LN2020 - Group 56

Rustam Zayanov, Mustafa Khalil

October 2020

Preprocessing, Models

First thing we have done was to remove punctuation, endline chars and extra spaces, we used regular expressions to perform this operation. Then one step is to transform the words into lowercase. The third possible step is transforming words into tokens using *word_tokenize* from nltk, and then We used StemPorter fron nltk to perform the stemming operation, and Stop words from corpus of nltk, we had to remove some words from this list because we think it is important for the classification (like where, what, ...). Some of the strategies in the following sections, required the texts to be feed as vectors, in order to do that, we use Count vectorization from sklearn, followed by TF;IDF vectorizer, note that it should be done on the train and dev datasets combined in order to normalize the resulted vectors in the same space. We tested different strategies with differnt parameter, starting from **K Nearest Neighbors** where A question will be classified with the label of the majority of the k nearest questions from the train dataset with respect to l2 distance between vectors, so the datasets will be vectorized using TF;IDF. The second strategy we tested was **Decision Trees**. A Decision tree is built over the training dataset using ID3 algorithm, The Third model is **Random Forests** which is an ensemble of many small decision trees that takes the decision using voting by Bagging, And finally we tested **Support Vector Machine** (over multiclass dataset) , which constructs an optimal separator between each class space.

Evaluation & Error Analysis

The used metric to evaluate each model is its accuracy over the dev dataset. The best results by each model are shown in the table , we concluded that Support Vector Machine is the best classifier for Coarse labels classification, and Random Forest for Fine labels classification, the two classifiers were giving similar results, but is slightly better than the other, and with big difference from the other classifiers, and the preprocessing was not very effective, as the results shows similar accuracies regardless of the applied preprocessing, which could be interpretted as the effective part of the preprocessing was done during the vectorization by TF;IDF.

Classifier	Accuracy - Coarse	Accuracy - Fine
SVM	82%	69%
Random Forest	81%	72%
Decision Tree	75%	62%
kNN	72%	61%

Table 1: Best Accuracies