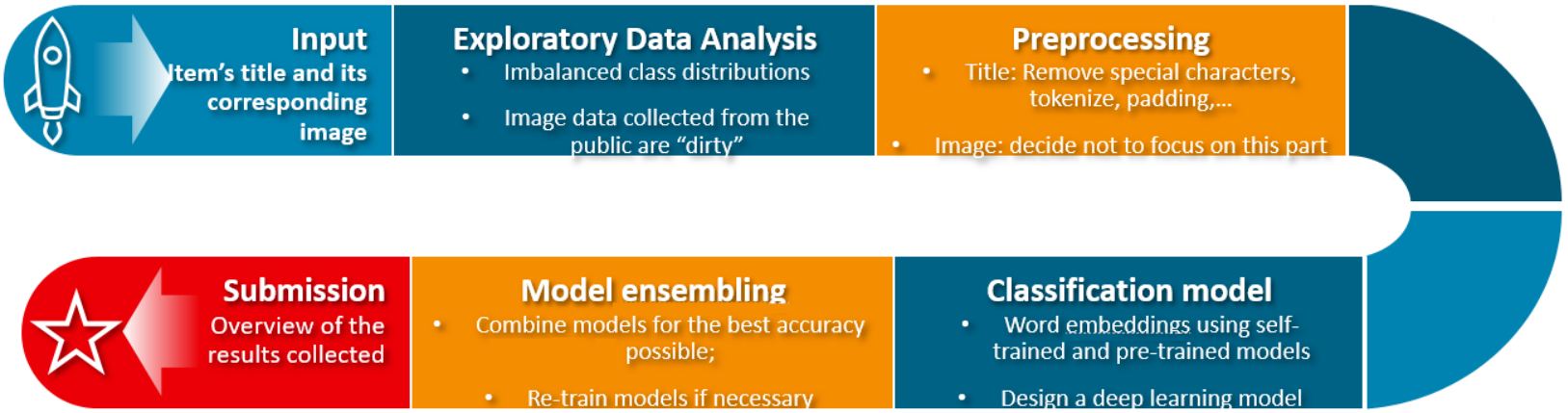
Supplementary Material for NDSC 2019

Group We S/U A-

Algorithm Pipeline



Exploratory Data Analysis

We found that the data is very imbalanced with around 25% of the categories fall below 1000 data points. The data also includes large amount of non-English words (in Bahasa Indonesia), especially for fashion products.

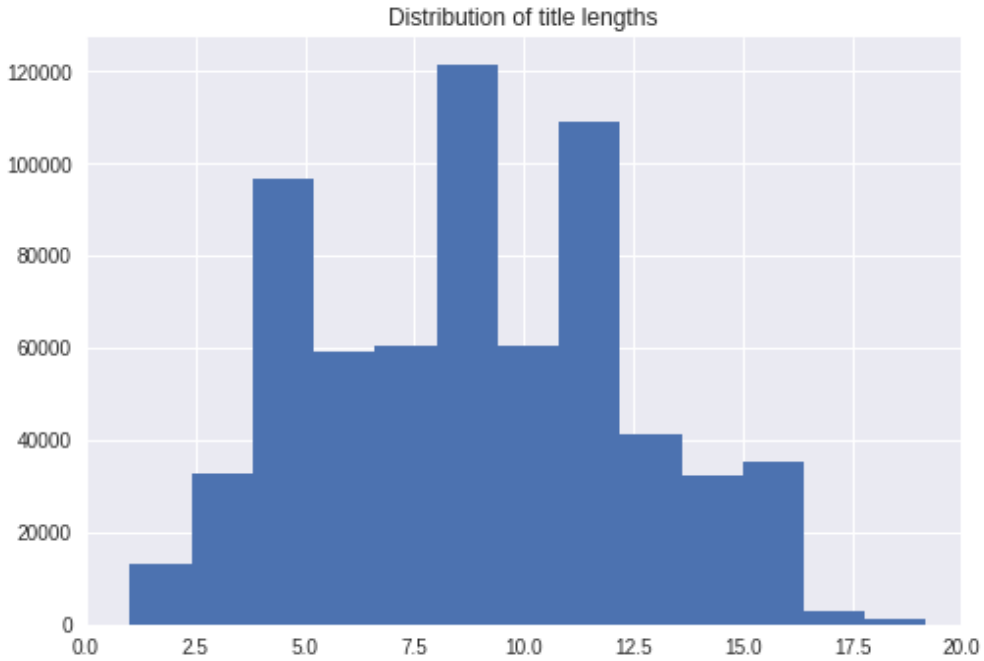
Very imbalanced class distributions in training data

```
185717    promo murah holika milky sweet peko hard cover...
214477          maybelline fresh mate bb cushion 03 natural
375359               dress pesta chiffon hitam h&m original
59739     bestseller kos0922 make up concealer liquid fo...
135175                      blush on cushion bioaqua koreak
200541       ori ponds men oil control face moisturizer 20ml
124132                                      bedak aprilskin
516288                    samsung galaxy s7 flat 32gb gold
639363    new model oppo a37 juni 2016gratis mmc 4 gb pa...
157776    er ar catrice all matt plus shine control powd...
```

Item titles not in English are commonly seen

Preprocessing

For data cleaning, we remove some special characters, words with length 1 and words containing numbers at the beginning. The text is then tokenized and padded into sequences of length 20.

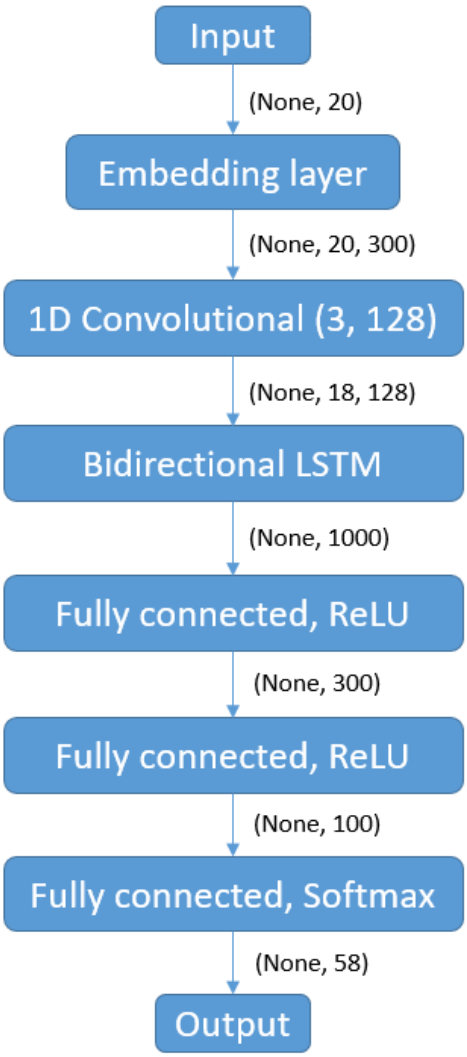


Classification model

We use the popular Python framework Keras for our Deep Learning model. For the weight of embedding layer, we use a Word2Vec model which is separately trained on the both train and test text data together with some other pre-trained word embeddings as below:

- GoogleNews-vectors-negative300
- glove.840B.300d
- paragram_300_sl999
- wiki-news-300d-1M

Other layers are kept the same, drop out is used in embedding and fully connected layers to prevent overfitting.



The model which uses custom Word2Vec embedding on the dataset gives the best result with test accuracy 0.75458 on the public leaderboard.

Observations:

- Overall pretrained embeddings seem to give worse results compared to non-pretrained model.
- The performance of the different pretrained embeddings are almost similar.

Besides the above model, we also try the latest state-of-the-art model Bidirectional Encoder Representations from Transformers (BERT), https://arxiv.org/abs/1810.04805. This gives us 0.75127 on the public leaderboard.

Model ensembling
To improve the accuracy, we use combine models that are trained with different word embeddings by calculating weighted sum of different models. The best combination for us is **9:20:24:12:0.81:0.71** for Word2Vec embedding, paragram_300_sl999 embedding, wiki-news-300d-1M embedding, BERT model, glove.840B.300d embedding, and GoogleNews-vectors-negative300 embedding, respectively. This gave us an accuracy score of 0.76589 on the public leaderboard.

Conclusion
- Even though the accuracy of the models with pre-trained embeddings are worse than our custom Word2Vec embedding, there is a good chance that they might capture different type of information from the data. And they do as we can see with the improved accuracy.
- It is hard to achieve a high accuracy because of misclassifications in both the training data and test data.

Other techniques that we did
We also applied these techniques but they didn't help to improve the accuracy in the public leaderboard.

- Oversampling: to deal with the unbalanced contribution of categories, we oversample classes that have fewer data (class 5x, 40, 48, 49, and many others). However, this led to overfitting.
- Item name hashing. A creative way to predict labels. We maintain a HashSet for the categories and some popular brand names, {‘Samsung’, ‘Apple’, ‘BB & CC cream’, ‘lip gloss’, ‘gloss’, ‘tint’, ‘lip tint’}. For example, if we see that there is a brand name like “Samsung” in the title, we just predict the item as Samsung. If we see words like “lipgloss”, “gloss”, “tint”, “liptint”, we immediately predict it as the given label. This should work well for correctly-labeled data; however, the test data (and training data) have many misclassifications, and by doing this we cannot predict the correct “incorrect” category that the test data wanted :).
- Other simple models like kNN, SVM. The validation accuracy is below 40%.
- Google Translate API: to translate Bahasa Indonesia item titles. However, Google limits the number of translations to about 1000 at a time, hence we cannot use it for our model.