NDSC 2019 Git Gud

Done by: Yang Zhenyan (Leader) [yang.zhenyan@u.nus.edu](mailto:yang.zhenyan@u.nus.edu), Brandon Phua [e0311208@u.nus.edu](mailto:e0311208@u.nus.edu)

# Preliminary studies

The main takeaway of our preliminary study of the dataset provided has a wide distribution in the number of records for each category, with Powder accounting for a large share (12.2%) of the dataset, while the smallest was SPC, at only 48 entries. This is a major issue as the extremely large variation in data led to training bias in machine learning. Another possible issue is the language on the titles on some of the products included Bahasa Indonesia, which we fear to be an issue which might increases the number of features and complicated the problem. In addition, the small variance in the words used in the titles was an issue, as the lack of enough unique words would result in a lower accuracy. The last issue we noted is the category ‘Others’.

# Algorithmic Body

## Preprocessing

All the titles in were processed to spilt the string of words into individual strings and stored them into a list. Then, from genism.models, Word2Vec was used to convert individual title lists to vectors of size 400, and using skip-gram as the training algorithm as we are trying to predict the category based on unique words, which may be infrequent. Then, the resulting word embedding is saved to a text file.

First, the training and validation data was split with a ratio of 0.7 and 0.15 respectively. The remaining 15% of the data was kept as testing data. Then the training data was split into batches and then transformed into numpy arrays.

## Model

We used LSTM (Long-Short Term Memory) to ignore and forget repeated words to give them less weightage, with a Bidirectional layer wrapper. Then, an Average Pooling 1D layer was added to control overfitting, which then led to another Bidireactional LSTM layer. After that, the subsequent layers were normal dense layers with the activation function set to Leaky Rectified Linear Unit, with a Dropout layer to control overfitting.

# Results analysis

When testing using stopwords, there was a decrease in the accuracy when stopwords are removed and not removed. Hence, we chose not to include stopwords for our machine learning model. In addition, we performed the worst in the various ‘others’ category, which affected the accuracy of our model significantly, it was very hard for the machine to tell the difference between others and other face cosmetics, as the tag does not reveal much features. We initially wanted to implement a two-stage classification, where one model is used to classify them into the three supercategories of fashion, mobile, and beauty, and another model for each category to classify the results of the previous round of classification, but unfortunately we did not have to time to do so.