International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I7P128

Volume 72 Issue 7, 260-265, July 2024
© 2024 Seventh Sense Research Group®

Original Article

Analysis of Various Measures of Text Similarity for Comparing Topics of Computer Science Syllabuses

Ritu Sodhi[1], Jitendra Choudhary[2], Ritu Jain[3], Ruby Bhatt[4], Ritesh Joshi[5], Anil Patidar[6]

[1,6] Computer Applications, Medi-Caps University, Indore, M.P., India.
[2,4] Computer Science, Medi-Caps University, Indore, M.P., India.
[3]Department of Computer Science, ISU-Engineering, ITM Skills University, Kharghar, Navi Mumbai, Maharashtra, India.
[5]Computer Applications, Ganpat University, Mehsana, Gujrat, India.

[1]Corresponding Author : ritu.sodhi@medicaps.ac.in

Received: 06 March 2024 Revised: 06 June 2024 Accepted: 02 July 2024 Published: 26 July 2024

Abstract - *Text similarity measures are used to find out how much different texts are similar. There is a need to compare text for document comparison, text classification, text summarizing, information retrieval, question-answer sessions, clustering documents, etc. There is also a need to compare computer science terms; while plagiarism checks, website contents, comparing syllabuses of the same subject, notes, books, etc. This research focused on the text similarity measures to compare text related to computer science terms. This research executed some of the lexical and semantic similarity measures for comparing topics of the syllabus of programming using Python. And found after executing various approaches that spacy using a large English model and cos_similarity together gives a better result. In the future, this research can be improved by including more similarity measures and by increasing the size of the dataset for comparison of computer science terms.*

Keywords - *Computer science, Python, Spacy, Syllabus, Text similarity.*

1. Introduction

One of the most significant and challenging methods in the field of artificial intelligence is Natural Language Processing (NLP); numerous applications of NLP require the outcomes of text mining [1,2]. There are many applications where the comparison between texts is needed [3,4,5]. The applications where sentence similarity is needed are the comparison of various documents, text summarization [6], text classification, question-answer retrieval [7], clustering documents, etc. Lexical and semantic analysis [8] is an important feature in natural language processing.

Most of the semantic analysis models have been developed only in English and European languages [9]. The Word2vec technique is used for semantic analysis [10].

1.1. Problem Statement

There are different applications where a comparison of the text related to computer science terms is needed, such as comparing different website contents, notes, syllabuses, plagiarism, etc.

1.2. Research Gap and Comparison with Existing Research Findings

Less research focuses on the comparison of computer science terms. There is a need to find the text similarity in the computer science syllabuses to find the topics common in various syllabuses.

1.3. The Novelty of Work

This work will help faculties and others to view the common topic of computer science syllabuses of different universities for their reference. This paper tested and analyzed the results of various text similarity measures available on the topics of the programming syllabus. Available text similarity measures are tested to see how they perform in computer science terms. This paper is organized as follows: Section 2 discusses the background and related work. Section 3 contains what is the problem. Section 4 has a research hypothesis. Section 5 contains materials and methods; and discusses different text similarity measures and the code on which testing performed. Section 6 contains the results and its analysis performed on various similarity measures. Section 7 contains the conclusion.

2. Background and Related Work

According to information theory, two text samples are considered similar if they have something in common. The similarity increases with increased commonality and vice versa.

2.1. Text Distance

It provides an insight into the semantic similarity of two text words based on their distance. Distance can be measured in three different ways based on the object's length, distribution, and semantics: length, distribution, and semantics.

2.2. Text Representation

Lexical and semantic analysis is an important feature in natural language processing. Lexical similarity is the degree to of two given strings are similar in their character sequence. If the result is one, which means the words are completely lexically the same. Zero says that there is nothing common between given strings. The lexical similarity between texts can be compared using string-based methods. Semantic similarity tells the similarity between text and document based on their meaning. For example, the words "cook" and "hook" are very much lexically similar, but they are semantically different. The pair words "Suzuki" and "Scooter" are lexically different, but they are related semantically.

Instead of character-by-character matching, semantic similarity assesses the resemblance between a text and a document based on their meaning. Measures that are knowledge and corpus-based are used to calculate semantic similarity. The idea of the hybrid approach is to combine string-based, corpus-based, and knowledge-based approaches to give better results. When Maulud et al. compared the state-of-the-art NLP tools, they discovered that sophisticated semantic methods are fairly accurate [8]. Gomaa and Fahmy discussed various text similarity measures. Text can be similar lexically or semantically.

The lexical similarity between texts can be compared using string-based methods. Semantic similarities between texts can be found by using Corpus-based and knowledge-based (WordNet) algorithms. This research [5] discussed String based, Corpus-based, and knowledge-based methods in detail. Atoum et al. compared three techniques: corpus-based, knowledge-based, and hybrid methods. The result of the comparison is that hybrid methods give results that are better as compared to corpus-based and knowledge-based methods [4]. Achananuparp et al. evaluate 14 existing methods for the semantic similarity between texts. In a low-complexity data set, linguistic measures are better than word overlap and TF-IDF measures. Word overlap and TF-IDF measures perform better in high-complexity data sets [3]. Chen et al. examined the performance of sentence similarity measures in the biomedical domain. In this research, Researchers try to find out the effectiveness of sentence similarity measures on PubMed documents for sentence ranking. Their experimental results show that neither lexical nor semantic measures provide the desired results for sentence ranking [11]. Quan et al. integrated semantic information, syntactic information, and the attention weight mechanism in a unified way and developed a new tree kernel, known as the ACVT kernel, that is used for sentence similarity [12]. Peng et al. Propose an Enhanced Recurrent Convolutional Neural Network (Enhanced-RCNN) model for sentence similarity. The architecture of Enhanced-RCNN is less complex as compared to the BERT model. According to Experimental results, Enhanced-RCNN outperforms the baselines and it also achieves competitive performance on two real-world datasets [13]. Above mentioned research is focused on similarity measures on various datasets. However, how they perform the comparison on the datasets related to computer science terms is the question for research.

3. Research Motivation

Various datasets on which the text similarity measures are well tested. However, they are not well focussed on how text similarity measures perform on the computer science terms dataset. There is a need to find out how the different similarity measures work on the dataset for computer science terms.

Table 1. Programming syllabus topics on which test similarity methods are tested

Topic 1	Topic 2
Formal Parameters	Formal Arguments
Method	Function
Loop	Iteration
Function Declaration	Function Prototype

Table 2. Text similarity measures applied to programming syllabus topics

1. Using spacy and similarity function	7. Using SentenceTransformer (all-mpnet-base-v2) and scipy.spatial	13. Using SentenceTransformer (nli-distilroberta-base-v2) and torch.nn
2. Using SentenceTransformer (distilbert-base-nli-mean-tokens) and util	8. Using SentenceTransformer (all-MiniLM-L6-v2) and scipy.spatial	14. Using tensorflow_hub and scipy.spatial
3. Using SentenceTransformer (all-mpnet-base-v2) and util	9. Using SentenceTransformer (nli-distilroberta-base-v2) and scipy.spatial	15. Using tensorflow_hub and torch.nn
4. Using SentenceTransformer (all-MiniLM-L6-v2) and util	10. Using SentenceTransformer (distilbert-base-nli-mean-tokens) and torch.nn	16. Jaccard Similarity
5. Using SentenceTransformer (nli-distilroberta-base-v2) and util	11.Using SentenceTransformer (all-mpnet-base-v2) and torch.nn	17. Using spacy and euclidean_distance
6. Using SentenceTransformer (distilbert-base-nli-mean-tokens) and scipy.spatial	12. Using SentenceTransformer (all-MiniLM-L6-v2) and torch.nn	18. Using spacy and cos_similarity

4. Research Hypothesis

This research will initiate the focus of different researchers to start focusing and testing how the text similarity measures work in computer science terms datasets.

5. Materials and Methods

This paper, executed and analyzed some of the text similarity measures for comparing the topics of the syllabus of programming. Table 1 contains some topics that have been compared. Topic 1 of the same row is compared with topic 2. For example, Formal Parameters are compared with Formal Arguments, etc. The text similarity measures executed are given in Table 2.

The first approach uses spacy and similarity functions. The code is given below.

```
import spacy
import spacy_sentence_bert
nlp = spacy.load('en_core_web_lg')
a1 = "Formal Parameters"
a2="Formal Arguments"
s1 = nlp(a1).similarity(nlp(a2))
print(a1 + "  " +a2 +" are similar " ,end=" ")
```

```
from sentence_transformers import
SentenceTransformer,util
import numpy as np
model = SentenceTransformer('distilbert-base-nli-mean-
tokens')
sentence1 = "Formal Parameters"
sentence2 = "Formal Arguments"
# Encode sentences to get their embeddings
embedding1 = model.encode(sentence1,
convert_to_tensor=True)
embedding2 = model.encode(sentence2,
convert_to_tensor=True)
# compute similarity scores of two embeddings
cosine_scores = util.pytorch_cos_sim(embedding1,
embedding2)
print("Sentence 1:", sentence1)
print("Sentence 2:", sentence2)
print("Similarity score:", cosine_scores.item())
```

The second approach is SentenceTransformer (distilbert-base-nli-mean-tokens) and util. In the approach, sentence transformer embedding is used, and the pytorch_cos_sim() method is used to find semantic similarity. The code is given above. Executed the second approach also on all-mpnet-base-v2, all-MiniLM-L6-v2, and nli-distilroberta-base-v2 models. Similarly, different embeddings and different lexical and semantic text similarity methods were used to compare the topics given in Table 1.

6. Results and Discussion

Python language for testing different text similarity measures on topics of the programming syllabus is used. The results of Text Similarity measures on different texts are given in Table 3. The first numeric value, 0.751400293, in Table 3 indicates that the Formal Parameter is 75% similar to the Formal Argument using spacy and similarity functions. Figure 1(a) shows the results of text similarity approaches for comparison of Formal Parameters and Formal Arguments, Figure 1(b) for Method and Function, Figure 1(c) for Loop and Iteration, and Figure 1(d) for Function Declaration and Function Prototype. Figure 1(e) shows which color shows which similarity measure in Figures 1(a), 1(b), 1(c), and 1 (d).

6.1. How and why the Results are Better Using Spacy and Cos_Similarity Measures

The result of spacy and cos_similarity gives results that are closer to 1. Because execution of the measures on semantic similar topics, conclude that spacy and cos_similarity together give better results as compared to the other measures for comparing computer science terms.

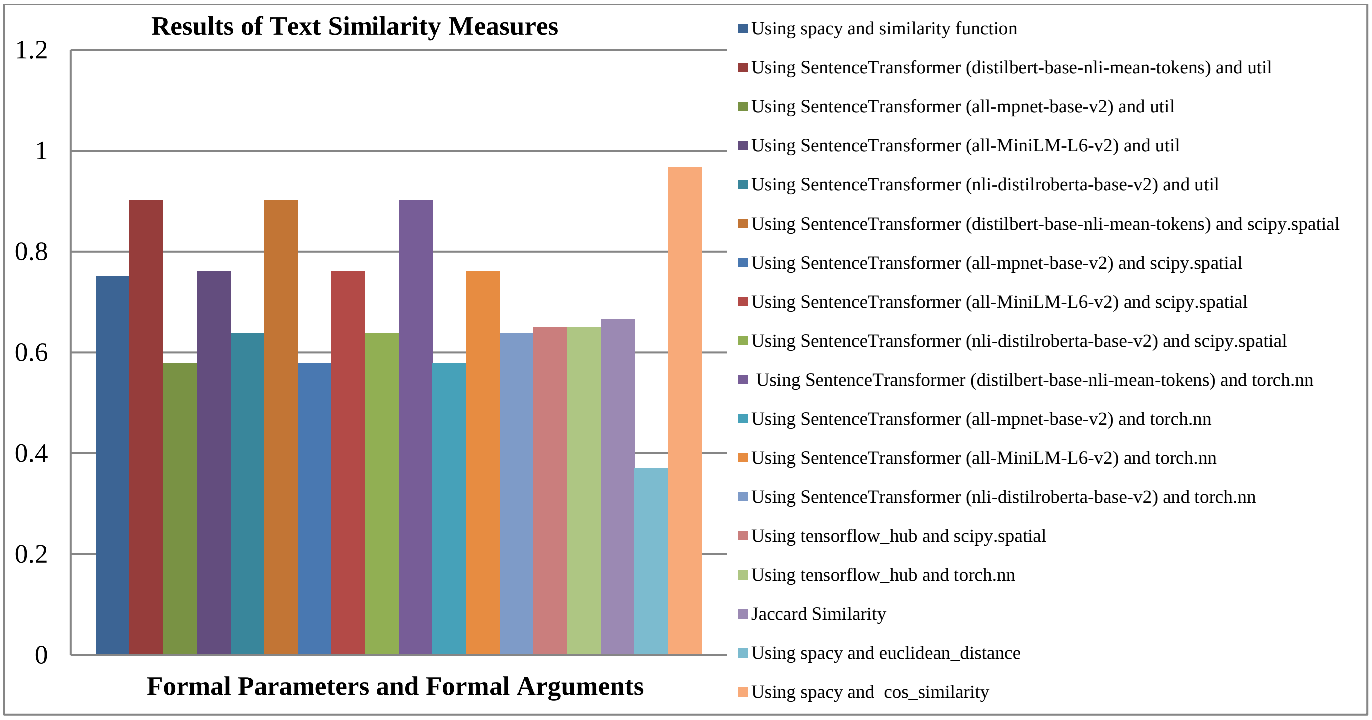
6.2. Limitations on Result Analysis

These text similarity measures are tested on only a limited dataset given in Table 1 related to computer science. The results may vary depending on the increasing data set.In this research, 18 different similarity measures on computer science terms are executed. This research is useful where there is a need to compare different computer science terms such as comparing syllabuses, checking plagiarism, comparing contents of websites, tutorial notes, etc. This research can be improved by including more similarity measures, testing similarity measures on large data sets, creating models for computer science terminologies, etc.

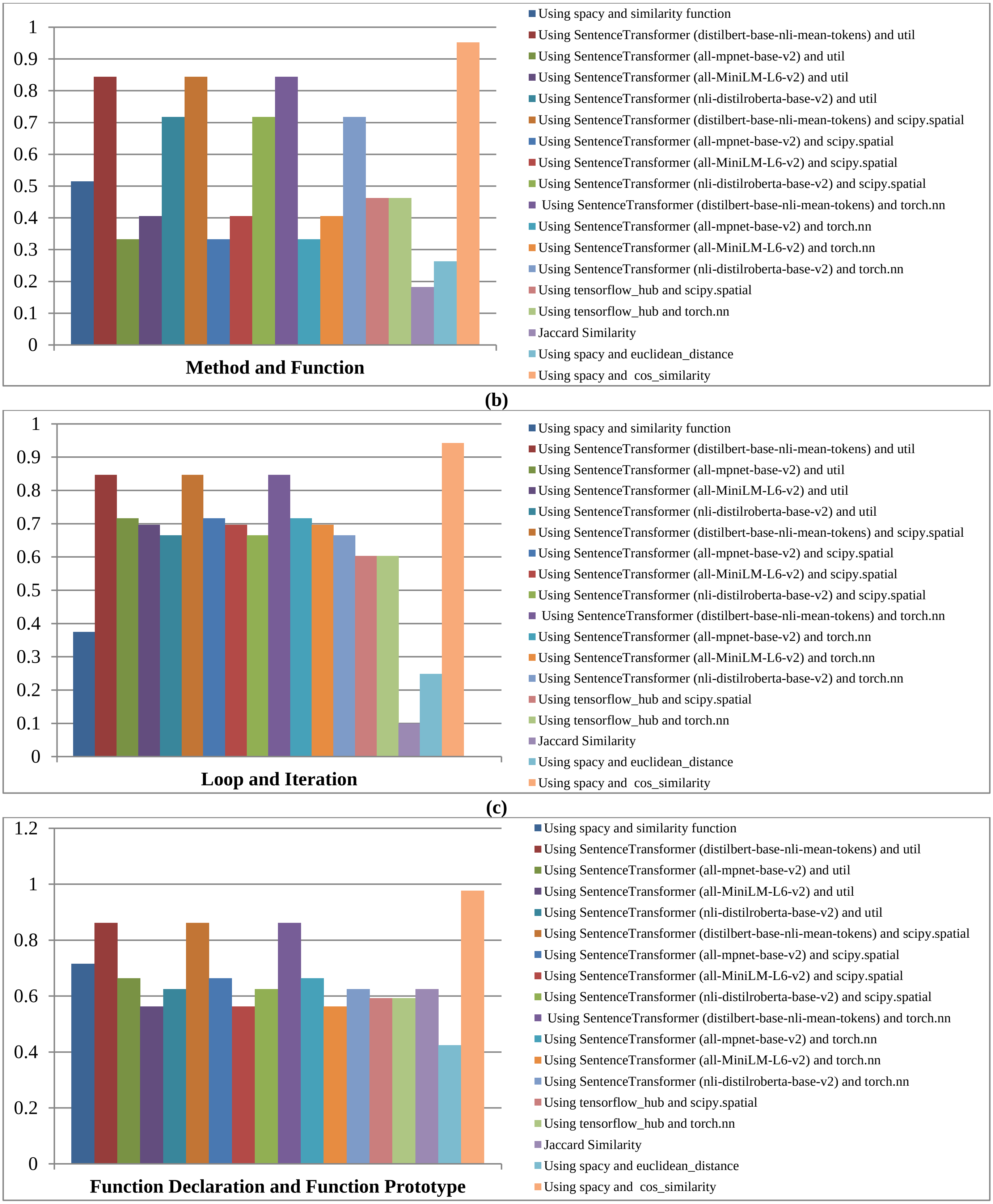
Table 3. Results of text similarity measures on programming syllabus topics

Topics Compared **Text Similarity Measures**	**s1=Formal Parameters** **s2=Formal Arguments**	**s1=Method** **s2=Function**	**s1=Loop** **s2=Iteration**	**s1= Function Declaration** **s2 = Function Prototype**
Using spacy and similarity function	0.751400293	0.514936602	0.374452757	0.71488921
Using SentenceTransformer (distilbert-base-nli-mean-tokens) and util	0.901824355	0.843984008	0.846786261	0.860680461

Using SentenceTransformer (all-mpnet-base-v2) and util	0.579260707	0.332026988	0.716405988	0.663579583
Using SentenceTransformer (all-MiniLM-L6-v2) and util	0.761538088	0.405633152	0.697149873	0.561831594
Using SentenceTransformer (nli-distilroberta-base-v2) and util	0.639154911	0.718146443	0.665073335	0.623886287
Using SentenceTransformer (distilbert-base-nli-mean-tokens) and scipy.spatial	0.901824236	0.843984663	0.846786082	0.860680461
Using SentenceTransformer (all-mpnet-base-v2) and scipy.spatial	0.579260409	0.332026839	0.716406226	0.663579285
Using SentenceTransformer (all-MiniLM-L6-v2) and scipy.spatial	0.761538088	0.405633122	0.697149634	0.561831594
Using SentenceTransformer (nli-distilroberta-base-v2) and scipy.spatial	0.63915503	0.718146622	0.665073097	0.62388593
Using SentenceTransformer (distilbert-base-nli-mean-tokens) and torch.nn	0.9018	0.844	0.8468	0.8607
Using SentenceTransformer (all-mpnet-base-v2) and torch.nn	0.5793	0.3320	0.7164	0.6636
Using SentenceTransformer (all-MiniLM-L6-v2) and torch.nn	0.7615	0.4056	0.6971	0.5618
Using SentenceTransformer (nli-distilroberta-base-v2) and torch.nn	0.6392	0.7181	0.6651	0.6239
Using tensorflow_hub and scipy.spatial	0.650274038	0.462715745	0.603160262	0.59192276
Using tensorflow_hub and torch.nn	0.6503	0.4627	0.6032	0.5919
Jaccard Similarity	0.666666667	0.181818182	0.1	0.625
Using spacy and euclidean_distance	0.370176863	0.263367262	0.248402597	0.423621895
Using spacy and cos_similarity	0.967	0.952	0.943	0.976



(a)



(b)

(c)

(d)

Fig. 1 Graphical representation of results of different text similarity measures.

7. Conclusion and Future Work

A comparison of computer science terms is needed while comparing syllabuses of the same subject, notes, website contents, books, plagiarism checks, etc. Different lexical and semantic text similarity measures exist. This paper tested 18 different text similarity measures on the topics related to the syllabus of the programming course.The topics that were experimented on are semantically similar. So, the measure that gives the result closer to value 1 is the better measure. In this analysis, it found that the use of spacy and cos_similarity together gives results that are closer to 1 for semantically similar topics. In the future, this research can be improved by testing more data sets and other text similarity measures.

References

[1] Xiaofang Liao, and Zijiang Zhu, "Classification of Natural Language Semantic Relations under Deep Learning," *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications*, Dalian, China, pp. 1025-1027, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Artem A. Maksutov et al., "Knowledge Base Collecting Using Natural Language Processing Algorithms," *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering*, St. Petersburg and Moscow, Russia, pp. 405-407, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Palakorn Achananuparp, Xiaohua Hu, and Xiajiong Shen, "The Evaluation of Sentence Similarity Measures," *Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science*, vol. 5182, pp. 305-316, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[4] Issa Atoum, Ahmed Otoom, and Narayanan Kulathuramaiyer, "A Comprehensive Comparative Study of Word and Sentence Similarity Measures," *International Journal of Computer Applications*, vol. 135, no. 1, pp. 10-17, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[5] Wael H. Gomaa, and Aly A. Fahmy, "A Survey of Text Similarity Approaches," *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13-18, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[6] Jiaxing Tan et al., "Sentence Retrieval with Sentiment-Specific Topical Anchoring for Review Summarization," *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore Singapore, pp. 2323-2326, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[7] Aliaksei Severyn, and Alessandro Moschitti, "Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks," *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, United States, pp. 373-382, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[8] Dastan Hussen Maulud et al., "State of Art for Semantic Analysis of Natural Language Processing," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 21-28, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Hamed Jelodar et al., "A Collaborative Framework Based for Semantic Patients-Behavior Analysis and Highlight Topics Discovery of Alcoholic Beverages in Online Healthcare Forums," *Journal of Medical Systems*, vol. 44, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Xiaolong Wang, Xingtong Dong, and Shuxin Chen, "Text Duplicated-Checking Algorithm Implementation Based on Natural Language Semantic Analysis," *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference*, Chongqing, China, pp. 732-735, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Qingyu Chen et al., "Sentence Similarity Measures Revisited: Ranking Sentences in PubMed Documents," *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, Washington DC, USA, pp. 531-532, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] Zhe Quan et al., "An Efficient Framework for Sentence Similarity Modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 853-865, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Shuang Peng et al., "Enhanced-RCNN: An Efficient Method for Learning Sentence Similarity," *Proceedings of The Web Conference 2020*, Taipei, Taiwan, pp. 2500-2506, 2020. [CrossRef] [Google Scholar] [Publisher Link]