Robust Sentiment Analysis on Twitter feeds

Rajendra Naidu Mannam

Harrisburg University

Robust Sentiment Analysis on Twitter feeds

Abstract

Sentiment analysis are crusial for startups/company growth. In this paper, we discussed about an important problem in sentiment analysis that is best way of representation of text for Models. We have taken the twitter dataset and performed analysis on three richer text representation which are Bag of Words, Word embedding- Word2Vec, Bert- Transformed based language models. In this binary classification project, logistic regression is used as classifier and confusion matrix and classification report are used to identify the best version of text representation.

Introduction

The Internet has increasingly become the platform for expressing opinions, beliefs and reporting problems. Social media acts as public forum in which people share mutual information with other members of the community. We often see individuals writing opinionated comments about many different topics ranging from feedback to news articles to product reviews and even criticizing celebrities and politicians. Most of the time, these comments are not restricted or limited to any structure, allowing writers to express their feelings in an accurate and complex way. These venues are a rich source of information in the form of unstructured text but it's hard to to extract value from this data unless it's organized in a certain way. Doing so used to be a difficult and expensive process since it required spending time and resources to manually sort the data or creating handcrafted rules that are difficult to maintain. People have started automating this task using machine learning called text classification (Aggarwal & Zhai, 2012) and it has proven to be a great alternative to structure textual data in a fast, cost-effective, and scalable way. Sentiment analysis is a sub-topic in natural language processing where the goal is predict the emotion on the underlying text (Pang, Lee et al., 2008). Even though the same algorithms of text classification can be readily applied to sentiment analysis, it is important to know that these are two different tasks. For example, text classification can provide a heads-up that trouble is coming when a new topic appears in your data. In

other words, if the word "spoiled" suddenly spikes in your restaurant chain's feedback, you should look into that area quickly. Sentiment scores also identify potential risks. In this case, a decline in sentiment score indicates that some aspect of your business has left your customers feeling negative toward you.

Hypothesis

In this work, we will be focusing on Sentiment Analysis which is quite popular in digital world these days. Companies often use these kinds of analysis to understand the reputation of their brand. By analyzing social media posts, product reviews, customer feedback, or NPS responses (among other sources of unstructured business data), they can be aware of how their customers feel about their product. They can also track specific topics and get relevant insights on how people are talking about those topics. The biggest challenge is to build technology to support sentiment analysis. The area of natural language processing is maturing every day and new algorithms and text representation techniques are being developed. It is also important to understand what kinds of feature representation yields to better model performance.

Search Questions

In this paper, we look at one popular micro-blog called Twitter and build model for classifying "tweets" into positive and negative sentiments. We will primarily build this model using three kinds of feature representations: 1) *Bag-Of-Words (BOW) models* which represents features based on word occurrence (frequency) in the corpus of text. 2) *Word-Vector models* which represents individual words as feature vectors. 3) *Transformer based language models* which represents the whole document as a feature vector and these models claim to capture the context of the document. For all of the representations, we use a simple linear classifier to classify between positive and negative sentiments.

We use *Sentiment140 Twitter data-set* in this work because Twitter data has several advantages over other data-sets because tweets are collected in a streaming fashion and therefore represent a true sample of actual tweets in terms of language use and content. This data also has a 140 character limit and this allows to measure the performance of

feature representation without worrying too much about length of the text samples (a common problem in NLP). We also chose this data-set in this work because it is well studied by other researchers.

The rest of the paper is organized as follows. In the Related work section, we discuss the theoretical and empirical analysis of various text representations in NLP and compare it to the previous research performed in this domain. In the Methodology section, we discuss about the details of the data-set, the sentiments used in the data and various text representations techniques.

Related Work

Sentiment Analysis

Sentiment analysis has been handled as a Natural Language Processing task at many levels of granularity. Starting from being a document level classification task (Pang & Lee, 2004; Turney, 2002), it has been handled at the sentence level (Zhang et al., 2004) and more recently at the phrase level (Aggarwal & Zhai, 2012).

A significant effort for sentiment classification on Twitter data (Barbosa & Feng, 2010). They use polarity predictions from three websites as noisy labels to train a model and use 1000 manually labeled tweets for tuning and another 1000 manually labeled tweets for testing. They however do not mention how they collect their test data. They propose the use of syntax features of tweets like retweet, hashtags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS of words.

(Gamon, 2004) perform sentiment analysis on feedback data from Global Support Services survey. One aim of their paper is to analyze the role of linguistic features like POS tags. They perform extensive feature analysis and feature selection and demonstrate that abstract linguistic analysis features contributes to the classifier accuracy.

Some of the early and recent results on sentiment analysis of Twitter data are (Pak & Paroubek, 2010). They collect data following a similar distant learning paradigm. They perform a different classification task though: subjective versus objective. For subjective data they collect the tweets ending with emoticons. For objective data they crawl twitter

accounts of popular newspapers like “New York Times”, “Washington Posts” etc. They report that POS and bigrams both help. These approaches, however, are primarily based on ngram models.

All of these approaches, over rely on handcrafted features but in this work, we will try to learn the automatic representation from the latest advancements in NLP.

Text Representation

Various representation techniques for text have been introduced over the course of time. In the recent years, none of these representations have been as popular as the word embeddings, such as Word2Vec (Mikolov, Chen et al., 2013) and GLoVE (Pennington et al., 2014), that took contextual usage of words into consideration. This has led to very robust word and text representations.

Text embedding has been a more challenging problem over word embeddings due to the variance of phrases, sentences, and text. Le (Le & Mikolov, 2014) developed a method to generate the embeddings that outperforms the traditional bag-of-words approach (Harris, 1954). More recently, deeper neural architectures have been developed to generate these embeddings and to perform text classification tasks (Kim, 2014) and some of these architectures involve sequential information of text, such as LSTMs (Palangi et al., 2016), BERT (Devlin et al., 2018).

Methods have been developed that use word embeddings to generate text embeddings without having to train on whole texts. These methods are less costly than the ones that train directly on whole text, and can be implemented faster.

Unweighted average word embedding (Wieting et al., 2015a) generated text embeddings by computing average of the embeddings of all the words occurring in the text. This is one of the most popular methods of computing text embeddings from trained word embeddings, and, though simple, has been known to outperform the more complex text embedding models especially in out-of-domain scenarios. Arora (Arora et al., 2017) provided a simpler method to enhance the performance of text embedding generated from simple averaged embedding by the application of PCA.

The unsupervised text embedding methods face the problem of importance-allocation of words while computing the embedding. This is important, as word importance determines how biased the text embedding needs to be towards the more informative words. DeBoom (De Boom et al., 2016) introduced a method that would assign importance to the words based on their tf-idf scores in the text.

Methodology

In a social media environment, people tend to add subtle hints and sarcasm in their messages to get their message/opinion across without risking offense to anyone in the vicinity. Conventional sentiment analyzers tend to mislabel such tweets, unless they are are provided with a richer representation of text. In this section, we first discuss one widely used social media data-set, we have discussed different performance measures we plan to use in this study and lastly, we did discuss various text representations that we used and the classification model details.

Participants

Twitter is a social networking and microblogging service that allows users to post real time messages, called tweets. Tweets are short messages, restricted to 140 characters in length. Due to the nature of this microblogging service (quick and short messages), people use acronyms, make spelling mistakes, use emoticons and other characters that express special meanings. Following is a brief terminology associated with tweets. **Emoticons**: These are facial expressions pictorially represented using punctuation and letters; they express the user's mood. **Target**: Users of Twitter use the "@" symbol to refer to other users on the microblog. Referring to other users in this manner automatically alerts them. **Hashtags**: Users usually use hashtags to mark topics. This is primarily done to increase the visibility of their tweets.

Procedures

We acquire 160,000 tweets from Twitter Sentiment140 data-set and this data publicly available. They collected the data by archiving the real-time stream. No language,

location or any other kind of restriction was made during the streaming process. In fact, their collection consists of tweets in foreign languages. They use Google translate to convert it into English before the annotation process. Each tweet is labeled by a human annotator as positive, negative, neutral or junk. The "junk" label means that the tweet cannot be understood by a human annotator. A manual analysis of a random sample of tweets labeled as "junk" suggested that many of these tweets were those that were not translated well using Google translate. They eliminate the tweets with junk label for experiments. This leaves us with a balanced sample of 160,000 tweets(80,000 tweets each from classes positive and negative) as shown in Figure1.

Measures

We have trained logistic regression as classification models with various feature representations in this study. Typically, The performance of any classification model can be defined as the percentage of outputs which are correctly predicted by the machine learning algorithm given a set of inputs. One important tool used to evaluate the performance of a classification task is the confusion matrix which visually represents the following things:

1. True Positives (TP) indicates the cases where the predicted labels are same as true labels for the positive class.

2. True Negatives (TN) indicates the cases where the predicted labels are same as true labels for the negative class.

3. False Positives (FP) indicates the cases where the predicted labels is not as same as true labels for the positive class.

4. False Negatives (FN) indicates the cases where the predicted labels is not as same as true labels for the negative class.

We have used this measures and calculate **Precision**, **Recall** and **F1-score** for each model trained and compare these metrics across various text representations.

Analysis

Once data pre-process techniques are performed on data-set, we have prepared a text representation that would encompass the essence of the comments in the training data, as well as work on newer texts with minimum loss of information. To test such a representation, we have used bag of words (to capture words with high emotional expression), Word vectors embedding (to make the representation semantically sound) and language model embedding (to capture context as well as to make the representation both semantically and syntactically sound).

Feature Representation:

1. **Bag of Words (BOW):**We have trained nltk part-of-speech tagger to extract nouns, verbs, adjectives, and adverbs from the comments. Words belonging to these categories hold most of the information that is present in the text. We, then, computed a term frequency-inverse document frequency (tfidf) matrix that would capture the word occurrence based on word count and its frequency in the training data.

2. **Word embedding models:** In order to capture the meaning behind the tweet, we have implemented an algorithm that would compute text embedding of the given comment by averaging across the vectors of the words occurring in the comment. The result is the center of the collection of words that we treat as the unweighted text embedding or text2vec (Wieting et al., 2015b). For word embedding, we used the word2vec representation (Mikolov, Sutskever et al., 2013). We have selected this representation, as it has performed really well in generating text embedding and word-related operations such as odd-word-out and contextually similar words. They capture the semantic qualities of the word, which are reflected in the vector. We want to leverage the use of pre-trained models in this work because, most of them are trained on Wikipedia or News which are grammatically richer than the twitter feeds and word embeddings helps to represent words in a n-dimensional vector space such that words which are syntactically or semantically close to each

other, are clustered together in the resulting embedded vector space.

3. **Transformer based Language Models:** Large transformer based Language models such as BERT which is pre-trained on massive amounts of text such as WIkipedia, DBpedia, online articles and books, provide richer text representation at different levels (word, sentence, document) and these representations often capture the contextual dependency in the corpus of text. We used the features from one such language models in this study and compared this with other approaches discussed.

Classifier: In this study, we used logistic regression as classifier. We trained several models using this classifier with various text representations and compared metrics using classification reports and Confusion matrix to identify the best feature representation for the sentiment analysis task.

Result

We discuss this section in the following way. First, we would like to give an overview on the dataset. Second, we talk about out preprocessing where we use latest natural language processing (NLP) techniques. Next, we talk about various ways of representing text or feature extraction and finally, we talk about our modeling methodology and report our results.

Data set overview:

We have selected the sentiment 140 dataset which is available in Kaggle. It contains 1,600,000 tweets extracted using the twitter API. This dataset contains 6 fields which are sentiment, ids, date, flag, use and, text. Our analysis will be performed on text data to detect the nature of tweet (positive or negative). To accomplish this analysis, text and sentiment fields are required as input and target respectively and the rest of fields are not useful for sentiment analysis, so those fields are removed. “Target” field has 2 classes that have been annotated with 0 as negative class, 4 as positive class and “text” field is

represented with English text along with URLs and each tweet has a 140-character limit This dataset is balanced, and data distribution graph can be visible in figure 1.

Data preprocessing:

We have used multiple NLP techniques in this step. Firstly, we made our corpus as a unique bag of words by removing stop words, stemming, and lemmatization. Removed the stop words in English like "a", "an". "the", etc. which provides little to no unique information that can be used for classification problems. Performed the lemmatization to get the morphological analysis of the words, for example, dataset contains words like rocks and better which can be converted to rock and good respectively. Applied stemming to reduce inflected words to their stem word (base or root form). For example, word "flying" can be assigned to "fly" when we apply stemming.

Feature extraction techniques:

Machine learning models cannot predict target class using raw data (text) as input. To make data understandable to machine learning models, we have converted the data into vector of floats and later assigned weight for their importance. We had computed three text representation technique to extract the feature which will be useful for classifier. Before extracting the features, we have split the dataset into training and testing sets. We have used three popular pre-trained text representation models which are Bag-of-Words (TF-IDF), Word embedding Model (Word2Vec) and Transformer based language Model (BERT).

1. **Bag-Of-Words/ TF-IDF:**"Term frequency- inverse document frequency" (TFIDF) matrix captures the frequency of each word and its importance. In order to achieve this, we have used sklearn's TfidfVectorizer class to transform text to feature vectors. In TFIDF representation, the words are represented based on the importance of that word in the corpus. For example, a frequently occurring word would be of less importance than that a word which occur less frequently. We have also used trained NLTK part of speech tagger to extract noun, verbs, adjectives and adverbs

from the tweet.

2. **Word2vec:** The basic idea behind this representation is that the words that occur in similar context tend to be closer to each other in vector space. The word2vec tool has two models: Skip-gram and continuous bag of words (CBOW). When we give word as tokens to skip-gram model predicts the neighboring words from the list of available words in database. In contrast, the CBOW model predicts the current word, by given the neighboring words in the tweet. In our approach, we have used a pre-trained skip-gram based word2vec model which was trained on news articles (Google News).

3. **Language Model:** Third Model is BERT (Bidirectional Encoder Representations from transformers) Model. BERT is a bidirectional model which means that BERT learns information from both the left and the right side of a token's context during the training phase. BERT can understand the meaning of a language since it is pre-trained on a large corpus of unlabeled text including the entire Wikipedia (that's 2,500 million words!) and Book Corpus (800 million words). BERT is pre-trained on two NLP task which are Masked Language Modeling, Next Sentence Prediction. Masked Language Models learn to understand the relationship between words and Next Sentence Prediction can understanding of the relationship between sentences. here we can use embeddings from BERT as embeddings for our tweet feed.

Result of modeling and performance measurements:

We have trained Logistic regression as our base classifier with three text presentations (Bag-Of-Words, Word2Vec, BERT). Classification metric and confusion matrix have been reported to compare results and identify the best text representation for sentiment analysis. BERT Modal provided accuracy of 76%, this dataset has a balanced set of classes and accuracy is sufficient to tell about model performances. To be sure, we also look at look precision, recall and f1-scores. Precision tells about the percentage of predicting positive tweet among all predicted classes. Recall tell us the percentage of predicting positive tweet among all actual positive tweets. From figure 4, we can observe BERT

model prediction. For positive tweet class, precision and recall are 76% and 76%. For negative tweets class, precision and recall are 77% and 76%. from Confusion matrix, BERT predicted positive tweets (TP: True positive) more accurate that is 38.33% than negative tweets (TN: true negative) which is 37.8this model predicted FP (False positive) and FN (False Negative) are 11.67% and 12.13% respectively.

From Figure 3, we can observe Word2vec predictions, this Model predicted 73% (accuracy) of test correctly, when logistic regression model is applied. For class 0 which is negative tweets,Precision (percentage of predicting Negative tweet among all predicted classes) and recall (percentage of predicting Negative tweet among all actual Negative tweets) are 76% and 71% respectively. For class 1 which is positive tweets,Precision (percentage of predicting positive tweet among all predicted classes) and recall (percentage of predicting positive tweet among all actual positive tweets) are 71% and 76% respectively. From Confusion matrix, Word2Vec predicted positive tweets (TP: True positive) more accurate that is 36.83% than negative tweets (TN: true negative) which is 36.50%. This model predicted FP (False positive) and FN (False Negative) are 14.83% and 11.83% respectively.

Bag-Of-Words/TF-IDF Model prediction can be observerd from Figure 2, the accuracy of this model is 73%. For class 0 which is negative tweets,Precision (percentage of predicting Negative tweet among all predicted classes) and recall (percentage of predicting Negative tweet among all actual Negative tweets) are 78% and 67% respectively. For class 1 which is positive tweets,Precision (percentage of predicting positive tweet among all predicted classes) and recall (percentage of predicting positive tweet among all actual positive tweets) are 70% and 79% respectively. From Confusion matrix, Bag-Of-Words/TF-IDF Model predicted positive tweets (TP: True positive) more accurate that is 38.67% than negative tweets (TN: true negative) which is 34.50%. This model predicted FP (False positive) and FN (False Negative) are 16.83% and 10% respectively.

After comparing the three text representations, we can see BERT model predicted sentiment better than word2vec and Bag of words. BERT was able to predict sentiment with accuracy of 76% while Bag-Of-Words and Word2Vec 74% and 73% respectively.

Conclusion

In the work, we discussed about an important problem in sentiment analysis and discussed how it can be improved upon with a richer text representation. We proposed the use of a popular social media dataset ‘twitter’, and performed analysis using various text representations. Later, trained them in logistic regression.We have reported results using standard classification metrics Precision, Recall and Accuracy. In our experiments, we have performed analysis using Bag-of-Words (TFIDF) representation which predicted 74% of test data. Next, we have performed analysis using word vector models which predicted 73% of test data. Finally, we have performed analysis using BERT models which predicted 76% of test data. The BERT Model Performed very well compared the two former methods. We are excited to state that the state-of-the-art text representation such as language model are said to perform the best on many NLP tasks, and we confirm that in our study on sentiment analysis.

References

Aggarwal, C. C. & Zhai, C. (2012). A survey of text classification algorithms. (C. C. Aggarwal & C. Zhai, Eds.). In C. C. Aggarwal & C. Zhai (Eds.), *Mining text data*. Springer. http://dblp.uni-trier.de/db/books/collections/Mining2012.html#AggarwalZ12b

Arora, S., Liang, Y. & Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings, In *5th international conference on learning representations, ICLR 2017, toulon, france, april 24-26, 2017, conference track proceedings*. https://openreview.net/forum?id=SyK00v5xx

Barbosa, L. & Feng, J. (2010). Robust sentiment detection on twitter from biased and noisy data, In *Proceedings of the 23rd international conference on computational linguistics: Posters*. Association for Computational Linguistics.

De Boom, C., Van Canneyt, S., Demeester, T. & Dhoedt, B. (2016). Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recogn. Lett.*, *80*(100), 150–156. https://doi.org/10.1016/j.patrec.2016.06.012

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gamon, M. (2004). Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis, In *Proceedings of the 20th international conference on computational linguistics*. Association for Computational Linguistics.

Harris, Z. (1954). Distributional structure. *Word*, *10*(23), 146–162.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Le, Q. & Mikolov, T. (2014). Distributed representations of sentences and documents, In *Proceedings of the 31st international conference on international conference on machine learning - volume 32*, Beijing, China, JMLR.org. http://dl.acm.org/citation.cfm?id=3044805.3045025

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space, In *Proceedings of iclr workshop 2013*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. (2013). Distributed representations of words and phrases and their compositionality, In *Advances in neural information processing systems*.

Pak, A. & Paroubek, P. (2010). Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives, In *Proceedings of the 5th international workshop on semantic evaluation*. Association for Computational Linguistics.

Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X. & Ward, R. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, *24*(4), 694–707. http://dl.acm.org/citation.cfm?id=2992449.2992457

Pang, B. & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, In *Proceedings of the 42nd annual meeting on association for computational linguistics*. Association for Computational Linguistics.

Pang, B., Lee, L. Et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), 1–135.

Pennington, J., Socher, R. & Manning, C. D. (2014). Glove: Global vectors for word representation, In *Empirical methods in natural language processing (emnlp)*. http://www.aclweb.org/anthology/D14-1162

Turney, P. D. (2002). 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*. Association for Computational Linguistics.

Wieting, J., Bansal, M., Gimpel, K. & Livescu, K. (2015a). Towards universal paraphrastic sentence embeddings. *CoRR*, *abs/1511.08198*.

Wieting, J., Bansal, M., Gimpel, K. & Livescu, K. (2015b). Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

Zhang, Y.-L., Zhao, Y.-C., Wang, J.-X., Zhu, H.-D., Liu, Q.-F., Fan, Y.-G., Wang, N.-F., Zhao, J.-H., Liu, H.-S., Ou-Yang, L. Et al. (2004). Effect of environmental exposure to cadmium on pregnancy outcome and fetal growth: A study on healthy pregnant women in china. *Journal of Environmental Science and Health, Part A*, *39*(9), 2507–2515.

Appendix

https://github.com/rajendranaidu495/ANLY699Project.git

Figures

Figure 1. Data distribution

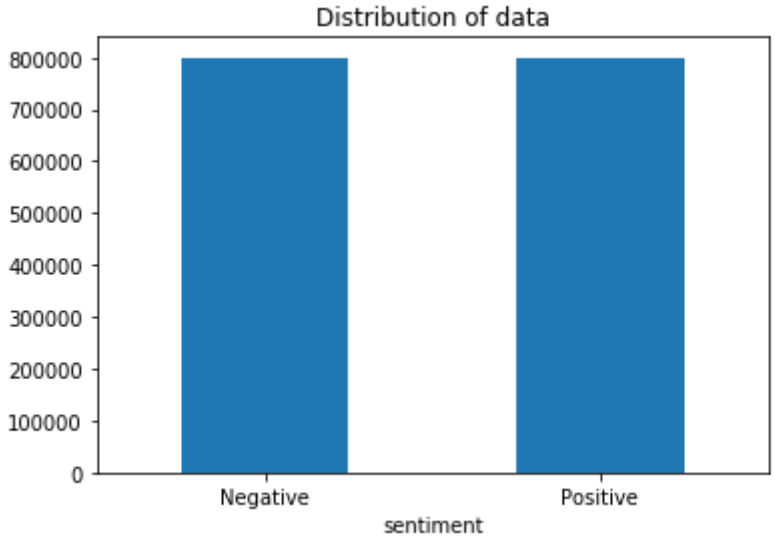

Figure 2. Bag-Of-Words/TF-IDF Model

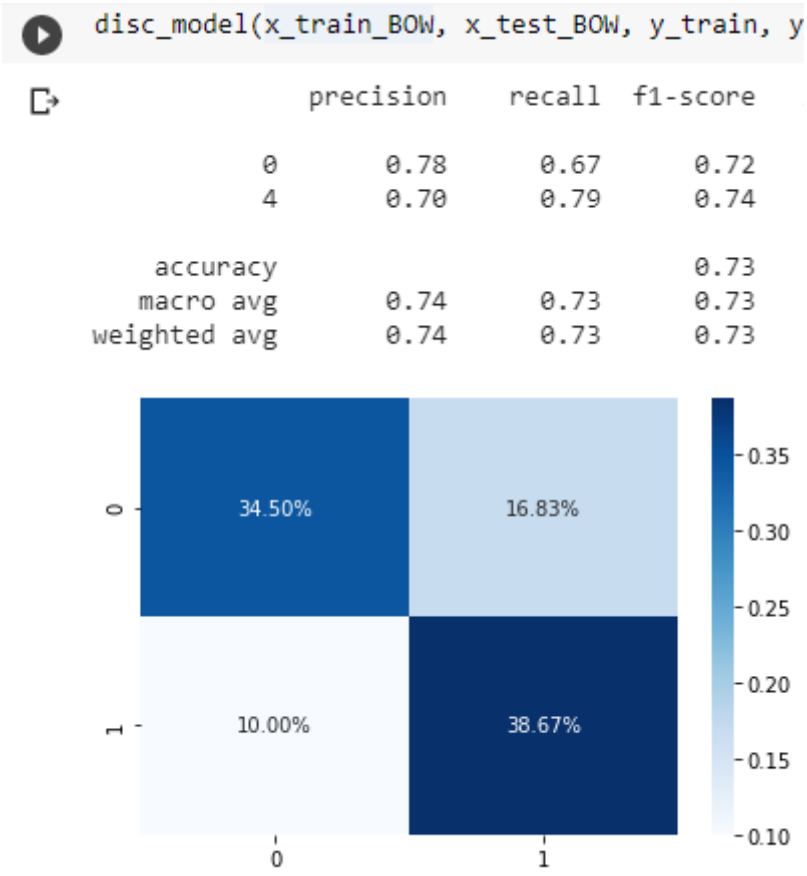

Figure 3. Word2Vec Model

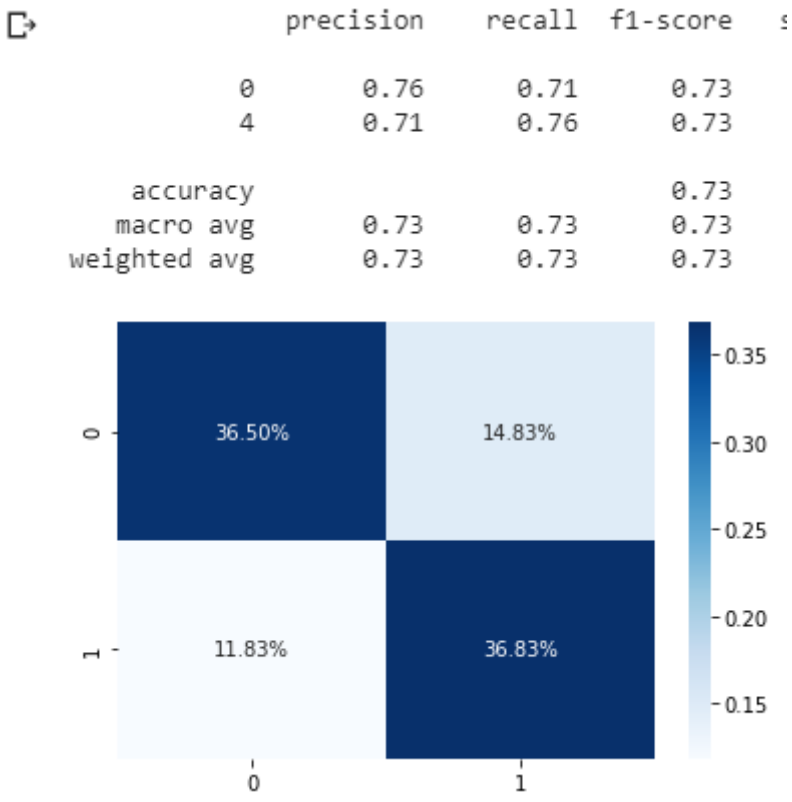

Figure 4. BERT Model

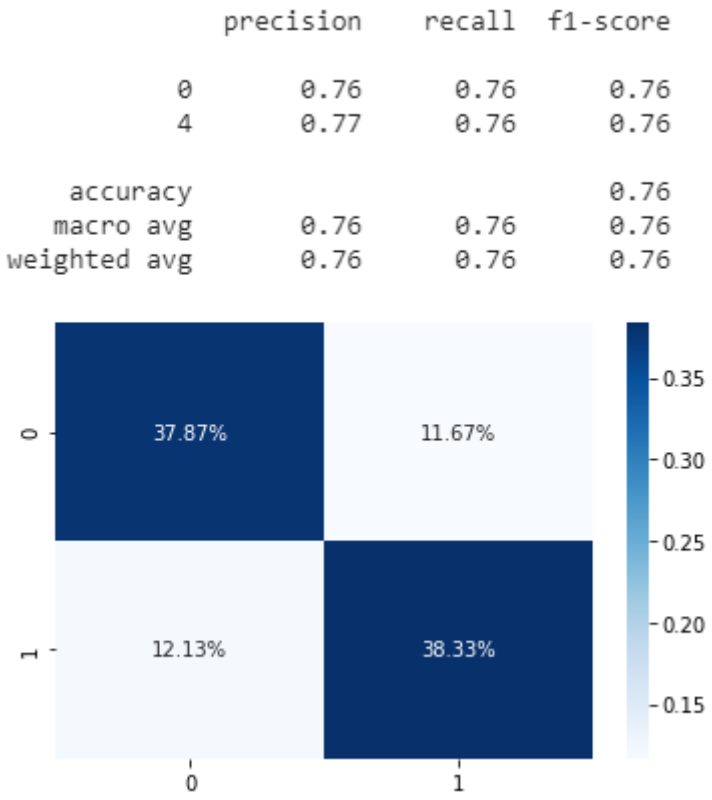

Figure 5. Compare between Models

Feature	Precision	Recall	Accuracy
Bag of words	78	67	74
Woed2Vec	76	71	73
BERT	**77**	**76**	**76**