FVEC feature and Machine Learning Approach for Indonesian Opinion Mining on YouTube Comments

Aina Musdholifah
Master Program of Computer Science
Department of Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
aina_m@ugm.ac.id

Ekki Rinaldi
Master Program of Computer Science
Department of Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
ekki.rinaldi@mail.ugm.ac.id

***Abstract*— Mining opinions from Indonesian comments from YouTube videos are required to extract interesting patterns and valuable information from consumer feedback. Opinions can consist of a combination of sentiments and topics from comments. The features considered in the mining of opinion become one of the important keys to getting a quality opinion. This paper proposes to utilize FVEC and TF-IDF features to represent the comments. In addition, two popular machine learning approaches in the field of opinion mining, i.e., SVM and CNN, are explored separately to extract opinions in Indonesian comments of YouTube videos. The experimental results show that the use of FVEC features on SVM and CNN achieves a very significant effect on the quality of opinions obtained, in term of accuracy.**

Keywords*— Machine Learning, CNN, FVEC, Opinion Mining, SVM

I. Introduction

Extracting the sentiment from Indonesian YouTube comments is still a challenge. Slangs and various dialect is the main problem. Rinaldi and Musdholifah [1] initiated Indonesian YouTube opinion mining using self-labelled Indonesian comment. The experiment conducted in three type of label given: SENTIMENT, TYPE, and ALL. SENTIMENT label based on overall sentiment given regardless its toward video or product. TYPE label experiment observe the type of comment whether it given toward product or video. ALL label consist the combination of both SENTIMENT and TYPE. This research proves that the overall result of classification using STRUCT method approach is better.

A term called sentiment analysis, or the mathematical taxonomy of statements' negative or positive connotations, gives companies potent ways to analyze cumulative language data across all sorts of communications. Opinion mining and sentiment analysis in the era of big data have been used in categorize the opinion into different sentiment and evaluating the mood of the public in general [2]. Various techniques have been developed over the years in different datasets and applied to various experimental settings and business cases.

Machine learning framework is an integrated system of programs. These programs learn from existing data and capable of predicting new observations. Machine learning learns from data rather than explicitly programmed instructions. Machine learning itself divided into supervised learning which need to learn its predecessor data to make such classification and regression, and unsupervised learning which treat the data as a space to learn the inherent structure of the data without explicitly labelled. This study used machine learning based technique to extract customer's opinion, train and classify on selected key words, named CNN and SVM.

Convolutional Neural Network or known as CNN rise its popularity in image processing with proven capability in various image data like MNIST [4]. CNN uses "neighborhood" method to analyze a pixel value to its surrounding, then extract the important feature from convoluted image. Yoon [5] make a breakthrough by implementing CNN for sentence classification because he see the neighborhood method can be utilized to extract sentiment in the comments. This research proves that CNN can be used in sentence classification with decent performance.

The decent performance and uniqueness of how CNN works bring the curiosity on combining CNN for sentence classification from Yoon [5] with FVEC additional feature from Rinaldi and Musdholifah [1]. Thus this research focused on combining CNN and FVEC then compares the performance with FVEC-SVM.

II. Related Works

Severyn [6] introduced FVEC and STRUCT feature in examine opinion mining in YouTube videos in English. The classification method used is SVM using the SHTK kernel function (Shallow Syntactic Tree Kernel). This study uses data from two domains, tablet domain and automobiles domain. The approach taken in classifying is divided into two, namely FVEC-SVM approach using the bag-of-words method and STRUCT-SVM approach using chunking method. Full task category reach 60,3% accuracy with STRUCT approach.

Rinaldi and Musdholifah [1] undertake the research similar to Severyn [6] with the following differences: (1) The comments are in Indonesian, (2) Only use smartphone domain, (3) The kernel functions used include linear, polynomial degree 2, polynomial degree 3, and RBF, (4) In the FVEC approach, lexicon approach is excluded. In this research, the FVEC-SVM performed better than STRUCT-SVM with 62.76% using linear kernel function in full task experiment.

Yoon [5] made a breakthrough by implementing CNN for sentence classification. CNN was popular for image classification by analyzing value of a pixel and its neighborhood using sliding window. This method of finding correlation of a feature with its surrounding undertake the similar problem on sentence classification. Yoon's work put text into word embedding, add padding so each sentence have a same length, and then send it to single-layer convolutional neural network. The performance was respectable compared to other famous text classification methods. Simple CNN with one layer of convolution performs remarkably well instead of tuning on hyper-parameters.

978-1-5386-8402-3/18/$31.00 ©2018 IEEE

Socher [7] proposed new model called the Recursive Neural Tensor Network (RNTN) Recursive Neural Tensor Networks is be able to take input phrases in any length. The phrase transformed into word vectors and parse tree, then by using tensor-based function for higher nodes vector computation. The RNTN accurately captures the sentiment and negation of the sentence. This research also includes two types of analyses: several large quantitative evaluations on the test set, and focuses on two linguistic phenomena that are important in sentiment. The top performance for all models was achieved between 25 and 35 words vector sizes dimensions with batch sizes between 20 and 30.

Socher [8] invented method for paraphrase detection called Recursive Autoencoders (RAE). RAE works by unfolding objective and learn feature vectors for phrases in syntactic trees. These features were used to calculate the similarity of word and phrase-wise between two sentences. This research introduced a novel dynamic pooling layer which computes a fixed-sized representation from the variable-sized matrices as input to the classifier since the length of the sentences vary. This method outperforms other approaches on MSRP paraphrase corpus. The RAE captures syntactic and semantic information as shown qualitatively with nearest neighborhood embedding and quantitatively on a paraphrase detection task. This representation captures sufficient information to determine the relationship of paraphrase on the MSRP dataset with a high accuracy.

Kalchbrenner [9] introduced Dynamic Convolutional Neural Network (DCNN). Dynamic k-Max was used for a global pooling operation over linear sequences. The network is capable to explicitly capturing short and long sentence relations. The network is easily applicable to any language since it does not rely on parse tree. This research conducted small-scale binary and multi-class sentiment prediction, six-way question classification and Twitter sentiment prediction by distant supervision, total four experiments. The model achieves outstanding performance in the first three tasks and more than 25% error reduction in the last task due to the strongest baseline.

Hermann dan Blunsom [10] experiment with learning of vector space representations of sentential semantics and the transparent interface between syntax and semantics provided by Combinatory Categorial Grammar (CCG) to introduce Combinatory Categorial Autoencoders. This research learns high dimensional embedding for sentences and evaluate them in a range of tasks, proving that the incorporation of syntax allows a concise model to learn representations that are both effective and general. This experiment explored a number of models, each of which conditions the compositional operations on different aspects of the CCG derivation. This experiment indicates a clear advantage for a deeper integration of syntax over models that only utilized the bracketing structure of the parse tree thought the most effective way for the compositional operators on the syntax remains unclear.

Wang et al. [11] proposed semantic clustering and convolutional neural network to model short texts based on novel method. The model uses pre-trained word embedding to produce extra knowledge, and multi-scale Semantic Units (SUs). Three pre-trained word embedding for initializing the lookup based on Senna, GloVe, and Word2Vec. The experiments are conducted on two benchmarks: TREC which contains 5,452 training dataset whereas the test dataset consists of 500 questions, and Google Snippets which consists of 10,060 training snippets and 2,280 test snippets from 8 categories. Three pre-trained words are conducted for each benchmark. This method achieves the highest result of 85.1% on Google snippets by Word2Vec and TREC achieve 97.2% when the GloVe word embedding is employed.

III. METHODS

This research compares the CNN performance in classifying YouTube comments towards SVM. In addition, FVEC features also added to test if it bring difference to learning model. The main experiment divided into two category: experiment using down-sampled data and experiment using whole data.

A. Data

The data used in this research was taken from [1]. This research organizing the data into two groups: whole data group and down-sampled data group. Whole data group uses all data and separated into train-test with 9:1 ratio, while down-sampled data group uses randomly selected comments from comments pool in order to get balanced data. This eliminate possibilities of bias model towards unbalanced data. The distribution of whole data group and down-sampled data group shown on Table 1 and Table 2. TRAIN data in each group k-folded and tested among TEST data to find the best accuracy can be reach for each groups. The down-sampled group TRAIN data folded into 5 while whole group data folded into 10.

Table 1: Down-sampled data group distribution

Label	*TRAIN*	*TEST*
Product-Positive	500	400
Product-Neutral	500	400
Product-Negative	500	400
Video-Positive	500	400
Video-Neutral	35	5
Video-Negative	500	400
Uninformative	500	400
Total	**3035**	**2405**

Table 2: Whole data group distribution

Label	*TRAIN*	*TEST*
Product-Positive	519	64
Product-Neutral	3863	397
Product-Negative	826	89
Video-Positive	850	98
Video-Neutral	35	5
Video-Negative	847	101
Uninformative	5349	612
Total	**12289**	**1366**

Since the length of the sentences vary, all sentences in the training data then padded to meet the longest sentence which contain the most word. Each word include padding tranformed into word embedding, replaced each word with numbers and to condition become n-word length.

For every comment processed, each k-word replaced with its representative in word embedding and padding is added to sentence to meet the n length. This transformation enable convolution layer to process the sentence.

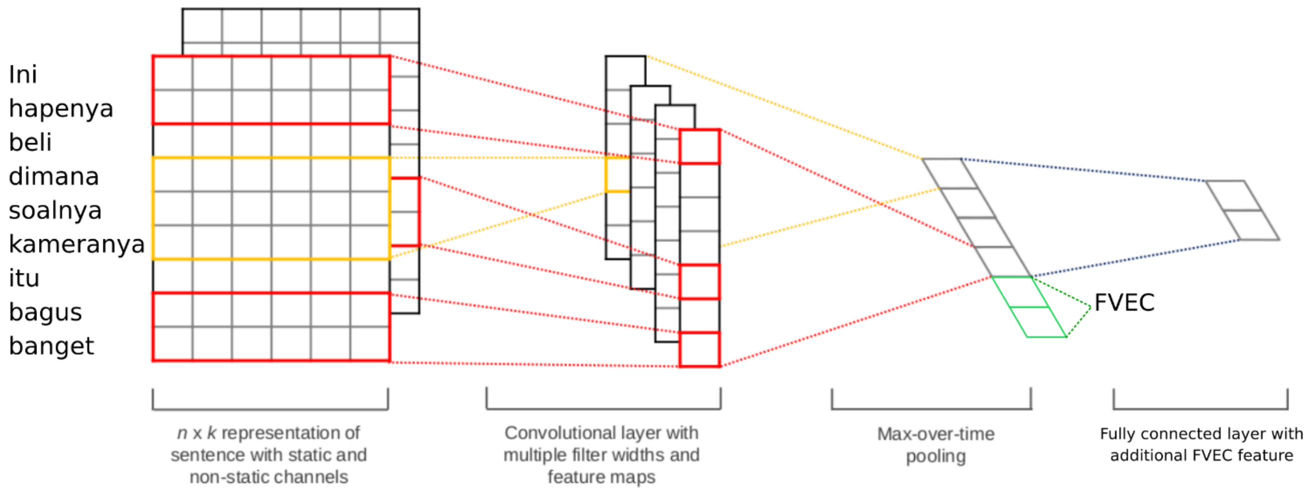

Figure 1: CNN architecture for sentence classification by Kim Yoon, modified by adding FVEC in fully connected layer

B. FVEC-CNN

CNN for sentence classification was introduced by Yoon [5] in various data. Each sentence tokenized then padding is added to make all sentence have a same length, make it *n* x *k* word representation. This form enable convolutional layer to process the sentences to learn corellation between sequential word

The model architecture shown in Figure 1, let $X_i \in R^k$ be the *k*-dimensional word vector corresponding to the *i*-th word in the sentence. A length of sentence of length *n* (padded if necessary) is represented as:

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n \qquad (1)$$

where $\oplus$ is the concatenation operator. In general, $x_{i:i+j}$ refer to the concatenation of words $x_i, x_{i+1}, \ldots, x_{i+j}$. A convolution operation involves $w \in R^{hk}$, which is applied to *h* words in order to produce a new feature. For example, a feature c_i is generated from a window of words $x_{i:i+h-1}$ by:

$$c_i = f\,(w \cdot x_{i:i+h-1} + b) \qquad (2)$$

here $b \in R$ is a bias term and f is a non-linear function similar to the hyperbolic tangent. This filter is applied to each possible window of words in the sentence $\{x_{1:h}, x_{2:h+1}, \ldots, x_{n-h+1:n}\}$ to produce a *feature map* with $c \in R^{n-h+1}$:

$$c = [c_1, c_2, \ldots, c_{n-h+1}] \qquad (3)$$

Max-over time pooling operation applied over the feature map and take the maximum value $\hat{c} = max\{c\}$ as the feature corresponding to capture the most important feature for each feature map.

Because FVEC has no correlation with the words sequences, additional FVEC feature added in the end of convolutional layer, beginning of fully connected layer. Figure 1 shows the architecture of Yoon's CNN architecture modified by adding FVEC in the beginning of fully connected layer.

For regularization, dropout employed on the penultimate layer with a constraint on l_2-norms of the weight vectors. Dropout prevents co-adaptation of hidden units by randomly dropping of the hidden units during forward-backpropagation. For output unit *y* in forward propagation, dropout uses formula 4 where $\circ$ is the multiplication operator of element-wise and $r \in \mathbb{R}^m$ is a *masking* vector. Gradients are back-propagated only through the unmasked units.

$$y = \boldsymbol{w} \cdot (\boldsymbol{z} \circ \boldsymbol{r}) + b, \qquad (4)$$

C. FVEC-SVM

Introduced by Severyn [6] and followed by Rinaldi and Musdholifah [1]. This method combines classic word weighting TF-IDF with additional feature cosine similarity between comments and the video title and counting the negation words in a comment.

Cosine similarity used to detect if a comment contains same product as the title or not because sometimes user name a product on a comment but it does not relate to the title. If it does, then there are probability that the comment talk about the product.

Negation used to inverse the polarity of the comments. If a positive comment contains one negation word, the polarity would be reversed to negative and vice versa.

Lower-cased unigram and bigram used as feature selection to quantify each item present in a comment then classic TF-IDF utilized to turn all features into vector space.

Since the data classified into seven classes, one-versus-rest SVM was performed to find the decision boundary of every class. To find the decision boundary of a class, other class considered as one negative class. Classification is done as much as number of class or seven times.

IV. Results

The data made into two groups, whole data groups and down-sampled data groups. Each group separated into two part, training data and testing data. Training data then k-folded in order to get the highest training accuracy. For whole data groups, 10-fold cross validation performed in training data while 5-fold cross validation performed for down-sampled training data.

A. Down-sampled Data Group

Figure 2 shows the CNN accuracy for training and validation on each fold for down-sampled data. This shows that FVEC relatively helps model to reach higher accuracy, even it was not significant. For validation data, CNN and SVM creates different slope. This may indicate any over-fitting in high training data accuracy.

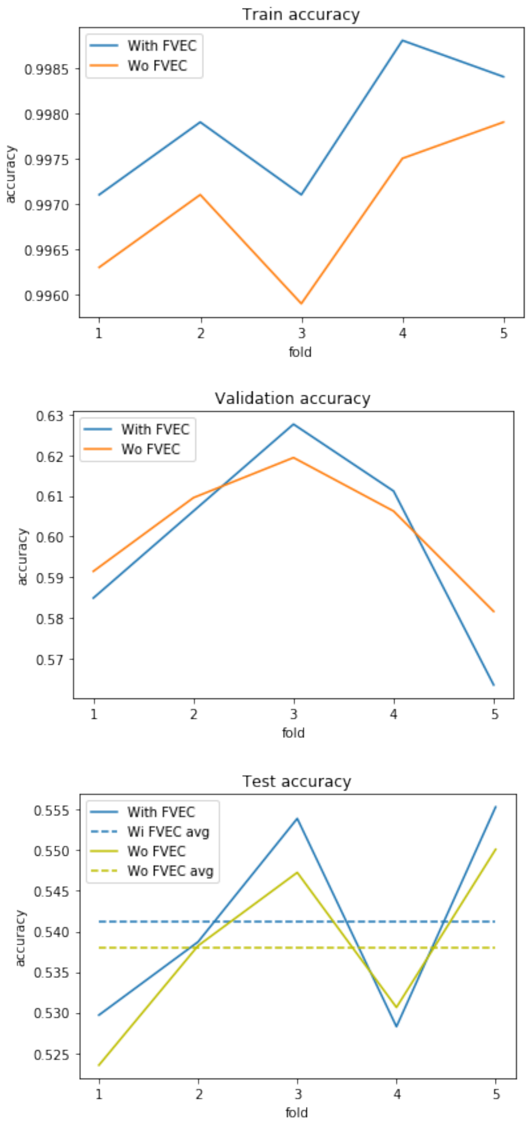

Figure 2: Result of Train, Validation, and Test of CNN performance on down-sampled data group

Figure 3 shows SVM accuracy on down-sampled data on each fold. On training data, FVEC inconsistently helps model to learn. However, FVEC relatively helps increase accuracy on validation data as well as test data. The figure also showed the comparison of SVM with and without FVEC. Once again, FVEC has been proven to increase the accuracy.

Finally, the performance comparison for FVEC-SVM and FVEC-CNN on down-sampled data group shown in Figure 4. FVEC-SVM outperformed FVEC-CNN with slight different around 0.01% difference. Therefore, FVEC-SVM still better than FVEC-CNN for sentence classification.

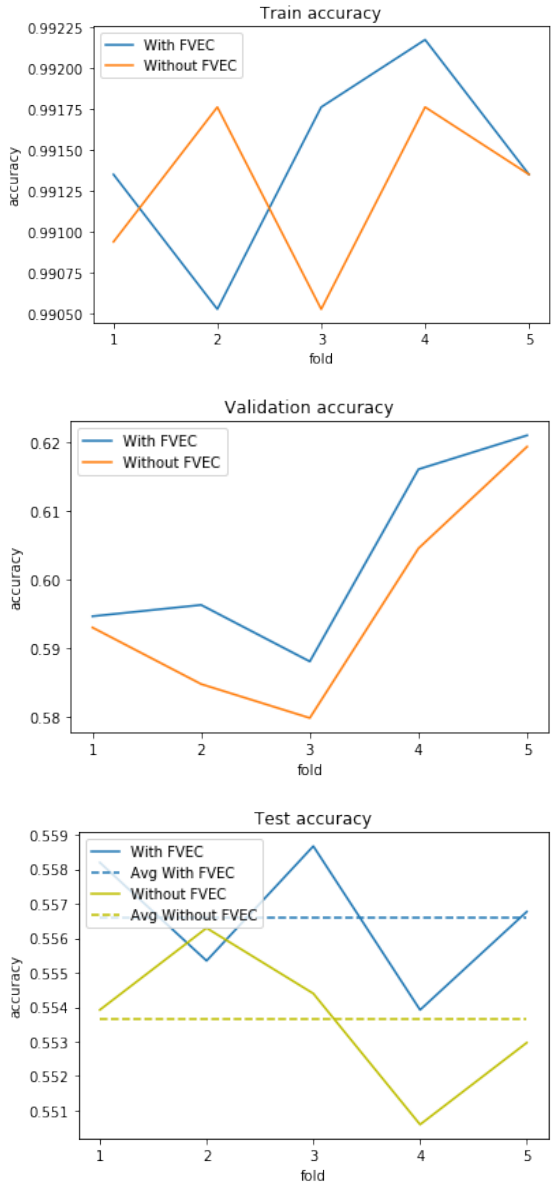

Figure 3: Result of Train, Validation, and Test of SVM performance on down-sampled data group

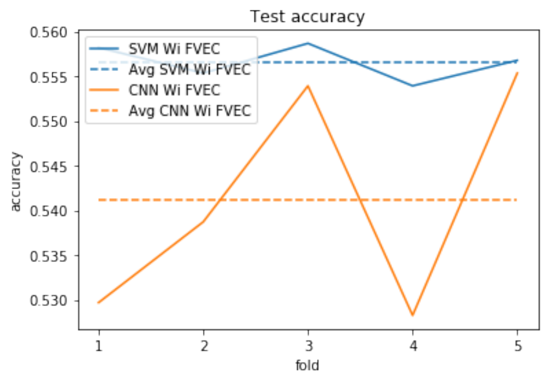

Figure 4: FVEC-SVM vs FVEC-CNN performance on down-sampled

B. Whole Data Group

Whole data group uses 9:1 ratio for training data and testing data on all data. The data selected randomly and proportionally. The training data then 10-folded in order to find the highest training accuracy. CNN and SVM both performed in this data group.

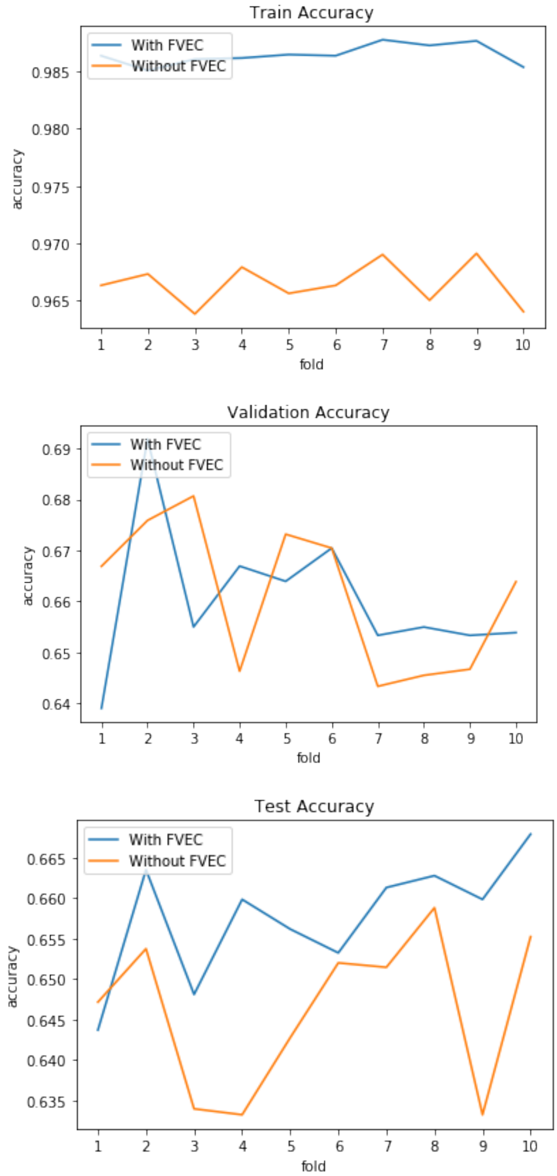

Figure 5: Result of Train, Validation, and Test of CNN performance on whole data group using 10 Fold Cross-Val

Figure 5 shows the result of training, validation, and testing data using CNN. In this case, FVEC has been proven to improve training data accuracy. The result on validation data accuracy seems fluctuate using FVEC, however it really help on predicting test data.

Performance of SVM on whole data group shown on figure 6. On training accuracy, the accuracy increased by utilized FVEC, this applies on validation data and training data as well.

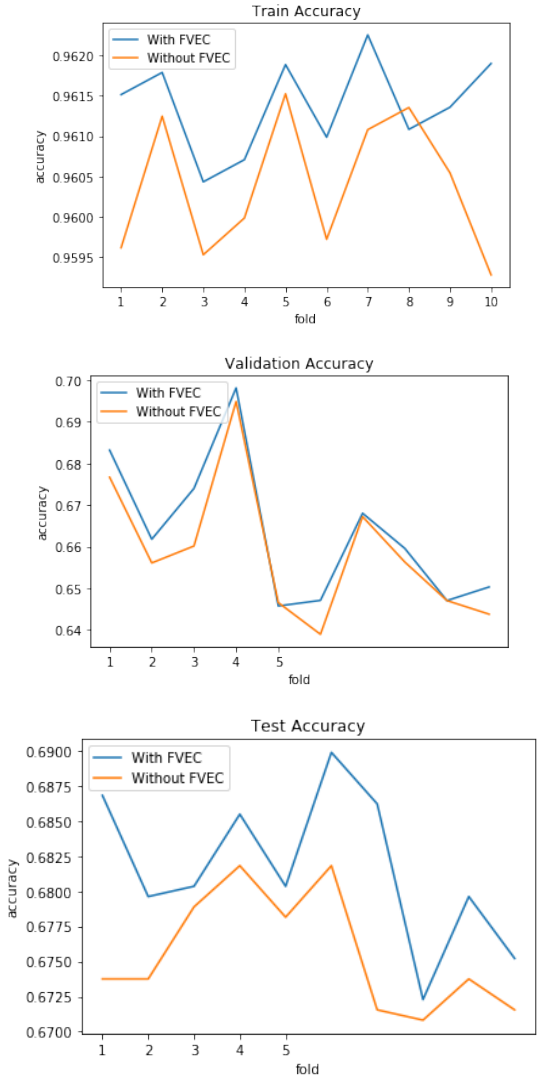

Figure 6: Result of Train, Validation, and Test SVM performance on whole data group using 10 Fold Cross-Val

Lastly, the comparison performance of FVEC-SVM and FVEC-CNN on whole data group shown on figure 7. The result shown that FVEC-SVM accuracy is definitely higher than FVEC-CNN accuracy on every folds.

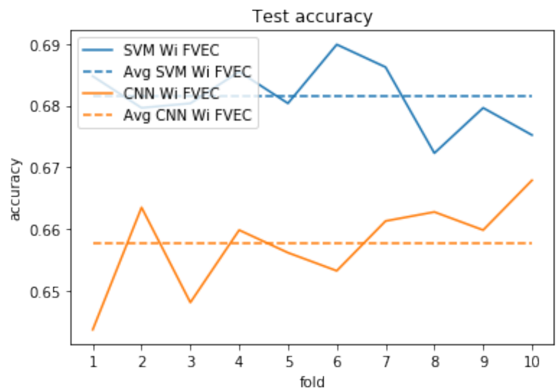

Figure 7: FVEC-SVM vs FVEC-CNN performance on whole data group using 10 Fold Cross Validation

V. Conclusion and Future Works

The proposed FVEC-CNN runs well on both down-sampled and whole data group with decent accuracy. However, FVEC-SVM has been proven to outperformed FVEC-CNN on both down-sampled and whole data group. By this means, CNN neighborhood method is not effective against the data compared to statistical based e.g. TF-IDF. In the future, research continues with method which robust to unstructured data e.g. LSTM, RNN, and Autoencoder.

References

[1] E. Rinaldi, A. Musdholifah, “FVEC-SVM for opinion mining on Indonesian comments of youtube video” 2017 International Conference on Data and Software Engineering (IcoDSE), 1-2 Nov 2017.

[2] S. Shayaa et al., "Sentiment Analysis of Big Data: Methods, Applications, and Open Challenges," in IEEE Access, 2018.

[3] L. Zhang, S. Wang, B. Liu., “Deep Learning for Sentiment Analysis : A Survey”. Computation and Language, Cornell University Library, 2018.

[4] X. Han., and Y. Li., “The Application of Convolution Neural Networks in Handwritten Numeral Recognition,” International Journal of Database Theory and Application, Vol.8, No.3 (2015), pp.367-376.

[5] K. Yoon, “Convolutional Neural Networks for Sentence Classification” Computation and Language (cs.CL); Neural and Evolutionary Computing (cs.NE), 25 Aug 2014.

[6] A. Severyn, A. Moschitti, O. Uryupina, K. Filippova, and B. Plank, “Opinion Mining on YouTube. In 52nd Annual Meeting of the Association for Computational Linguistics,” in Proceedings of the Conference ACL, vol. 1, pp. 1252-1261, 2014.

[7] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, C. Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Proceedings of EMNLP 2013.

[8] R. Socher, J. Pennington, E. Huang, A. Ng, C. Manning. 2011. Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In Proceedings of EMNLP 2011.

[9] N. Kalchbrenner, E. Grefenstette, P. Blunsom,. “A Convolutional Neural Network for Modelling Sentences,” In Proceedings of ACL 2014.

[10] K. Hermann, P. Blunsom. “The Role of Syntax in Vector Space Models of Compositional Semantics.” In Proceedings of ACL 2013.

[11] P. Wang, J. Xu, B. Xu, C. Liu, H. Zhang, F. Wang, H. Hao. “Semantic Clustering and Convolutional Neural Network for Short Text Categorization”. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 352–357, Beijing, China, July 26-31, 2015.