Learning Sentimental Weights of Mixed-gram Terms for Classification and Visualization

Tszhang Guo[1,2], Bowen Li[1], Zihao Fu[3], Tao Wan[1,4(✉)], and Zengchang Qin[1(✉)]

[1] Intelligent Computing and Machine Learning Lab, School of ASEE, Beihang University, Beijing 100191, China
guozikeng@foxmail.com, libowen.ne@gmail.com, tao.wan.wan@gmail.com, zengchang.qin@gmail.com

[2] Department of Automation, Tsinghua University, Beijing, China

[3] Alibaba Group, Beijing 100022, China
fuzihaofzh@163.com

[4] School of Biological Science and Medical Engineering, Beihang University, Beijing, China

Abstract. Sentimental analysis is an important topic in natural language processing and opinion mining. Many previous studies have reported to judge whether a term is with emotion or not. However, little work has been done in measuring degrees of sentiment for these terms. For example, the word *excellent* has stronger positive sentiment than the word *good* and *okay*. In this paper, we investigate how to model this intricate sentimental difference by assigning sentimental weights. A simple and effective model is proposed based on logistic regression to extract emotional terms associated with sentiment weights. Weighted terms can be used in sentiment classification and visualization by drawing emotional clouds of texts. The new model is tested using uni-gram, bi-gram and mixed-gram language models on two benchmark datasets. The empirical results show that the new model is highly efficient with comparable accuracy to other sentiment classifiers.

1 Introduction

Sentiment analysis is an important task in natural language processing (NLP) for analyzing people's opinions, attitudes and emotions towards certain services and products [1]. When we think of emotive reviews or comments, we are inclined to think of predicates of personal taste (*boring, fun*), exclamatives (*awesome, damn*) and other *emotional* words or terms that are more or less contributed to convey sentimental information of our opinions. Once we get the knowledge of these dominant emotional terms, we could judge the sentimental polarity (positive or negative) of a given sentence or document. Unfortunately, for particular utterance in a given context, such significant emotional terms are not always so apparent. We hope to find a way to automatically extract these emotional terms from corpus.

Sentiment classification can be simply considered as a text classification task. Most previous studies focused on using supervised machine learning methods

R. Booth and M.-L. Zhang (Eds.): PRICAI 2016, LNAI 9810, pp. 116–124, 2016.
DOI: 10.1007/978-3-319-42911-3_10

based on N-gram language model to do sentiment classification [2,3]. In movie reviews, we can often see terms like *worth watching*, that expresses strong positive emotion. Though the word *watching* is emotionally neutral, it will be assigned with positive weight alone with *worth*. It may become troublesome in sentences containing *watching* only. In this paper, we develop a mixed-gram model by embedding both uni-gram and bi-gram together to tackle this problem. Therefore, the bi-gram term *worth watching* will be assigned with large positive weight without separating them. We propose a simple but effective approach to extract sentiment terms by assuming that every distinct term in the corpus has an unique sentiment weight and the sentiment of a sentence or document is determined by a nonlinear combination of sentiment weights. We employ logistic regression to estimate sentiment weights of terms and use these weights for predictions.

For sentiment weight extraction, some studies have been done include the following: [4] did the work on tackling the problem of determining the semantic orientation (or polarity) of words; [5] proposed an approach to find subjective adjectives using the results of word clustering according to their distributional similarity; [6] proposed a double propagation algorithm to expand opinion words and extract target words. There are also some related work in sentiment lexicon extraction: [7,8] showed that supervised learning methods can achieve state-of-the-art results for lexicon extraction. In the domain-specific lexicon extraction, [9] got significant improvement by using active learning method. [10] used logistic regression (LR) to get the terms' weights corresponding to different ratings. [11] focused on the relevant weights of sentences in a given document for aspect rating prediction.

In terms of representation of documents, BoW could nicely reduce a piece of text with arbitrary length to a fixed length vector. In recent years, by exploiting the co-occurrence pattern of words, embedding model was employed to gain lower-dimensional, compact and meaningful vectors for words or documents [12]. Deep neural network approaches like convolutional neural network (CNN) can also bring significant improvement to sentiment classification task [13]. However, deep learning (DL) methods are always computationally expensive. In this paper, we hope to develop a simple and efficient learning model for intuitive sentiment visualization of a text with comparable results to the DL methods in sentiment classification.

2 Polarity Model of Sentiment

Given a sentimental text of being positive or negative, it is easy to see that the sentiment contributions of its consisting terms are different. Some terms like *excellent, good* occur more often in positive documents, and terms like *bad, horrible* occur more often in negative documents. This implies that such terms have high sentimental contributions and should be assigned with large sentiment weights. For some objective nouns and action verbs like *take, walk*, very likely they appear equally in both positive or negative documents, therefore, they contribute less to the sentiment of a text. Such terms are neutral and should

be assigned with small sentiment weights. In this polarity model, we assume that the sentiment of a sentence is a function of the sentimental weights of its consisting terms. A term can be either one word in uni-gram model or two words in bi-gram model, or even mixed-gram of both.

Given a sentence of N different terms, the associated sentiment weight of term t_i is denoted by w_i for $1 \leq i \leq N$. The sentiment score h of the given sentence is:

$$h = f\left(\sum_{i=1}^{N} w_i x_i\right) = f(\mathbf{w}^T \mathbf{x}) \tag{1}$$

where x_i is the feature value of a given term t_i. It could be term frequency, binary value (appears in the particular document or not) or the TF-IDF value of the term.

Moreover, the sentiment polarity of a term x_i can be defined according to its sentiment weight w_i. It is not easy to set thresholds among positive, neutral and negative, as it is quite data dependent. We will test thresholding based on sentiment weights in Sect. 3. Function $f(\cdot)$ is a nonlinear function to smooth the linear combination of the sentimental weights.

2.1 Sentiment Weight Learning

In this section, we are going to use logistic regression (LR) and Gradient Descent algorithm (GD) to learn the sentiment weight based on given training data. We use $\mathbf{w} = \{w_1, w_2, \ldots, w_N\}$ to represent the weight vector where $w_i : i \in \{1, 2, \ldots, N\}$ is the sentimental weight of term t_i, where $T = \{t_1, t_2, \ldots, t_N\}$ is a set of terms in a corpus based on uni-gram or bi-gram. We use $y \in \{0, 1\}$ to represent the document's sentiment label (0 for negative and 1 for positive), and h to represent the sentiment score of the given document. $\mathbf{x}_j = \{x_{j1}, x_{j2}, \ldots, x_{ji}, \ldots, x_{jN}\}$denotes the j^{th} document's feature vector where x_{ji} represents the i^{th} term's feature value in the j^{th} document. We can initialize $\mathbf{w}$ randomly and calculate sentiment score by using logistic function:

$$h_j = \frac{1}{1 + \exp\left(-\mathbf{w}^T \mathbf{x}_j\right)} \quad ; \quad 1 \leq j \leq M \tag{2}$$

where M is the total number of documents in the training corpus. In order to minimize the squared error: $E = \frac{1}{2}\sum_{j=1}^{M}(h_j - y_j)^2$. we can update $\mathbf{w}$ given h_i, y_i and x_i using the Gradient Descent algorithm: $\mathbf{w} := \mathbf{w} - \alpha(h_j - y_j)\mathbf{x}_j$ where α is the learning rate. We iterate the process until convergence and use the final $\mathbf{w}$ to predict the new unseen document's score $\hat{h}$ by Eq. (2). We then can predict the sentiment label by:

$$y = \begin{cases} 0 & if \quad \hat{h} < 0.5 \\ 1 & otherwise \end{cases} \tag{3}$$

In addition, the computational complexity of LR is $O(|V|)$, where $|V|$ is the size of vocabulary.

2.2 Sparsity Constraints

As we have discussed, given a text, there are few words with strong sentiment. Most words including nouns and verbs are neutral. Therefore, we can put a constraint of sparsity in learning sparse sentimental weights using logistic regression. We can use L-BFGS method to minimize the following cost function in ℓ_1 (or ℓ_2) norms:

$$\min_{\mathbf{w}} \left[C||\mathbf{w}||_1 - \frac{1}{M} \sum_{j=1}^{M} (y_j \log(h_j) + (1 - y_j) \log(1 - h_j)) \right] \tag{4}$$

By such constraints, we hope to learn a sparse vector $\mathbf{w}$ in which zero values represent neutral terms and terms with non-zero values carry the sentiment.

2.3 Mixed-gram Model

Previous work shows that bi-gram generally performs better than uni-gram, since bi-gram has more semantic information of word order or word position [3]. As we have seen before, terms like *good movie* or *bad script* often appear in pairs in reviews or comments. In uni-gram model, since the word *movie* appears together with *good* in positive examples, it is likely to be assigned sentimentally positive value. The neutral word *movie* may become problematic in classification. However, if we consider the same example in bi-gram model, the positive sentiment value will be assigned to *good movie*. The word comes after *moive* could be very random and it won't be biased on certain special bi-gram terms starting with *movie*. Ideally, weights should be assigned to sentimentally segmented terms. For example, *This is a good movie, actors are excellent!* should be segmented to: *This*|*is a*|*good movie*|*actors*|*are*|*excellent*|! This may bring a new problem of sentimental segmentation and it is beyond the scope of this paper. In this paper, we simply use mixed-gram model that is a mixture of both uni-gram and bi-gram.

2.4 Visualization of Emotional Word Cloud

Word cloud is a visualization form for text that is recognized for its aesthetic, social, and analytical values. Here, we are concerned with deepening its analytical value for visual comparison of documents. A word is expected to have the same color and position across word clouds. This aims to reduce the cognitive effort needed for comparing word clouds. However, it has two shortcomings. First, it only seeks to synchronize the appearance of each distinct word. This is problematic, as text frequently uses different words to refer to the same concept. Second, its synchronization of all word clouds imposes sizeable runtime requirement that prevents real-time generation of word clouds. These issues arise because word clouds are still high-dimensional representations, with dimensionality the size of the vocabulary. In this paper, we use the color (from green to red) feature to represent sentimental weights and the size of a word is determined by its TF-IDF value.

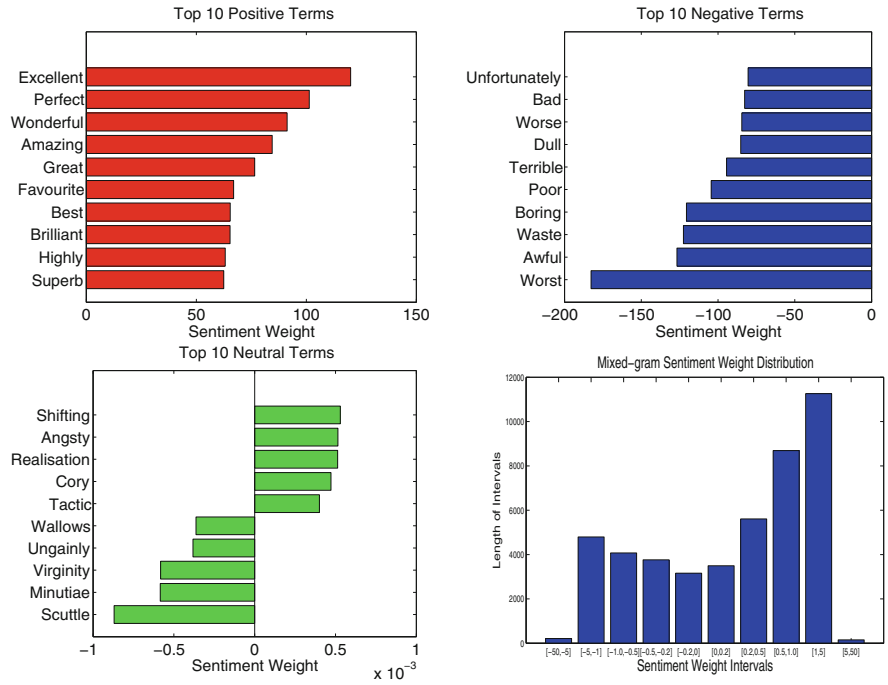

Fig. 1. Upper: Top 10 positive and negative terms in the IMDB corpus. *Excellent* is the strongest positive term and *Worst* is the strongest negative term. Below: Top 10 neutral terms and weight distribution of mixed-gram on the IMDB dataset (Color figure online)

3 Experimental Studies

We choose two benchmark datasets on sentiment classification in our experiments. The first is the IMDB dataset of online movie reviews [14], it contains 25000 reviews (12500 positives and 12500 negatives) for training and 25000 (12500 positives and 12500 negatives) for test. The second dataset is the Product Review including DVD, electronics, books and kitchens, and each of them contains 1000 positive and 1000 negative reviews. Stop words are not removed in our text preprocessing.

3.1 Sentiment Term Extraction

We first test the model on the IMDB dataset. Sentiment weight for each term is estimated based on uni-gram, bi-gram and mixed-gram, respectively. The top 10 results of uni-gram are shown in Fig. 1 and top 20 sentiment terms of mixed-gram are listed in Table 1. It is easy to see that our model successfully extracts terms with strong sentiment and the results are intuitively agree with human

perceptions. Weights for positives terms are positive values and weights for negative terms are negative values. Weights for neutral terms are close to zero. The learned term weights have a big variance, which indicates that a small set of terms carrying strong sentiment than other terms.

Table 1. Top 20 sentiment terms based on the mixed-gram model.

Polarity	Sentiment terms (Mixed-gram)
Positive	great, the best, excellent, perfect, wonderful, amazing, a bit, a great well worth, is a, a must, fun, my favorite, today, very good brilliant, definitely worth, is great, very well, superb
Negative	the worst, bad, worst, awful, boring, poor, no, terrible waste, nothing, waste of, at all, worse, not even, dull, horrible poorly, stupid, annoying, lame

Fig. 2. Emotional word clouds of positive (first 2 figures) and negative reviews (the last figure) generated from the IMDB dataset. (Color figure online)

Particularly, in the mixed-gram model, *well worth*, *a must*, *not even* and *at all(not ... at all)* are valuable patterns to determine the sentiment orientation. It should be noted that stop words cannot be filtered to recognize such patterns. Though some neutral uni-grams or bi-grams are wrongly classified as positive (e.g. *today*, *is a*) or negative (e.g. *script*). One possible reason is that these neutral uni-grams or bi-grams occur frequently in some sentimental contexts. For example, *job* is more likely to occur in the forms of *good job* or *great job*; *script* is more likely to be discussed when people want to criticize a movie. Some misclassified neural words (e.g. *today*) may simply come from the bias of the dataset. And top 20 terms also demonstrate that mixed-gram behave better than uni-gram intuitively.

Furthermore, sentiment weights can help to draw *emotional word clouds* of positives reviews and negative reviews. The color of term shows its sentiment; the deeper the color is, the stronger sentiment the term has. We use cold colors (blue, cyan) to represent negativity and warm colors (red, yellow and orange) to

Table 2. Classification results under sparse constraints using logistic regressions with different N-gram language models.

N-gram	No constraints	ℓ_1 Norm	ℓ_2 Norm
Uni-gram	11.22 %	11.14 %	10.94 %
Bi-gram	11.03 %	11.00 %	10.84 %
Mixed-gram	9.77 %	9.61 %	9.58 %

Table 3. Performance comparison of LR+mixed-gram-ℓ_2 to other approaches on the IMDB dataset.

Sentiment classification model	Error	Sentiment classification model	Error
BoW(bnc) [14]	12.20 %	LDA [14]	32.58 %
Full+Unlabeled+BoW [14]	11.11 %	WRRBM [15]	12.58 %
WRRBM+BoW(bnc) [15]	10.77 %	MNB-uni [3]	16.45 %
MNB-bi [3]	13.41 %	SVM-uni [3]	13.05 %
SVM-bi [3]	10.84 %	NBSVM-uni [3]	11.71 %
NBSVM-bi [3]	**8.78 %**	PV+Unlabled [12]	**7.42 %**
LR-mixed-gram-ℓ_2 (Our model)	**9.58 %**		

represent positivity. The fontsize of a term is proportional to its TF-IDF value. Two sample emotional word clouds of positive and negative reviews in IMDB are shown in Fig. 2.

Table 2 shows the classification results under sparse constraints of ℓ_1 and ℓ_2 norms, respectively. Classification results of ℓ_2 norm is generally better than ℓ_1 and classical logistic regress without sparse constraints. Many researchers have reported their results on the IMDB dataset. In particular, one of the most significant improvement recently was the work of [3] in which they found that bi-gram feature works the best and yields a considerable improvement of 2 % in error rate. Another important contribution is [15] in which they combine a Restricted Boltzmann Machines model with BoW. The best result so far was reported by [12] in which deep learning was used and it involves a big computing resources. The method we proposed (LR+mixed-gram+ℓ_2) is the simplest one and with least computational time (Table 3).

We also conduct experiments on the Product Review dataset. In our experiments, like what we have done to the IMDB dataset, we do not remove any stopwords or apply any stemming in preprocessing for fair comparison to approaches proposed in [2]. We don't handle the problem of orthographic mistakes, abbreviations, idiomatic expressions or ironic sentences either. From Table 4, we can find that our model also has comparable performance to the baseline approaches. Though it performs slightly worse than SVM in classifying Books and DVDs, it performs well in Electronics and Kitchen.

Table 4. Comparisons to baseline approaches on the Product Review dataset.

Category	ANN	SVM	LR-mixed-gram
Books	18.3 %	17.2 %	19.8 %
DVD	18.4 %	16.3 %	19.9 %
Electronics	16.3 %	15.1 %	15.6 %
Kitchen	14.8 %	13.6 %	13.8 %

4 Conclusions

In this paper, we propose a model using Logistic Regression with Gradient Descent algorithm for extracting sentiment terms and learning sentiment weights. We assume the sentiment of a sentence or documents is a function of sentiment weights of consisting terms. The extracted sentiment terms can be drawn as emotional clouds. In sentiment classification, we have tested our model based on different N-gram models and find that the mixed-gram model outperforms both uni-gram and bi-gram models. Extensive experimental results show our proposed method can extract precise sentiment terms and achieve a high level accuracy in classification on given benchmark datasets.

References

1. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of EMNLP, pp. 79–86 (2002)
2. Fattah, M.A.: New term weighting schemes with combination of multiple classifiers for sentiment analysis. Neurocomputing **167**, 434–442 (2015)
3. Wang, S., Manning, C.D.: Baselines and bigrams: simple, good sentiment and text classification. In: Proceedings of ACL, pp. 90–94 (2012)
4. Kaji, N., Kitsuregawa, M.: Building lexicon for sentiment analysis from massive collection of HTML documents. In: Proceedings of EMNLP, pp. 1075–1083 (2007)
5. Wiebe, J.M.: Learning subjective adjective from corpora. In: Proceedings of AAAI, pp. 735–740 (2000)
6. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. In: Proceedings of ACL, pp. 9–27 (2011)
7. Jin, W., Ho, H.H.: A novel lexicalized hmm-based learning framework for web opinion mining. In: Proceedings of ICML, pp. 465–472 (2009)
8. Li, F., Huang, M., Zhu, X.: Sentiment analysis with global topics and local dependency. In: Proceedings of AAAI, pp. 1371–1376 (2010)
9. Park, S., Lee, W., Moon, I.C.: Efficient extraction of domain specific sentiment lexicon with active learning. Pattern Recogn. Lett. **56**, 38–44 (2015)
10. Potts, C., Schwarz, F.: Affective 'this'. Linguist. Issues Lang. Technol. **3**(1), 1–30 (2010)
11. Pappas, N., Popescu-Belis, A.: Efficient extraction of domain specific sentiment lexicon with active learning. In: Proceedings of EMNLP, pp. 455–466 (2014)
12. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of ACL (2014). arXiv:1405.4053

13. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of EMNLP, pp. 1746–1751 (2014)
14. Maas, A.L., Daly, R.E., Pham, P.T.: Learning word vectors for sentiment analysis. In: Proceedings of ACL, pp. 142–150 (2011)
15. Dahl, G.E., Adams, R.P., Larochelle, H.: Training restricted boltzmann machines on word observations (2012). arXiv:1202.5695