Mao, Y.S., Zhang, L.Y., & Li, Y.R. (2019). Finding product problems from online reviews based on BERT-CRF model. In *Proceedings of The 19th International Conference on Electronic Business* (pp. 231-241). ICEB, Newcastle upon Tyne, UK, December 8-12

Finding Product Problems from Online Reviews Based on BERT-CRF Model

(Full Paper)

Yusheng Mao, Wuhan University, China, yushmao95@163.com
Liyi Zhang*, Wuhan University, China, lyzhang@whu.edu.cn
Yiran Li, Wuhan University, China, yiran_li94@sina.com

ABSTRACT

Online product reviews contain a lot of valuable information regarding product problems, which are very useful for producers to find product pain points and improve product quality. However, many studies focus only on the sentiment polarity of the product aspect, ignoring specific product problem information in online reviews. In this paper, product aspects and specific problem information are extracted from online reviews to help producers find the specific pain points of products. We call this task *Review Problem Mining* (RPM). At the same time, existing methods of review information extraction depend heavily on manually constructed features or large-scale data. To address these limitations, we proposed a new joint model BERT-CRF which integrates the popular pre-trained language model BERT and conditional random fields (CRF). The proposed method introduces external knowledge through BERT to reduce the model's dependence on training data and uses CRF to model the dependencies among tags. To verify the validity of our method, we constructed a dataset from JD.com and carried out the experiments. Experimental results show that the proposed method is highly effective.

Keywords: Review problem mining; information extraction; BERT; CRF; deep learning; natural language processing.

*Corresponding author

INTRODUCTION

With the development of internet technology and the popularization of e-commerce, more and more consumers like to shop online and post reviews on specific products. Therefore, there are a large number of reviews on e-commerce platforms. Online reviews provide consumers with more product information, which helps them make more informed purchasing decisions. Producers can also get timely feedback from consumers through online reviews, which can help producers understand consumer needs, find product pain points, and promote product innovation (H. Zhang, Rao, & Feng, 2018). However, it is difficult for ordinary people to process a massive volume of online reviews and extract consumer opinions in them. Thus, automatic review processing and opinion extraction techniques are needed. Sentiment analysis grows out of this need. Sentiment analysis (also called opinion mining or review mining) is the task of detecting, extracting and classifying opinions, sentiments, and attitudes concerning different topics, as expressed in textual input (Montoyo, MartíNez-Barco, & Balahur, 2012). It helps achieve a variety of goals, such as obtaining market intelligence, measuring customer satisfaction, identifying consumer needs, and more (Ravi & Ravi, 2015).

Researchers mainly study sentiment analysis from three levels: document-level, sentence-level, and aspect-level (L. Zhang, Wang, & Liu, 2018). The document-level and sentence-level sentiment analysis can identify the sentiment polarity of a document or sentence. However, consumers usually express different opinions on different aspects of the product in their reviews (e.g. "The screen is nice, but the performance is poor"), and document-level and sentence-level sentiment analysis cannot provide such detailed opinion information. Therefore, aspect-based sentiment analysis (ABSA) has attracted the attention of many researchers (Schouten & Frasincar, 2015). Compared with document-level and sentence-level sentiment analysis, aspect-based sentiment analysis is more fine-grained, it can identify the sentiment polarity of consumers to product aspects. (Hu & Liu, 2004).

However, Liu (2015) pointed out that for higher analysis needs, it is not enough to identify aspects and the polarities of each aspect only, more detailed analysis of reviews is needed. For example, in the sentence "The sound quality of the mobile phone is bad, there are some murmurs." the "sound quality" is a product aspect and "bad" is a sentiment word, aspect-based sentiment analysis techniques can identify consumers' negative opinion about the sound quality of the mobile phone. However, it is difficult for the producer to improve the product only based on the negative opinion, they need to know the specific problems of the product. In this sentence, we call " there are some murmurs " as a problem of the sound quality. In practical application, it is of great significance to identify problems in reviews, because producers can understand the specific pain points of products and services according to the problems, make targeted decisions to improve products and services quality (Liu, 2015).

Therefore, product aspects and specific problem information are extracted from online reviews in this work (e.g. Sound quality: There are often murmurs), and we call the task *Review Problem Mining* (RPM). Compared to aspect-based sentiment analysis, which only concerns sentiment polarity in aspect-level, RPM concentrates on the extraction of specific problems in reviews. Therefore, RPM can provide producers with more fine-grained and targeted product problem information to help them improve their products.

Review information extraction is typically modeled as a sequence labeling task. The most commonly used model is conditional random fields (Lafferty, McCallum, & Pereira, 2001). However, the effect of conditional random fields (CRF) model depends heavily on manually constructed features, the process of constructing features is inefficient. With the development of deep learning technology, many studies apply deep learning to review information extraction recently (L. Zhang, Wang, & Liu, 2018). Deep learning models can automatically learn the features of the text, but they also require large-scale data to train their models, the lack of annotated review data limits the performance of them.

Recent studies have shown that the use of language models pre-trained with large-scale data is useful for a variety of NLP tasks (Devlin *et al.*, 2018; Peters *et al.*, 2018; Radford *et al.*, 2018). Therefore, to address the existing limitations of current methods, we proposed a joint model BERT-CRF which integrates the popular language model BERT (Devlin *et al.*, 2018) and CRF to extract the aspects and problems from online reviews. Bidirectional Encoder Representations from Transformers (BERT) is one of the critical innovations in the field of natural language processing, and it has achieved great success in many NLP tasks. Its main idea is to learn a lot of prior language knowledge from a large amount of unsupervised corpus through the language model task, and then to fine-tune on specific downstream tasks. Therefore, it can perform well on limited data. The proposed method fine-tunes BERT to learn the features of online reviews and uses CRF to model the dependencies among tags. To evaluate the proposed method, we constructed a dataset from JD.com, which contained 7645 annotated reviews. Experimental results show that the proposed method outperforms other popular methods at present, such as BI-LSTM-CRF.

The main contributions of this paper are as follows: (1) It proposed a new task called *review problem mining* (RPM), which can help producers find problems in products and services efficiently. (2) It proposed the BERT-CRF model for RPM based on the limitations of current methods. Experimental results show the effectiveness of the proposed method. This method can also be applied to some other review information extraction tasks such as aspect extraction (AE).

RELATED WORK

Existing research has produced numerous methods for various tasks of review information extraction, which include both supervised learning and unsupervised learning methods. Unsupervised learning methods usually rely on the dictionary or syntactic rules. Hu and Liu (2004) used association rules to extract frequent nouns and noun phrases as aspects, and the adjectives closest to aspects as sentiment words. Qiu *et al.* (2009) proposed a double propagation method, given a set of sentiment words, which can use the dependencies between the aspect words and the sentiment words to identify new aspects and sentiment words in the reviews. Samha, Li, and Zhang (2014) identified aspects by querying related domain aspect name and synonym information in the WordNet dictionary. García-Pablos, Cuadros, and Rigau (2018) developed an unsupervised opinion mining system based on the topic model, which can automatically identify aspect words and sentiment words in reviews by given aspect and sentiment seeds. Im *et al.* (2019) proposed a method called confirmatory aspect-based opinion mining, which first splits the reviews into a set of clauses, then parses the clauses into a set of aspect-sentiment word pairs. These unsupervised methods most rely on the manual construction of dictionaries and syntactic rules, which is very time-consuming. At the same time, they crucially depend on the grammatical accuracy of the sentences.

Supervised learning methods are successfully applied to review information extraction. Jin, Ho, and Srihari (2009) proposed a lexicalized HMM model, which can identify aspects and sentiment words in reviews and determine the sentiment polarity. Jakob and Gurevych (2010) adopted conditional random fields (CRF), introduced features such as part of speech tags, tokens, short dependency paths, word distances to extract aspects from online reviews. Xiang, He, and Zheng (2018) used the kmeans++ algorithm to acquire multi-feature embedding and word clustering features and input these additional features to the CRF model for training. Laddha and Mukherjee (2018) proposed a hybrid method, LDA-CRF, which uses CRF to extract aspects and sentiment words, and then predicts sentiment scores through a regression method. These methods depend on manual construction and selection of features, which requires a lot of manual effort.

With the development of deep learning technology, many scholars apply deep learning to review information extraction. Poria, Cambria, and Gelbukh (2016) used a 7-layer deep convolutional neural network (CNN) to extract aspects from the reviews. Meanwhile, they developed a set of linguistic patterns and integrated them into the model. Wang *et al.* (2016) proposed a joint model RNCRF to extract aspects and sentiment words from the reviews, it can learn the high-level features and double propagates information between aspect and sentiment words. Jabreel, Hassan, and Moreno (2018) proposed a Bidirectional Gated Recurrent Unit (GRU) network model, the model can identify aspects in the tweet and the polarities of tweet towards each aspect. Wu *et al.* (2018) proposed a hybrid unsupervised method to extract aspects. The method first uses linguistic rules to extract phrase blocks as candidate aspects, and then it uses these texts with extracted chunks as pseudo annotated data to train a GRU network. Al-Smadi *et al.* (2019) used a character-level bidirectional long short-term memory (LSTM) along with CRF to extract aspects from reviews. These deep learning methods need many annotated data to train their models. The lack of annotated review data limits the performance of these methods.

Related researches mainly focus on the extraction of aspects and sentiment words, ignoring the extraction of specific product problems. Unlike them, RPM focuses on the extraction of specific problems in reviews, which helps producers find the pain points of products and improve the quality of products. Meanwhile, existing methods of review information extraction rely

heavily on manual constructed features or large-scale training data. In order to address these limitations, we proposed a new joint model BERT-CRF. Experimental results show the effectiveness of our method.

MODEL DESCRIPTION

The pre-training of large language models requires large-scale data and sufficient computing resources. Therefore, redesigning and pre-training a language model is very expensive and unnecessary for most research works. In this paper, we choose Google's open-source pre-trained language model BERT as the basis of our model. In addition, there are strong dependencies among tags in the sequence labeling task, so we introduce the CRF layer to model the dependencies. In this section, we first introduce the basic architecture of BERT and CRF and then introduce the proposed joint model BERT-CRF in detail.

BERT

BERT (Devlin *et al.*, 2018) is one of the most critical developments in the recent progress of the pre-trained language model (Peters *et al.*, 2018; Radford *et al.*, 2018). The idea behind the progress is that even if the word embeddings are trained from a large-scale unsupervised corpus, it is not enough to learn the contextual representation of the word only through limited supervised data on end tasks. BERT is based on a multi-layer bidirectional transformer encoder (Vaswani *et al.*, 2017), it aims to learn deep contextual representations of words by pre-training on large-scale unsupervised data. It adopts a fine-tuning approach, which can be applied to many downstream NLP tasks.

Input representations

The input of BERT can be a single sentence or a pair of sentences, which are processed in slightly different ways. Since our task is a sequence labeling task, we will introduce the processing mode of a single sentence. In the task of a single sentence, each sentence is converted into a token sequence. The first and last tokens of each token sequence are special tokens ([CLS] and [SEP]). In the pre-training process of BERT, [CLS] is used to encode the information of the whole token sequence, and [SEP] is used to distinguish two sentences. They are not crucial for our task, but in order to be consistent with the pre-trained language model, they need to be preserved. For Chinese, each of the remaining tokens in the token sequence is a Chinese character. As shown in Figure 1, for each token in a token sequence, its input embedding is the sum of corresponding token embedding, segment embedding, and position embedding. Token embedding is related to the specific token. Segment embedding is intended to indicate which sentence the token belongs to, and all segment embeddings are the same for a single sentence. Position embedding is intended to represent the position of the token in the token sequence.

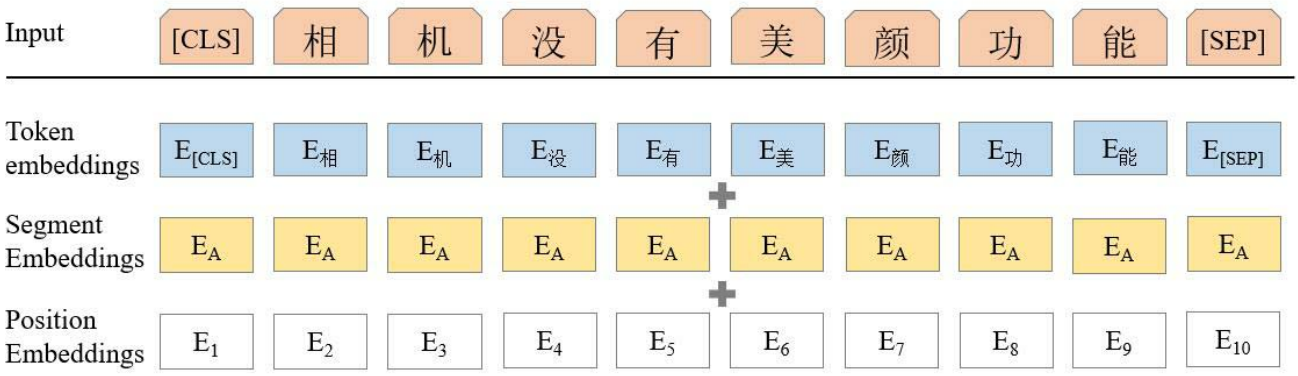

Figure 1: The input representations of BERT

BERT architecture

BERT uses a multi-layer bidirectional transformer encoder instead of BI-LSTM. Compared to LSTM that processes the sequence step by step, the transformer can process the entire sequence in parallel, which speeds up the calculation. Figure 2 shows the architecture of BERT. Given a token sequence $s=(t_1,t_2,\cdots,t_N)$, its input embeddings are formulated as $\vec{x}=(\vec{x}_1,\vec{x}_2,\cdots,\vec{x}_N)$, where $\vec{x}_i$ is the input embedding for t_i, N is the number of tokens in the token sequence. Then $\vec{x}$ goes through multi-layer transformer blocks. Finally, we obtain the hidden states of the last layer (the output of the last transformer block) $\vec{h}=(\vec{h}_1,\vec{h}_2,\cdots,\vec{h}_N)$. These hidden states can be considered as the features of the text, and they can be used for various NLP tasks such as sentence classification and sequence labeling.

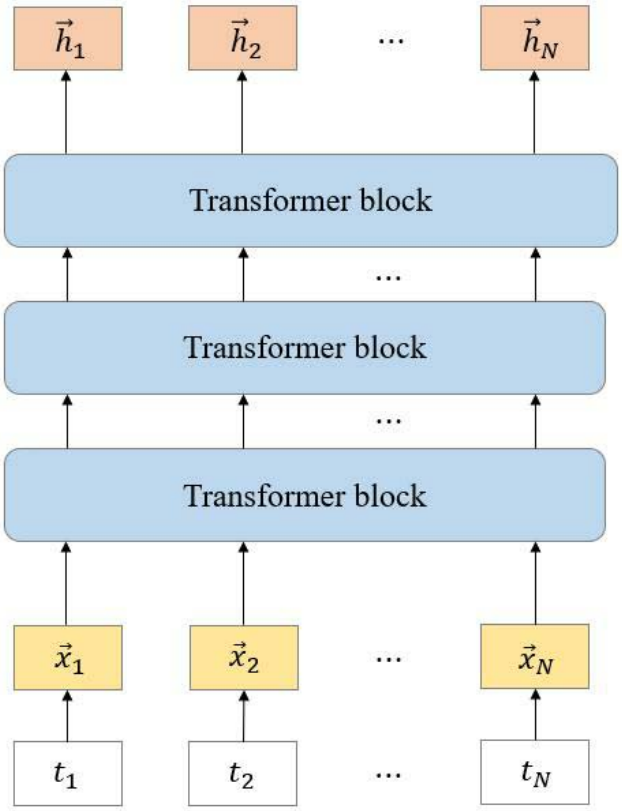

Figure 2: The architecture of BERT

Each transformer block includes a self-attention layer and a fully connected layer. transformer relies entirely on the self-attention mechanism to obtain context information so that it can compute in parallel. Figure 3 shows the architecture of a transformer block. Because introducing the transformer is not the focus of this paper, we will omit an exhaustive description of the transformer and refer readers to (Vaswani *et al.*, 2017).

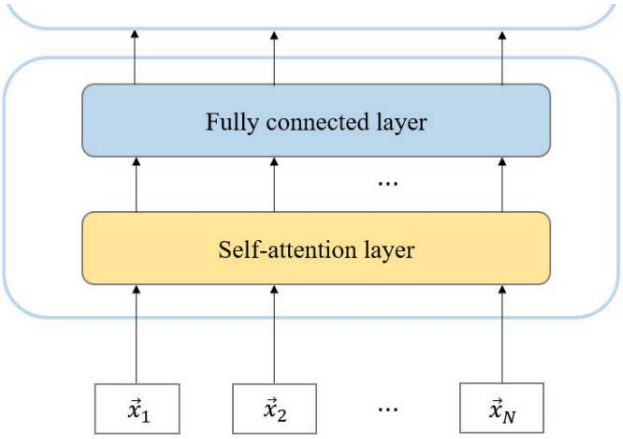

Figure 3: The architecture of a transformer block

BERT was pre-trained on the large-scale document-level corpus using masked language model task (Taylor, 1953) and predicting the next sentence task. For Chinese, the parameters of BERT are set as follows:

BERT$_{BASE}$, Chinese:12-layer, 768-hidden, 12-heads (in transformer), 110M parameters.

The Chinese model was pre-trained on Chinese Wikipedia articles, and we will fine-tune this language model on our dataset.

CRF

Conditional random fields (CRF) (Lafferty, McCallum, & Pereira, 2001) is a conditional probability distribution model, it calculates the joint probability of a tag sequence under a given observation sequence. Linear-chain conditional random fields (linear-chain CRF) is the most commonly used conditional random fields model in the field of natural language processing,

which is widely used in various sequence labeling tasks (CRF mentioned in this paper refers to linear-chain CRF). Its basic architecture is shown in Figure 4. Given an observation sequence $x=(x_1, x_2, \cdots, x_N)$ and a tag sequence $y=(y_1, y_2, \cdots, y_N)$, the probability $P(y|x)$ of the tag sequence corresponding to x being y is as follows:

$$P(y|x)=\frac{1}{Z(x)}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1}, y_i, i)+\sum_{i,l}\mu_l s_l(y_i, x, i)\right) \quad (1)$$

$$Z(x)=\sum_{y}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1}, y_i, i)+\sum_{i,l}\mu_l s_l(y_i, x, i)\right) \quad (2)$$

Where $Z(x)$ is a normalized factor, $t_k(y_{i-1}, y_i, i)$ is the transition function, representing the score of the transition from the tag y_{i-1} at the position $i-1$ to the tag y_i at the position i. $s_l(y_i, x, i)$ is the state function, representing the score of the tag is y_i on the position i of the tag sequence. The values of these two functions are usually 1 or 0. λ_k and μ_l are the weights of the transition function and state function.

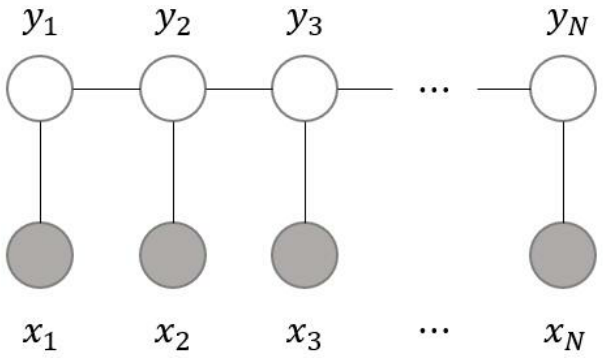

Figure 4: Linear-chain CRF

BERT-CRF Model

Although BERT can well learn the features of the text, it cannot model the dependencies among output tags (e.g. meaningful tags always start with B). Therefore, it is necessary to model the dependencies among tags using CRF. We combine BERT and a CRF layer to form the BERT-CRF model, which is shown in Figure 5. This model can efficiently learn the features of text via BERT and obtain sentence-level tag information via a CRF layer. The CRF layer has a transition matrix as parameters. With such a layer, we can effectively use the surrounding tags to predict the current tag.

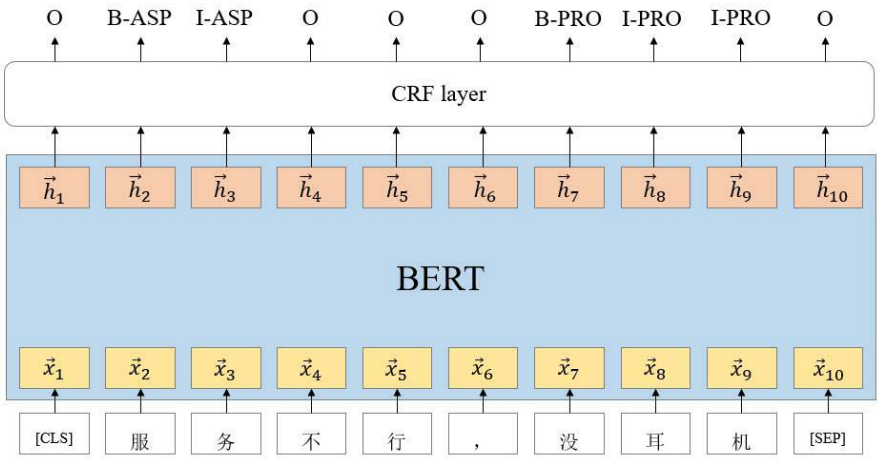

Figure 5: The architecture of BERT-CRF

Given a token sequence $s=(t_1,t_2,\cdots,t_N)$, its input embeddings are $\vec{x}=(\vec{x}_1,\vec{x}_2,\cdots,\vec{x}_N)$. After a multi-layer transformer encoder, we first obtain the hidden states of BERT's last layer as $\vec{h}=(\vec{h}_1,\vec{h}_2,\cdots,\vec{h}_N)$, where $\vec{h}_i\in\mathbb{R}^H$, H is BERT's hidden dimensions, equal to 768. Then $\vec{h}$ is converted to the matrix φ by the parameter matrix W_φ:

$$\varphi=\vec{h}^{\mathrm{T}}\bullet W_\varphi \tag{3}$$

Where $W_\varphi\in\mathbb{R}^{H\times K}$, K is the number of tags. $\varphi\in\mathbb{R}^{N\times K}$ is the score matrix, and φ_{i,y_i} is the score of the tag is y_i on the position i of the tag sequence. To model the dependencies among tags, we introduce a state transition parameter matrix $A\in\mathbb{R}^{K\times K}$, A_{y_{i-1},y_i} is the score of the transition from the tag y_{i-1} to tag y_i. We now denote BERT's parameters as θ, and the parameters of BERT-CRF are $\tilde{\theta}=\theta\cup W_\varphi\cup\mathrm{A}$. For a prediction sequence $\hat{y}=(\hat{y}_1,\hat{y}_2,\cdots,\hat{y}_N)$, we formulate its score to be:

$$sc(s,\hat{y},\tilde{\theta})=\sum_{i=1}^{N}A_{\hat{y}_{i-1},\hat{y}_i}+\sum_{i=1}^{N}\varphi_{i,\hat{y}_i} \tag{4}$$

The softmax function over all possible tag sequences produces the probability $P(\hat{y}\mid s,\tilde{\theta})$ of the sequence $\hat{y}$:

$$P(\hat{y}\mid s,\tilde{\theta})=\frac{\exp(sc(s,\hat{y},\tilde{\theta}))}{\sum_{\tilde{y}\in Y_s}\exp(sc(s,\tilde{y},\tilde{\theta}))} \tag{5}$$

Where Y_s is the set of all possible tag sequences for s. We denote the correct tag sequence as $y=(y_1,y_2,\cdots,y_N)$, during the training, we maximize the log-probability $L(\tilde{\theta})$ of y:

$$L(\tilde{\theta})=\log P(y\mid s,\tilde{\theta}) \tag{6}$$

As can be seen from the above formula, we want the model to generate a meaningful tag sequence. When decoding, dynamic programming (Rabiner, 1989) can be used to obtain the optimal tag sequence. we predict the tag sequence that obtains the maximum score given by:

$$y^*=\underset{\tilde{y}\in Y_s}{\operatorname{argmax}}\,sc(s,\tilde{y},\tilde{\theta}) \tag{7}$$

The training procedure of the BERT-CRF model is shown in Algorithm 1. In each epoch, we divide the data into multiple batches and feed one batch into the model each time. For each batch, we first run the BERT-CRF model forward pass. As a result, we obtain a loss $L(\tilde{\theta})$. We then run the BERT-CRF model backward pass to compute gradients for all model parameters $\tilde{\theta}$. Finally, we update all parameters $\tilde{\theta}$.

Algorithm 1: BERT-CRF training procedure

1: **for** each epoch **do**
2: **for** each batch **do**
3: 1) BERT model forward pass
4: 2) CRF layer forward pass
5: 3) calculate the loss
6: 4) CRF layer backward pass
7: 5) BERT model backward pass
8: 6) update parameters
9: **end for**
10: **end for**

EXPERIMENTS

In this section, we will introduce our experiments. The primary process of the experiments is shown in Figure 6. Besides, in

order to verify the effectiveness of our method, we also compared three commonly used methods of review information extraction at present and reported the comparison results.

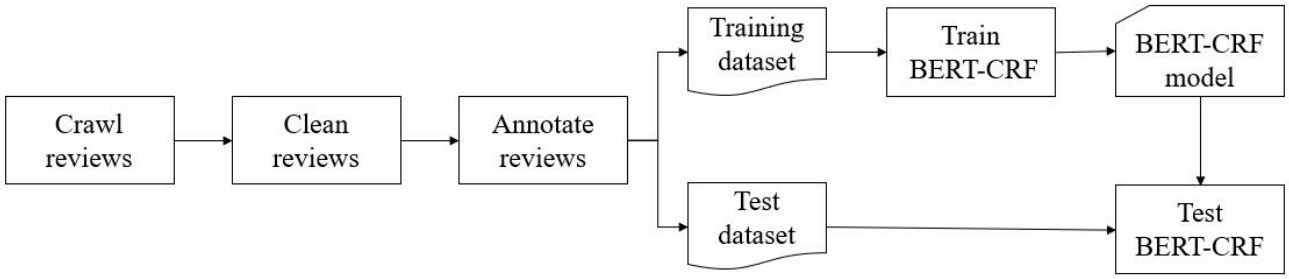

Figure 6: The experimental process

Dataset

Data collection and preprocessing

As there are no existing datasets for RPM, we constructed a dataset. We developed a web crawler using python and used the crawler to crawl 190251 mobile phone reviews from JD.com, then we cleaned the original reviews and got 172200 valid reviews.

The review cleaning process includes: (1) Delete the system default praise; (2) Delete the reviews without Chinese characters; (3) Convert uppercase English letters in the reviews to lowercase English letters; (4) Convert the traditional characters in the reviews to simplified characters; (5) Delete special symbols in the reviews, such as emoticons; (6) Delete duplicate reviews; (7) Delete the reviews with length less than 3.

We did not carry out Chinese word segmentation, because :(1) BERT was pre-trained based on Chinese characters, our input format needs to be consistent with the pre-trained language model. (2) For Chinese, especially social network corpus, the effect of existing word segmentation tools is not very ideal. Inaccurate segmentation results will mislead training and reduce the effect of model, deep learning models should learn such knowledge from data.

Data annotation

Since BERT-CRF is a supervised learning method, data annotation is required. We annotated 7600 reviews, the length distribution of which is shown in Figure 7.

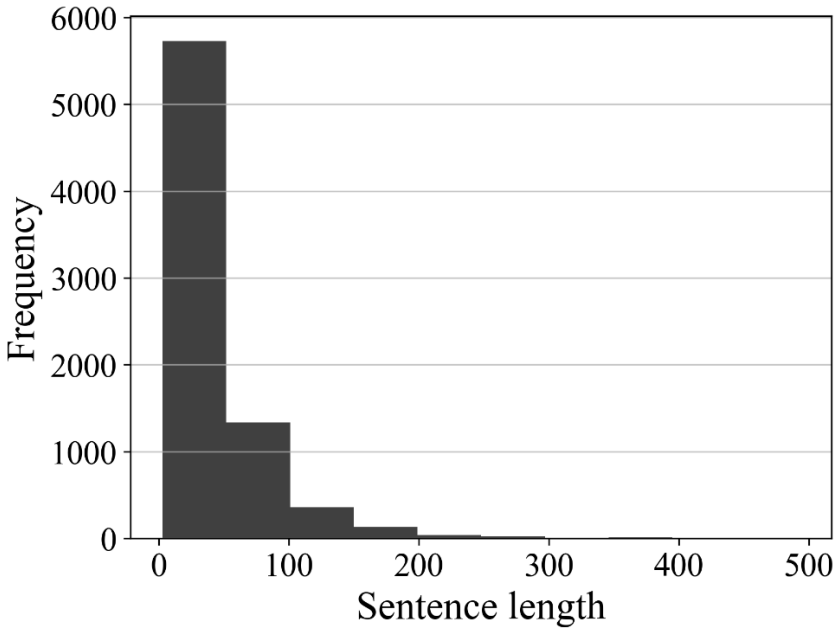

Figure 7: The length distribution of reviews

Neutral and negative reviews are usually shorter than positive reviews. To ensure the balance of the data, there are more neutral reviews and negative reviews than positive reviews in the annotated reviews. The specific distribution is shown in Figure. 8.

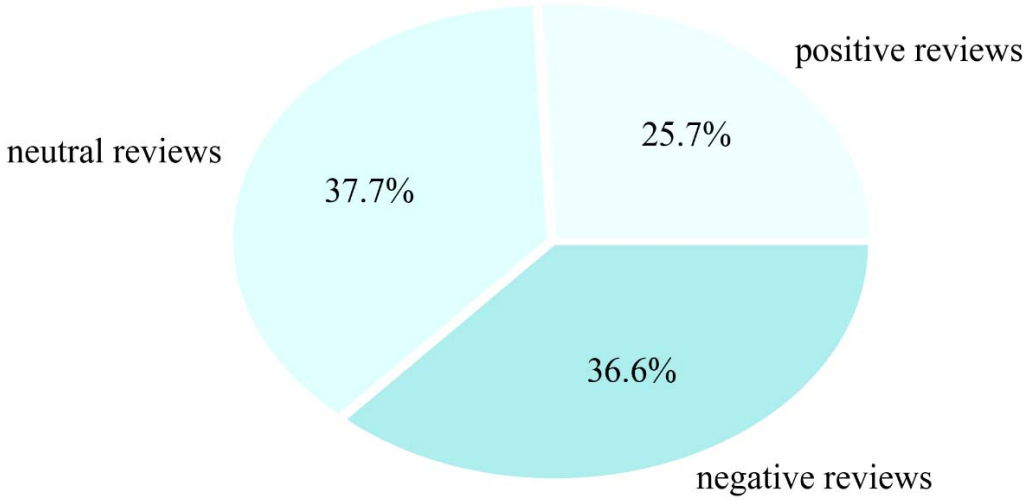

Figure 8: The distribution of positive reviews, neutral reviews, and negative reviews

In the sequence labeling task, each character in the text is assigned a tag. Because the aspects and problems are usually composed of multiple characters, a unified tagging scheme is needed. In this paper, we follow the BIO (Beginning, Inside, and Other) tagging scheme. We use ASP for aspect, PRO for problem and O for others, and then we have five tags {B-ASP, I-ASP, B-PRO, I-PRO, O}, where B-, I- indicate beginning and inside positions of the aspect or problem.

To ensure the quality of the annotation, 7,645 reviews were annotated by the same annotator. Then we divided the annotated reviews data into the training dataset and test dataset according to the ratio of 7:3. Table 1 shows the basic information on the training dataset and test dataset.

Table 1: The basic information on the training dataset and test dataset

	reviews	tokens	B-ASP	I-ASP	B-PRO	I-PRO
training set	5353	208171	8286	11967	2050	2922
test set	2292	90175	3497	5060	930	1290

Table 2 shows an example of the training data.

Table 2: An example of the training data

充	电	口	部	分	有	白	点
B-ASP	I-ASP	I-ASP	O	O	B-PRO	I-PRO	I-PRO
，	屏	幕	偶	尔	会	黑	屏
O	B-ASP	I-ASP	O	O	O	B-PRO	I-PRO

Pre-trained character embeddings

BERT-CRF does not require pre-trained character embeddings as input, but the models for comparison require that for better results. Therefore, we adopt the 172200 reviews after cleaning as character embeddings training corpus and then used the skip-gram model (Mikolov *et al.*, 2013) to generate character embeddings. The dimension of each character embedding is set to 100, and the window size is set to 5.

Experimental Settings

We program in python and develop the model using the deep learning framework TensorFlow. Our experiment is implemented on the hardware with NVIDIA GeForce GTX 1080Ti.

Hyper-parameters

We adopt $\textbf{BERT}_{\textbf{BASE}}$ as the basis for our model, it has 12 layers, 768 hidden dimensions, and 12 heads of self-attention. We apply dropout to $\vec{h}$ with a dropout rate of 0.1. The max sequence length is set to 128. We use Adam optimizer with the learning rate of 5e-5, warmup proportion of 0.1. We train 10 epochs on the training dataset with a batch size of 32.

Compared methods

In order to verify the effectiveness of the proposed method, we compared several commonly used methods of review information extraction at present. We have three baselines: CRF, BI-LSTM-CRF, BI-GRU-CRF.

CRF: Conditional random fields (Lafferty, McCallum, & Pereira, 2001) is a classic method of sequence labeling. We use the open-source tool CRF++ to implement the CRF model and provide the context characteristics of each character to the CRF model for training.

BI-LSTM-CRF: BI-LSTM-CRF (Lample *et al.*, 2016) can use past and future input features via a BiLSTM network and sentence-level tag information via a CRF layer. LSTM (Graves & Schmidhuber, 2005; Hochreiter & Schmidhuber, 1997) is one of the most common variants of recurrent neural networks (RNN), it is better at discovering and exploiting long-range dependencies in data than RNN. We use the pre-trained character embeddings as the input of BI-LSTM-CRF.

BI-GRU-CRF: This method is similar to BI-LSTM-CRF, except that it uses the GRU cell instead of the LSTM cell. GRU (Cho *et al.*, 2014) cell is a simplification of LSTM cell, and it can also model long-range dependencies in data. We also use the pre-trained character embeddings as the input of BI-GRU-CRF.

The hyper-parameter settings for all models are shown in Table 3.

Table 3: The hyper-parameter settings

hyper-parameter	BERT-CRF	BI-LSTM-CRF	BI-GRU-CRF	CRF
character embedding dimensions	768	100	100	-
hidden dimensions	768	100	100	-
max sequence length	128	128	128	128
optimizer	Adam	Adam	Adam	-
learning rate	5e-5	1e-3	1e-3	-
epochs	10	60	60	200
dropout rate	0.1	0.6	0.5	-
cost parameter	-	-	-	3

Evaluation metrics

In this paper, we use precision, recall, and F1 score as the evaluation metrics of the experimental results. In practice, we prefer to identify as many problems as possible in reviews, so recall and F1 score are major evaluation metrics. Meanwhile, in a sequence labeling task, we pay more attention to meaningful tags, so the tag O is not considered in the calculation. The evaluation metrics are calculated by the following formulas:

$$\text{precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{recall} = \frac{TP}{P} \tag{9}$$

$$F1 = \frac{2}{1/\text{precision} + 1/\text{recall}} \tag{10}$$

Where TP indicates that the example is positive and the prediction result is true. FP indicates that the example is positive, but the prediction result is false. P indicates all positive examples. In this paper, the positive examples refer to the aspects or problems in the reviews. F1 score is twice the harmonic mean of precision and recall.

Table 4: The experimental results

	aspects			problems		
	precision	recall	F1	precision	recall	F1

CRF	94.04	93.09	93.56	72.72	51.92	60.59
BI-GRU-CRF	92.51	93.71	93.11	75.06	73.83	74.44
BI-LSTM-CRF	91.64	94.83	93.21	75.28	73.06	74.16
BERT-CRF	**94.37**	**95.85**	**95.11**	**76.66**	**78.25**	**77.45**

Result Analysis

We report all models' performance on test data in Table 4. We observed that the proposed joint model BERT-CRF converges faster, and its precision, recall, and F1 score are better than other models. We believe that the main reason is that BERT was pre-trained on the large-scale corpus, it has much prior knowledge so that it can perform better on limited annotated examples.

We noticed that the extraction effect of all models on the problems is much lower than that on the aspects. We suspect that there are two reasons: (1) The number of problems in the training data is less than the number of aspects, the lack of training examples leads to the reduction of extraction effect. (2) The aspect expressions in the reviews are relatively fixed and straightforward, with apparent features. However, the problem expressions are very complex, often a phrase or clause, which makes them challenging to extract.

We found that the precision of the CRF model is well, but the recall is significantly lower than other models, which indicates that the generalization ability of the CRF model is insufficient. We suspect the reason is that CRF is a shallow machine learning model, which relies heavily on manually constructed features. Therefore, it does not well identify examples that do not appear in the training data. Deep learning models can automatically learn the high-level features of the text, such as semantic features, so they have better generalization ability. Generalization ability is beneficial for problem extraction because problem expressions are usually very diverse and arbitrary, the powerful generalization ability enables the model to identify the examples that do not appear in the training data.

CONCLUSION

We proposed a new task called *review problem mining* (RPM). Compared with aspect-based sentiment analysis, which focuses on the sentiment polarity of aspect, RPM focuses more on the extraction of product problem information in reviews, which is helpful for producers to find the specific pain points of products and improve the quality of products. Meanwhile, to address the limitations of current methods of review information extraction, we proposed a new joint model BERT-CRF, which introduces external knowledge through BERT to reduce the model's dependence on data. To verify the effectiveness of the proposed method, we constructed a dataset from JD.com and carried out experiments. Experimental results show that the proposed method is highly effective. This method can also be applied to other review information extraction tasks such as aspect extraction and sentiment word extraction.

However, this study still has some limitations. For example, the BERT used in our work was pre-trained on Wikipedia articles and has almost no understanding of the review text, which leads to BERT's lack of domain knowledge and limits the effect of the model. Future research can post-train BERT on a large-scale review corpus in the domain of e-commerce, which may make the model perform better. Moreover, some problem expressions in reviews are very long, so how to extract these long problem expressions is also an important research topic in the future.

ACKNOWLEDGMENT

This work was supported by grant No. 71874126, 71373192 from the National Natural Science Foundation of China.

REFERENCES

Al-Smadi, M., Talafha, B., Al-Ayyoub, M., & Jararweh, Y. (2019). Using long short-term memory deep neural networks for aspect-based sentiment analysis of arabic reviews. *International Journal of Machine Learning and Cybernetics, 10*(8), 2163-2175.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

García-Pablos, A., Cuadros, M., & Rigau, G. (2018). W2vlda: Almost unsupervised system for aspect based sentiment analysis. *Expert Systems with Applications, 91*, 127-137.

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

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

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 168-177). ACM

Im, J., Song, T., Lee, Y., & Kim, J. (2019). Confirmatory aspect-based opinion mining processes. *arXiv preprint arXiv:1907.12850*.

Jabreel, M., Hassan, F., & Moreno, A. (2018). Target-dependent sentiment analysis of tweets using bidirectional gated recurrent neural networks. In *Advances in hybridization of intelligent methods* (pp. 39-55): Springer.

Jakob, N., & Gurevych, I. (2010). Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 1035-1045). Association for Computational Linguistics.

Jin, W., Ho, H. H., & Srihari, R. K. (2009). A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of the 26th annual international conference on machine learning* (pp. 465-472). Citeseer.

Laddha, A., & Mukherjee, A. (2018). Aspect opinion expression and rating prediction via lda–crf hybrid. *Natural Language Engineering, 24*(4), 611-639.

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. New York, NY: Cambridge University Press.

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).

Montoyo, A., MartíNez-Barco, P., & Balahur, A. (2012). Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments. *Decision Support Systems, 53*(4), 675-679.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Poria, S., Cambria, E., & Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems, 108*, 42-49.

Qiu, G., Liu, B., Bu, J., & Chen, C. (2009). Expanding domain sentiment lexicon through double propagation. In *Twenty-First International Joint Conference on Artificial Intelligence*.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257-286.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowledge-Based Systems, 89*, 14-46.

Samha, A. K., Li, Y., & Zhang, J. (2014). Aspect-based opinion extraction from customer reviews. *arXiv preprint arXiv:1404.1982*.

Schouten, K., & Frasincar, F. (2015). Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering, 28*(3), 813-830.

Taylor, W. L. (1953). "Cloze procedure": A new tool for measuring readability. *Journalism Bulletin, 30*(4), 415-433.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

Wang, W., Pan, S. J., Dahlmeier, D., & Xiao, X. (2016). Recursive neural conditional random fields for aspect-based sentiment analysis. *arXiv preprint arXiv:1603.06679*.

Wu, C., Wu, F., Wu, S., Yuan, Z., & Huang, Y. (2018). A hybrid unsupervised method for aspect term and opinion target extraction. *Knowledge-Based Systems, 148*, 66-73.

Xiang, Y., He, H., & Zheng, J. (2018). Aspect term extraction based on mfe-crf. *Information, 9*(8), 198.