University of Lisbon

Master in Computer Science and Engineering

Question Classification NLP Project Presentation

Author:
Laura Vicente da Costa ist89633
Yu Cheng ist97282

Professor:
Prof. Luísa Coheur
Prof. Nuno João Neves Mamede

Group 68
25 of October of 2020

Contents

Chapter 1

Model Developed

For the **preprocessing** of the corpus obtained from the train and dev file, we created a function preprocess which will perform several tactics of the data processing:

1. At first, we will process with the text cleaning by removing special characters and keeping common dot words
2. Next we will remove the stop words from the corpus
3. finally we choose stemming or lemmatization (one of them, both incorporated doesn't make sense) to complement the data process.

For the **feature engineering**, we partially follow the article `"High-Performance Question Classification Using Semantic Features"` of University of Stanford.
Their strategy consists in using the informer tagger to predict an informer for a given question, and then extract a set of informer features and a set of word features from all the words in the question and then these two sets of features are then combined into single feature vector.
Although we didn't created informer features, we obtained word Features from Train Data, which would be fed to our feature vector and the possible features that we considered are Named-Entity Recognition, grams, lemmas, POS Tags and Orthographic Features using shape.
According to the article, including Named-Entity Recognition would improve the accuracy of the fine label, but after teting the model with all the combination, since we didn't include the informer features, we were not able to achieve the score stated in the article, we decide to use the combination of uni-gram+bi-gram, lemmas, POS Tags and Orthographic Features (removing NER) as it provides the best score of accuracy in our model.
Finally, the literature study over the years has shown that Linear SVM has the best performance in the question classification problem, hence we are opting this as our classifier.

Chapter 2

Resultant Accuracy

The resulting Accuracy is approximately 84% for the coarse label and 79% for the coarse-fine label which is the score we obtained for the best coarse-fine label accuracy.
This is reached by choosing stemming in the preprocessing.
And having the feature vector with the following word features: uni-gram, lemmas, POS Tags and Orthographic Features

Chapter 3

Error Analysis

First. As we proceed with the analysis of the corpus generated from the preprocessing of the text, we found out there are many difficulty to accomplish a full cleaning of the corpus, for instance: it's not efficient and practical to include all the dot word in the list of words that we will keep, the words in parenthesis should be considered as a unique token as they could be a name of a book, song, movie... and issues concerning the numbers(conversion between units, currency, years). The accuracy would improves if we could manage to clean the corpus taking into account all the factors.
Second. Our feature list has limitation and is predicting only according to the word feature that we include. For instance, there are wrong prediction that could be correct according to a certain measure. To overcome this limitation, we should adapt to the context of the corpus and consider more measure

Chapter 4

References

1.Methods and function obtained from
https://towardsdatascience.com/text-analysis-feature-engineering-with-nlp-502d6ea9225d
2.Strategy and consideration for the improvement of the accuracy from
https://nlp.stanford.edu/courses/cs224n/2010/reports/olalerew.pdf