Identification of Text Similarity Based On Context

Shreya Saloni Verma[1], Abdullah Sarguroh[2], Jyotsana Rawat[3]

[1]Student, Dept. of IT Engineering, Pillai College of Engineering, New Panvel, Maharashtra, India
[2]Student, Dept. of IT Engineering, Pillai College of Engineering, New Panvel, Maharashtra, India
[3]Student, Dept. of IT Engineering, Pillai College of Engineering, New Panvel, Maharashtra, India

---***---

Abstract- *Text similarity computing plays an important role in natural language processing. The similarity calculation of short text is influenced by the small feature of text words and the accuracy is low. so it is a common improvement method to calculate the similarity of short texts with word semantic similarity. The word similarity calculation method combines two word semantic similarity by some strategies. Instead of doing a word for word comparison, we also need to pay attention to context in order to capture more of the semantics. Calculating similarities between texts that have been written in English language is still one of the most important challenges facing natural language processing. The proposed system will find the similarity between two English texts by using similarity measures techniques: Semantic similarity measure, Cosine similarity measure and N-gram. In our proposed system we will design English Semantic Net that stores the keywords for a specific field, by this network we can find semantic similarity between words according to specific equations.*

Keywords-- **Text Similarity, Context Based Similarity.**

1. INTRODUCTION

We always need to compute the similarity in meaning between texts.

Search engines need to model the relevance of a document to a query, beyond the overlap in words between the two. For instance, question-and-answer sites such as Quora or Stack overflow need to determine whether a question has already been asked before.

In legal matters, text similarity tasks allow to mitigate risks on a new contract, based on the assumption that if a new contract is similar to an existent one that has been proved to be resilient, the risk of this new contract being the cause of financial loss is minimised. Here is the principle of Case Law principle. Automatic linking of related documents ensures that identical situations are treated similarly in every case. Text similarity fosters fairness and equality. Precedence retrieval of legal documents is an information retrieval task to retrieve prior case documents that are related to a given case document.

In customer services, AI systems should be able to understand semantically similar queries from users and provide a uniform response. The emphasis on semantic similarity aims to create a system that recognizes language and word patterns to craft responses that are similar to how a human conversation works. For example, if the user asks “What has happened to my delivery?” or “What is wrong with my shipping?”, the user will expect the same response.

Text similarity has to determine how ‘close’ two pieces of text are both in surface closeness [lexical similarity] and meaning [semantic similarity].

For instance, how similar are the phrases “the cat ate the mouse” with “the mouse ate the cat food” by just looking at the words?

- On the surface, if you consider only word level similarity, these two phrases appear very similar as 3 of the 4 unique words are an exact overlap. It typically does not take into account the actual meaning behind words or the entire phrase in context.

- Instead of doing a word for word comparison, we also need to pay attention to context in order to capture more of the semantics. To consider semantic similarity we need to focus on phrase/paragraph levels (or lexical chain level) where a piece of text is broken into a relevant group of related words prior to computing similarity. We know that while the words significantly overlap, these two phrases actually have different meanings.

2. LITERATURE SURVEY

[1] A New Context-Based Similarity Measure for Categorical Data Using Information Theory

In this paper, they have proposed a new unsupervised similarity measure for categorical data based on the information theoretic approach. The new proposed measure could be able to integrate the information of relations between attributes through co-occurrence content. In order to evaluate our measure, an experiment has been conducted to compare its effectiveness with other categorical similarity measures. The results have shown that the new proposed measure has a competitive performance compared to the others especially when handled with highly correlated data sets. For the future work, they intend to improve o measure so that it could automatically select a good β parameter value according to different data sets. It is also useful to evaluate our measure with different applications and techniques as well.

[2] A Comparative Analysis of Text Similarity Measures and Algorithms in Research Paper Recommender System

The purpose of the experiments conducted in this paper was to test the performance of data mining algorithms that are going to be used to develop their research paper recommender system. They tested 3 algorithms (Random Forest, Recursive Partitioning and Boosted tree) for their accuracy and efficiency. When they evaluated the performance of all the three algorithms, the rpart algorithm proved to be more efficient and accurate when compared to the other two counterparts that had a very poor performance. Farther accuracy of the prediction was conducted, and the rpart machine learning algorithm was selected and used in the classification of research papers due to its shortest running time to classify the datasets. The similarity between the research papers was accomplished by utilising the cosine similarity. Having measured the cosine similarity, the measures can be taken to collect top-k most similar papers.

[3] Short Text Similarity Calculation Using Semantic Information

In this paper, a semantic similarity calculation method based on corpus and knowledge is proposed for the application of short text. The pseudo-code of the word semantic similarity algorithm includes the three parts and the short text semantic similarity algorithm pseudo-code. The word similarity method combines Word2Vec to calculate corpus semantic similarity and using the improved Li similarity measure in WordNet to compute knowledge-based semantic similarity by some strategies, and also uses an extensible common synonyms dictionary to modify the strategies. Finally, the short text semantic similarity is calculated based on the improved word semantic similarity method.

[4] A Paraphrase and Semantic Similarity Detection System for User Generated Short-Text Content on Microblogs

In this paper, they have proposed a machine learning based approach towards this. They have proposed a set of features that, although well-known in the NLP literature for solving other problems, have not been explored for detecting paraphrase or semantic similarity, on noisy user-generated short-text data such as Twitter. They apply support vector machine (SVM) based learning.

[5] A Survey of Text Similarity Approaches

This survey discusses the existing works on text similarity through partitioning them into three approaches; String-based, Corpus-based and Knowledge-based similarities. Furthermore, samples of combination between these similarities are presented. This survey represents the most popular string similarity measures which were implemented in the SimMetrics package.

[6] A Semantic Similarity Approach to Paraphrase Detection

This paper presented a novel approach to the problem of paraphrase identification. Our method makes use of WordNet-based lexical similarity measures applied differently from previous approaches. The system was evaluated on the Microsoft Research Paraphrase Corpus and found to outperform previously reported approaches. The work here has focused on the use of lexical similarity for paraphrase detection, essentially using a bag-of-words model.

[7] Context Based Similarity Matching

In this paper by J.Luo present a robust algorithm designed to detect a class of compound colour objects given a single model image. A compound colour object is defined as having a set of multiple, particular colours arranged spatially in a particular way, including flags, logos, cartoon characters, people in uniforms, etc. This approach is based on a particular type of spatial-colour joint probability function called the colour edge co-occurrence histogram.

3. PROPOSED SYSTEM

The proposed methodology considers the text as a sequence of words and deals with all the words in sentences separately according to their semantic and syntactic structure. The information content of the word is related to the frequency of the meaning of the word in a lexical database or a corpus.

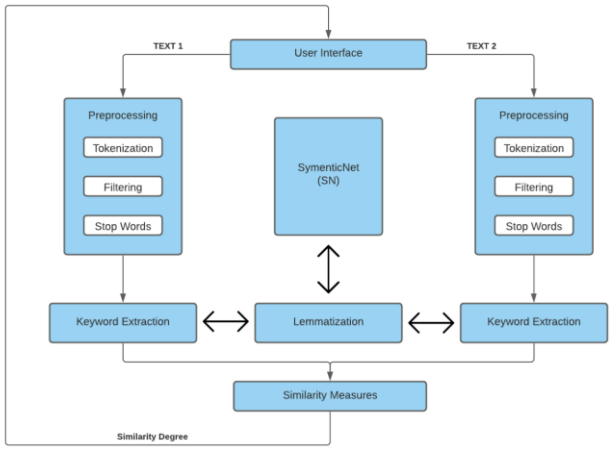


Fig -1: Proposed Diagram

The main steps of the proposed system are:

➔ Text Prepossessing

- Lemmatization
- Similarity Measures

The preprocessing is the first step of the proposed system, including three stages, these stages are:

- Tokenization
- Stop words removal
- Filtering
- Text Prepossessing

The input to the proposed system is an English text, it consists of sentences.

- Tokenization

In the tokenization part, the sentence is converted to a list of tokens, according to the spaces between English words or stop marks. There are various tokenization techniques available which can be applicable based on the language and purpose of modeling.

There are different methods and libraries available to perform tokenization. NLTK, Gensim, Keras are some of the libraries that can be used to accomplish the task.

We have used NLTK(NATURAL LANGUAGE TOOLKIT) library. Word_tokenize and sent_tokenize are very simple tokenizers available in NLTK.

Sent_tokenize() will string into multiple sentences. The sent_tokenize function uses an instance of PunktSentenceTokenizer from the nltk.tokenize.punkt module, which has already been trained and thus very well knows to mark the end and beginning of sentence at what characters and punctuation.

word_tokenize() function is a wrapper function that calls tokenize()

Algorithm-

```
from nltk.tokenize import sent_tokenize, word_tokenize

word_tokens = word_tokenize(str1)
```

Example-

str1 = “The state is having a crisis, medical emergency and population outbreak.“

Output- [‘The’, ‘state’, ‘is’, ‘having’, ‘a’, ‘crisis’, ‘,’, ‘medical’, ‘emergency’, ‘and’, ‘population’, ‘outbreak’, ‘.’]

- Stop Words Removal

After converting the input English text to a list of tokens, the next step is stop words removal. The stop words will be removed. The stop words can be defined as words that don't have any remarkable importance, or any word that don’t give any importance in finding the similarity could be considered as a stop word.

Algorithm-

```
        from nltk.corpus import stopwords,wordnet

        for words1 in word_tokenize(str1):

    if words1 not in stop_words:

        filtered_sentence1.append(words1)
```

Example-

str1 = “The state is having a crisis, medical emergency and population outbreak.“

Output- [‘The’, ‘state’, ‘having’, ‘crisis’, ‘,’, ‘medical’, ‘emergency’, ‘population’, ‘outbreak’, ‘.’]

- Filtering

The tokens without the alphanumeric part being removed such as !,?,:,;, as it does not give much importance to the sentence.

Algorithm-

```
for words1 in word_tokenize(str1):

    if words1 not in stop_words:

      if words1.isalnum():

        filtered_sentence1.append(words1)
```

Output-

str1 = “The state is having a crisis, medical emergency and population outbreak.“

Output- [‘The’, ‘state’, ‘having’, ‘crisis’, ‘medical’, ‘emergency’, ‘population’, ‘outbreak’]

- Lemmatization

Lemmatization will reduce the inflected words properly ensuring that the root word belongs to the language. Lemmatization will consider the context and convert the word to its meaningful base form. WordNet Lemmatizer that uses the WordNet Database to look up lemmas of words. Wordnet is a large, freely and publicly available lexical database for the English language aiming to establish structured semantic relationships between words.

We first tokenize the sentence into words using nltk.word_tokenize and then we will call lemmatizer.lemmatize() on each word.

Algorithm-

```
        from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

for i in filtered_sentence1:

    lemm_sentence1.append(lemmatizer.lemmatize(i))

    print(lemm_sentence1)
```

Examples-

-> rocks : rock

-> corpora : corpus

-> better : good

- Finding Similarity

After the lemmatization process of text1 and text2, we will find similarity degrees between them by using the Synset:

Synset-

It is a special kind of a simple interface that is present in NLTK to look up words in WordNet. Synset instances are the groupings of synonymous words that express the same concept. Some of the words have only one Synset and some have several.

Algorithm-

```
from nltk.corpus import wordnet

syn = wordnet.synsets('day')[0]

print ("Synset name : ", syn.name())

print ("\nSynset meaning : ", syn.definition())

print ("\nSynset example : ", syn.examples())
```

Output-

```
Synset name :   day.n.01

Synset meaning :   time for Earth to make a complete rotation on its axis

Synset example :  ['two days later they left']
```

Wu & Palmer Similarity-

The wup_similarity method is short for Wu-Palmer Similarity, which is a scoring method based on how similar the word senses are and where the Synsets occur relative to each other in the hypernym tree. One of the core metrics used to calculate similarity is the shortest path distance between the two Synsets and their common hypernym.

It calculates relatedness by considering the depths of the two synsets in the WordNet taxonomies, along with the depth of the LCS (Least Common Subsumer).

The score can be 0 < score <= 1. The score can never be zero because the depth of the LCS is never zero.

Algorithm-

```
from nltk.corpus import wordnet

syn0 = wordnet.synsets('hello')[0]

syn1 = wordnet.synsets('selling')[0]

print ("hello name : ", syn0.name())

print ("selling name : ", syn1.name())

syn0.wup_similarity(syn1)
```

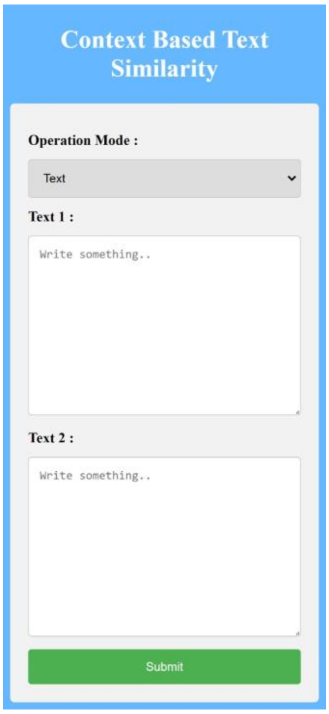
Output-

```
hello name :   hello.n.01

selling name :   selling.n.01

0.26666666666666666
```

hello and selling are apparently 27% similar! This is because they share common hypernyms further up the two.

4. RESULT

In this paper, we have briefly presented a Identification of text similarity based on context, we have explained the objectives of the proposed system, we have conducted a literature survey of previous works. We have briefly explained the existing system architecture and the proposed architecture. Further, the report also explains all the tools and technologies implemented in the proposed system such as HTML, CSS, JS, Python.



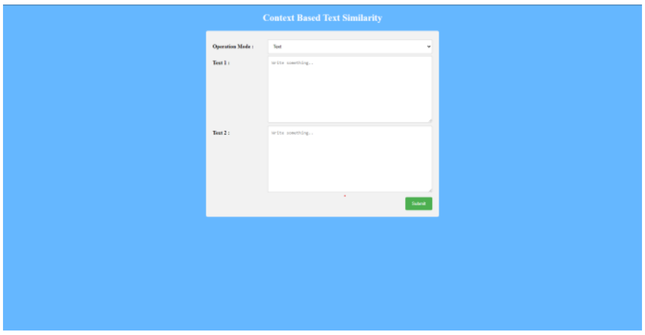
Context Based Text Similarity

Operation Mode : Text

Text 1 : Write something..

Text 2 : Write something..

Fig -2: Main page - Mobile View



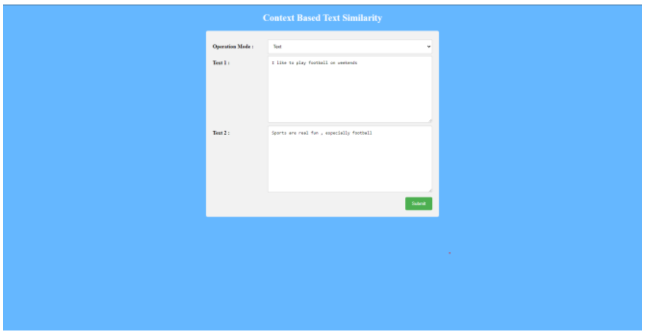
Context Based Text Similarity

Operation Mode : Text

Text 1 : Write something..

Text 2 : Write something..

Fig -3: Main page - Web View



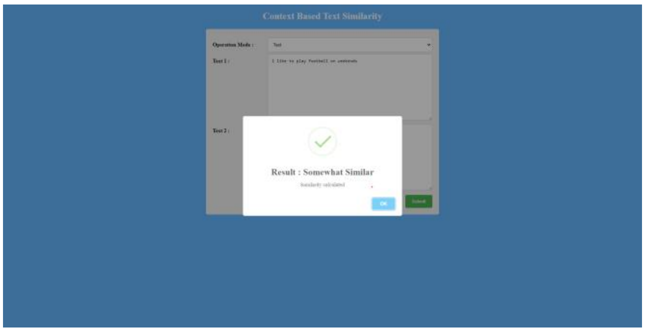
Context Based Text Similarity

Operation Mode : Text

Text 1 : I like to play football on weekends

Text 2 : Sports are real fun , especially football

Fig -4: Input



Context Based Text Similarity

Operation Mode : Text

Text 1 : I like to play football on weekends

Text 2 :

Result : Somewhat Similar

Similarity calculated

Fig -5: Output

ACKNOWLEDGEMENT

We would like to thank our guide Prof. Shubhangi Chavan who helped us regarding any and all queries while working on the project. In every phase of the project her supervision and guidance shaped this report to be completely perfect.

We would like to express our gratitude to our Head Of Information Technology Department Dr. Satishkumar Varma for giving us this opportunity and for motivating us to do innovative things that will be beneficial for our future.

We would like to thank everyone who provided us with an opportunity to work on projects like this thereby increasing our knowledge. We would like to thank our Principal Dr. Sandeep Joshi who always encouraged and motivated us.

REFERENCES

[1] Thanh-Phu Nguyen1(B), Mina Ryoke, and Van-Nam Huynh1 ”A New Context-Based Similarity Measure for Categorical Data Using Information Theory” Asahidai, Nomi, Ishikawa 923-1292, Japan

[2] Maake Benard Magara, Sunday O. Ojo, Tranos Zuva "A Comparative Analysis of Text Similarity Measures and Algorithms in Research PaperRecommender Systems" Information Communications Technology and Society (ICTAS), (2018).

[3] Haoyu Pu, Gaolei Fei, Hailin Zhao, Guangmin Hu, Chengbo Jiao, Zhoujun Xu "Short Text Similarity Calculation Using Semantic Information" Big Data Computing and Communications, (2017).

[4] Ishrath Jahan C, Abitha E “Context Based Similarity Matching” International Journal of Science and Research (IJSR)

[5] Kuntal Dey, Ritvik Shrivastava, Saroj Kaus “A Paraphrase and Semantic Similarity Detection System for User Generated Short-Text Content on Microblogs” IBM Research India, NIST Delhi , IIT Delhi

[6] Wael H. Gomaa And Aly A. Fahmy“A Survey of Text Similarity Approaches”, International Journal of Computer Applications, (2018)

[7] Samuel Fernando and Mark Stevenson “A Semantic Similarity Approach to Paraphrase Detection” ,University of Sheffield