Classification of semantic relationship between pair of nominals using machine learning algorithms.

Team name: DataBox

Team Members:-Amit Vyas
-Dhanraj Kotian
-Prajwal Nayak
-Ankith Prabhu

Solution Description:

The problem statement given to us is to build a classifier model which when given a sentence with two highlighted words as the input, outputs of which predefined category, the sentence belong to. The language we have chosen is Python, to implement the library scikit-learn with which we were comfortable.

- Initially we started off by pre-processing the raw data by removing the tags and the indexes in the sentences, finally getting it in a CSV file format.
- Next, we tried to get the bag of words of the training data and hence the feature vectors which will help to train the classifier.
- Then, with the pre-processed test data as input to the classifier, we implemented various algorithms like random forest, SVM, Logistical Regression, Naive Bayes etc. out of which Logistical Regression gave the highest accuracy of 63.38% (58.85% considering the direction).

How to run the files:

The system running our code should have python 2.7 along with the following libraries:

- numpy
- pandas

- nltk
- Beautiful Soup
- sklearn
- re (comes with python)

These packages/libraries can be installed through pip.

All the source code is present in the submitted folder. We have two kinds of results, one which predicts without considering the direction (the order of e1 and e2 tags) and the one which considers the direction (optional). The files have been named accordingly. The files PreProcessTrain and TestFileGeneration preprocess the data. Next the Train file ({train with no direction},Train_with_no_direction.py file) trains the classifier by taking input from the pre-processed train file. Next, it takes the test data, classifies it and calculates the number of matches with the actual output, giving its accuracy,confusion matrix and the f1_score. The direction oriented train file also work in the same way(only difference is that it also predicts the order) and is optional. Also the output(prediction) of the code is written in the CSV file named Category_Output along with the actual output to facilitate comparison.

The Classification of categories with direction is also present and its optional.

An alternate approach was to use different ML algorithms out of which Logistic Regression gave the best option. The other ML algorithms in the test file is commented. Our other approach was to use cosine similarity to classify sentences as can be seen in the alternate approach folder.

It is advised to first run the *PreProcess*, *Test Generation* and *Test Keys* python files first before running the training file. All the python files can be run through the terminal.

Thank you.