2012 IEEE 12th International Conference on Data Mining Workshops

Fine-grained Product Features Extraction and Categorization in Reviews Opinion Mining

Sheng Huang, Xinlan Liu, Xueping Peng, Zhendong Niu
The School of Computer Science and Technology
Beijing Institute of Technology
Beijing 100081, China
huangsheng2009@gmail.com, {0330334, pengxp, zniu}@bit.edu.cn

***Abstract*—With the growth of user-generated contents on the Web, product reviews opinion mining increasingly becomes a research practice of great value to e-commerce, search and recommendation. Unfortunately, the number of reviews is rising up to hundreds or even thousands, especially for some popular items, which makes it a laborious work for the potential buyers and the manufacturers to read through them to make a wise decision. Besides, the free format and the uncertainty of reviews expressions, make fine-grained product features extraction and categorization a more difficult task than traditional information extraction techniques. In this work, we propose to treat product feature extraction as a sequence labeling task and employ a discriminative learning model using Conditional Random Fields (CRFs) to tackle it. We innovatively incorporate the part-of-speech features and the sentence structure features into the CRFs learning process. For product feature categorization, we introduce the semantic knowledge-based and distributional context-based similarity measures to calculate the similarities between product feature expressions, then an effective graph pruning based categorizing algorithm is proposed to classify the collection of feature expressions into different semantic groups. The empirical studies have proved the effectiveness and efficiency of our approaches compared with other counterpart methods.**

Keywords-product features; extraction and categorization; conditional random fields; similarity calculation

I. INTRODUCTION

The user-generated reviews have increasingly played an important role in market intelligence, e-commerce and recommendation, since they contain valuable opinions originated from other users' experiences. For instance, in e-commerce sites, before a potential buyer decides to purchase an item, he tends to consult other buyers' suggestions and usage experiences. As the user feedback and market investigation, online sites also encourage consumers to comment on various aspects of the products. Nowadays, most online e-ecommerce websites, such as Amazon, Ebay, and Taobao, have encouraged the customers to post their reviews freely. As more and more people become comfortable with e-commerce, the reviews' number has grown rapidly and in fact reached to hundreds or even thousands for some popular items, which makes it a rather hard work for the potential buyers to read through them to make a wise decision. The large number of reviews also makes it hard for product manufacturers to keep track of customer opinions and sentiments on their products and services.

Fine-grained reviews opinion mining has always been a challenging problem. During the past several years, many studies had tried to normally define and solve this problem [1-16]. Since consumers are encouraged to express their comments freely and people usually use different words to describe the same features in the reviews, for example, "picture quality" and "image clarity" are both feature expressions referring to the picture/photo aspect in cameras reviews, the customer reviews show high levels of arbitrariness and uncertainty. This also makes it a more difficult task than traditional information extraction and analysis.

The techniques aiming at solving this task can be mainly classified into two categories: statistic information-based approaches and machine learning-based methods. The existing statistical techniques [2-7] usually base on manually designed extraction rules and use high-frequency noun phrases as the candidate product features. But they can produce too many non-product features and miss many low-frequency terms. The machine learning techniques [8-15] try to treat it as a probabilistic learning process, while the features selection is always a tough problem.

In this paper, we present a CRFs-based probabilistic learning model to extract product features (also known as feature expressions below) from the review sentences sequence. The part-of-speech features and the sentence structure features are clearly incorporated into the CRFs learning process. Then we describe two categories of similarity measures to categorize product features into different feature groups. The combination of both categories of similarity measures is employed to obtain the best performance. We propose a syntactic dependency approach for distributional context extraction and a bipartite graph-based method for distributional similarity calculation. The experimental results show that our approaches achieve excellent performance compared with other existing methods.

The rest of this paper is structured as follows: Related works are described in section 2. Section 3 and 4 describe our product features extraction and categorization strategies respectively. Section 5 gives out the experiments and results analysis. At last, we conclude our work in section 6.

978-0-7695-4925-5/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDMW.2012.53

II. Related Work

The related work mainly focuses on product features extraction from product reviews and product features categorization since we focus on these two tasks in this work. They are necessary to classify opinion polarity and produce effective feature-based opinion summarization.

Product features extraction and categorization is firstly studied as showed in [2]. Hu and Liu [2] presented a two-step statistical strategy that selects highly frequent feature words by using association rules. Popescu and Etzioni [3] improved Hu and Liu's work by removing frequent noun phrases that may not be real features. Their method achieved a better precision, but a small drop in recall. Wu et al. [4] selected noun phrases and verb phrases as candidate product features and introduced a language model to filter candidate product features with the assumption that product features are mentioned more often in product reviews than in general texts. Qiu et al. [5] also only extracted nouns or noun phrases as candidate opinion targets. However, the limitation of these methods is that they usually achieve normal recall performance but very bad precision, and fail to identify infrequent entities effectively.

Topic modeling is also frequently used in aspect and sentiment analysis of online reviews recently. Brody and Elhadad [8] presented an unsupervised manner for extracting aspects and determining sentiments in review text. They took a two-step approach by first detecting product aspects by sentence based local LDA and then identifying aspect-specific opinion words using polarity propagation. But they only considered sentences that contain only single aspect. Zhao et al. [9] used a MaxEnt-LDA hybrid by incorporating a supervised discriminative maximum entropy model into an unsupervised generative topic model to jointly discover both aspects and aspect-specific opinion words simultaneously. Zhai et al. [10] extended a constrained-LDA model with the ability to process large scale constraints, and proposed two methods to extract constraints automatically. However, these methods have represented each feature aspect by single words, while ignoring the phrasal structures.

Guo et al. [11] proposed an unsupervised product-feature categorization method with multilevel latent semantic association (called mLaSA). The first LaSA model generated the latent semantic structure for each product-feature, and the second LaSA model categorized product features according to their latent semantics structures and contexts snippets in the reviews. Zhai et al. [12] formulated the unsupervised feature clustering problem into a semi-supervised learning task, and an EM algorithm based on naïve Bayesian classification was adopted to solve it. Jin et al. [13] presented a novel supervised machine learning system built under the framework of lexicalized HMMs. Although it is proved more effective than previous works, it is still not able to model arbitrary, dependent features of the input data sequence.

Li et al. have successfully proposed and applied matrix factorization algorithms into some opinion mining tasks. Li et al. [23] proposed to build a constrained non-negative tri-factorization model of the term-document matrix, with the knowledge learned from domain-independent sentiment-laden terms and domain-dependent unlabeled data, in conjunction with a few labeled documents. Li et al. [24] applied non-negative matrix tri-factorizations into cross-domain sentiment analysis task. Document labels were incorporated into matrix factorization framework via a least squares penalty incurred by a certain linear model that enables direct and explicit knowledge transfer across different domains. Cambria et al. [22] devoted to develop a Chinese common and common sense knowledge base to assist Chinese sentiment analysis, based on content translation of some existing English common knowledge.

The latest product feature extraction work is done in [14]. They have also attempted to adopt linear-chain CRFs for opinion mining. However, compared with our method, the experimental results show that they still have room for improvement.

III. Proposed Extraction Strategy

A. CRFs-Learning Model

Conditional random fields (CRFs) are a discriminative undirected probabilistic graphical models framework for labeling and segmenting sequential data [14, 17, 18]. A CRF defines a single log-linear distribution over label sequences given a particular observation sequence. Lafferty and MaCallum [18] defined a CRF on observations X and random variables Y as follows: Let $G = (V, E)$ be a graph for which V indicates the nodes and E indicates the edges. Let $Y = (Yv)_{v\ \ V}$, so that Y is indexed by the vertices of G. Then (X, Y) is a conditional random field in case, when conditioned on X, the random variables Yv obey the Markov property with respect to the graph:

$$p(Y_v \mid X, Y_w, w \neq v) = p(Y_v \mid X, Y_w, w \sim v) \qquad (1)$$

Where $w \sim v$ means w and v are neighbors in G. Formally, the joint distribution over the label sequence Y given X has the form:

$$p_\theta(y \mid x) \propto \exp\left(\sum_{e \in E,k} \lambda_k f_k(e, y|_e, x) + \sum_{v \in V,k} \mu_k g_k(v, y|_v, x) \right) \qquad (2)$$

B. Product Features Types Definitions

In this work, we define that product features are products, product components, function of products, properties of products, properties of product components and related objects. For example, in digital camera domain, "picture quality", "lens cap" and "shutter delay" are all product features.

To capture the lexical and the semantic information contained in the training data, we define three types of features for the CRFs learning process: Word Features, Part-of-Speech (POS) Features and Sentence Structure Features.

(1)Word Features: To avoid the influence of different variants of a word, we use the stem of the word instead of the different variants. And we use the stems in a [-1, +1] window as word features.

(2)POS Features: POS tag defines the part-of-speech (pos) information of each word in the sentences. We use Stanford part-of-speech tagger to generate pos tags for the words in each sentence. To shrink the pos tags space and reduce computation complexity, we represent all adjective-related pos tags (like JJ, JJS and JJR) as JJ, all noun-related pos tags (like NN, NNS, NP and NPS) as NN, all adverb-related pos tags (like RB, RBR, RBS) as RB, all verb-related pos tags (like VB, VBD, VBG, VBN, VBP and VBZ) as VB. Specifically, we use pos tags in [-1, +1] window as POS features.

(3)Sentence Structure Features: Since the word features and the POS features involve only single words and the partition windows capture only neighboring words, we need to learn the syntactic relations between words spanning long distance. Sentence structure features encode the syntactic structure information among the words in the sentences. Syntactic dependency parser is used to extract the syntactic dependency relations for each sentence. We have defined seven structure relation tagging rules in Table I.

For example, given the sentence "The video is good, but sounds are not very loud.", its dependency relations are as follows: det(video-2, The-1), nsubj(good-4, video-2), cop(good-4, is-3), root(ROOT-0, good-4), nsubj(loud-11, sounds-7), cop(loud-11, are-8), neg(loud-11, not-9), advmod(loud-11, very-10), and conj_but(good-4, loud-11). Since "good" and "video" have the dependency relations "nsubj(good-4, video-2)" and "cop(good-4, is-3)", they are tagged "CADJ" and "CNN" respectively. Since "sounds" and "loud" have the dependency relations "nsubj(loud-11, sounds-7)" and "cop(loud-11, are-8)", but there also exists "neg(loud-11, not-9)", so "loud" should be tagged "CADJNEG". And it is modified by "very" since there exists "advmod(loud-11, very-10)", so the word "very" should be tagged "CADV".

C. Tagging Schema

We have defined four kinds of labels for the semantic roles of the words: F, O, MOD and B. The details of the labels are showed in Table II. F denotes the words that are related to product features; O is indicative of the words that are related to consumer opinions-contained expressions; MOD describes the degree words that are used to decorate opinion words; And B is used to declare the background words in the sentence.

IV. Proposed Categorization Strategy

A. Similarity Calculation

The critical challenge of product features categorization is how to measure the similarity and relatedness among different feature expressions. The measures used to solve this challenge can be roughly classified into two categories: those relying on semantic knowledge resources (e.g. WordNet, thesauri or encyclopedias) and those inducing distributional properties from corpora. The semantic knowledge measures capture the domain-independent lexical and semantic similarities. E.g., "picture" and "image" bear high similarity in WordNet with pos tag "NN". The distributional context measures capture the distributional context-based similarities in large corpora.

In this work, we introduce a measure for each category: WordNet-based similarity calculation for the first category and bipartite graph-based similarity calculation for the second category. A syntactic dependency approach is proposed to extract the distributional context for each feature expression, and a simple but effective graph pruning-based strategy is employed to cluster these feature expressions into semantic groups. The effectiveness of each measure is described in the experiment section.

WordNet-based similarity calculation: WordNet is a widely-used lexical database of English in the NLP area. Since it has been widely-used to measure the similarity of two words [12, 19-21], we also utilize WordNet in our feature expressions categorization task. Since WordNet is constructed according to the part-of-speech tags, we calculate pair-wise similarities between the component words with the same POS tag for each pair of feature expressions. The similarity between two feature expressions is calculated as the average value of all their component words similarities. The common used Jiang&Conrath algorithm is used to calculate the pair-wise similarities of the component words. Eq. (3) describes this algorithm.

$$Jcn = \frac{1}{IC(concept_1) + IC(concept_2) - 2 * IC(Ics)} \tag{3}$$

Where Ics represents the least common subsumer of $Concept_1$ and $Concept_2$, and IC denotes the information content.

Syntactic dependency approach: Compared with context windows approach that only captures distributional context in neighboring windows, syntactic dependency approach can capture long distance syntactic context. The syntactic parser is used to parse each sentence and construct it into a dependency tree. For each product feature expression f_e, we collect sets of governing words (e.g., the parent, grand-parent etc.) as well as collections of descendants (e.g., immediate children, grandchildren etc.). These words are then encoded as a contextual vector for f_e. The detailed syntactic dependency approach is shown in Algorithm 1.

Algorithm 1: Syntactic dependency distributional context extraction
Input: Feature Expression f_e;
A set of sentences S that contain f_e

For each sentence s which contains f_e in S:
1. *Parse s into a set of syntactic dependencies with the syntactic parser*
2. *Construct a dependency tree out of the set of syntactic dependencies and find out the root node*
3. *For each term t in feature expression f_e, collect the parent node and ancestor nodes*
4. *For each term t in feature expression f_e, collect the children nodes and descendant nodes*
5. *Construct these nodes into a contextual vector*

Output: the dependency contextual vector of f_e

Table II. Statistics of the semantic role labels

F	Product Features
O	Opinion expressions
MOD	Degree words that decorate opinion words
B	Otherwise

Table I. Statistics of the sentence structure tagging rules

Sentence Structure Types	Syntactic Dependency Relations	Tagging Rules
adjective-noun phrase structure	amod(noun, adjective)	noun: NPNN + adjective: NPADJ
subject-predicative structure	nsubj(adjective, noun) + cop(adjective, link verb)	noun: CNN + adjective: CADJ
adjective-complement structure	nsubj(verb, noun) + acomp(verb, adjective)	noun: CNN + adjective: CADJ
parataxis-nouns phrase	nn(noun1, noun2)	giving noun2 with the same tag as noun1
coordinate structure	conj_and(noun1, noun2) conj_and(adjective1, adjective2)	giving noun2 with the same tag as noun1 giving adjective2 with the same tag as adjective1
negative structure	neg(adjective, negative word)	adjective: NPADJ → adjective: NPADJNEG adjective: CADJ → adjective: CADJNEG
degree adverb-modifying-adjective structure	advmod(adjective, adverb)	adjective: NPADJ → adverb: NPADV adjective: CADJ → adverb: CADV

Bipartite Graph-based Similarity Calculation: Different from traditional TF-IDF similarity calculation, we transform the distributional contextual vectors into a bipartite graph model <*A*, *E*, *B*>: We denote the collection of product features as node set *A*, and the collection of contextual words as node set *B*; If the contextual word cw_j occurs in the distributional context of feature expression fe_i, there exists an edge from node fe_i in *A* to node cw_j in *B*. If there are *m* edges from *A* to node cw_j in *B*, then each of those m edges gets *1/m* weight score. The similarity is calculated by Eq. (4).

$$similarity_{context}(fe_1, fe_2) = \sum_{cw \in CooccurSet(fe_1, fe_2)} \frac{2}{OccurFreq(cw)} \quad (4)$$

Where $CooccurSet(fe_1,fe_2)$ is the collection of words that co-occur in the distributional contexts of fe_1 and fe_2. $OccurFreq(cw)$ represents the number of feature expressions which contain context word *cw* in their distributional contexts.

B. Categorizing Step

To aggregate and categorize the collection of feature expressions into some semantic feature groups, we introduce a simple but effective graph pruning-based categorizing strategy. We construct the set of product features into an weighted undirected graph $G = (V, E)$: *V* is the set of product features, each graph node represents a product feature; *E* is the set of weighted edges between nodes, each weighted edge represents the similarity between two feature expressions. Given the feature groups number *K*, then we repeatedly find and prune the minimum weighted edge until the connected branches number is more than *K*. The whole categorizing algorithm is presented in Algorithm 2.

Algorithm 2: product features categorization approach

Input: Product feature expressions set $Set_{pf}=\{ fe_1, fe_2, fe_3, \ldots\ldots, fe_n\}$

The product feature categorizations number K

1. *Calculate the similarity matrix for* Set_{pf}, *and construct it into a weighted undirected graph, graph nodes represent product features and edges represent the similarities between features;*
2. *Calculate the connected branches number CB, if CB <= K, go to step 3, otherwise go to step 4;*
3. *Find the minimum weighted edge and prune it, then go to step 2.*
4. *Construct each of the K connected branches into a aspect set, the categorization result is* $AspectSet_{pf} = \{Set_1, Set_2, \ldots\ldots, Set_K\}$.
5. *Return* $AspectSet_{pf}$ *and exit.*

Output: Categorization result $AspectSet_{pf} = \{Set_1, Set_2, \ldots\ldots, Set_K\}$

V. Empirical Evaluation

In this section, we mainly give out the empirical evaluation of our product features extraction and categorization strategies on real product reviews dataset. The dataset preparation and processing is firstly introduced, the extraction results and the categorization results are given out respectively secondly. At last, to illustrate the usability and effectiveness of our strategies, we analyze and give out the results comparison with some state-of-art counterpart methods.

A. Dataset setup

We have constructed a dataset of digital camera reviews contains four different components. Table III has shown the statistics of the reviews dataset. HL is selected from the two sets of digital camera reviews used in [2]; the other three components: YW1, YW2 and YW3, are crawled from online Amazon Website. A crawler has fetched the web pages and extracted all the review units. All the four dataset components have been tagged manually sentence by sentence as described in section 3.2.

B. Product Features Extraction Results

1) Extraction Baselines and Settings

To evaluate our product features extraction strategy, we introduce two baseline methods: NVPs+LM and CRFs. We denote our proposed strategy as CRFs+SSI, which represents we incorporate the sentence structure information into the CRFs learning process. The specification of two baseline methods is as follows:

Table III. Statistics of the reviews dataset

Dataset Components	No. of Reviews	No. of Sentences
HL	79	978
YW1	100	1098
YW2	163	974
YW3	182	1355

NVPs+LM: A primary idea of feature expressions extraction is based on the part-of-speech tags of natural language text. Through analyzing the dataset, we observe that most of the feature expressions are either noun phrases (NPs) or verb phrases (VPs). We introduce the product aspects extraction algorithm in [4] as a baseline, which selects all NPs and VPs as candidate product features. To shrink the size of candidate set, a language model is utilized to filter out low-score NPs or VPs chunk.

CRFs: To evaluate the contribution of sentence structure information, we employ the basic CRFs model used for product aspects extraction in [14] as another baseline system for comparison.

The common used Precision, Recall and F-Measure are exploited as the evaluation metrics in this work.

2) Extraction Results

Table IV shows the experimental results of Precision, Recall and F-Measure of CRFs+SSI strategy on the four dataset components respectively. Table V shows the results comparison of CRFs+SSI strategy with the two baseline methods.

C. Product Features Categorization Results

1) Categorization Baselines and Settings

By analyzing the labeled corpus, we find that many feature expressions are phrases consisting of multiple words and sharing some common words, e.g., “shutter delay” and “shutter lag”, “picture quality” and “picture clarity”, “price tag” and “price” etc. From the perspective of linguistics, these feature phrases with sharing words are more likely to belong to the same groups. We can exploit this pre-existing knowledge to assist the similarities calculation. The Eq. (5) is utilized to calculate this sharing-words similarity between two feature expressions.

$$Similarity_{sharingwords}(fe1,\ fe2) = \frac{Num_{common}}{Num_{fe1} + Num_{fe2} - Num_{common}} \quad (5)$$

Table IV. Results of the four dataset components by CRFs+SSI strategy

Data Set	Precision	Recall	F-Measure
HL	0.75	0.675	0.711
YW1	0.812	0.705	0.755
YW2	0.839	0.73	0.781
YW3	0.814	0.744	0.777
Average(Avg)	0.804	0.714	0.756

Table V. Results comparison of three different extraction strategies

Methods	Precision	Recall	F-Measure
NVPs+LM	0.428	0.855	0.570
CRFs	0.709	0.573	0.634
CRFs+SSI	0.804	0.714	0.756

In this section, we compare the effectiveness of two kinds of similarity measures described in section 4.1. Since sharing-words similarity belongs to pre-existing language knowledge, we always incorporate it into both kinds. At last, we give out the experimental performance of combining all these three sources of knowledge.

Sharing-words+WordNet: Combine sharing-words and WordNet-based similarities into the feature expressions similarity calculation.

Sharing-words+Distributional Context: Combine sharing-words and bipartite graph-based distributional similarities into the feature expressions similarity calculation.

Combination of All: Combine sharing-words, WordNet-based and bipartite graph-based distributional similarities into the feature expressions similarity calculation.

2) Categorization Evaluation Metrics and Results

As motivated by [12], we utilize Entropy and Purity to evaluate the categorization quality. The Entropy and the Purity are defined as following: Given a data set *DS*, its gold partition is $G = \{g_1, \ldots, g_j, \ldots, g_k\}$, where k is the given number of categories. The groups partition *DS* into k disjoint subsets: $DS_1, \ldots, DS_i, \ldots, DS_k$.

Entropy: For each resulting category, we measure its entropy using Eq. (6), where $P_i(g_j)$ is the proportion of g_j data points in DS_i. The total entropy of the whole categorizing (which considers all categories) is calculated by Eq. (7).

Purity: Purity measures the extent that a category contains only data from one gold-partition. Each resulting category’s purity is computed with Eq. (8). The total purity of the whole categorizing is computed with Eq. (9).

$$entropy(DS_i) = -\sum_{j=1}^{k} Pi(g_j)\log_2 P_i(g_j) \quad (6)$$

$$entropy_{total} = \sum_{i=1}^{k} \frac{|DS_i|}{|DS|} entropy(DS_i) \quad (7)$$

$$purity(DS_i) = \max_j P_i(g_j) \quad (8)$$

$$purity_{total} = \sum_{i=1}^{k} \frac{|DS_i|}{|DS|} purity(DS_i) \quad (9)$$

Table VI shows some categorization examples of product feature expressions categorizing results. Table VII shows the results comparison of different similarity calculation strategies during categorizing process.

Table VII. Results comparison of different similarity calculation methods

Method	Entropy	Purity	Avg
Sharing-words+WordNet	1.426	0.866	1.146
Sharing-words+Distributional Context	2.815	0.809	1.812
Combination of All	1.354	0.830	1.092

Table VI. Categorization examples of product features categorizing results

Categorization	Related feature expressions	Categorization	Related feature expressions
shutter delay	speed, shutter, shutter button, shutter delay, lag, shutter lag, shutter speed, shutter lag, lag time	lens cap	Lens, Lens cover, lens cap design, lens barrel, lens cap
battery	battery, battery duration, standard battery, rechargeable battery, battery indicator, battery charger, backup battery, battery life, charge system	photo / image	photo, picture, photo quality, picture quality, picture clarity, image, flash photo, photograph, raw image, image format
auto focus	auto focus, focus, focus system, light focus, focus range, auto focus delay, auto focus assist light	resolution	resolution, resolution setting, high resolution, resolution limit

D. Results Analysis

Table IV and V show the effectiveness and efficiency of our product features extraction strategy. From Table IV, we can compare the Precision, Recall and F-Measure of extraction results based on the four dataset components respectively. We can observe that our proposed CRFs+SSI strategy performs quite well on all the dataset components. YW1, YW2 and YW3 dataset components get almost the same performance, which is a little better than the HL component. We employ the average performance over all the four dataset components as the performance of our CRFs+SSI strategy over the whole review dataset. Table V shows result comparisons of three different extraction strategies. We could observe that our proposed CRFs+SSI strategy is much better than the NVPs+LM and CRFs baselines, which are the start-of-art techniques for product features extraction in reviews opinion mining. Compared by Recall, NVPs+LM gets the best Recall score which verified the observation that most of product features are in single phrases, but it achieves the worst Precision and F-Measure. Compared by Precision and F-Measure metrics, when part-of-speech and SSI features are incorporated into CRFs, our CRFs+SSI achieves the best extraction performance.

Table VI and VII show the effectiveness and efficiency of our product features categorization approach. From table VI, we can see that the product expressions are categorized according to feature groups precisely. For example, feature expressions-"photo", "picture", "photo quality", "picture quality", "picture clarity", "image", "flash photo", "photograph", "raw image" and "image format" etc., all have been categorized into the photo/image group correctly. Table VII shows the effectiveness comparisons of different similarity calculation strategies for the feature expressions categorization process. We can obtain the lowest Entropy and Average value when combine all these three measures for similarity calculation. We can see that the combination of sharing-words-based and WordNet-based similarity measures also achieves rather good performance, which proves that they play important roles in the combination of three. This is in accordance with their usages in [12, 20].

VI. Conclusions

In this paper, we have studied fine-grained product features extraction and categorization in online product reviews opinion mining. We propose to treat feature expression extraction as a sequence labeling task and employ a CRFs-based probabilistic model to tackle it. The part-of-speech features and sentence structure features are incorporated into the CRFs learning process innovatively. As feature expressions categorization is an unsupervised process, we utilize the semantic knowledge-based and the distributional context-based similarity measures to calculate the similarities and categorize product features into different feature groups. A combination of both measures is presented to achieve the best performance at last. The experimental results have demonstrated the effectiveness of our proposed model.

Acknowledgment

This work is partially supported by the grant from Key Foundation Research Projects of Beijing Institute of Technology (grant no.3070012231001), Basic Research Projects of Beijing Institute of Technology (grant no. 3070012211209), Graduate Students Research and Application Programs, Beijing Municipal Commission of Education (grant no.1320037010601) and the 111 Project of Beijing Institute of Technology.

References

[1] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," Proc. Proceedings of the First ACM International Conference on Web Search and Data Mining, ACM, Feb. 2008, pp. 231-240.

[2] M. Hu, and B. Liu, "Mining and summarizing customer reviews," Proc. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Aug. 2005, pp. 168-177.

[3] A. Popescu, and O. Etzioni, "Extracting product features and opinions from reviews," Proc. Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Oct. 2005, pp. 339-346.

[4] Y. Wu, Q. Zhang, X. Huang, and L. Wu, "Phrase dependency parsing for opinion mining," Proc. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning, Association for Computational Linguistics, Aug. 2009, pp. 1533-1541.

[5] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," Computational Linguistics, vol. 37, Mar. 2011, pp. 9-27.

[6] M. Hu, and B. Liu, "Opinion extraction and summarization on the web," Proc. Proceedings of the 21st National Conference on Artificial Intelligence, AAAI Press, Jul. 2006, pp. 1621-1624.

[7] B. Liu, M. Hu, and J. Cheng, "Opinion observer: analyzing and comparing opinions on the Web," Proc. Proceedings of the 14th International Conference on World Wide Web, ACM, May. 2005, pp. 342-351.

[8] S. Brody, and N. Elhadad, "An unsupervised aspect-sentiment model for online reviews," Proc. Proceedings of 2010 Human Language Technologies Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Jun. 2010, pp. 804-812.

[9] X. Zhao, J. Jiang, H. Yan, and X. Li, "Jointly modeling aspects and opinions with a MaxEnt-LDA hybird," Proc. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Oct. 2010, pp. 56-65.

[10] Z. Zhai, B. Liu, H. Xu, and P. Jia, "Constrained LDA for grouping product features in opinion mining," Proc. Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, May. 2011, pp. 448-459.

[11] H. Guo, H. Zhu, Z. Guo, X. Zhang, and Z. Su, "Product feature categorization with multilevel lantent semantic Association," Proc. Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM, Nov. 2009, pp. 1087-1096.

[12] Z. Zhai, B. Liu, H. Xu, and P. Jia, "Clustering product features for opinion mining," Proc. Proceedings of the 4th ACM International Conference on Web Search and Data Mining, ACM, Feb. 2011, pp. 347-354.

[13] W. Jin, H. H. Ho, and R. K. Srihari, "OpinionMiner: a novel machine learning system for web opinion mining and Extraction," Proc. Proceedings of the 15th ACM SIGKDD International Conference on

Knowledge Discovery and Data Mining, ACM, Jun. 2009, pp. 1195-1204.

[14] L. Qi, and L. Chen, “A linear-chain CRF-based approach for web opinion mining,” Proc. Proceedings of the 11th International Conference on Web Information Systems Engineering, Springer, Dec. 2010, pp. 128-141.

[15] Y. Lu, and C. Zhai, “Opinion integration through semi-supervised topic modeling,” Proc. Proceedings of the 17th International Conference on World Wide Web, ACM, Apr. 2008, pp. 121-130.

[16] J. Yu, Z. Zha, M. Wang, and T. S. Chua, “Aspect ranking: identifying important product aspects from online consumer reviews,” Proc. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association of Computational Linguistics, Jun. 2011, pp. 1495-1505.

[17] A. McCallum, “Efficiently inducing features of conditional random fields,” Proc. Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, Aug. 2003, pp. 403-410.

[18] L. John, M. Andrew, and P. Fernando, “Conditional random fields: probabilistic models for segmenting and labeling sequence data,” Proc. Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufmann Publishers, Jun. 2001, pp. 282-289.

[19] J. Jiang, and D. Conrath, “Semantic similarity based on corpus statistics and lexical taxonomy,” Proc. Proceedings of International Conference of on Research in Computational Linguistics, Association for Computational Linguistics, 1997, pp. 19-33.

[20] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa, “A study on similarity and relatedness using distributional and WordNet-based approaches,” Proc. Proceedings of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association of Computational Linguistics, May. 2009, pp. 19-27.

[21] T. Pedersen, S. Patwardhan, and J. Michelizzi, “WordNet::Similarity: measuring the relatedness of concepts,” Proc. In Demonstration Papers at Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting, Association for Compuational Linguistics, May. 2004, pp. 38-41.

[22] E. Cambria, A. Hussain, T. Durrani, and J. Zhang, “Towards a chinese common and common sense knowledge base for sentiment analysis,” Proc. Proceedings of the 25th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems: advanced research in applied artificial intelligence, Springer, Jun. 2012, pp. 437-446.

[23] T. Li, Y. Zhang, and V. Sindhwani, “A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge,” Proc. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Conference on Natural Language Processing of the AFNLP, Association of Computational Linguistics, Aug. 2009, pp. 244-252.

[24] T. Li, V. Sindhwani, C. Ding, and Y. Zhang, “Briding domains with words:opinion analysis with matrix tri-factorizations,” Proc. Proceedings of the SIAM International Conference on Data Ming, SIAM, Apr. 2010, pp. 293-202.