Sentiment Classification for Unlabeled Dataset using Doc2Vec with JST

Sangheon Lee
Dept. of Information and Industrial Engineering, Yonsei University
50 Yonsei-ro, Seodaemun-gu
Seoul, Korea
sangheon@yonsei.ac.kr

Xiangdan Jin
Dept. of Information and Industrial Engineering, Yonsei University
50 Yonsei-ro, Seodaemun-gu
Seoul, Korea
hyangdan0@yonsei.ac.kr

Wooju Kim
Dept. of Information and Industrial Engineering, Yonsei University
50 Yonsei-ro, Seodaemun-gu
Seoul, Korea code
wkim@yonsei.ac.kr

ABSTRACT

Supervised learning require sentiment labeled corpus for training. But it is hard to apply automatic sentiment classification system to new domain because labeled dataset construction costs a lot of time. Meanwhile, researches using Doc2vec based document representation beat out other sentiment classification researches. However, these document representation methods only represent documents' context or sentiment. In this paper, we proposed supervised learning scheme for unlabeled corpus and also proposed document representation method which can simultaneously represent documents' context and sentiment.

CCS Concepts

• **Information systems→Sentiment analysis.**

Keywords

Sentiment analysis; Doc2Vec; JST; machine learning; natural language processing

1. INTRODUCTION

At present, the method of sentiment analysis is mainly divided into two categories. One is based on the emotion dictionary, and the other is based on machine learning which is used for large scale corpus. Dictionary based approach needs a good labeled emotion dictionary. There are many English dictionaries, but Korean dictionaries are not completed to use directly. Machine learning based approach requires a lot of manual tagging of the corpus as a training set, the classifier is constructed to decide the polarity of sentiment by extracting the feature of the text. Sentiment classification using unsupervised learning algorithm can classify the corpus which don't have sentiment label. When using unsupervised learning, the corpus would clustered base on corpus itself, so the accuracy of using unsupervised learning could lower than using supervised learning. Sentiment classification using supervised learning can classify the corpus which have sentiment label. When using supervised learning, the prediction model is learned about the corpus with true label, so it can predict the true answer for corpus. The algorithm using supervised learning should be more correct than unsupervised learning or semi-supervised learning. Semi-supervised learning approaches use unlabeled corpus to improve classification accuracy by labeling unlabeled corpus and using the label information. Semi-supervised learning approaches are meaningful because they use only small amount of labeled corpus and can get comparable classification accuracy to classification accuracy of supervised approaches[4]. But both supervised learning and semi-supervised learning require sentiment labeled corpus for training. But it is hard to apply automatic sentiment classification system to new domain because labeled dataset construction costs a lot of time. On the other hand, researches about document representations based on deep learning technology has been conducted. Research about sentiment classification based on Doc2vec shows high performance than Bag-of-Words [5]. L. Park [1] uses Doc2vec model to improve the performance of sentiment analysis by learning about polarity (i.e., positive, negative) vector. But we requires both context based vector representation and sentiment polarity based vector representation.

In this paper, we proposed concatenate vector to represent document and use it as document's feature. Also, we proposed semi-supervised learning approach for small amount of labeled dataset and large amount of unlabeled dataset. We generate initial labeled dataset using unsupervised learning approach, extend labeled training dataset using semi-supervised learning approach and concatenate vector. Finally, we choose the best classifier and doc2vec model and classify test documents' sentiment.

2. RELATED WORK

In this section, we discuss document representation methods and unsupervised/semi-supervised learning approach.

2.1 Document Representation Methods

Existing researches[1,2] represent document as bag-of-words model; that is, represents document's vector using unigram, bigram features. Because bag-of-words based document representation methods represents document vector using occurrence and frequency of features, they couldn't represent any words' context and documents' context. In order to deal with these two problem, [5] proposed not keyword based feature, but document based feature. When we use document-based feature; document vector using Paragraph Vector, we don't have to put up with any information loss by using all components of documents because all words in document are fully used for learning

[1] https://www.lucypark.kr/slides/2015-pyconkr/#1

ICEC '16, August 17 - 19, 2016, Suwon, Republic of Korea.

ACM 978-1-4503-4222-3/16/08…$15.00
DOI: http://dx.doi.org/10.1145/2971603.2971631

document vector. We can also comprehend document's complex sentiment in a comprehensive angle since Para2Vec represents its feature to multi-dimensional vector (not scalar value).

2.1.1 Doc2Vec

2.1.1.1 Document Vector

Doc2vec generate document vectors and word vectors which appears in training dataset. Every document in training dataset will be represented as word set and tag. Word set is all tokens in each document and tag is the identifier for each document. Document vector matrix **D** is a vector matrix for all tags in training data, each document vector is represented by a column of matrix **D**. Word vector matrix **W** is a vector matrix for all tokens in training data, each word vector is represented a column of matrix **W**.

Document vector is generated by the context in a document and the whole document. We train doc2vec model with the training data, which is tagged by document number. Then the model will generate document vectors for each document. The size of matrix D is the same as the numbers of documents. And then use the inference function in doc2vec to predict the vectors for the test dataset. The inference stage is to get a document vector for a new document. This is also obtained by gradient descent. In this step, the parameters for the rest of model, the word vectors **W** and the softmax weights are fixed.[5]

2.1.1.2 Polarity Vector

These document vectors can represent the contextual information but can't represent the information about sentiment polarity. For example, for two sentences, "It sounds good" and "It sounds bad", in these sentences, "good" and "bad" is the same in context location, but their sentiment polarity is completely opposite. But the model may see them similar. In order to complete this situation, we make another vector for document called polarity vector. Polarity vector is presented about document polarity. In this method, there are only two types of tags in training dataset, positive and negative. Every document is tagged by positive or negative then after training the model will has two document vectors, a positive document vector and a negative document vector. Then we infer all the training and test dataset with this doc2vec model. The inference vector for each document is polarity vector.

2.2 Unsupervised Learning Approach

Unsupervised learning approaches classify unlabeled instances to several classes. JST(Joint Sentiment/Topic Model) [6] is prominent in unsupervised sentiment classification area. JST is a topic model which adds sentiment layer into LDA [3]. [6] uses sentiment-labels' probability distribution for each document generated by JST model to classify each document's sentiment polarity. However, JST has two problems. First, human has to determine sentiment-labels' sentiment polarity by looking up word distribution given sentiment-label and topic. Second, JST can't classify instance accurately if difference between instance's positive-sentiment probability and instance's negative probability is small. In this work, we proposed sentiment labels' sentiment score to automatically determine sentiment-labels' sentiment polarity. Also, we determines confidently classified instances by using threshold and add only confidently classified instances to labeled training set to improve classification accuracy.

2.3 Semi-supervised Learning Approach

Supervised learning methods are highly dependent to large amount of labeled dataset. If we don't have any labeled dataset, we couldn't apply supervised learning methods. Researchers studied semi-supervised learning approaches by using unlabeled instances and found that these approaches perform better. [4] is one of the semi-supervised learning approach. It trains a classifier with a small amount of labeled data and iteratively train it by augmenting n most confidently classified unlabeled test samples to labeled training dataset.

3. PROPOSED METHODOLOGY

3.1 Supervised Learning Approach

3.1.1 Preprocessing

Before Training Doc2vec we first do pre-processing with the data. There are two steps in pre-processing, tokenization step and POS tagging step. First we dividing the text data into tokens. Then use Twitter class in Python to annotate every token with POS tags. In order to distinguish between homonyms, we use "token/POS tag" format to training doc2vec model. We don't remove any tokens like punctuation marks, emoticons and stops words. We treat all tokens in the context is meaningful and use them to training doc2vec model.

3.1.2 Concatenate Vector

We concatenate document vector and polarity vector as document feature vector. This vector can represent context information and sentiment polarity. We use this vector to train machine learning algorithm and classify document polarity. In our experiments we use SVM (support vector machine) to classify the document polarity.

3.2 Semi-supervised Learning Approach

If we don't have any labeled dataset, we couldn't apply supervised learning methods. Though self-training requires smaller amount of initial labeled dataset than supervised methods, self-training performs poorly if amount of initial labeled dataset is very small. Therefore we proposed semi-supervised learning approach for small amount of labeled dataset and large amount of unlabeled dataset. Procedure of semi-supervised learning approach is shown as Figure 1. First, generate initial labeled training dataset. Second, iteratively label training instances with sentiment label and stop repetition in stopping condition. Third, select best model among each models generated in each iteration with classification performance. Finally, classify test dataset using selected model.

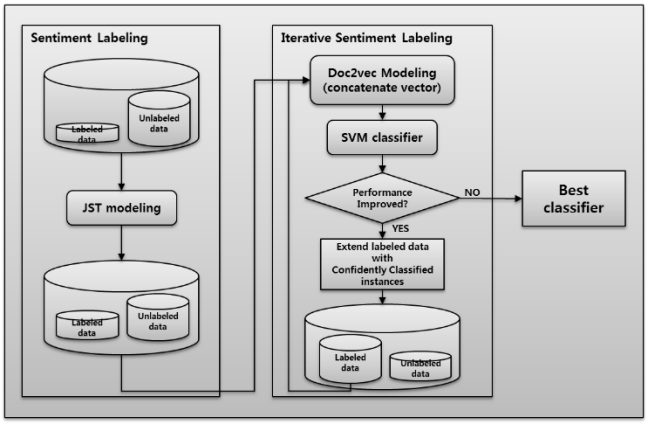

Figure 1. Procedure of Semi-supervised Learning Approach

3.2.1 Sentiment Labeling using JST

In order to apply self-training for unlabeled dataset, we constructed initial labeled training dataset by sentiment labeling using JST.

3.2.1.1 JST modeling

We pre-processed training documents to leave only noun, verb, adjective, adverb tokens. Using the pre-processed training documents, we learned JST model with 2 sentiment labels and 10 topics.

3.2.1.2 Determine Sentiment Labels' Sentiment Polarity

JST generates P(w|l,z), words' probability distribution given sentiment label l and topic z. [6] uses top N words in words' probability distribution given sentiment label l and topic z to determine sentiment polarity of sentiment label l. To determine sentiment polarity of sentiment label automatically, we use small amount of labeled development dataset. The number of determining sentiment polarity of sentiment labels is only 2, assign positive sentiment label to label1 or label2, vice versa. We determine sentiment polarity of sentiment labels as follows. First, determine development dataset's sentiment polarity with 2 cases. Then, select the case which has better classification performance of development dataset.

3.2.1.3 Determine confidently classified instances

[6] classify sentiment polarity of each document by comparing sentiment labels' probability distribution given document, P(l|d). That is, [6] classify a document's sentiment polarity as positive if positive probability given the document, P(positive|d) is higher than negative probability given the document, P(negative|d) and vice versa. But if difference between P(positive|d) and P(negative|d) is very small, we can't accurately classify sentiment polarity of the document.

In this paper, we use threshold τ to determine confidently classified instances. We define document d as confidently classified instance if the instance meets the equation (1).

$$|P(positive|d) - P(negative|d)| > \tau \quad (1)$$

We determine the threshold τ with the maximum τ among the cases which have the best classification performance for selected development instances by threshold τ.

Confidently classified instances are added to labeled training dataset, L and not-confidently classified instances are added to unlabeled training dataset, U.

3.2.2 Iterative Sentiment Labeling using Self-Training and Concatenate Vector

Iterative sentiment labeling is as follows. First, we use doc2vec to get document vector and use this vector to classify the polarity of document. We will use doc2vec model to present document as document vector and polarity vector in feature space for sentiment classification. Next, select most confidently classified instances(10%) to label them, then remove them from labeled training dataset and put it into unlabeled training dataset. Repeat iterative sentiment labeling until all unlabeled data are labeled or moving average-based estimated classification performance of development dataset decreases.

3.2.2.1 Training

We generate document vector and polarity vector for labeled training documents in each iteration. Then we concatenate document vector and polarity vector as document feature vector. This vector can represent context information and sentiment polarity. We use this vector to train machine learning algorithm and classify document polarity. In our experiments we use SVM (support vector machine) to classify the document polarity.

3.2.2.2 Labeling

By predicting sentiment label for unlabeled training instances using SVM model in each iteration, we can get positive probability and negative probability for each instances. We select 10% most confidently classified instances and label these instances with sentiment label in each iteration. The labeled instances in this iteration will be added to the labeled training dataset and the unconfident data in this iteration will be in the unlabeled training dataset again.

3.2.2.3 Stopping Criteria

Self-training repeats unless there is no unlabeled training instance. In this paper, we apply heuristic stopping criteria to shorten the running time. Stopping criteria is as follows. If moving-average based estimated classification performance (for labeled development dataset) decreases, stop iterative sentiment labeling.

3.2.2.4 Model Selection

When iterative sentiment labeling is stopped, we select doc2vec models and SVM model in each iteration by comparing classification performance for labeled development dataset.

3.2.3 Sentiment Classification

In this step, we use selected doc2vec model to generate concatenate vectors of test instances. Then predict sentiment of test instances using selected SVM model.

4. EXPERIMENT

We did experiments for supervised learning approach and semi-supervised learning approach for 2 types of dataset.

4.1 Supervised Learning Approach

4.1.1 Data

We use Naver movie review data to evaluate our supervised learning approach(concatenate vector). The data is constructed by the method from [7]. Every review contains at least one token, each review is rated by score from 1 to 10, we use the reviews ratings from 1 to 4 as negative samples and use the reviews rating from 9 to 10 as positive samples. Reviews which ratings from 5 to 8, we treat them as neutral and didn't use at this experiment. We use 150,000 reviews for training with 74,827 positive reviews and 75,173 negative reviews. Use 50,000 reviews with 25,173 positive reviews and 24,827 negative reviews. All data have three columns, id, content, and label. We use this label to evaluate the test data.

4.1.2 Experiment

In the proposed method, feature vector is represented by two kinds of vectors, document vector and polarity vector. Document vectors are learned from doc2vec using 150,000 training data. In this doc2vec model, we set parameters as follows. Algorithm we use PV-DM algorithm, window size set as 8, it means we consider a paragraph vector and 7 word vectors to predict a word in the context. Vector size set as 300. For 50,000 test data, we use inference to predict the document vector. Then the 300-dimensional vector is "document vector" in our model.

To get polarity vector, we use another doc2vec model. In this model, when we training doc2vec model, set the parameters the same as getting document vectors. Then we called the 300-dimensional vector "polarity vector". Finally we concentrate document vector and polarity vector, a new 600- dimensional

vector to classify the document sentiment. In our experiment, we use SVM classifier in sentiment classification step.

4.1.3 Experiment Result

We compare out result with the most popular algorithm in sentiment classification called TF-IDF. In TF-IDF, feature is the word in all data and the TF-IDF score for feature value. There are 3,677 features in this feature space.

And we also do the experiment for only document vector and polarity vector, which feature size is 300 for each. The last experiment is for the concentrate vector which has the size of feature is 600. The performance measure in this experiments are as follows:

$$Accuracy = \frac{(tp+tn)}{(tp+fp+fn+tn)}$$

$$Recall(pos) = \frac{(tp)}{(tp+fp)}$$

$$Recall(neg) = \frac{(tn)}{(fn+tn)}$$

$$Precision(pos) = \frac{(tp)}{(tp+fn)}$$

$$Precision(neg) = \frac{(tn)}{(fp+tn)}$$

$$F1-score = \frac{2*Precision(pos)*Recall(pos)}{(Precision(pos)+Recall(pos))}$$

Table 1. Confusion Matrix

Total population		True condition	
		Condition positive	Condition negative
Predicted condition	Predicted condition positive	True positive	False positive
	Predicted condition negative	False negative	True negative

Table 2. Experiment Result (Concatenate Vector)

Experiment	TF-IDF	Document Vector	Polarity Vector	Concatenate Vector
Accuracy	0.713	0.764	0.759	0.793
Recall(pos)	0.617	0.763	0.758	0.791
Recall(neg)	0.809	0.766	0.761	0.793
Precision(pos)	0.767	0.768	0.762	0.795
Precision(neg)	0.676	0.761	0.756	0.790
F1-score	0.683	0.765	0.760	0.793

Table 2 shows the result of experiment. Using doc2vec shows high performance than TF-IDF, and the concentrate vector shows high accuracy than that experiments only use document vector or polarity vector. It was applicable for all performance measure.

4.2 Semi-supervised Learning Approach

4.2.1 Data

We use several categories news article data to evaluate. We choose 200 news articles randomly, four people read news articles and labeled them manually, then use majority voting to label the news article. We use 50 of them as development data and 150 of them as test data.

4.2.2 Experiment

First, we need to select the first training data set which will use in iterative sentiment labeling. It can be selected from JST modeling. We use 10 topics and 2 sentiment label to training JST model. Based on the 50 development data, we can select the first 3,753 labeled data and 6,097 unlabeled data to start self-training. Using the classifier learned by 3,753 labeled data, we classify the 6,097 unlabeled data. Then select 1,000 most confidently classified instances, remove these data from unlabeled data set and extend the labeled data set. Each iteration, record the classifier's performance and moving-average about 50 development data. If moving-average based estimated classification performance decreases, stop iterative sentiment labeling. But in this experiment, the moving-average is increase until there are no unlabeled data in the whole training data. So we stop the experiment when where are no more unlabeled data. Then we select the best performance classifier as best classifier and use it to classify the final test data set.

Table 3. Accuracy of Development Dataset in Each Iteration

# of Iteration	0	1	2	3	**4**	5	**6**	7
# of labeled instances	3,753	4,753	5,753	6,753	**7,753**	8,753	**9,753**	9,850
Accuracy (development dataset)	0.5600	0.5000	0.6200	0.5600	**0.6800**	0.6200	**0.6800**	0.6600
Estimated accuracy		0.5300	0.5600	0.5600	0.5840	0.5900	0.6029	**0.6100**

We will compare it with the following three kinds of experiments.

The first is PMI score, we use PMI score proposed in [8]. Second one is labeling data using only JST model. We labeled news articles base on the probability resulted in JST only. The Third one use both JST and the method we proposed as supervised learning, but it doesn't repeat the sentiment labeling method, it labels the whole training data only one time and use JST only. Then use the method we proposed in supervised learning to classify 150 test data.

4.2.3 Experiment Result

First, we can see that in unsupervised learning method, JST is more accurate than PMI score, so we use JST to select the first training set. And the method we proposed in this paper shows the best performance in all measures.

Table 4. Experiment Result (Proposed Method)

Experiment	Phrase PMI	JST	JST (labeling) +SVM	Proposed Method
Accuracy	0.5068	0.5600	0.5467	0.6533
Recall(pos)	0.5893	0.6140	0.5614	0.6754
Recall(neg)	0.2500	0.3889	0.5000	0.5833
Precision(pos)	0.7097	0.7609	0.7805	0.8370
Precision(neg)	0.1636	0.2414	0.2647	0.3621
F1-score	0.6439	0.6796	0.6531	0.7476

5. CONCLUSION

In this paper, we proposed concatenate vector to represent document and use it as document's feature. Also, we proposed semi-supervised learning approach for small amount of labeled dataset and large amount of unlabeled dataset. We generate initial labeled dataset using unsupervised learning approach, extend labeled training dataset using semi-supervised learning approach and concatenate vector. Finally, we choose the best classifier and doc2vec model and classify test documents' sentiment.

We experiment with proposed concatenate vector and semi-supervised learning approach to evaluate them. Experiment results shows that concatenate vector performs better than document vector and polarity vector in Korean Naver movie review dataset and semi-supervised learning approach performs better than JST and PMI. In future work, we will evaluate proposed semi-supervised learning approach using Korean Naver movie review dataset to analysis influence of sentiment labeling.

6. ACKNOWLEDGMENTS

This work (Grants No.C0258734) was supported by Business for Academic-industrial Cooperative establishments funded Korea Small and Medium Business Administration in 2015.

This work is financially supported by Korea Ministry of Land, Infrastructure and Transport(MOLIT) as 「U-City Master and Doctor Course Grant Program」.

7. REFERENCES

[1] B. Pang and L. Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, 2004, Association for Computational Linguistics, 271-278.

[2] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, Morristown, NJ, USA, 2002, Association for Computational Linguistics, 79-86.

[3] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *The Journal of machine Learning research*, *3(Jan)*, 2003, 993-1022.

[4] Gao, W., Li, S., Xue, Y., Wang, M., and Zhou, G. Semi-supervised Sentiment Classification with Self-training on Feature Subspaces. In *Chinese Lexical Semantics*, 2014, Springer International Publishing, 231-239.

[5] Le, Q. V., and Mikolov, T. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, 2014, 1188-1196.

[6] Lin, C. and He, Y. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, ACM, 375-384.

[7] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, 2011, Association for Computational Linguistics, 142-150.

[8] TURNEY, Peter D. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting on association for computational linguistics, 2002, Association for Computational Linguistics, 417-424.