Kernel-based Sentiment Classification for Chinese Sentence

Linlin Li Tianfang Yao

Department of Computer Science and Engineering,
Shanghai Jiao Tong University, China
ridingwind@sjtu.edu.cn yao-tf@cs.sjtu.edu.cn

Abstract

Recent years have seen a large growth in the online customer reviews. Classifying these reviews into positive or negative ones would be helpful in business intelligence applications and recommender systems. This paper aims to solve the sentiment classification at a fine-grained level, i.e. the sentence level. The challenging aspect of this problem that distinguishes it from the traditional classification problem is that sentiment expression is more free-style. Classification features are more difficult to determine.

In this paper, we propose a kernel-based method to make it is feasible for incorporating multiple features from word, n-gram and syntactic levels. Experiment results show that our method is effective, and it outperforms the very competitive n-gram method.

1. Introduction

With the widely spread use of Internet, a large amount of information is available online. Lots of research focuses on data mining of the online information resource, such as topic categorization, text summarization. However, recent years have seen rapid growth in review sites where the obvious feature of posted articles is their sentiment, or overall opinion towards the subject matter. Classifying these articles into positive or negative ones would be helpful in business intelligence applications and recommender systems, where user input and feedback could be quickly summarized.

In this paper, we presented an effective machine learning strategy for classifying the reviews into positive or negative ones. Our method differs from lots of former research in that our processing unit is sentence, which means that we are trying to solve the sentiment classification problem using a more fine-grained analysis.

A lot of classification problems are identifiable from some keywords, such as in topic-based classification problem, some keywords that have high TF-IDF scores may be used for topic-based classification, but sentiment expressions are quite contextual.

Although there are a lot of sentiment expressions containing sentiment words, there are still a large proportion of sentences that don't have sentiment words but still convey positive or negative attitude. For example:

这车值 130000RMB ? !

(This car is worthy 130,000 RMB?!)

No sentiment word appears in this sentence, but we can still find a negative attitude expressed by the author. From our study, sentiment detection is a complex problem which needs to be done in several processing levels, which are word, phrase, syntactic and semantic levels.

In this paper, we proposed a kernel-based method. Our main contribution is that we utilize the kernel-based method to makes it is feasible for incorporating multiple features from word, n-gram and syntactic levels. Experiment results show that our method is effective, and it wins the competition with the very competitive n-gram method.

The following sections are arranged in this way: in section 2, we introduce some related work; section 3 gives a brief introduction of the kernel method; in section 4, we give our methodology; section 5 shows the experiment results and some analysis; in the last two sections, we talk about the conclusion and the future work.

2. Related Work

Sentiment classification could be done in either word/phrase level, sentence level or document level.

Some of this work focuses on classifying the semantic orientation of individual words or phrases, using linguistic heuristics or a pre-selected set of seed words [1, 4]. [1] develops an algorithm for automatically recognizing the semantic orientation of adjectives. [12] identifies *subjective* adjectives (or sentiment adjectives) from corpora. Some other researchers use WordNet for word semantic orientation study. The adjective synonym and antonym set of indicators in WordNet are used to predict sentiment orientations [2, 3]. [8] uses distances among lexicons in WordNet to reveal relations of affective or emotive meanings, and to predict sentiment orientations. Our work is different from their ones in that we are doing sentiment detection in sentence level other than word/phrase level.

Most of the past work on document level sentiment-based categorization assumes that an entire document is only about one subject, and applies existing text classification algorithms. They often involve the use of the manual or semi-manual construction of discriminate-word lexicons [13, 14, 15]. [6] compares the performance of three machine learning methods (Naive Bayes, maximum entropy classification, and SVM). [16] applies text categorization only on the subjective portions of the document. [4] uses the average “semantic orientation” of the phrases to predicate the overall orientation of the review. [17] analyzes emotional affect of various corpora computed as average of affect scores of individual affect terms in the articles. [18] extends a sentiment classification model that utilizes unlabeled documents and human-provided information as well as labeled documents. [6, 9] integrated n-gram features to the document level classifier and achieved a high performance. Most of the sentiment classifiers assume that each document has only one subject, and the subject of each document is known. However, these assumptions are not always true, especially for web documents. As a result, researches aimed at more fine-grained level, the sentence level.

Some researchers have tried document level models on sentence level analysis, but results turned out to be poor [9], which proved that additional features are needed to be incorporated for sentence level classification. [7, 10] uses substrings extracted from constitute or dependency structures and improves the performance of the sentence level model. [11] also mentions that it is necessary to integrate more features on sentence level sentiment classification. In this paper, we will introduce our method for how to incorporate multiple features in sentiment classification at sentence level.

3. Kernel Method

Many machine learning algorithms involve only the dot product of vectors in a feature space, in which each vector represents an object in the object domain. Kernel methods can be seen as a generalization of feature-based algorithms, in which the dot product is replaced by a kernel function (or kernel) $\Psi(X, Y)$ between two vectors, or even between two objects. Mathematically, as long as $\Psi(X, Y)$ is symmetric and the kernel matrix formed by Ψ is positive semi-definite, it forms a valid dot product in an implicit Hilbert space. In this implicit space, a kernel can be broken down into features, although the dimension of the feature space could be infinite. So kernel function is a nice choice to solve the high dimensional feature classification problem.

Kernel functions have many nice combination properties as well. For example, the sum or product of existing kernels is a valid kernel. With these combination properties, we can combine individual kernels

representing information from different sources in a principled way.

Figure 1 shows the way how kernel method works compared with the traditional feature-based algorithm.

Figure 1. Kernel-based VS feature-based algorithm

4. Methodology

In this section, we will introduce the methodology and the idea behind it. Firstly, several related concepts will be introduced, and then, we will give the explicit definition of our kernels.

4.2. Definitions

Word (w) is defined as a 3-tuple $w = (t, pos, s)$, where t is the original string of the word; pos is the part-of-speech; s is the semantic orientation, with value 1 meaning positive, -1 negative, and 0 neutral.

The similarity between two $w(s)$ is defined as:

$$K_w(w, w') = \begin{cases} \beta_1 \times I(w.t, w'.t) + \beta_2 \times I(w.pos, w'.pos); & w.s = 0, w'.s = 0 \\ \beta_1 \times I(w.t, w'.t) + \beta_2 \times I(w.pos, w'.pos) + \beta_3 \times I(w.s, w'.s); & \text{otherwise} \end{cases}$$

where,

$$I(t_1, t_2) = \begin{cases} 1, t_1 = t_2 \\ 0, t_1 \neq t_2 \end{cases}$$

where β_i is a weight, and $\sum_i \beta_i = 1$.

Sequence (seq) is defined as the sequence of several consecutive words,

$$seq = (w_1, \dots, w_{key}, \dots, w_n),$$

where w_{key} is the key word which influences the similarity of seq more than other types of words and should be assigned more weight. We can also consider seq as an n-gram.

The similarity between two $seq(s)$ is defined as:

$$K_s(\text{seq}, \text{seq}') = \beta_1 \times K_w(\text{seq}.w_{\text{key}}, \text{seq}'.w_{\text{key}}) +$$

$$\beta_2 \times \frac{1}{n-1} \sum_{i=1}^{n-1} K_w(\text{seq}.w_i, \text{seq}'.w_i)$$

where β_i is a weight, and $\sum_i \beta_i = 1$.

Node (n) is defined as a node in the syntactic tree,
 $n = (w, l, p, c)$

where w is the word in the node; l is the dependency relation to the parent node; p is the parent node; $c = (n_1, n_2, \dots, n_i)$ is the child list whose element is also a type of node.

PathNode (pn) is defined as the node in the syntactic path:

$$pn = (n, e)$$

where n is the syntactic node; e is the direction of the dependency, with value 1 for a backward dependency and 0 for a forward dependency.

Path (pa) is defined as the syntactic path from the source node n_1 to the destined node n_n ,

$$pa = (pn_1, \dots, pn_{\text{key}}, \dots, pn_n),$$

where pn_{key} is a key pn which influences the similarity between two $pa(s)$ more than any other pn .

The similarity between two $pn(s)$ is defined as:

$$K_{pn}(pn, pn') = ((\beta_1 \times I(pn.n.l, pn'.n.l) + \beta_2 \times K_w(pn.n.w, pn'.n.w)) \times I(pn.e, pn'.e))$$

The similarity between two $pa(s)$ is defined as:

$$K_p(pa, pa') = \beta_1 \times K_{pn}(pa.pn_{\text{key}}, pa'.pn_{\text{key}}) + \beta_2 \times \sum_{pn_i \notin pn_{\text{key}}} \sum_{pn_j \notin pn_{\text{key}}} K_{pn}(pa.pn_i, pa'.pn_j)$$

where β_i is a weight, and $\sum_i \beta_i = 1$.

Union Similarity (US) is the similarity between two unions whose elements belong to the same category. It is calculated as the follows:

When $s_2 \geq s_1$,

$$\text{Sim}(U_1, U_2) = \frac{1}{s_2} (s_2 - s_1) * \delta +$$

$$\frac{1}{s_2} \sum_i \max_j \{\text{Sim}(e_i, e_j)\}$$

where

$$e_i \in U_1, 1 \leq i \leq s_1;$$

$$e_j \in U_2, 1 \leq j \leq s_2;$$

where s_1 is the element number of the first union; s_2 is the element number of the second union; δ is a very little

positive number meaning the similarity value between a non-empty value and the empty value. $\text{Sim}(e_i, e_j)$ is the similarity value between two elements. When $s_1 \geq s_2$, we change the position of corresponding parameter and got a similar definition.

Then, we define three word unions:

U_s^w : All the sentiment words;

U_a^w : All the adjectives except the sentiment ones;

U_v^w : All the verbs except the sentiment ones.

Given $w_i^m \in U_m^w, m = \{s, a, v\}$, we define seq_i^m, pa_i^m as the corresponding seq and pa of w_i^m , which satisfy

$$\text{seq}_i^m = (w_{i-n/2}^m, \dots, w_i^m, \dots, w_{i+n/2}^m);$$

pa_i^m : Path from w_i^m to its nearest noun or verb.

We define three seq unions

$$U_s^{\text{seq}} = \{\text{seq}_i^s\}_i, U_a^{\text{seq}} = \{\text{seq}_i^a\}_i, U_v^{\text{seq}} = \{\text{seq}_i^v\}_i,$$

and three pa unions

$$U_s^{\text{pa}} = \{pa_i^s\}_i, U_a^{\text{pa}} = \{pa_i^a\}_i, U_v^{\text{pa}} = \{pa_i^v\}_i.$$

4.2. Kernels

In our study, sentiment words, adjective and verb are good indicators of sentiment. We choose these three word unions from each input sentence, and make use of the information from the word, n-gram, and syntactic levels to calculate the similarity between two input sentences.

Now, we define our kernel functions. Since the kernel function is to measure the similarity between two instances, the effective kernel function should have high values between instance with the same polarity and low values between instances with opposite polarities.

We define three kernels to match polarity examples at word, phrase and syntactic levels. Using the notation just defined, we can write the three kernels as follows:

1) Word kernel

$$\Psi_1(I_1, I_2) = \sum_{k=s,a,v} \beta_k \text{Sim}(I_1.U_k^w, I_2.U_k^w),$$

where

$$\beta_i \text{ is a weight; } \sum_{i=1}^3 \beta_i = 1.$$

We define the word kernel to compute the similarity between two instances by computing the similarity of the three specific word unions: the sentiment word union, the adjective union, and the verb union. By assigning different weights to different unions, we can observe the importance of each word union for sentiment classification.

2) N-gram kernel

$$\psi_2(I_1, I_2) = \sum_{k=s,a,v} \beta_k \text{Sim}(I_1 \cdot U_k^{\text{seq}}, I_2 \cdot U_k^{\text{seq}})$$

where

$$\beta_i \text{ is a weight; } \sum_{i=1}^3 \beta_i = 1.$$

What makes the n-gram kernel different from the word kernel is that we choose the n-gram instead of the single word as the unit for similarity calculation. The intuition behind this is that instances sharing the same polarity may have similar n-grams.

3) Path kernel

$$\psi_3(I_1, I_2) = \sum_{k=s,a,v} \beta_k \text{Sim}(I_1 \cdot U_k^{\text{pa}}, I_2 \cdot U_k^{\text{pa}})$$

where

$$\beta_i \text{ is a weight; } \sum_{i=1}^3 \beta_i = 1.$$

In the path kernel, we compute the similarity between two input sentences by calculating some key syntactic sub-structures. We use three heuristics (i.e. three kinds of sub-structures) as the basis to compute path kernel

- Path from sentiment word to its nearest noun or verb;
- Path from adjective to its nearest noun or verb;
- Path from verb to its nearest noun.

5. Experiment

5.1. Resource

We build our corpus using data from *Chinacars*¹. Experiments were carried out on 589 positive and 600 negative instances with hand-annotated labels. In this section we will compare the performance of different kernel setups trained with SVM. The SVM package we used is SMO, and the classification tool we use is *weak* [24]. Evaluation of kernels was done on the training data using 10-fold cross-validation.

Three pre-processing tools we used are word segmentation tool, part-of-speech tagger and syntactic parser, all of which are shared by Harbin Institution of Technology.

Sentiment words are gotten from HowNet. All the words in HowNet with the “desired” or “undesired” attribute are taken as sentiment words. We got 8078 positive words and 8457 negative ones, of which 77.6% are adjectives, and 20.8% are nouns.

5.2. Results

The standard that we use to evaluate the classification results are precision, recall and F-Score. Their definitions are as table 1.

We built three baselines for our system. In baseline 1, the classifier gives a random label either positive or negative to the input instances. Baseline 2 is an n-gram classifier. In baseline 3, the classifier assigns the labels according to the number of the sentiment words, positive when positive words more, negative when negative words more, or neutral when the numbers are the same.

Table 1. Evaluation standard

		classifier	
		1	0
label	1	a	b
	0	c	d

$$\text{Class 1: precision} = \frac{a}{a+c} \quad \text{recall} = \frac{a}{a+b}$$

$$\text{Class 0: precision} = \frac{d}{b+d} \quad \text{recall} = \frac{d}{c+d}$$

Table 2 shows the performance of the three baselines. We can see that n-gram gains the best results, which agree with a lot of former studies on classification study.

Table 3 shows the performance of our kernel functions. Our system, in which we made combined uses of the information that indicates sentiment, outperforms the best baseline system. To show how our system works, the following two sentences are given as an example:

这车漂亮而且省油。

(This car is nice-looking and energy-saving.) (1)

这车省油。

(This car is energy-saving.) (2)

If the sentiment dictionary does not contain the word “省油 (energy-saving)”, but the word “漂亮 (nice-looking)”, we will fail to get the sentiment of the second sentence. Our system solves the problem by calculating the similarity of the two sentences. Sentences with high similarity values will be classified as the same class.

In the above example, the sentiment of sentence 1 is easy to find since it contains the sentiment word “漂亮 (nice-looking)”. Sentence 2 will be classified as the same group with sentence 1 since they share the same adjective word “省油 (energy-saving)”, and have a relatively high similarity value.

In our kernels, ψ_2 and ψ_3 are the extension of ψ_1 since they do not only take the words into consideration but also the context or syntactic information. The fact that ψ_2 outperforms ψ_1 indicates that some context information is good indicators of sentiment. The very disappointing

¹ <http://www.chinacars.com/>

Table 2. Performance on three baselines

	B1			B2			B3		
	Prec.	Recall	F-score	Prec.	Recall	F-score	Prec.	Recall	F-score
Pro.	50.6%	50.8%	50.7%	68.2%	51.4%	58.7%	64.1%	58.2%	61.0%
Con.	51.5%	51.3%	51.4%	72.9%	84.5%	78.2%	79.0%	16.3%	27.0%

Table 3. Performance on our kernel function

	ψ_1			ψ_2			ψ_3		
	Prec.	Recall	F-score	Prec.	Recall	F-score	Prec.	Recall	F-score
Pro.	80.2%	63.3%	70.8%	83.1%	63.7%	72.1%	79.5%	59.1%	67.8%
Con.	70.2%	84.7%	76.7%	71.0%	87.3%	78.3%	67.9%	85.0%	75.5%

Table 4. SVM performance on different feature weight setups

Performance		Pro.			Con.		
		Prec.	Recall	F-score	Prec.	Recall	F-score
β	(1, 0, 0)	80.3%	54.5%	64.9%	66.0%	86.8%	75.0%
	(0, 0.5, 0.5)	62.8%	65.2%	64.0%	64.5%	62.2%	63.3%
	(1/3, 1/3, 1/3)	80.2%	63.3%	70.8%	70.2%	84.7%	76.7%

part of our experiment is that ψ_3 lost the competition with ψ_1 . We find there is a high error rate of the syntactic parser by manual sample checking. We think this high parsing error rate may be the main cause of ψ_3 's poor performance. The wrong parsed structure failed to indicate the sentiment information. Another possibility is that our definition of ψ_3 is not sufficient to capture the sentiment information. Therefore, the improvement of the definition could be a good future work.

Table 4 shows the performance of the SVM on different weight setup within kernel ψ_1 . $\beta = (a, b, c)$ is the weight vector. a , b , and c are the weights of the sentiment words, the adjectives, and the verbs. We could get two conclusions from this experiment result: (1) sentiment words are good indicators of sentiment; (2) adjective and verb help to improve the performance of sentiment classification.

6. Conclusion

Our kernel method is an effective way for sentiment classification. It outperforms the most widely believed best system, the n-gram. It is also different from lots of sentiment classification methods, since it is not just relied on a fixed sentiment dictionary. Our proper defined kernel functions can capture sentiment information from n-gram and syntactic levels. These kinds of information are closer to the semantic analysis and more suitable for sentiment classification task, which is semantic demanding and different from traditional topic-based classification task.

7. Future Work

As mentioned above, the kernel function ψ_3 didn't gain the performance as we expected. We want to further address this problem in our future work. We may develop some methods to decrease the influence by parser error, such as choosing smaller substructures. We will also try to modify the form of our syntactic kernel function to observe the result of different syntactic kernel setups.

Acknowledgements

This work gets a lot of support from the Language Technology Lab of the German Research Center for Artificial Intelligence (DFKI). Here we would like to express our heartfelt thanks to DFKI for all its help throughout the work.

References

- [1] Vasileios Hatzivassiloglou and Kathy McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 174C181.
- [2] Soo-Min Kim, and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pp. 1367-1373, Geneva, Switzerland.
- [3] Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271-278.
- [4] Peter D. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of*

- the Association for Computational Linguistics (ACL-02), pp. 417-424, Philadelphia, Pennsylvania.
- [5] Peter D. Turney and Michael L. Littman. 2002. Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus.
- [6] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, pp. 79-86, Philadelphia, Pennsylvania.
- [7] Taku Kudo and Yuji Matsumoto. 2004. A boosting algorithm for classification of semistructured text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pp. 301-308, Barcelona, Spain.
- [8] Jaap Kamps, Maarten Marx, Robert J. Mokken and Marten de Rijke. 2002. Words with attitude. In *Proceedings of the 1st International Conference on Global WordNet*, Mysore, India
- [9] Kushal Dave, Steve Lawrence, David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International World Wide Web Conference (WWW2003)*, Budapest, Hungary. Available at <http://www2003.org>.
- [10] Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. 2005. Sentiment classification using word subsequences and dependency sub-trees. In *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-05)*, pages 301-310.
- [11] Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pp. 412-418, Barcelona, Spain.
- [12] J. M. Wiebe. Learning subjective adjectives from corpora. In *Proceedings of the AAAI Conference*, 2000.
- [13] S. Das and M. Chen. Yahoo! for amazon: Extracting market sentiment from stock message boards. In *Proceedings of the APFA*, 2001.
- [14] R. M. Tong. An operational system for detecting and tracking opinions in on-line discussion. In *SIGIR Workshop on Operational Text Classification*, 2001.
- [15] A. Huettner and P. Subasic. Fuzzy typing for document management. In *Proceedings of the ACL Conference (Software Demonstration)*, 2000.
- [16] B. Pang and L. Lee. A sentiment education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL Conference*, 2004.
- [17] L. Rovinelli and C. Whissell. Emotion and style in 30-second television advertisements targeted at men, women, boys, and girls. *Perceptual and Motor Skills*, 86:1048-1050, 1998.
- [18] P. Beineke, T. Hastie, and S. Vaithyanathan. The sentimental factor: Improving review classification via human-provided information. In *Proceedings of the ACL Conference*, pages 263-270, Barcelona, Spain, 2004.
- [19] Shubin Zhao, Ralph Grishman. Extracting relations with integrated information using kernel methods. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, p.419-426, June 25-30, 2005, Ann Arbor, Michigan
- [20] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, Chris Watkins. Text classification using string kernels, *The Journal of Machine Learning Research*, 2. p.419-444, 3/1/2002
- [21] Aron Culotta, Jeffrey Sorensen. Dependency tree kernels for relation extraction. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, p.423-es, July 21-26, 2004, Barcelona, Spain
- [22] Razvan C. Bunescu, Raymond J. Mooney. A shortest path dependency kernel for relation extraction. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, p.724-731, October 06-08, 2005, Vancouver, British Columbia, Canada
- [23] Basili R., Cammisa M. and Moschitti A. A Semantic Kernel to classify text with very few training examples. *ICML-2005*
- [24] <http://www.cs.waikato.ac.nz/ml/weka/>