Text Semantic Similarity - Group 3

Machine Learning Project
Aneri Sheth (1401072)
Himanshu Budhia (1401039)
Raj Shah (1401050)
Twinkle Vaghela (1401106)

***Abstract*—Machine Learning has found its place in the technological world rapidly since the past few years. Text Semantic Similarity is one of the examples of Machine Learning with various applications in natural language processing,information retrieval and digital education. Text Semantic Similarity is a measure of the degree of semantic equivalence between two pieces of text. In this report, importance, limitations, approaches for text semantics and applications are discussed and results for comparing textual data in two files are shown.**

***Index Terms*—Corpus-based method, Lemmatizing, Tokenizing, StopWords, Semantics Similarity, Natural Language Processing**

I. Introduction

THE fundamental challenge in natural language processing or plagiarism checking softwares is to find out the meaning of text. Semantic similarity is important for various purposes - plagiarism checking, information retrieval and enabling machines to answer questions. But for machines, it is difficult to determine the semantics similarity. Due to advancement in Machine Learning, machines are getting better at Text Semantics and various algorithms are proposed. Semantic Analysis is not about teaching the machine, it is about getting them to learn.

The idea is to take two text files and then to separate the words in the sentences. Once the words are separated, the next step will be to find clues in the contextual data based on the data already there with the machine. For example, if the word 'happy' is encountered, then all the related words within the database will be searched to best match our data. Till now, the text semantic algorithms are considered and further research is going on.

II. Literature Review

Text semantics has found its application in Information Retrieval, Plagiarism Checking, and many more. Text Semantic models in Information Retrieval are based on a bag-of-words representation of text, where large documents are simply represented as the frequency distribution of their terms. This representation disregards syntax or word order in the text and encodes only what topic the text is about.
Semantic Similarity Using Corpus-based Word Similarity and String Similarity uses the Longest Common Sequence string matching algorithm. Existing methods have focused on either large documents or individual words. This paper focuses on two short phrases or paragraphs. Another major outcome is a rich literature for Semantics, describing around 300 systems that have been evaluated over a span of the years 2012−2015. A vast majority of the best-performing systems at apply a regression algorithm that predicts similarity as a linear function of a wide array of text similarity measures. The methods include Semantic Overlap and n-gram overlap which are top-performing systems since the past few years.
There are Deep Learning Models like Recursive Neural Networks and Recurrent Neural Networks which also emerged as approached for text semantics. Recurrent Neural Network takes input as set of tokens which produces hidden vectors. Then we produce a vector whose values represent probability that the sentence pair belongs to the same category. The loss is calculated based on probability. Recurrent Neural Network is a powerful deep learning method.Recursive Neural Network takes input as a binary tree. The nodes of this tree represents the words in the sentence and we infer the binary tree using calculated parse of the sentence. The vector is calculated from the binary tree structure and the rest is similar as recurrent neural network.

III. Natural Language Processing

Natural Language processing is a wide domain covering concepts of Computer Science, Artificial Intelligence and Machine Learning. It is used to analyze text or how humans speak. One of the applications of NLP is Semantic Analysis(Understanding the meaning of text).

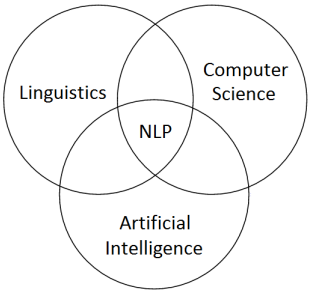

Fig.1 Overview

IV. Corpus-based Approach

This approach uses semantically annotated corpora to train Machine learning algorithms to decide which word to use in which context. Corpus-based methods are supervised learning approaches when the training data is trained by the algorithms. The corpora and the lexical resource used is WordNet.

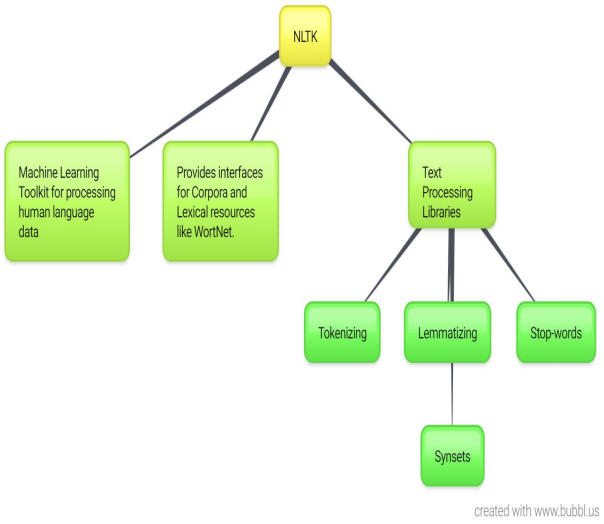

Fig.2 NLTK

Sentence 1 - A cemetery is a place where dead people's bodies or their ashes are buried.
Sentence 2 - A graveyard is an area of land, sometimes near a church, where dead people are buried.

- **Tokenizing** - Splitting sentences and words from the body of text. Tokenzing are of two types sentence tokeninzing and word tokenizing. Word tokenizing - separates every word in a sentence. Words are separated by space after the word, i.e.after every word there is a space. It counts punctuation as a separate token/word (,.!?etc) Currently, it is done only for one sentence and not implemented for sentence tokenizing yet.

```
['A', 'cemetery', 'place', 'dead',
'people', "'s", 'bodies', 'ashes',
'buried', '.']
['A', 'graveyard', 'area', 'land',
',', 'sometimes', 'near', 'church'
, ',', 'dead', 'people', 'buried',
'.']
```

Fig. 3 Tokenizing

- **Stop Words** - Sometimes, some extremely common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary entirely. These words are called stop words . The general strategy for determining a stop list is to sort the terms by collection frequency (the total number of times each term appears in the document collection), and then to take the most frequent terms, often hand-filtered for their semantic content relative to the domain of the documents being indexed, as a stop list , the members of which are then discarded during indexing. An example of stop list is shown in the figure. Using a stop list significantly reduces the number of postings that a system has to store. Stop words can be filtered from the text to be processed. There is no universal list of stop words in nlp, however the nltk module contains a list of stop words. We then remove the stop words from the sentence.

a an and are as at be by for from has he in is it its of on that the to was were will with **etc**

```
['A', 'cemetery', 'place', 'dead', 'people
', "'s", 'bodies', 'ashes', 'buried', '.']


['A', 'graveyard', 'area', 'land', ',', 's
ometimes', 'near', 'church', ',', 'dead',
'people', 'buried', '.']
```

Fig. 4 Stop Words (List and Output)

- **Lemmatizing** - The goal of lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma . Lemmas create actual words. If confronted with the token 'saw' lemmatization would attempt to return either see or saw depending on whether the use of the token was as a verb or a noun. Lemmatization commonly only collapses the different inflectional forms of a lemma. So, your root stem, meaning the word you end up with, is not something you can just look up in a dictionary, but you can look up a lemma. The only major thing to note is that lemmatize takes a part of speech parameter, "pos." If not supplied, the default is "noun." This means that an attempt will be made to find the closest noun of that word.

```
['A', 'cemetery', 'is', 'a', 'placing', '
where', 'dead', 'people', "'s", 'body', '
or', 'their', 'ash', 'are', 'buried']


['A', 'graveyard', 'is', 'an', 'area', 'o
f', 'land', ',', 'sometimes', 'near', 'a'
, 'church', ',', 'where', 'dead', 'people
', 'are', 'buried', '.']
```

Fig. 5 Lemmatizing

- **Synsets** - WordNet is a lexical database for the English language, and is part of the NLTK corpus. We can use WordNet alongside the NLTK module to find the meaning of words, synonyms, antonyms and more. Synonyms are word that have similar meaning, therefore a synonym set or synset, is a group of synonyms. The lemmas will be synonyms, and then we can use .antonyms to find the antonyms to the lemmas. Next, we can also easily use WordNet to compare the similarity of two words and their tenses, by incorporating the Wu and Palmer method for semantic relatedness.

```
word1 :  ['Area']
word2 :  ['Place']
Similarity index : 0.9333333333333333
word1 :  ['Bull']
word2 :  ['Hair']
Similarity index : 0.5333333333333333
```

Fig. 6 Synsets Example

V. Previous Results

This section includes the results and simulation that are generated in MATLAB. Text Semantics Similarity can be done either on text files (part of Natural Language Processing) or the text can be compared from images (like Optical Character Recognition).

- Comparing two text files (MATLAB Simulation) - From the output above, we see that by taking two input text files and separating the space (a delimiter) between the characters, we can compare the characters.
- Two images having textual data can also be compared. This is done using OCR (Optical Character Recognition) . This algorithm was explored for comparing text semantics from PDFs (in this case, the input will be images)

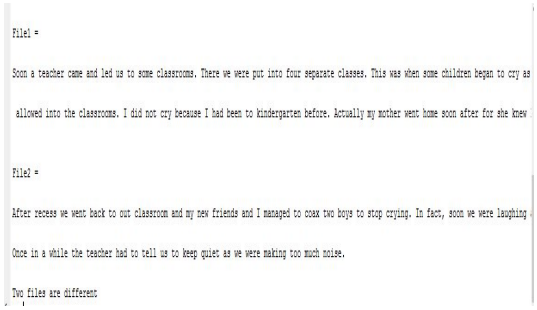

Fig. 7 Text Comparison (For different text files)

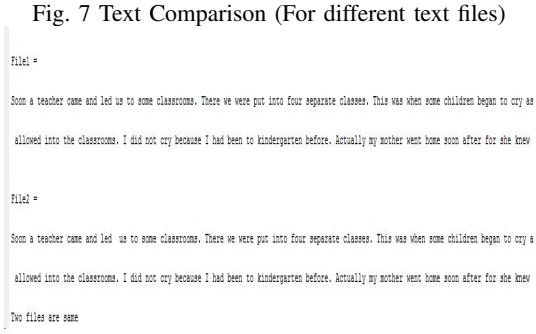

Fig.8 Text Comparison (For similar text files)

VI. Results

The previous results included comparison of text files using Optical Character Recognition (OCR). However, semantics similarity is different as we discussed in earlier sections. The results for Semantics Similarity are discussed in this section.

- Let S1 be ”I was given a card by her in the garden” and S2 be ”In the garden, she gave me a card.”
- For semantic analysis, two phrases/sentences are taken. The two sentences are similar, dissimilar or somewhat similar.
- After that, set of stopwords are defined for English language.
- After eliminating the special characters and punctuations and then removing all the stop words and lemmatizing, we get S1={I, given, card, garden} and S2={In, garden, gave, card}.
- Only 2 tokens {garden, card} in S1 exactly match tokens in S2 and so we remove those 2 words (garden and card) from both S1 and S2.
- After lemmatizing, we find the synonyms of the lemmatized words which are called synsets. Then, we compare first word of S1 with all the words of S2 and continue this iteratively and find the similarity index of each word with words in the S2.
- We find the mean of the computed similarity indexes and thus we we anaylze the semantic similarity using machine learning.
- If the similarity index is less than 0.65, the sentences are labeled as ’Not Similar’, if it is between 0.65 and 0.8, the sentences are labeled as ’Somewhat Similar’ and more than 0.8, the sentences are ’Similar’.

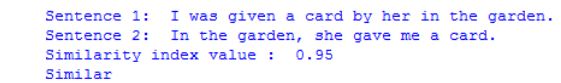

Fig.9 Similar Semantics

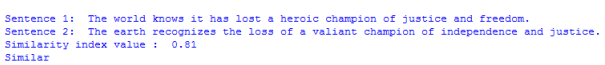

Fig.10 Similar Semantics

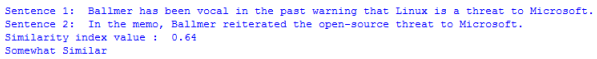

Fig.11 Somewhat Similar Semantics

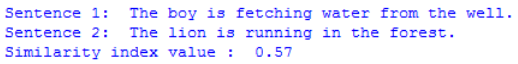

Fig.12 Dissimilar Semantics

VII. Discussion

- Semantics Similarity has been done for sentences and phrases. However, for paragraphs and short texts will need complex algorithms for separating of sentences and finding their semantics similarity.
- This is done on strings generated on the IDE. For images, we need image processing techniques along with Natural language processing.
- We find similarity word by word and thus we may get false positives and negatives.
- Our implementation does not consider spellings. To implement that, Longest Common Subsequence (LCS) Algorithm can be used.

VIII. Future Work

- We are looking forward to compare the entire text file instead of a string and find similarity between them on the basis of text semantics.
- Performing OCR on the pdfs so that it becomes easy to find plagiarism rate in the research paper and many other online stuffs.
- We would try to decrease the false positive and negative rates by using sentence- sentence similarity instead of word-word similarity.

REFERENCES

[1] T. graph, "Text semantic analysis and semantic graph", Stackoverflow.com, 2017. [Online]. Available: http://stackoverflow.com/questions/35666726/text-semantic-analysis-and-semantic-graph. [Accessed: 27- Feb- 2017].

[2] "How to do Semantic Keyword Research Using NLP and Text Analysis - AYLIEN", AYLIEN, 2017. [Online]. Available: http://blog.aylien.com/how-to-do-semantic-keyword-research-using-nlp-and/. [Accessed: 01- Mar- 2017].

[3] "Understanding Semantic Analysis (And Why This Title is Totally Meta) - Boomtrain", Boomtrain, 2017. [Online]. Available: https://boomtrain.com/understanding-semantic-analysis/. [Accessed: 01- Mar- 2017].

[4] [Online]. Available: https://cs224d.stanford.edu/reports/SanbornAdrian.pdf. [Accessed: 05- Mar- 2017].

[5] [Online]. Available: http://nlp.stanford.edu/pubs/wordwalk-textgraphs09.pdf. [Accessed: 02- Mar- 2017].

[6] "How does OCR document scanning work?", Explain that Stuff, 2017. [Online]. Available: http://www.explainthatstuff.com/how-ocr-works.html. [Accessed: 15- Feb- 2017].

[7] "Image Processing in MATLAB Tutorial 6: OCR in Natural Images", YouTube, 2017. [Online]. Available: https://www.youtube.com/watch?v=JJLDOO4Xh8Y. [Accessed: 24- Feb- 2017].

[8] "Maximally stable extremal regions", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Maximallystableextremalregions. [Accessed: 26- Feb- 2017].

[9] "Natural Language Processing With Python and NLTK p.1 Tokenizing words and Sentences", YouTube, 2017. [Online]. Available: https://youtu.be/FLZvOKSCkxY?list=PLQVvvaa0QuDf2JswnfiGkliBInZnIC4HL. [Accessed: 08- Apr- 2017].