Product Feature Extraction via Topic Model and Synonym Recognition Approach

Jun Feng, Wen Yang, Cheng Gong, Xiaodong Li and Rongrong Bo

July 18, 2019

Product Feature Extraction via Topic Model and Synonym Recognition Approach

Jun Feng[1], Wen Yang[1], Cheng Gong[1], Xiaodong Li[1], Rongrong Bo[2]

[1] College of Computer and Information, Hohai University, Nanjing, 211100, China
[2] School of Foreign Languages, Nanjing Medical University, Nanjing, 211166, China
rrbo@njmu.edu.cn

Abstract. As e-commerce is becoming more and more popular, sentiment analysis of online reviews has become one of the most active areas in text mining. The main task of sentiment analysis is to analyze the user's attitude towards different product features. Product feature extraction refers to extracting the product features of user evaluation from reviews, which is the first step to achieve further sentiment analysis tasks. The existing product feature extraction methods do not address flexibility and randomness of online reviews. Moreover, these methods have defects, such as relying on labor, low accuracy and recall rate. In this study, we propose a product feature extraction method based on topic model and synonym recognition. Firstly, we set a threshold that TFIDF value of a product feature noun must reach to filter meaningless words in reviews, and select the threshold by grid search. Secondly, considering the co-occurrence rule of different product features in reviews, we propose a novel product similarity calculation, which also per-forms weighted fusion based on information entropy with a variety of general similarity calculation methods. Finally, compared with traditional methods, the experimental results show that the product feature extraction method proposed in this paper can effectively improve F1 and recall score of product feature extraction.

Keywords: Product Feature Extraction, LDA, Synonym Recognition, Shopping Reviews

1 Introduction

With the rapid advance of e-commerce technology, online shopping has gradually become the preferred way of daily consumption. At the same time, a large number of reviews on various products and services has shared over the Internet. It is important to analyze the emotional information expressed from reviews, which can not only help manufacturers find defects in their products but also help consumers while making purchase decisions. For online product reviews, sentiment analysis including sentiment extraction, sentiment classification, sentiment retrieval and summarization. As an essential first step towards achieving sentiment classification and deeper retrieval and induction, product feature extraction refers to extracting the product features of user evaluation from reviews.

With the urgent need of fine-grained sentiment analysis in practical applications, product feature extraction has gradually become a research hotspot. Hu and Liu [1] classify product features into implicit features and explicit features, while the current research mainly focuses on extracting explicit features. Extracting explicit features can be divided into two categories: supervised and unsupervised. If the annotation data is sufficient and accurate, the supervised explicit feature extraction method can achieve better results. Yu et al. [2] used SVM for product feature extraction, clustered similar product features and ranked all product features according to importance. The common unsupervised extraction method is determining product features by mining nouns and noun phrases that occurred frequently. For example, Popescu et al. [3] applied the word frequency to extract product features and tried to filter non-character words. With the development of the topic model [4], more and more scholars use the topic model to extract product features. Mamoru et al. [5] used the improved topic model DTM to analyze the patterns of product features that users are concerned with over time. However, the topic model can only extract coarser-grained global features. The existing product feature extraction methods are not address characteristics of flexibility and randomness of online reviews, and there are still defects such as relying on labor, low accuracy and recall rate.

In this paper, we propose a product feature extraction method based on topic model (LDA, Latent Dirichlet Allocation) and synonym recognition, which makes up for the defect that the topic model can only extract coarse-grained global product features. According to the appearance of product features in reviews, we define the product feature similarity rules and propose a product similarity calculation method. We also perform weighted fusion based on information entropy with a variety of general similarity calculation methods. The experimental results show that our algorithm has better performance of product feature extraction compared with the traditional method.

The rest of the paper is structured as follows. In Section 2, we discuss the related work. Section 3 presents our method for extracting the product features from shopping reviews. Section 4 describes our experimental setup and results. We conclude in Section 5.

2 Related Work

For product feature extraction in mining reviews, the most representative study began with Hu and Liu [5] in 2004 who summarize product feature evaluation of digital products (e.g., "battery life", "screen", etc.). Meanwhile, they also divided product features into two categories, one is to explicit product features, and the other is implicit features. The explicit feature refers to the evaluation object or feature expressed by users in reviews with specific words, such as the review "this phone is small in size, light in weight and easy to carry", in which "size" and "weight" are explicit features directly expressed by the user [6]. At present, the research on product feature extraction mainly focuses on extracting explicit features. The explicit feature extraction methods include two categories [7]: supervised extraction methods and unsupervised extraction methods.

The supervised methods treat product feature extraction as a sequence labeling or classification problem, the morphology, the Part-of-Speech and the syntactic relationship between words in the corpus, the distance between the words in the sentence, the position and other information are regarded as features for sequence learning or classification. Therefore, many supervised algorithms (e.g., conditional random fields, hidden Markov models, support vector machines, etc.) can be applied to product feature extraction. Li et al. [8] proposed Skip-CRF and Tree-CRF based on Conditional Random Field (CRF), used Skip-CRF to solve the long-distance dependence between vocabulary, and used Tree-CRF to learn the grammatical relationships contained in reviews. Yu et al. [2] applied SVM classification to extract product features from the annotated data. In addition, they clustered the similar product features, and ranked the extracted product features according to the score of contribution and frequency of occurrence.

The unsupervised explicit feature extraction methods mainly include three kinds of ways: based on statistical features such as word frequency, product feature and relationship between emotional words, and topic models [9]. Hu and Liu [1] believe that the vocabulary used by users to describe evaluation objects in a specific field is relatively concentrated, and they are generally nouns or noun phrases. Therefore, Hu and Liu extracted nouns and noun phrases that appear frequently in reviews as product features. Moreover, they treated the adjectives, which are closer to product features, as emotional words. Commonly, there is a specific relationship between product features and emotional words. Emotional words are used to modify product features. Therefore, nouns that near emotional words are likely to be product features. Qiu et al. [10] used the dependence relationship between seeds emotional words and features to extract product features, and used propagation algorithm to extract new emotional words and product features through the extracted product features.

With the rapid development of the Topic Model [4], more and more scholars use it to extract product feature. Mei et al. [11] proposed a joint model based on PLSA topic model for product feature extraction [10], while others are almost based on the extension of the LDA topic model. Lin and He [11] proposed an extended model of the LDA, a topic-emotional joint model, which can mine the topic and emotional information in reviews at the same time. Brody and Elhadad [12] used the topic model to identify product feature, and used adjectives that close to the product feature as emotional words. Zhao et al. [13] proposed the LDA extension model named MaxEnt-LDA, which combined the maximum entropy model with the LDA, and used product feature and emotional words for modeling. However, the topic model is mainly concerned with the appearance of high-frequency global product features and emotional words.

Due to the different expression habits of people, different words or phrases in product reviews can describe the same product features [14]. Only by identifying synonyms of these product features, we can better extract product features and summarize viewpoints. Semantic lexicons such as " Synonym Lin ", " HowNet " and " WordNet " [15,16,17] are often used to identify synonyms for product features. For example, Tian Jiu Le et al. [15] used the number of synonymous items in the "Cilin" to calculate the semantic similarity between words. However, it is not ideal to use synonym lexicons to identify synonyms of product features, because some nouns that describe the same product features in product reviews have not been judged as synonyms by lexicons. In addition to based synonym lexicon, some scholars have proposed a series of methods for calculating the similarity of product features, such as TFIDF similarity [18], SimRank similarity [19] etc. The TFIDF similarity considered the context information of the product features in reviews, it taked the word around the product feature as the feature of its vector representation, and used the TFIDF value of the word as the weight of its vector. The TFIDF similarity calculation formula is as follows:

$$TFIDF_{sim}(d_i, d_j) = \frac{\sum_{k=1}^{n} w_{ik}, w_{jk}}{\sqrt{(\sum_{k=1}^{n} {w_{ik}}^2)(\sum_{k=1}^{n} {w_{jk}}^2)}} \tag{1}$$

The SimRank algorithm is a structural similarity algorithm proposed by Jeh et al. The main idea is to construct the connected relationship of product features in online reviews, and used the graph structure to calculate the similarity of product features. With the rise of deep learning, word2vec [20] has been used by more and more scholars to mine text semantics. After word2vec training, considering the cosine distance between word vectors as the similarity of words, which is the most mature and widely used application of word2vec. For example, Luo et al. [21] used word2vec to calculate the similarity between words in the domain text, and realized the clustering of domain words on this.

However, these lexicons or similarity calculation methods described above do not consider the expression characteristics of the shopping reviews. In this paper, we analyzed the expression characteristics of the shopping reviews, proposed a novel method for calculating the similarity of product features, compared and fused with the similarity calculation methods based on TFIDF and word2vec.

3 Model

Due to consider the word frequency in parameter inference process of topic model, the product features extracted by LDA in the model are mostly nouns (global features) with highe occurrences. And, it ignored the specific interpretation and description (local features). To solve the problems mentioned above, we propose a product feature extraction method based on LDA and synonym recognition. Figure 1 shows the framework of our method. Firstly, we use TFIDF to filter nouns that do not represent any product features. Secondly, we use LDA to extract the global features and the parameters of the LDA topic model are selected by grid search. Finally, we get the all features by using global features for synonym discovery.

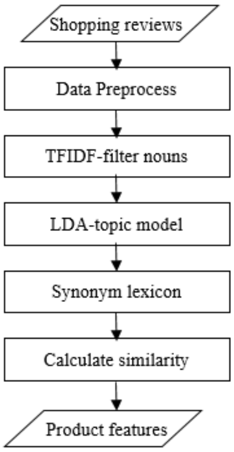

Fig. 1. Framework of method used for extracting product features.

3.1 Extract Global Product Features by LDA

Data Preprocessing

Due to the online customer reviews are unstructured data, they cannot be analyzed directly. It is necessary to preprocess the data, in which word segmentation and Part-of-Speech tagging are particularly important.

Word segmentation is the process of dividing a text into word-based sequences according to certain rules. Two adjacent words in an English sentence are simply separated by a blank space, but words involve more difficult tenses. Therefore, the existing open source software can be used to restore the lexical stems of words, such as the most commonly used natural language processing toolkit NLTK [22]. However, due to the difference in language structure, there is no blank space within Chinese sentences. Individual words have to be identified through word segmentation first. In this case, it is necessary to use existing Chinese word segmentation open source software such as jieba [23] word segmentation software, ik tokenizer [24] and so on. Moreover, to achieve the best word segmentation effect, users often need to add a custom dictionary to identify words in some uncommon field.

According to the observation, product features generally appear in the form of nouns and noun phrases in shopping reviews, while the emotional words used to express opinions are usually adjectives or verbs. In this paper, we use the jieba toolkit for Chinese word segmentation and morphological tagging, and use NLTK to extract stems and fonts of English vocabulary. However, whether in English or Chinese text, there are a large number of words that do not contain any meaning. For example, in Chinese, "是 ", "的", "了", etc., "is", "the", etc. in English. In this case, we need to use the stop words processing operation and filter out these words, which can reduce the negative impact of these meaningless high frequency words on our text analysis. So we use the stop words list as the Chinese stop words, and select the English word contained in the stop words module in NLTK as the English stop word.

Nouns Filtering

The product features are often presented in nouns or noun phrases. Because of the colloquial and non-standard expressions in reviews, in the case of only considering Part-of-Speech, not all the nouns in shopping reviews can be used as candidate product features, and these nouns are usually not included in any existing stop words list. To overcome this problem, it is necessary to filter the nouns and noun phrases obtained after the word segmentation (remove many nouns that do not represent product features, such as "things", "time", etc.). For example, for such a review "This hotel is good, I will recommend it to my mother." Only considering the Part-of-Speech, the term "chance" will be regarded as a product feature. But this term is not the product that users care about in this review. Moreover, due to the irregularity and colloquialism of shopping reviews, such term occupies a high percentage in reviews. When using LDA for product feature extraction, we need to filter these terms. Therefore, it is important to screen out and eliminate such nouns before using LDA to train the corpus.

The object of evaluation should be a term that appears frequently in reviews of a class of goods and rarely appears in others. Therefore, we use TFIDF to select the object of evaluation. The calculation formula of TFIDF is as follows:

$$TFIDF(word) = TF(word) \times IDF(word) \tag{2}$$

Where $TF(word)$ is the frequency of words appearing in the document; $IDF(word)$ is the inverse document frequency, which reflects whether words are only often appearing in a document. It is worth noting that the same word has different TFIDF values in different documents. However, the object that TFIDF considered is the document, not the single review. Fortunately, although there are many reviews for each type of product, we only need to combine the reviews of similar products into one document.

According to Equation (2), we can get the TFIDF values of all words in different documents. At the same time, the nouns with higher TFIDF values are related to features of the evaluated products. For example, we perform a TFIDF calculation on a large number of reviews about the hotels. It can be found that the term such as "restaurant", "bathroom" and "air conditioner" are closely related to the hotel and the TFIDF value is higher. When commenting on a hotel, we inevitably refer to nouns such as “restaurant” and “bathroom”, which are rarely used to review the other products. We set a threshold ε as the TFIDF value that the evaluation object must reach in the experiment.

Model parameter setting

By using the LDA model, the probability distribution for each vocabulary under each topic can be obtained as shown:

$$P(w|z = j) = \{p_{wj1}, p_{wj2}, \dots, p_{wjv}\} \tag{3}$$

Where $P(w|z = j)$ is the probability distribution of each vocabulary under topic j; p_{wji} is the probability of vocabulary w_i under topic j. After getting all the words in each topic of probability distributions, according to equation (4) to get the global feature set S_1:

$$S_1 = \{w_{ij} | \, p_{ij} \gg \sigma\} \tag{4}$$

Parameters of the LDA model need to be input: 1) Hyperparameters α, β of the LDA model; 2) the total number of topics K; 3) TFIDF threshold ε for initial noun filtering; 4) probability threshold σ of global feature candidate words under each topic; Next, the settings of the above four types of parameters are described separately.

First, the α and β of this paper are artificially specified, using empirical values [25], α is K/5 0, and β is 0.01.

Next, using the preplexity to determine the value of the K. In information theory, the preplexity is an indicator, which to measure the quality of the probability distribution or probability model to predicting samples. When using LDA to model a document, the author of the LDA model, D.blei, took the perplexity to determine the number of topics. The definition of preplexity can be expressed by the following formula:

$$perplexity = e^{\frac{-\sum \log(p(w))}{N}} \tag{5}$$

Where $p(w)$ represents the probability of occurrence of each word in the test set, and is specifically calculated into the LDA model as follows:

$$p(w) = \sum_{z,d} p(z|d) \times p(w|z) \tag{6}$$

Where z represents the topic that has been trained; where d represents each document in the test set. The denominator N in the formula (5) represents the number of all words contained in the test set (the total length). For LDA topic model, the lower the perplexity, the better model performance. In this paper, we use the log-perplexity function in the gensim module in Python to calculate the perplexity.

For the TFIDF threshold ε of the initial noun filtering and the probability threshold σ parameter of the global feature candidate under each topic, we draw on the method in [14], determined in the following way.

Since the threshold ε determines the TFIDF values of words in reviews, the larger the ε, the more nouns are filtered out, ε is in the range of (0,1). Similarly, the threshold σ limits the probability of occurrence of words under each topic, the larger the σ, the fewer nouns selected as global features, and the range of σ is (0,1). Therefore, for a given number of topics K, setting ε $= m\beta$, σ $= n\beta$. For product feature extraction, it is mainly tests and compares from three indicators: accuracy, recall and F1 index. The fixed K, α and β, F1can be considered as a function of m and n, the following formula:

$$F1 = f(m, n) \tag{7}$$

Therefore, in the process of model training, we can use the grid search to adjust the parameters. In all candidate parameters, the final result is the best performing parameter after trying each possibility by loop traversal. However, it should be noted that the original data set is divided into a training set and a test set, the test set is used to measure the quality of the model in addition to the adjustment parameters, which will result in a final score that is better than the actual result. Therefore, we need to divide the training set again, into a training set and a verification set. In that, the training set is used for model training, the verification set to adjust parameters, and the test set to measure the quality of the final model.

Finally, set the parameters of the model, input the review text into the LDA topic model, and take the top 5 most probable nouns or noun phrases for each topic. Similarly, take the hotel comment as an example, and the results are shown in the following table.

Table 1. The topic model extracts the global features of hotel reviews

Topic 1	Topic 2	Topic 3
service	location	restaurant
hotel	hotel	facility
attitude	air conditioning	environment
air conditioning	distance	breakfast
customer service	subway	hygienic

From Table 1 above, it is possible to obtain that the nouns mined by LDA, and they are mostly global product features, (e.g., "hotel", "location", "environment", "air conditioning", "restaurant", etc.), which are used frequently in reviews on hotels. However, for some specific local features (e.g., "temperature" or "noise", etc.), it has not been successfully mined. Therefore, the next part of this paper will study local feature extraction based on synonym discovery.

3.2 Extract Local Product Features by Synonym recognition

After training through the LDA model, a global feature set of product reviews can be obtained, but it not means that other nouns are not the product features. Due to the different expression habits, different words or phrases are often used in product reviews to describe the same product feature. For example, the "shape" and "appearance" in the clothing reviews indicate the same product feature. The reviews on a class of products often contain hundreds or thousands of product features. It is time-consuming to manually label synonyms of product features, so we need to find an automatic method to identify synonyms of product features.

Synonym Lexicon and Product Feature Similarity

For the expansion of product features, synonymous supplementation of product features based on synonym lexicons, which is the most popular on research and the most convenient method. Considering the validity and convenience of lexicons, we use a public synonym lexicon to supplement product features that mined by the LDA.

For Chinese, select the Cilin as the synonym lexicon and the WordNet as the English synonym dictionary. Although the two lexicons are not organized in the same way, they can both search for a set of synonyms for a word. The synonym expansion algorithm is as follows:

Algorithm 1. Mine product features by synonym.

```
Input:  LDA extracts global feature set $S_1$, $Synonym$(synonym lexicon),
        D corpus (sets of reviews)
Output: The extended global feature set $S_2$
While $S_1$ is not change
    for each word $w_i$ in $S_1$
        If $w_i$ has near-synset $t$ in $Synonym$
            put $t$ in $S_1$ if $t$ appear in D & $t$ not in $S_1$
    end for
end while, $S_2 = |S_1|$
return $S_2$
```

First, we traverse each word w_i in the global feature S_1. Next, we search for a synonym set t of the word according to the synonym lexicon, add words in the synonym set t that have appeared in corpus to the set S_1, and continue to traverse S_1 until it no longer changes. Finally, we remove the words that are repeated, and get a set of feature words that are augmented by the synonym lexicon.

Although we use the synonym lexicon to supplement product features in part, there are still some limitations in them. Because some words or phrases that describe the same feature appear in shopping reviews, they are not synonyms in the synonym lexicon. For example, "appearance" and "styling" represent the same feature in digital product reviews, but they can not classified as synonyms in a synonym lexicon. The main reason is that the synonym lexicon is universal, not the corpus for product reviews, which contain synonyms usually in accordance with common sense rather than synonyms for product features in shopping reviews. Therefore, we focus on product review corpus, mining the similarities between features, and further expanding the product features.

In shopping reviews, feature synonyms that describe the same product feature tend to have similar contexts. However, the similarity based on the semantic dictionary does not consider the context, that is, it does not take advantage of context information in product review features. By analyzing the product features in corpus, we find a rule in reviews that users always like to start from global evaluation and then fall on the local. For example, reviews such as " The hotel bathroom is not good, the toilet is broken. ", while similar product features typically appear in the same review at the same time. Therefore, for product features in shopping reviews, we propose the similarity principles:

- Rule 1: If product features m_i and m_j appear in the same review, they are considered to have potential similarities, such as “restaurant food".

- Rule 2: If product features m_i and m_j have a side-by-side relationship or affiliation, they are considered to have strong similarities, such as “sheets and pillows”, “toilet in the bathroom”, etc.
- Rule 3: If product features m_i and m_j have a turning point, they are considered not to have similarities, such as "although the location is far, but the price is low", "location" and "price" do not have similarities.

For Rule 2 and Rule 3, it is easy to use the syntactic dependency analysis and adversative relation [26] to judge the two product features (noun/noun phrase) in reviews. In this paper, we use Stanford's syntax-dependent analysis tree Stanford parser [27] to analyze Comment corpus, the product feature similarity algorithm (MRBPF) proposed in this paper is as follows:

Algorithm 2. Mine relationship between product features (MRBPF)

```
Input    $M = \{m_1, m_2, \dots, m_p\}, M' = \{m_1', m_2', \dots, m_t'\}, D = \{d_1, d_2, \dots, d_n\}$
Output:  Matrix $C[t][p]$
$i = 1, C[t][p] = zero\ matrix$
while $i \le t$ do
    $j = 1$
    while $j \le p$ do
        for each $d \in D$ do
            if $(m_i', m_j)$ $in$ $d$ && $m_i' \ne m_j$ then
                $C[i][j] = C[i][j] + 1$
            if $(m_i', m_j)$ $hasrelationA$ $in$ $d$ && then
                $C[i][j] = C[i][j] + 1$
            if $(m_i', m_j)$ $hasrelationB$ $in$ $d$ && then
                $C[i][j] = C[i][j] - 1$
        $j = j + 1$
    $i = i + 1$
for each row  $row \in C[t][p]$ do
    row = normal(row)
return  $C_1$
```

Where $M' = \{m_1', m_2', \dots, m_t'\}$ is the product feature set obtained through Section 3.1, $M = \{m_1, m_2, \dots, m_p\}$ is the noun set of all the nouns in the corpus (excluding the noun below the threshold in Section 3.1), $D = \{d_1, d_2, \dots, d_n\}$ is the shopping reviews set, $hasrelationA$ represents meeting Rule 2, $hasrelationB$ represents meeting Rule 3. $normal$ is a normalization function, it is used to ensure that each dimension in the vector is within the interval $[0, 1]$. Finally, we use the matrix C to store the similarity between global feature nouns and the remaining nouns, where 0 means that the two words are completely dissimilar, and the closer to 1, the more similar.

Similarity Fusion based on Information Entropy

However, considering the diversity of similarity calculation methods, the same two words will get different similarities under different models and calculation methods. Therefore, in order to mine product features as comprehensively and accurately as possible, we consider two other common similarity calculation methods, based on the word2vec method and the TFIDF-based method, and the similarity results are also stored in the matrices C_1, C_2 and C_3 as same structure. Similarly, we normalize each row of the matrix, and mark the similarity of the same words as 0. Therefore, through the matrices C_1, C_2 and C_3, we can get three different similarities of MRBPF, word2vec and TFIDF proposed in this paper, which are respectively recorded as sim_m, sim_w and sim_t.

Define w_1, w_2 and w_3 respectively to represent the weight of three kinds of similarity. However, as there is little knowledge about the difference between different similarity degrees, we adopt a popular method to determine the weight of three different similarity degrees.

In the performance evaluation, each indicator contains different amounts of information, which leads to different resolution of the evaluation system. A basic idea of the entropy weight method is to determine the objective weight according to the variability of the indicator. In general, if the information entropy of an indicator is smaller, it indicates that the index is more variability, the more information is provided, the greater the role that can be played in the comprehensive evaluation, and the greater the weight. In the shopping reviews, the similarity calculation is performed for one product feature and other product features, and the similarity difference between different product features should be large, rather than only a small range does not change substantially. Therefore, we use the entropy weight method to calculate the weights of the above three similarities. The specific steps are as follows:

- The similarity obtained by the three different methods is normalized, because we need to calculate the entropy of each similarity, it is necessary to standardize the values of all similarities. Among them, C, C_1 and C_2 store the similarity of any two product features in the two sets M' and M, and each matrix size is $t * p$. For ease to calculate, the three kinds of similarity matrix transformed into three arrays X_1, X_2, X_3. Where $X_i = \{x_1, x_{2,\dots,}x_{t*p}\}$, $x_1 \sim x_n$ represents all $t * p$ similarities obtained by each similarity algorithm. To calculate information entropy, it needs to be standardized. It is assumed that the standardized value is Y_1, Y_2, Y_3. The formula is as follows:

$$Y_{ij} = \frac{X_{ij} - min(X_i)}{max(X_i) - min(X_i)} \tag{8}$$

- Calculate the information entropy of each index. According to the definition of information entropy in information theory, the formula for calculating the entropy value e_k of the kth similarity is as follows:

$$e_k = \frac{1}{\ln t*p} \sum_{i=1}^{t*p} p_{ik} \ln \frac{1}{p_{ik}} \tag{9}$$

If $p_{ik} = 0$, then $\lim_{p_{ij}\to 0} p_{ij} \log(p_{ij}) = 0$.

- Determine the weight of each indicator. According to the calculation formula of information entropy, the information entropies of the three similarities can be calculated as e_1, $e_2$and e_3, respectively. The weight of each similarity is calculated by information entropy, and the formula is as follows:

$$w_i = \frac{1-e_i}{k-\sum e_i}(i = 1,2,3) \tag{10}$$

Therefore, by calculating the information entropy contained in the similarity of product features under different similarity algorithms, the weights of the three similarities can be obtained. Finally, the similarity between the two product features can be calculated by the following formula. This paper named this similarity calculation method MMRBPF:

$$similarity(m_i, m_j') = w_1 sim\_m(m_i, m_j') + w_2 sim\_w(m_i, m_j') + w_3 sim\_t(m_i, m_j') \tag{11}$$

In addition, it should be noted that the MMRBPF can obtain similarity because it is weighted and summed based on three different similarity methods, and the obtained similarity value may be greater than 1. For the sake of comparison, the same be normalized, and the final result is stored in a matrix in same structure. Finally, we continue to mine product features, use the similarity between global features and residual feature nouns. For each product feature in the set, a noun with a similarity greater than 0.8 is selected as a new product feature noun to supplement.

4 Experiments

4.1 Dataset

Because of the testing of product feature extraction method, we need a test set that labeled the product features included in each review. However, the large number of reviews and the tedious work of labeling, there is no public and convincing test set for everyone to use in the field of Chinese. Therefore, this paper mainly tests the product feature extraction method in the field of English product reviews. The dataset collected by Hu and Liu is widely used by many researchers, and the reviews contained in the corpus are shown in the following table.

Table 2. Hu and Liu's collection of product reviews

Product name	Number of sentences	Number of product features
Digital camera(Canon)	597	237
Digital camera(Nikon)	346	174
Cell phone(Nokia)	546	302
MP3 player(Creative)	1716	674

DVD player(Apex)	740	296

4.2 Evaluation Metric

In this paper, we use the accuracy rate P, recall rate R and F_1 values that are widely used at present. The evaluation formula is:

$$P = \frac{FP}{F} * 100\% \tag{12}$$

$$R = \frac{FP}{FE} * 100\% \tag{13}$$

$$F_1 - measure = \frac{2 \times P \times R}{P+R} \tag{14}$$

In the above formula, FR represents the number of correct product features extracted, F represents the total number of product features extracted, and FE represents the number of real product features contained in the actual corpus.

4.3 Experimental Setup and Results Analysis

The experiments in this paper are divided into two parts: In the first part, the test uses TFIDF to filter the effect of nouns on product features by threshold ε. In the second part, to verify the effectiveness of the proposed method in product feature extraction, we compare the product feature similarity algorithm proposed in this paper with the traditional similarity algorithm.

Test 1: Testing the validity of the TFIDF threshold ε

In order to test the influence of different threshold ε on product feature extraction, it is found that the ε selected by the final model is mostly around 0.3. Therefore, the value of ε starts from 0. It is gradually increased from 0.05 to 0.3, and the change of the product feature extraction P, R, F_1 is separately calculated. Attention, the other parameters of the model are fixed at this time. α and β are determined by manual experience, and K is determined by the degree of perplexity. As for the probability threshold σ of global feature candidate for each topic, determined by performing the grid search optimal parameter on the training set. Next, for the subsequent synonym expansion, we adopted the MMRBPF similarity algorithm proposed in this paper.

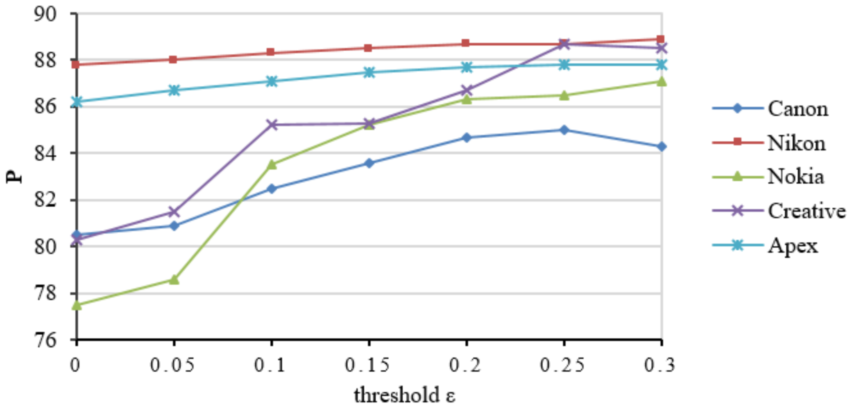

Fig. 2. Accuracy rate P with threshold ε change line graph

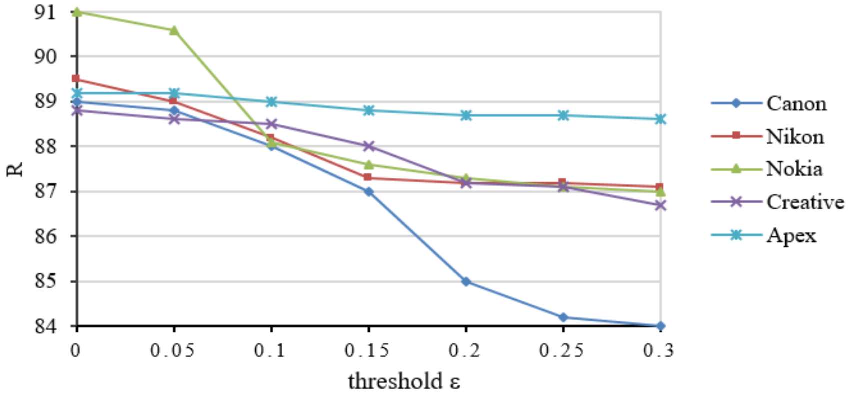

Fig. 3. The recall rate R varies with the threshold ε

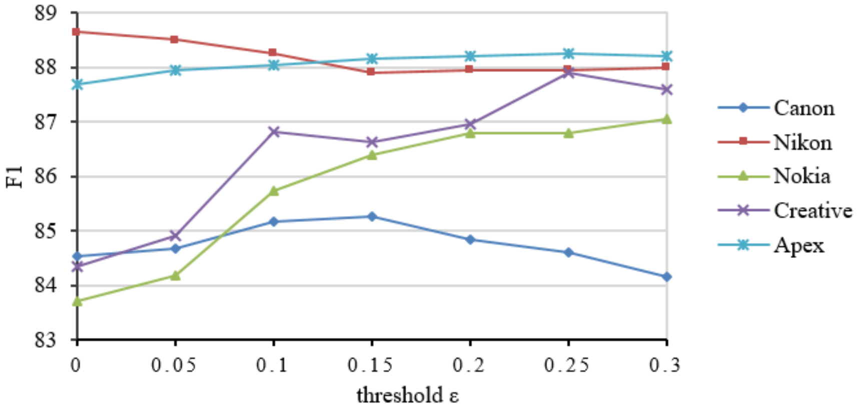

Fig. 4. F1 line graph with threshold value ε

The experimental results are shown in Figures 2, 3, and 4. With the gradual increase of ε, the accuracy P of the five types of products has been improved, but the recall rate of the products has been reduced to extent. For F1, four of the five categories of products have been improved (relative to the threshold ε of 0), and only F1 of Nikon's products has declined slightly.

Thus, using the TFIDF to set threshold ε to filter the nouns in corpus can significantly improve the accuracy of product feature extraction. Next, as the threshold ε increases, some of the correct product features are filtered due to the low frequency of occurrence. Therefore, the recall of product feature extraction will continue to decrease, and the improvement of accuracy is partly due to the decrease in the number of candidate product features extracted, that is, F. In addition, with the increase of the threshold ε, Creative and Nokia's F1 is greatly improved compared to threshold ε of 0, Cannon's F1 is slightly increased, but Nikon's F1 is slightly decreased. This is because the total number of sentences and vocabulary contained in Creative and Nokia is relatively rich, but Nikon contains relatively few sentences and vocabulary. As the threshold ε increases, the impact of the recall reduce is greater than the accuracy increase in Nikon reviews. In summary, we adopt TFIDF to set the threshold ε to filter the product feature extraction method is effective, which can improve the accuracy and favor the F1 promotion of certain commodity categories, and more suitable for the number of comments and vocabulary richer corpus. From above, the optimal threshold ε for different categories of goods is different, and there is no rule. Therefore, it is necessary to select the parameter of the grid search on the training set.

Test 2: Test the effectiveness of the MRBPF and MMRBPF methods for product feature extraction.

We compare MRBPF and MMRBPF with two general word similarity algorithms are used: based on word2vec and based on TFIDF. For these four different similarity calculation methods, we set the remaining parameters of the model to be consistent.

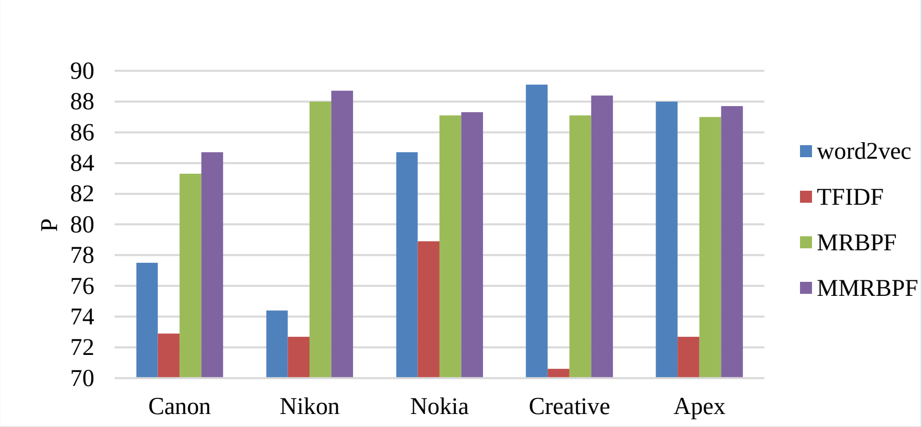

Fig. 5. Product feature extraction accuracy rate P under different similarity algorithms

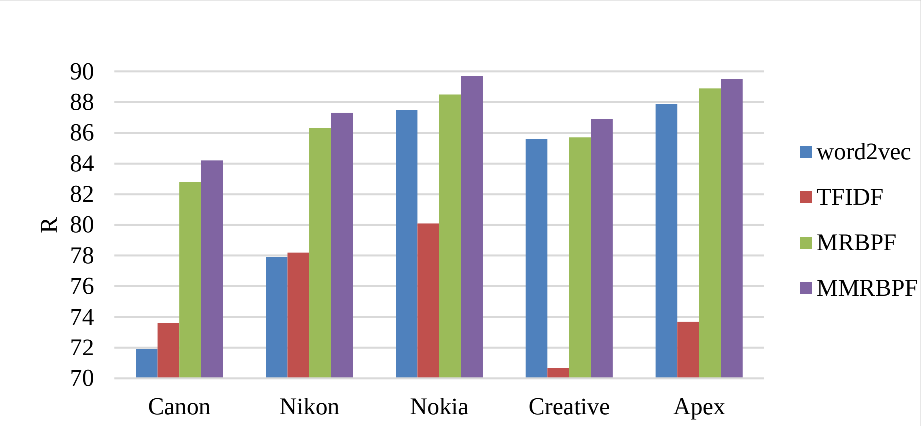

Fig. 6. Product feature extraction recall rate R under different similarity algorithms

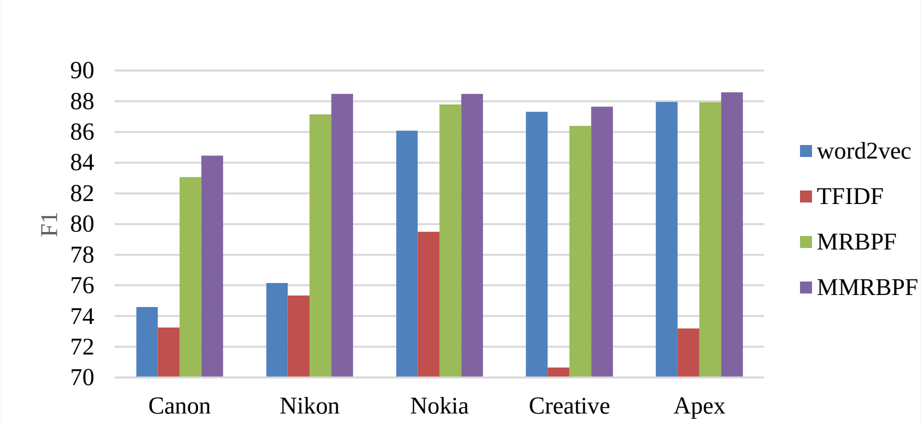

Fig. 7. Product Feature Extraction F1 under Different Similarity Algorithms

The experimental results are shown in Figures 5,6, and 7. For the accuracy rate P, the semantic similarity based on TFIDF is the worst in the five categories of products. In addition, the word2vec method performs better in Creative and Apex because of the richer reviews, but poor performance in Nikon and Cannon due to fewer reviews. While the similarity algorithm MRBPF and MMRBPF algorithm proposed in this paper are relatively stable, and show good performance in product feature extraction accuracy of five types of products. For the recall rate R, the performance of TFIDF is still the worst, word2vec is not good in the Nikon and Cannon products with less reviews. The MRBPF and MMRBPF methods proposed in this paper are relatively stable, and MMRBPF achieved the best results in the recall rate R of the five types of product feature extraction. Similarly, for product feature extraction F1, the performance MRBPF and MMRBPF method are relatively stable, MMRBPF also have achieved the best results.

Thus, using the TFIDF similarity to extract product features is not ideal because only the contextual TFIDF values are considered. Using the word2vec to calculate the similarity extraction feature. Although it works well in some corpus, it dependents on the number of corpora. If the corpus resources are not rich enough, the performance of word2vec will drop dramatically. While the MRBPF method has achieved good and stable effects in different types of products based on the specific characteristics of product reviews. Next, because the MMRBPF method proposed in this paper combines the above three similarity algorithms, although the accuracy rate is not always optimal, the recall rate is improved, and the final F1 results are also optimal. The average of the weights (30 times) obtained by the three kinds of similarities through the information entropy is shown:

Table 3. Average weight of three different similarities

Product name	word2vec	TFIFF	MRBPF
Digital camera(Canon)	0.301	0.015	0.684
Digital camera(Nikon)	0.183	0.065	0.752
Cell phone(Nokia)	0.341	0.135	0.524
MP3 player(Creative)	0.379	0.154	0.467
DVD player(Apex)	0.314	0.113	0.573

From above, the weights of the similarities obtained by the five types of commodities based on TFIDF are very small. Word2vec has a smaller weight in Cannon and Nikon with fewer reviews, and another way to explain the dependence of the word2vec method on the number of corpus. Due to the MMRBPF based on three similarity fusions has improved in the recall rate and F1 compared to the three independent similarity algorithms, it is reasonable and effective to use the information entropy to weight the different similarities.

5 Conclusions

Product feature extraction is an essential part of sentiment analysis of online product reviews. In this paper, we propose a product feature extraction method based on LDA and synonym recognition. Firstly, we consider the TFIDF value of a product feature noun must reach as threshold to filter out meaningless words, and use grid search to determine the threshold. Secondly, considering the co-occurrence rule of different product features in the reviews, we propose a novel product feature similarity calculation algorithm MRBPF. Moreover, we propose another similarity calculation algorithm MMRBPF by weighting fusion of MRBPF with two popular similarity calculation methods TFIDF and word2vec. Finally, we conduct experiments on English product shopping reviews for product feature extraction.

From the experimental results, we find that the F1 and accuracy of product extraction will improve by setting TFIDF threshold because that can filter nouns that do not represent any product features. Compared with TFIDF and word2vec, the MRBPF improves the accuracy though the corpus is not sufficient. In addition, we find that the

MMRBPF has the best recall and F1, thus proving the effectiveness of the similarity fusion based on information entropy.

References

1. Hu Minqing, Liu Bing.: Mining and summarizing customer reviews. In: KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.168-177. Association for Computing Machinery, Seattle, WA, United states (2004).
2. Yu Jianxing, Zha Zhengjun, Wang Meng.: Aspect ranking: identifying important product aspects from online consumer reviews. In: ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol.1, pp. 1496-1505. Association for Computational Linguistics (ACL), Portland, OR, United states (2011).
3. Popescu Ana-Maria, Etzioni Orena.: Extracting product features and opinions from reviews. Natural language processing and text mining, pp.9-28, Springer London (2007).
4. Blei David, Carin Lawrence, Dunson David.: Probabilistic topic models. Journal 27(6), 55-65 (2010).
5. Emoto, Mamoru.: Method for Extraction of Purchase Behavior and Product Character Using Dynamic Topic Model. In: Proceedings - 16th IEEE International Conference on Data Mining Workshops, ICDMW 2016, pp.778-782. IEEE Computer Society, Barcelona, Spain (2017).
6. Rana Toqir A, Cheah Yu-N.: Aspect extraction in sentiment analysis: comparative analysis and survey. Journal 46(4), 459-483 (2016).
7. Qian Liu.: Research on approaches to opinion target extraction in opinion minging. Southeast University (2016).
8. Li Fangtao, Han Chao, Huang Minlie.: Structure-aware review mining and summarization. In: Proceedings of the 23rd international conference on computational linguistics, vol.2, pp. 653-661. Tsinghua University Press, Beijing, China (2010).
9. Schouten, Kim, and F. Frasincar.: Survey on Aspect-Level Sentiment Analysis. IEEE Transactions on Knowledge and Data Engineering Journal 28(3), 813–830 (2016).
10. Qiu GA, Liu B, Bu JJ.: Opinion word expansion and target extraction through double propagation. Journal 37(1), 9-27 (2011).
11. Mei Qiaozhu, Ling Xu, Wondra M.: Topic sentiment mixture:modeling facets and opinions in weblogs. In: 16th International World Wide Web Conference, WWW2007, pp. 171–180. Association for Computing Machinery, Banff, AB, Canada (2007).
12. Brody Samuel, Elhadad Noemie.: An Unsupervised Aspect-Sentiment Model for Online Reviews. In: NAACL HLT 2010 - Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Main Conference, pp. 804–812. Association for Computational Linguistics (ACL), Los Angeles, CA, United states (2010).
13. Zhao W, Jiang J, Yan H.: Jointly Modeling Aspects and Opinions with a MaxEnt-LDA Hybrid. In: EMNLP 2010 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, pp. 804–812. Association for Computational Linguistics (ACL), Cambridge, MA, United states (2010).
14. Baizhang Ma, Zhijun Yan.: Product features extraction of online reviews based on LDA model. Journal 20(1), 98-103 (2014).

15. Tian Jiule, Zhao Wei.: Words Similarity Alogrithm Based on Tongyici Cilin in Semantic Web Adaptive Learning System. Journal 28(06), 602-608 (2010).
16. Xu Linhong, Lin Hongfei Pan.: Constructing the Affective Lexicon Ontology. Journal 2 (2008), 602-608 (2010).
17. Lopez-Arevalo I, Sosa-Sosa V J, Rojas-Lopez F.: Improving selection of synsets from WordNet for domain-specific word sense disambiguation. Computer Speech & Language41(C),128-145 (2017).
18. Xi Yahui.: Recognizing the Feature Synonyms in Product Review. Journal 30(4), 150-158 (2016).
19. Jeh Glen, Widom Jennifer.: SimRank: a measure of structural-context similarity. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 538–543. Association for Computing Machinery, Edmonton, Alta, Canada (2002).
20. Mikolov Tomas, Chen Kai, Corrado Greg.: Efficient estimation of word representations in vector space. Computer Science, 1301-3781 (2013).
21. Luo Jie, Wang Qinglin, Li Yuan.: Word clustering based on word2vec and semantic similarity. In: Proceedings of the 33rd Chinese Control Conference, CCC 2014, pp. 804–812. IEEE Computer Society, Nanjing, China (2014).
22. Natural Language Toolkit. NLTK3.4 documentation, http://www.nltk.org/, 2018/11/7
23. Python Software Foundation. jieba0.39, https://pypi.org/project/jieba/, 2017/08/28
24. Github. Kunshan Wang.ik-analyzer, https://github.com/wks/ik-analyze, 2011/04/14
25. Turney P D, Littman M L.: Measuring praise and criticism: Inference of semantic orientation from association. Journal 21(4), 315-346 (2003).
26. Wu Shuang.: Sentiment Polarity Unit Extraction of Web-based Financial Information Based on Dependency Parsing. Jiangxi University of finance & economics (2015).
27. De Marneffe M.C, Manning C.D.: The Stanford typed dependencies representation. In: Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation, pp. 1-8 (2008).