Final Problem Resolution

When we were provided with the problem statement, we unanimously decided to apply a Recurrent Neural Network (using LSTMs) to solve the problem. A Recurrent Neural Network was our first choice because it takes into account the sequence of the words while training.

We started out by using the word2vec architecture for passing the corresponding integer value of each word of the corpora into the RNN model. We were able to classify the data across 2 categories, but were not being able to do it for 10 labels. We switched between different approaches and frameworks (TensorFlow, Keras), but to no avail.

Finally, we decided to adopt the Machine Learning technique-SVM. We implemented the SVM algorithm using Scikit-Learn.The SVM classfier ran on top of the sklearn pipeline and we used TF-IDF and count vectorization.We used GridSearchCV on a Multinomial Naive Bayes classifier to fine tune the parameters for the TF-IDF function.We ran through several trial runs using various error functions and optimizers for the given data.

Later, when we were going through several blog posts regarding comparison of performances (between SVM and RNNs), we stumbled upon the fact that Deep Learning techniques are only as good as the vastness of the provided dataset, i.e, it performs well only when size of dataset is huge. The given dataset doesn’t seem to be big enough for giving substantial results using Deep Learning techniques.

We reached a final accuracy of 63% with our present architecture, and would love your suggestions on improving this score.

Dependencies:

1. numpy

2. scikit-learn

3. scipy

Instructions to run the project:

Download all dependencies and then run the Jupyter Notebook.