

Assignment 2: Question Pair Identification

On websites like Quora, users can ask any question and other users will try to answer this question. If new questions are asked, it would be helpful to automatically determine whether a question with the same intent has already been asked. This will be your task for this assignment.

You are given manually labeled data, consisting of pairs of questions. Each pair is labeled with a class 0/1 whether they have the same intent or a different one. Then you should create the features for a classifier and in the second step you should use a classifier to classify the examples. You should create the feature set yourself, while you can use an existing classifier from sklearn for the classification.

Use the training data to train your model and report the scores on the test data. If you need to try different hyper-parameter, use the validation set to select the best ones.

Framework

The basic framework already exists. You find the code in a git repository. Clone the code by:

```
git clone https://jniehues@bitbucket.org/jniehues/nlp2021.git
```

In addition, you will need the libraries sklearn and scipy

The data is in asgmt2/data

You have three sets, train.*, test.* and valid.*. Each set contains 3 files. They always have one example per line.

\$set.q1.txt: The first question of a question pair

\$set.q2.txt: The second question of a question pair

\$set.class.txt: Class for this question pair

You find the code in asgmt2/src

Files:

- train.py: The main file for training and evaluating the classifier
- FeatureCreator.py: Template of how to create features. One method will take the training data and create all possible features. The other will create the features for a specific example
-

Tasks:

1. In FeatureCreator, a single BoW uni-gram feature set is implemented. Extend this implementation in two ways:
 - a. Implement a FeatureCreator for individual BoW for each sentence. Then the features for the classifier are the union of both feature set.

- b. Implement a filter on the word features based on the frequency in the training data. Once take only take the most frequent words and in a second experiment ignore the frequent words
2. Train a classifier on your different features and evaluate them on the test data. (To faster get some initial results, it might be helpful to do also some experiments where you use the validation set as your training set.)
 - a. Use a Naïve Bayes classifier (MultinomialNB())
 - b. Use a logistic regression (LogisticRegression)
 - c. Discuss the results in the report

Try the classifiers and plot in a histogram, try different features and different parameters and compare the accuracies
3. Create a feature set that that looks at the difference between the questions instead of the question individually
 - a. It should use the following features for each word of the vocabulary
 - i. How often does the word occur both questions
 - ii. How more often does the word occur in one question than in another question
 - b. Discuss your results. Take into consideration that you are using linear models
4. Improve the results by using a different representation of the questions
 - tokenize them
 - represent the questions in another form than an array

Submission

The submission to Canvas should respect the following guidelines:

- Submit a pdf file with the report and a zip file with the code
- Write the first and last name in the beginning of the report
- Submitting the assignment as a zip file and not as rar
- When using Jupiter notebooks, add the reports also as PDF
- Name the zip file with the code and the pdf file with the report with lastname_firstname.zip / lastname_firstname.pdf