**PROJECT -2**

**Group Ipsy:**

* Sri Chaitra Kareddy (sxk180037)
* Sukruti Hosudurg (sxp180014)

**Problem Statement:**

The motive of this project is to train a model to give the similarity score for two chunks of text are. The similarity score takes an integer value between 1 and 5 (included). The higher the score, the more similar the two chunks are.

**Proposed Solution:**

The solution that we came up to is building a model using the bagging technique on base estimator support vector machine (SVM). Two features were given to this model as input per data point. Here is a description for the features:

Feature 1:

For every token in a sentence, it’s hyponyms, hypernyms and synonyms are considered, and this set is considered to be an enriched version of that sentence.

For two enriched sets of two different sentences the Jaccard similarity is then used to get a value of similarity score for this particular feature.

Feature 2:

For every token in a sentence, it’s hyponyms, hypernyms and synonyms are considered, and this set is considered to be an enriched version of that sentence similar to Feature 1.

These enriched sets of two sentences are passed to scikit-learn’s implementation of the tf-idf vectorizers. This gives us the vector representation of the enriched sets. Cosine similarity is then used on these vectors to get a value of the similarity score for this particular feature.

The model works on these two features of the data and then gives us a similarity score between 1-5 for the given two sentences tested.

**Description of the Similarity Measures used:**

* Jaccard Similarity:

The Jaccard Index, also known as the Jaccard similarity coefficient, is a statistic used in understanding the similarities between sample sets. The measurement emphasizes similarity between finite sample sets and is formally defined as the size of the intersection divided by the size of the union of the sample sets. The mathematical representation of the index is written as:

* A close up of a logo

  Description automatically generated
* Cosine Similarity:

The cosine similarity between two vectors (or two documents on the Vector Space) is a measure that calculates the cosine of the angle between them. This metric is a measurement of orientation and not magnitude, it can be seen as a comparison between documents on a normalized space because we’re not taking into the consideration only the magnitude of each word count (tf-idf) of each document, but the angle between the documents.

**Description on the machine learning model/technique used:**

Bagging (Ensemble):

Bootstrap Aggregation (or Bagging for short), is a simple and very powerful ensemble method. Bagging is the application of the Bootstrap procedure to a high-variance machine learning algorithm, typically decision trees.

Support Vector Machine (SVM):

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

**Full Implementation Details:**

Programming tools/Third Party software:

* NLP Libraries/tools: Spacy,nltk.

1. ML sources: scikit-learn
2. Programming language used: python.

**Result and Error Analysis:**

The accuracy value obtained by our model: 0.3367(approx. 34%)

Based on our analysis, since we haven’t considered a lot of semantic, syntactic features the error could have been this high. As we suggested in the potential improvements incorporating the usage of the syntactic information and semantic information.

**Problems Encountered/Challenges faced:**

We faced multiple problems while completing the project that includes technical difficulties while trying to use pre-existing NLP resources and adding them to our system. Apart from these technical issues, the primary issues of our concern include:

* Trying to find a good similarity measure by going through several resources available online.
* Finalizing on the machine learning model to pick by implementing several different ones and checking the similarity score was an equally difficult challenge.

**Pending Issues:**

We have several issues pending such as giving similarity for a bag of words using the functions that we wrote in task2.py using Jaccard and cosine similarity measures. It was taking a very long time to run using all the functionalities in task 2, hence we eliminated the thought of running the code using the functions for finding hyponyms, hypernyms, meronyms and holonyms.

**Potential Improvements:**

* We haven’t fully utilized the potential of the syntactic parsing/dependency parsing available and there is quite a lot of scope for improvement in that area.
* We also haven’t utilized the POS (Parts of Speech) tags for the words available which could have improved the stature of the model.
* All in all, there is scope for potential improvement on the semantic end as well as the syntactic end.