International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-017-0757-6

ORIGINAL ARTICLE

A hybrid model for opinion mining based on domain sentiment dictionary

Yi Cai[1] · Kai Yang[2] · Dongping Huang[1] · Zikai Zhou[1] · Xue Lei[1] · Haoran Xie[3] · Tak-Lam Wong[4]

Received: 1 April 2017 / Accepted: 1 December 2017

Abstract
Sentiment classification is an application of sentiment analysis, which is a popular research field in NLP. It can classify documents into different categories according to their sentiments. For a sentiment classification task, the first step is to extract sentimental features from documents, and then classify them using some classifiers. In the first step, a traditional way to extract sentimental features is to apply sentiment dictionaries. However, sentiment words may have different sentiment tendencies in different contexts, and traditional sentiment dictionaries does not consider this situation where wrong sentiment tendencies may be selected for sentiment words. In our research, we find that sentiment words will not have diverse meanings when they associate with the nearby aspects and entities in documents. Then, we propose a three layers sentiment dictionary, which can associate sentiment words with the corresponding entities and aspects together to reduce their multiple meanings. In the second step of the sentiment classification task, many classification models, such as SVM, GBDT, can be used to classify documents according to the extracted sentiment words. However, different classifiers have different weaknesses. A Stacking-based hybrid model is applied to combine SVM and GBDT together to overcome their weaknesses and reach higher performance. This hybrid model contains two layers, and the output of the first layer will become the input of the second layer. The first layer will generate different classification results according to different classifiers, while the second layer will automatically learn how to select a probable one as the final result. The experimental results show that our hybrid model outperforms the baseline single models.

Keywords Opinion mining · Hybrid model · Natural language processing

1 Introduction

With the development of the Internet, there is a huge demand on sentiment analysis and opinion mining [15]. Sentiment classification is one of the most popular research fields in opinion mining. It usually contains the following two steps: sentimental features extraction step and classification step.

In the sentimental features extraction step, a popular way is to apply sentiment dictionary to extract sentiment words as the features of documents. However, the traditional sentiment dictionaries have some drawbacks. As we know that some words may have different meanings according to their contexts. There exist some sentiment words that have different sentimental meanings when describing different things. For example, the word 'high' in two sentences shown in Table 1. When describing the quality of the car, 'high' shows positive sentiment. On the contrary, when describing the fuel-consumption, negative sentiment can be seen from 'high'. The traditional sentiment dictionary can only find out sentiment words like 'high', but ignore its different sentimental meanings in different contexts. This may result in information loss during the sentiment features extraction process, and finally decline the performance of sentiment classification.

To solve this problem, Liu et al. consider that sentimental meanings of sentiment words are associated with

The preliminary version of this article has been published in ASC 2017 conjunction with BIGCOMP 2017 [27].

✉ Yi Cai
ycai@scut.edu.cn

[1] South China University of Technology, Guangzhou, China

[2] City University of Hong Kong, Hong Kong, Hong Kong

[3] The Education University of Hong Kong, Hong Kong, Hong Kong

[4] Douglas College, New Westminster, Canada

Published online: 12 December 2017

Table 1 An example of multi-meaning sentiment words

Sentence	Sentiment
The quality of this product is high	Positive
The fuel-consumption of this car is high	Negative

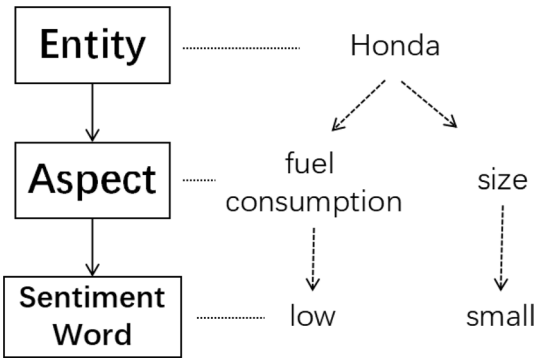

Fig. 1 Three layer model

entities of documents they describe [15]. Thus they propose a method to find out entities in the document at first, and then use a sentiment dictionary to find sentiment words corresponding to these entities. However, this way has a problem that the sentiment words may be not related to the nearby entities. For example, in the sentence "Honda's small car has low fuel consumption and small size", words 'low' and 'small' are not related to the entity 'Honda', but related to 'fuel consumption' and 'size' respectively. We call phrases like 'fuel consumption' and 'size', which belong to the entity 'Honda', as 'aspect'. Therefore, the sentiment words are not directly related to the entities, instead, they are more related to the aspects of these entities. That is to say, aspects of entities should also be considered when constructing sentiment dictionaries. Different from the traditional sentiment dictionary which only contain sentiment words, we construct a three-layer sentiment dictionary in this paper, where records are stored by a form of 3-tuples, for example, '(Honda, size, small)'. That is to say, the proposed sentiment dictionary does not only contain sentiment words, but also words or phrases which are aspects and entities. As shown in Fig. 1, the first layer contains words which are regarded as entities like 'Honda', and then the next layer is the aspects belonging to these entities like 'fuel consumption' and 'size'. Below the aspects, the last layer is the sentiment words related to these aspects. In example shown in Table 1, the sentiment word 'high' with different sentiment tendencies will be stored in the proposed dictionary using different tuples: '(Product, quality, high)' and '(Car, fuel-consumption, high)'. When sentiment words like 'high' will be extracted according to its nearby entities or aspects, they will less likely to have multi-meanings, thus these sentiment words can become important features for deciding the sentiment tendencies of documents.

However, different domains may have different entities or aspects. For example, there will not exist such aspect like 'fuel-consumption' in documents which domain is about sports. In addition, different domains may have different idiomatic or special sentiment words, which is hard to collected by some sentiment dictionaries from other domains. Therefore, it is necessary to construct a domain specific sentiment dictionary. Since there are many forums or blogs for different domains on the Internet, we construct the domain specific dictionary by extract '(Entities, Aspects, Sentiment words)' tuples from these external data.

After extracting features according to the proposed domain specific sentiment dictionary, the next step of sentiment classification is to classify documents using the extracted features. To classify documents into positive or negative sentiments, many traditional classification algorithms can be applied, such as Support Vector Machine (SVM) or Gradient Boosting Decision Tree (GBDT). However, these classification algorithms have their own strengths or weaknesses. SVM does not perform well when data is sparse, whereas GBDT tends to overfit data sets. Therefore, these weaknesses limit the performance of each models. A popular solution is to combine these single models together and construct a hybrid model, which is named as ensemble learning. However, how to select a proper results from these models is a challenge. In this paper, we use stacking approach, which proposes a automatic results selecting way, to combine SVM and GBDT together. The stacking approach contain two layers. In the first layer, several classifiers will be conducted and generate different classification results. These results will become the input features for a classifier in next layer, which will automatically learn how to select a most appropriate results. In the experiment, we compare the performance of the hybrid model with other baseline models. The result shows that our proposed hybrid model is better than other baseline models since it can overcome the weakness of single classifiers.

In this paper, we have the following contribution:

- A domain specific sentiment dictionary is constructed which can deal with problems caused by multi-meanings of domain sentiment words.
- A effective hybrid model, which combines different classifiers together, is proposed to improve the performance of sentiment classification by a automatical learning approach to select proper results from these classifiers.
- Several experiments are designed to demonstrate the effectiveness of our proposed hybrid model, and the

experimental result shows that the hybrid model outperforms the single models.

2 Related work

2.1 Sentiment analysis and opinion mining

There are many studies about sentiment analysis and opinion mining. Methods based on sentiment dictionary, or topic models have been commonly used in this field. Sentiment dictionary is most widely used to analyze sentiments [18]. This method judges opinions of documents according to the sentiment words in the dictionary. Liu et al. summarize the usage of sentiment analysis in [14], which declares that it can be applied at document sentiment classification, sentence subjectivity classification, sentiment lexicon generation, opinion summarization, and opinion search or retrieval, etc. In [1], sentiment analysis is introduced to process people's online social data, which can automatically capture the sentiments of the public about social events, marketing campaigns, product preferences or the financial market prediction.

There are many tools for sentiment analysis. WordNet [15] is the most widely used English dictionary, and HowNet [3] is a Chinese dictionary. Except for these dictionary-based methods, Lin et al. propose a sentiment analysis model based on topic model. Besides, Maas et al. construct word vectors according to the distribution of documents, and classify documents into positive or negative sentiment applying these word vectors. In addition, sentiment analysis can be applied in some semantic search schemes [6–8, 25]. Since the sentiment of the context can reveal semantic information to some extent, it can be taken into consideration to make semantic search more effective.

2.2 Classification algorithms

2.2.1 Support vector machine

Support Vector Machine (SVM) [19] is a new learning approach proposed by Vapnik et al. according to statistical learning theory. Its greatest feature is based on structural risk minimization criteria, constructing optimal classification hyperplanes by maximizing classification interval to improve the performance. It can solve problems caused by nonlinearity and high dimension of features [10]. SVM obtains the classes of samples by calculating the hyperplanes according to the samples in the region.

SVM aims to find the best decision hyperplane that separates data into different classes. In the two-category case, the basic idea behind training process is to search a decision hyperplane, represented by $\overrightarrow{w}$, that not only separates the data of one class from those of the other class, but also maximizes the distance (i.e. margin) between two hyperplanes defined by support vectors; letting $y_i \in \{0, 1\}$ be the correct class label of an input sample $\overrightarrow{x_i}$, the hyperplanes can be written as follows:

$$\overrightarrow{w_i} = \sum_{i=1}^{n} \alpha_i y_i \overrightarrow{x_i} \tag{1}$$

where the α_i's value is obtained by solving the dual optimization problem, and n is the number of input samples. The $\overrightarrow{x_i}$ is a support vector of the hyperplane $\overrightarrow{w}$ if and only if the α_i is greater than zero. And the classification procedure is to decide which side of the hyperplane that input data fall in. Compared with decision trees, SVMs are more capable of handling non-linear classification by implicitly mapping input into high dimensional feature spaces with appropriate kernels like Gaussian kernel, etc.

2.2.2 Gradient boosting decision tree

Gradient Boosting Decision Tree (GBDT) [4, 5] is also named as Multiple Additive Regression Tree (MATR). GBDT consists of multiple simple decision trees, the final prediction results is decided by the output of all simple decision trees. GBDT uses Gradient Boosting method to integrate all simple decision trees together. GBDT has high generalization ability, which performs well in classification when the number of feature is less than 400.

2.3 Term weighting schemes

Term weighting schemes have been used in Vector Space Model (VSM) to evaluate the weights of words in word vectors. They can be classified into supervised schemes and unsupervised schemes [12]. The supervised schemes exploit category information of training documents while unsupervised schemes do not. There are many unsupervised schemes widely used in Information Retrieval (IR) tasks, such as *tf*, $tf \cdot idf$ [22] and some variants [13, 17].

However, these schemes ignore the categories labels of each document. On the contrast, supervised schemes use the documents labeled with category information. Some supervised schemes are proposed recently, e.g., $iqf \cdot qf \cdot icf$ [20], *rf* [12] and some variants [11]. $iqf \cdot qf \cdot icf$ can be represented as:

$$iqf \cdot qf \cdot icf = log\left(\frac{N}{tp + fn}\right) \times log(tp + 1) \times log\left(\frac{|C|}{cf} + 1\right), \tag{2}$$

where tp is the number of documents that contain word w in the positive category, while fn is the number of documents

that contain word w in the negative category. cf represents the category frequency. N is the number of documents in the whole collection. $|C|$ is the number categories.

Wang et al. propose some entropy-based term weighting schemes such as *bdc* which are based on the entropy of terms in categories [24]. Wang et al. declare that *bdc* outperforms the state-of-the-art schemes, e.g. $tf \cdot idf$, $iqf \cdot qf \cdot icf$ and rf, in text categorization tasks.

2.4 N-gram model

N-gram model [2] is a probabilistic language model. A word can be predicted based on its previous N-1 words in a document. This is a Markov model [21] which assumes that the next word w_i depend on the probability of the previous N−1 words $w_{i-(N-1)}^{i-1} = \{w_{i-1}, \ldots, w_{i-(N-1)}\}$. The approximate prediction of a sequence w_1^n is calculated as follows:

$$P(w_1^n) = \prod_{k=1}^{n} P\left(w_k | w_{k-(N-1)}^{k-1}\right) \tag{3}$$

An N-gram is called unigram when $N = 1$, bigram when $N = 2$, and trigram when $N = 3$. Bigram and trigram model are mostly used in text classification tasks.

2.5 Word2Vec

Traditional opinion mining methods judge the opinion according to the grammar of documents, and regardless of the semantics of the document. Methods like Word2Vec [16] can be applied to supplement semantic features of documents.

Word2Vec is a toolkit based on deep learning technique [9], and is developed by Mikolov et al. [16]. The input of Word2Vec is a text corpus as input and the output is the word vectors. It first constructs a vocabulary from the training text corpus and then learns vector representation of words [26]. The result word vectors can reflect the semantics of the corresponding words. These word vectors can be used as features in many natural language processing applications. For this paper, the model has a window size of 5 and the dimension of word vectors is set to 100.

3 Overall framework

The main task of sentiment classification is to classify documents into different sentiments. It is a popular research field in opinion mining. In this paper, we propose a new sentiment classification model. Our work contain the following two parts: features extracting process and sentiment classification, which are shown in Fig. 2. We extract the relevant features at first, and then use the supervised machine learning algorithm for classification. In the features extraction step, sentiment words will be extracted at first. A three layer sentiment dictionary will be constructed in this step to extract sentiment words. Except for sentiment features, we apply other features of documents as a supplement, for example, features extracted by bag-of-word model or N-grams. For the reason that models like Word2Vec can reflect the semantic information of words in documents, we also apply Word2Vec model to generate more features for documents. After that, features of documents are used to train the hybrid classification model which combines SVM and GBDT together.

4 Features extracting process

4.1 Three layers model

Traditional sentiment analysis methods are based on sentiment dictionary. However, according to our observation, there exist some extracted sentiment words that are not related to the main entity of the documents. These words will result in poor performance in sentiment classification tasks. Liu et al. propose a method to find out entities in the document, and

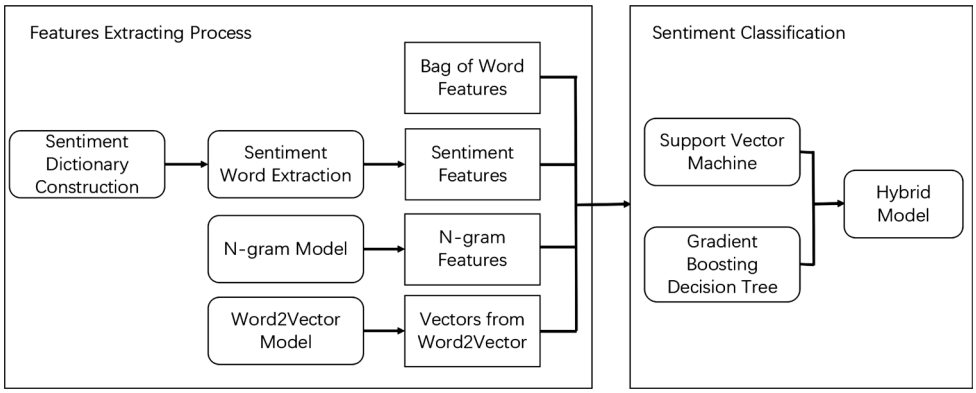

Fig. 2 Overall framework

then use sentiment dictionary to find out sentiment words that related to these entities [15]. However, the sentiment words may be not directly related to the nearby entities. For example, in the sentence 'Honda's small car has low fuel consumption and small size', words 'low' and 'small' are not related to the entity 'Honda', but related to phrases 'fuel consumption' and 'size' respectively. We call phrases like 'fuel consumption' and 'size', which belong to the entity 'Honda', as 'aspect'. Therefore, the sentiment words are not directly related to the entities, instead, they are more related to the aspects of these entities. Since traditional sentiment dictionary only contain sentiment words, they cannot be used to find out entities or aspects from documents. In this paper, we construct a three-layer sentiment dictionary, which not only contain sentiment words, but also aspects and entities. The structure of this dictionary is shown in Fig. 1. The first layer contain words which are regarded as entities, and then the next layer is the aspect that belong to these entities. The last layer contains the sentiment words that related to the corresponding aspects.

To construct the three-layer dictionary, a training dataset is needed. Traditional dictionaries are constructed based on some commonly used dataset. However, the relation between entities, aspects and sentiment words may be different in different domain, for the reason that different domains have different idiomatic expressions. In addition, the common-used training sets do not contain the newly-born buzzwords. Therefore, when a document from a specify domain contains idiomatic expressions or the newly-born buzzwords, these words cannot be extracted from the sentiment dictionary, which will result in the poor performance in sentiment analysis. According to our observation, most of specify domains have their corresponding forums on the internet. Those idiomatic expressions and newly-born buzzwords can be found in these forums. Therefore, in this paper, we develop web spiders to obtain textual data from forums to construct a domain specify dictionary for opinion mining. For example, to analyse the sentiment of document from a specify domain about 'car', we obtain data from a famous car forum 'auto home' (http://www.autohome.com.cn). After obtain the textual data from this forum, term weighting schemes like $tf - idf$ will be conducted, and we will obtain the weights of words in the dataset. $tf \cdot idf$ can be represented as follows:

$$tf \times idf = f_{t,d} \times log\frac{N}{n_t}, \tag{4}$$

where $f_{t,d}$ is the frequency of word t in document d. N is the total number of documents and n_t is the number of documents contain word t. Words with high weights represent that these words are important in the forum, thus they can be selected and put into the domain dictionary.

4.2 Entity and aspect extraction

Given a document, we find out entities words first, and then find out aspects of the entities. Secondly, sentiment words will be found to describe the sentiments of aspects. Finally, we obtain the sentiments of entities by combining sentiments of their aspects together. Our proposed three layers model can solve the problem caused by ambiguous words. For example, in car field, the word 'high' may have two different sentiments. 'high fuel consumption' express a negative opinion, while 'high chassis' is positive. In our proposed three layers model, the sentiment words will be attached to the corresponding aspects. Hence, 'high' will be considered a positive sentiment words when it belongs to aspect 'chassis', but it will become a negative sentiment words for aspect 'fuel consumption'.

4.3 Sentiment words extraction

After obtaining aspects, we aim to find out sentiment words related to these aspects. There are many ways to extract sentiment words. We apply syntax parsing algorithms to find out words which have subject, attribute or predicate relation with aspects words. Normally, these words are the sentiment words. For example shown in Fig. 3, we use parsing algorithms to find out the structure of the Chinese sentences. The meaning of the sentence '因为其油耗低, 体积小, 所以常见于本田的小型汽车上。' is: "Since its low fuel consumption and small size, it is common in Honda's small car". And then adjectives and adverbs will be discovered and regarded as sentiment words. Hence, in the example, words '低 (low)' and '小 (small)' are the sentiment words. These sentiment words will be stored in the dictionary using

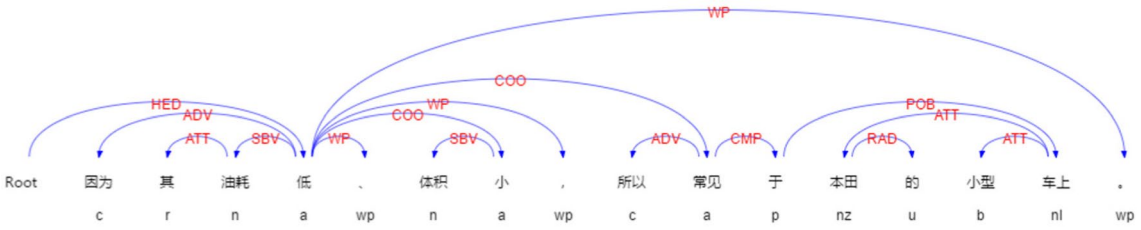

Fig. 3 An example about sentiment words extraction

the form of 3-tuples: '(Entity, Aspect, Sentiment word)'. In the example above, sentiment words like '低 (low)' and '小 (small)' will be constructed into the following format: '(本田 (Honda), 油耗 (fuel-consumption), 低 (low))','(本田 (Honda), 体积 (Size), 小 (small))'.

Traditional sentiment dictionary like Hownet can only identify sentiment words, but it can not reveal the emotional intensity of these words. However, the emotional intensity is also important for the sentiment classification task. For example, word 'best' can express stronger emotion than the word 'good', thus documents with the word 'best' are more probable to express positive emotion than that with the word 'good'. To consider emotional intensity into our sentiment dictionary, we count the frequency of words belonging to positive or negative emotion. Then, we calculate the probability of these words appearing in documents with positive and negative sentiments. Words with higher probability will have stronger emotional intensity. The examples are shown in Table 2. The word '百看不厌' (Never boring) has high probability belonging to positive words, thus it have strong emotional intensity.

4.4 Other features for classification

As is shown in Fig. 2, we select the following features as the input of the classifiers.

- Bag of Word Model: The Bag of Word Model is a simplifying representation in natural language processing. In this model, a document is regarded as a disorder set of vocabulary, which ignores the order between words in the document. We apply term weighting schemes to extract the most important words for documents, and these words will become features to represent the corresponding documents.
- Extracted Sentiment Words: We extract sentiment words applying the constructed three layer dictionary in Sect. 4.
- N-gram: A single word in some phrases may not reflect the sentiment tendency of the document, and all words in phrases should be integrated together. We combine these words together and generate a N-gram, where N is the length of the phrases.
- Vectors generated by Word2Vec: Semantics information can also reflect the sentiment of a document. Word2Vect can transfer a document into a vector which can reflect the semantics of the document.

The reason we apply N-gram as features is that unigram will split phrases into different single words, and the original sentiment tendency of the phrases will change when we see these single words. For example, the Chinese phrases '不可靠' ('not reliable') will be split into words '不' ('not') and '可靠' ('reliable'). For the reason that '可靠' ('reliable') is a positive word, and phrases '不可靠' ('not reliable') is a negative phrases, if we only apply unigram '可靠' ('reliable') as the feature, it is hard to obtain a accurate result. To solve this problem, we combine the neighboring words into bigrams, and count the frequency that these bigrams appear in positive and negative documents in the training set. In addition, we calculate the probability of these bigrams belongs to positive and negative sentiments. The example is shown in Table 3. In the first bigram, words '颜值' ('looking') and '高' ('good') are combined into a bigram '颜值+高' ('good looking'). This bigram appear 2057 times in documents with positive sentiment and 10 times in negative sentiment. Therefore, we can calculate the its probability that belongs to positive sentiment is 0.9952, and that of negative sentiment is 0.0048. The four dimensions in Table 3

Table 2 Emotional intensity of sentiment words

	Word	Positive frequency	Negative frequency	Positive probability	Negative probability
1	舒适 (Comfortable)	16995	2682	0.8638	0.1363
2	漏油 (Oil leak)	115	1921	0.0565	0.9435
3	百看不厌 (Never boring)	305	1	0.9967	0.0033

Table 3 Features of bigrams

	Bigrams	Positive frequency	Negative frequency	Positive probability	Negative probability
1	'颜值' + '高' (good looking)	2057	10	0.9952	0.0048
2	'拉' + '风' (cool)	1450	36	0.9758	0.0242
3	'发动机' + '噪声' (Engine noise)	23	275	0.0772	0.9228

are regarded as features to classify documents into different sentiments.

However, N-gram model only take the sequence of words in documents into consideration, and ignore the semantics of these words. Therefore, we apply Word2Vec to supplement the features in the semantic perspective. Word2Vec is a useful tool to represent words with vectoxrs. It use deep learning technique to map a word into K-dimension vector space, which provides a useful way to represent textual data. Documents with similar vector will be regarded as semantically similar. For the reason that sentiment information belongs to semantic information to some extent, vectors generated by Word2Vec can reflect the sentiment information. In this paper, we use the training set to train a Word2Vec model. Then, words will be transfer into K-dimension vectors. After that, we add all word vectors belonging to the same document together and obtain a new vector which can represent the document. The K-dimension will act as sentiment features and be input into the classifiers.

5 Sentiment classification model

The main task of sentiment analysis is to classify documents into different sentiments, thus it is a classification task. In this section, we propose a hybrid sentiment classification model combining SVM [19] and GBDT [4] together. Since the SVM we used is slightly different from the traditional SVM, we introduce the SVM basing on tf-bdc first. And then the hybrid model will be introduced.

5.1 SVM model basing on tf-bdc

Wang et al. propose a entropy-based term weighting schemes, called balanced distributional concentration (*bdc*) [24]. *bdc* can be calculated by the following equation.

$$bdc(t) = 1 - \frac{BT(t)}{log(|C|)} = 1 + \frac{\sum_{i=1}^{|C|} \frac{p(t|c_i)}{\sum_{i=1}^{|C|} p(t|c_i)} log \frac{p(t|c_i)}{\sum_{i=1}^{|C|} p(t|c_i)}}{log(|C|)} \tag{5}$$

where $|C|$ is the number of categories and c_i is the ith category. *bdc* can measure the important of a word to a category. If a word is assigned high *bdc* value, the ability of this word distinguishing other categories is high, thus this word will be important in the classification task. In addition, if a word have high frequency in a document, this word is also important for the classification task. Therefore, we combine term frequency (*tf*) and *bdc* together to evaluate the importance of a word in the classification process, and get $tf - bdc$, which is calculated as follows:

$$tf - bdc = tf \times bdc \tag{6}$$

5.2 Hybrid model

There are many single models dealing with the classification task. For example, GBDT consists of multiple simple decision trees, and the results of all trees are superimposed to create the final predictions. It uses Gradient Boosting algorithm [4] to iteratively optimize the final results. Thus it has high generalization ability, and is suitable for the classification tasks whose number of features is less than 400.

However, the single models like SVM and GBDT have some defaults. SVM performs well when classifying sentences which have simple structure and strong opinion tendency, but it has poor performance for those complicated sentences. On the other hand, GBDT performs well for long sentences with many sentiment words.

We use an example shown in Table 4 to make this clear. In this example, the first sentence has strong opinion tendency, thus SVM can get the correct sentiments. The third sentence is a long sentence, and it contains many sentiment words. In this case, GBDT gets the correct result, while SVM gets the wrong result. Hence, these two models have both strengths and weaknesses. If these two model are combined together and overcome their own weaknesses, we can obtain a highly effective sentiment analysis model.

As shown in examples of Table 4, different models are good at dealing with different situations. Stacking approach [23] can be applied to learn the situations that these models are good at. Then, when facing a specific situation, a result

Table 4 Comparison of SVM and GBDT

ID	Sentences	SVM	GBDT	Correct
1	轩逸：天呐，我上榜了，我要发朋友圈去！(XuanYi: God! I get on the list! I want to share it with my friends!)	pos	neg	pos
2	【大事件】长安cs75—1.5t 劲越登场 ([Big News] Changan cs75-1.5t has launched to market.)	pos	pos	pos
3	中控台上有着8 英寸的触摸屏也有着出色的质感，无论是 触摸操作的反应速度还是显示的效果都对得上豪华二字， 不仅如此，冠道为了兼顾各种使用情况特意将屏幕设计成多角度可调，无论多刺眼的阳光都不会影响使用 (The center console has an 8-inch touch screen and also has a good texture, which looks luxury; moreover, the screen specially designed to Multi-angle adjustable in order to consider the various circumstances, thus no matter how dazzling the sun will not affect the use of the screen.)	neg	pos	pos

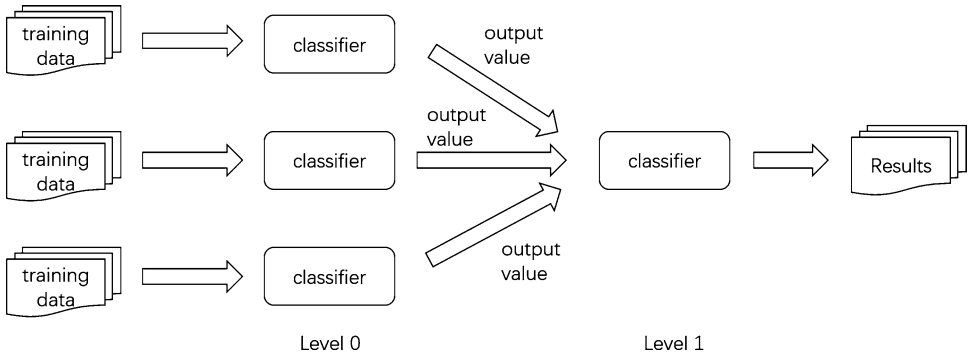

Fig. 4 Training process of Stacking

from the model which are good at dealing with this situation will be selected. In this paper, we combine SVM and GBDT based on Stacking approach. It is a two layer model, as shown in Fig. 4. In the first layer, single classifiers will be conducted and generate the corresponding results. These results will become the input features of the classifier in the second layer, which will generate a final classification result.

In this paper, we apply SVM and GBDT respectively in Level 0. Then, in Level 1, SVM model are also applied. The training set is separated into two equally parts. One part is used to train SVM and GBDT in Level 0, and the other part is to train the SVM model in Level 1. The classifiers in Level 0 will output the classification results of documents belonging to positive or negative sentiments. In Level 1, the output value of classifiers in Level 0 will be input into another classifier, linear SVM, and the final prediction will be obtained through the output of linear SVM. For example, SVM and GBDT in Level 0 may get the result 'positive' and 'negative' respectively, and then the input features of SVM in Level 2 is < 'positive', 'negative' >. Finally, SVM in Level will output a final result according to the input features.

6 Experiments

6.1 Evaluation metric

We use Precision, Recall and F1-measure as our evaluation metric. The outputs of our model are the entity-sentiment pair. We find out entities from documents, and assign each entity a sentiment. The Precision value is calculated as follows:

$$P = \frac{tf}{fp + fn_2}, \tag{7}$$

where P is the Precision value, and tp is the number of correct entity-sentiment pairs, while fp is that of incorrect entity-sentiment pairs. fn_2 is the number of incorrect entities. The Recall value is calculated as follows:

$$R = \frac{tf}{fp + fn_1}, \tag{8}$$

where R is the Recall value and fn_1 is the number of entities that have not been found out. The F_1 value is calculated as follows:

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{9}$$

6.2 Dataset

A dataset in a specify domain 'automobile' is used in this paper to illustrate the effectiveness of our proposed model. This dataset is derived from Chinese news from automotive field, and the main task of our paper is to analyse the sentiment tendency of these news. We obtain this dataset from some famous website in China using web crawler technique, Scrapy, which is a crawler framework developed by Python and released in https://github.com/scrapy/scrapy. Applying Scrapy, we obtain totally 35,100 texts talking about automobiles. Then, these texts are manually labeled into positive, neutral or negative according to their sentiment tendency. The distribution of the labeled data is as follows: 24.8% of texts are labeled with positive, 66.71% are labeled with neutral and 8.49% are labeled with negative. Examples of the positive, neutral and negative samples are shown in Table 5. Finally, these data are divided into two set, where 80% of them are assigned to training set, and 20% of them are testing set.

Table 5 Examples of dataset

Texts	Label
长安汽车旗下全新MPV凌轩即将耀世登场 (Changan Automobile's new MPV Ling Xuan is about to debut.)	Positive
汽车之家网为您提供全国上汽大众经销商 (Autohome.com will provide you with the national SAIC dealer.)	Neutral
最终我们放弃了迈腾 (Eventually we gave up MAGOTAN.)	Negative

6.3 Data preprocessing

Data preprocessing is a necessary process before analysing textual data. The reason is that textual data is unstructured, and contain a lot of dirty data which will have negative effect on the analysing results. The main task of data preprocessing is to improve the quality of data by eliminating dirty data. In addition, English texts contain spaces between words, which make it easy to extract words. However, there do not exist any spaces between words in Chinese text. Therefore, word segmentation process is necessary in data preprocessing.

The main steps for data preprocessing is shown as follows:

- Domain Data Extension: extend textual data in automotive field using crawler technique.
- Data Cleaning: eliminate stop words and structure data.
- Chinese Words Segmentation: segment words after cleaning data.

The first step is to extend domain data. In order to construct the domain specific sentiment dictionary, we obtain external data from a automotive forum 'auto home'. This forum contains a lot of reviews about cars, and each review contain a ranking given by the user. It is reasonable that reviews with high rankings show positive emotion, while that with low rankings are negative emotion. Therefore, we label the reviews with high rankings as positive sentiment and that with low rankings as negative. These labeled data will be applied to construct the sentiment dictionary and enrich the training set.

To obtain these data, we construct a multithreading and high concurrency web crawler framework based on Scrapy and a unstructured database, Redis. The structure of our crawler framework is shown in Fig. 5. It contains five parts: Scrapy Engine, Scheduler, Downloader, Spiders and Item Pipeline. Scrapy Engine controls the operation of the other four parts. To obtain a resource from Internet, the Scheduler will first send a request to Downloader, and this task will enter the queue of Downloader. Then, the downloader will obtain the corresponding resources, and deliver it to the Spiders. After that, the obtain data will be processed and put into the Pipeline. Users can obtain these data from the Item Pipeline. In order to store these data, Redis is applied, which is a high-performance in-memory database. It stores most of data into the memory, thus it can fast access data and provide stable data storage service meanwhile. Therefore, data crawled by Scrapy will be stored in Redis. In general, our crawler framework obtains about 320,000 reviews from auto-home.

In the second step, we first eliminate stop words from the dataset according to a list of stop words. In addition, some parts in the Chinese news are useless for judging their sentiments.

The third step is to segment Chinese word. The task of Chinese word segmentation is to segment continuous Chinese characters into individual words. In English texts, there are spaces between words as the separators. However, Chinese texts do not contain these separators. Thus, Chinese words segmentation is necessary when processing Chinese texts. We apply a Chinese words segmentation tool, called Jieba, which has been widely used in Chinese NLP tasks and can be accessed free from https://pypi.python.org/pypi/jieba/. It uses dynamic programming algorithm to find out the suitable Tire-Tree route and the maximum word segmentation based on word frequency.

6.4 Comparison of hybrid model with baseline models

In order to verify the effectiveness of our proposed hybrid model, we design several baseline models for comparison. Since radial basis function and linear kernel function are two of most popular function for SVM model, we apply them as the baseline models in the experiment. The baseline models we apply are as follows: (a) Baseline model 1:

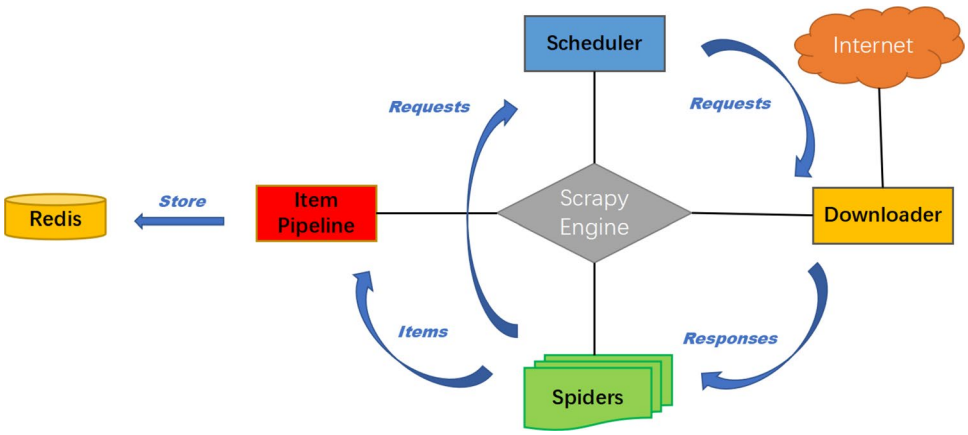

Fig. 5 Crawler framework

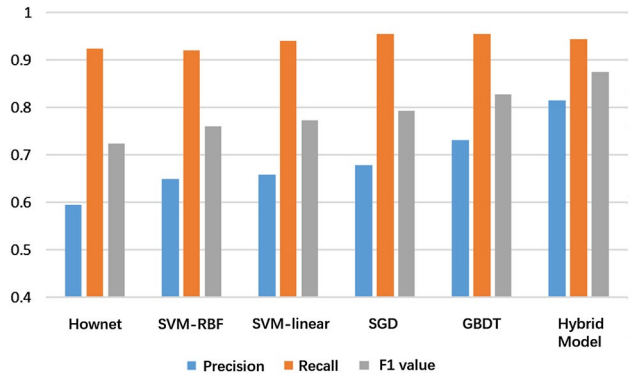

Fig. 6 Comparison of hybrid model with baseline models

This model only applies a commonly used dictionary, called Hownet [3], to analyze sentiments; (b) Baseline model 2: This model only applies SVM model with radial basis function (RBF); (c) Baseline model 3: This model only applies SVM model with linear kernel function; (d) Baseline model 4: This model only applies SDG using Bagging as optimization method; (e) Baseline model 5: This model only applies GBDT.

The experimental results are shown in Fig. 6. From the experimental results, baseline model based on dictionary performs worst. We explore the reason and find that domain words in car field are not contained in Hownet. Therefore, many sentiment words in the automobile field cannot be found, which results in the bad performance of this model. Besides, single SVM models also have poor performance, for the reason that the training dataset contains many sparse vectors. However, SVM with radial basis function (RBF) tends to over-fit, thus SVM with linear kernel function outperform that with RBF. GBDT uses Boosting approach to integrate decision trees and reach the highest performance among baseline models. However, it tends to over-fit, thus this method also has room for improvement. The experimental result shows that the Hybrid model outperforms the baseline models. We explore the reason why the hybrid model performs better. We know that different single classification models are good at dealing with different kinds of data. The Stacking approach we applied in hybrid model can learn from data and know which classifier is more appropriate for a specific kind of data. Therefore, the Stacking approach can select a proper result from a classifier which is good at dealing with the corresponding kind of data. For example, Stacking approach can learn that when data is sparse, the precision of SVM becomes low, while that of GBDT is high. Thus when dealing with sparse data, the results of GBDT will be chosen.

6.5 Comparison of regulation-based and stacking-based hybrid models

Our proposed model applies Stacking technique to combine different single models. However, there exist other approach to construct the hybrid models, for example, models can be merged together according to some regulations. To see which hybrid models perform well, we construct a Regulation-based hybrid model and compare its performance with the Stacking-based one in this sub-section.

The Regulation-based hybrid model is based on the following regulations. (a) If GBDT can obtain a certain result, this result will become the final result. (b) If GBDT obtain a uncertain result, SVM will be conducted, and the result of SVM will be output as the final result.

The experimental results are shown in Fig. 7. Apparently, Stacking-based model performs better than regulation-based one. The performance of regulation-based model even worse than that of single GBDT model. On the other hand, the results also show that the Stacking-based hybrid model outperforms the single models. We try to explore the reason that stacking-based model is better than regulation-based model. As we have discuss before, the stacking-based hybrid model will choose a proper results form different single models according to the characteristic of input data. That is to say, in different situations, different strategies will be applied to choose proper classification results. To some extent, this is similar to the regulation-based model. However, the difference is that the strategies for the regulation-based model are pre-defined, while that for the stacking-based model are automatically learned from data. In this way, the stacking-based model can find out some latent regulations which are not intuitive for humans.

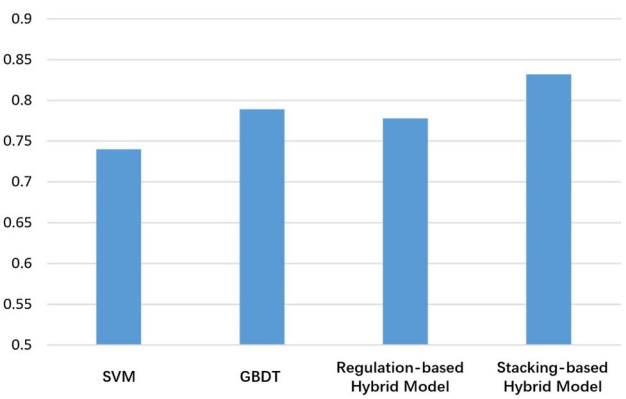

Fig. 7 Comparison of regulation-based and stacking-based hybrid models (F1 values)

6.6 The effect of different features on classification results

In Sect. 4.4, we discuss the effect of different features on the sentiment classification tasks. In this sub-section, we conduct several experiments to find out a best combination of features for our proposed model. We use the following features in this experiment: unigrams, bigrams, sentiment words, vectors obtained by Word2Vec.

As is shown in Table 6, features with ticks are applied in the corresponding settings. We compare the performance of our proposed model using different settings. The experimental results are shown in Fig. 8. We can get the following observation:

- The performance of bigrams (Setting 2) is higher than that of unigrams (Setting 1), and the combination of these two model (Setting 6) outperforms each individual ones. The reason why bigrams perform better is that it can reduce the ambiguity caused by individual words.
- The performance of our model increases by 2% when adding Word2Vec features (Setting 4). The reason is that these features bring semantic information into the classification process.
- After considering the sentiment words, the performance of models increases by 1.8% (Setting 8).

In general, the proposed model using all the features has best performance. Since sentiment words or Word2Vector can both reflect the semantic information of the textual data, these features can improve the performance of model. In addition, bigrams performs best between the four features when they are applied individually. Since bigrams can reflect the context information of a document, it is also important for sentiment classification task.

7 Conclusion

In this paper, we first construct a domain specific sentiment dictionary, so that some idiomatic expression or newly-born buzzwords in the domain can be contained in this dictionary. In addition, this dictionary has the following three layer structure: entities, aspects and sentiment words. Thus the extracted sentiment words can be associated with entities. Besides, we apply N-gram, Word2Vec to extend features. Then, these features will be put into classifiers. Since single classifiers like SVM or GBDT have their strengths or weaknesses. To overcome the weaknesses of single models, we apply Stacking approach to combine SVM and GBDT together, and reach a better performance. We also conduct several experiments to demonstrate the effectiveness of our proposed models. The first experiment is to compare the performance of hybrid model with baseline model like single SVM and GBDT. The second experiment is the comparison of Regulation-based or

Table 6 Combination of different features

	Unigram	Bigram	Sentiment words	Word2Vector
Setting 1	√			
Setting 2		√		
Setting 3			√	
Setting 4	√	√		√
Setting 5				√
Setting 6	√	√		
Setting 7	√			√
Setting 8	√	√	√	√

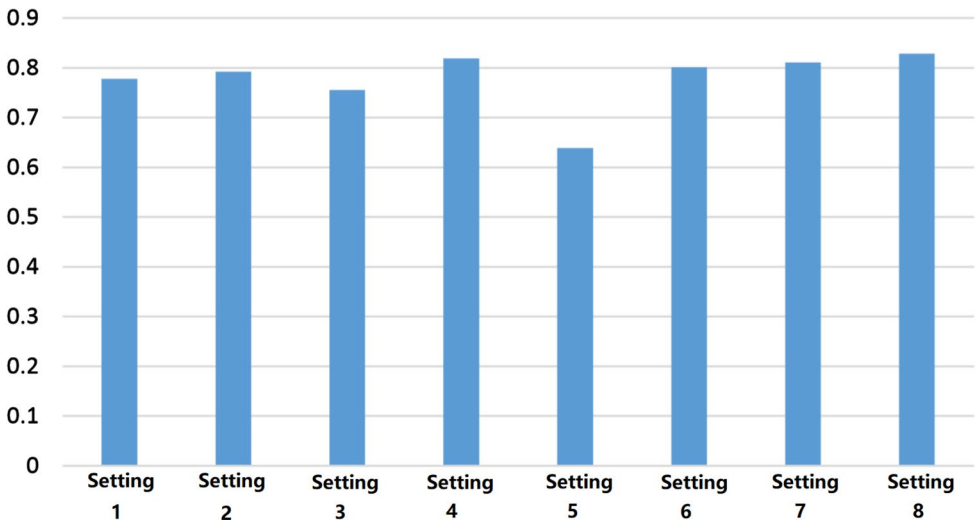

Fig. 8 Comparison of different combination of features (F1 values)

Stacking-based Hybrid model. We also compare the effect of different features on classification results.

Acknowledgements This work is supported by the Fundamental Research Funds for the Central Universities, SCUT (No. 2017ZD048), Tiptop Scientific and Technical Innovative Youth Talents of Guangdong special support program (No. 2015TQ01X633), Science and Technology Planning Project of Guangdong Province, China (No. 2017B050506004), Science and Technology Program of Guangzhou (International Science & Technology Cooperation Program No. 201704030076), and the Internal Research Grant (RG 66/2016-2017) and the Funding Support to ECS Proposal (RG 23/2017-2018R) of The Education University of Hong Kong.

References

1. Cambria E (2016) Affective computing and sentiment analysis. IEEE Intell Syst 31(2):102–107
2. Cavnar WB, Trenkle JM et al (1994) N-gram-based text categorization. Ann Arbor MI 48113(2):161–175
3. Dong Z, Dong Q (2006) Hownet and the computation of meaning. World Scientific, Singapore
4. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29:1189–1232
5. Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378
6. Fu Z, Huang F, Sun X, Vasilakos A, Yang C-N (2016) Enabling semantic search based on conceptual graphs over encrypted outsourced data. IEEE Trans Serv Comput PP:1–1
7. Fu Z, Ren K, Shu J, Sun X, Huang F (2016) Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Trans Parallel Distrib Syst 27(9):2546–2559
8. Fu Z, Wu X, Guan C, Sun X, Ren K (2016) Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Trans Inf Forensics Secur 11(12):2706–2716
9. Goldberg Y, Levy O (2014) word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722
10. Hofmann T (1999) Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '99. ACM, New York, NY, USA, pp 50–57
11. Ko Y (2012) A study of term weighting schemes using class information for text classification. In: Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval, ACM, pp 1029–1030
12. Lan M, Tan CL, Su J, Lu Y (2009) Supervised and traditional term weighting methods for automatic text categorization. Pattern Anal Mach Intell IEEE Trans 31(4):721–735
13. Leopold E, Kindermann J (2002) Text categorization with support vector machines. How to represent texts in input space? Mach Learn 46(1–3):423–444
14. Liu Bing (2012) Sentiment analysis and opinion mining. Synth Lect Hum Lang Technol 5(1):1–167
15. Liu B, Hu M, Cheng J (2005) Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of the 14th international conference on world wide web, WWW '05. ACM, New York, NY, USA, pp 342–351
16. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: NIPS'13 Proceedings of the 26th international conference on neural information processing systems, vol 2, 5–10 Dec 2013, Lake Tahoe, Nevada, pp 3111–3119
17. Paik JH (2013) A novel tf-idf weighting scheme for e ective ranking. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, SIGIR '13. ACM, New York, NY, USA, pp 343–352
18. Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1–2):1–135
19. Papadimitriou CH, Tamaki H, Raghavan P, Vempala S (1998) Latent semantic indexing: a probabilistic analysis. In: Proceedings of the seventeenth ACM SIGACT–SIGMOD–SIGART symposium on principles of database systems, ACM, pp 159–168
20. Quan X, Wenyin L, Qiu B (2011) Term weighting schemes for question categorization. Pattern Anal Mach Intell IEEE Trans 33(5):1009–1021
21. Rabiner L, Juang B (1986) An introduction to hidden Markov models. IEEE ASSP Mag 3(1):4–16
22. Sparck Jones K (1972) A statistical interpretation of term specificity and its application in retrieval. J Doc 28(1):11–21
23. Wang G, Hao J, Ma J, Jiang H (2011) A comparative assessment of ensemble learning for credit scoring. Expert Syst Appl 38(1):223–230
24. Wang T, Cai Y, Leung H, Cai Z, Min H (2015) Entropy-based term weighting schemes for text categorization in VSM. In: Tools with artificial intelligence (ICTAI), 2015 IEEE 27th international conference. IEEE, Vietri sul Mare, Italy, pp 325–332
25. Xia Z, Wang X, Sun X, Wang Q (2016) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans Parallel Distrib Syst 27(2):340–352
26. Xue B, Fu C, Shaobin Z (2014) A study on sentiment computing and classification of sina weibo with word2vec. In: Big Data (BigData Congress), 2014 IEEE international congress. IEEE, Anchorage, AK, USA, pp 358–363
27. Yang K, Cai Y, Huang D, Li J, Zhou Z, Lei X (2017) An effective hybrid model for opinion mining and sentiment analysis. In: Big data and smart computing (BigComp), 2017 IEEE international conference. IEEE, Jeju, South Korea, pp 465–466