CSE574: Sentiment analysis of political Indonesian tweets

Stephanie N. Richter
Department of Linguistics
University at Buffalo
snrichte@buffalo.edu

Georgius Aland Abibenar Feltama
Department of Computer Science and Engineering
University at Buffalo
georgius@buffalo.edu

May 5, 2020

1 Project Details

Team Name:	Machine Belajar
Project Name:	Sentiment analysis of political Indonesian tweets
Team Members:	Stephanie N. Richter; Georgius Aland Abibenar Feltama
GitHub Repo:	https://github.com/stephrichter/machine-belajar

2 Problem Statement

Our team is tackling the problem of performing analysis on a low-resource language – for us, that will be Indonesian. English is the most widely studied language in the world; anyone can very quickly and easily set up almost any model for any NLP task in English, with ample support from previous research and resources contributed by the NLP community at large. This is not true for a language like Indonesian. We are exploring different methods of categorizing Indonesian text (specifically tweets of political discourse) for sentiment analysis. For the first milestone, we used K-means clustering (KMC) on a Word2Vec model to find sentiment values of tweets, to questionable results; for the second milestone, we improved the results of our KMC model and have investigated a supervised approach, specifically a convolutional neural network (CNN) which utilizes a Word2Vec model. We chose a CNN model as we had not seen it specifically in the Indonesian sentiment analysis literature. For the third milestone, we have examined a recurrent neural network (RNN) model, which does not utilize word vectors at all.

We are working in Python 3, on Google Colaboratory. We are using `gensim` to load the Word2Vec model and `sklearn` for KMC; for the CNN model, we are using these same models as well as `keras`; for the RNN, we are mostly using `tensorflow`. We are also using `pandas` and `numpy` for general data manipulation, and other such basic Python libraries.

3 Method

Due to the small size of the labeled dataset we have found (discussed in Section 4), we had first thought that a supervised task was not realistic. We instead first performed an unsupervised task on the categorization of various words in a Word2Vec model using KMC. We used a pretrained Word2Vec model using the CBOW architecture trained on an Indonesian Wikipedia page dump.[1] We then formed two clusters within the model with the hope that the clusters would roughly distinguish themselves in regard to the polarities.

While we only had 5478 pre-labeled tweets with binary judgements, we thought it worthwhile to still attempt to use these tweets in a supervised task to compare the results with our KMC model. Thus, we trained a CNN on a subset of the labeled data and tested it on the remainder. This model performed significantly better than we would have thought it would; it also performed better than KMC. For our third model, we have decided to forego using word vectors and instead dedicate

We chose KMC for our first model because of the few resources available for analysis of Indonesian text. There have been somewhat successful sentiment analysis experiments done with lexicon-based methods. However, these are not really machine learning-based. We use a pretrained Word2Vec model which has "learned" the distributions and similarities of text, and assume that there will be some distinction of the positivity and negativity of certain words included within. There are flaws with this, as we discuss in Section 4.1.

4 Results

We decided to focus on only the categorization of positive and negative tweets. We use the same Word2Vec model for both KMC and CNN.

We first performed out analyses solely on the pre-labeled tweet set that we have found online. For our second set of tests, we append our set of political tweets to the pre-labeled set. This is discussed further in Section 5.

4.1 KMC

Results were largely obtained by following the guide in the corresponding footnote.[2] The initial results for KMC from Milestone 1 were, frankly, awful, as

[1] The exact model we used can be found here: `https://github.com/Kyubyong/wordvectors`

[2] `https://github.com/rafaljanwojcik/Unsupervised-Sentiment-Analysis`

can be seen from Table 1. However, we were able to slightly improve the overall results by more carefully selecting parameters for the cluster selection.

From these still-poor results, it is clear to us that the Word2Vec model we have utilized is not going to work for this task.[3] Wikipedia pages are almost always written in a neutral register, and so sentiment values are not very conducive for this domain. Moreover, Wikipedia pages contain many formal words, as well as much native English text. (We were doubtful, yet hopeful that adequate sentiment values could be obtained from these Wikipedia vectors.) As can be seen from Figure 1, the clusters which KMC produced are not very separable, and so do not point to having very good results.

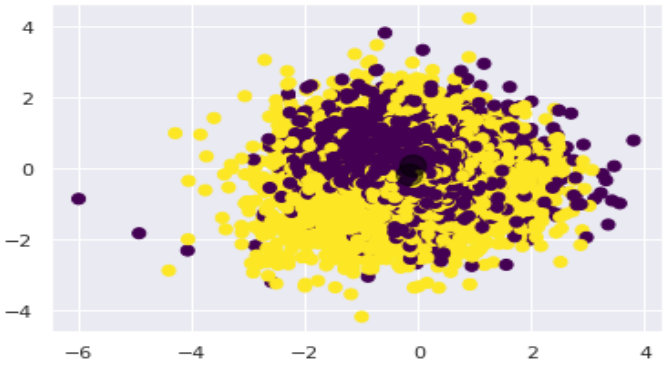

Figure 1: Word vector clustering results of KMC method, with two cluster centers shown as dark spots in the center of the image. Purple dots represent negative words; yellow, positive.

With the addition of our strictly political tweets, we see very similar results to those we had without the political tweets, as seen in Table 2. It is debatable if including these tweets has made any difference at all; we would likely need more political tweets in order to see a larger difference. Additionally, a Word2Vec model trained using the skip-gram architecture may behave better. Nawangsari, Kusumaningrum, and Wibowo (2019) notes that models trained using skip-gram, in general, will be better-suited for sentiment analysis, as skip-grams rely on unique words just as sentiment analysis typically does. Political discourse uses many infrequent words, and so a skip-gram model may behave better at this.

[3]We had planned to pursue a potentially better word vector model that was trained on both Wikipedia and Common Crawl (the FastText vectors which are avilable here: `https://fasttext.cc/docs/en/crawl-vectors.html`), but the conversion of FastText vectors into Word2Vec vectors proved very challenging.

	acc	prec	recall	F1
KMC-M1	0.506	0.465	0.297	0.362
KMC-M2	0.491	0.474	**0.692**	0.563
CNN-M2	0.687	0.668	0.657	0.662
RNN-M3	**0.734**	**0.846**	0.629	**0.721**

Table 1: Results from the pre-labeled set of 5478 positive and negative tweets, comparing both tests of the KMC model, and the CNN and RNN models for Milestone 1, 2, and 3 (M1, M2, and M3, respectively).

	acc	prec	recall	F1
KMC-M3	0.494	0.484	0.690	0.568
CNN-M3	**0.681**	0.645	0.713	0.677
RNN-M3	0.656	**0.739**	**0.746**	**0.743**

Table 2: Results from the combination of the pre-labeled 5478 positive and negative tweets and our 227 well-formed tweets, for a total of 5705. We compare KMC, CNN, and RNN results on this total set.

4.2 CNN

Results were largely obtained by following the guide in the corresponding footnote.[4] For our first run, the model was trained on 90% of the 5478 labeled tweets that we had immediate access to, and was tested on the remainder. After some experimentation, we found the best results by training the model for 3 epochs on a batch size of 64, as our tweet set is still very small. The results were relatively good compared to what we had accomplished previously on KMC, as can also be seen in Table 1.

However, when training the model with our 5705-tweet long total dataset, we saw some increased performance on average, as seen in Table 2. This model was significantly better at recall. As Le, Moeljadi, Miura, and Ohkuma (2016) notes, using a normalizer for CNN likely will not help performance when performing sentiment analysis, and so we have not used it here.

4.3 RNN

Results were largely obtained by following the guide in the corresponding footnote.[5] For our first run, the model was trained on 90% of the 5478 labeled tweets that we had immediate access to, and was tested on the remainder. After some experimentation, we found the best results by training the model for 5 epochs on a batch size of 64, as our tweet set is very small. The results we obtained were the best out of all three models we tested, as seen in Table 1. The model was surprisingly precise in its judgements, though recall suffered; overall, however, it prevailed as the most successful of the three models.

[4] `https://towardsdatascience.com/cnn-sentiment-analysis-1d16b7c5a0e7`

[5] `https://towardsdatascience.com/sentiment-analysis-using-rnns-lstm-60871fa6aeba`

When training the model with our 5705-tweet long total dataset, we saw an interesting change in performance, as seen in Table 2. This time, the model was significantly better at recall, and had more even performance overall.

5 Dataset

As Indonesian is a low resource language, we had great difficulty in locating a satisfactory dataset for sentiment analysis. As far as we have been able to ascertain, there are no large human-judged datasets of Indonesian reviews or tweets available for public download. The best we have found is a small set of tweets available on GitHub which includes roughly 10,000 preprocessed tweets with labels of positive, neutral, or negative sentiment, and roughly 450,000 unlabeled tweets.[6]

This is not ideal for a supervised learning task, as there are relatively few labeled tweets and it would be unrealistic for our team to label many more. Additionally, it is unclear what domain these tweets were found in, or how the researchers found these particular tweets. As we are primarily interested in tweets of notable political interest, the sentiment expressed in these labeled tweets may not reliably expand to the political realm as well.

It is for these reasons that we decided to pursue an unsupervised task initially, and decided to use these labeled tweets as part of our test set. Now, we have relaxed our intuition and decided to attempt a supervised task.

To aid us in Milestone 3, we have acquired a small set of specifically political tweets labeled by our team. We have used the Twitter API in Python to extract tweets by following hashtags that correspond with Indonesian political topics or events. These political tweets ranged in dates from January 1 2020 until April 14 2020. The search for tweets was based on hashtag searches of recently relevant political topics in Indonesia: *#IndonesiaButuhPemimpin, #pemerintah, #pemilu2019, #jokowi,* and *#politikindo*. From this search, we have gathered 1,928 tweets about Indonesian politics.

The selected tweets were then labeled using TextBlob, a basic NLP tool for English, to obtain a basic first-level label (positive, negative, neutral) for any tweets which may contain a fair amount of English text, which is relatively common in Indonesian. From the total of 1,928 tweets, we were able to see a breakdown of roughly 1500 neutral tweets, 360 positive tweets, and 136 negative tweets. We then manually checked many of these tweets to obtain a larger chunk of human-rated tweets, for a total of 227 negative and positive tweets to be added to out dataset. The tweets were then saved into CSV format with the original tweets. The original tweets have been pre-processed to remove noise, such as URLs, username mentions, hashtags, symbols, and emoji; we do the same for tweets which we have obtained, for a total of 5705 tweets.

Because we are unaware of exactly where the original tweets came from, and our 227 tweets still make up a very small portion of the total, there is some

[6]Details on this dataset can be found here: `http://ugm.id/idsadataset`

danger for irregularity in our data posing an issue for our final results. However, we find the difference in average performance to still be notable.

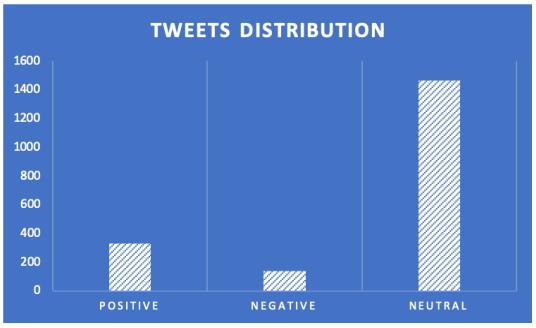

Figure 2: Sentiment distribution of 1,928 Indonesian political tweets.

References

Le, T. A., Moeljadi, D., Miura, Y., & Ohkuma, T. (2016, December). Sentiment analysis for low resource languages: A study on informal Indonesian tweets. In *Proceedings of the 12th workshop on Asian language resources (ALR12)* (pp. 123–131). Osaka, Japan: The COLING 2016 Organizing Committee.

Nawangsari, R. P., Kusumaningrum, R., & Wibowo, A. (2019). Word2Vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study. *Procedia Computer Science*, *157*, 360 - 366. (The 4th International Conference on Computer Science and Computational Intelligence (ICCSCI 2019) : Enabling Collaboration to Escalate Impact of Research Results for Society) doi: https://doi.org/10.1016/j.procs.2019.08.178