EXPERIMENT DOCUMENTATION

RESTAURANT SENTIMENT ANALYSIS

Author : Randy Zakya Suchrady

Preprocessing

In this experiment, the preprocessing steps are as follows:

1. Case Folding
   Case Folding is a common preprocessing step, the reason behind it is to make words with different case becomes similar e.g. “Hello” and “hello” have the same meaning. The advantage of using this method is making the dataset vocabulary richer (the frequency of certain word becomes bigger), but the disadvantage is, some text is written in a specific cases that resemble sentiment e.g. “kamu” and “KAMU” where the first word probably resembles neutral or positive sentiment whereas the second one may resemble negative sentiment. In this experiment, I will assume that cases don’t say much about sentiment.

2. Remove Non Alphabet and Non Space
   This is also a common preprocessing step, the reason behind it is to reduce the vocabulary dimension. The advantage is explained in the reason. The disadvantage is there may be some characters like punctuation that shows sentiment e.g. “kenapa” and “kenapa???” where the second one is likely to show disappointment or negative sentiment. In this experiment, I will assume that non alphabets and non space don’t say much about sentiment.

3. Remove Multiple Consecutive Character
   This preprocessing step is used because in Indonesia, many people write down a word by exaggerating their text. By removing this, it allows the vocabulary frequency table to become bigger. The disadvantage, similar to previous steps, multiple consecutive character may polarize into a sentiment e.g. “kenapa” and “kenapaaaa” where the second one likely to show disappointment or negative sentiment. In this experiment, I will assume that multiple consecutive characters don’t say much about sentiment.

4. Normalization
   In Indonesia, it is common for people (especially in social media) to write down in a non formal way and this causes problems because the vocabulary dimension may become bigger. To tackle it, I use a public dataset to clean some non formal words, this may reduce the dimension of the vocabulary. The disadvantage is also the same as the previous steps, this step may remove non formal words that resemble different sentiment with their original formal form. In this experiment, I will assume that non formal words don’t say much about sentiment.

After the preprocess steps, the dataset is vectorized using Tfidf Vectorizer. I choose Tfidf Vectorizer because I assume some words appear more than other words for certain sentiment, so Tfidf Vectorizer may become a solution.

There are other preprocessing steps such as stemming, removing stop words, and lemmatize. Unfortunately, those steps didn't make significant improvement.

Model Selection

In the modeling step, I choose the best model among 5 models:

1. Complement Naive Bayes Classifier
   The reason for using this model is to test the dataset whether one word with another has a relationship. If the model performance is good, the dataset may have an independent relationship from one word to another in the text. This model is also fast in the training step and also very light in size.

2. XGBoost Classifier
   The reason for using this model is because this model is one of the best ensemble model in machine learning. This type of model also has a more elegant way to train the model than random forest.

3. RandomForest Classifier
   The reason for using this model is because random forest is one of the most common ensemble model. By creating many trees, random forest can tackle overfitting that may appear in decision tree.

4. SVM Classifier
   This model has an elegant way to train. The SVM Classifier works best for binary classification because the SVM in default is a model to solve binary classification problems. This model has a fancy way of interpreting data in n dimensional space.

5. Recurrent Neural Network
   This model is the popular model for NLP problems. This model tackles the problem in a standard neural network because this model solves sequential data properly by using the gates in the units. There are multiple types of RNN tested in this experiment : LSTM, GRU, Bidirectional LSTM, Bidirectional GRU

Performance Metrics

From the 5 model, I use cross validation with 5 fold using accuracy and f1-score metrics. The result shows SVM has a higher score than other types of model in the default hyperparameter (without hyperparameter tuning). The reason behind choosing accuracy is because accuracy is

the common way to interpret a classification problem. The reason in using f1-score is because f1-score measures better in a certain task or for an imbalance dataset. In this experiment for analyzing sentiment, there may be a certain objective where some sentiment is more valuable than the other for example if the model is use for gathering insights from customers from the restaurant, so the restaurant owner may know if some feedbacks may be worth for evaluate his or her business (in this experiment is restaurant).

Model Performance Analysis

Validation

By traversing the traditional (non deep learning) models and the type of preprocessing, results in SVM with the validation accuracy score 85.2% and the validation f1-score 89.8%. On the other hand, the RNN model has a 81.96% accuracy and 86.61%. The SVM then becomes the chosen model for hyperparameter tuning.
In the hyperparameter tuning phase, the f1-score becomes 90.41% by using the following hyperparameter:

1. C : 1
2. Gamma : 1
3. Kernel : RBF
4. Class weights = {negative: 1.5344827586206897, positive: 0.7416666666666667}}

Testing

In the test phase, the model shows a good performance as follows:
1. Accuracy score : 89.18%
2. F1 Score Binary : 91.8%
3. F1 Score Micro : 89.18%
4. F1 Score Macro : 87.96%

Different f1-score average is used to evaluate the model for different task that must the model solved. Must be noted that the dataset that is used for the training phase is imbalance with 1200 positive sentiments and 580 negative sentiments.