Expert Systems with Applications 156 (2020) 113465

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A deceptive review detection framework: Combination of coarse and fine-grained features

Ning Cao [a], Shujuan Ji [a,*], Dickson K.W. Chiu [b], Mingxiang He [a], Xiaohong Sun [c]

[a] Key Laboratory for Wisdom Mine Information Technology of Shandong Province, Shandong University of Science and Technology, Qingdao, China
[b] Faculty of Education, The University of Hong Kong, China
[c] Audit Office, Shandong University of Science and Technology, Qingdao, China

ARTICLE INFO

Article history:
Received 1 February 2020
Revised 4 April 2020
Accepted 17 April 2020
Available online 22 April 2020

Keywords:
Deceptive reviews detection
LDA topic model
Deep learning
Coarse-grained features
Fine-grained features

ABSTRACT

Electronic commerce has become a popular shopping mode. To enhance their reputations, attract more customers, and finally obtain more benefits, dishonest sellers often recruit buyers or robots to post a large number of deceptive reviews to mislead users. According to the interpretability of learning results, existing methods for detecting deceptive reviews can be mainly divided into explicit feature-based mining ones and neural network-based implicit feature mining ones. The nature of these works is accurate text classification based on coarse-grained features (e.g., topic, sentence, and document) or fine-grained features (e.g., word). To take full merits of existing approaches, this paper proposes a new framework that explores a method to combine the coarse-grained features and the fine-grained features. In this framework, the coarse-grained implicit semantic features of the topic distribution are learned by the concatenation of a Latent Dirichlet Allocation (LDA) topic model and a 2-layered neural network. The fine-grained implicit semantic features from the word vectors representation of the reviews are parallelly learned by a deep learning framework. Finally, these two granular features are combined and adopted to train a Support Vector Machine (SVM) classifier for detecting whether a review is deceptive or not. To verify the effectiveness and performance of this framework, we derive three models by specifying three popular deep learning models, such as TextCNN, long short-term memory (LSTM), and Bi-directional LSTM (BiLSTM) to learn the fine-grained features. Experimental results on a mixed-domain dataset and balanced/unbalanced in-domain datasets show that all the combination models are superior to the corresponding baseline models considering single features.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Nowadays, shopping online has become part of our life. Buyers on e-commerce platforms often refer to online product reviews when shopping. Driven by various profits, sellers often collude with review spammers in writing deceptive reviews (Jindal & Liu, 2008; Ott, Choi, Cardie, & Hancock, 2011). Deceptive reviews not only are unfair to legitimate businesses participating in market competition but also will mislead consumers' shopping decisions and cause damage to consumers' rights. Therefore, the detection of deceptive reviews is vital. However, an experimental study shows that the accuracy of human recognition of deceptive reviews is only 57.3% (Ott et al., 2011). The design of automatic methods for accurately detecting deceptive reviews not only can crackdown on the speculative behavior of dishonest sellers, but it is also conducive to the construction of social trust systems and helpful for consumers to make better purchasing decisions.

Researchers have done lots of work in designing automatic deceptive review detection algorithms. They deem deceptive opinion spam detection as a classification problem (Ott et al., 2011; Ren, Ji, & Zhang, 2014). Commonly used features are word bag features, POS (part-of-speech features), LIWC (Linguistic Inquiry and Word Count), and so on. Though such features may be able to perform well, the sparsity of features makes it difficult to extract-coarse semantic information over a review (Ren & Ji, 2017). Many topic models are also used to identify deceptive reviews as they can obtain coarse semantic information of the reviews. For example, Li, Cardie, and Li (2013) used an extended Latent Dirichlet Allocation (LDA) topic model to detect deceptive reviews. Li, Xie, Sun, Ma, and Bai (2011) implemented text categorization using the LDA topic model. In their work, the topic distribution is used as features to complete the text classification task. In recent years,

* Corresponding author.
E-mail addresses: 836300237@qq.com (N. Cao), jane_ji2003@aliyun.com (S. Ji), dicksonchiu@ieee.org (D.K.W. Chiu), hmx0708@163.com (M. He), 2079038050@qq.com (X. Sun).

https://doi.org/10.1016/j.eswa.2020.113465
0957-4174/© 2020 Elsevier Ltd. All rights reserved.

neural networks have become more and more widely used in natural language processing (NLP). Deep learning methods based on word vectors can obtain fine-grained features in reviews. However, due to the neglect of coarse-grained text features, the semantic features extracted using deep learning methods have some limitations on the expression of the coarse semantic information.

To solve the problem that implicit semantic of reviews are not comprehensively obtained and used, this paper proposes a novel framework that combines both coarse and fine-grained features to detect deceptive reviews. The contributions of this framework are as follows:

(1) To extract the coarse-grained features, we first use the LDA topic model to obtain the features of the topic distribution of reviews, then apply a 2-layered BP neural network to obtain the topic-based implicit semantic features from the topic distribution of reviews.
(2) To parallelly extract the fine-grained features, we use a deep learning model to obtain the implicit semantic features from the word vectors representation of reviews. Moreover, we combine the extracted coarse-grained and fine-grained features to realize the final text classification.
(3) To verify the performance of this framework, we derive three deep learning models, i.e., TextCNN, LSTM, and BiLSTM, as models in the left part of our framework for extracting the fine-grained implicit semantic features from the word vectors representation of reviews.
(4) The experimental results on a mixed-domain dataset and balanced/unbalanced in-domain datasets show that the models combining coarse and fine-grained features outperform all the baselines that only consider coarse-grained or fine-grained features in accuracy, precision, recall, and F1-score. In particular, the experimental results further show that our framework is appropriate to be applied in real-life mixed unbalanced e-commerce environments.

In the remaining parts, Section 2 introduces the related work. Section 3 presents the proposed framework in this paper. Section 4 illustrates the experimental settings and analyses the experimental results. Section 5 summarizes this paper with our future work direction.

2. Related work

Generally, there are two categories of methods for detecting deceptive reviews. The first category is explicit feature-based mining ones based on features engineering and classification. Such features include explicit fine-grained ones based on words and explicit coarse-grained ones based on topics. Therefore, this kind of methods have good interpretability. The second category methods mainly use neural networks as feature extractors and classifiers to identify deceptive reviews. The features extracted by neural networks are implicit and have poor interpretability. The extracted features include implicit fine-grained ones based on words and the implicit coarse-grained ones such as topics, sentences, and documents. Here we review the related work from these two methods.

2.1. Explicit feature-based mining methods

According to features extracted, explicit feature-based mining methods can be divided into fine-grained ones based on words and coarse-grained ones based on topics. Jindal and Liu (2008) made a preliminary attempt in the field of deceptive review recognition. They analyzed the explicit characteristics of reviews and found that it was almost impossible to distinguish deceptive reviews manually as there was a lack of labeled datasets. Therefore, they regarded duplicated or nearly duplicated comments as deceptive reviews and trained logistic regression classifiers based on reviews, reviewers, and products. Yoo and Gretzel (2009) analyzed the differences in language structures between 40 real hotel reviews and 42 deceptive reviews and found that it is difficult to distinguish deceptive and truthful reviews based on structural properties. Ott et al. (2011) collected a dataset containing real reviews from 20 hotels and deceptive reviews written by Amazon Mechanical Turkey (AMT). They used the POS, n-gram, and LIWC features to train naive Bayes and support vector machine classifiers. Feng, Banerjee, and Choi (2012) used CFG (Context Free Grammar) parse trees to extract the syntactic features of reviews to identify deceptive reviews. They designed experiments and verified that deep syntactic features could improve the detection performance of deceptive reviews. Fusilier, Cabrera, Montes, and Rosso (2013) proposed a model based on PU-learning (learning from positive and unlabeled examples), which solved the problem of lack of labeled data to some extent. Li, Ott, Cardie, and Hovy (2014) expanded the gold standard deceptive review dataset and published a new gold dataset, which includes data from hotels, restaurants, and doctors. The authors improved the SAGE model and used Unigram, POS, and LIWC as features to implement in-domain and cross-domain experiments. Martinez-Torres and Toral (2019) designed a deceptive review detection method based on the bag of words. They collected the bag of words according to 3 methods: (i) selecting all the existing words; (ii) selecting the attributes that can be uniquely associated to the deceptive or the non-deceptive classes; (iii) selecting a polarity oriented attribute, which extended the deceptive and the non-deceptive classes to four classes by adding the sentiment polarity. Six different classifiers were trained and tested using TF-IDF value of each bag of words. They found that some words are associated with the target classes, which is helpful for identifying the deceptive reviews, especially when the sentiment orientation of reviews is considered.

Using topic models to achieve text categorization is another explicit feature-based mining methods. The topic models can be used to obtain the topic distribution of the reviews, which can serve as an explicit feature to detect deceptive reviews. Blei, Ng, and Jordan (2003) proposed the LDA topic model based on Bayesian rules. In their model, the meaning of a document comprises different topics, and each word in the document is related to a specific topic. Later, scholars made improvements based on LDA. For instance, Lin and He (2009) proposed the Joint Sentiment/Topic (JST) model by adding a sentiment layer to the LDA model, and they thought that topics rely on sentiment. Jo and Oh (2011) proposed an Aspect and Sentiment Unification Model (ASUM) similar to JST, except that all words in a sentence of ASUM come from the same topic distribution. Dong et al. (2018) proposed an unsupervised topic-sentiment joint probabilistic model (UTSJ) and used it for the detection of deceptive reviews. Unlike JST, the sentiment in UTSJ depends on topics. Jia, Zhang, Wang, and Liu (2018) used LDA to obtain extended terms to detect deceptive reviews, and they verified the effectiveness of the LDA model for deceptive review detection. Zhang, Wang, and Wang (2017) combined word embedding and the LDA topic model to deal with short text classification tasks. They first used the LDA topic model to obtain a topic-word distribution matrix, expanded the term for each word in the original text and calculated the weight of the extended term, and finally used the k-nearest neighbors (KNN) algorithm to train the model. Qiu, Jia, Liu, and Fan (2019) presented an HVCH (Hot topic detection based on the VSM combined) fusion model algorithm to infer topics from the Microblog.

Besides, some scholars have studied deceptive information detection from the viewpoint of feature selection. Wang, Liu, and

Zhu (2014) proposed a two-step based hybrid feature selection method for spam filtering. Kiliroor and Valliyammai (2019) considered the features in email data as well as social networks for spam filtering. Wang and Zhou (2018) proposed a Multi-level spam SMS recognition method, which comprises three methods that extract symbolic features, text similarity features, and pattern features of emails, respectively.

2.2. Neural network-based implicit feature mining methods

Neural networks have an excellent performance in natural language processing. Compared with explicit feature-based mining methods, neural networks can automatically extract features and represent them as continuous real-valued vectors. Commonly used models in the field of deceptive information identification are convolutional neural network (CNN), recurrent neural network (RNN), and gated recurrent neural network (GRNN).

For example, Kim (2014) used CNN in text classification. He first transformed text into word vectors and then used CNN in training. In particular, when training the convolutional neural network, word vectors can be dynamically fine-tuned. Moreover, he showed that the effect of dynamic word vectors is better than the static word vectors. Some researchers also used CNN to capture implicit features of text (Johnson & Zhang, 2014; Kalchbrenner, Grefenstette, & Blunsom, 2014). Tang, Qin, and Liu (2015) used GRNN (Chung, Gulcehre, Cho, & Bengio, 2015) to model reviews and implement sentiment classification. Liu, Qiu, and Huang (2016) used RNN and multi-task learning for text classification. Lai, Xu, Liu, and Zhao (2015) used RNN to extract contextual information and CNN to represent text for text classification. Considering the hierarchical structure of documents, Yang et al. (2016) used the attention mechanism and RNN to achieve document classification. Different from the above works solely using neural networks to realize text classification, some researchers combined the LDA model and neural networks to improve the classification performance. For example, Xian-yan, Rong-yi, Ya-hui, and Zhenguo (2019) proposed a multilingual short text classification model. They firstly spliced topic vectors and word vectors to represent text, and then they used BiLSTM and CNN for text classification. In terms of coarse-grained feature extraction, Ren and Ji (2017) first extracted sentence features from word vectors representation of reviews by CNN, and then document features from sentence features by GRNN. Similarly, Li, Qin, Ren, and Liu (2017) also learned document representation to detect deceptive reviews, and they used a weighted-average scheme to obtain the document feature as they consider different sentences have different importance. Besides, in their models, POS features and first-person pronouns are considered.

2.3. Summary

The explicit feature-based mining methods listed in Section 2.1 are designed based on fine-grained word features or coarse-grained topic features. These methods have the merit of good interpretability, together with the limitation of sparsity in text representations. The neural network-based implicit feature mining methods that are designed based on fine-grained word features, coarse-grained sentence features or document features have two advantages: they do not need manual extraction of features, and the performance of neural network-based implicit feature mining methods usually outperform explicit feature-based ones. However, neural network-based implicit feature mining methods generally lack interpretability.

Different from existing works (Dong et al., 2018; Li et al., 2017; Ren & Ji, 2017) that focus on extracting solely coarse-grained or fine-grained information such as topic-sentiment, sentence or document, we propose a new framework, which synchronously extracts both coarse-grained and fine-grained features and combine them to realize deceptive reviews detection. This framework takes full advantage of the merits of the LDA topic model in extracting explicit coarse topic information as well as neural network-based models in extracting implicit fine-grained semantic information. In other words, our framework makes full use of the features of different granularity to extract the hidden implicit features in the review text more comprehensively. Different from existing non-neural network machine learning methods, this new methodology can automatically extract fine-grained features instead of manually. Different from existing neural networks-based machine learning methods, this new methodology can obtain more comprehensive information by extracting coarse-grained topic features.

3. Our approach

This section introduces our framework for detecting deceptive reviews based on combining coarse and fine-grained features. Fig. 1 depicts the architecture of this framework.

The left part of this architecture aims at learning the fine-grained feature of reviews, which is implemented by Algorithm 1 according to three steps. First, the reviews after preprocessing are transformed into the word vectors representation (see Step 1 in Algorithm 1). Secondly, a deep neural network model can be used to train the word vectors representation of the reviews in the training set (see Steps 2–5 in Algorithm 1). Finally, the output of the full connection layer is chosen as the implicit fine-grained feature of the reviews (see Steps 6–8 in Algorithm 1). Notably, in the second step in the process of Algorithm 1, we do not prescribe a specific deep neural network. That is because we want to test the applicability of our framework over different deep neural networks.

The right part of this framework focus on the learning of coarse-grained features, which is implemented by concatenating the LDA model and a 2-layered BP neural network (denoted as LDA-BP). Algorithm 2 illustrates the detailed steps. First, the LDA topic model is used to obtain the explicit topic distribution features of the preprocessed reviews (see Steps 1–4 in Algorithm 2). Secondly, based on the explicit topic distribution features, a double-layers BP neural network is used for training (see Steps 5–8 in Algorithm 2). Finally, the output of the hidden layer of the neural network is taken as the implicit coarse-grained topic features of the reviews (see Steps 9–11 in Algorithm 2).

The top part of this framework combines the learned fine-grained and coarse-grained features and trains an SVM classifier, which is implemented by Algorithm 3 with two steps. First, the two kinds of features are spliced into a vector. Next, an SVM classifier is trained by taking the combined vectors as input. SVM classifiers are chosen for three reasons. (1) SVM classifiers can map the input vector to a high-dimension feature space, and it has a high generalization ability (Cortes & Vapnik, 1995). (2) SVM classifiers are suitable for dealing with text classification problems (Joachims, 1998). (3) Many related researches (e.g., Blei et al., 2003; Ott et al., 2011; Li et al., 2014) used SVM or combined SVM classifiers with neural networks and made good achievements (Chen & Zhang, 2018; Wang & Qu, 2017). As our framework needs to address two granularity properties (i.e., topics and words) that are high-dimensional, therefore, the SVM classifier is suitable for our purpose. Notably, in the left part of this framework, no specific deep neural network is specified. In the realization of this framework, we can realize a particular model by specifying an actual deep neural network.

Algorithm 1: Fine-grained feature acquisition algorithm

```
Input: preprocessed reviews set $R^p$, pre-trained word vector
  matrix M
Output: Fine-grained feature $F^f$ of reviews
Process:
Get word vectors representation of reviews R from
preprocessed reviews $R^p$ and M;
Select a certain deep neural network model $d_j \in D$;
// D is a set of deep neural network models that can be
used for mining the fine-grained features
For t = 1 to n:        //n is the epochs number of the $d_j$:
  Train the deep neural network $d_j$;
End for
For each review $r_i \in R$:
  Get the fine-grained features $F^f$ in the fully connected
  layer of $d_j$;
End for
Return $F^f$
```

Algorithm 2: Coarse-grained feature acquisition algorithm

```
Input: preprocessed reviews set $R^p$, the number of topics K,
  hyper-parameters $\alpha$ and $\beta$
Output: Coarse-grained features $F^c$ of reviews
Process:
Train the LDA topic model taking preprocessed reviews
$R^p$ as input;
For each review $r_i \in R$:
  Get the topic distribution $\theta_i \in \vec{\theta}$ of each review $r_i$ using
  the trained LDA topic model;
End for
For i = 1 to n:        //n is the epochs number of the 2-layer
neural network
  Get the input topic distribution $\theta_i \in \vec{\theta}$ of each review $r_i$;
  Train the 2-layer neural network;
End for
For each review $r_i$ in reviews R:
  Get the topic features $F^c$ in the hidden layer of the
  neural network;
End for
Return $F^c$
```

Algorithm 3: Features fusion and classification algorithm

```
Input: Fine-grained feature $F^f$ of reviews R, Coarse-grained
  feature $F^c$ of reviewsR
Output: SVM classifier
Process:
Merge the fine-grained features and coarse-grained
features of reviews R;
For i = 1 to n:        // n is the epochs number of the Support
Vector Machine:
  Get input final merged feature F of the reviews R;
  Train the SVM classifier;
End for
Return SVM classifer
```

4. Experiment

To verify the performance of our framework, we have designed four sets of experiments. As the performance of the LDA topic models is affected by the number of topics, the first set of experiments is to select the appropriate number of topics. The second set of experiments is an in-domain one on the gold standard small dataset (Li et al., 2014), which aims at benchmarking the performance of our framework in three different domains. The third set of experiments is also an in-domain one on the Yelp dataset, which target at testing the performance of our framework on a large-scale dataset. To analyze the impact of balanced dataset and unbalanced dataset on classification performance, we construct the balanced/unbalanced datasets based on the original gold standard small dataset used in the second experiment and the large Yelp dataset used in the third experiment, respectively. We design two sub-experiments based on these two balanced/unbalanced datasets. To verify the effectiveness and performance of our framework, we design the fourth set of experiments, which is a mixed-domain one over the gold standard small dataset (Li et al., 2014).

4.1. Data and experimental settings

We use two sets of experimental datasets in this paper. One dataset is a small one with the statistics shown in Table 1, which is the gold standard deceptive review dataset published by (Li et al., 2014). This dataset includes three different domains, i.e., hotels, restaurants, and doctors. The other dataset is the Yelp restaurant reviews (Mukherjee, Venkataraman, Liu, & Glance, 2013). It is a large dataset comprising 8303 deceptive reviews and 58,716 real reviews. The statistics about the original Yelp restaurant dataset is listed in the first row of Table 3.

To realize the aims of the designed four sets of experiments, we construct 3 kinds (7 subsets in total) of experimental datasets over the original small gold standard dataset. First, neglecting the data from experts, we construct a balanced dataset and an unbalanced dataset over the datasets of hotels, restaurants, and doctors, respectively. Secondly, we mixed the original datasets of hotels, restaurants, and doctors to form a dataset of mixed domains. Table 2 listed the statistics of the 7 experimental datasets in which the domain name labeled with "y" denotes the balanced dataset, and the domain name labeled with "n" denotes the unbalanced dataset. Besides, based on the original Yelp restaurant dataset, we derive a balanced Yelp dataset and an unbalanced one. The second row and third row of Table 3 show the statistics of the derived Yelp datasets in which Yelp^y and Yelp^n represent the balanced and unbalanced dataset respectively. In the last column of Tables 2 and 3, we list the set of experiments in which we use each set of dataset.

To verify the performance of our framework, in the second, third, and fourth set of experiments, we derives three deep learning models from our framework, i.e., TextCNN, LSTM, and BiLSTM to learn the fine-grained feature because all these models are excellent and popular ones in extracting implicit text features (Graves & Schmidhuber, 2005; Hochreiter & Schmidhuber, 1997; Kim, 2014). Table 4 lists the structure of the deep learning models used in our experiments. According to these settings, the three tested models of our framework are denoted as LDA-BP$\oplus$LSTM, LDA-BP$\oplus$Bi-LSTM and LDA-BP$\oplus$TextCNN.

As this paper aims at detecting the deceptive review from the perspective of review text, and the Unigram, POS, and LDA topic models are commonly used explicit methods that are good at extracting features such as word bag, part-of-speech and topic distribution; therefore we select these feature-based mining methods as baselines. Moreover, the LDA-BP model is selected as another

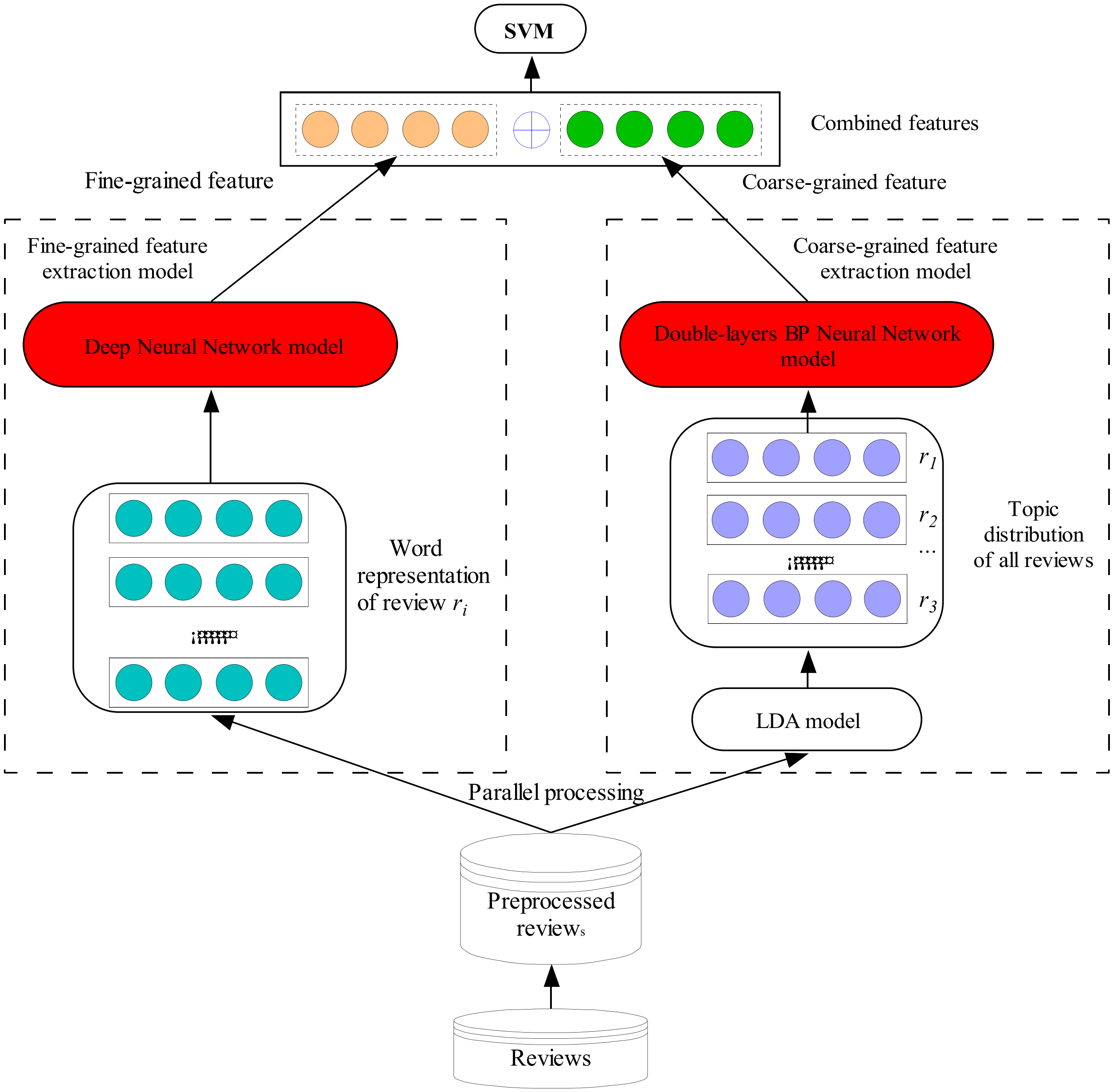
SVM
Combined features
Fine-grained feature
Coarse-grained feature
Fine-grained feature extraction model
Deep Neural Network model
Coarse-grained feature extraction model
Double-layers BP Neural Network model
Word representation of review r_i
r_1
r_2
…
r_3
Topic distribution of all reviews
LDA model
Parallel processing
Preprocessed reviews
Reviews

Fig. 1. The framework architecture based on combining coarse and fine-grained features.

Table 1
Statistics of the golden standard small dataset.

Domain	Turkey	Expert	Customer	Total reviews
Hotel	800	280	800	1880
Restaurant	200	0	200	400
Doctor	356	0	200	556

baseline because it is the right part of our derived models, which also only focus on learning the explicit and implicit coarse-grained features**.** Besides, we choose the TextCNN, LSTM, and BiLSTM as the baselines because all these models are recent excellent and popular models used for extracting implicit text features, which are also used in the left part of our framework. It should be noted that as the models of LDA, Unigram, POS, LDA-BP$\oplus$LSTM,

Table 2
Statistics of experimental datasets constructed based on the small-dataset.

Dataset	Deceptive	True	Deceptive%	Total reviews	Used experiments
Hotel^{y}	800	800	50	1600	1st, 2nd set
Restaurant^{y}	200	200	50	400	1st, 2nd set
Doctor^{y}	200	200	50	400	1st, 2nd set
Hotel^{n}	550	800	40.7	1350	2nd set
Restaurant^{n}	135	200	40.3	335	2nd set
Doctor^{n}	135	200	40.3	335	2nd set
Mixed	1636	1200	57.7	2836	1st, 4th set

Table 3
Statistics of Yelp-related datasets.

Dataset	Deceptive	True	Deceptive%	Total reviews	Used experiments
Original Yelp	8303	58,716	12.4	60,719	
Yelp^{y}	8303	8303	50	16,606	1st, 3rd set
Yelp^{n}	5535	8303	40	13,838	3rd set

Table 4
The deep learning model structure used in this paper.

Model	Structure
LSTM	one LSTM layer, one fully connected layer
Bi-LSTM	one Bi-LSTM layer, one fully connected layer
TextCNN	Four diffirent filter size (2, 3, 4, 5), one fully connected layer

LDA-BP⊕Bi-LSTM, and LDA-BP⊕TextCNN only focus on feature learning, an SVM classifier is needed.

Word vectors representation (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013; Pennington, Socher, & Manning, 2014) has been widely used in various natural language processing tasks. The word vectors used in this paper is downloaded from the web,[1] which is pre-trained by the word vector learning algorithm (Mikolov et al., 2013) using the Google News dataset. Pre-trained word vectors are trained on a large-scale real corpus background, with richer information and stronger representation capabilities so that they can be adapted to tasks in different domains. In the process of deep learning model training, we set the word embedding layer as fine-tuning, which is to fine-tune the pre-trained word vectors during the training process. Ren and Ji (2017) experimentally verified that a fine-tuned word embedding layer works best.

4.2. Evaluation metrics

All the experiments are validated by five-fold cross-validation. Accuracy, precision, recall, and F1-score were used as evaluation criteria, which are defined in Eqs. (1) to (4), respectively. They are calculated using macro-average. TP (True Positive) is the number of samples that are correctly classified as positive. FP (False Positive) is the number of samples that are wrongly classified as positive. TN (True Negative) is the number of samples that are correctly classified as negative. FN (False Negative) is the number of samples that are wrongly classified as negative. Accuracy (A) represents the proportion of correctly classified samples to the total number of samples. Precision (P) is the proportion of samples that are correctly predicted for a certain type of predicted sample. Recall (R) is the proportion of samples that are correctly predicted for a particular type of actual sample. F1-score (F) reflects both accuracy and recall rate.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

4.3. Selection of the optimal number of topics

Since the LDA topic model used in this paper needs to specify the number of topics, we design the first set of experiments to analyze the influence of different numbers of topics on the identification of deceptive reviews for selecting a suitable number of topics. In the parameter calculation process of the LDA topic model, we use the Gibbs Sampling method (Porteous et al., 2008) because this method has a more precise estimation.

4.3.1. Optimal number of topics for small datasets

We implement experiments in the three balanced domains of the small datasets (i.e., Hotel^{y}, Restaurant^{y}, and Doctor^{y}) and test out seven different numbers of topics (i.e., 10, 20, 30, 50, 100, 150, 200) for comparison. Tables 5–7 show the experimental results on the datasets of Hotel^{y}, Restaurant^{y}, and Doctor^{y} with different numbers of topics, respectively.

The optimal number of topics for the Hotel^{y} and Doctor^{y} datasets is 20, while that for the Restaurant^{y} dataset is 10. Table 8 shows the experimental results of the Mixed dataset with different topic numbers. On the Mixed dataset, the optimal number of topics is 30. As the influence of the number of topics on these datasets is

Table 5
Performance of LDA on the Hotel^{y} Dataset under various numbers of topics.

Topics number	A	P	R	F
10	0.802	0.805	0.802	0.801
20	**0.816**	**0.816**	**0.816**	**0.816**
30	0.789	0.789	0.789	0.789
50	0.784	0.785	0.784	0.784
100	0.763	0.769	0.763	0.762
150	0.763	0.765	0.763	0.763
200	0.74	0.748	0.74	0.738

Table 6
Performance of LDA on the Restaurant^{y} Dataset under various numbers of topics.

Topics number	A	P	R	F
10	**0.773**	**0.773**	**0.772**	**0.772**
20	0.75	0.754	0.75	0.749
30	0.747	0.761	0.747	0.741
50	0.745	0.749	0.745	0.744
100	0.735	0.739	0.735	0.734
150	0.738	0.741	0.738	0.736
200	0.718	0.718	0.718	0.717

Table 7
Performance of LDA on the Doctor^{y} Dataset under various numbers of topics.

Topics number	A	P	R	F
10	0.762	0.769	0.762	0.761
20	**0.765**	**0.775**	**0.765**	**0.762**
30	0.715	0.73	0.715	0.71
50	0.718	0.721	0.718	0.715
100	0.685	0.698	0.685	0.679
150	0.67	0.692	0.67	0.657
200	0.685	0.713	0.685	0.674

[1] code.google.com/p/word2vec.

Table 8
Performance of LDA on the Mixed Dataset under various numbers of topics.

Topics number	A	P	R	F
10	0.687	0.722	0.644	0.634
20	0.724	0.734	0.695	0.698
30	**0.748**	**0.753**	**0.725**	**0.73**
50	0.715	0.735	0.682	0.682
100	0.698	0.731	0.658	0.652
150	0.66	0.732	0.605	0.573
200	0.659	0.725	0.605	0.575

not very different, and the performance is generally well when the number of topics is 20, we choose 20 as the number of topics in the small dataset.

4.3.2. Optimal number of topics for large dataset

As the reviews in the Yelp^{y} dataset involve richer and more detailed contents, the optimal number of topics in the Yelp^{y} dataset should be larger than the small dataset. Therefore, we select several large numbers of topics, i.e., 20, 50, 100, 150, 200, 250, and 300 for testing. Table 9 shows the results for the Yelp^{y} dataset with different topic numbers. We chose 250 as the number of topics in the large-dataset, as this yields the best classification performance.

4.4. In-domain results and analysis on the small datasets

In the in-domain experiments on the small dataset, this paper compares the effectiveness and performance of each model on the datasets of balanced/unbalanced hotels, restaurants, and doctors. The following sub-sections illustrate the experimental results in detail.

Table 10 lists the experimental results on the Hotel dataset. It shows that, on the balanced Hotel^{y} dataset, the coarse-grained LDA-related topic models achieve good results (all the values of A, P, R, F are larger than 0.8), which are better than those of the fined-grained Unigram, POS, and LSTM models. These results indicate that coarse-grained topic information helps identify the deceptive review. The Bi-LSTM model is as good as LDA-related models, and the TextCNN model is better than all the other models for all metrics. As for the unbalanced dataset, the performance of coarse-grained LDA-related topic models decreases significantly. The difference is that the performance of the fine-grained models either remain stable or fluctuate slightly up and down. No matter balanced and unbalanced datasets are considered, in contrast to the original LSTM, Bi-LSTM, and TextCNN models that only consider fine-grained features, the three models (i.e., LDA-BP$\oplus$LSTM, LDA-BP$\oplus$Bi-LSTM, LDA-BP$\oplus$TextCNN) derived from our framework gain correspondingly higher performance than the fine-grained ones. This is because fine-grained and coarse-grained features are taken into account in our framework, both of them help identify the deceptive review. In particular, compared with the original TextCNN model, the performance of LDA-BP$\oplus$TextCNN model correspondingly improves (1.7%, 1.4%, 1.8%, 1.7%) on the balanced dataset and improves (1.3%, 0.8%, 1.7%, 1.5%) on the unbalanced dataset. Moreover, compared with the coarse-grained LDA topic model, the performance of the LDA-BP$\oplus$TextCNN model correspondingly improves (4.3%, 4.4%, 4.4%, 4.3%) on the balanced dataset and improves (7.2%, 6.0%, 9.2%, 8.5%) on the unbalanced dataset.

Table 9
Performance of LDA on the Yelp^{y} Dataset under various numbers of topics.

Topics number	A	P	R	F
20	0.803	0.803	0.803	0.803
50	0.812	0.813	0.812	0.812
100	0.832	0.832	0.832	0.832
150	0.83	0.831	0.83	0.83
200	0.833	0.835	0.833	0.833
250	**0.835**	**0.837**	**0.835**	**0.835**
300	0.832	0.836	0.832	0.832

Table 11 lists the experimental results on the Restaurant dataset, which is smaller than the Hotel dataset. Similar to the results from the Hotel dataset, the coarse-grained LDA-related topic models achieve good performance on the balanced dataset and worse performance on the unbalanced dataset. When dealing with balanced/unbalanced datasets, the performance of all fine-grained models remain stable or fluctuate slightly, which means that the LDA topic model is not suitable to deal with unbalanced datasets. Similar to the results on the Hotel dataset, our derived combination models (LDA-BP$\oplus$LSTM, LDA-BP$\oplus$Bi-LSTM, LDA-BP$\oplus$TextCNN) outperform the corresponding fine-grained ones and coarse-grained ones. In particular, for the balanced Restaurant^{y} dataset, the LDA-BP$\oplus$TextCNN correspondingly improves the fine-grained TextCNN model and the coarse-grained LDA model by (5.7%, 3.6%, 5.7%, 6.4%) and (6.5%, 6.8%, 6.5%, 6.6%), respectively. For the unbalanced Restaurant^{n} dataset, compared with the balanced dataset, the performance decreases slightly for all the models except for the Unigram and the POS. That is probably because these two models are particularly good at dealing with unbalanced data. The POS model is superior to the other baselines, which proves that the part-of-speech feature of reviews can help us identify the deceptive review. Moreover, on the unbalanced Restaurant^{n} dataset, the LDA-BP$\oplus$TextCNN model obtains the best performance among the three derived models and outperform the fine-grained TextCNN model and the coarse-grained LDA model by (4.5%, 4.8%, 6.2%, 6.9%) and (7.5%, 7.6%, 8.5%, 9.4%), respectively.

Table 12 lists the experimental results on another small dataset, i.e., the Doctor dataset. Similar results are achieved on the Doctor

Table 10
Experimental results on the Hotel Dataset.

Model type		Model	Balanced Hotel^{y} dataset				Unbalanced Hotel^{n} dataset			
			A	P	R	F	A	P	R	F
Coarse-grained	Explicit feature mining	LDA	0.816	0.816	0.816	0.816	0.784	0.794	0.756	0.764
	Ex-&implicit feature mining	LDA-BP	0.813	0.814	0.813	0.813	0.77	0.763	0.758	0.76
Fine-grained	Explicit feature mining	Unigram	0.768	0.77	0.768	0.768	0.793	0.788	0.779	0.782
		POS	0.778	0.778	0.778	0.777	0.796	0.792	0.78	0.784
	Implicit feature mining	LSTM	0.786	0.786	0.786	0.786	0.779	0.794	0.757	0.761
		Bi-LSTM	0.814	0.826	0.814	0.812	0.829	0.836	0.812	0.817
		TextCNN	0.842	0.846	0.842	0.842	0.843	0.846	0.831	0.834
Coarse and Fine-grained fusion		LDA-BP$\oplus$LSTM	0.796	0.796	0.796	0.796	0.799	0.769	0.784	0.787
		LDA-BP$\oplus$Bi-LSTM	0.841	0.842	0.841	0.84	0.836	0.835	0.823	0.827
		LDA-BP$\oplus$TextCNN	**0.859**	**0.86**	**0.86**	**0.859**	**0.856**	**0.854**	**0.848**	**0.849**

Table 11
Experimental results on the Restaurant Dataset.

Model type		Model	Balanced Restaurant^{y} dataset				Unbalanced Restaurant^{n} dataset			
			A	P	R	F	A	P	R	F
Coarse-grained	Explicit feature mining	LDA	0.75	0.754	0.75	0.749	0.734	0.752	0.692	0.693
	Ex & implicit feature mining	LDA-BP	0.747	0.755	0.747	0.747	0.684	0.674	0.671	0.671
Fine-grained	Explicit feature mining	Unigram	0.747	0.751	0.747	0.747	0.77	0.763	0.756	0.757
		POS	0.772	0.777	0.772	0.772	0.785	0.778	0.773	0.775
	Implicit feature mining	LSTM	0.765	0.782	0.765	0.761	0.764	0.762	0.741	0.744
		Bi-LSTM	0.75	0.769	0.75	0.745	0.746	0.756	0.72	0.719
		TextCNN	0.758	0.786	0.758	0.751	0.764	0.78	0.715	0.718
Coarse and Fine-grained fusion		LDA-BP⊕LSTM	0.795	0.8	0.795	0.794	0.785	0.781	0.766	0.77
		LDA-BP⊕Bi-LSTM	0.772	0.777	0.772	0.772	0.788	0.791	0.772	0.775
		LDA-BP⊕TextCNN	**0.815**	**0.822**	**0.815**	**0.815**	**0.809**	**0.828**	**0.777**	**0.787**

Table 12
Experimental results on Doctor Dataset.

Model type		Model	Balanced Doctor^{y} dataset				Unbalanced Doctor^{n} dataset			
			A	P	R	F	A	P	R	F
Coarse-grained	Explicit feature mining	LDA	0.765	0.775	0.765	0.762	0.746	0.797	0.698	0.699
	Ex & implicit feature mining	LDA-BP	0.767	0.771	0.767	0.767	0.69	0.676	0.67	0.67
Fine-grained	Explicit feature mining	Unigram	0.735	0.74	0.735	0.734	0.752	0.749	0.732	0.735
		POS	0.715	0.719	0.715	0.714	0.737	0.738	0.708	0.712
	Implicit feature mining	LSTM	0.77	0.794	0.701	0.71	0.743	0.738	0.725	0.725
		Bi-LSTM	0.802	0.811	0.763	0.769	0.752	0.809	0.728	0.735
		TextCNN	0.775	0.8	0.721	0.717	0.755	0.818	0.706	0.698
Coarse and Fine-grained fusion		LDA-BP⊕LSTM	0.802	0.799	0.766	0.775	0.755	0.749	0.743	0.744
		LDA-BP⊕Bi-LSTM	0.818	0.811	0.791	0.798	0.815	0.831	0.777	0.783
		LDA-BP⊕TextCNN	**0.827**	**0.823**	**0.797**	**0.806**	**0.818**	**0.832**	**0.792**	**0.799**

dataset compared with ones on the Restaurant dataset. The LDA-BP⊕TextCNN model gains the best performance among the three derived models on either the balanced or the unbalanced Doctor dataset. Compared with the TextCNN model, the LDA-BP⊕TextCNN model correspondingly improves (5.2%, 2.3%, 7.6%, 8.9%) and (6.3%, 1.4%, 8.6%, 10.1%) on the balanced and unbalanced datasets, respectively. Moreover, compared with the coarse-grained LDA topic model, the performance of the LDA-BP⊕TextCNN model correspondingly improves (6.2%, 4.8%, 3.2%, 4.4%) and (7.2%, 3.5%, 9.4%, 10.0%) on the balanced and unbalanced datasets, respectively. Our framework makes full use of the features of different granularity to extract the implicit features in the review text more comprehensively, and the results show the effectiveness of our approach.

Notably, the performances metrics A, P, R, F on the Restaurant and Doctor datasets are correspondingly inferior to the ones on the Hotel dataset, no matter which model is adopted. That is because the Hotel dataset is big enough to train deep learning models and the LDA model to obtain features based on word vectors and features based on topic distribution, respectively. In summary, in the in-domain experiments on small-scale datasets, we can draw the following conclusion.

Conclusion 1 The three models (LDA-BP⊕LSTM, LDA-BPBi-LSTM, LDA-BP⊕TextCNN) derived from our framework are better than the baselines ones that only consider the fine-grained deep learning model or the coarse-grained LDA-based topic models on both balanced and unbalanced datasets. Moreover, the LDA-BP⊕TextCNN model gains the best performance.

4.5. In-domain results and analysis on the large dataset

To test the performance of the framework given in this paper on real large-scale datasets, we set up an in-domain experiment on the Yelp dataset. Table 13 lists the experimental results on the Yelp dataset, which is much larger than the Hotel dataset. Similar results are obtained on the large Yelp dataset compared with ones

Table 13
Experimental results on Yelp Dataset.

Model type		Model	Balanced Yelp^{y} dataset				Unbalanced Yelp^{n} dataset			
			A	P	R	F	A	P	R	F
Coarse-grained	Explicit feature mining	LDA	0.835	0.837	0.835	0.835	0.798	**0.827**	0.759	0.77
	ex & implicit feature mining	LDA-BP	0.835	0.836	0.835	0.835	0.82	0.81	0.807	0.81
Fine-grained	Explicit feature mining	Unigram	0.758	0.762	0.758	0.757	0.755	0.745	0.748	0.746
		POS	0.794	0.796	0.794	0.793	0.787	0.793	0.758	0.766
	implicit feature mining	LSTM	0.837	0.837	0.837	0.837	0.821	0.816	0.808	0.811
		Bi-LSTM	0.823	0.824	0.823	0.823	0.806	0.802	0.795	0.797
		TextCNN	0.842	0.843	0.842	0.842	0.827	0.823	0.813	0.817
Coarse and Fine-grained fusion		LDA-BP⊕LSTM	0.841	0.841	0.84	0.84	0.823	0.817	0.811	0.814
		LDA-BP⊕Bi-LSTM	0.829	0.829	0.829	0.829	0.819	0.814	0.808	0.81
		LDA-BP⊕TextCNN	**0.845**	**0.845**	**0.845**	**0.845**	**0.829**	0.824	**0.816**	**0.819**

on small datasets described in Section 4.4. The results of the unbalanced dataset are worse than those of the balanced dataset: the Unigram and POS models perform worse on the unbalanced Yelp^n dataset than on the balanced Yelp^y dataset, which yields the opposite result on small datasets. This is probably because the Unigram and POS models are not suitable for dealing with unbalanced large-scale datasets.

Among the three derived models, the LDA-BP⊕TextCNN model gains the best performance on the balanced/unbalanced Yelp datasets, and all of them outperform the models that consider solely coarse-grained features or fine-grained features. Compared with the TextCNN model, the LDA-BP⊕TextCNN model correspondingly improves (0.3%, 0.2%, 0.3%, 0.3%) and (0.2%, 0.1%, 0.3%, 0.2%) on the balanced and unbalanced datasets, respectively. This is because our framework not only considers fine-grained semantic information from word vectors representation of reviews, but also takes into account the coarse-grained semantic information from topic distribution of reviews. Compared with the benchmarked deep learning models, the improvement of our framework on the large-scale Yelp dataset is not so obvious. This is because the deep learning methods can learn the fine-grained features well on large datasets. However, the combination of coarse-grained topic features still improves the performance a bit. Compared with the coarse-grained LDA topic model, the performance of the LDA-BP⊕TextCNN model correspondingly improves (1.0%, 0.8%, 1.0%, 1.0%) and (3.1%, −0.3%, 5.7%, 4.9%) on the balanced and unbalanced datasets, respectively. In particular, the three derived models improve the performance well when unbalanced dataset are involved, especially in Recall (5.7%).

Therefore, based on the results of the in-domain experiments on large-scale dataset, we can draw the following conclusion.

Conclusion 2 The three models derived from our framework are superior to baselines in most metrics on real-life large-scale balanced/unbalanced datasets.

4.6. Mixed-domain results and analysis

To test the performance of our framework in mixed domains, we perform further experiments on a mixed-domain dataset. Table 14 lists the experimental results on the Mixed dataset. The three derived models from our framework outperform the models that only consider coarse-grained features or fine-grained features. In particular, the LDA-BP⊕TextCNN model has the best performance (83.1%, 83.1%, 82.1%, 82.5%) among the three derived models. Compared with the TextCNN and the coarse-grained LDA models, the LDA-BP⊕TextCNN model correspondingly improves (2.0%, 0.2%, 3.1%, 2.8%). and (10.7%, 9.7%, 12.6%, 12.7%), respectively. Notably, compared with coarse-grained LDA topic models and other explicit feature-based mining methods, the LDA-BP⊕TextCNN model in this paper achieves more than 10% improvement in Accuracy, Precision, Recall, and F1 scores. The results show that the single granularity features may not be enough to distinguish deceptive reviews. Therefore, in the mixed-domain experiments, the following conclusion can be drawn.

Conclusion 3 The three derived models can effectively deal with deceptive reviews detection on the mixed-domain dataset. Moreover, the LDA-BP⊕TextCNN model achieves very good performance on unbalanced mixed dataset, which is near to the real-life electronic commerce datasets.

Comprehensively considering the experimental results in Sections 4.4–4.6, we can get the following conclusions.

Conclusion 4 No matter for in-domain or mix-domain application, large or small dataset, balanced or unbalanced dataset, the overall performance of the derived models from our framework is better than other models using solely coarse-grained or fine-grained features. Therefore, our framework has generally good applicability.

Conclusion 5 Our framework that combines coarse-grained and fine-grained features can effectively improve the performance of

Table 14
Experimental results on Mixed Dataset.

Model type		Model	Mixed dataset			
			A	P	R	F
Coarse-grained	Explicit feature mining	LDA	0.724	0.734	0.695	0.698
	Ex & implicit feature mining	LDA-BP	0.726	0.727	0.706	0.709
Fine-grained	Explicit feature mining	Unigram	0.714	0.711	0.694	0.697
		POS	0.735	0.732	0.719	0.722
	Implicit feature mining	LSTM	0.765	0.767	0.751	0.753
		Bi-LSTM	0.816	0.82	0.807	0.809
		TextCNN	0.811	0.829	0.79	0.797
Coarse and Fine-grained fusion		LDA-BP⊕LSTM	0.778	0.774	0.768	0.77
		LDA-BP⊕Bi-LSTM	0.824	0.823	0.816	0.818
		LDA-BP⊕TextCNN	**0.831**	**0.831**	**0.821**	**0.825**

Table 15
The time complexity of each model.

Model type	Model name	Classifier needed	Time complexity
Coarse-grained models	LDA	SVM	O(LDA) + O(SVM)
	LDA-BP	/	O(LDA) + O(BP)
Fine-grained models	Unigram	SVM	O(Unigram) + O(SVM)
	POS	SVM	O(POS) + O(SVM)
	LSTM	/	O(LSTM)
	Bi-LSTM		O(Bi-LSTM)
	TextCNN		O(TextCNN)
Combination models	LDA-BPLSTM	SVM	max(O(LDA-BP), O(LSTM)) + O(SVM)
	LDA-BP⊕Bi-LSTM		max(O(LDA-BP), O(Bi-LSTM)) + O(SVM)
	LDA-BP⊕TextCNN		max(O(LDA-BP), O(TextCNN)) + O(SVM)

the models considering a single type of features. Moreover, the improvement of the coarse-grained LDA topic model is correspondingly greater than the fine-grained deep learning model.

4.7. Time complexity comparison

We compare the time complexity of our derived models with the baselines. Table 15 shows the comparison results of each model. Among the coarse-grained feature extraction models, the time complexity of the model structure of LDA combined with SVM (O(LDA)+O(SVM)) is lower than the LDA-BP model (O(LDA)+O(BP)). Among the fine-grained feature extraction models, the time complexity of the deep learning model is higher than that of Unigram and POS models, also higher than that of the coarse-grained LDA and LDA-BP models.

The time complexity of our derived combination models is determined by the corresponding fine-grained deep learning model. That is because the fine-grained model and the coarse-grained model are operated in parallel. Compared with the corresponding fine-grained deep learning model, the time complexity of our derived models is increased by O(SVM). For example, compared with the coarse-grained LDA model, the time complexity of the LDA-BP⊕TextCNN model increases by O(TextCNN)–O(LDA). Therefore, we can conclude that the time complexity of our derived models is slightly higher than the single coarse-grained or fine-grained models alone.

5. Conclusion

This paper proposes a deceptive review text detection framework that combines coarse and fine-grained features. To verify the effectiveness and performance of this framework, the typical LDA topic model, the explicit fine-grained feature mining models Unigram and POS, as well as excellent deep learning-based implicit feature mining models such as TextCNN, LSTM, and Bi-LSTM are selected and compared. Besides, to further verify the performance of this framework, a submodel (LDA-BP) of this framework is also used as a baseline. Comprehensive experiments are designed and implemented based on the gold standard dataset and the Yelp dataset. Experimental results show that our derived models achieve better detection performance on the different in-domain of balanced/unbalanced datasets than corresponding baselines. In particular, the performance of our derived models in the mixed-domain is significantly better than the baselines. On the large-scale Yelp dataset, our derived models can also achieve some improvements. Moreover, these kinds of datasets we experimented with are very close to real-life applications, especially on the mixed-domain dataset. Therefore, we can conclude that our framework has good effectiveness and performance and is appropriate to apply in real-life e-commerce environments.

Due to the high flexibility of our framework, it can be further improved by integrating other better deep neural network algorithms and coarse-grained algorithms in the future. For example, as the LDA topic model used in this paper can only extract the topic information, we plan to apply some sentence-based or document-based models (Li et al., 2017; Ren & Ji, 2017) as coarse-grained feature extraction tools. Besides, as sentiment provides important information in the reviews, which should also be considered when extracting coarse features from reviews, the topic-sentiment joint probabilistic models (Dong et al., 2018) can also be combined with the deep learning models. Besides, only a specified classifier SVM is adopted in our framework in this paper. To further analyze the influence of classifiers on the performance of our framework, we will repeat our experiments by using other classifiers such as random forest, logistic regression, naive Bayes, and KNN.

Our experiments verified that our framework performs well in single or mixed domains such as hotels, restaurants, and doctors. It is interesting to further verify its performance by considering more domains such as education and health (Wu & Yu, 2020), in which the topics are much broader, the accuracy and recall are more relevant to data. Moreover, this accurate deceptive opinion filteration algorithm can also be used in other security and privacy fields (Ho, See-To, & Chiu, 2020; Hung et al., 2007; Wu et al., 2020; Wu, Chen, Wang, Meng, & Wang, 2019) and in realizing accurate influence analysis of products (Qiu, Yu, Fan, Jia, & Gao, 2019). We are also interested in applying user contextual information (Hong, Chiu, Shen, Cheung, & Kafeza, 2007) and reputation (Chiu, Leung, & Lam, 2009) as well as using Big Data (Liu, Sun, Wang, & Wu, 2019) and social network analysis (Su, Lin, Chen, & Lai, 2020) for deception detection.

CRediT authorship contribution statement

Ning Cao: Software, Writing - original draft. **Shujuan Ji:** Conceptualization, Methodology. **Dickson K.W. Chiu:** Writing - review & editing. **Mingxiang He:** Validation. **Xiaohong Sun:** Data curation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This paper is supported in part by the Natural Science Foundation of China (No. 71772107, 61502281), Qingdao social science planning project (No. QDSKL1801138), the National Key R&D Plan (No. 2018YFC0831002), Humanity and Social Science Fund of the Ministry of Education (No. 18YJAZH136), the Key R&D Plan of Shandong Province (No.2018GGX101045), the Natural Science Foundation of Shandong Province (Nos. ZR2018BF013), the Innovative Research Foundation of Qingdao (No. 18-2-2-41-jch), Shandong Education Quality Improvement Plan for Postgraduate, the Leading talent development program of Shandong University of Science and Technology and Special funding for Taishan scholar construction project.

References

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*(Jan), 993–1022.

Chen, Y., & Zhang, Z. (2018). Research on text sentiment analysis based on CNNs and SVM. In *2018 13th IEEE conference on industrial electronics and applications (ICIEA)* (pp. 2731–2734). IEEE.

Chiu, D. K. W., Leung, H. F., & Lam, K. M. (2009). On the making of service recommendations: An action theory based on utility, reputation, and risk attitude. *Expert Systems with Applications, 36*(2), 3293–3301.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2015). Gated feedback recurrent neural networks. In *International conference on machine learning* (pp. 2067–2075).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

Dong, L. Y., Ji, S. J., Zhang, C. J., Zhang, Q., Chiu, D. W., Qiu, L. Q., & Li, D. (2018). An unsupervised topic-sentiment joint probabilistic model for detecting deceptive reviews. *Expert Systems with Applications, 114*, 210–223.

Feng, S., Banerjee, R., & Choi, Y. (2012). Syntactic stylometry for deception detection. In *Proceedings of the 50th annual meeting of the association for computational linguistics: short papers (Vol. 2)* (pp. 171–175). Association for Computational Linguistics.

Fusilier, D. H., Cabrera, R. G., Montes, M., & Rosso, P. (2013). June). Using PU-learning to detect deceptive opinion spam. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis* (pp. 38–45).

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks, 18*(5–6), 602–610.

Ho, K. K., See-To, E. W., & Chiu, D. K. (2020). ‘‘Price Tag” of risk of using E-payment service. *Journal of Internet Commerce*. https://doi.org/10.1080/15332861.2020.1742482 (in press).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Hong, D., Chiu, D. K., Shen, V. Y., Cheung, S. C., & Kafeza, E. (2007). Ubiquitous enterprise service adaptations based on contextual user behavior. *Information Systems Frontiers, 9*(4), 343–358.

Hung, P. C., Chiu, D. K., Fung, W. W., Cheung, W. K., Wong, R., Choi, S. P., ... Cheng, V. S. (2007). End-to-end privacy control in service outsourcing of human intensive processes: A multi-layered Web service integration approach. *Information Systems Frontiers, 9*(1), 85–101.

Jia, S., Zhang, X., Wang, X., & Liu, Y. (2018). Fake reviews detection based on LDA. In *2018 4th international conference on information management (ICIM)* (pp. 280–283). IEEE.

Jindal, N., & Liu, B. (2008). Opinion spam and analysis. In *Proceedings of the 2008 international conference on web search and data mining* (pp. 219–230).

Jo, Y., & Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on web search and data mining* (pp. 815–824).

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (pp. 137–142). Berlin, Heidelberg: Springer.

Johnson, R., & Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv:1412.1058.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modeling sentences. arXiv preprint arXiv:1404.2188.

Kiliroor, C. C., & Valliyammai, C. (2019). Social network based filtering of unsolicited messages from e-mails. *Journal of Intelligent & Fuzzy Systems, 36*(5), 4037–4048.

Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. In Twenty-ninth AAAI conference on artificial intelligence.

Li, J., Cardie, C., & Li, S. (2013). Topicspam: A topic-model based approach for spam detection. In *Proceedings of the 51st annual meeting of the association for computational linguistics (Vol. 2: Short papers)* (pp. 217–221).

Li, J., Ott, M., Cardie, C., & Hovy, E. (2014). Towards a general rule for identifying deceptive opinion spam. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (Vol. 1: Long papers)* (pp. 1566–1576).

Li, K., Xie, J., Sun, X., Ma, Y., & Bai, H. (2011). Multi-class text categorization based on LDA and SVM. *Procedia Engineering, 15*, 1963–1967.

Li, L., Qin, B., Ren, W., & Liu, T. (2017). Document representation and feature combination for deceptive spam review detection. *Neurocomputing, 254*, 33–41.

Lin, C., & He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on information and knowledge management* (pp. 375–384).

Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101.

Liu, X., Sun, R., Wang, S., & Wu, Y. J. (2019). The research landscape of big data: A bibliometric analysis. Library Hi Tech, ahead of print. doi: 10.1080/15332861.2020.1742482.

Martinez-Torres, M. R., & Toral, S. L. (2019). A machine learning approach for the identification of the deceptive reviews in the hospitality sector using unique attributes and sentiment orientation. *Tourism Management, 75*, 393–403.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. (2013). What yelp fake review filter might be doing? *Seventh international AAAI conference on weblogs and social media*.

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies (Vol. 1)* (pp. 309–319). Association for Computational Linguistics.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., & Welling, M. (2008). Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 569–577).

Qiu, L., Jia, W., Liu, H., & Fan, X. (2019). Microblog hot topics detection based on VSM and HMBTM model fusion. *IEEE Access, 2019*(7), 120273–120281.

Qiu, L., Yu, J., Fan, X., Jia, W., & Gao, W. (2019). Analysis of influence maximization in temporal social networks. *IEEE Access, 7*, 42052–42062.

Ren, Y., & Ji, D. (2017). Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences, 385*, 213–224.

Ren, Y., Ji, D., & Zhang, H. (2014). Positive unlabeled learning for deceptive reviews detection. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 488–498).

Su, Y. S., Lin, C. L., Chen, S. Y., & Lai, C. F. (2020). Bibliometric study of social network analysis literature. Library Hi Tech, ahead of print. doi: 10.1108/LHT-01-2019-0028.

Tang, D., Qin, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1422–1432).

Wang, W., & Zhou, D. (2018). A multi-level approach to highly efficient recognition of Chinese spam short messages. *Frontiers of Computer Science, 12*(1), 135–145.

Wang, Y., Liu, Y., & Zhu, X. (2014). Two-step based hybrid feature selection method for spam filtering. *Journal of Intelligent & Fuzzy Systems, 27*(6), 2785–2796.

Wang, Z., & Qu, Z. (2017). Research on web text classification algorithm based on improved CNN and SVM. In *2017 IEEE 17th international conference on communication technology (ICCT)* (pp. 1958–1961). IEEE.

Wu, D., & Yu, F. (2020). Data for better health (Guest editorial). Library Hi Tech, 38 (2), ahead of print.

Wu, T. Y., Chen, C. M., Wang, K. H., Meng, C., & Wang, E. K. (2019). A provably secure certificateless public key encryption with keyword search. *Journal of the Chinese Institute of Engineers, 42*(1), 20–28.

Wu, T. Y., Lee, Z., Obaidat, M. S., Kumari, S., Kumar, S., & Chen, C. M. (2020). An authenticated key exchange protocol for multi-server architecture in 5G networks. *IEEE Access, 8*, 28096–28108.

Xian-yan, M., Rong-yi, C., Ya-hui, Z., & Zhenguo, Z. (2019). Multilingual short text classification based on LDA and BiLSTM-CNN neural network. In W. Ni, X. Wang, W. Song, & Y. Li (Eds.), *International conference on web information systems and applications* (pp. 319–323). Cham: Springer.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 1480–1489).

Yoo, K. H., & Gretzel, U. (2009). Comparison of deceptive and truthful travel reviews. In W. Höpken, U. Gretzel, & R. Law (Eds.), *Information and communication technologies in tourism* (pp. 37–47). Vienna: Springer.

Zhang, Q., Wang, H., & Wang, L. (2017). Classifying short texts with word embedding and LDA model. *Data Analysis and Knowledge Discovery, 32*(12), 27–35.