UI at SemEval-2020 Task 8: Text-Image Fusion for Sentiment Classification

Andi Suciati
Universitas Indonesia
Depok, Indonesia
andi.suciati@ui.ac.id

Indra Budi
Universitas Indonesia
Depok, Indonesia
indra@cs.ui.ac.id

Abstract

This paper describes our system, UI, for task A: Sentiment Classification in SemEval-2020 Task 8 Memotion Analysis. We use a common traditional machine learning, which is SVM, by utilizing the combination of text and images features. The data consist text that extracted from memes and the images of memes. We employ n-gram language model for text features and pre-trained model, VGG-16, for image features. After obtaining both features from text and images in form of 2-dimensional arrays, we concatenate and classify the final features using SVM. The experiment results show SVM achieved 35% for its F1 macro, which is 0.132 points or 13.2% above the baseline model.

1 Introduction

SemEval-2020 Task 8 Memotion Analysis consists three tasks, which are Sentiment Classification, Humor Classification, and Scales of Semantic Classes[1]. For this research, we chose to do the task A because we have interest in sentiment analysis area. In task A: Sentiment Classification, we need to classify the sentiment of given English memes to three sentiment categories, which are positive, negative, and neutral. Sentiment analysis is one of NLP task that conducted for extracting the sentiment, emotions, or judgement of reviews and classified it. Mainly, sentiment analysis task requires dataset in form of text. However, the opinion that people express is not only in form of text, such as comments, reviews, but also in other form, for example, image with text, or can be called memes. According to Davidson (2012), an internet meme is a piece of culture, typically a joke, which gains influence through online transmission. In social media, usually the memes are expressed in images that contain short text which express the author opinion towards object. This makes classifying the sentiment of memes a challenge since the opinion in memes can be expressed explicit or implicitly.

In this study, we conducted sentiment classification task on memes by combining the features from images and text, then classified it as positive, negative, or neutral. By using memes as data, it is believed that we can obtained better sentiment classification results because the other data source such as images, can enhance the robustness of models (Yu and Jiang, 2019). The data that used were obtained from SemEval 2020 Task 8: Memotion Analysis (Sharma et al., 2020). As we want to see how traditional machine learning works in this case of classification, we compared four machine learning algorithms and a neural network. Our system shows that it can achieve high scores in the testing set compared to the baseline scores even though the score was still considered low score. The rest of this paper is arranged as follows: In section 2, we review the related works with our study. Section 3 talks about the data that we used, including training and testing data. We describe the research steps that applied in this work and the result from that we obtained in section 4. In section 5, we analysis the result and error. Then we conclude our research in the section 6.

[1] https://competitions.codalab.org/competitions/20629

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1195–1200
Barcelona, Spain (Online), December 12, 2020.

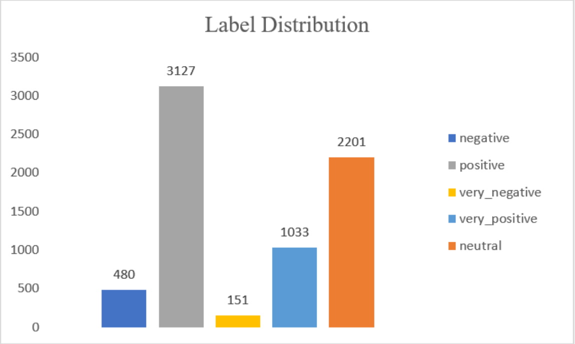

Figure 1: Label distribution

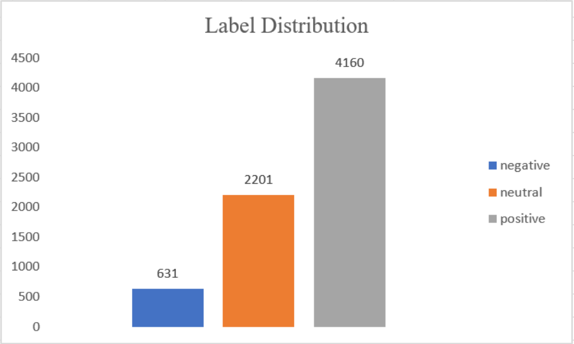

Figure 2: Label distribution after merging label

2 Related Work

There are several researches that using text and images as the features for their research. For in-stance, a study conducted by Sabat et al. (2019) for detecting the hate speech using Multi-Layer Perceptron and calculating its scores. The results show highest scores were obtained by image and text features fusion compared to only using images or text features. Next is the study conducted by Audebert et al. (2019) about document classification using two document datasets that contain digitized documents. The features that used were document embedding combined features extracted using pre-trained model, MobileNetV2. The results show that the overall performances from combination of the features from text and images are better than only using image or text as features in both datasets. Another study conducted by Yu and Jiang (2019). They proposed a target-oriented multimodal using a pre-trained language representations, BERT architecture, for detecting the sentiment. The study also shows the good performances while using images and text features for sentiment classification. In contrast, a research by Gomez et al. (2020) about multimodal model to detect hate speech cannot outperform the textual model even though images are useful for hate speech detection. According to them, the reason can be caused by noisy data, complexity of multimodal relations, and small set of samples.

In our research, we focus of our research is to see the performance of machine learning when classifying using the combination of text and visual features. We compared few algorithms before selecting the best model and use n-gram language model and a pre-trained model, VGG-16 (Simonyan and Zisserman, 2014) for extracting text and images features respectively, before combining and classifying the sentiment.

3 Data

In this part, we show the descriptions of datasets that we used. There are three datasets that provided by SemEval-2020 Task 8, which are trial dataset, training dataset, and testing dataset. However, in our research, we only used training data for model building, then predict the classes for testing data.

3.1 Training Data

The training dataset consist a dataframe contains 6992 rows and 11 attributes, and 6992 images. We found two columns that have missing values including 161 missing values on 'tex_tocr' and 5 rows on 'text_corrected' column. The initial labels of the data were separated into five categories, which are very positive, positive, neutral, negative, and very negative. However, since the label value for positive and very positive is ”1”, as well as both negative and very negative label is ”-1”, those categories were merged and the final labels consist three categories (positive, neutral, negative). Figure 1 and Figure 2 show the distribution of label in each class is imbalanced. Before merging the label (Figure 1), the very_negative and negative labels were very low while positive label was dominating. After merging the label (Figure 2), the total of positive labels almost twice of neutral labels, and for negative labels, positive labels almost seven times higher than negative labels.

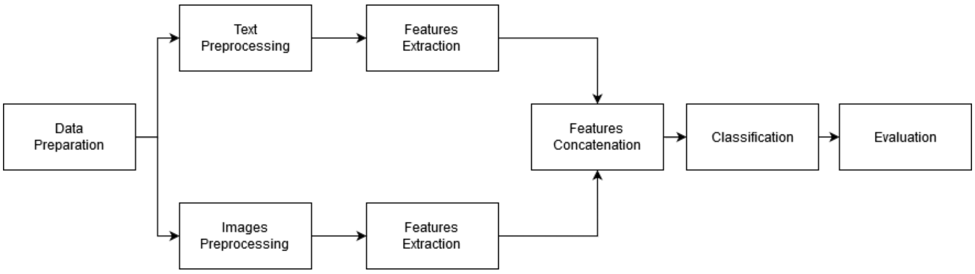

Figure 3: Flow of system

3.2 Testing Data

The testing dataset consist a dataframe that contains 1878 rows and 4 attributes while the image dataset contains 2000 items. There are 19 missing values on ‘OCR_extracted_text’ and 18 rows in ‘corrected_text’ columns. Our final system will predict the label of the testing data and the score is calculated by the organizer.

4 System Description

This section will describe the flow of our system we made in this study that will be implemented in training and testing data. Our system uses few libraries such as Pandas (Wes McKinney, 2010) and NumPy (van der Walt et al., 2011) for data processing, Scikit-Learn (Pedregosa et al., 2011) for text pre-processing, feature extraction for text, and classification models. We also use Keras (Chollet and others, 2015) for image pre-processing, and feature extraction for image. In classification step for training data, we compared three machine learning algorithms for classification which are Multinomial Naïve Bayes, Support Vector Machine (Boser et al., 1992), and Random Forest (Breiman, 2001), and neural network Multi-Layer Perceptron. After that, we selected the best model and predict the sentiment of testing data. The flow of our system classification model consists six stages that can be seen at Figure 3.

- Stage 1 (Data Preparation): At the very beginning, we input training data from SemEval 2020 Task 8. Since the data have two text columns, “text_ocr” and “text_corrected”, we chose “text corrected” as our base text and made new “text” column from it. After that, we filled the null rows in our “text” column with “text_ocr” and dropped the rows that still remained null in “text” column. Then, we filtered out images that are not in data-frame list names, since there were images that are not listed in it. For the testing data, we did the same steps, but we did not drop the null rows from its dataframe and keep the remain null rows because we need to predict all the instances in the testing data. We also merged the positive and very positive labels as well as negative and very negative in this stage.

- Stage 2 (Text and Image Pre-processing): In this stage, the images and text were pre-processed separately. The text pre-processing techniques we implemented including lowercasing the text, remove the numbers, punctuations, and whitespace. We did not remove stopwords and apply stemming step because the text from the memes are relatively short, so, we assume that we will need all the words that appear, and we do not want to miss the information that provided by data. For instance, if there is phrase ‘this isn’t bad’, if we remove the stopwords, it can be changed to ‘bad’ only, since ‘this’, ‘is’, and ‘not’ are stopwords. While the true sentiment of the phrase could be ‘neutral’, or ‘positive’, the models could be misclassified it into ‘negative’ sentiment. As for images, the pre-processing includes resizing the images into 224x224 pixels, expanding the dimensions, and subtracting the mean RGB values since we will use pre-trained model, VGG-16, for extracting the image features.

- Stage 3 (Feature extractions): After pre-processing step, both text and images features were extracted separately. The feature extraction method for text is using n-gram language model which is unigram-bigram, while for images, we implement one of images pre-trained models, VGG-16, by using Keras. After that, we flattened the result into 2-dimensional array.

Model	Text			Text + Images			Images		
	Acc	F1 weighted	F1 macro	Acc	F1 weighted	F1 macro	Acc	F1 weighted	F1 macro
MNB	0.508	0.477	0.322	0.407	0.429	**0.327**	0.401	0.424	**0.325**
SVM	0.513	0.481	**0.328**	0.456	0.457	**0.327**	0.454	0.454	0.324
RF	**0.583**	0.455	0.266	0.576	**0.463**	0.275	0.571	**0.465**	0.279
MLP	0.534	**0.485**	0.319	**0.594**	0.443	0.248	**0.594**	0.443	0.248

Table 1: Comparison of algorithms performances

- Stage 4 (Feature concatenation): After obtained the features from both text and images in 2-dimensional arrays, we concatenated them using NumPy function. Then, we used these concatenated features as final features for classifying the sentiment of memes.

- Stage 5 (Classification): In this part, we classified the sentiment of text and images using concatenated features that merged in previous stage using Multinomial Naïve Bayes (MNB), Super Vector Machine (SVM), Random Forest (RF), and Multi-Layer Perceptron (MLP). The sentiment polarities that we used are positive, negative, and neutral. Since the training data are imbalanced, we applied 10 folds stratified cross-validation method. By doing so, the percentage of distribution for each class in every fold is equal for model.

- Stage 6 (Evaluation): In last stage, we evaluated the classification results. This stage aims to see the performances of models while classifying the training data. The metric scores we evaluated are accuracy (Acc) and F1 measures[2] (F1 weighted and macro). We used two F1 scores because we want to see the performance of the models when the imbalanced label is considered by models using F1 weighted, and while the models only measure the true label of each instances with F1 macro. Besides, the prediction results for testing data also will be assessed with F1 macro. In addition, we compared the performances for every model while classifying using with text features only, and images features only. After that, we chose the best model and then predict the sentiment of testing data.

5 Experimental Result and Analysis

In this section, we show our experimental results in all datasets that provided. The first part shows the results using training data when classified using text features only, combination of text and images features, and images features only. For the second part, it shows the official scores from organizer that achieved by our model when predicting the testing data.

5.1 Training Data Result

By seeing the result of training in Table 1, SVM achieved the best F1 macro score for predicting the sentiment by using text features, while MLP obtained best F1 weighted and RF for accuracy scores. When classifying sentiment with combination of text and images features, MNB and SVM obtained same F1 macro score. For accuracy and F1 weighted, MLP and RF attained high scores for each metric respectively. For images features only, MNB gained the best score for F1 macro, RF for F1 weighted, and MLP for accuracy.

For our final model, we chose SVM based on its performances in three scenarios. RF and MLP may lead in scores for accuracy as well as F1 weighted, however, both of algorithms attained low scores for F1 scores. Those scores prove that imbalance data influence both RF and MLP, consider the training set are imbalanced data. For SVM, it indeed got same F1 macro score with MNB by using text and images features, but the accuracy and F1 weighted from SVM are higher than MNB. Besides, the performance of MNB with images features only, just slightly higher than SVM, which is only 0.001 or 0.1% difference.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

Model	F1 macro	F1 micro
Baseline	0.218	0.308
Our system	**0.340**	**0.470**

Table 2: Testing F1 scores result

In addition, the F1 macro from MNB with text features is 0.006 or 0.6% lower than SVM, while SVM also has higher F1 weighted and accuracy compared to MNB.

5.2 Testing Data Result

From the testing result in Table 2, we can see our model achieved 0.148 and 0.162 points higher than the baseline for F1 macro and F1 micro, respectively. The system is using the combination of text and images features, then using SVM as the classifier. Surprisingly, the F1 macro score it had obtained was higher than its score from training data. It can be happened if the label distribution from testing data is more balanced than training data. Furthermore, the result also shows that by combining text features from n-gram language model and image features extracted using pre-trained model, we can achieve better result than the baseline scores, although the score still considered as bad score as it was below 50% or 0.5.

5.3 Error Analysis

There are several reasons that may affect the score results. First, as we can see before, the data are imbalanced since label distribution was dominated by positive label, following by neutral and negative label. That means the models were mainly learning about positive data and more highly predict the label of testing data as positive than negative or neutral. Second is sometimes the text that extracted are too short and not explicitly expressed the sentiment of memes, for example, the memes that contain phrase “monday got me like”. If we just take the text into the account, the only label that is match is neutral. However, if there is another meme contains same phrase which labelled as positive or negative, the model may become confused then misclassify it. Third, there are many variations of words appear such as slang words and abbreviations. In illustration, ’goooood’ and ’gud’, even though they have same meaning, the machine will treat them as different words if we do not implement the word normalization techniques. Last is an image, usually, can be used to make more than one meme. In instance, if a meme in training data and a meme in testing data use same image but different words and label, the model more likely predict the label of meme in testing data similar to label in training data.

6 Conclusion

In this research, we built UI, a traditional machine learning based system for classifying the sentiment of memes by utilizing the combination of text and images features. The text features were extracted by implementing n-gram language model while images features were using VGG-16. After that, we examined the performance of three machine learning algorithms using training data for classifying memes before selecting the final model. The algorithms are Multinomial Naïve Bayes (MNB), Super Vector Machine (SVM), and Random Forest (RF), also a neural network Multi-Layer Perceptron (MLP). By the comparison of their scores, we chose SVM as our final algorithm, and it achieved 0.350 points or 35% for its F1 macro, which is 0.132 points or 13.2% than the baseline model.

Acknowledgements

The authors acknowledge the PUTI Prosiding research grant NKB-870/UN2.RST/HKP.05.00/2020 from Directorate Research and Community Services, Universitas Indonesia.

References

Nicolas Audebert, Catherine Herold, Kuider Slimani, and Cédric Vidal. 2019. Multimodal deep networks for text and image-based document classification. *arXiv preprint arXiv:1907.06370*.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA. Association for Computing Machinery.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

François Chollet et al. 2015. Keras. `https://keras.io`.

Patrick Davison. 2012. The language of internet memes. *The social media reader*, pages 120–134.

Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas. 2020. Exploring hate speech detection in multimodal publications. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1470–1478.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Benet Oriol Sabat, Cristian Canton Ferrer, and Xavier Giro-i Nieto. 2019. Hate speech in pixels: Detection of offensive memes towards automatic moderation. *arXiv preprint arXiv:1910.02334*.

Chhavi Sharma, Deepesh Bhageria, William Paka, Scott, Srinivas P Y K L, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020. SemEval-2020 Task 8: Memotion Analysis-The Visuo-Lingual Metaphor! In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain, Sep. Association for Computational Linguistics.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition.

S. van der Walt, S. C. Colbert, and G. Varoquaux. 2011. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2):22–30.

Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Jianfei Yu and Jing Jiang. 2019. Adapting bert for target-oriented multimodal sentiment classification. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5408–5414. International Joint Conferences on Artificial Intelligence Organization, 7.