Sentiment classification of SNS comments with CNN

Seonmi Lee, 20205196

I. INTRODUCTION

Recently, people use social network services(SNS) like Tweeter or Instagram in their daily life. People started to build public or semi-pubic in SNS where they can share their emotion and opinions freely than off-line circumstances.[1] As SNS becomes a field to exchange one's emotion, SNS can be used for a tool analyze people's sentiments. There are many ways to express one's emotion online. Among them, text has been researched as one of the best methods to analyze sentiments.

There are many studies about text sentiment analysis in the field of deep learning using a lot of data from popular portal site which forms large community like Twitter. Among them, binary sentiment classification which divide text into positive or negative group are performed by many kinds of models for the sake of marketing. However, it is necessary to classify sentiments in more categories so that it can be used in many ways like tracking user's moods continuously or regulating indiscreet expressions online.[2] Also, there are a lot of sources to perform supervised learning for sentiment classification in English, but not many in Korean.[3]

Therefore, the purpose of this study is to conduct a model to classify multi sentiments from Korean SNS comments using Convolutional Neural Network(CNN). This study uses convolutional neural network(CNN) for natrual language sentiment classification. To compare final model, machine learning models like Naïve Bayes and support vector machines were used.

II. PROBLEM DESCRIPTION

A. Data preparation

Data set was needed to train the model. It has crawled from 'AI hub', a universal platform which provide AI data for development of AI technology. [4] This web site had provided short sentences sentence collected from SNS and online comments. Each sentence has its own sentiment labels. There are 7 kinds of sentiments; 'Scared', 'Frightened', 'Angry', 'Sad', 'Neutral', 'Happy', 'Disgusted'. Each sentiment occupies similar proportion in the data set. (Figure 1) All the sentences had been labeled one of the 7 sentiments. The number of sentences was 38,594 and words in each sentence was 5.3(SD = 3.1). The data had divided into 3 groups for training, validation, and testing in 6:2:2 proportions.

B. Preprocessing

Because the data set has rough sentences which have a lot of useless words like typos. Therefore, the sentences are needed to be refined before training. There were 3 steps to refine text data before embedding the sentences.

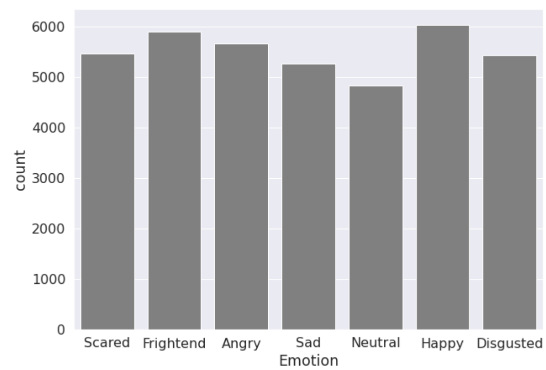

Fig. 1. Graph shows emotion distribution in the data

- Canonicalization: Sentences have many kinds of things like emoticons and punctuation symbols. In this step, all the things except the words were removed. Also, unpaired Jamo was removed also. Therefore, there only left well constructed word in each sentence.
- Tokenization: English does not have lots of variation in word expression, but Korean has a lot of variation of words caused by Josa. Therefore, words in sentences were tokenized into morphology. In htis model, we used koNLPy which is morphological analyzer for Korean. Among them, we used Okt which is well used for informal expressions expecially people use in SNS. [5]
- Regulation of sentence length: Too short or too long sentence is not appropriate for modeling because there can be too much zeros which can lead confusion to the model. Therefore, the length of the sentences were regulated from over 2 words and under 10 words. The limitation 10 was three forth point from the sentence length distribution.

C. Evaluation

The model for this study shoult be evaluated compared with other supervised learning models. There are many machine learning methods for supervised learning like Naive Bayes classifier, K-Nearest Neighbors, Neural Network, Support Vector Machine(SVM), and Maximum Entropy. In the previous study from Pang, SVM was appeared to have best performance for sentiment analysis among them. [9] However, after that, there were some researches that Naive Bayes classifier shew better performance than SVM. [10][11] Therefore, Naive Bayes classifier and SVM were used to compare the model performance in this study. Training the

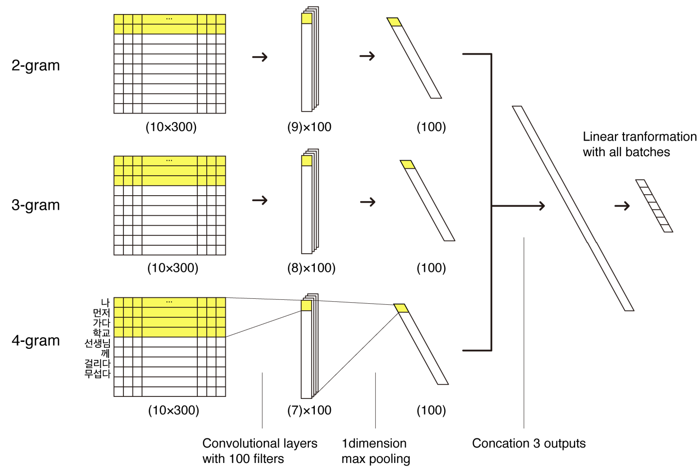

Fig. 2. Scheme for the model this study used

machine learning model, there was no validation set so that only training and testing sets were used. Accuracy and f1 score was used for general metrics.

Also, there are 7 sentiments to be classified not binary classification, so confusion matrix was drawn to see whether the model could classify sentiments on the same level. From confusion matrix, recall, precision and f1-score of all sentiments were calculated.

III. METHOD

A. Embedding

To train the model, the words should be vectorized first. The words was embedded using word dictionary. This word dictionary was constructed by word vectors which is previously trained on Wikipedia by fasttext. The words were trained by skip-gram method which performs embedding in a unit of charactor first, then make word vectors by integration of charactor vectors.[6] From this reason, it performs better than embedding method like n-gram fasttext, or word2vec. Therefore, based ob this pre-trained word vectors, we built our word dictionary using training set. In our model, we used this word dictionary for embedding the words.

B. Model

In this study, we used Convolutional Neural Networks(CNN) to train the embedded sentences and classify to 7 sentiments. CNN had been verified to be effective to natural language processing(NLP) and could draw excellent result in semantic parsing [7] which means enabling to get meaning from natural language. However, language is sequential data so that Recurrent Neural Network(RNN) is also frequently used in NLP task. However, RNN is useful for long sequential data but this study focuses on short sentence analysis. Moreover, it is effective to use CNN models when we want to execute a keyphrase recognition task like sentiment detection.[8] Therefore, CNN was used to design the model for this study.

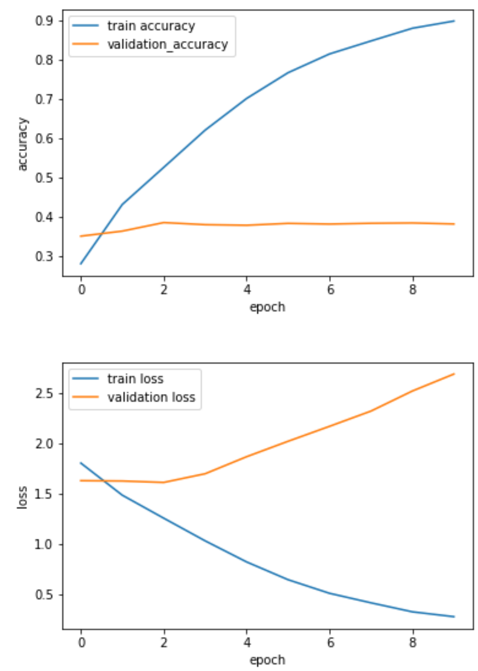

Fig. 3. Changes of training and validation accuracy(upper) and loss(lower)

The Figure 2 shows the simple scheme for the model. In our model, 3 convolutions are done together with 3 different kernels. The kernel sizes are (2,300), (3,300), (4,300) each. This shows that we will consider 2, 3, 4, words in a unit of statistical learning which means 2, 3, 4-gram. The second dimension of the kernel is same with word embedding vector length, so it is possible to obtain a number as a result of 2, 3, or 4-gram convolution. ReLU was used as activation function in this model. Then, 1 dimension max pooling is performed to extract maximum unit which affects most in the sentence from the n-gram results. Then, we concate 3 n-gram results together and perform linear transformation with 7 outcomes because there are 7 sentiments to be classified.

Before doing linear transformation, we do regularization by dropping less than 0.4 values to remove noise. It means that if there are too many detailed information in the final steps, the model can be confused by weak signals not needed to be considered. These can lead to overfitting, so we put dropout in this model for regularization method.

IV. EXPERIMENT RESULT

A. Training result

The pre-processed data set was trained through our model. The number of epochs was 50 and batch size for a mini batch was 20. The Figure 3 shows the train and validation loss and accuracy as epoch number increase. It shows that the validation accuracy drops after epoch 2 which is very early and loss difference between training and validation tends to be larger continuously. The large gap between training and validation loss means overfitting. Therefore, we used a model trained before third epoch for test set. To get more specific performance of the model, confusion matrix was drawn. From the confusion matrix, recall and precision for each sentiment was calculated. Table 1 shows the result.

TABLE I
RECALL AND PRECISION OF EACH SENTIMENT

Sentiment	Recall	Precision	f1-score
Angry	0.39	0.35	0.37
Happy	0.63	0.57	0.60
Surprised	0.22	0.43	0.29
Disgusted	0.30	0.28	0.29
Scared	0.35	0.39	0.37
Sad	0.50	0.36	0.42
Neutral	0.20	0.21	0.20

From the above table, it shows that this model can predict 'Happy' mostly (f1-score = 0.60), and 'Neutral' worst (f1-score = 0.20). It shows that 'Neutral' is hard to be predicted with CNN because there are less representative word to express 'Neutral'.

B. Model comparison

There are 2 machine learning models; Naive Bayes classifier and SVM to be compared with the new model. Accuracy and f1-score was calculated to compare those 3 models. Table 2 shows the result. NB means Naive Bayes classifier and SVM means Support Vector Machine. According to this result, it is shown that Naive Bayes and SVM has similar accuracy and f1-score but CNN has slightly lower than them. In fact, deep learning model performs significantly better than machine learning method when the data is large enough. However, the data which is used for the study was only about 40,000 sentences with 7 categories. Therefore, the data set was not large enough considering the number of classes.

TABLE II
ACCURACY AND F1-SCORE FOR 3 METHOD; CNN, NAIVE BAYES, SVM

	CNN	NB	SVM
Accuracy	0.37	0.442	0.431
F1-score	0.37	0.435	0.434

TABLE III
F1-SCORE OF EACH SENTIMENT IN 3 MODELS

Sentiment	NB	SVM	CNN
Angry	0.45	0.42	0.37
Happy	0.68	0.67	0.60
Surprised	0.39	0.40	0.29
Disgusted	0.31	0.33	0.29
Scared	0.46	0.46	0.37
Sad	0.49	0.47	0.42
Neutral	0.24	0.28	0.20

Table 3 shows each sentiment's f1-score for all 3 models. Naive Bayes model could classify 'Happy' best(f1-score = 0.69) and 'Sad','Scared' and 'Angry' next in about 0.48 f1-score. The result of SVM shew similar patterns with Naive Bayes. This pattern is very similar with CNN model. Also, all the models have very low f1-score for 'Neutral', compared with other sentiments.

V. CONCLUSIONS

The purpose of this study was to classify multi sentiments from Korean SNS comments using Convolutional Neural Network(CNN). In this study, we used Korean morphological analyzer koNLPy and pre-trained word vectors which are trained using fasttext based on wiki data. In our model, there were 3 convolutions had progressed, with different size of the kernels. After training, accuracy and f1- score turned out to be 0.37 which means low performance. However, it is needed to check confusion matrix also to check whether the model classified sentiments in balance. From the confusion matrix, it appears that 'Neutral' which is hard to distinguish representative expressions, could not be classified. However, 'Happy'(f1-score = 0.60) showed better performance. This shows drawbacks of CNN which picks one important token without considering sequence. Therefore, it is required to make Recurrent Neural Network model which considers sequence also and compare its performance together under the multi-sentiment circumstances.

This model had compared with other machine learning models, Naive Bayes classifier and SVM. The performance of CNN model was not effective compared with other models with our data set. However, it is hard to say CNN model is not better than machine learning models because the data set used in this study was too small. From this reason, the future work after gathering more data should be done.

In conclusion, the CNN model performed not good. However, CNN model tends to be useful to distinguish extreme sentiments like 'Happy', and 'Sad'. Therefore, if this model is developed more with more data set, then it can be helpful for many things like like regulation of SNS comments that needs extreme sentiments detection.

REFERENCES

[1] Boyd, D. M., Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. Journal of computer-mediated communication, 13(1), 210-230.

[2] Döveling, K. (2015). Emotion regulation in bereavement: Searching for and finding emotional support in social network sites. New Review of Hypermedia and Multimedia, 21(1-2), 106-122.

[3] Yuyoung Kim, Min Song. (2016). A Study on Analyzing Sentiments on Movie Reviews by Multi-Level Sentiment Classifier. Intelligent information research, 22(3), 71-89.

[4] AI Hub intro: AI Hub. (n.d.). Retrieved June 19, 2020, from http://www.aihub.or.kr/intro/vision

[5] Wiki word vectors · fastText. (n.d.). Retrieved June 19, 2020, from https://fasttext.cc/docs/en/pretrained-vectors.html

[6] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135-146.

[7] Yih, W. T., He, X., Meek, C. (2014, June). Semantic parsing for single-relation question answering. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 643-648).

[8] Yin, W., Kann, K., Yu, M., Schütze, H. (2017). Comparative study of cnn and rnn for natural language processing. arXiv preprint arXiv:1702.01923.

[9] Pang, B., Lee, L., Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

[10] Ferguson, P., O'Hare, N., Davy, M., Bermingham, A., Sheridan, P., Gurrin, C., Smeaton, A. F. (2009). Exploring the use of paragraph-level annotations for sentiment analysis of financial blogs.

[11] Pak, A., Paroubek, P. (2010, May). Twitter as a corpus for sentiment analysis and opinion mining. In LREc (Vol. 10, No. 2010, pp. 1320-1326).