Group 12: Filipe Sousa 90714 Pedro Vilela 90762

APPROACH: MLClassification

Description

Our algorithm uses a Support Vector Machine to learn a model from the TRAIN dataset and then to predict the correct classes in the DEVELOP dataset. We use unigrams, bigrams, trigrams and the first word of each question as features to input in the SVM learning algorithm.

For each type of N-gram we multiply it by a different weight, so it will have a different importance in the learning algorithm.

For pre-processing, we transform all the sentences into tokens, where one token corresponds to one word of a sentence. We also remove from the sentences, from both datasets, the punctuation and some suffixes from the tokens that improved the accuracy of our algorithm.

We convert all tokens into lowercase in order to count bigrams, trigrams and the first word in the sentence. We don't change the case to count the unigrams, as we found out it reduces the accuracy of our algorithm. We think it may be due to the unigrams' case storing important information about where in the sentence a word is in (mainly "start of the sentence" vs. "in the middle of the sentence").

To evaluate the efficiency of our model, we used accuracy and recall as evaluation metrics.

What we tried:

- First we tried an approach based on N-grams and similarity measures (the cosine similarity between the TF-IDFs of each word, considering each category as a "file"), but we gave up on it because it didn't have a good enough accuracy. The accuracy of this attempt was 73.080967% for Coarse granularity, and 50.262881% for Fine;
- With the SVM approach, we tried using larger N-grams (4+), and pairs of words not directly adjacent, but the results (accuracy, recall) were worse than our final approach;
- We also tried different combinations of the weights for each N-gram and we found out the combination that yielded the best accuracy for the DEVELOP dataset was the following:

Granularity \ N-gram type	Unigrams	Bigrams	Trigrams	First word in sentence
Coarse	2	1.15	0.22	2
Fine	1.5	1	0.22	1.75

… which is the combination we used in our final version.

Accuracy

The accuracy of our final model is 87.171399% for Coarse granularity, and 82.649842% for Fine.

Error analysis

The recall for each one of the Coarse categories was very similar except for "ABBR". We think one of the main sources for this error was the fact that this class didn't have enough examples to train the model in comparison with the others (59 train examples out of the 4001 in total); our recall for this class was 44.444444% whilst for our second worst class (ENTY) it was 81.531532%.

For each of the Fine categories, the recall was much more variable than in the Coarse case. We had some categories with 0% recall, others with 100% and others in-between. We consider a source for these errors was, besides the one we described in the Coarse case, the fact that some classes have few common words among their examples. For instance, the class "ENTY:veh" has many different vehicle names to memorize in only 20 examples (motorcycle, aircraft, gunship, boat, yacht, among others), and we had a low score of 25% recall in this class.