TSSRD: A Topic Sentiment Summarization Framework Based on Reaching Definition

Xiaodong Li, Chenxin Zou, Pangjing Wu, and Qing Li, *Fellow, IEEE*

Abstract—Exposure to massive information in daily lives makes it necessary for people to obtain major points efficiently, promoting the development of text summarization technology. However, existing sentiment-based text summarization methods only pay attention to the sentiment polarity of either a single sentence or a whole document, ignoring changes of sentiments along with sentences or sentiment flow across the whole document. To incorporate the above two aspects into the summarization process to generate high-quality summaries, we propose a topic sentiment summarization framework based on reaching definition (TSSRD). In the framework, we first use topic models to model documents and calculate topic sentiment embeddings. Then, we analyze document structures from different perspectives to design data flow diagrams, in which improved reaching definition is used to analyze sentiment changes and sentiment flow. Finally, topic sentiment summaries are generated based on sentiments in steady states of the reaching definition. To evaluate our summarization framework, we introduce an extrinsic evaluation method. In this method, a sentiment classifier is trained by the topic sentiment summaries, and accuracy of the sentiment classification is used as a quality score. Experimental results demonstrate that our summarization framework is at least 2.32% better than baselines on IMDb and Amazon datasets.

Index Terms—Reaching definition, sentiment analysis, summarization

1 Introduction

With the development of the Internet, people can receive massive textual information in their daily lives, but only a few parts of the information are what they need. So people usually spend a lot of time filtering and processing information. To extract major points from texts more efficiently, researchers propose automatic text summarization generation technologies which help reduce the cost of information extraction.

In 1950, scholars began to study automatic text summarization. With the development of text processing, more levels of semantic information are used to generate better summaries. Sentiment analysis plays a very important role in most text summarization methods. For documents full of sentiments, sentiment analysis can help improve summary quality by identifying and quantifying sentiments in documents [1]. Most of existing extractive text summarization methods are based on sentiment analysis [2], [3] to build relevant evaluation indicators and use these indicators to score sentences. They finally extract top-k sentences with the highest scores to form a final summary [4].

However, these summarization methods based on sentiment analysis have a common problem: they just consider the sentiment polarity of each single sentence or a whole document, but ignore changes of sentiments between sentences and how sentiments flow across all sentences of the document. There are always sentiment changes between two different sentences, which can be named *local change*. Specific structures of a document not only express relationships between literal meanings but also imply sentiment flow in the whole document, which can be regarded as *global change*. For example, an author may have a mind to organize contents and positions of sentences to better explain his intentions and express sentiments, and his sentiment changes can be inferred from the document structures. To model the local changes and the global changes of sentiments, we propose a topic sentiment summarization framework based on reaching definition (TSSRD). In this framework, we first use topic models to model documents and calculate topic sentiment embeddings. Then, we reveal change regularities and flow paths of sentiments by analyzing document structures from perspectives of *readers* and *writers* in order to design data flow diagrams. The traditional reaching definition [5] is improved based on differences between sentiments and assignments of variables to analyze sentiment changes and flow in the designed diagrams. Finally, we generate topic sentiment summaries according to sentiments in steady states of the reaching definition.

To evaluate our proposed framework, we use an extrinsic evaluation method. In this method, a sentiment classifier is trained by the generated summaries, and accuracy of the sentiment classification is used as a quality score. Experimental results demonstrate that the proposed framework is at least 2.32% better than baselines on IMDb[1] and Amazon[2] datasets.

- Xiaodong Li, Chenxin Zou, and Pangjing Wu are with the College of Computer and Information, Hohai University, Nanjing 210098, China. E-mail: {xiaodong.c.li, pangjing.wu}@outlook.com, zoucx@hhu.edu.cn.
- Qing Li is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong 999077, China. E-mail: csqli@comp.polyu.edu.hk.

Manuscript received 30 Jan. 2022; revised 7 June 2022; accepted 18 June 2022. Date of publication 24 June 2022; date of current version 13 Sept. 2023.
This work was supported in part by the National Natural Science Foundation of China under Grant 61602149, and in part by the Fundamental Research Funds for the Central Universities, China under Grant B210202078.
(Corresponding author: Chenxin Zou.)
Recommended for acceptance by The Guest Editors of the Special Issue on Neuro-symbolic Artificial Intelligence for Sentiment Analysis.
Digital Object Identifier no. 10.1109/TAFFC.2022.3186015

[1] http://ai.stanford.edu/~amaas/data/sentiment/
[2] https://nijianmo.github.io/amazon/

Our main contributions are as follows:

1) We take the local and global changes of sentiments in a document ignored by previous summarization methods into consideration, and generate better summaries with sentiments.
2) We propose the TSSRD summarization framework, which uses the reaching definition to analyze the sentiment changes between sentences and the sentiment flow across all sentences, and generates topic sentiment summaries when sentiments reach steady states.
3) We reveal the change regularities and flow paths of sentiments implicated in specific structures of documents from two different perspectives: *readers* and *writers*.
4) We introduce an extrinsic evaluation method to evaluate our framework, which uses a sentiment classifier to calculate a quality score for the summarization framework. The experimental results show that our framework performs statistically significantly better than baselines.

The remaining sections of this paper are arranged as follows. Section 2 introduces some work related to our framework. Section 3 proposes the TSSRD summarization framework. Section 4 describes the summarization and evaluation experiments, and gives some related discussions. Section 5 gives our conclusion and indicates the future work.

2 Related Work

Since 1950, automatic text summarization technology has appeared. In the beginning, text summarization based on statistical information was the most common. Luhn *et al.* [6] proposed an extractive text summarization method based on the frequency of words and phrases. They focused on high-frequency words but ignored auxiliary words. Later, with the development of various text analysis technologies, many related technologies are also used in text summarization methods. Among them, topic analysis and sentiment analysis are commonly used. Topic models help mine the latent semantics of documents. Sentiment analysis provides a new perspective to understand semantics. Besides, joint sentiment-topic models emphasize the relationship between topic and sentiment, which is more conducive to fine-grained semantics understanding.

2.1 Text Summarization Based on Topic Analysis

Topic analysis-based text summarization usually uses topic models to model documents. The Latent Dirichlet Allocation (LDA) topic model [7] is the most commonly used. There are also some neural topic models, such as NQTM [8] and CRNTM [9]. In existing extractive text summarization methods, topic models are usually used for sentence screening. Scholars always take the outputs of topic models as sentence evaluation indicators directly or use them to construct better indicators.

Outputs of the LDA model are document-topic and topic-word distributions, which can be directly used to score sentences. According to the topic distribution of sentences, Roul *et al.* [10] constructed a two-dimensional weight matrix. Rows represented sentences and columns represented topics. Under each column, sentences with top-k weights were selected for the first round of sentences screening. Later, Roul [11] made further optimization on the basis of [10]. He found that the topic number can be arbitrary and the most appropriate value can not be determined. To address this issue, he made use of topic-word distribution to calculate topic numbers before first-round sentences screening. Wu *et al.* [12] were different from [10] and [11], they identified topic words according to the topic-word distribution and then selected sentences related to topic words as candidates. Gialitsis *et al.* [13] regarded text summarization as a two-classification task. According to the distribution of words in each topic, a sentence feature vector whose length equaled the topic number was constructed as the input of a classifier. The classifier predicted whether the sentence was a part of the golden summary.

Outputs of topic models can also be used to further build better evaluation indicators. Rani *et al.* [14] designed three indicators: redundancy rate, inclusive topic diversity, and exclusive topic diversity according to the sentence-topic distribution and the topic-word distribution. They selected sentences based on them to form different candidate summaries and finally chose one which retained the most information. Xu *et al.* [15] also designed three sentence scoring methods based on frequencies and weights of topic words and Okapi BM25. According to these three scoring methods and similarity between sentences, the most appropriate sentences were selected iteratively to form summaries. Khurana *et al.* [16] introduced Non-negative Matrix Factorization to calculate the entropy of topics in a latent space and selected informative sentences from important topics as a summary through information theoretic principle. Besides, topic analysis can help calculate edge weights in some graph models used for summarization. Belwal *et al.* [17] first selected top-k words under each topic as keywords according to the topic-word distribution. Then they calculated phrases similarity of two sentences and the similarity between the union of phrases and key phrases of whole documents by WordNet to determine the edge weights in the graph. Finally, the weighted PageRank algorithm in TextRank [18] was used to select top-k sentences to form final summaries. In contrast, Li *et al.* [19] proposed the Self-Present Sentence Relevance (SPSR) algorithm, which used cosine similarity of two sentences as edge weights and selected sentences according to their contribution to the weights of all edges connected to them. Jang *et al.* [20] calculated sentence importance and sentence similarity by leveraging deep representations. Then they introduced integer linear programming to extract sentences as summaries.

2.2 Sentiment Analysis

Sentiment analysis is to explore implicit sentiment information in text and make a correct judgment on sentiments of the text. It belongs to a task of natural language processing [21]. Most existing methods are based on deep learning. The analysis process of these methods is similar. They first obtain sentiment related feature information from the input text, and then use these features to identify the sentiments. Their differences lie in the features considered and the ways of feature extraction. Basiri *et al.* [22] used Bi-LSTM and Bi-GRU based on attention to obtain long-term and short-term dependencies respectively, and used CNN to extract important features. Li *et al.* [23] adopted the idea similar to RNN and obtained context information of current time according to current utterance and previous context information. Then LSTM and CNN were used for feature extraction to obtain sentiment features. Similarly, Gan *et al.* [24] designed a scalable multi-channel dilated CNN-BiLSTM to obtain different higher-level context information. Song *et al.* [25] proposed a semantic perception and refinement network to capture context information and filter noise. LSTM is widely used in sentiment analysis. Huang *et al.* [26] found that emotions would affect memory process and it was directly reflected in the forget gate of LSTM. Therefore, they added an emotion estimator to the original LSTM to help the forget gate judge how much information should be retained. This improvement balanced information memory and forgetting. In the above methods, the features are automatically extracted by neural network. Besides, researchers also designed some important features based on experience, such as Log Term Frequency and Modified Inverse Class Frequency [27]. To comprehensively consider the artificially designed features and the features automatically extracted by neural network, Ray *et al.* [28] proposed an ensemble-based method. They captured these two kinds of features by integrating two different BERT and a random forest model. In addition to the information of text itself, external knowledge is often introduced to help analyze sentiments. Zhao *et al.* [29] added relevant triples to the input text and used BERT for encoding by the soft-position embedding strategy based on a sentiment knowledge graph.

In summary, there are two main differences between the above sentiment analysis methods and our framework. On the one hand, aims are different. The above methods are to predict sentiments of text as correctly as possible. However, our framework is to summarize sentiments. Using the obtained sentiment summaries to predict sentiments only helps to evaluate the performance of the framework. On the other hand, the explainability of sentiment analysis is different. The above methods are supervised and they belong to deep learning methods. The process of sentiment analysis is unexplainable. In our framework, we can clearly know how sentiments change at each step and how sentiments flow across full text while using reaching definition to analyze sentiments. The whole process is explainable. There are also explainable sentiment analysis methods. For example, Vashishtha *et al.* [30] calculated sentiment scores of phrases according to different collocations between words and selected keywords through fuzzy entropy to analyze sentiments. The aim of this method is also different from our framework, but the results of sentiment prediction are better on IMDb while evaluating our framework.

2.3 Text Summarization Based on Sentiment Analysis

Human beings are rich in sentiments. People always express their sentiments unconsciously while describing something. These sentiment information implies their preferences [31]. When summarizing the text (e.g., reviews) containing sentiments, it is necessary to make a sentiment analysis. The obtained summaries should contain the main sentiments of original documents.

Existing methods usually divide sentiments into three categories: positive, negative, and neutral. The sentences that can best express sentiments are selected to form a summary. Akhtar *et al.* [2] used SentiWordNet to calculate sentence sentiment scores according to words in sentences, and selected the sentences with the highest top-k scores from each sentiment type as summaries. Mandal *et al.* [3] introduced a PSO-based approach (PsoSA). They used the clustering algorithm based on particle swarm optimization to cluster sentences. The sentence sentiment score also obtained from SentiWordNet was used as the fitness function. A cluster was selected first according to the diversity and coverage, and then some sentences having high scores in the cluster were chosen to generate summaries. Chatterjee *et al.* [32] used the online sentiment dictionary AFINN-111 to calculate sentence sentiment scores as the input of neural networks. The most important sentences were selected by the genetic algorithm to form final summaries.

Both SentiWordNet and AFINN-111 are universal dictionaries, which are trained from annotated data. The quality and size of these data will indirectly impact the results of sentiment analysis [33]. While using the dictionaries to analyze sentiments, there may be two problems: coverage and specificity. Existing dictionaries cannot cover all sentiment words. A same sentiment word may express different sentiments in different corpora. Both of these problems will lead to poor summary performance. To solve the coverage problem, Abdi *et al.* [34] proposed two ways: one was to combine multiple sentiment dictionaries, standardize calculation results to the same interval, and then get a unified result; the other was to calculate sentiment scores of synonyms obtained from WordNet. To address the specificity issue, existing methods mainly relied on the semantic information of the text itself to analyze sentiments with the help of machine learning. Ali *et al.* [35] integrated semantic features into a Bayesian classifier to identify the sentiments of documents. In addition, some extra information, such as idioms [36], modifiers [37], and predefined rules [38], is helpful for sentiment analysis. Bompotas *et al.* [38] used a predefined rule-based sentiment annotator and LSTMs to analyze the sentiment polarity of sentences. Li *et al.* [39] constructed a sentiment classifier based on hierarchical attention neural network. Original documents and the corresponding standard summaries were used to train the classifier.

2.4 Joint Sentiment-Topic Models

The summarization methods based on sentiment analysis usually only focus on sentiments of sentences. However, a sentence may contain multiple topics, and the sentiment of each topic is different [40]. Focusing only on sentence sentiments will lead to the loss of some topic sentiments. Therefore, the joint sentiment-topic models are proposed to analyze text more finely-grained.

Earlier, Lin *et al.* [41] proposed a joint model (JST) that added a sentiment layer based on LDA. While generating the words, it drew a sentiment label according to the document-sentiment distribution first, then drew a topic based on the sentiment-topic distribution, and finally generated a word in the light of the sentiment-topic-word distribution. Dermouche *et al.* [42] thought topics from different sentiments should be independent. They swapped the sentiment layer and topic layer of JST to build a new joint model, which was more in line with people's writing habits that they considered topics first, and then expressed their sentiments about the topic through words. Dong *et al.* [43] used the topic feature and sentiment feature from this model to construct a classifier for detecting deceptive reviews. Besides, potential topic and sentiment information obtained from the joint models can also help companies make useful decisions [44] and find new product opportunities [45].

Many problems (e.g., text sparse [46], [47], time sequences capturing [48], and multimodal data [49]) of these joint sentiment-topic models have been well solved. But sentiment changes are rarely noticed. Rahman *et al.* [50] introduced a hidden Markov model that took sentences as observable states. Two variables were sampled to judge whether there was a sentimental change between two sentences. However, they only focus on the existence of changes. It is far from enough. Our model quantifies these changes and concentrates more on how much sentiments has changed. We consider not only the local sentiment changes between two sentences but also the sentiment flow in whole documents, so as to enable our model to analyze sentiment more comprehensively.

2.5 Reaching Definition Analysis

Reaching definition is a kind of data-flow analysis technology, which calculates the data-flow set of each point in terms of all variables in the program [5]. The reaching definition can be fully used for program optimization [51] and code analysis [52] by establishing and solving a reaching definition data-flow equation. In the reaching definition data-flow analysis, the definition of variable A refers to the assignment or possible assignment of a statement to A. The position of this statement is called the definition point of A. When a definition point d of variable A reaches a certain point p, it means that there is a path from d to p, and d is not "killed" on this path, that is, the variable is not reset. Intuitively speaking, there is a path from d to p in the flow diagram and there are no other definitions of A in that path.

The reaching definition equation is a simultaneous establishment of the following two data-flow equations,

$$OUT[B] = gen[B] \cup (IN[B] - kill[B]), \tag{1}$$

$$IN[B] = \cup(OUT[b]) \qquad b \in P[B], \tag{2}$$

where,

- $IN[B]$ and $OUT[B]$ are definition points of each variable at the entrance and exit of node B, respectively.
- $gen[B]$ are definition points "generated" by B, which means the definition points in node B that can reach the exit of B.
- $kill[B]$ are definition points that satisfy the following conditions:
  - They can reach the entrance of B;
  - The variables related have been reset in B.

In short, they are the definition points that B "kills".

- $P[B]$ are the precursors of node B.

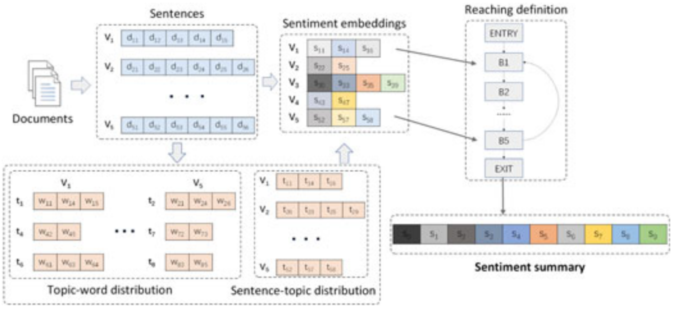


Fig. 1. The workflow of the TSSRD framework. First, the framework preprocesses documents using a topic model. The sentence-topic distribution and topic-word distribution are obtained from the model. Next, the sentiment embedding of each topic is calculated through a sentiment dictionary. Then, document structures are analyzed based on the sentence-topic distribution to construct a data flow diagram where sentences are taken as nodes. The improved reaching definition is used to analyze the sentiment changes and the sentiment flow in the diagram. Finally, when the outflow sentiments of each node no longer change, i.e., in steady states of the reaching definition, the output of the last node is regarded as the final topic sentiment summary.

Topic analysis-based or sentiment analysis-based summarization methods only focus on one of topics and sentiments but ignore the other. Joint sentiment-topic models concentrate on what sentiments are under one topic rather than the difference between them. Reaching definition is a mature analysis technology for data flow. However, to our best knowledge, it has never been compared with summarization. Natural language is similar to program. In a program, the data, whose carrier is variables, changes continuously according to the instructions and becomes one that meets the requirements. In a natural language document, the sentiments, whose carrier is words, also change and finally become one the document would like to express. This is similar to a summary process. In the TSSRD, we take topics as the carriers of sentiments and improve reaching definition according to the difference between program data and sentiments. The improved reaching definition is used to analyze the sentiment changes and the sentiment flow, and obtain a topic sentiment summary.

3 TSSRD Framework

In the TSSRD framework, we consider the sentiment changes between sentences and the sentiment flow across whole documents, which are ignored by existing summarization methods and joint sentiment-topic models. The workflow of the TSSRD is shown in Fig. 1. More details about the framework are introduced in the following five parts.

3.1 Task Definition

The TSSRD aims to summarize single-document texts. The input is a document, which is a set of sentences. To model finely-grained sentiments within a sentence, we mine topics of the sentence by topic models and calculate topic sentiments based on topic-word distribution. The sentiment changes (i.e., the local change) can be analyzed according to sentiments of the same topic between two sentences, and the sentiment flow (i.e., the global change) can be analyzed according to sentiments of the same topic in the whole document. The output summary is a sentiment embedding, which is different from summaries represented by natural languages. In the rest of this section, we will introduce four modules of the TSSRD. For the convenience of reading and

TABLE 1
Notation Interpretation

Notations	Interpretation
V	set of all sentences in a document
W^t	set of topic weights in sentences
W^e	set of word weights under topics
S^t	set of topic sentiments in sentences
v	a sentence
t	a topic
w^t	weight of a topic
e	a word
w^e	weight of a topic
s^t	sentiment of a topic
s^e	sentiment of a word
$\|D\|$	dimension of a sentiment dictionary
$\mathcal{S}$	new assignment of a variable
v_u	a sentence positioned after v_r
v_r	a sentence positioned before v_u
w_u	sentiment weight of v_r
w_r	sentiment weight of v_u
B	a node in sentiment flow diagram
B_u	the node that corresponds to v_u
B_r	the node that corresponds to v_r
s^t_u	assignment of a variable in B_u
s^t_r	assignment of a variable in B_r
I_l	set of the last node indices in all branch paths
h	node depth in a path
s^o	output sentiments of a node
α, β	weighted average parameters
θ	topic sparsification parameter

understanding, we summarize notations used in this paper and corresponding interpretations in Table 1.

3.2 Document Representation Module

The TSSRD framework uses topic models to model documents at sentence level. Each sentence in a document is represented as sentence-topic distribution and topic-word distribution. The topic sentiments of sentences will be further calculated in the subsequent modules. Finally, a document can be represented as a quadruple $< V, W^t, W^e, S^t >$, where,

- $V = \{v_i \mid \text{the } i\text{th sentence}\}$;
- $W^t = \{w^t_{ij} \mid \text{the weight of } j\text{th topic in sentence } v_i\}$;
- $W^e = \{w^e_{jk} \mid \text{the weight of } k\text{th word under topic } t_j\}$;
- $S^t = \{s^t_{ij} \mid \text{the sentiment of } j\text{th topic in sentence } v_i\}$.

3.3 Sentiment Calculation Module

To calculate topic sentiments of sentences, we introduce a sentiment dictionary. The sentiment dictionary has analyzed sentiments of a large number of words and phrases in advance and stores their sentiment embeddings in the dictionary. We are able to get sentiment embeddings of words under each topic by a query operation,

$$s^e = query(e), \tag{3}$$

where e is a word and s^e is the sentiment embedding of e. The dimension of sentiment embeddings depends on the dimension of the sentiment dictionary. Therefore, the sentiment embedding of each word e_{ij} under topic t_j in sentence v_i is,

$$s^e_{ij} = query\big(e_{ij}\big) \qquad s^e_{ij} \in \mathbb{R}^{|D|}, \tag{4}$$

where $|D|$ represents the dimension of the sentiment dictionary. Supposing topic sentiments can be obtained by linear superposition of word sentiments within each topic, the sentiment embedding of topic t_j in sentence v_i can be represented by an average value of the word sentiment embeddings,

$$s^t_{ij} = \frac{1}{|t_j|} \sum_{e_{ij} \in t_j} query\big(e_{ij}\big), \tag{5}$$

where $|t_j|$ represents the number of words under topic t_j.

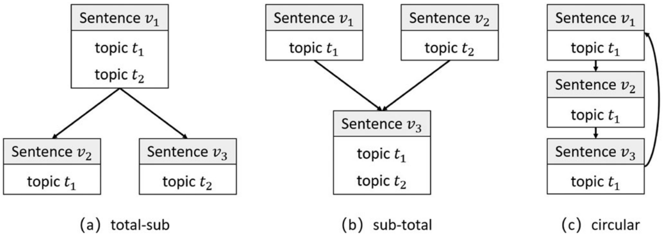
3.4 Reaching Definition Analysis Module

In the TSSRD framework, the reaching definition analysis module is used to analyze the sentiment changes between sentences and the sentiment flow in whole documents. This module can be further divided into two sub-modules: sentiment flow diagram design module and sentiment flow analysis module.

3.4.1 Sentiment Flow Diagram Design Module

This sub-module is to analyze document structures from perspectives of *readers* and *writers* and reveal change regularities and flow paths of sentiments to design a sentiment flow diagram. As *readers*, we usually read a document from front to back repeatedly until we have a clear idea about the document. Sentiments flow in sentence order and return to the first sentence when they reach the last one. As *writers*, we always carefully arrange the contents and positions of sentences to describe intentions and express sentiments to the greatest extent. There are three common structures for organizing sentences: total-sub, sub-total, and circular structure. In the total-sub structure, an overview is given first, and then each point is described. On the contrary, each point is stated first, and then a summary is given in the sub-total structure. The circular structure usually plays an emphasis role, i.e., describing the same topics repeatedly. Examples of the structures are given in Fig. 2 for better understanding. We parse these structures in documents through relationships between topics of adjacent sentences. More details are shown in Algorithm 1.

After document structure analysis, a document can be parsed into a tree-like structure. We take each sentence in the document as a node, each topic in the sentence as a variable in the node, and each sentiment embedding of the topic as an assignment of the related variable to construct a sentiment flow diagram. Based on document structures, the structure of the sentiment flow diagram is designed as a single-path structure or a multi-path structure from different perspectives. For example, suppose that there are three sentences v_1, v_2 and v_3 in a document. The sentence v_1 is about topics t_1 and t_2, v_2 about t_1, and v_3 about t_2. The three sentences correspond to three nodes B_1, B_2, and B_3 in the sentiment flow diagram respectively. From the perspective of *readers*, the sentiment flow path is: from B_1 to B_2, from B_2 to B_3, and from B_3 back to B_1. The structure of the diagram is a single-path structure. From the perspective of *writers*, the path is: from B_1 to B_2 and B_3, and from B_2 and B_3 back to



(a) total-sub (b) sub-total (c) circular

Fig. 2. Examples of (a) total-sub, (b) sub-total, and (c) circular structure.

B_1. The structure of the diagram is a multi-path structure. It is worth noting that the multi-path structure is equivalent to the single-path structure when there are not any special structures in the document.

Algorithm 1. Document Structure Analysis

```
Require: Sentences $V = \{v_1, v_2, \ldots, v_m\}$.
Ensure: Structure $U$.
 1: for $v_i \in V$ do
 2:   $vt_i \leftarrow LDA(v_i)$;                      ▷get topics of each sentence
 3:   add $vt_i$ to $VT$;
 4: end for
 5: for $i \leftarrow 1$ to length($VT$) do
 6:   for $j \leftarrow i-1$ to 1 do                    ▷traverse forward
 7:     if $vt_j \subset vt_i$ and $continuity\_flag1 == 0$ then
 8:       $left\_proper\_subset++$;
 9:     else
10:       $continuity\_flag1 = 1$;
11:     end if
12:     if $vt_j = vt_i$ and $continuity\_flag3 == 0$ then
13:       $equal\_set++$;
14:     else
15:       $continuity\_flag3 = 1$;
16:     end if
17:   end for
18:   for $j \leftarrow i+1$ to length($VT$) do         ▷traverse backward
19:     Similar to 7 to 16;
20:   end for
21:   if $left\_proper\_subset >= 2$ then
22:     add 1 to $U$;                     ▷there is a "sub-total" structure
23:   else if $right\_proper\_subset >= 2$ then
24:     add 2 to $U$;                     ▷there is a "total-sub" structure
25:   else if $equal\_set >= 2$ then
26:     add 3 to $U$;                     ▷there is a circular structure
27:   end if
28: end for
29: remove duplicate elements of $U$;
```

3.4.2 Sentiment Flow Analysis Module

This sub-module is to analyze the sentiment changes and the sentiment flow using the reaching definition based on the designed sentiment flow diagram. The reaching definition equation is shown in Section 2.5 *Reaching Definition Analysis*. For ease of understanding, we suppose there are two sentences v_r and v_u in a document and they correspond to two nodes B_r and B_u in the sentiment flow diagram respectively. v_u is positioned after v_r and B_u is after B_r. In the traditional reaching definition, if there are the same variables in B_r and B_u, the assignments to these variables in B_u will replace those in B_r. This process is named "kill" process. But sentiments are different from the above assignments. While people are reading, the sentiments contained in v_u cannot directly replace those in v_r. It should be a slowly changing process and needs to combine two sentences to judge how sentiments change. Therefore, we improve the "kill" process of the traditional reaching definition.

Instead of replacing the assignment directly, we recalculate a new assignment by incorporating assignments of the same variables in B_r and B_u. Generally, key points are gradually described and placed in a later position in a document. Therefore, we set sentiment weights for sentences and suppose that the more backward the position of a sentence is, the greater the sentiment weight of the sentence is. Based on the above assumption, we design three weighted average methods from two perspectives: sentiment neutralization and sentiment superposition.

1) *Sentiment neutralization*

- Static weighted average method. The sentiment weight of v_r is set as α and that of v_u is $(1-\alpha)$. According to the above assumption, $(1-\alpha)$ should be greater than α, so the condition $\alpha < 0.5$ needs to be met. Suppose that there is a same variable t_a in B_r and B_u. Then a new assignment of t_a is,

$$\mathcal{S} = \alpha \times s_r^t + (1-\alpha) \times s_u^t, \tag{6}$$

  where s_r^t and s_u^t represent the assignments of t_a in B_r and B_u, respectively. If sentiments flow to B_u, $\mathcal{S}$ will be assigned to t_a in B_u. This assignment process is the same for B_r.

- Dynamic weighted average method. A fixed weight is set in the static weighted average method, which means that all the sentences which are positioned before v_u have the same impact on the sentiments of v_u. Instead, we design sentiment weights according to the relative positions of sentences. The sentiment weights will change with sentence positions. We set a parameter β, then the sentiment weights are denoted as,

$$w_u = \frac{\beta}{\beta + \beta \times (1-\beta)^{|span|}}, \tag{7}$$

$$w_r = \frac{\beta \times (1-\beta)^{|span|}}{\beta + \beta \times (1-\beta)^{|span|}}, \tag{8}$$

  where, w_r and w_u represent the sentiment weights of v_r and v_u, respectively. $|span|$ represents the distance between two sentences. The new assignment of t_a is,

$$\mathcal{S} = w_u \times s_u^t + w_r \times s_r^t. \tag{9}$$

2) *Sentiment superposition* From the perspective of sentiment neutralization, the new assignment $\mathcal{S}$ of t_a is between the original assignments s_u^t and s_r^t. While people are reading, the intensity of sentiments may improve. So from the perspective of sentiment superposition, we change how the new assignment is calculated, which depends on the positions of sentences.

The sentiment weights w_r and w_u are the same as above. The new assignment is,

$$\mathcal{S} = \begin{cases} s_u^t + w_r \times s_r^t, & \text{if flowing to } B_u, \\ s_r^t + w_u \times s_u^t, & \text{if flowing to } B_r. \end{cases} \tag{10}$$

If sentiments flow to B_u, $\mathcal{S}$ calculated by the first equation will be assigned to t_a in B_u. If sentiments flow to B_r, the second equation will be used to calculate $\mathcal{S}$ to assign t_a.

We mainly focus on sentiments while using the improved reaching definition to analyze sentiments. In addition to sentiments, we have also taken information at different levels and context information into consideration. Specifically, word sentiments are calculated by a sentiment dictionary at word level. Topics are selected and weighted at topic level. Sentences are regarded as nodes in the sentiment flow diagram to analyze sentiment changes between sentences at sentence level. The sentiment flow diagram designed based on the document structure is used to analyze sentiment flow across the whole document at document level. Besides, the context information of sentiments is retained adequately after we leverage different weighted average methods to improve the "kill" process of the traditional reaching definition.

Algorithm 2. Topic Sentiment Summarization

```
Require: Sentences $V = \{v_1, v_2, \ldots, v_m\}$; node $B = \{B_1, B_2, \ldots, B_m\}$; topic weight $T$; topic-word distribution $W$; definition generated in node $B^{gen}$; definition killed in node $B^{kill}$; output definition of node $OUT[B]$; input definition of node $IN[B]$;
Ensure:topic sentiment summary $S'$.
for $v_i \in V$ do
    $T, W \leftarrow LDA(v_i)$;
    $B \leftarrow calculateTopicSentiment(T, W)$;
end for
for $B_i \in B$ do
    $B^{gen} \leftarrow calculateDefinitionGenerated(B_i, B)$;
    $B^{kill} \leftarrow calculateDefinitionKilled(B_i, B)$;
    $OUT[B_i] \leftarrow \emptyset$;
end for
while ($OUT[B]$ changes) do
    for $B_i \in B$ do
        $IN[B_i] \leftarrow \cup_{P \text{is one of} B_i's\ precursors} OUT[P]$;
        $OUT[B_i] \leftarrow B_i^{gen} \cup (IN[B_i] - B_i^{kill})$;
    end for
end while
$S' \leftarrow$ output definition of the last node;
```

3.5 Summary Generation Module

In the TSSRD framework, the summary generation module is used to generate final topic sentiment summaries. When sentiments in each node reach steady states of the reaching definition, i.e., assignments of variables no longer change, we take the output of the last node as the final topic sentiment summary. From the perspective of *readers*, the sentiment flow diagram is designed as a single-path structure, where there is only one last node. The output of the last node is the final summary. From the perspective of *writers*, the diagram is designed as a multi-path structure, where all the branch paths have the last nodes. We weight the outputs of the last nodes in all branch paths to obtain the final summary in terms of their depth in the paths. The final summary is denoted as,

$$Summary = \frac{\sum_{i \in I_l} h_i s_i^o}{\sum_{i \in I_l} h_i}, \tag{11}$$

where, I_l represents the indices of the last nodes in all branch paths, h is the depth, and s^o is the output sentiment of the last node. The whole process of generating topic sentiment summaries by the reaching definition is shown in Algorithm 2. After obtaining summaries, an extrinsic evaluation method is introduced to evaluate the TSSRD and baselines. In this method, we use summaries to train a sentiment classifier such as SVM. Performance of summarization methods is evaluated by classification results of the classifiers.

3.6 Case Study

To further clarify and explain the proposed framework, we select a piece of movie review from IMDb and conduct the summarization as a case study. The selected review is labeled negative. There are 4 sentences, i.e.,

1. *A mean spirited, repulsive horror film about 3 murderous children.*
2. *Susan Strasberg is totally wasted in a 5-minute cameo, even though she receives star billing.*
3. *If your a Julie Brown fan, you'll want to check it out, since she's naked in a couple of shots.*
4. *All others, avoid.*

First, we use a trained LDA model to obtain the topic distribution and topic-word distribution of each sentence. Next, the sentiment embedding of each topic is calculated. The results are,

1. **topic2**$(\ 0.000,\ 0.000,\ 0.000,\ 0.000,\ 0.000)$
   topic3$(\ 0.078,\ 0.090,\ 0.066,\ 0.000,\ 0.000)$
   topic8$(-0.067, -0.103,\ 0.000,\ 0.000, -0.102)$
2. **topic4**$(-0.246, -0.215,\ 0.000,\ 0.000, -0.273)$
3. **topic2**$(\ 0.025,\ 0.082, -0.100, -0.017,\ 0.111)$
   topic4$(\ 0.000,\ 0.000,\ 0.000,\ 0.000,\ 0.000)$
4. **topic2**$(-0.119,\ 0.000,\ 0.000,\ 0.000, -0.119)$

Then, a single-path diagram is built to explore the sentiment changes and sentiment flow according to the reaching definition analysis module. Finally, the output of the last node is the topic sentiment summary.

The whole process is similar to the sentiment changes while people are reading. There is more than one sentence containing the sentiments of topic 2 and 4, so the sentiment changes are mainly under these two topics, which is shown in Figs. 3 and 4. Each reading step represents the output of one node. The final value of each sentiment dimension under topic 4 is negative. Except for "Introspection" and "Sensitivity," the final value of other dimensions under topic 2 are also negative. The sentiment of the whole document is negative, which is consistent with its negative label. The third sentence contains positive sentiments under topic 2, so there are two positive values. This case study demonstrates that our framework can summarize the existing sentiments to a great extent.

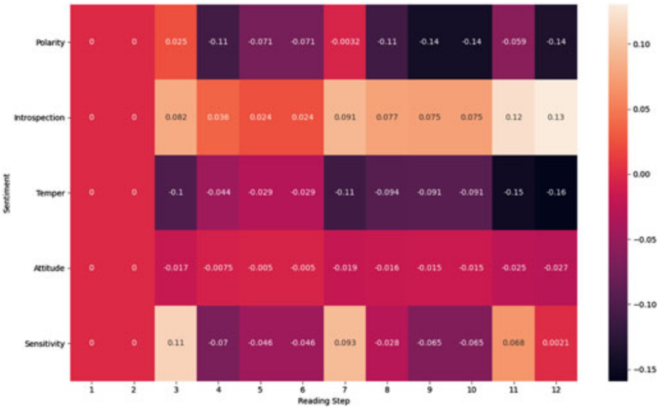


Fig. 3. Sentiment changes of topic 2.

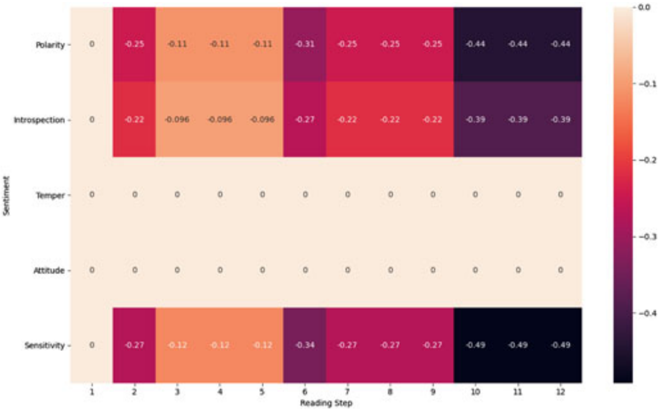


Fig. 4. Sentiment changes of topic 4.

4 Experiments and Discussions

This section introduces related experiments of the TSSRD framework and discusses some factors that have an impact on the framework. An extrinsic evaluation method is used to calculate the summarization framework quality score. We carry out experiments to evaluate the TSSRD and compare it with several baselines.

4.1 Datasets

IMDb dataset consists of 50,000 reviews about movies. The reviews in the corpus are divided into positive and negative parts according to the polarity of sentiments with 25,000 reviews respectively.

Amazon Electronics dataset consists of 1,689,188 reviews about commodities. The reviews are rated on a scale from 1 to 5. Among them, there are 108,725 reviews rated 1, 82,139 reviews rated 2, 142,257 reviews rated 3, 347,041 reviews rated 4 and 1,009,026 reviews rated 5. To make the reviews among different ratings balanced, we categorize those with ratings of 1, 2 and 3 as negative reviews and those with ratings of 4 and 5 as positive reviews. We randomly select 25,000 reviews from two polarities respectively for experiments. Specifically, according to the proportion of reviews with different ratings, we randomly selected 8,160 reviews rated 1, 6,164 reviews rated 2, 10676 reviews rated 3, 6398 reviews rated 4 and 18,602 reviews rated 5.

4.2 Experimental Setup

4.2.1 *Topic Sentiment Summarization*

We set up the topic sentiment summarization experiment by following steps:

1) Each document is segmented into sentences. Each sentence is segmented into words. Each word is restored as its original form and stop words are removed.
2) The corpus is divided into a training set U_{train} and a test set U_{test}, in which there are 12,500 positive reviews and negative reviews respectively. U_{train} is used to train a LDA topic model and U_{test} is used to generate topic sentiment summaries.
3) The trained topic model is used to calculate sentence-topic distribution and topic-word distribution for each sentence of documents in U_{test}. To filter unimportant topics, we introduce a topic sparsification parameter θ and set it as $\theta = 0.1$. Topics whose weights are greater than θ are kept and those whose weights are less than θ are ignored.
4) According to the topic-word distribution, SenticNet 6 [53] is used to calculate sentiment embeddings of the words under the reserved topics. SenticNet 6 contains five sentiment dimensions, namely sentiment polarity value and four fine-grained sentiments, introspection, temper, attitude, and sensitivity.
5) The α and β are regarded as weighted average parameters of the weighted average methods. We set seven values for these parameters and select one that makes the framework perform the best. In the best framework, the improved reaching definition is used to analyze the sentiment changes and the sentiment flow. The output of the last node in the steady state is taken as the final topic sentiment summary.

According to different structures of sentiment flow diagrams and different weighted average methods in the reaching definition analysis module, we design four variant frameworks of the TSSRD. There are five summarization frameworks in total and their differences are shown in Table 2.

4.2.2 *Evaluation*

In the framework, we make use of the improved reaching definition to analyze sentiments based on topic sentiments calculated by SenticNet 6 and get topic sentiment summaries of documents according to sentiments in the steady state. After obtaining summaries, we use an extrinsic evaluation method to evaluate the TSSRD framework and its four variants. First, we divide the topic sentiment summary set U^a into a training set U^a_{train} and a test set U^a_{test} according to the ratio of 8:2. The summary set U^a is obtained by passing U_{test} through the summarization framework. Then, we use the training set U^a_{train} to train a SVM classifier. During the training, we use a parameter grid search method to select the best key parameters of SVM. The radial basis function (RBF) is used as the kernel function. The search space of penalty parameter C is $\{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and kernel function parameter γ is $\{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$. To evaluate the performance of selected parameters, we conduct 5-fold

TABLE 2
Differences Among Five Proposed Summarization Frameworks

Frameworks	Diagram structure	Reaching definition	Weighted average method
TSSRD	single-path	traditional	-
TSSRD-*fix-*	single-path	improved	static
TSSRD-*index-*	single-path	improved	dynamic
TSSRD-*index+*	single-path	improved	superposition
TSSRD*-*index+*	multi-path	improved	superposition

cross-validation experiments and repeat 10 times. Finally, we use the trained SVM classifier to test U^a_{test} and calculate the accuracy of classification by the following equation,

$$accuracy = \frac{n_- + n_+}{N}, \tag{12}$$

where n_- denotes the number of samples whose sentiment is negative and the classifier's sentiment classification result is also negative, n_+ denotes the number of samples whose sentiment is positive and the result of sentiment classification is also positive, and N denotes the total number of samples in U^a_{test}. We use the accuracy of the classifier to evaluate the performance of the summarization frameworks.

To further prove the effectiveness and superiority of the frameworks, we compare them with eight baselines, i.e., Lead,[3] TextRank [18], SPSR [19], GbTM [17], PsoSA [3], E-Summ [16], LFIP-SUM [20], and SLS [54]. We set the summary length to 3, which means there are three sentences in each summary. After obtaining summaries, we also use the LDA topic model and SenticNet 6 to calculate sentiment embeddings, and train a SVM classifier to evaluate the summarization methods.

4.3 Results

We carry out topic sentiment summarization experiments on IMDb and Amazon Electronics. An extrinsic evaluation method is used to evaluate the performance of our proposed frameworks, which are compared with several baselines.

To explore the most suitable values of weighted average parameters (i.e., α and β) in different weighted average methods, we set seven values from 0.1 to 0.7 with step size 0.1. The TSSRD-*fix-*, TSSRD-*index-*, and TSSRD-*index+* are used for summarization and evaluation experiments. We also conduct 5-fold cross-validation experiments 10 times. The experimental results are shown in Figs. 5 and 6, illustrating the performance of the three frameworks when the weighted average parameters are set to different values on IMDb and Amazon. On IMDb, the best values of weighted average parameters in the TSSRD-*fix-*, TSSRD-*index-* and TSSRD-*index+* are 0.5, 0.1 and 0.2, respectively; on Amazon, the best values are 0.5, 0.3 and 0.2, respectively.

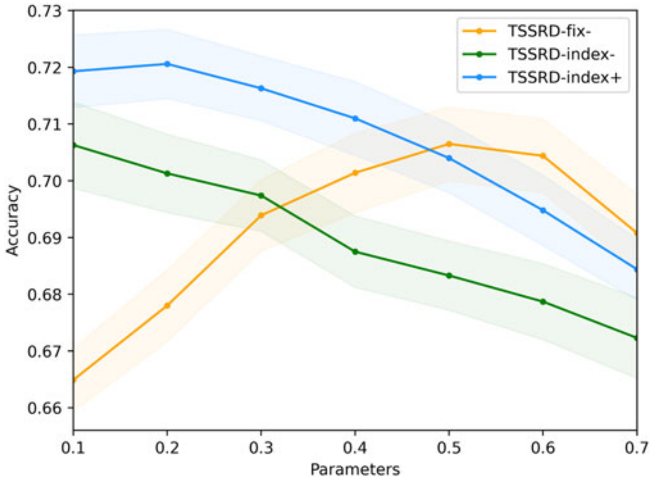


Fig. 5. Performance of frameworks under different weighted average parameters on IMDb.

After getting the best values of weighted average parameters and obtaining the best performance of the TSSRD-*fix-*, TSSRD-*index-*, and TSSRD-*index+*, we further use the TSSRD for the same experiments. Then all these four frameworks are compared with the baselines. The cross-validation results are shown in Table 3. The bold numbers represent the best performance among the frameworks on the same dataset, and the underlined numbers represent the second-best performance. The experimental results demonstrate that the TSSRD-*index+* has the best performance on IMDb and Amazon. To verify whether the performance of the TSSRD-*index+* is statistically significantly better than those of other frameworks, we compare the cross-validation results of the TSSRD-*index+* with those of other frameworks by *t*-test, whose results are shown in Table 4. The *t*-test results show that the TSSRD-*index+* is statistically significantly better than the TSSRD, TSSRD-*fix-* and TSSRD-*index-*, and also better than the baselines.

Table 5 shows the results of the summarization frameworks on the test set. They demonstrate that the performance of the TSSRD-*index+* is also better than other frameworks on IMDb and Amazon. The TSSRD-*index+* is at least 2.32% better than the baselines.

The above experiments show that the TSSRD-*index+* is the best among the TSSRD and its first three variants, and is also better than the baselines. The difference between the TSSRD*-*index+* and TSSRD-*index+* is the sentiment flow diagram. If a document itself does not have any special structures, the sentiment flow diagrams constructed by the two

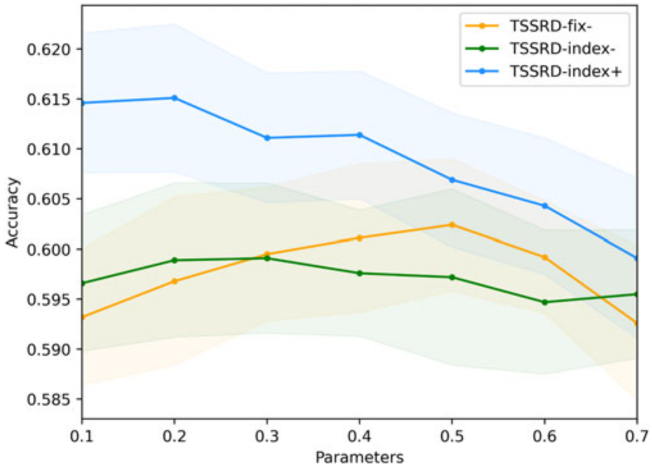


Fig. 6. Performance of frameworks under different weighted average parameters on Amazon.

[3] The first k sentences of a document.

TABLE 3
Cross-Validation Results of Summarization Frameworks

Frameworks	IMDb		Amazon	
	avg.	std.	avg.	std.
Lead	0.6653	0.0070	0.5780	0.0064
TextRank	0.6738	0.0073	0.5807	0.0083
GbTM	0.7065	0.0234	0.5960	0.0144
PsoSA	0.6549	0.0082	0.5761	0.0065
SPSR	0.6625	0.0065	0.5934	0.0074
E-Summ	0.6431	0.0063	0.5702	0.0100
LFIP-SUM	0.6106	0.0056	0.5826	0.0072
SLS	0.6861	0.0062	0.5953	0.0069
TSSRD	0.6344	0.0068	0.5862	0.0077
TSSRD-*fix-*	0.7065	0.0065	0.6024	0.0066
TSSRD-*index-*	0.7063	0.0076	0.5991	0.0075
TSSRD-*index+*	**0.7206**	0.0061	**0.6151**	0.0074

TABLE 4
T-Test for Comparing TSSRD-*Index+* With Other Summarization Frameworks

Frameworks	IMDb	Amazon
Lead	3.6411E-42	1.0261E-31
TextRank	7.0935E-37	2.2243E-28
GbTM	6.5021E-05	7.5399E-14
PsoSA	5.0640E-42	7.8166E-34
SPSR	5.5334E-41	2.2511E-19
E-Summ	6.1207E-50	2.8434E-27
LFIP-SUM	1.5575E-55	4.7584E-26
SLS	1.4275E-31	3.6263E-18
TSSRD	2.9163E-51	1.1387E-24
TSSRD-*fix-*	4.4513E-14	7.6840E-12
TSSRD-*index-*	1.4756E-14	2.9410E-15

frameworks are the same. Therefore, while analyzing the sentiment flow path of a document, we judge whether the document contains special structures according to the document structure analysis algorithm. If not, this document will be filtered out and not participate in the subsequent experiments. The amount of original samples and selected samples is shown in Table 8. The amount of selected samples is fewer than the original samples, but it is still sufficient for training SVMs according to [55], [56]. The experimental setups of the TSSRD*-*index+* and TSSRD-*index* + are consistent. We set the weighted average parameter to 0.2, and use the two frameworks to conduct summarization and evaluation experiments. The cross-validation results are shown in Tables 6 and 7 shows the results on the test set. They demonstrate that on IMDb, the TSSRD*-*index+* is better than the TSSRD-*index+*. On Amazon, the TSSRD*-*index+* is also better on the test set although it performs a little poorly from the cross-validation results, which means that the TSSRD*-*index+* has better generalization.

TABLE 5
Results of Test Set

Frameworks	IMDb	Amazon
Lead	0.6552	0.5807
TextRank	0.6584	0.5793
GbTM	0.6694	0.5807
PsoSA	0.6526	0.5802
SPSR	0.6737	0.5958
E-Summ	0.6228	0.5787
LFIP-SUM	0.6063	0.5802
SLS	0.6786	0.5948
TSSRD	0.6404	0.5856
TSSRD-*fix-*	0.6872	0.5976
TSSRD-*index-*	0.7016	0.6142
TSSRD-*index+*	**0.7272**	**0.6190**

TABLE 6
Cross-Validation Results of TSSRD-*Index+* and TSSRD*-*Index+*

Frameworks	IMDb		Amazon	
	avg.	std.	avg.	std.
TSSRD-*index+*	0.7187	0.0082	**0.6209**	0.0161
TSSRD*-*index+*	**0.7194**	0.0085	0.6194	0.0135

TABLE 7
Results of TSSRD-*Index+* and TSSRD*-*Index+* on Test Set

Frameworks	IMDb	Amazon
TSSRD-*index+*	0.7201	0.6153
TSSRD*-*index+*	**0.7218**	**0.6186**

4.4 Results Analysis and Discussion

4.4.1 Topic Model

One important part of the frameworks is the LDA topic model. To explore the influence of different topic models on the frameworks, we replace the LDA in the TSSRD-*index+* with neural topic models NQTM [8] and CRNTM [9]. Other setups are the same. We carry out 5-fold cross-validation experiments 10 times and compare the results of the LDA with two neural topic models by *t*-test, whose results are shown in Tables 9 and 10. Table 11 shows the results on the test set. All the results demonstrate that the framework with LDA is statistically significantly better than those with two neural topic models on IMDb. For Amazon, the framework with NQTM is the best.

The NQTM adopts a new topic distribution quantization approach and negative sampling decoder to model texts, which makes it suitable for mining latent topics of shorter documents. In IMDb, each document contains 11 sentences and each sentence contains 26 words on average. In Amazon, each document contains 7 sentences and each sentence contains 22 words on average. The documents are shorter in Amazon. Therefore, the NQTM has better performance on Amazon. On IMDb with longer documents, the traditional LDA is better.

4.4.2 Topic Sparsification Parameters

To explore the influence of topic sparsification parameters on the performance of the summarization frameworks, we set values of the topic sparsification parameter from 0.1 to 0.5, and the step size is 0.1. The topic number is 10. The TSSRD is used for experiments. The experimental results are shown in Figs. 7 and 8. They demonstrate that on IMDb, when the topic sparsification parameter is 0.4, the performance of the

TABLE 8
Amount of Original Samples and Selected Samples

Datasets	Original samples		Selected samples	
	pos	neg	pos	neg
IMDb	12,500	12,500	7,342	7,845
Amazon	12,500	12,500	3,058	4,126

TABLE 9
Cross-Validation Results of the TSSRD-*Index+* Framework With Different Topic Models

Topic models	IMDb		Amazon	
	avg.	std.	avg.	std.
CRNTM	0.6863	0.0082	0.5509	0.0108
NQTM	0.6991	0.0067	**0.6356**	0.0074
LDA	**0.7206**	0.0061	0.6151	0.0074

TABLE 10
T-Test for Comparing LDA With CRNTM and NQTM

Topic models	IMDb	Amazon
CRNTM	1.5176E-29	6.4908E-37
NQTM	5.3240E-20	1.7200E-18

TABLE 11
Results of the TSSRD-*Index+* Framework With Different Topic Models on Test Set

Topic models	IMDb	Amazon
CRNTM	0.6880	0.5437
NQTM	0.7028	**0.6336**
LDA	**0.7272**	0.6190

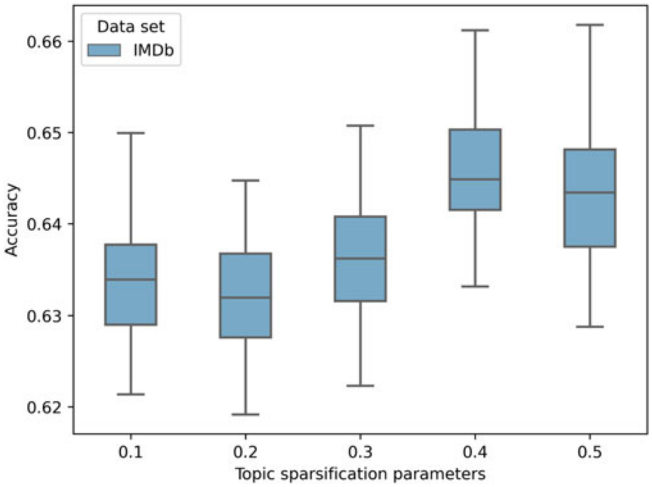


Fig. 7. Results of topic sparsification parameters exploration on IMDb.

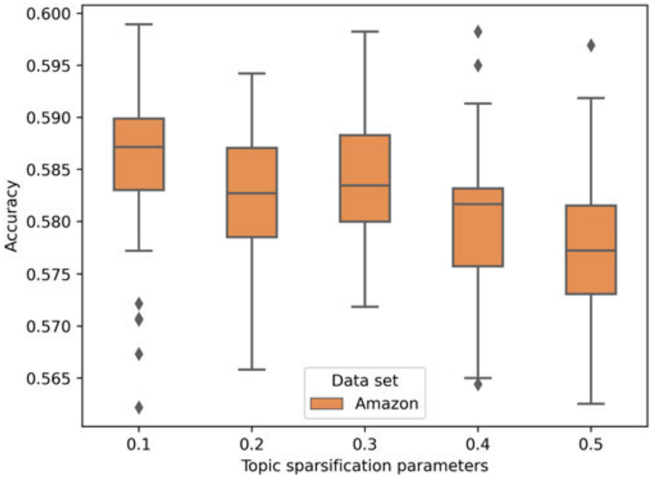


Fig. 8. Results of topic sparsification parameters exploration on Amazon.

framework is the best. When the parameter is 0.5, the performance is second-best. On Amazon, when the topic sparsification parameter is 0.1, the performance is the best. When the parameter is 0.3, the performance is second-best.

The values of topic sparsification parameters that make the best and second-best performance of the summarization framework are different on different corpora, which may be due to the different topic distributions of documents. On IMDb, the topic distribution is relatively concentrated, and the weight of important topics is relatively great. On Amazon, the topic distribution is scattered, and the weight of important topics is small, which may be only a little greater than that of unimportant topics. Therefore, on IMDb, the topic sparsification parameter that makes the summarization framework perform best will be greater; on Amazon, the topic sparsification parameter will be smaller.

4.4.3 Topic Number

To explore the influence of topic number, we set the topic number from 3 to 17 respectively, and the step size is 2. The topic sparsification parameter is set to 0.1. The TSSRD-*index+* is used for experiments and the weighted average parameter is set to 0.2. The experimental results are shown in Figs. 9 and 10. They demonstrate that on IMDb, when the topic number is 13, the performance of the framework is the best. When the number is 10, the performance is second-best. On Amazon, when the topic number is 15, the performance is the best. When the number is 13, the performance is second-best.

The topic numbers that make the best and second-best performance of the summarization framework vary with different corpora. Maybe it is because the actual number of topics involved in the different corpora is different. When the topic number set is less than the actual number, the actual important topics and unimportant topics may be divided into the same topic due to the limitation of the topic number, resulting in impurities in the topic and affecting the generation of summaries. When the topic number set is greater than the actual number, the actual important topics may be overdivided and divided into different topics. The information of important topics cannot be fully utilized to generate summaries. The actual number of topics involved in IMDb is probably 13 while the number in Amazon may be 15.

4.4.4 Document Structures

To explore the influence of document structures, we design two groups of experiments. The difference between them

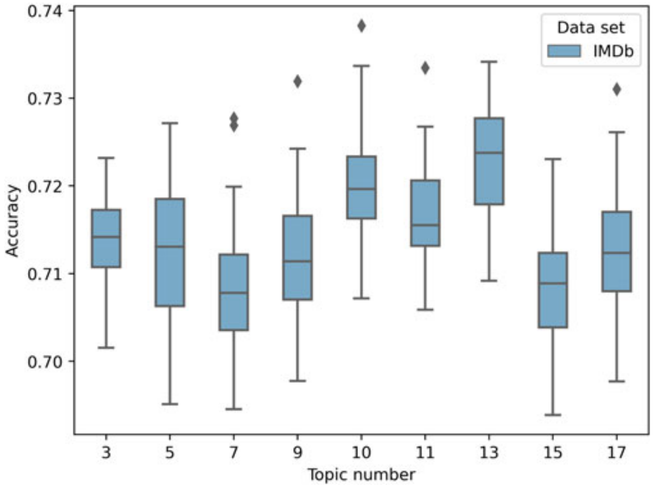


Fig. 9. Results of topic number exploration on IMDb.

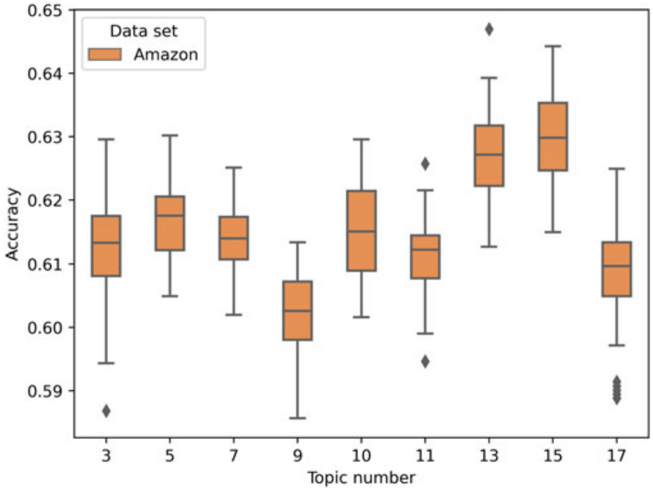


Fig. 10. Results of topic number exploration on Amazon.

TABLE 12
Cross-Validation Results on the Original Samples

Frameworks	IMDb		Amazon	
	avg.	std.	avg.	std.
TSSRD-*index+*	0.7206	0.0061	**0.6151**	0.0074
TSSRD*-*index+*	**0.7218**	0.0063	0.6148	0.0063

TABLE 13
Results of Test Set on the Original Samples

Frameworks	IMDb	Amazon
TSSRD-*index+*	**0.7272**	**0.6190**
TSSRD*-*index+*	0.7266	0.6155

TABLE 14
Cross-Validation Results on Further Processed Samples

Frameworks	IMDb		Amazon	
	avg.	std.	avg.	std.
TSSRD-*index+*	0.7074	0.0112	0.6075	0.0179
TSSRD*-*index+*	**0.7103**	0.0101	**0.6208**	0.0150

TABLE 15
Results of Test Set on Further Processed Samples

Frameworks	IMDb	Amazon
TSSRD-*index+*	0.7090	0.6092
TSSRD*-*index+*	**0.7111**	**0.6163**

lies in experimental samples. In the first group of experiments, we do not preprocess the samples, and carry out summarization and evaluation experiments using the TSSRD*-*index+* and the TSSRD-*index+* on the original samples, whose results are shown in Tables 12 and 13. In the second group of experiments, we only retain documents with special structures and more than 10 sentences, on which summarization and evaluation experiments are conducted. The results are shown in Tables 14 and 15. To avoid occasionality, we sample the test set randomly 10 times. The average of 10 results is taken as the final result. The experimental results show that when the samples are not preprocessed, the TSSRD-*index+* is better. After further processing the samples, the TSSRD*-*index+* is obviously better than the TSSRD-*index+*. It means that the TSSRD*-*index+* is more suitable for topic sentiment summarization on longer documents with special structures.

The reason for the above experimental results may be the different characteristics of datasets. Both IMDb and Amazon are reviews from users. They are very different from formal articles, especially for document structures. The reviews written by users usually have no special structures. The formal articles are works after authors' careful consideration, and they often need to rely on rigorous structures to explain points or plots. Therefore, there may be only a small number of documents that really contain special structures on both datasets. When the samples are not preprocessed, most documents without special structures have a negative impact on the TSSRD*-*index+*. After filtering the samples without special structures, the advantages of the TSSRD*-*index+* framework are revealed.

4.4.5 Borderline Reviews on Amazon

On Amazon, the reviews are rated on a scale from 1 to 5. We have categorized reviews with ratings of 1, 2 and 3 as negative reviews and those with ratings of 4 and 5 as positive reviews. Such a categorization method makes full use of the dataset, but the boundary between two polarities is vague, which leads to that the reviews categorized as positive may be negative or the reviews categorized as negative may be positive. Assigning borderline reviews to a definite polarity may affect sentiment classification. To explore the influence of borderline reviews on experimental results, we take reviews with ratings of 1 and 2 as negative reviews and those with ratings of 4 and 5 as positive reviews, ignoring those with ratings of 3. We resample reviews from Amazon and conduct summarization and evaluation experiments using the proposed frameworks. 5-fold cross-validation and *t*-test are used to compare the performances of the frameworks on the resampled samples with those on the original samples, and results are shown in Tables 16 and 17. Table 18

TABLE 16
Cross-Validation Results on the Original Samples and the Resampled Samples

Frameworks	Original samples		Resampled samples	
	avg.	std.	avg.	std.
TSSRD	0.5862	0.0077	**0.6101**	0.0072
TSSRD-*fix-*	0.6024	0.0066	**0.6417**	0.0073
TSSRD-*index-*	0.5991	0.0075	**0.6359**	0.0080
TSSRD-*index+*	0.6151	0.0074	**0.6598**	0.0067

shows the results on the test set. All these results demonstrate that classification of sentiment polarity is statistically significantly better on the resampled samples. It illustrates that borderline reviews exist in the original experimental samples and they have a negative influence on sentiment classification. It is better to exclude these reviews.

5 Conclusion and Future Work

In this paper, we propose a topic sentiment summarization framework based on reaching definition (TSSRD). The framework concentrates more on the sentiment changes and the sentiment flow ignored by existing summarization methods. To better incorporate the above two aspects into summarization, we introduce the reaching definition compiler. Based on the difference between sentiment changes in documents and assignments of variables in programs, the "kill" process of the traditional reaching definition is further improved to make it more suitable for sentiment analysis. Specifically, we propose three weighted average methods from the perspectives of sentiment neutralization and sentiment superposition to change the assignments of variables while sentiments are flowing. In addition, we also propose to analyze the document structures from the perspectives of readers and writers to reveal the change regularities and flow paths of sentiments in documents. In the evaluation of summarization frameworks, an extrinsic evaluation method is introduced. The experimental results demonstrate that: 1) under the same conditions, the TSSRD-*index+* framework has the best performance, which is at least 2.32% higher than the baselines; 2) the TSSRD*-*index+* framework is better than TSSRD-*index+* while summarizing topic sentiments on longer documents with special structures; 3) the proposed summarization framework is sensitive to topic model, topic number and topic sparsification parameters. In other words, the framework will have better performance when the parameters are set to appropriate values.

In future work, we will explore more factors that could influence the framework performance, and further analyze their internal relationships to get the best combination of parameters. We also plan to design a unified framework that can summarize sentiments on different kinds of documents, which will help machines better understand natural language from the sentiment level.

TABLE 17
T-Test for Comparing Results on the Resampled Samples With Those on the Original Samples

Frameworks	*p*-value
TSSRD	8.5648E-22
TSSRD-*fix-*	5.0984E-30
TSSRD-*index-*	1.9012E-30
TSSRD-*index+*	1.2213E-33

TABLE 18
Results on Test Sets of the Original Samples and the Resampled Samples

Frameworks	Original samples	Resampled samples
TSSRD	0.5856	**0.5995**
TSSRD-*fix-*	0.5976	**0.6187**
TSSRD-*index-*	0.6142	**0.6227**
TSSRD-*index+*	0.6190	**0.6320**

References

[1] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, "Affective computing and sentiment analysis," in *A Practical Guide to Sentiment Analysis*, Berlin, Germany: Springer, 2017, pp. 1–10.

[2] N. Akhtar, N. Zubair, A. Kumar, and T. Ahmad, "Aspect based sentiment oriented summarization of hotel reviews," *Procedia Comput. Sci.*, vol. 115, pp. 563–571, 2017.

[3] S. Mandal, G. K. Singh, and A. Pal, "PSO-based text summarization approach using sentiment analysis," in *Proc. Comput. Commun. Signal Process.*, 2019, pp. 845–854.

[4] A. Nenkova and K. McKeown, "A survey of text summarization techniques," in *Mining Text Data*, Berlin, Germany: Springer, 2012, pp. 43–76.

[5] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques and Tools*, Noida, India: Pearson Education India, 2007.

[6] H. P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Develop.*, vol. 2, no. 2, pp. 159–165, 1958.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[8] X. Wu, C. Li, Y. Zhu, and Y. Miao, "Short text topic modeling with topic distribution quantization and negative sampling decoder," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2020, pp. 1772–1782.

[9] J. Feng, Z. Zhang, C. Ding, Y. Rao, and H. Xie, "Context reinforced neural topic modeling over short texts," 2020, *arXiv:2008.04545*.

[10] R. K. Roul, S. Mehrotra, Y. Pungaliya, and J. K. Sahoo, "A new automatic multi-document text summarization using topic modeling," in *Proc. Int. Conf. Distrib. Comput. Internet Technol.*, 2019, pp. 212–221.

[11] R. K. Roul, "Topic modeling combined with classification technique for extractive multi-document text summarization," *Soft Comput.*, vol. 25, no. 2, pp. 1113–1127, 2021.

[12] Z. Wu *et al.*, "A topic modeling based approach to novel document automatic summarization," *Expert Syst. Appl.*, vol. 84, pp. 12–23, 2017.

[13] N. Gialitsis, N. Pittaras, and P. Stamatopoulos, "A topic-based sentence representation for extractive text summarization," in *Proc. Workshop MultiLing Summarization Across Lang., Genres Sources*, 2019, pp. 26–34.

[14] R. Rani and D. Lobiyal, "An extractive text summarization approach using tagged-LDA based topic modeling," *Multimedia Tools Appl.*, vol. 80, no. 3, pp. 3275–3305, 2021.

[15] B. Xu, H. Lin, H. Hao, Z. Yang, J. Wang, and S. Zhang, "Generating user-oriented text summarization based on social networks using topic models," in *Proc. Chin. Nat. Conf. Soc. Media Process.*, 2016, pp. 186–193.

[16] A. Khurana and V. Bhatnagar, "Investigating entropy for extractive document summarization," *Expert Syst. Appl.*, vol. 187, 2021, Art. no. 115820.

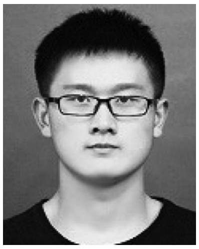
[17] R. C. Belwal, S. Rai, and A. Gupta, "A new graph-based extractive text summarization using keywords or topic modeling," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 10, pp. 8975–8990, 2021.

[18] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2004, pp. 404–411.

[19] X. Li, S. Zhu, H. Xie, and Q. Li, "Document summarization via self-present sentence relevance model," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2013, pp. 309–323.

[20] M. Jang and P. Kang, "Learning-free unsupervised extractive summarization model," *IEEE Access*, vol. 9, pp. 14358–14368, 2021.
[21] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowl.-Based Syst.*, vol. 226, 2021, Art. no. 107134.
[22] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Gener. Comput. Syst.*, vol. 115, pp. 279–294, 2021.
[23] W. Li, W. Shao, S. Ji, and E. Cambria, "BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis," *Neurocomputing*, vol. 467, pp. 73–82, 2022.
[24] C. Gan, Q. Feng, and Z. Zhang, "Scalable multi-channel dilated CNN–BiLSTM model with attention mechanism for chinese textual sentiment analysis," *Future Gener. Comput. Syst.*, vol. 118, pp. 297–309, 2021.
[25] W. Song, Z. Wen, Z. Xiao, and S. C. Park, "Semantics perception and refinement network for aspect-based sentiment analysis," *Knowl.-Based Syst.*, vol. 214, 2021, Art. no. 106755.
[26] F. Huang, X. Li, C. Yuan, S. Zhang, J. Zhang, and S. Qiao, "Attention-emotion-enhanced convolutional LSTM for sentiment analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2021.3056664.
[27] H. Zhao, Z. Liu, X. Yao, and Q. Yang, "A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach," *Inf. Process. Manage.*, vol. 58, no. 5, 2021, Art. no. 102656.
[28] B. Ray, A. Garain, and R. Sarkar, "An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews," *Appl. Soft Comput.*, vol. 98, 2021, Art. no. 106935.
[29] A. Zhao and Y. Yu, "Knowledge-enabled bert for aspect-based sentiment analysis," *Knowl.-Based Syst.*, vol. 227, 2021, Art. no. 107220.
[30] S. Vashishtha and S. Susan, "Highlighting keyphrases using senti-scoring and fuzzy entropy for unsupervised sentiment analysis," *Expert Syst. Appl.*, vol. 169, 2021, Art. no. 114323.
[31] E. Zangerle, C.-M. Chen, M.-F. Tsai, and Y.-H. Yang, "Leveraging affective hashtags for ranking music recommendations," *IEEE Trans. Affective Comput.*, vol. 12, no. 1, pp. 78–91, Jan.–Mar. 2021.
[32] N. Chatterjee, G. Jain, and G. S. Bajwa, "Single document extractive text summarization using neural networks and genetic algorithm," in *Proc. Sci. Inf. Conf.*, 2018, pp. 338–358.
[33] V. Basile, N. Novielli, D. Croce, F. Barbieri, M. Nissim, and V. Patti, "Sentiment polarity classification at EVALITA: Lessons learned and open challenges," *IEEE Trans. Affective Comput.*, vol. 12, no. 2, pp. 466–478, Apr.–Jun. 2021.
[34] A. Abdi, S. M. Shamsuddin, S. Hasan, and J. Piran, "Automatic sentiment-oriented summarization of multi-documents using soft computing," *Soft Comput.*, vol. 23, no. 20, pp. 10551–10568, 2019.
[35] S. M. Ali, Z. Noorian, E. Bagheri, C. Ding, and F. Al-Obeidat, "Topic and sentiment aware microblog summarization for Twitter," *J. Intell. Inf. Syst.*, vol. 54, no. 1, pp. 129–156, 2020.
[36] I. Spasić, L. Williams, and A. Buerki, "Idiom-based features in sentiment analysis: Cutting the gordian knot," *IEEE Trans. Affective Comput.*, vol. 11, no. 2, pp. 189–199, Apr.–Jun. 2020.
[37] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Pipelined neural networks for phrase-level sentiment intensity prediction," *IEEE Trans. Affective Comput.*, vol. 11, no. 3, pp. 447–458, Jul.–Sep. 2020.
[38] A. Bompotas, A. Ilias, A. Kanavos, C. Makris, G. Rompolas, and A. Savvopoulos, "A sentiment-based hotel review summarization using machine learning techniques," in *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innovations*, 2020, pp. 155–164.
[39] H. Li, Y. Wang, X. Mou, and Q. Peng, "Sentiment classification of financial microblogs through automatic text summarization," in *Proc. Chin. Automat. Congr.*, 2020, pp. 5579–5584.
[40] X. Wang, L. Kou, V. Sugumaran, X. Luo, and H. Zhang, "Emotion correlation mining through deep learning models on natural language text," *IEEE Trans. Cybern.*, vol. 51, no. 9, pp. 4400–4413, Sep. 2021.
[41] C. Lin, Y. He, R. Everson, and S. Ruger, "Weakly supervised joint sentiment-topic detection from text," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 1134–1145, Jun. 2012.
[42] M. Dermouche, L. Kouas, J. Velcin, and S. Loudcher, "A joint model for topic-sentiment modeling from text," in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, 2015, pp. 819–824.
[43] L.-Y. Dong *et al.*, "An unsupervised topic-sentiment joint probabilistic model for detecting deceptive reviews," *Expert Syst. Appl.*, vol. 114, pp. 210–223, 2018.
[44] A. Alamsyah, W. Rizkika, D. D. A. Nugroho, F. Renaldi, and S. Saadah, "Dynamic large scale data on twitter using sentiment analysis and topic modeling," in *Proc. 6th Int. Conf. Inf. Commun. Technol.*, 2018, pp. 254–258.
[45] B. Jeong, J. Yoon, and J.-M. Lee, "Social media mining for product planning: A product opportunity mining approach based on topic modeling and sentiment analysis," *Int. J. Inf. Manage.*, vol. 48, pp. 280–290, 2019.
[46] S. Xiong, K. Wang, D. Ji, and B. Wang, "A short text sentiment-topic model for product reviews," *Neurocomputing*, vol. 297, pp. 94–102, 2018.
[47] X. Fu, X. Sun, H. Wu, L. Cui, and J. Z. Huang, "Weakly supervised topic sentiment joint model with word embeddings," *Knowl.-Based Syst.*, vol. 147, pp. 43–54, 2018.
[48] F. Huang, C. Yuan, Y. Bi, and J. Lu, "Exploiting long-term dependency for topic sentiment analysis," *IEEE Access*, vol. 8, pp. 221963–221974, 2020.
[49] F. Huang, S. Zhang, J. Zhang, and G. Yu, "Multimodal learning for topic sentiment analysis in microblogging," *Neurocomputing*, vol. 253, pp. 144–153, 2017.
[50] M. M. Rahman and H. Wang, "Hidden topic sentiment model," in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 155–165.
[51] A. N. Masud and F. Ciccozzi, "More precise construction of static single assignment programs using reaching definitions," *J. Syst. Softw.*, vol. 166, 2020, Art. no. 110590.
[52] P. Tonella, G. Antoniol, R. Fiutem, and E. Merlo, "Variable precision reaching definitions analysis for software maintenance," in *Proc. 1st Euromicro Conf. Softw. Maintenance Reengineering*, 1997, pp. 60–67.
[53] E. Cambria, Y. Li, F. Z. Xing, S. Poria, and K. Kwok, "SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 105–114.
[54] X. Li, P. Wu, C. Zou, H. Xie, and F. L. Wang, "Sentiment lossless summarization," *Knowl.-Based Syst.*, vol. 227, 2021, Art. no. 107170.
[55] R. Moraes, J. F. Valiati, and W. P. G. Neto, "Document-level sentiment classification: An empirical comparison between SVM and ANN," *Expert Syst. Appl.*, vol. 40, no. 2, pp. 621–633, 2013.
[56] P. Kalarani and S. Selva Brunda, "Sentiment analysis by POS and joint sentiment topic features using SVM and ANN," *Soft Comput.*, vol. 23, no. 16, pp. 7067–7079, 2019.

Xiaodong Li received the PhD degree in computer science from the City University of Hong Kong. He is an associate professor with the College of Computer and Information, Hohai University. His research interests include artificial intelligence, machine learning, information retrieval, and data mining, especially Big Data applications to algorithmic trading in finance and water resource management in hydrology. He has served as a PI of a subproject in the National Key R&D Program of China, and a PI of a project in the National Natural Science Foundation of China.



Chenxin Zou received the bachelor's degree in computer science and technology from Hohai University, where he is currently working toward the postgraduate degree with the College of Computer and Information. His research interests include machine learning and knowledge graph, especially construction of domain knowledge graph. He has participated in the FinTech Research Group, under the supervision of Dr. Xiaodong Li.

Pangjing Wu received the bachelor's degree in information and computing science from Hohai University. He is currently working toward the postgraduate degree with the College of Computer and Information, Hohai University. His research interests include reinforcement learning, financial engineering, especially Big Data applications to algorithmic trading in finance. He has participated in the FinTech research group under supervision of Dr. Xiaodong Li.

Qing Li (Fellow, IEEE) received the MSc and PhD degrees from the University of Southern California, Los Angeles in computer science. He is currently a chair professor with the Department of Computing, Hong Kong Polytechnic University. His research interests include multi-modal data management, conceptual data modeling, social media, Web services, and e-learning systems. He has authored more than 400 publications. He is actively involved in the research community. He is a fellow of IEE/IET, U.K., and a distinguished member of CCF, China. He served as a conference and program chair/co-chair for numerous major international conferences. He also sits in the Steering Committees of DASFAA, ER, ACM RecSys, IEEE U-MEDIA, and ICWL. He has served as an associate editor for a number of major technical journals, including the *IEEE Transactions on Knowledge and Data Engineering*, *ACM Transactions on Internet Technology, Data Science and Engineering*, *World Wide Web*, and the *Journal of Web Engineering*.