Assessing Sentence Similarity using Lexical and semantic Analysis for Text Summarization using Neural Network.

Abujar Onkon
Jahangirnagar University
Savar, Dhaka, Bangladesh
sheikhabujar@gmail.com

Md. Shahidul Islam
Jahangirnagar University
Savar, Dhaka, Bangladesh
Shahidul555@gmail.com

Abu Abed Md. Shohaeb
Jahangirnagar University
Savar, Dhaka, Bangladesh
shohaeb@gmail.com

Abstract— This paper has presented sentence similarity measure using lexical and semantic similarity. Degree of similarity was mentioned and implemented in the proposed method. There are few resources available for Bengali language. More development on Bengali language is just more than essential. Bengali WordNet is not stable as like other WordNet available for English language. The key challenges of Natural language Processing is to identify the meaning of any text. Text Summarization is one of the most challenging applications in the field of Natural Language Processing. An expert Text Summarizer need proper analysis of given input text. To identify the degree of relationship among input sentences will help to reduce the inclusion of unimportant sentences in summarized text. This is the objective of this research, to identify similar sentences. Result of summarized text always may not identify by optimal functions, rather a better summarized result could be found by measuring sentence similarities. The current sentence similarity measuring methods only find out the similarity between words and sentences. These methods states only syntactic information of every sentence. There are two major problems to identify similarities between sentences; such problems were never addressed by previous proposed strategies: provide the ultimate meaning of the sentence and added the word order, approximately. In this paper, the main objective was tried to measure sentence similarities, which will help to summarize any Language text, though specially considered for English and Bengali language. The experiment exhibited a proposed method of measuring English and Bengali sentence similarity. Results will states the outstanding performances of our proposed algorithms. Text summarization follows two different methods: Extractive and Abstractive method. Sentence similarity can play a vital role in both, Abstractive and Extractive text summarization approach. Through a proper measurement of sentence similarity, centroid sentences could be extracted and considered as a main and/or leading sentence.

Keywords—Sentence Similarity, Text Summarization, Benglai Summarization, Sentence clustering,Deep Learning.

I. INTRODUCTION

Text Summarization is a tool that attempts to provide a gist or summary text of any given text as input, automatically. It helps to understand any large document in a very short time, by getting the main idea and/or information of entire text from a summarized text. To produce the proper summarization, there are several steps to follow, Such as: Lexical Analysis, Semantic analysis and Syntactic analysis. Few well known features like: Sentence clustering, Word segmentations, frequent words identification, Similarity measures, etc. Possible methods and research findings regarding sentence similarity is stated in this paper. Several key factors were mentioned in details. Bengali language has very different sentence structural forms and analyzing those Bengali alphabets may found difficult in various programming platforms. The best way of preprocessing the Bengali and English sentences before deep analysis, is using Unicode [2]. Sentence could be identified in a standard form, it will help to identify and/or modify sentence or words structure as needed. Tokenization

The degree of measuring sentence similarity is being measured by method of identifying sentence similarity as well as large and/or short text similarity. Sentence or text similarity is a very important for various fields, such as: text summarization, text categorization, text mining, search results optimization, and in various fields of natural language processing. Sentence similarity measures should state information like: does any two or more sentences are either fully matched in lexical form or in semantic form, sentence could be matched partially or we could found any leading sentence. Identifying centroid sentence is one of the major tasks to accomplish [1]. Few sentences may contain some of major or important words which may not be identified by words frequency. So, only depending on word frequency may not always provide the expected output, though several times most frequent words may relate with the topic models.

Sentence could be written in different form, sometimes sentences may found in a form of same meaning though written by different words. Or sometimes sentence may lead or redirect towards other sentences. Those related sentences may avoid while preparing a better text summarizer [3]. But related or supporting sentences may add a value to the leading sentences. Finally most leading sentence and relationship between sentences could be determined. Relationship regarding text similarity could be measured in such ways, such as: word to word, word to sentence, sentence to word and sentence to sentence.

Semantic measures are rapidly using in semantic web technologies as a particular solution regarding ontological based solutions. It is very much needed to define the relationship between same domains using ontological analysis [4]. There are several frameworks of semantic web related ontological technology, such as: N-Triples, which is known as a resource description framework (RDF), RDF Schema and the web ontology based language. Recently, the use of WordNet becoming more and more flexible. There are a lot of ontological group sets in various languages are available to use. To express the cognitive synonyms which are also known as synsets, we could use WordNet. Where XML Schemas are widely dependent on ontological explanation and representation. Several semantic processing fields are widely adopted in several text summarization techniques, only purpose of producing a better summarized output. Few Gene ontology (GO) are introducing to improve the result of ontology generation or representation. Semantic spaces could be divided into two different space models, as Noun space and other one is verb space model. WordNet similarity measure defines the values of every space vector. This is one of the best approach rather working with the most frequent words results.

In this paper, we have discussed several important factors regarding assessing Sentence and/or text similarity. Major findings are mentioned in details and more importantly a possible deep learning methods and model were stated here. Possible benchmarking methods were discussed and analyzed by using several online Bengali dataset, mostly used from News portals and other possible online resources, available in public domain. Several experiment results were stated and explained with necessary measures. The rest of the paper is being organized as follows: literature review, proposed method, result experiments and the last section is drawing the chapter-conclusion and possible future work.

II. LITERATURE REVIEW

The basic feature of text summarization would be either abstractive or extractive, approach. Extractive method applies several manipulation rules over word, sentence or paragraph. Based on weighted values or other measures, extractive approach choose appropriate sentence. In this method summarize sentence will also pick from input text. In other words, Abstractive method extracts knowledge from the given input text. A fully new summarized sentence will be generated based on the analysis of sentence knowledge. Lexical, Semantic and syntactic analysis play vital role for generating appropriate summarized word. This may not look like, exact sentences collected from input text. Rather it will look like, similar to human generated summarized output. Abstractive summarization requires several weights like, sentence fusion, constriction and basic reformulation (Mani & Maybury, 1999; Wan, 2008).

Oliva et al. (2011) introduced a model SyMSS, which measure sentence similarity by assessing, how two different sentences systaltic structure influence each other. Syntactic dependence tree help to identify the rooted sentence, as well as the similar sentence. This methods state that, every word in a sentence has some syntactic connections and this will create a meaning of every sentence. To composing sentences, semantic information will be obtained to find out the phrase. WordNet could be done the same process to find out the composing sentence.

The combination of LSA (Deerwester et al., 1990) and WordNet (Miller, 1995) to access the sentence similarity in between every words were proposed in Han et al.(2013). They have proposed two different methods to measure sentence similarity. First one makes a group of words – known as the align-and-penalize approach and the Second one is known as SVM approach, where the method applies different similarity measures using n-gram similarity. Such as: skip-bigram, bi-gram, tri-gram and uni-gram. And to produce the final similarity by using Support Vector Regression (SVR), they use LIBSVM (chang and Lin, 2011), as another similarity measure.

A threshold based model always returns the similarity value between 0 and 1. Mihalcea et al. (Mihalcea et al., 2006) represents all sentences as a list of bag of words vector and they consider first sentence as a main sentence. To identify word-to-word similarity measure, they have used highest semantic similarity measures in between main sentence and next sentence. The process will continue repeated times until the second main sentence could be found, during this process period. And finally an arithmetic weighted result will be combined and the threshold value will be equal to 0.5 to identify or measure paraphrases. Sentence similarity value more than 0.5 will be considered as a tagged paraphrase.

Das and Smith (Das and Smith, 2009) introduced a probabilistic model which states Syntax and semantic based analysis. In addition, a hidden loose alignment will be created in between of two different sentences (in a tree position structure). If the proposed posterior classifier probability crossed the value 0.5, those all of the pair of words will be considered as a group of paraphrase.

Heilman and Smith (Heilman and Smith, 2010) introduces as new method of editing tree, which will contain syntactic relations between input sentences, will identify paraphrases. To transform one existing tree into a new generated tree, a logistic regression classifier model will process nine different states like insertion, delete and modification. The logistic regression classifier will be trained to find out a short sequence.

To identify sentence based dissimilarity, a supervised two phase framework has been represented using semantic triples (Qiu et al., 2006). In three different phases they have processed the entire proposed methods; in first step they have calculated the sentence similarity by using semantic measures. By using a greedy manner related words pair group will be added in a token. The second phase is responsible to identify the unpaired words or if any other information exists, need to collect. To identify wither paraphrase or not, a Support Vector Machine (SVM) classifier will be applied over the unpaired data set, as well as numerical expressions, semantic similarity nodes and word trees, to identify whether they are semantically similar to other tuples of words in the form of a similar sentence. Several contextual features like expected initial target sentence length and the pair of similar or dissimilar token counts. Support Vector Machine (SVM) can combine distributional, shallow textual and knowledge based models using support vector regression model.

A supervised similarity measures acquire a very appropriate results following the same domain measures. A large number of data set or corpus is required for their training purpose. To handle the further process, it was needed to identify the sentence pattern. The large corpus will be used to identify the similar sentence of processed sentence.

This paper used an unsupervised method to measure sentence similarity. Different methods and algorithms were applied as a part of proposed model. Algorithm of sentence similarity matrix representation will be processed by three different part of text analysis: Lexical, Syntactic and Semantic methods. Besides that this paper states different possible approach for text summarizer. A better text summarizer needs a tool to measure and/or identify the percentage of similarity between sentences. As no sentences will not be hundred percent semantically similar. So if, the sentence similarity requires semantic methods, the maximum tuples of similarity measure values will be considered as a similar sentence. In several other researches, it was found that that the average sentence similarity value should be more than 0.5 in the range of 0 to 1.

III. PROPOSED METHOD

This Section represents a new proposed sentence similarity measuring model for English and Bengali language. The assessing methods, sentence representation and degree of sentence similarity has been explained in detail. The necessary steps required specially for Bangla language, has been considered while developing the proposed model. This model will work for measuring English and Bengali sentence similarity. The sentence structure and lexical form are very different for Bangla language. The semantic and syntactic measures also can add more values. The concept of working with all those necessary steps will help to produce better output, in every aspect. In this research - lexical methods has been applied, and untimely a perfect expected result has been found.

A. Lexical Layer Analysis:

The lexical layer has few major functions to perform, such as: Lexical representation and Lexical similarity. Both of these layers have several other states to perform. The Fig. 1 is the proposed model for lexical layer.

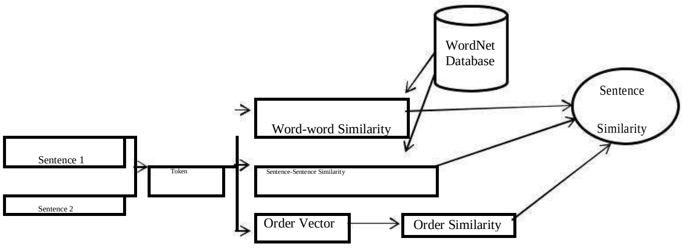

Fig. 1. Lexical Layer analysis model.

Figure 1 introduces the sentence similarity measures for lexical analysis. Different sentences will be added into a token. A word-to-word and sentence-to-sentence analyzer will perform together. An order vector will add all those word and/or sentence order in a sequence based on similarity measures. With the reference of weighted sum, the order of words and sentence will be privileged. A WordNet database will send lexical resources to word-to-word and sentence-to - sentence processes. Ultimately based on the order preference, the values from three different states (Word-word Similarity, Sentence-Sentence Similarity and Order Similarity) will generate the similar sentence output.

1. Lexical Analysis: This Sate splits sentence and words into different tokens for further processing.

2. Lemmatization: This is a step to convert and/or translates each and every token into a basic form, exactly from where it belongs to. The very same verb form in the initial form.

3. Stemming: Stemming is the state of word analysis. Word-word and sentence-to-sentence both methods need all their contents (text/word) in a unique form. Where every word will be treated as a rooted word. Such as : play, player – both words are different as word, though in deep meaning those words could be considered as a branch words of the word “Play”. By using a stemmer, we could have found all those text in a unique form before further processing. The confusion of getting different words in structure but same in inner meaning, will reduce. So, it is a very basic part of text preprocessing modules.

4. Levenshtein similarity (Lev.): Lev. counts the minimum number of similarity requires for the operation of insertion, deletion and modification of every character, which may require transforming from a string to another string. The similarity will be calculated based on the Eq. (1).

$$\text{LevSim}= 1.0-(\text{Lev.Distance}(W1,W2)/\text{maxLength}(W1,W2)) \quad (1)$$

5. Similarity between Words: The degree of relationship helps to produce a better text summarizer by analyzing text similarity. As Text is combination of words and/or sentences. The degree of measurement could be word-word, word – sentence, sentence – word and sentence –sentence. In this state, we had discussed the similarity between two different words. Such as there is a set of Word : $W= \{W_1,W_2,W_3,W_4,\ldots..W_n\}$. only similarity will be checked between two different words. The similarity between words could be measured by algorithm
1. The value of path will be dependent of distance values. And LevSimilairty (LevSim) value could be found from Eq. 1.

Algorithm 1- Similarity between Words

```
if Path_measure(W1,W2) < 0.1 then
    W_similarity = LevSim(W1,W2)
else
    W_similarity = Path_measure(W1,W2)
end if
```

A proposed similarity algorithm in stated in Algorithm 2.

Algorithm 2- Proposed algorithm

```
matrix = nmatrix(nsize(M)xsize(N))
total_sim = 0
k = 0
for buk ∈ A do
    for bu j ∈ B do
        matrix(k, j) = sim (tk, t j )
    end for
end for
for line(matrix) and has column(matrix) do
    total_sim = total_sim + large_sim(matrix)
    sim(matrix)) k++
end for
partial_sim = total_sim/k
return partial_sim
```

The Algorthm-2 receives the token on two different X,Y as input text. Then it will create a matrix representation of m*n dimensions. Variable total_sim (total similarity) and k (which is the value of iteration) will initially become 0. The variable total_sim adds more value between 0 to 1 to, and then will calculate the similarity of pair of sentences. The output is partial similarity which is the value of division of total similarity and k (iteration).

IV. EXPERIMENT RESULT

Several English and Bengali texts were tested though the proposed lexical layer to find out the sentence similarity measure. Texts are being collected from online resource, for example: wwwo.prothom-alo.com, bdnews24.com, etc. our python web crawler initially saved all those web (html content) data into notepad file. We have used Python – Natural Language Tool Kit (NLTK: Version– 3). Table 1 and Table 2 represents similarity matrix written in both English and Bengali language.

Table 1. Similarity Matrix

	Class	Education	Book	school
Class	1.0	0.15	0.87	0.39
Education	0.15	1.0	0.12	0.56
Book	0.87	0.41	1.0	0.37
school	0.39	0.78	0.95	1.0

Table 1. Similarity Matrix

স্কুল	পড়া	বই	পরীক্ষা
1.0	0.78	0.88	0.16
0.78	1.0	0.82	0.47
0.88	0.31	1.0	0.89
0.16	0.24	0.73	1.0

Table 1 and Table 2 contain a single sentence similarity matrix values. An English sentence was used in Table 1 as well as a Bengali sentence in Table 2. Both provide expected output. As the output was also analyzed manually. In both cases, output found satisfactory.

V. CONCLUSION AND FUTURE WORK

This paper represents sentence similarity measures using lexical similarity. Degree of similarity were mentioned and implemented in the proposed method here. This research found suitable output in the unsupervised approach. Though a huge dataset will be required to implement the supervised learning methods. There are other sentence similarity measures, could be done by semantic analysis and syntactic analysis. Both of these analysis if could be done together including lexical similarities, a better result could be found.

A. Selecting a Template (Heading 2)

First, Reference

[1] Rafael Ferreira et al. “Assessing Sentence Scoring Techniques for Extractive Text Summarization”, Elsevier Ltd., Expert Systems with Applications 40 (2013) 5755-5764.

[2] Abujar, Sheikh, and Mahmudul Hasan. "A comprehensive text analysis for Bengali TTS using Unicode." Informatics, Electronics and Vision (ICIEV), 2016 5th International Conference on. IEEE, 2016.

[3] Abujar, Sheikh, et al. "A Heuristic Approach of Text Summarization for Bengali Documentation." 8th International Conference on Computing, Communication and Networking (8th ICCCNT), 2017 8th International Conference on. IEEE,2017.

[4] Lee, Ming Che. "A novel sentence similarity measure for semantic-based expert systems." Expert Systems with Applications 38.5 (2011): 6392-6399.

[5] Mani, Inderjeet, and Mark T. Maybury, eds. Advances in automatic text summarization. Vol. 293. Cambridge, MA: MIT press, 1999.

[6] Oliva, Jesús, et al. "SyMSS: A syntax-based measure for short-text semantic similarity." Data & Knowledge Engineering 70.4 (2011): 390-405.

[7] Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R., 1990. Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. 41 (6), 391–407.

[8] Han, L., Kashyap, A.L., Finin, T., Mayfield, J., Weese, J., 2013. UMBC EBIQUITY-CORE: semantic textual similarity systems. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Association for Computational Linguistics, Atlanta, Georgia, USA, June, pp. 44–52.

[9] Miller, G.A., 1995. Wordnet: a lexical database for English. Commun. ACM 38, 39–41.

[10] Chang, C.-C., Lin, C.-J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (May (3)), 27, 1–27.

[11] Mihalcea, R., Corley, C., Strapparava, C., 2006. Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1. AAAI Press, Boston, Massachusetts, pp. 775–780.

[12] Heilman, M., Smith, N.A., 2010. Tree edits models for recognizing textual entailments, paraphrases, and answers to questions. In: Proceedings of 399 Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1011–1019.

[13] Heilman, M., Smith, N.A., 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In: Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1011– 1019.

[14] Qiu, L., Kan, M.-Y., Chua, T.-S., 2006. Paraphrase recognition via dissimilarity significance classification. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 18–26.

[15] Dzikovska, Myroslava O., et al. ”Intelligent tutoring with natural language support in the Beetle II system.”Sustaining TEL: From Innovation to Learning and Practice. Springer Berlin Heidelberg, 2010. 620-625.

[16] Jurgens, David, Mohammad Taher Pilehvar, and Roberto Navigli. ”SemEval-2014 Task 3: Cross-level semantic similarity.” SemEval 2014 (2014): 17.

[17] Mikolov, Tomas, et al. ”Extensions of recurrent neural network language model.” Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011.

[18] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. ”Glove: Global vectors for word representation.” Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014) 12 (2014).

[19] Rashtchian, Cyrus, et al. ”Collecting image annotations using Amazon’s Mechanical Turk.” Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk. Association for Computational Linguistics, 2010.

[20] Socher, Richard, et al. ”Parsing natural scenes and natural language with recursive neural networks.” Proceed-ings of the 28th international conference on machine learning (ICML-11). 2011