Bulletin of Electrical Engineering and Informatics
Vol. 11, No. 5, October 2022, pp. 2958~2967
ISSN: 2302-9285, DOI: 10.11591/eei.v11i5.3485

An insight into the intricacies of lingual paraphrasing pragmatic discourse on the purpose of synonyms

Jabir Al Nahian[1], Abu Kaisar Mohammad Masum[1], Muntaser Mansur Syed[2], Sheikh Abujar[3]
[1]Department of Computer Science and Engineering, Faculty of Science and Information Technology, Daffodil International University, Dhaka, Bangladesh
[2]Department of Computer Engineering and Science, School of Computer Science and Engineering, Florida Institute of Technology, Florida, United States
[3]Department of Computer Science, School of Computer Science and Engineering, Florida Institute of Technology, Florida, United States

Article Info

Article history:

Received Dec 11, 2021
Revised Apr 11, 2022
Accepted Aug 10, 2022

Keywords:

Natural language processing
NLTK
Paraphrasing algorithm
Paraphrasing technique
Synonym's paraphrase

ABSTRACT

The term "paraphrasing" refers to the process of presenting the sense of an input text in a new way while preserving fluency. Scientific research distribution is gaining traction, allowing both rookie and experienced scientists to participate in their respective fields. As a result, there is now a massive demand for paraphrase tools that may efficiently and effectively assist scientists in modifying statements in order to avoid plagiarism. natural language processing (NLP) is very much important in the realm of the process of document paraphrasing. We analyze and discuss existing studies on paraphrasing in the English language in this paper. Finally, we develop an algorithm to paraphrase any text document or paragraphs using WordNet and natural language tool kit (NLTK) and maintain "Using Synonyms" techniques to achieve our result. For 250 paragraphs, our algorithm achieved a paraphrase accuracy of 94.8%.

Corresponding Author:

Jabir Al Nahian
Department of Computer Science and Engineering, Daffodil International University
DIU Rd, Dhaka 1341, Bangladesh
Email: jabir15-10414@diu.edu.bd

1. INTRODUCTION

Alternate methods to present the same idea is known as paraphrases. A technique for computerized paraphrase learning is both practical and linguistically interesting [1]. Variability in presentation is a major difficulty for many NLP applications from a practical standpoint. The process of developing a fluent output paragraph from an inputting statement that communicates the same concept in such an alternative form is known called paraphrasing. It is indeed a major issue in natural language processing (NLP), featuring applications ranging from information retrieval to conversational agents to summarization. According to Prentice *et al.* [2] recognized the usage of online paraphrase tools, but they were also curious about the likelihood of online language translation tools being used. They used a sample of content provided to the students as a trigger for the paper as a parent paper to test the outputs of these tools. This article was subjected to six free online paraphrase tools as well as six sequential language translations using the Google Translate™ tool. Rogerson *et al.* [3] they first learned about paraphrasing techniques through a student's casual statement. Several natural language processing techniques, including information retrieval, dialogue systems, and relay on question answering. Due to the complexity of natural language, automatically creating correct and distinct appearing paraphrases remains a difficult research challeng [4].

We discuss three techniques for paraphrasing. Using synonyms, we can paraphrase sentences. Most words have multiple meanings based on context, and we must consider the synonym that best reflects the

correct meaning for the given situation. Here; "may" is replaced with "is likely to" "put upward pressure on "is replaced with "push up" and a verb can be replaced with a noun from the same word family. Also, an adjective can be replaced by a noun. The third technique is "Changing the grammatical structure" We take the sentence "Progress has been slower than was anticipated in the early 1986's", after grammatical structure change, we can write it "Progress has not been as quick as expected in the early 1986's". Using this approach, we can paraphrase any sentences. In this case, one grammatical structure for expressing a comparison ("slower than") has been swapped with another ("not as quick as"). The technique of paraphrasing in the English language has been studied extensively, and a few ways have been established.

Liu *et al.* [4] proposed an Unsupervised Paraphrasing by Simulated Annealing technique. Simulated Annealing is a unique method for achieving Unsupervised paraphrasing. They treat paraphrase creation as an optimization problem and offer a complex fitness function that takes into account paraphrase language, expression diversity, and semantic similarity fluency. The united professional sales association (UPSA) then performs a series of local edits to analyze the sentence space for this goal. Their approach is unsupervised and therefore does not need parallel corpus for train, making it adaptable to a variety of domains. Sulistyaningrum *et al.* [5] look into the challenges mechanical engineering vocational education students have with paraphrasing in academic writing classes, as well as the usage of online paraphrasing tools to help them overcome those difficulties. The information was gathered through two questionnaires given to students in response to the two issues outlined earlier. Using paraphrase tools to hide plagiarized work, as described by Wahle *et al.* [6], is a serious danger to scientific integrity. We compare the efficiency of five pre-trained word embedding models with machine learning classifiers and state-of-the-art neural language models to enable the identification of machine paraphrased content. We used several settings of the tools SpinBot and SpinnerChief to examine preprints of Wikipedia articles, research papers, and graduation theses that we paraphrased. Long former, the effectively approach, scored an average F1 of 80.99 percent (F1=99.68 percent for SpinBot and F1=71.64 percent for SpinnerChief cases), whereas human assessors scored F1=78.4 percent for SpinBot and F1=65.6 percent for SpinnerChief cases. We show that automated categorization overcomes the flaws of popular text-matching tools like Turnitin and PlagScan. An interview revealed that, Participants regularly used synonyms while paraphrasing, but they hardly ever altered the syntactic structures [7]. Siddique *et al.* [8] developed progressive unsupervised paraphrasing (PUP): a revolutionary deep reinforcement learning-based unsupervised paraphrase generating technique. PUP generates a bootstrap para which gets warmer the Deep reinforcement learning model using a variationally auto-encoder. The seed para is then gradually fine-tuned by PUP, driven by our innovative optimization method, which quantifies the accuracy of the obtained parts in each repetition without using parallel texts by combining semantic adequacy, expression diversity, and language fluency measurements. We found out by studying paraphrasing research that there are three techniques for paraphrasing; Changing the grammatical structure, using synonyms, and changing the form of words are the best paraphrasing technique [9]. Roy *et al.* [10] developed another paraphrase model just from an unidentified monolingual dataset. They developed a baseline form of the vector-quantized variationally auto-encoder to achieve this. They compared paraphrasing recognition, creation, and training augment to MT-based techniques. In every case, monolingual paraphrasing trumps unsupervised translation. The results of comparing with supervised translations were even less clear. For recognition and enhancement, monolingual phrasing is intriguing; for production, supervised translation is preferred. Barzilay *et al.* [11] created a paraphrasing model of paragraph paraphrasing in sentence synthesis, which is different from and so more complex than changing the form of words paraphrased. Multiple-sequence realignment is used in their approach to match sentences from unstructured text comparative datasets.

A variety of NLP approaches have been used to solve Paraphrase generation. But at this time, we could not find the "using synonym" technique to paraphrase paragraphs. And also, nobody includes their measurement accuracy in their paper [12], [13]. The system for free parameters of noun compounds is described by Afantenos *et al.* [14]. Their method combines the power of an unsupervised distributional word space representation with the accuracy of a supervised maximum-entropy classification algorithm; the distributional model produces a representation for a given compound noun, which is then used by the classifier to generate a set of suitable paraphrases. Witteveen *et al.* [15] present a practical method for doing the task of paraphrase on a range of texts and subjects utilizing a big language model. It is shown that their method can produce paraphrases not only at the sentence level but also for longer passages of text, like paragraphs, without the need to divide the material into smaller portions. Alassir *et al.* [16] suggest a technique for paraphrasing French phrases. Their approach is based on transducers and dictionaries. It entails substituting synonyms or antonyms for parts of the sentence's terms or moving to the passive form. Furthermore, in order to avoid the punctuation form case ambiguity, they separated these words into two portions - one for terms that begin with a vowel and the other for words that do not begin with a vowel. Ganitkevitch *et al.* [17] expand bilingual paraphrastic extraction to syntactic paraphrases and show that it is capable of learning a range of generic paraphrastic transformations such as passivation, dative shift, and

topicalization. They demonstrate how our model's feature set, development data, and parameter estimation method may be enhanced to adapt it to a variety of text creation applications. They demonstrate this adaptation by applying our paraphrasing model to the problem of sentence compression and achieving results that are comparable to those of cutting-edge compression methods. Several recent studies [18]-[28], we find a lot of research gaps in paraphrasing.

In this research paper, we offer a “Using Synonyms technique” algorithm for paraphrasing any nglish-language text composition. Improved accuracy is also a goal of English Grammar rules and the Natural Language Tool Kit. By including WordNet into the technique, we were able to solve this difficulty. In the following part, we'll go over a few scenarios in which WordNet and NLTK change is crucial. The following is how the rest of the paper is organized: Section 2 delves into the many techniques used in the methodology as well as the complete methodology. Section 3 describes the many resources that were used in the algorithm's evaluation. In section 4, the experimental method and dataset utilized in the experiment are described in depth. Section 5 contains the evaluation and outcomes. Finally, in section 6, we bring this study to a close.

2. METHOD

Until now, we've talked about many parts of speech and sentence structure in the English language. This section covers our methods for locating paraphrase paragraphs as well as the similarity between old and machine-generated paragraphs. We used python programming language to complete the whole work. Now, here we described our all-process step by step. Our algorithms follow Figure 1 flow chart to achieve its result.

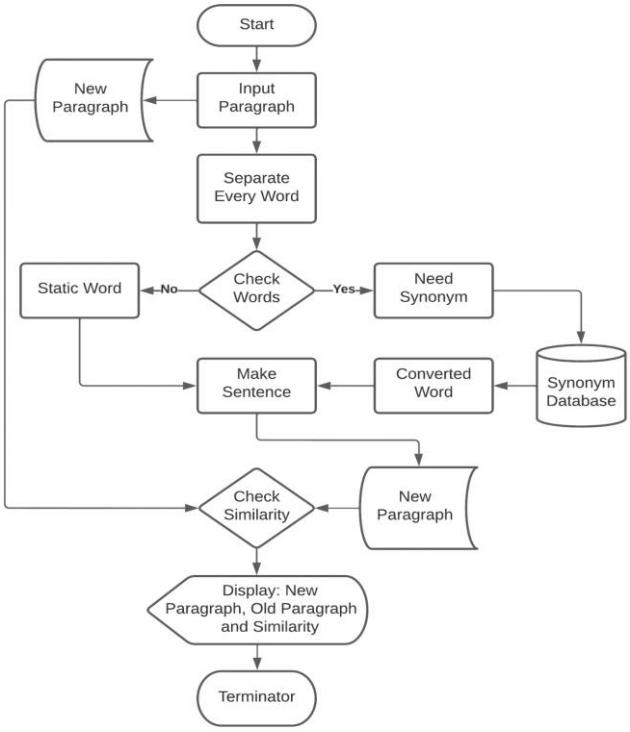

Figure 1. Algorithm flowchart

2.1. Implementation procedure

2.1.1. Split word

In this process, we used the split () method to split a string into a list. When we split a text into a list of words, we get a list with every sentence's word as an element. Splitting "I love programming" into a series of words, for example, yields ["I", "love", "programming"]. If we take a string, we can break it down into multiple smaller strings. There should be at least one dividing character in the text, which can be a space or

other punctuation. The split method defaults to using space as a separator. When we call the method, we get a list of all the substrings.

2.1.2. Listing words

Every text document has one or more sentences. Every sentence is made with its parts of speech. After reviewing sentence structures, we know that some words or some parts of speech do not need to change. Because these parts of speech or words provide the same meaning all time. That's why we create a list of specific types of words and parts of speech. We create here six lists "pronouns", "preposition", "auxilaryVerb", "numbers" and "randomWord" for no need to change Pronouns, Preposition, Auxiliary Verb, Numbers, WH Question, Random words. In Table 1 we show some lists of words. After listing all unchanging words, we can get our changeable words and parts of speech. Like Verb, Adjective, Adverb, Conjunction, Interjection, and other changeable words. After getting this data we separate two types of data. We don't change unchangeable words and we change changeable words following the below methods.

Table 1. Listing all unchanged words

Unchanged words & Parts of speech	Words
Pronouns	I, we, you, he, she, it, they, me, us, her, him, them, mine, ours, yours, hers, his, theirs, my, our, your, her, his, their, myself, yourself, herself, himself, itself, ourselves, yourselves, themselves, such, that, these, this, and those.
Preposition	About, anti, around, as, at, because, but, by, for, from, in, into, minus, of, off, on, onto, per, since, then, though, to, toward, under, underneath, unlike, until, upon, versus, via, with, within, and without.
Auxiliary Verb	Am, is, are, was, were, being, been, will, has, have, had, having, does, do, did, shall, should, would, and could.
Numbers	Zero, one, two, three, four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen, twenty, thirty, forty, fifty, sixty, seventy, eighty, ninety, and hundred.
WH Question	What, when, where, which, who, whom, whose, why, how
Random Word	Not, no, a, an, and the

2.1.3. Convert words

It's a major and important method for this work. Because we used here "Using Synonyms" techniques to convert all changeable words. Most words have multiple meanings based on context, and we must consider the synonym that best reflects the correct meaning for the given situation. Example: "I love football.". After splitting it's like ["I", "love", "football"]. So here "I" is a pronoun, we know "love" is an abstract noun or a verb. But here "love" is a verb and "football" is a noun and an object. So here we cannot change "I" and "football" but we can change "love" by replacing them with an exactly similar word. So here, the natural language tool kit (NLTK) package has been used to accomplish the necessary preprocessing task. After replacing the word "love" our algorithm gives a new string "I enjoy reading". We can see another example: "I have ten takas", In this example, the sentence cannot change by our algorithm. As we see here all parts of speech are not changeable. Using the natural language tool kit (NLTK) we change all the changeable words.

2.1.4. Reassemble paragraphs

We reassemble entire paragraphs in the final phase. Every paragraph contains one or more sentences. That's why we need to reassemble paragraphs. Because at first, we split all words accordingly. After converting changeable words, we replace those words with their own indexes. Our algorithm pseudocode contains all the processes below.

Algorithm: Paraphrasing and Similarity Checking Algorithm

Input: "stringData" is the input paragraph. We make six lists of prepositions, WH-Question, numbers, auxiliary verbs, pronouns, random words.

Output: After applying the following procedure we get a new paragraph and similarity result between old text and New text.

a. Paraphrasing function

Step 1: START

Step 2: INPUT STRING stringData
Step 3: LISTING STRINGS getChangedSentence() (unchanged parts of speech)
preposition, pronouns, auxiliary verb, wh-question, numbers, and random words
Step 4: SPLIT stringData to WORDS wordlist[]
Step 5: Check wordlist[] with getChangedSentence()
- If word match with unchanged words, it doesn't need change
- Else word need change
Step 6: Rest wordlist[] CHNAGE WITH getSynonym(word) using Wordnet and NLTK Tools
Step 7: newString = Marge wordlist[] in its own index
Step 8: DISPLAY newString
Step 9: COMPARE newString with stringData SIMILARITY USING SIMILARITY CHECKING ALGORITHM
Step 10: STOP

b. Similarity checking function

Step 1: START
Step 2: INPUT STRING stringData
Step 3: INPUT STRING newString
Step 4: SPLIT stringData ODW[]
-Words count ODC
Step 5: SPLIT newString NDW[]
-Words Count NDC
Step 6: COMPARE ODW[] with NDW[]
Similar Words Count SWC
Step 7: CHECK Similarity
Similarity =(ODC-SWC)/ODC
Step 8: DISPLAY Similarity
Step 9: STOP

Following this, we compare two paragraphs for similarity. We check the old paragraph and New paragraph similarity percentages using our similarity checking method. Here, matchcount is a variable that counts the matched words. len(wordList) contains the main paragraph length. Finally, we subtract the main paragraph word length to match the words. And divided by len(wordList). After counting similarity, we convert it into a percentage multiplied by a hundred. Finally, we subtract the presentence between 100. At last, we get our similarity presentence between the old paragraph and the new paragraph.

3. RESOURCES USED

3.1. English wordnet

Synonymy is the most common relationship between terms in WordNet [29]. Open English WordNet is a lexical networking system for the English language that divides words into synsets and connects them using meronymy, hypernymy, and antonymy interactions. It's designed for use in natural language processing applications and presents deep lexical information about just the English language inside the graphical form. We imported the English WordNet to download the Natural language Tool Kit [30] in our program. We use this tool to convert changeable words.

3.2. Sentence structure

In this research, we work in the English language that's why we need to read English grammar more and more. English grammar is a book containing a description of the rules of the English language. At this point here we discuss sentence functionality and its structures.

A sentence is a group of words at least contain a subject and a verb and makes complete sense by itself. It may also include an object or a compliment, and the words must be appropriately ordered. Every sentence can be divided into two parts: Subject and Predicate. The person or thing about or which something is said in a sentence is called the subject. And what is said about the subject in a sentence is called a predicate. Example: "The sky is blue", Here "The sky" is a subject, and "is blue" is a predicate.

We know that there are three types of sentences according to the structure. There is a simple sentence, a complex sentence and the last one is a compound sentence. Here we discuss the three types of structures of sentences. A simple sentence contains a subject, a finite verb, and an object is called a simple sentence. Example: "They play Football", here "They" is a subject, "play" is a finite verb and last "football" is an object of this sentence. A compound sentence is a sentence in which two or more principal clauses relate to coordinating conjunction. Here we can see another example: "I went to Dhaka and meet my uncle".

In this sentence “I went to Dhaka” and “I meet my uncle” are the two principal clauses and they are added by one conjunction “and”. On the other hand, a sentence that has a principal clause and a subordinate clause is called a complex sentence. If we see an example, it is clear to identify complex sentences. “It is unbelievable what the magician showed us yesterday”. Now we are clear on how we can handle a paragraph’s sentence and its structures.

3.3. Parts of speech

After gaining information about sentences and their types of the structure now we need to discuss Parts of Speech. We know that every sentence contains parts of speech. Because every part of speech is very important for any sentence also our research. The sentence is a ‘Combination of words.’ on the other hand the parts of speech are ‘classes of words’.

In the English language, there are eight parts of speech: noun, pronoun, verb, adjective, adverb, preposition, conjunction, and interjection. The parts of speech identify how a word acts both grammatically and in terms of meaning inside a sentence. Whenever employed in different situations, a single word can serve as more than one element of speech. When using a dictionary, knowing the parts of speech is crucial for finding the accurate definition of a word. So, now we discuss every part of the speech. In every sentence, we get one or more nouns. The name of a thing, place, person, or concept is a noun. Articles (the, a, and an) are frequently used together with nouns, though not always. Common nouns don't really begin with an uppercase, but proper nouns must. Plural and Singular nouns, as well as abstract and concrete nouns, are all possible. Possession is indicated by adding's to nouns. Within such a statement, nouns can play a variety of roles, including subject complement, subject, indirect object, direct object, and object of a preposition. Example: Dhaka and horse. On the other hand, the word which is used instead of a noun is called a pronoun. It is the predecessor of a pronoun when it is used to replace a particular noun. The predecessor for the pronoun "she is the girl" in the preceding sentence.

The pronoun "She" is used here. Personal pronouns relate to distinct things or people, while possessive pronouns imply provenance, reflexive pronouns highlight the other noun or pronoun, relative pronouns provide a subordinate clause, and demonstrative pronouns define, refer to, or point to nouns. A verb is a word or a group of words that show what someone or something does. In short, a verb indicates an action or an event, or a state. A primary verb is present, as well as one or more supporting verbs. ("She can sing.") The major verb is sing, and the supporting verb is can) In terms of number, a verb must correspond to its subject. To represent tense, verbs manifest in various. A noun or pronoun’s place, quality, number, or quantity is described by an adjective. It generally responds to questions such as what kind, how many, or which one there are. A word that changes an adjective, an adverb, or some other verb is called an adverb. It usually responds to the following points: how, where, when, why, under what circumstances, and to what extent. A preposition is a word often placed before a noun or a pronoun to show its relationship with another noun on pronoun or to some other word in a sentence. (till tonight, by the tree, among our buddies, discussing the book). As a result, a preposition appears in every preposition-al sentence. Most often, the prepositional phrase serves as an adverb or an adjective. The most used prepositions are listed in Table 2. Conjunction seems to be a word that connects two or more phrases, sentences, or words to show the link between both components. but, or, and, nor, yet, so, for are all regulating conjunctions that link grammatically similar items. Although, while, since, because, and other subordinating conjunctions join sentences that are not similar. There are also other kinds of conjunctions. The words that express sudden feelings of mind like joy, sorrow, surprise, hatred, and exclamation. are called interjection. It's frequently followed with an exclamation mark. We have shown in Table 2 that all parts of speech of a sentence include all parts of speech. After this study, we are very clear about sentence structure and parts of speech with those examples to continue our research to build a paraphrasing algorithm and after paraphrasing, this algorithm also calculates the similarity between two documents.

Table 2. Identify all parts of speech for an example

Example: The young girl brought me a very long letter from the teacher, and then she quickly disappeared. Oh my!	
Parts of Speech	Words
Noun	Girl, letter, teacher
Pronoun	Me, she
Verb	Brought, disappeared
Adjective	Young, long
Adverb	Very, then, quickly
Preposition	from
Conjunction	and
Interjection	Oh my!

4. EXPERIMENT

4.1. Dataset

On a test set of 250 randomly chosen paragraphs from the internet, we put our algorithm to the test. This test dataset has 250 paragraphs. Because certain words and parts of speech in English are not transferable. As a result, we manually double-checked the unchangeable words which showed in Table 1. And after collecting the dataset we apply our algorithm.

4.2. Implementation

Algorithm was written in Python 3.8 and all the process is shown in Figure 2. Our python script scans every sentence in the database in the first step. The essential preprocessing activities were completed using the natural language tool kit (NLTK) package. Within the natural language tool kit (NLTK) package, we have used split() to split parts of speech for each word of the sentence. Then the unchangeable words are checked by the list. After checking all changeable words are converted using NLTK. Then our algorithm reassembles the whole paragraphs. Another python script has been used to check similarities between old paragraphs and algorithm-generated new paragraphs.

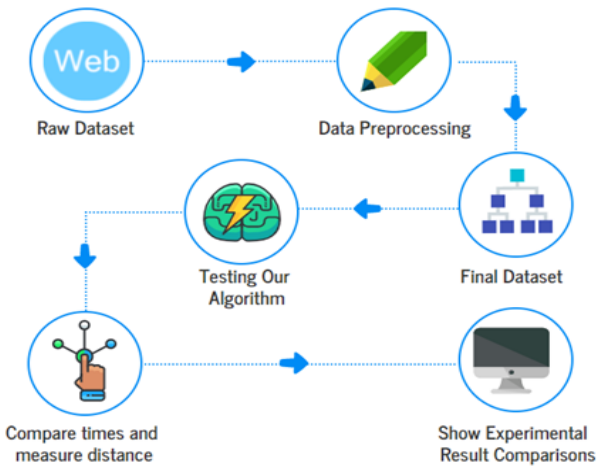

Figure 2. Experimental diagram

5. RESULTS AND ANALYSIS

In our approach, we paraphrase the whole paragraph keeping the right sentence structures and periodic parts of speech. This research has both advantages and cons. One of the major advantages is that our algorithm converts every changeable word with its proper synonym and replaces its own index also. This approach has a critical shortcoming in that it relies on WordNet and NLTK. Our algorithm works on a maximum of 1000 words. In Table 3. Shows the input and output of our algorithm. We input a sentence to our algorithm and the algorithm generates the output of a new para-phrased text and also measures the similarity in percentage between the two texts.

Any paragraph can be paraphrased using our technique. It succeeded at paraphrasing 237 paragraphs out of 250. On the other hand, it converts all changeable words properly. Leading to a shortage of synonyms in the NLTK package, our algorithm fails for 13 paragraphs. For the dataset, our algorithm's accuracy is 94.8%. Table 4. shows a comparison of the time limits of several paraphrase tools and techniques. Table 4. shows that our approach produces superior results and adheres to time constraints.

Table 3. Input and output of the algorithm

Input Text	Output	Paraphrased
The young girl brought me a very long letter from the teacher, and then she quickly disappeared.	The young girl presented me with a lengthy epistle from the instructor, and then she rapidly faded.	45.36%

We show our analysis in a graph chart Figure 3. Using the synonyms technique, we get the best result from other paraphrasing tools because they use Grammatical structure and Change the form of words techniques. And, we get the maximum accurate meaning of the sentence. We know that "grammatical structure and changing the form of words techniques", those techniques don't give accurate meaning to any sentence or paragraph and also do work properly for large sentences.

We see after using those techniques sentences or text documents break grammatical rules as well as their main structure. But our algorithm which means the synonyms technique gives the best result in any text document. Because it is not changing any grammatical rules. In Table 5, we compare our algorithm with the best online document paraphrasers and get the best result. We take our dataset to calculate this result. After analyzing this process, we get our algorithms paraphrasing time, and it's better than the other online tools. We compare our similarity measurement function also with cosine similarity [31], Jaccard Similarity [32] in Table 5.

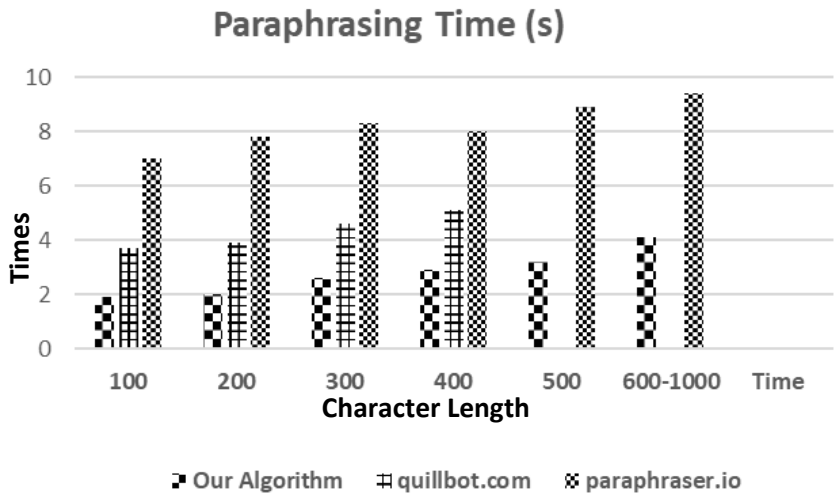

Figure 3. Bar graph for comparisons

Table 4. Compare with other paraphrasing tools

Characters	Paraphrasing Time		
	Our Algorithm	quillbot.com	paraphraser.io
100	1.9s	3.7s	7.0s
200	2.0s	3.9s	7.8s
300	2.6s	4.6s	8.3s
400	2.9s	5.1s	8.0s
500	3.2s	×	8.9s
600-1000	4.1s	×	9.4s

Table 5. Compare similarity measurements with our similarity measurement function (SFM)

Similarity Measurements in Percentance	
Similarity measurement function (SMF)	45.36%
Cosine similarity	41.12%
Jaccard similarity	39.14%

6. CONCLUSION

For the English language, we developed an algorithm. Using the synonyms dictionary dataset from WordNet and Natural Language Tool Kit. To keep sentence structures and periodic parts of speech consistent, English grammar rules were applied. To paraphrase a new text document, we've covered practically every type of text document. We will continue to improve this algorithm in the future to improve accuracy, work on other parts of speech, and also increase character limits.

REFERENCES

[1] R. Barzilay and K. McKeown, "Extracting paraphrases from a parallel corpus," *In Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, ACL* '01, pp. 50–57, 2001, doi: 10.3115/1073012.1073020.

[2] F. M. Prentice and C. E. Kinden, "Paraphrasing tools, language translation tools and plagiarism: an exploratory study," *International Journal for Educational Integrity*, vol. 14, no. 1, pp. 1-16, 2018, 10.1007/s40979-018-0036-7.

[3] A. M. Rogerson and G. McCarthy, "Using Internet based paraphrasing tools: Original work, patchwriting or facilitated plagiarism?," *International Journal for Educational Integrity*, vol. 13, no. 2. 2017, 10.1007/s40979-016-0013-y.

[4] X. Liu, L. Mou, F. Meng, H. Zhou, J. Zhou, and S. Song, "Unsupervised paraphrasing by simulated annealing," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 302-312, 2020, doi: 10.18653/v1/2020.acl-main.28.

[5] S. D. Sulistyaningru and N. Inayah, "Employing Online Paraphrasing Tools to Overcome Students' Difficulties in Paraphrasing," *In STAIRS: English Language Education Journal*, vol. 2, no. 1, pp. 52-59, 2021.

[6] J. P. Wahle, T. Ruas, T. Foltýnek, N. Meuschke, and B. Gipp, "Identifying Machine-Paraphrased Plagiarism," *International Conference on Information, iConference 2022: Information for a Better World: Shaping the Global Future*, vol. 13192, pp. 393-413, doi: 10.1007/978-3-030-96957-8_34.

[7] D. N. Chi and X. N. C. M. Nguyen, "Paraphrasing in academic writing: A case study of Vietnamese learners of English," *Language Education in Asia*, vol. 8, no. 1, pp. 9-24, 2017, doi: 10.5746/leia/17/v8/i1/a02/na_mai.

[8] A. B. Siddique, S. Oymak, and V. Hristidis, "Unsupervised paraphrasing via deep reinforcement learning," In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1800-1809, 2020, doi: 10.1145/3394486.3403231.

[9] F., 2021. Updates, Insights, and News from FutureLearn | Online Learning for You. FutureLearn.*https://www.futurelearn.com/info/courses/english-for-study-intermediate/0/steps/35241*

[10] A. Roy, and D. Grangier, "Unsupervised paraphrasing without translation," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6033-6039, 2019, doi: 10.18653/v1/P19-1605.

[11] R. Barzilay and L. Lee, "Learning to paraphrase: An unsupervised approach using multiple-sequence alignment," In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 16-23, 2003.

[12] R. Sato, M. Yamada, and H. Kashima, "Re-evaluating Word Mover's Distance," *Proceedings of the 39th International Conference on Machine Learning*, vol. 162, pp. 19231-19249, 2022.

[13] I. Dokmanic, R. Parhizkar, J. Ranieri and M. Vetterli, "Euclidean Distance Matrices: Essential theory, algorithms, and applications," in *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 12-30, Nov. 2015, doi: 10.1109/MSP.2015.2398954.

[14] T. V. D. Cruys, S. D. Afantenos, and P. Muller, "MELODI: A supervised distributional approach for free paraphrasing of noun compounds," In *7th International Workshop on Semantic Evaluation (SemEval 2013) in: 2nd Joint Conference on Lexical and Computational Semantics (SEM 2013)*, June 2013.

[15] S. Witteveen and M. Andrews, "Paraphrasing with large language models," in *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pp. 215–220, Nov. 2019, doi: 10.18653/v1/D19-5623.

[16] A. Alassir, S. Dardour and H. Fehri, "Paraphrasing Tool Using the NooJ Platform," *Springer International Publishing: In International Conference on Automatic Processing of Natural-Language Electronic Texts with NooJ*, pp. 163-173, June 2021.

[17] J. Ganitkevitch, C. Callison-Burch, C. Napoles, and B. Van Durme, "Learning sentential paraphrases from bilingual parallel corpora for text-to-text generation," In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 1168-1179, July 2011.

[18] R. Cao *et al*., "Unsupervised dual paraphrasing for two-stage semantic parsing," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6806–6817, July 2020, doi: 10.18653/v1/2020.acl-main.608.

[19] G. Ponkiya, R. Murthy, P. Bhattacharyya, and G. Palshikar, "Looking inside Noun Compounds: Unsupervised Prepositional and Free Paraphrasing using Language Models," In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 4313-4323, November 2020, doi: 10.18653/v1/2020.findings-emnlp.386.

[20] Q. Du and Y. Liu, "Foregrounding learner voice: Chinese undergraduate students' understanding of paraphrasing and source use conventions for English research paper writing," *Language Teaching Research*, p.13621688211027032, 2021, doi: 10.1177/13621688211027032.

[21] I. Böschen, "Evaluation of JATSdecoder as an automated text extraction tool for statistical results in scientific reports," *Scientific Reports*, vol. 11, no. 1, p. 19525, pp. 1-12, 2021, doi: 10.1038/s41598-021-98782-3.

[22] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Systems with Applications*, vol. 165, p. 113679, March 2021, doi: 10.1016/j.eswa.2020.113679.

[23] T. Waltzer and A. Dahl, "Students' perceptions and evaluations of plagiarism: Effects of text and context," *Journal of Moral Education*, vol. 50, no. 4, pp. 436-451, 2021, doi: 10.1080/03057240.2020.1787961.

[24] E. Hovy and D. Marcu, "Automated text summarization," *The Oxford Handbook of computational linguistics*, p. *583598*, 2005.

[25] I. Mani, D. House, G. Klein, L. Hirschman, T. Firmin, and B. M. Sundheim, "The TIPSTER SUMMAC text summarization evaluation," In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 77-85, June 1999.

[26] H. R. Bernard, and G. Ryan, "Text analysis," *Handbook of methods in cultural anthropology*, vol. 613, 1998.

[27] J. C. Eichstaedt *et al*., "Closed-and open-vocabulary approaches to text analysis: A review, quantitative comparison, and recommendations," *Psychological Methods*, vol. 26, no. 4, p. 398, 2021, doi: 10.1037/met0000349.

[28] M. Bednarek and G. Carr, "Computer-assisted digital text analysis for journalism and communications research: Introducing corpus linguistic techniques that do not require programming," *Media International Australia*, vol. 181, no. 1, pp. 131-151, 2020, doi: 10.1177/1329878X20947124.

[29] C. Fellbaum, "WordNet and wordnets," In *Brown, Keith et al. (eds.), Encyclopedia of Language and Linguistics, Second Edition, Oxford: Elsevier*, pp. 665-670, 2005.

[30] Bird, Steven. "NLTK: the natural language toolkit." In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, pp. 69-72. 2006.

[31] B. Li and L. Han, "Distance weighted cosine similarity measure for text classification," In *International conference on intelligent data engineering and automated learning: Springer Berlin Heidelberg*, pp. 611-618, October 2013.

[32] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of Jaccard coefficient for keywords similarity," In *Proceedings of the international multiconference of engineers and computer scientists*, vol. 1, no. 6, pp. 380-384, March 2013.

BIOGRAPHIES OF AUTHORS

Jabir Al Nahian received the Bachelor's degree in Computer Science and Engineering from Daffodil International University, Bangladesh. He is currently Researcher in the Computational Intelligence Lab, Bangladesh specializing in providing machine learning solutions for expertise. His current research interests lie in the area of data science, natural language processing, machine learning, deep learning and particularly in areas pertaining to their application for the Bangla language. He is an active researcher and reviewer several International conferences and journals. He can be contacted at email: jabirnahian009@gmail.com.

Abu Kaisar Mohammad Masum completed his B.Sc. from Daffodil International University (DIU) in Bangladesh. Now he is a Lecturer in the Dept. of CSE, DIU. Previously he worked as Research Assistant (RA) in Apurba-DIU Research & Development Lab. He has a number of Scopus indexed publications in international and national journals and conference proceedings. Broadly, his methodological research focuses on the Application of Machine Learning, Natural Language Processing (NLP), and Data mining. He currently works on different areas of NLP and Adaptive algorithms. Mr. Masum has received Best Researcher Award-2021 organized by DIU and awarded as Best Performed Faculty Member of DIU. He also received 'In Recognition of Scholar Publication in Reputed Indexed Journal' award for the year of 2019 by DIU. Mr. Masum is a Supervisor of the DIU - NLP and Machine Learning Research LAB. He can be contacted at email: abu.cse@diu.edu.bd.

Muntaser Mansur Syed is a PhD candidate in the department of Computer Science and Engineering at the Florida Institute of Technology. His research interests include machine learning on edge devices; Internet of Things and distributed sensor networks. Muntaser has previously interned at Goldman Sachs and Nvidia, in the latter as a Machine Learning GPU advocate. Muntaser is also an avid participant at hackathons, having competed in over 200 such events over the past four years. He can be contacted at email: msyed2011@my.fit.edu.

Seikh Abujar has obtained his M.Sc. degree in Computer Science from Jahangirnagar University, Bangladesh. Previously he completed his B.Sc. degree in Computer Science and Engineering at Daffodil International University (DIU), Bangladesh. Currently pursuing his PhD in Computer Science at Florida Institute of Technology in Melbourne, Florida. At present he is working as a Lecturer in the department of Computer Science and Engineering in Independent University, Bangladesh (IUB). Prior to joining IUB, he was a faculty member at DIU from September 2017 to January 2021 and at Britannia University (BU) from February 2015 to August 2017. He has a number of articles published in several Scopus Indexed proceedings of IEEE, Springer and Elsevier, and has served on several International conferences and journals as a reviewer. Broadly, his methodological research focuses on Application of machine learning and Data mining. He currently works on different areas of Natural Language Processing and Adaptive algorithms. Many of Mr. Abujar's and his students' research work have won "Best Research Paper - Award" in many International Conferences. During his career at DIU, he was Founder and Supervisor of the DIU-NLP and Machine Learning Research LAB. He can be contacted at email: sabujar2021@my.fit.edu.