Here we represent our feature extractor and classical classifiers

Available Classifiers

- Naive Bayes
- Logisitic Regression
- XGBoost

To Train

1. Download [Quora Question Pairs Dataset].

2. Place it into `dataset` directory in the parent directory.

3. Install imblearn.

```
pip install imblearn
```

4. extract features from data

```
python main.py --do_data
```

5. Train the classifier

```
python main.py -classifier <classifier version (naive_bayes,
logistic_regression, xgboost)>
```

e.g. to train `naive_bayes` classifier

```
python main.py -classifier naive_bayes
```

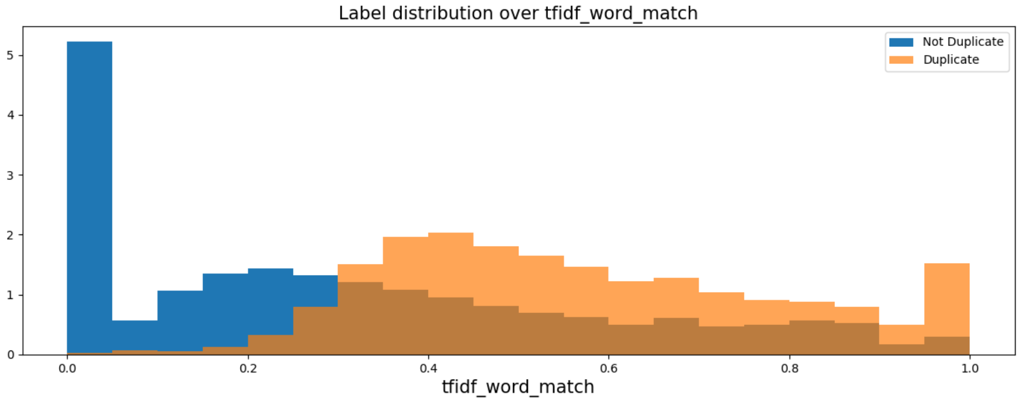
Feature Extractor Details

Set of Features

1. **TfIdf on shared words** For each question pair,

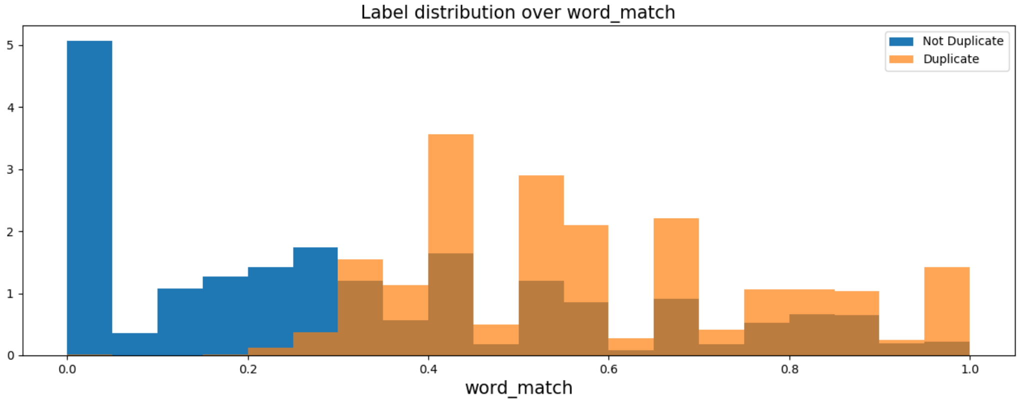
- Stop words are removed.
- Shared words are extracted.
- TfIdf feature is extracted from the shared words only.

- This allows not to be biased to the common words that are more likely to be shared, because of their commonalities.



2. **Word Match Share** For each question pair,

- Stop words are removed.
- Shared words are extracted.
- Ratio of shared words is calculated, `R = (2 * number_of_shared_words) / (number_of_words_in_question1 + number_of_words_in_question2)`.



3. **Jaccard** For each question pair,

- Stop words are removed.
- Shared words are extracted (intersection).
- All set of words in both questions are extracted (union).
- Ratio of shared words is calculated, `R = number_of_shared_words / number_of_union_words`. image

4. **Word Count Difference** For each question pair, absolute difference between number of words in questions is calculated. image

5. **Word Count Ratio** For each question pair,

- Word Count of both questions are calculated.
- Ratio of Counts is calculated, `R = min_word_count / max_word_count`. image

6. **Unique Word Count Difference** For each question pair, absolute difference between number of unique words in questions is calculated. image

7. **Unique Word Count Difference without Stop words** For each question pair,

- Stop words are removed.
- Absolute difference between number of unique words in questions is calculated. image

8. **Word Match Count** For each question pair, number of shared words is calculated. image

9. **Unique Word Count** For each question pair, number of unique words from both questions is calculated. image

10. **Unique Word Count Ratio** For each question pair,

- Unique Word Count of both questions are calculated.
- Ratio of Counts is calculated, `R = min_word_count / max_word_count`. image

11. **Unique Word Count Ratio without Stop words** For each question pair,

- Stop words are removed.
- Unique Word Count of both questions are calculated.
- Ratio of Counts is calculated, `R = min_word_count / max_word_count`. image

12. **Same Start Word** For each question pair, check whether both questions start with the same word or not. image

13. **Character Count Difference** For each question pair, absolute difference between number of characters in questions is calculated. image

14. **Character Count Ratio** For each question pair,

- Character Count of both questions are calculated.
- Ratio of Counts is calculated, `R = min_character_count / max_character_count` image

Feature Selection

We applied a `Sequential Backward Selection` Approach to select the best representative features, and it ended-up with these features:

- Word Match Share
- Word Count Difference
- Word Count Difference without Stop words
- Unique Word Count Difference
- Unique Word Count Difference without Stop words
- Unique Word Count
- Unique Word Count Ratio
- Same Start Word
- Character Count Difference
- Character Count Ratio

Classifiers

We splitted the dataset as `90% training` and `10% validation`, and here are the results

- **Naive Bayes** Training Accuracy: 59% Validation Accuracy: 58.8% Validation AUC: 0.612 ROC Curve: image

- **Logistic Regression** Training Accuracy: 70.27% Validation Accuracy: 70.46% Validation AUC: 0.783 ROC Curve: image

- **XGBoost** Training Accuracy: 78.26% Validation Accuracy: 78.32% Validation AUC: 0.874 ROC Curve: image