NLP Project Report

Team Pandas

Jiten Girdhar- jxg170021
Divya Sharma- dxs180031

PROBLEM DESCRIPTION:

To design and implement a model that determines how similar 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. This problem is also known as the Semantic textual Similarity problem.
As a reference, here are some examples:

Sentence 1: Birdie is washing itself in the water basin.
Sentence 2: The bird is bathing in the sink.
Score: 4
Comment: Both sentences convey the message that a bird is taking a bath.

Sentence 1: The young lady enjoys listening to the guitar.
Sentence 2: The young lady enjoys playing the guitar.
Score: 2
Comment: Both sentences involve a lady and a guitar, but convey different actions i.e. listening to the guitar and playing the guitar respectively.

PROPOSED SOLUTION:

For this we use Machine Learning based approach to solve the problem of STS. We have extracted the following features from the data :

1. Number of Common words(bag of words):
   We tokenized sentences followed by lemmatization of their tokens and removal of stop words. Then we kept a count of the number of words which were common in both sentences.

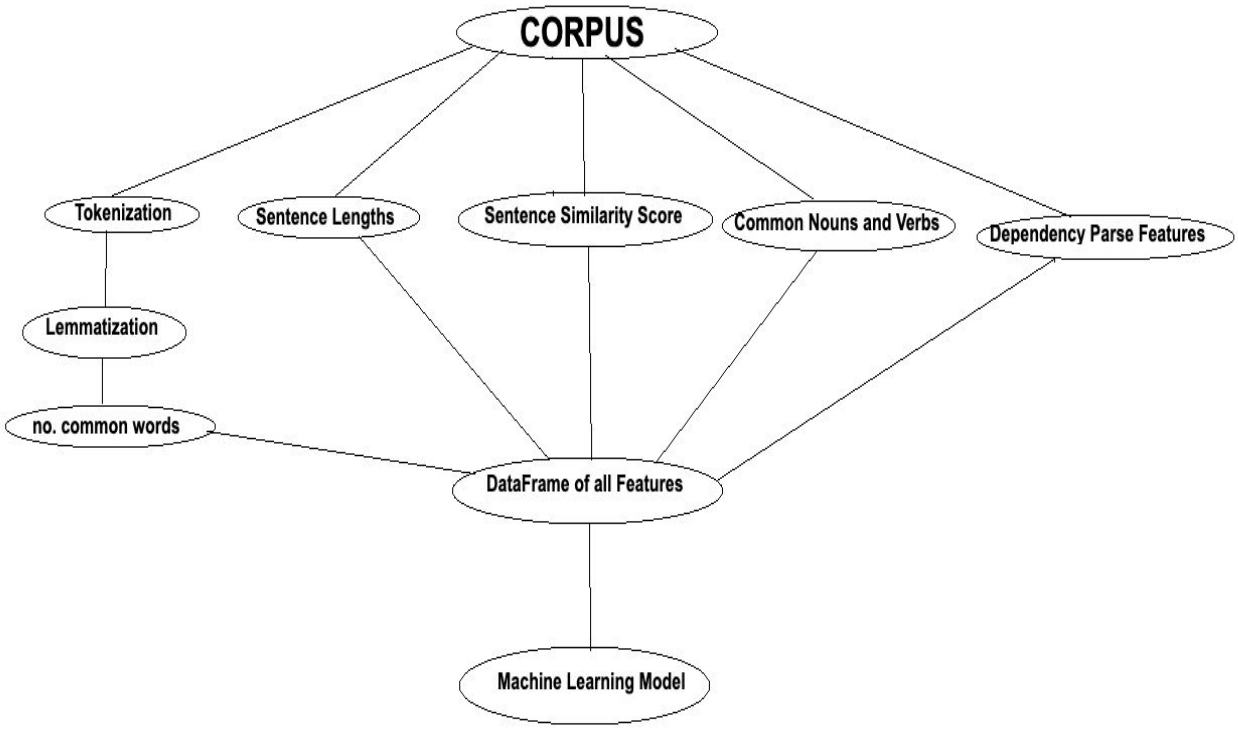
2. Lengths of sentences: The lengths of left and right sentences seem to have substantial amount of feature importance. So we used them as 2 seperate features.
3. Sentence similarity based on the words used:
   We obtained similarity between each word of sentence1 with each word of sentence2 and recorded the maximum similarity between them and averaged them out.
4. Nouns and Verb similarity across both sentences:
   We found all the Nouns and verbs in both the sentences and derived their similarity scores using the wup_similarity function in NLTK. We then average the similarities for all nouns and verbs. We used this parameter because it considers the depths of the two synsets in the WordNet taxonomies, along with the depth of the LCS (Least Common Subsumer)
5. Dependency Parse Features :
   We derived the Dependency parse trees using Spacy. We then filtered out the subject, object and root for each sentence and then match the corresponding feature for the left and the right sentence. This led us to 6 new features subject/object/root similarity check and subject/object/root similarity score.

We use an ensemble of Random Forest models trained on the given dataset for this problem.

TOOLS USED:

For this project we have used NLTK library extensively to extract various language features.
We used Spacy for dependency parse features.
Sklearn for machine learning models and model statistics
Numpy and Pandas to transfer and/or collect data across the feature extraction phases.

ARCHITECTURE DIAGRAM:



RESULTS AND ERROR ANALYSIS:

1. We are able to classify the class with label 1 relatively distinctly. We can verify this assumption because we got better precision for class 1 samples considering that their total number is itself very small (34) :
   1.0 0.44 0.32 0.37 34 (class, prec, recall, F1, total)
2. The model is definitely having trouble in classifying the class with label 2. This is evident when we compare the results with class 1 results. We think that there is enough data to label samples as 1 but not enough for the model to learn the difference between 1 and 2.
   2.0 0.28 0.07 0.12 95 (class, prec, recall, F1, total)

3. This is a recurrent theme in our model that we can classify with good precision vs count for classes 1 and 5 (extremes) but the rest of them need more data to differentiate between them.

	precision	recall	f1-score	support
1.0	0.44	0.32	0.37	34
2.0	0.28	0.07	0.12	95
3.0	0.35	0.33	0.34	326
4.0	0.36	0.58	0.44	358
5.0	0.69	0.45	0.55	396
accuracy			0.43	1209
macro avg	0.42	0.35	0.36	1209
weighted avg	0.46	0.43	0.42	1209

Correctly classified examples to elaborate:

s_532: (For score 1)

The technology-laced Nasdaq Composite Index <.IXIC> jumped 26 points, or 1.78 percent, to 1,516.
The broad Standard & Poor's 500 Index .SPX was up 8.79 points, or 0.96 percent, at 929.06.

s_518: (For score 2)

Shares of Coke closed New York Stock Exchange trading Thursday at $44.01.
In morning trading on the New York Stock Exchange, Coca-Cola shares were down 34 cents at $43.67.

s_171: (For score 3)

The European Union has got to do something and do it quickly.
The European Union itself and do so quickly.

s_15: (For score 4)

I am one of those Members who attends sittings quite faithfully.
I belong to the Members who are present rather honourably.

s_4: (For score 5)

The vote will take place today at 5.30 p.m.
The vote will take place at 5.30pm

4. Both Decision Tree and Random Forest seemed to perform well but the bootstrapped sampling of the random forest technique helped us to improve the model. The sampling of features also helped decrease variance. All other models ignored one class or the other:

Naïve Bayes Classifier:

	precision	recall	f1-score	support
1.0	0.05	0.12	0.07	34
2.0	0.18	0.26	0.21	95
3.0	0.37	0.10	0.16	326
4.0	0.35	0.50	0.41	358
5.0	0.55	0.55	0.55	396
accuracy			0.38	1209
macro avg	0.30	0.31	0.28	1209
weighted avg	0.40	0.38	0.36	1209

Logistic regression Classifier :

	precision	recall	f1-score	support
1.0	0.00	0.00	0.00	34
2.0	0.00	0.00	0.00	95
3.0	0.26	0.06	0.10	326
4.0	0.34	0.77	0.47	358
5.0	0.60	0.45	0.51	396
accuracy			0.39	1209
macro avg	0.20	0.21	0.18	1209
weighted avg	0.37	0.39	0.33	1209

Due to these results we decided to use only random forest classifier for this project, throughout the report it can be assumed that 'model' refers to a random forest classifier.

5. The dataset given to us represents an unbalanced classification problem, hence we had to take certain steps to address this issue.

6. As we can see the number of 1 and 2 classes is very small. But probably due to the features our classifier can recognise 1. So, in order to increase some precision we took a subsampling approach. In this we clubbed the classes 2 and 3 into 1 class(0) and trained a classifier on this. Next we clubbed the samples of 2 and 3 and trained another classifier on this data(this data now has classes 0->(2 and 3) 1, 4 and 5). Next whenever the classifier 2 gave 0 as an output we used the output of classifier 1 and in this way we got these results :

	precision	recall	f1-score	support
1.0	0.54	0.21	0.30	34
2.0	0.19	0.14	0.16	95
3.0	0.32	0.45	0.38	326
4.0	0.39	0.50	0.44	358
5.0	0.76	0.41	0.54	396
accuracy			0.42	1209
macro avg	0.44	0.34	0.36	1209
weighted avg	0.48	0.42	0.43	1209

This approach gave us a pearson coefficient score of 0.347

PROBLEMS FACED:

1. WordNet takes a different set of POS Tags than the ones generated by nltk. So, we wrote the code to change the nltk POS tags into those provided by WordNet to be able to use the functions provided by WordNet. Moreover, not all words are there in WordNet, so, we kept a check on that as well.
2. The dataset was divided among five classes which made it difficult for the machine learning model to learn. The number of datapoints for classes 1 and 2 were not many and also not in proportion with number of datapoints of other classes.

3. Because of the problem listed above, many machine learning models like Naive Bayes, Logistic Regression etc. could not perform well. Therefore, we made use of Tree based classifiers like Random Forest algorithm which gave us better results.
4. S_7: {'u.s.': 'JJ', 'prosecutors': 'NNS', 'have': 'VBP', 'arrested': 'VBN', 'more': 'JJR', 'than': 'IN', '130': 'CD', 'individuals': 'NNS', 'and': 'CC', 'seized': 'VBN', '$': '$', '17': 'CD', 'million': 'CD', 'in': 'IN', 'a': 'DT', 'continuing': 'VBG', 'crackdown': 'NN', 'on': 'IN', 'internet': 'NN', 'fraud': 'NN', 'abuse': 'NN', '.': '.'}

   S_7: {'more': 'JJR', 'than': 'IN', '130': 'CD', 'people': 'NNS', 'have': 'VBP', 'been': 'VBN', 'arrested': 'VBN', 'and': 'CC', '$': '$', '17': 'CD', 'million': 'CD', 'worth': 'NN', 'of': 'IN', 'property': 'NN', 'seized': 'VBN', 'in': 'IN', 'an': 'DT', 'internet': 'NN', 'fraud': 'NN', 'sweep': 'NN', 'announced': 'VBD', 'friday': 'NN', 'by': 'IN', 'three': 'CD', 'u.s.': 'JJ', 'government': 'NN', 'agencies': 'NNS', '.': '.'}

   This is the tokenization by NLTK.

   We used Spacy for dependency parsing and it seemed to split the sentences in a different way. The Spacy seems to be taking the ‘u.s.’ as ‘u.s’ which leads to keyerrors in python.

5. We were not able to incorporate word sense disambiguation in our sentence similarity function. This also happened for the nouns and verbs similarity.
6. Complications regarding dependency Parse Trees, the errors in Spacy and NLTK models which generate parse trees seep into our model.

PENDING ISSUES:

1. We used Spacy for dependency parsing and it seemed to split the sentences in a different way. Spacy seems to be taking the ‘u.s.’ as ‘u.s’ which leads to keyerrors in python. This shows that there is a certain extent of error in the standard library models itself which seeps into our models.
2. There is need to get more clarity about how to handle the words and tags which are not there in WordNet.
3. Lack of enough data- Our model is able to classify the samples to extreme classes in a better way which goes on to show that we need more data to show the model how to differentiate between 1 and 2, 2 and 3 , 3 and 4. With enough data, we can also try Deep learning approaches to this problem.

POTENTIAL IMPROVEMENTS:

1. Finding an efficient way of dealing with words not present in wordnet
2. Inclusion of prepositions in analyzing the semantic similarity of sentences because prepositions alone can cause semantic dissimilarity between sentences to a certain extent.
3. Getting more training data preferably proportionate to the number of classes.