Semantic Analysis for Similarity in Short Text

Anjali Doneria
200200056
adoneri@ncsu.edu

Anupama Kesari
200199472
akesari@ncsu.edu

Smit Doshi
200206525
stdoshi@ncsu.edu

1 Background & Introduction

Today, Google, Quora and Wikipedia are the go-to engines if you are stuck at a problem and need a solution or you are just an inquisitive person who likes to explore the different domains. Given the vast amount of users that use these platforms, there is no doubt that there is redundancy and duplicacy of questions with similar intent across the web. So, the main idea behind this project is to develop a data mining algorithm that would span across a vast set of sample questions to classify them for similarity based on semantic similarity.
One of the important use case of this classification could be recommendations of semantically similar questions/statements to user for his ease of exploration.This brings us to the tasks of analyzing the textual content on the webpages. Thus, this project undertakes comparing two given statements by performing semantic and structural analysis of the text and checking whether they are similar in meaning or not by binary classification. The inspiration for this idea is Kaggle.
A lot of research has been going on to understand short text. It has widespread application in domains like information retrieval, document clustering, duplication detection. But short text analysis has its own challenges. They do not generally follow the grammatical syntaxes as the normal sentences and hence their semantic analysis becomes harder. Two short texts can be similar either lexically or semantically. If they follow the same structure and sequence of characters, they are said to be lexically similar. If they are similar in meaning then they are said to be semantically similar. [1] compares different algorithms to find out text similarity but the challenge lies in finding the semantic similarity between the two documents. This paper only states the approaches for lexical analysis of given literature. [2] proposes an algorithm to discover the syntactic similarity between texts by using n-grams and calculating the distance between them to find their closeness. It is a good approach to similarity detection but we need to do analysis between sentences and not the characters for our purpose. To move ahead, [3] proposes an algorithm of building a co-occurence network using Probase to find the semantics of a short text. It is a really good approach to build a network words with vertices and their relatedness as edges but its application gets limited since it uses Probase which is not an open-source library for us to make use of. [4] states a novel approach to paraphrase detection. Paraphrases use synonymous words to convey similar meaning of the given sentence. Hence, this approach exploits the concept of semantic similarity between two given texts by building a similarity matrix and calculate similarity based on a formula. It makes use of WordNet library to fetch the similarity between the tokens. Our algorithm is inspired by the work done in these research papers and we are connecting dots to extend the suggested methods to find semantic similarity between two texts in simpler and easier way.
Natural Language Analysis is of three types[5]:

- Syntactic Analysis: in this stage input text is checked for syntax accuracy and structured representation of the possible parses are generated. Grammar is used to determine the syntax by generating a parse tree which is generated by using two general methods of context-free grammar and top-down parser.
- Semantic Analysis: In this stage, initial representation of the meaning of the sentence is obtained from the parsing. This meaning is generally the context-independent meaning which implies that the sentence is regarded as stand-alone disregarding any knowledge of any previous sentences.

- Contextual Analysis: In this stage, the meaning of the text is elaborated based on the contextual and world knowledge by working out references of expressions. This also deals with determining the goals underlying the utterances to formulate a more relevant reply for the particular situation.

The existing research for text classification can be broken down into the following[1]:

- String-Based Similarity: they operate on string sequences and character composition. This measures similarity or dissimilarity between two texts for approximate string comparison. Longest Common Substring (LCS), Damerau-Levenshtein, n-gram are some of the approaches for the same.
- Term-Based Similarity: they calculate similarity or dissimilarity between two given sentences based on the similarity of terms used in the sentence. Block distance, cosine similarity, jaccards coefficient are few approaches for calculating term-based similarity between the texts.
- Corpus-Based Similarity: it is a semantic similarity measure that determines the similarity between the texts by gaining information from a large corpora (which is the collection of written and spoken texts for language analysis). Latent Semantic Analysis (LSA), Hyperspace Analogue to Language (HAL) are some of the popular corpus-based similarity measures. LSA assumes that the words that are close in meaning occur in similar texts. A matrix containing word counts per paragraph is constructed from a large piece of text and a mathematical technique is used to reduce the number of columns while preserving the similarity structure among the rows. The comparison between words is then done by taking the cosine of the angle between two vectors formed by the two rows.

String-based and term-based similarity measures fail to identify similarity between sentences using different but synonymous words. Hence corpus based similarity becomes important to predict similarity between sentences using similar intent by use of synonymous words.
Traditional methods stated above calculate similarity based on the number of shared terms in articles, so if these are applied to measure similarity in short texts, some limitations may arise:

- Short texts will be converted to very high-dimensional space and lead to formation of sparse vectors resulting in inaccurate calculation.
- Information of shared terms in short texts is rare causing system to generate low semantic similarity score.
- Stopwords are ignored but in short text, they deliver information about the structure of sentences.

2 Proposed Method

The proposed algorithm aims to analyse the tokens of the two texts and decide if they are similar in meaning. Two texts could be tagged similar based on the structural and semantic similarity. We are going to make use of both the concepts to decide if the two questions in our use case are similar.

2.1 Lexical Analysis of Text to Find Syntactic Similarity

We will be using shingling to tokenize our data. K-shingles refer to all the possible consecutive substrings of length k from a given string of text of length n. For example, for text, 'earth rotates around the sun', k-shingles with k = 3 will be following,
"earth rotates around", "rotates around the", "around the sun"
Using k-shingles, we will tokenize the input documents and then create a 'relation' matrix to visualize the relationship between the generated shingles and the documents. It consists of all the possible shingles as the row and doc 1 and doc 2 as the columns. Thus, this matrix is of dimension ix2 where i is the number of possible unique shingles across both the documents. It stores the mapping of presence of the given shingles in either or both the documents as labels 0 and 1. If a shingle is present in both the documents, they are closer in structure. Using the example from [4],

- Document 1: "The sky is blue and the sun is bright"

- Document 2: “The sun in the sky is bright”
- 3-Shingles: “The sky is” , “sky is blue” , “is blue and” , “blue and the” , “and the sun” , “the sun is” , “sun is bright”, “The sun in” , “sun in the” , “in the sky” , “the sky is” , “sky is bright”

Table 1: Relation matrix

Shingles	Document 1	Document 2 (μm)
“The sky is”	1	1
“sky is blue”	1	0
“is blue and”	1	0
“blue and the”	1	0
“and the sun”	1	0
“the sun is”	1	0
“sun is bright”	1	0
“The sun in”	0	1
“sun in the”	0	1
“in the sky”	0	1
“the sky is”	1	1
“sky is bright”	0	1

See Table 1, we can make out that the given documents are closely similar but the extent of similarity needs to be calculated. So, using this relation matrix, we can calculate Jaccard’s coefficient to get the syntactic similarity score for the input documents. Let shi1 and shi2 be the two sets of shingles generated for the use case, the mathematical formula for similarity measurement using boolean model would be,

$$\frac{shi1 \cap shi2}{shi1 \cup shi2}$$

So, it can be said that Jaccard coefficient is equal to the ratio of intersection of the two set to their union. For our case where we have two documents, following cases could arise:

- Case 1: Both documents(columns) contain 1
- Case 2: One of the document is 1 and other is 0
- Case 3: Both are labelled 0

So, as per the concept of the Jaccard coefficient, similarity would be given as,

$$\frac{Case1}{Case1 + Case2}$$

This score would always range from 0 to 1. For above case this measure = 1/12 = 0.083

2.2 Semantic Analysis of Text

In this algorithm, as explained in [4], we will understand the short text by dividing it into a collection of terms and then try to under the semantic of each term. So, the task of short text understanding is formulated as follows:

- Text segmentation - break down the short text into best segments
- For given text segments, create a similarity matrix with similarity scores between terms
- The available two short texts are treated as binary vectors where every term is represented with a 1 if it is present and 0 if absent.
- Using the similarity matrix and the binary vectors, $\vec{a}$ and $\vec{b}$, we find similarity as

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a}W\vec{b}}{|\vec{a}||\vec{b}|}$$

  where W is the semantic similarity matrix.

In the semantic similarity matrix, W, every element w_{ij} represents the similarity score between terms p_i and p_j. The score is some lexical measure which if similar, matrix will also be similar. Diagonal elements (w_{ij} where i = j) will be 1 since the terms p_i and p_j will be the same.

The similarity matrix value, w_{ij} is computed by the wup metric which computes the similarity of the nodes as a function of the path length from the least common subsumer (LCS) of the nodes ad the depths of the two synsets in the WordNet taxonomies. If you have two nodes, the LCS is defined as the most specific node which both share as an ancestor. If node 1 (N_1) is 'car' and the other is 'boat' ($_{N2}$) then the LCS would be 'vehicle'. The similarity between nodes N1 and N2 is given by

$$sim_{wup} = \frac{2 * depth(LCS(N_1, N_2)}{depth(N_1) + depth(N_2)}$$

where depth gives the depth of the node in the WordNet.
The similarity matrix for two sample texts "The dog sat on the mat" "The mutt sat on the rug" maybe given as

Table 2: Similarity matrix

	dog	mat	mutt	rug	sat (μm)
dog	1	0	0.8	0	0
mat	0	1	0	0.9	0
$mutt$	0.8	0	1	0	0
rug	0	0.9	0	1	0
sat	0	0	0	0	1

where the p_is are "dog", "mat", "mutt", "rug", and "sat".

The rationale behind using both the above approaches for this project was simplicity of understanding and implementation. The first approach of syntactic similarity is inspired by [6] and that made sense for implementation because we have to do term-based analysis of the two given questions. Shingles are preferred over n-gram approach because shingles help in tokenization of words than characters unlike n-grams. A relation matrix is build and the distance between the terms is calculated using jaccard coefficient. This helps in classification of the two given texts as similar and dissimilar. The texts that are classified as dissimilar by syntactic analysis are then analysed semantically to judge if they have similar intent using different synonymous words. This is done using the method listed in [4] because the paraphrasing detection pertains to the goal of the project of semantic analysis of the text for similarity. So, as per the research paper, if two texts are paraphrases of each other, then they are referring to the similar context/intent. This is the algorithm that we have applied in our proposed method by calculation of distance between terms based on the similarity matrix obtained using WordNet library.

3 Plan and Experiment

The plan is to divide the project into different phases of data cleaning, data transformation, modeling and visualization phase with following logistics

3.1 Data Set

The train dataset consists of a range of a pair of questions(q1,q2) with the classification labels defined by 0: not similar and 1: Is similar.
Dataset contains 404290 records (question pairs) where 255027 have 0 classification label and 149263 have label 1. We use stratified sampling of 70:30 (training set : test data) to get a training set containing 2830003 records with 178519 records with label 0 and 104484 records with label 1.

3.2 Description of Testbed

The Test Dataset consists of test cases of question pairs where the algorithm will determine whether the pair of questions are similar or not. The test data has 121287 records with 76508 records with label 0 and 44779 records with label 1.

3.3 Hypothesis

The classification of two given questions based on similar intent. That is, if two questions have similar meaning or reference, then they should be classified as label 1.

3.4 Experimental Design

3.4.1 Data Preprocessing

- Many of the textual question data pairs consists of HTML entities which can be parsed using packages to replace them by standard HTML tags or by the direct approach of context free grammar
- Removal of punctuation marks
- Removal of stop words
- Converting all characters of the input string to lowercase
- Trim the trailing white spaces from the beginning and end of the text
- Change extra whitespace to single whitespace

3.4.2 Parse-1

Lexical analysis of the given two short texts for syntactic similarity using the algorithm described in proposed method above.

3.4.3 Parse-2

After parse 1 to determine the structural similarity between the questions, we will determine the semantic similarity between the two by using the semantic analysis algorithm described above in the proposed method. This will be performed only if the lexical analysis of the two texts is labelled as 0

4 Results

By syntactic analysis of the given dataset, we achieved an accuracy of 62.4% using shingles analysis and calculating distance between the two given short texts using Jaccards coefficient. The confusion matrix for the sample set of 10,000 data is shown in Table 3:

Table 3: Confusion matrix for Parse 1

	0	1
0	2628	959
1	2801	3612

From the semantic analysis we achieve accuracy of 65% on an average for multiple samples of 100,000 records. Due to system constraints it was not possible to use all of the 404290 records. Accuracies of the order of 65% we found with results such as in Table 4

Table 4: Confusion matrix for Parse 2

	0	1
0	49028	13718
1	21845	15409

Table 5: Prediction Statistics

	Precision	Recall	F1-Score	Support (μm)
$Class0$	0.69	0.78	0.73	62746
$Class1$	0.53	0.41	0.46	37254
$Avg/Total$	0.63	0.64	0.63	100000

The accuracies for smaller sample sets of size 10,000, 25,000, and 50,000 statement pairs were also evaluated and it was found that their accuracies ranged similar to this. It was observed that the accuracy may have been impacted by the way WordNet evaluated the similarity measure where in the homonyms identified and evaluated may have been affected by the Part of Speech they belonged to and this may be further worked on in order to attempt to improve the performance of the model.

5 Conclusion

This was an interesting project to take up for understanding the niches of natural language processing. We learnt about different kinds of analysis stages for natural text, about short text and the challenges it brings along for natural language analysis. We reviewed a variety of literature for classification of two short texts for similarity. It varied from research that has been done in this domain to classify by syntactic similarity which is the similarity by structural analysis of the sentences. If two texts are structured similarly and use the similar words, we classify them under label 1. We explored the significance of different text preprocessing ranging from stemming to use of n-grams and shingles. We learnt the practical implementation of different distance metrics learnt during the lectures and how to go around those on real-world data. Later on, we moved to understanding the semantic similarity part of the given questions and how can use of different words in texts with similar intent lead to their misclassification. So, we went through a set of researches done in this domain where we came across different approaches for semantic analysis of text which parsed from building a co-occurence matrix using Probase to building a similarity matrix using WordNet library.

6 References

[1] Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. International Journal of Computer Applications, 68(13).

[2] Kaur, A. (2015). A Novel Approach For Syntactic Similarity Between Two Short Text. International Journal of Scientific & Technology Research, 4(06), 2277-8616.

[3] Hua, Wen, Zhongyuan Wang, Haixun Wang, Kai Zheng, & Xiaofang Zhou. "Short text understanding through lexical-semantic analysis." In Data Engineering (ICDE), 2015 IEEE 31st International Conference on, pp. 495-506. IEEE, 2015.

[4]Fernando, Samuel, & Mark Stevenson. "A semantic similarity approach to paraphrase detection." Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics. 2008

[5] Natural Language Processing... Understanding what you say *Tan Chin Tuck*. Retrieved on 1st December,2017 from `http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol1/ctt/`.

[6] Text Similarity *Ethan Liu*, Retrieved on 7th November,2017 from `http://ethen8181.github.io/machine-learning/clustering_old/text_similarity/text_similarity.html`.

Github Clone Link : `https://github.com/doneria-anjali/textSemanticAnalysis.git`