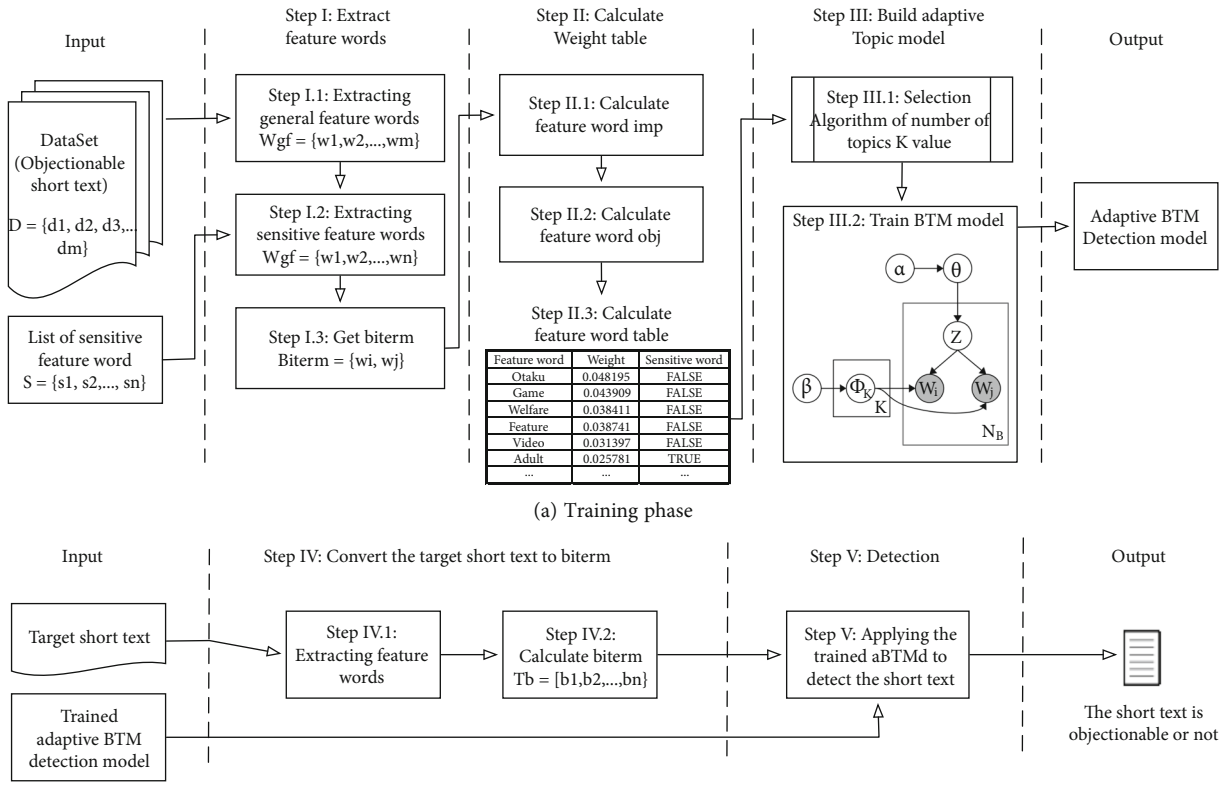
The new method does not require a complete keyword list, which makes the proposed approach different from traditional keyword-based filtering methods. By setting the seed words in advance, based on calculation, the framework can obtain an objectionable feature dataset, which can effectively reflect the sensitivity of the corpus. Different from current classification-based detection methods, we use the aBTMd to build a one-class classification model for detecting objectionable short text. This approach is helpful for constructing an adaptive topic model with a combination of objectionable feature selection abilities.

Detailed information about the proposed framework will be introduced in subsequent sections. Table 1 shows some symbols used in the following sections.

4. Building a Sensitive Word Dataset

This paper proposes an improved method for constructing a feature dataset by referencing feature word frequencies and word cooccurrence relationships based on the feature word extraction method proposed by Zeng et al. [9].

Because feature words have different effects on objectionable short text detection depending on their sensitivities, it is necessary to set different weights. This paper classifies sensitive words as general feature words and sensitive feature words. General feature words refer to relatively concealed words that express objectionable intentions. A sensitive



(a) Training phase

(b) Detection phase

Figure 1: Overview of the BTM-based adaptive objectionable short text filtering framework: the training phase generates an adaptive BTM-based detection model (aBTMd), and the detection phase uses the aBTMd to determine whether a target short text is objectionable.

Table 1: Symbol description.

Symbol	Description
$D = \{d_1, d_2, \cdots, d_m\}$	Corpus
$W = \{w_1, w_2, \cdots, w_n\}$	List of feature words
$\mathrm{OC}(w_i, w_j)$	The number of simultaneous occurrences of each pair of words in each document in corpus D
M	Feature word relationship matrix
n	Number of feature words in the feature vocabulary
d	Damping factor between 0 and 1; usually 0.85
$\mathrm{Imp}(i)$	Word importance vector; each element represents the importance of the corresponding feature word w_i in the sequence (recorded as $\mathrm{imp}(i)$)
$\mathrm{Obj}(i)$	The sensitivity of a feature word w_i
$Wg(i)$	The weight value of word w_i
$S = \{s_1, s_2, \cdots, s_n\}$	List of sensitive feature words
$\mathrm{SW} = \{\mathrm{sw}_1, \mathrm{sw}_2, \cdots, \mathrm{sw}_n\}$	Weight list of sensitive feature words
d_i	Corpus document i
Z	Topic implied in the document
$\mathrm{biterm}=\{w_i, w_j\}$	Word pairs consisting of word i and word j
n_z	Number of word pairs under topic Z
$n_{w_i z}$	Number of times topic Z is assigned to word w_i
STC	Sensitive topic criterion
$\mathrm{STCV}(t)$	The sensitivity of topic t
K	The topic number

Input: $W = \{w_1, w_2, \cdots, w_n\}$, $D = \{d_1, d_2, \cdots, d_m\}$.
Output: $Imp(w)$
1: init matrix $Imp(w) = [(1/n), (1/n), \cdots, (1/n)]$
2: **for** i in $\{1, 2, \cdots, n-1\}$ **do**
3: **for** j in $\{i+1, i+2, \cdots, n\}$ **do**
4: **for** k in $\{1, 2, \cdots, k\}$ **do**
5: **if** $(w_i \in d_k) \wedge (w_j \in d_k)$ **then**
6: $OC(w_i, w_j)$ ++;
7: **end if**
8: **end for**
9: **end for**
10: **end for**
11: Calculated with the following formula:

$$M(i,j) = OC(w_i, w_j) / \sum_{j=1, j \neq i}^{n} OC(w_i, w_j)$$

12: **while** $Imp(w)_p - Imp(w)_q > \Delta$ **do**
13:

$$Imp(w)_p = dM(i,j) Imp(w)_q + (1-d) Imp(w)_q$$

14: **end while**
15: **return** $Imp(w)$

Algorithm 1: Feature word **Imp** calculation algorithm.

feature word is a word with a clear indication of objectionable information that is determined manually in advance.

The weight of a feature word is determined by two aspects. One is the frequency at which the feature word appears in the corpus, and the other is the sensitivity of the feature word itself. To better distinguish the weights of general feature words and sensitive feature words during filtering, based on the existing research, this paper divides the weight of each feature word into an importance degree (Imp) and objectionable degree (Obj). Imp refers to the importance of the feature word in the corpus. Obj refers to the degree of the feature word as a basis for objectionable text judgment.

First, the feature matrix consists of the relationships between feature words, and then, the importance levels of feature words in the training set are calculated by the PageRank algorithm. To better distinguish between the sensitivities of different sensitive feature words and general feature words, this paper utilizes the TF-IDF algorithm and introduces statistical word frequencies to the training set, and the sensitivities are divided into different levels for calculation purposes. Finally, the weight Wg of the corresponding feature word is obtained by multiplying its Imp and Obj values.

4.1. Calculation of the Imp of a Feature Word. **Imp** refers to the importance of a feature word in the input corpus. To obtain the Imp of a feature word in the corpus, this paper uses the PageRank algorithm for calculation. Algorithm 1 shows how to calculate this value. Firstly, we need to create an undirected graph. The nodes of the graph are feature words, and the edges of two nodes describe the word cooccurrence relationship between the two feature words. Then, the undirected graph is transformed into a matrix by using the PageRank random browsing model. Finally, we calculate the convergence matrix after the graph matrix reaches a steady state.

4.2. Calculation of the Obj of a Feature Word. **Obj** refers to the importance of feature words while judging objectionable text. The obj of feature words is divided into the obj of sensitive feature words and the obj of general feature words.

Sensitive feature words have different levels of semantics, especially in Chinese. Different sensitive feature words have different tones, and their occurrence frequencies in the training set are also different. Words with high sensitivity and low frequency have good text discrimination. For those words with a high frequency of occurrence, it is clear that they do not have such good text discrimination as the former.

Therefore, to better reflect the weights of feature words in the corpus and better distinguish objectionable text, this paper introduces the TF-IDF algorithm to calculate the obj values of feature words in the training set according to the frequencies of sensitive feature words appearing in the corpus.

Algorithm 2 shows how to calculate the obj values of feature words. Based on the idea of the TF-IDF algorithm, if the TF of a sensitive feature word appearing in a document is very high but the feature word does not appear in other documents, then the word can distinguish this document from other documents very well. If the number of documents containing sensitive feature words is smaller than the number of other documents, the IDF value is higher, and the ability to differentiate between sensitive feature words is stronger. By calculating the TF-IDF value of each sensitive feature word in the corpus, the degree of the model to distinguish objectionable text for each feature word can be measured. On this basis, the obj values of the features of sensitive words are set in the interval of (0.5, 1).

The obj calculations of general feature words depend on the cooccurrence relationships between the sensitive feature words and general feature words in the given corpus. Based on the $OC(w_i, w_j)$ calculated in section A, which is the

Input: $W = \{w_1, w_2, \cdots, w_n\}$, $S = \{s_1, s_2, \cdots, s_m\}$, $OC(w_i, w_j)$, $C = \{c_{obj}, c_1, c_2, \cdots, c_p\}$.
Output: $Obj(w)$
1: **for** i in $\{1, 2, \cdots, n\}$ **do**
2: **if** $w_i \in S$ **then**
3: $calculate\, tf - idf_i$ of w_i in C
4: $s_w(i) = tf - idf_i/2 + 0.5$
5: **else**

$$p_w(i) = \sum_{j=1, j\in S}^{n} s_w(j)\left(OC(w_i, w_j)/\sum_{j=1, j\neq i}^{n} OC(w_i, w_j)\right)$$

6: **end if**
7: **end for**
8: calculate $Obj(i)$:

$$Obj(i) = \begin{cases} s_w(i) & \text{if } w_i \text{ in } S \\ p_w(i) & \text{if } w_i \text{ not in } S \end{cases}$$

9: **return** $Obj(w)$

Algorithm 2: Feature word **Obj** calculation algorithm.

Input: corpus D
Output: K
1: Set the maximum K value: $K = K_{\max}$.
2: Train the BTM model using corpus D; generate ***biterm*** – ***topic*** distribution, ***document*** – ***biterm*** distribution, etc.
3: Calculate $STC(K)$ using Formula (7).
4: $K = K$-1.
5: Repeat steps (2)-(5) until K=2.
6: Select the K value that makes $STC(K)$ reach its peak value as the final result.
7: **return** K

Algorithm 3: Adaptive BTM algorithm.

number of times word w_i and word w_j appear in the corpus document simultaneously, the sensitivity of the general feature word can be calculated by Formula (4). S represents the list of sensitive feature words, and $p_w(i)$ represents the calculated sensitivity of the ith feature word.

$$p_w(i) = \sum_{j=1, j\in S}^{n} s_w(j) \frac{\mathrm{OC}(w_i, w_j)}{\sum_{j=1, j\neq i}^{n} \mathrm{OC}(w_i, w_j)}. \tag{4}$$

Finally, this paper obtains the sensitivity of each feature word (Obj) through

$$Obj(i) = \begin{cases} s_w(i) & \text{if } w_i \text{ in } S. \\ p_w(i) & \text{if } w_i \text{ not in } S. \end{cases} \tag{5}$$

4.3. Sensitive Word Feature Dataset. After calculating the Imp and Obj of each feature word, the weight of every feature word is obtained by multiplying the two values. The weight of a feature word can be obtained by

$$Wg(i) = \mathrm{Imp}(i) * \mathrm{Obj}(i). \tag{6}$$

By sorting the weights of the feature words, the first T nodes are selected as the feature words for model training.

5. Adaptive BTM-Based Topic Model

In this paper, we present an adaptive BTM topic model for quickly and automatically detecting objectionable short text.

5.1. Establishment of the Adaptive BTM Topic Model. A BTM does not provide a way to determine the value of K. Based on the idea regarding the sensitivity of objectionable short text, this section proposes an adaptive modelling method based on a BTM to determine the optimal value of K. This paper designs a standard to measure whether a topic is sensitive and defines it as a sensitive topic criterion (STC). Algorithm 3 shows the process of building the adaptive BTM.

First, we need to derive the weight of each biterm. As shown in Formula (7), biterm_i is composed of the word pair (w_p, w_q). The weight of biterm_i is the product of the weights of words w_p and w_q. We use $b(i)$ to represent the weight value of biterm_i and $wg(i)$ to represent the weight value of word w_i.

$$b(i) = \mathrm{w}g(p) * wg(q). \tag{7}$$

$\mathrm{STCV}(t)$ indicates the sensitivity of topic t, which is calculated by the sensitivity of biterm_i and the probability of biterm_i under topic t. The calculation is as shown in

Formula (8). T represents the number of feature words, and $p(b_i \mid z_t)$ represents the probability of biterm$_i$ under sensitive topic t.

$$\mathrm{STCV}(t) = \sum_{i=1}^{T} b(i)p(b_i \mid z_t). \tag{8}$$

Finally, the calculation process for obtaining $\mathrm{STC}(K)$ is shown in Formula (9). $\mathrm{STC}(K)$ can reflect the sensitivity of the topic model obtained through BTM training when the number of topics is set as K. If those word group terms, which have high sensitivity levels, also have substantial probabilities when the topic has been given, then the model tends to reflect the sensitivities of the topics very well. Therefore, the peak value of $\mathrm{STC}(K)$ reflects the optimal parameter value K of the BTM that is suitable for the training text corpus, and the value of K is obtained by using

$$\mathrm{STC}(K) = \sum_{i=1}^{K} \mathrm{STCV}(i), \tag{9}$$

$$K = \underset{k}{\operatorname{argmin}} \mathrm{STC}(K). \tag{10}$$

5.2. Objectionable Short Text Filtering. This paper judges an unknown text by calculating its similarity to the adaptive topic model generated above. If the similarity is more significant than a particular value, it can be judged as a black sample. Otherwise, it is determined as a white sample. In the BTM, Formula (11) can be used as a basis for determining whether the input sample is an objectionable text.

$$p(d \mid M) = \int p(\theta \mid \alpha) \left(\prod_{j=1}^{K} \sum_{z_j} p(z_j \mid \theta) p(b \mid z_j, \beta) \right) d\theta. \tag{11}$$

The final result $p(d \mid M)$ represents the probability that the unknown text is generated by the topic model.

When $p(d \mid M)$ is more significant than a manually set threshold r, this indicates that the sample has a high degree of similarity to the topic model and can be judged as objectionable text.

5.3. Example. In order to better understand the proposed BTM-based adaptive objectionable short text filtering framework in this paper, we take a text dataset D containing pornography as an example to illustrate. As the first step, we extract all the feature words in dataset D, filter the words with obvious pornographic meanings as a list of sensitive feature words W_{sf}, and the rest as a list of general feature words W_{gf}. We can obtain the biterm list through W_{gf} and W_{sf}. In the second step, we can calculate the obj and imp of each feature word by the method proposed in this paper and then calculate the Wg of each word. We can get the feature weight table as shown in Table 2. In the third step, we use the feature weight table to determine the K values of the BTM model and construct aBTMd. Finally, we can use aBTMd to detect whether the unknown text is pornography.

Table 2: Feature weight table.

S/N	Feature word	Weight	Sensitive word
1	Otaku	0.048195	False
2	Game	0.043909	False
3	Welfare	0.038411	False
4	Feature	0.034741	False
5	Video	0.031397	False
6	Adult	0.025781	True
7	Beauty	0.019537	False
8	Play	0.019114	False
9	Watch	0.01832	False
10	Adult movie artifact	0.017636	True
11	Broadcast	0.017163	False
12	Resource	0.017132	False
13	Picture	0.013816	False
14	Necessary	0.012849	False
15	Veteran playboy	0.01241	False
16	HD	0.011707	False
17	Film	0.011647	False
18	Artifact	0.011508	False
19	Player	0.011375	False
20	Leisure	0.010474	False
21	Enjoy	0.010227	False
…	…	…	…

6. Evaluation

The experiments follow the steps described below. First, we collect application description information on web pages and construct a dataset for modelling and evaluation. Then, feature words are extracted, and the model is trained based on the constructed dataset. The effectiveness of the method proposed in this paper is evaluated through the following experiments.

6.1. Dataset Creation. Due to the lack of public objectionable short text datasets for the semantic detection of objectionable content, we must manually create a dataset. The application descriptions with pornographic information are designated as references for creating a dataset. We collect application descriptions from the Internet and select 1200 objectionable application descriptions and 5000 normal application descriptions through keyword filtering and manual checking.

We construct a text dataset (DS1, 1200 messages) with objectionable application description information and another text dataset (DS2, 5000 messages) with normal application description information. The dataset is publicly available at https://github.com/buptnsrc/chinese_objectionable_short_corpus.

Table 3: Datasets.

Dataset name	Training dataset	Testing dataset
DS2_DS1_1	DS_i, $i = 2, 3, 4, 5, 6, 7, 8, 9, 10$	DS_1, DS2
DS2_DS1_2	DS_i, $i = 1, 3, 4, 5, 6, 7, 8, 9, 10$	DS_2, DS2
DS2_DS1_3	DS_i, $i = 1, 2, 4, 5, 6, 7, 8, 9, 10$	DS_3, DS2
DS2_DS1_4	DS_i, $i = 1, 2, 3, 5, 6, 7, 8, 9, 10$	DS_4, DS2
DS2_DS1_5	DS_i, $i = 1, 2, 3, 4, 6, 7, 8, 9, 10$	DS_5, DS2
DS2_DS1_6	DS_i, $i = 1, 2, 3, 4, 5, 7, 8, 9, 10$	DS_6, DS2
DS2_DS1_7	DS_i, $i = 1, 2, 3, 4, 5, 6, 8, 9, 10$	DS_7, DS2
DS2_DS1_8	DS_i, $i = 1, 2, 3, 4, 5, 6, 7, 9, 10$	DS_8, DS2
DS2_DS1_9	DS_i, $i = 1, 2, 3, 4, 5, 6, 7, 8, 10$	DS_9, DS2
DS2_DS1_10	DS_i, $i = 1, 2, 3, 4, 5, 6, 7, 8, 9$	DS_10, DS2

We split the text dataset DS1 into 10 folds, nine of which are used as the training set, and the remaining data are mixed with the DS2 dataset to generate several new datasets for testing.

Finally, we obtain the dataset DS2_DS1_i ($i = 1, 2, \cdots, 10$), as shown in Table 3. Each dataset contains two parts: one part contains 9 folds of DS1 as the training dataset, and the other part contains DS2 and the ith-fold of DS1 as the testing dataset.

6.2. Experimental Settings and Method

6.2.1. Feature Dataset Evaluation Method. In this paper, we build a feature dataset based on word frequencies and word cooccurrence relationships. Then, we build the aBTMd proposed in this paper based on the constructed feature dataset. To verify that the feature dataset constructed in this paper has a good effect on the detection of objectionable short text, we build 5 feature datasets with and without semantic features. We use a keyword-based filtering method to test the objectionable short text detection effect of the model on these 5 feature datasets.

The first method is a traditional feature word extraction method that does not consider semantics. Taking DS2_DS1_2 as an example, the construction method is as follows. We select all common words in DS1 and DS2. Then, the word frequencies of the words appearing in DS1 are sorted. We select the words that appear in the top R rankings of DS1 to create a list R_list. R is an adjustable parameter. By setting $R = 100$, 150, and 200, we can obtain the lists R_1, R_2, and R_3, respectively.

The second method is based on semantic feature word extraction. The keyword list contains three types of keywords: obvious keywords with objectionable meanings, hidden keywords with objectionable meanings, and logical keywords with objectionable meanings combined with other words. The feature dataset constructed by the feature word extraction method developed in Reference [9] is named R_a, and the feature dataset constructed using the method proposed in this paper to extract feature words is named R_b, which contains sensitive word values and weights.

The criteria for objectionable short text judgment with the keyword-based filtering method are defined as follows: we consider a sentence objectionable if the ratio of the number of keywords (T_1) in the document to the total number of words (T_2) is above a threshold value r_d.

$$r = \frac{T_1}{T_2} > r_d. \tag{12}$$

6.2.2. aBTMd Evaluation Method. To evaluate our self-adaptive algorithm, comparative experiments are performed with the aBTMd and an LDA-based detection model (LDAd) [12]. Based on the feature datasets R_a and R_b, we use BTM version v0.5 (Available from https://github.com/xiaohuiyan/BTM.) to build the adaptive detection model proposed in this paper. We utilize two detection models: aBTMd-Ra and aBTMd-Rb. We also build the LDA-based detection model utilized in Reference [12] on feature datasets R_a and R_b and obtain detection models LDAd-Ra and LDAd-Rb, respectively. Intelligence-based analysis filtering methods, such as an SVM, KNN, and logistic regression algorithms, are not suitable for the detection of objectionable short text, so we do not compare our aBTMd with these approaches.

The criteria for objectionable short text judgement with the aBTMd and LDAd refer to Formula (12).

6.2.3. Evaluation Indices. An evaluation index is used to illustrate the ability of the constructed feature dataset and detection model to detect objectionable short text. We use the receiver operating characteristic (ROC) curve and area under the ROC curve (AUC) for evaluation. Regarding the ROC curve, the ordinate represents the detection rate (DR), and the abscissa represents the false alarm rate (FR). They also represent detection precision (true positive rate) and false positive rate, respectively. The AUC refers to the size of the area under the ROC curve in the coordinate system. The larger the AUC value is, the better the detection effect.

We conduct a detection test on each dataset in DS2_DS1_i ($i = 1, 2, \cdots, 10$) using different detection models. For the test involving each dataset, we calculate the DR and FR by changing the thresholds r_d in Formula (12) and r in Formula (13). And we then record the average value as the performance index of the entire dataset. The process is as follows.

For each dataset in DS2_DS1_i ($i = 1, 2, \cdots, 10$), we can obtain five feature datasets, including R1_i ($R = 100$), R2_i ($R = 150$), R3_i ($R = 200$), Ra_i, and Rb_i. Ra_i and Rb_i are sensitive feature datasets containing 200 keywords with weights.

During the training phase, we train the detection model to obtain a detector $\mathcal{D}ji$ based on the dataset DS2_DS1_i and the feature dataset Rj_i ($i = 1, 2, \cdots, 10$, $j = 1, 2, 3, a, b$).

During the detection phase, by adjusting the threshold $rd\_ji\_m\{rd\_ji\_m = 0.1, 0.2, 0.3 \cdots, 1 \mid m = 1, 2, 3 \cdots, 10\}$, for each dataset DS2_DS1_i, we can obtain a pair (FRji_m, DRji_m) via Formulas (14) and (15) according to $\mathcal{D}ji$ (DS2_DS1_i, rd_ji_m).

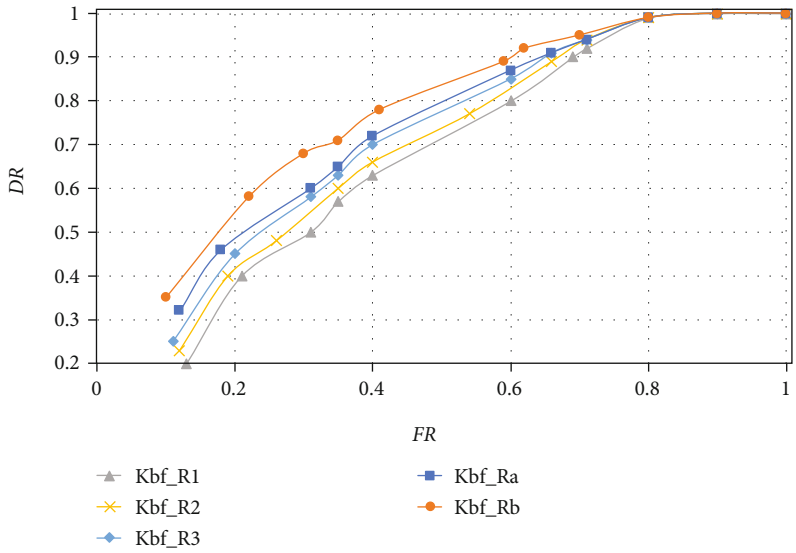


Figure 2: ROC graphs of the Kbf_Ri detection models.

Then, we can calculate the average (FRj_m, DRj_m) of the detection model $\mathcal{D}j(rd\_j\_m)$ on the entire dataset with Formulas (16) and (17).

$$\mathrm{FR}ji\_m = \frac{\mathrm{TP}ji\_m}{\mathrm{TP}ji\_m + \mathrm{FN}ji\_m}, \tag{13}$$

$$\mathrm{DR}ji\_m = \frac{\mathrm{FP}ji\_m}{\mathrm{FP}ji\_m + \mathrm{TN}ji\_m}, \tag{14}$$

$$\mathrm{FR}j\_m = \frac{\sum_{i=1}^{n} \mathrm{FR}ij\_m}{n}, \tag{15}$$

$$\mathrm{DR}j\_m = \frac{\sum_{i=1}^{n} \mathrm{DR}ij\_m}{n}. \tag{16}$$

In the above formulas, TPji_m represents the true positive count obtained when detector $\mathcal{D}ji$ is used for detection on dataset DS2_DS1_i and feature dataset Rj_i, and the threshold is set to rd_ji_m. Similarly, TNji_m represents the true negative count, FNji_m represents the false negative count, and FPji_m represents the false positive count in the same situation. In this paper, the value of n is 10.

6.3. Detection Performance. We use the training dataset in DS2_DS1_i ($i = 1, 2, \cdots, 10$) to generate feature datasets R_1, R_2, R_3, R_a, and R_b. Based on these feature datasets, we build a keyword-based filtering model (Kbf), the aBTMd, and the LDAd, and we use the test dataset in DS2_DS1_i for testing purposes.

Our experiments are divided into three parts: verifying the validity of the sensitive feature dataset, verifying the effectiveness of the adaptive BTM, and evaluating the aBTMd at a global scale.

6.3.1. Validation of the Sensitive Feature Dataset. To show that the sensitive feature word extraction method proposed in this paper can extract the feature words in objectionable short text more accurately than other approaches, we use the keyword-based filtering method to construct a detection model (Kbf) on the feature datasets and verify it on the test dataset.

We follow the steps in the following to perform our experiment:

(i) For each dataset in DS2_DS1_i ($i = 1, 2, \cdots, 10$), we can obtain five feature datasets, including R1_i ($R = 100$), R2_i ($R = 150$), R3_i ($R = 200$), Ra_i, and Rb_i

(ii) We use a keyword-based filtering method to build the detector Kbf_Rj_i based on the dataset DS2_DS1_i and the feature dataset Rj_i ($i = 1, 2, \cdots, 10$, $j = 1, 2, 3, a, b$)

(iii) By adjusting the threshold $rd\_ji\_m\{rd\_ji\_m = 0.1, 0.2, 0.3 \cdots, 1 \mid m = 1, 2, 3 \cdots, 10\}$, for each Kbf_R$j_i$, we can obtain a series of (FRji_m, DRji_m) according to Formulas (14) and (15)

(iv) Then, we can refer to Formulas (16) and (17) to calculate the average (FRj_m, DRj_m) according to the detection model Kbf_R$j(rd_j_m)$ over the entire dataset DS2_DS1_i ($i = 1, 2, \cdots, 10$)

(v) Finally, we can draw the ROC curves of the detection models Kbf_R1, Kbf_R2, Kbf_R3, Kbf_Ra, and Kbf_Rb, as shown in Figure 2

From Figure 2, it can be seen that the detection results obtained with different feature datasets vary greatly. By comparing the AUCs of Kbf_R1, Kbf_R2, and Kbf_R3, the Kbf_R3 model has the best detection effect. By comparing the AUCs of Kbf_R3, Kbf_Ra, and Kbf_Rb, the Kbf_Rb model has the best detection effect.

As we can see, when the number of selected feature words is increased, the effect of the keyword filtering algorithm is improved. The feature datasets R_1, R_2, and R_3 are generated according to the word frequencies, and R_a and

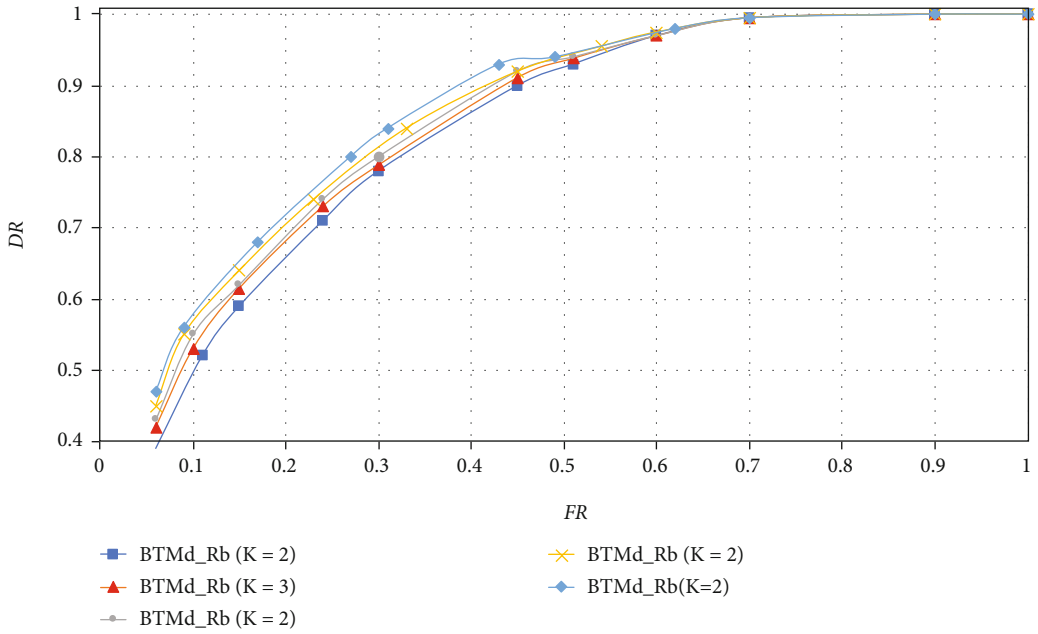


Figure 3: ROC graphs of the BTMd and aBTMd.

R_b are generated according to the word frequencies and semantics, so they contain sensitive word weights. We know that R_a and R_b can better preserve corpus features, and the detection models built on them can filter objectionable short text effectively. At the same time, the detection effect of the keyword-based filtering model generated by the feature dataset R_b extracted in this paper is better than that of the model generated by the feature dataset R_a extracted in [9].

6.3.2. The Effectiveness of the Adaptive BTM. For a comparison with the traditional BTM-based detection method, we set the number of topics in the BTM to $K = 2, 3, 4, 8$ to evaluate the impact of this parameter on the detection performance. When constructing the BTM-based detection model, we perform the following operations for $K = 2, 3, 4, 8$ and the adaptive values (aBTMd):

(i) For each dataset in DS2_DS1_i ($i = 1, 2, \cdots, 10$), we can obtain feature datasets Rb_i

(ii) We use the BTM to build detectors BTMd_Rb_i ($K = 2$), BTMd_Rb_i ($K = 3$), BTMd_Rb_i ($K = 4$), BTMd_Rb_i ($K = 8$), and aBTMd_Rb based on the dataset DS2_DS1_i and feature dataset Rb_i ($i = 1, 2, \cdots, 10$)

(iii) By adjusting the threshold $rd\_i\_m\{rd\_i\_m = 0.1, 0.2, 0.3 \cdots, 1 \mid m = 1, 2, 3 \cdots, 10\}$, for each detection model, we can obtain a series of (FRi_m, DRi_m)

(iv) Then, we can refer to Formulas (16) and (17) to calculate the average (FR_m, DR_m) according to the detection model BTMd_Rb (K, rd_m) over the entire dataset DS2_DS1_i ($i = 1, 2, \cdots, 10$)

(v) Finally, we can draw the ROC curves of the detection models BTMd_Rb ($K = 2$), BTMd_Rb ($K = 3$), BTMd_Rb ($K = 4$), BTMd_Rb ($K = 8$), and aBTMd_Rb, as shown in Figure 3

It should be noted that the K value (number of topics) selected by the BTM-based adaptive model is 6. We can see that when the K value is closer to 6, the larger the AUC value is, the better the detection effect. Therefore, the K value selected by the aBTMd can more effectively filter objectionable short text.

In this paper, the BTM-based adaptive filtering framework generates a weighted sensitive feature dataset based on seed words and the original input dataset and automatically calculates the appropriate K value. This process retains the sensitive word meanings of the original dataset to a certain extent and is completed automatically. Therefore, the aBTMd is applicable to other similar datasets.

6.3.3. Global Evaluation of aBTMd. We experimentally compare the detection effects of the aBTMd and LDAd with respect to objectionable short text. On the basis of experiment B, we use feature datasets R_a and R_b to train the LDA-based topic model [12] to obtain LDAd_Ra and LDAd_Rb, respectively, and we use feature dataset R_b to train the adaptive BTM model to obtain aBTMd_Rb. We use the method in Experiment B to obtain the ROC curves for the three detection models, as shown in Figure 4.

As we can see from Figure 4, aBTMb_Rb has the largest AUC value, so its detection effect is the best.

The LDA model encounters the data sparsity problem in short documents. The BTM learns the topics by directly modelling the generation of word cooccurrence patterns in the whole corpus. The BTM uses biterms to represent documents instead of words, which are used in the LDA model.

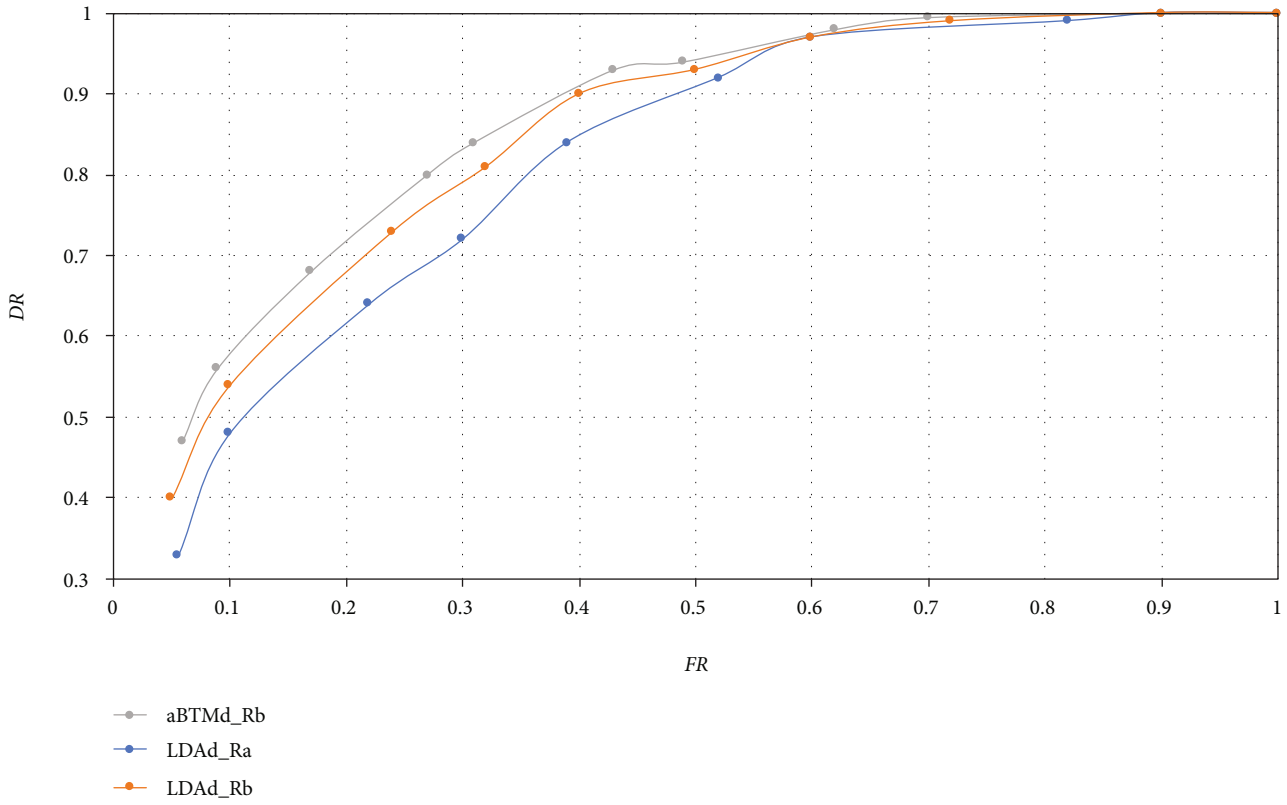


Figure 4: ROC graphs of the LDAd and aBTMd.

Table 4: Performance.

Detection method	Feature selection (ms)	Model inference (ms)	Test (ms)
tBTMd	—	8238	7440
aBTMd	51687	39423	6667
Kbf	—	—	6123

The proposed method can solve the data sparsity problem. When the FR falls in the interval [0.1, 0.4], the detection results of the aBTMd are significantly better than those of the LDAd.

6.4. Complexity Analysis and Discussion. To further evaluate the performance of the proposed method, we check its time complexity. The experiment is conducted on a personal computer configured with an Intel Core i5 7500 CPU @ 3.4 GHz and 8 GB of memory. We test three models: Kbf, the traditional BTM ($K = 4$) tBTMd, and the aBTMd. The evaluation is based on all datasets DS2_DS1_i ($i = 1, 2, \cdots 10$) and calculates the average elapsed time required for each model. The results are shown in Table 4.

The aBTMd proposed in this paper must find a suitable K value during the model derivation process, so it requires a long running time. The aBTMd consumes more time than the tBTMd. In addition, the feature selection process requires additional calculation time. However, from the perspective of the detection process, because the feature space is relatively small, the detection time is faster than that of the tBTMd. Because model training can be performed offline, the complexity of the model training process is acceptable.

7. Conclusions

In this paper, we provide an adaptive method for calculating the number of topics in a BTM model without setting it in advance for different datasets. By introducing the sensitivities of feature words, the importance levels of words are quantified. Then, we develop an aBTMd that can effectively detect objectionable short text with a high detection rate and a low false detection rate. How to include synonyms in the feature dataset and combine the dataset with the topic model will be the focuses of future work. The dataset used in this paper can be obtained through https://github.com/buptnsrc/chinese_objectionable_short_corpus.

Data Availability

The dataset used in this paper can be obtained through https://github.com/buptnsrc/chinese_objectionable_short_corpus.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] K. S. Myers, "Wikimmunity: fitting the communications decency act to Wikipedia," vol. 1, Social Science Electronic Publishing, 2006.

[2] Z. Zhou, T. Song, and Y. Jia, "A high-performance URL lookup engine for URL filtering systems," in *IEEE International Conference on Communications*, Cape Town, May 2010.

[3] M.-S. Lin, C.-Y. Chiu, Y.-J. Lee, and H.-K. Pao, "Malicious URL filtering - a big data application," in *2013 IEEE International Conference on Big Data*, Silicon Valley, CA, USA, October 2013.

[4] D. Sahoo, C. Liu, and S. C. H. Hoi, *Malicious URL detection using machine learning: a survey*, 2017.

[5] D. B. Skillicorn, "Beyond keyword filtering for message and conversation detection," in *International Conference on Intelligence and Security Informatics*, pp. 231–243, Springer, Berlin, Heidelberg, 2005.

[6] C. Zimmer, C. Tryfonopoulos, and G. Weikum, "Exploiting correlated keywords to improve approximate information filtering," 2008.

[7] H. Gu, W. Wang, P. Liu, S. Zhang, J. Liu, and C. Wang, "A system for web page sensitive keywords detection," in *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems. IEEE*, pp. 370–374, Shenzhen China, Hong Kong, China, 2014.

[8] R. P. Ganar and S. Ardhapurkar, "Prediction of civil unrest by analysing social network using keyword filtering: a survey," in *2016 Online International Conference on Green Engineering and Technologies (IC-GET). IEEE*, pp. 1–4, Coimbatore, India, 2016.

[9] J. Zeng, J. Duan, and C. Wu, "Adaptive topic modeling for detection objectionable text," in *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). IEEE*, pp. 381–388, Atlanta, GA, USA, 2013.

[10] C. M. Chen, H. M. Lee, and C. C. Tan, "An intelligent webpage classifier with fair feature-subset selection," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 8, pp. 967–978, 2006.

[11] M. Haddoud, A. Mokhtari, T. Lecroq, and S. Abdeddaïm, "Combining supervised term-weighting metrics for SVM text classification with extended term representation," *Knowledge and Information Systems*, vol. 49, no. 3, pp. 909–931, 2016.

[12] J. Duan and J. Zeng, "Web objectionable text content detection using topic modeling technique," *Expert Systems with Applications*, vol. 40, no. 15, pp. 6094–6104, 2013.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[14] Y. Chen, W. Li, W. Guo, and K. Guo, "Popular topic detection in Chinese micro-blog based on the modified LDA model," in *2015 12th Web Information System and Application Conference (WISA). IEEE*, pp. 37–42, Jinan, China, 2015.

[15] L. I. Li, L. Yu-Lan, and Y. Rui-Bo, *Interactive Text Theme Mining Based on LDA Model-Take Customer Service Chat as an Example*, Information Science, 2018.

[16] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A bi-term topic model for short texts," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 1445–1456, Rio de Janeiro, Brazil, 2013.

[17] R. Du, R. Safavi-Naini, and W. Susilo, "Web filtering using text classification," in *The 11th IEEE International Conference on Networks, 2003. ICON2003. IEEE*, pp. 325–330, Miami, FL, USA, 2003.

[18] X. Jin, Y. Li, T. Mah, and J. Tong, "Sensitive webpage classification for content advertising," in *Proceedings of the 1st international workshop on Data mining and audience intelligence for advertising*, pp. 28–33, San Jose, California, 2007.

[19] N. Agarwal, H. Liu, and J. Zhang, "Blocking objectionable web content by leveraging multiple information sources," *Acm Sigkdd Explorations Newsletter*, vol. 8, no. 1, pp. 17–26, 2006.

[20] M.-Y. Kan and H. O. N. Thi, "Fast webpage classification using URL features," 2005.

[21] L. H. Lee and H. H. Chen, "Collaborative cyberporn filtering with collective intelligence," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 1153-1154, Beijing, China, 2011.

[22] F. Ali, P. Khan, K. Riaz et al., "A fuzzy ontology and SVM-based web content classification system," *IEEE Access*, vol. 5, pp. 25781–25797, 2017.