A Hybrid Approach to Paraphrase Detection Based on Text Similarities and Machine Learning Classifiers

Conference Paper · May 2022

DOI: 10.1109/MIUCC55081.2022.9781678

2 authors:

Mena Hany
October University for Modern Sciences and Arts
1 PUBLICATION **1** CITATION
SEE PROFILE

Wael Gomaa
Beni Suef University
20 PUBLICATIONS **1,005** CITATIONS
SEE PROFILE

A Hybrid Approach to Paraphrase Detection Based on Text Similarities and Machine Learning Classifiers

Mena Hany
Faculty of Computer Science
MSA University
Giza, Egypt
mena.hany@msa.edu.eg

Wael H. Gomaa
Faculty of Computer and Artificial Intelligence, Beni-Suef University
Faculty of Computer Science, MSA University
Egypt
wahassan@msa.edu.eg

2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-6677-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/MIUCC55081.2022.9781678

Abstract—In the realm of natural language processing (NLP), paraphrase detection is a highly common and significant activity. Because it is involved in a lot of complicated and complex NLP applications like information retrieval, text mining, and plagiarism detection. The proposed model finds the best combination of the three types of similarity techniques that are string similarity, semantic similarity and embedding similarity. Then, inputs these similarity scores that range from 0 to 1, to the machine learning classifiers. This proposed model will be benchmarked on "the Microsoft research paraphrase corpus" dataset (MSRP) and from this approach for paraphrase detection problem, the accuracy acquired is 75.78% and F1-Score of 83.01%.

Index Terms—Paraphrase Detection, Text Similarity, Machine Learning, Natural Language Processing

I. INTRODUCTION

NLP is the field of making the computer understand, use and produce the human language and it is used in many things like chatbots and plagiarism detection systems. NLP has many topics and ideas for projects that can help people automate their day-to-day lives. One of these topics is paraphrase detection. The ability to determine if two statements have the same semantic meaning or not is known as paraphrase detection and paraphrase detection is one of the basic operations in NLP that is involved in many more complicated NLP operations like information retrieval, text mining and plagiarism detection, although it is a basic operation of many NLP tasks. It is not an easy or trivial problem to solve and many old researchers faced many problems in solving the paraphrase detection to use it in other NLP operations. And one day-to-day life example of systems that use paraphrase detection is Turnitin to help people like teachers and professors [1]. And for example when teachers are revising the research or assignments of their students. They could wish to see if this research or homework is plagiarized from an online source or if the students are plagiarizing from one another. And here comes the great use of paraphrase detection which is the most important step in plagiarism detection. The paraphrase detection model will be formed from the combination of NLP techniques and machine learning techniques. This paper will mainly focus on paraphrase detection between two sentences. The method that will be utilized is to compare two sentences and then use machine learning techniques to determine whether one of the statements is paraphrased from the other. There are generally three main techniques of NLP that are used to measure the likeness of two sentences:

First, the string similarity techniques usually work on measuring the similarity score by comparing the characters of the sentence with the other sentence or by comparing the whole sentence with the other sentence [2].

Second, the semantic similarity techniques usually work on measuring the similarity score by comparing the meaning of the whole sentence with the other one. This comparison usually returns to a dictionary or something like Wikipedia or WORDNET to take the meaning of the sentences and make the comparisons [2].

Third, the embedding similarity techniques usually work on measuring the similarity score by converting the two sentences to two vectors after that the technique compares those two vectors with each other to get the similarity score [3]. There are many machine learning techniques like classic machine learning like classifiers, deep learning with neural network and finally the pre-trained models [4]. The second section will then give a literature review as well as background information on prior research in the same field. The third section will talk about the methodology and the system architecture of the project. The fourth section will discuss the data, tools and environment that will be used in the paper, also will talk about the setup of the experiments, the experiments themselves and the results of those experiments. The fifth section will be a discussion of the acquired results.

II. RELATED WORK

Mainly these previous works are divided into two categories which are unsupervised and supervised and those previous works worked on MSRP dataset. The unsupervised usually targets the problem of paraphrase detection by using different similarity techniques. The other approach of targeting the problem of paraphrase detection always uses machine learning and deep learning techniques to solve this problem and the supervised has labeled data to use. The two usual evaluation scores used by these two approaches are the accuracy score and the F1-Score. Accuracy score is how many correctly classified instances of the data while the F1-Score is a measure of the test's accuracy [5].

A. Unsupervised Methods

Some researchers used unsupervised methods to tackle the paraphrase detection problem by using LSA, WORDNET similarity like (lesk, Lin, jcn and other) and some semantic similarity techniques like Resnik, J & C and L & C and also uses cosine similarity and tf-idf weighting [6]. Hassan [7] used the encyclopedic method to get the similarity of the words by using Wikipedia and used also the LSA (latent semantic space) method, the SSA (salient semantic space) and the ESA (explicit semantic space). Rus et al [8] used a method that maps the two sentences into a graph in

978-1-6654-6677-6/22/$31.00 ©2022 IEEE

three stages. First preprocess (tokenization, lemmatization, parsing) the two sentences, second the actual mapping from the text to the graph is called dependency graph creation and it uses the information from the parsing in the preprocessing stage and the last stage is final graph generation which uses the dependency graph and does on it some refinements to get the final graph. After mapping the text into a graph they use the Graph Subsumption method to get the decision if whether these two sentences are paraphrased or not. Islam and Inkpen [9] introduced a new model by using string and semantic similarity techniques together they named this model Semantic Text Similarity (STS) and they used string similarity techniques like the longest common subsequence (LCS) and normalized longest common subsequence (NLCS). In semantic similarity, the semantic similarity matrix between words. Milajevs et al [10] their model convert the two sentences into two vectors and then compare these two vectors with each other by the cosine similarity method and if the similarity value exceeds a certain threshold. These two sentences are considered paraphrased. They tried different thresholds with and without lemmatization. Fernando and Stevenson [11] used the different WORDNET similarity methods like ("JCN", "LCH", "LESK", "LIN", "RES" and "WUP") in the matrix similarity approach.

In table I, a list of unsupervised previous works was summarized to show different paraphrase detection methods on the MSRP dataset. The table is divided into four columns: the first column lists the previous work's reference number, the second column lists the paraphrase detection method, and the third and fourth columns list the accuracy and F1-Score, respectively.

Table I: Different models of unsupervised methods on MSRP dataset

Ref. No.	Methods	Accuracy	F1-Score
[6]	Vector based similarity (Baseline) – MCS	65.4% - 70.3%	75.3% - 81.3%
[7]	ESA – LSA – SSA	67.0% - 68.8% - 72.5%	79.3% - 79.9% - 81.4%
[8]	RMLMG	70.6%	80.5%
[9]	STS	72.6%	81.3%
[10]	Vector-based similarity	73.0%	82.0%
[11]	**matrixJcn**	**74.1%**	**82.4%**

B. Supervised Methods

Kozareva and Montoyo [12] the researchers used a combination of string similarity techniques as they used the "n-grams", "skip-gram" and "longest common subsequence (LCS)" and semantic similarity techniques like WORDNET similarity methods to extract features from the two sentences and then fed these features to machine learning techniques as they used three machine learning modules that are "SVM (support vector machine)", "K-NN (K Nearest Neighbors)" and Maximum Entropy (MaxEnt). Qiu et al [13] introduced a new model by combining the sentence similarity and dissimilarity. Their model has three phases and the first phase is preprocessing the sentences by using Charniak parser and assert and get the predicates of the sentences and the second phase they get the similarity between the predicates by using Thesaurus and the final phase is to input these predicates into dissimilarity classifier to judge if the two sentences are paraphrased or not. Zia and Wasif [14] their model has three stages. The first stage is the preprocessing by the "part of speech (POS)" tagging and followed by removing the stop words. The second stage is feature extraction by the longest common subsequence (LCS) and WordNet semantic heuristics, and the last stage is inputting the features to the weka tool that has many classifiers and specifically they used the logistic regression model. Blacoe and Lapata [15] their model represents the sentences in three types of vectors. The first type represents the pair of input sentences via concatenation or subtraction, and the second type represents a vector of encoding the words of the sentence, and the last type represents a vector of four informational items that are the cosine similarity of the sentence vectors, the length of the first sentence, the length of the second sentence and lastly the unigram overlap of the two sentences. After that, these three vectors are fed with different combinations to four composition models that are the "distributional memory (DM)", "the neural language model (NLM)", the recursive auto encoder (RAE) and the simple distributional semantic space (SDS) that gives the best results. Ji and Eisenstein [16] they introduced a new weighting method that is called TF-KLD and used a combination of feature set that has the unigram and bigram similarity methods and other methods that they called the fine-grained features. They used the weighting method with the fine-grained features and input them into the support vector machine. El Desouki et al [17] they approached the problem of paraphrase detection in three steps. In the first step, they inputted the sentences into two types of text similarity methods and the first is the string similarity algorithms and the second is the semantic similarity algorithm which is called skip-thought. Through these methods, they get similarity values and then in the second step they input the similarity values to the weka tool that has a lot of classifiers and the last step they used the select attribute method from the weka to select the best combination of the text similarity algorithms and it resulted in 7 algorithms that they called the CombineBest. They inputted the CombineBest into weka again and got the best results with the VotedPerceptron. Finch et al [18] there model has three steps. First, they tokenized the sentences and after that, they used stemming technique to return the word to its original root. Finally, in the last step, they used the output from the machine transition evaluation techniques that are WER, BLEU, PER and NIST. The output of those techniques is used to feed the support vector machine classifier to judge if the sentence is paraphrased or not. Wan et al [19] they used many features to feed different classifiers from the weka tool. The features are different "N-gram" techniques, "Dependency relation techniques", "Dependency Tree edit distance techniques" and surface techniques. To give a total of 17 used techniques to feed 5 classifiers of the weka that are the NavieBayes, a clone of the C4.5 decision tree classifier that is called J48, a support vector machine with a polynomial kernel that is called SMO, K-Nearest Neighbor that is called IBK and lastly the baseline technique that is called ZeroR but

they only reported the result from the support vector machine as it outperformed other classifiers.

In table II, a list of supervised previous works summarized to show different paraphrase detection methods on the MSRP dataset. The table is divided into four columns: the first column lists the previous work's reference number, the second column lists the paraphrase detection method, and the third and fourth columns list the accuracy and F1-Score, respectively.

Table II: Different models of supervised methods on MSRP dataset

Ref. No.	Methods	Accuracy	F1-Score
[12]	KM	76.6%	79.6%
[13]	QKC	72.0%	81.6%
[14]	ParaDetect	74.7%	81.8%
[15]	SDS	73.0%	82.3%
[16]	**TF-KLD**	**80.4%**	**85.9%**
[17]	CombineBest	76.6%	83.5%
[18]	FHS	75.0%	82.7%
[19]	WDDP	75.6%	83.0%

III. METHODOLOGY

Fig 1 shows the process of our proposed model and it has 3 processes the preprocessing of the dataset and after that, the preprocessed data is inputted to the three text similarity techniques to get the similarity scores and lastly these similarity scores will be inputted into machine learning to get the result of the data that the data is paraphrased or not and the next subsections will explain the overview in details.

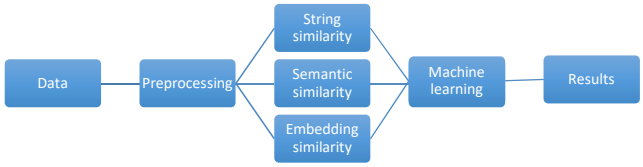

Fig 1: System Overview

A. *Preprocessing*

First is the step of preprocessing where we remove special characters, change the sentence to lowercase, use stemming, use lemmatization and remove stop words. Where the stemming and the lemmatization are to convert the words of the sentence to their root form.

B. *Text similarity techniques*

The second step will input the preprocessed sentences into three similarity techniques that are:

First, the string similarity techniques that take the two sentences and compare the difference in the characters and words in the two sentences and the abydos library in python has over 200+ string similarity techniques like Levenshtein similarity and Damerau-Levenshtein-similarity and other more string similarity techniques.

Second, the semantic similarity techniques that take the two sentences and compare the difference in the meaning of the two sentences and the nltk and spacy libraries in python have different semantic similarity techniques like wup and lin and other more semantic similarity techniques.

Third, the embedding similarity techniques that take the two sentences and convert the two sentences to their vectors and then compare these vectors with each other to get the similarity score and the sentence transformers library in python has many pre-trained models that make the embedding similarity techniques like bert-base-nli-mean-tokens and all-mpnet-base-v2 and other more pre-trained models. These similarity scores that are produced are between 1 and 0. 1 being 100% percent similar to each other and 0 being has no similarity with each other.

C. *Hybrid model*

The third step is to take these similarity scores with different combinations and all of them to be fed to different machine learning classifiers from the skleran library in python to decide whether the two sentences are paraphrased or not and these classifiers are like Logistic Regression and Extra Tree Classifier and other classifiers.

IV. EXPERIMENT AND RESULTS

The tests were carried out on an i7 9th Gen processor with 16GB of RAM and an NVIDIA GTX 1660ti 6GB GPU. All the experiments were done on the same dataset with a 10-fold K-fold validation method.

A. Data-set

The proposed model will use "the Microsoft research paraphrase corpus" (MSRP) dataset. This dataset contains 5801 pairs of sentences and these sentences are extracted from news sources on the web and from each given news article, only one sentence has been taken. Each pair of sentences is combined with human annotations to indicate whether the pair of sentences are semantically equivalent or not and 3900 pairs (67%) of the original 5801 pairs were annotated as paraphrased pairs. In table III, there are some examples of the dataset.

Table III: Dataset examples

#1 String	#2 String	Quality
Amrozi accused his brother, whom he called "the witness", of deliberately distorting his evidence.	Referring to him as only "the witness", Amrozi accused his brother of deliberately distorting his evidence.	1
Around 0335 GMT, Tab shares were up 19 cents, or 4.4%, at A$4.56, having earlier set a record high of A$4.57.	Tab shares jumped 20 cents, or 4.6%, to set a record closing high at A$4.57.	0
The Nasdaq had a weekly gain of 17.27, or 1.2 percent, closing at 1,520.15 on Friday.	The tech-laced Nasdaq Composite .IXIC rallied 30.46 points, or 2.04 percent, to 1,520.15.	0
Revenue in the first quarter of the year dropped 15 percent from the same period a year earlier.	With the scandal hanging over Stewart's company, revenue the first quarter of the year dropped 15 percent from the same period a year earlier.	1

B. Experiments

In table IV, the different experiments done on the MSRP dataset with lowercase, remove stop words and lemmatization as preprocessing stage and the text similarity stage consist of three different types of text similarities which consist of different combinations of 168 string similarity techniques, 8 bert models for embedding similarity techniques and lastly 3 wordnet algorithms and 1 spacy algorithm for semantic similarity techniques and those similarity scores are inputted to 3 classifiers that are Linear SVC, Logistic RegressionCV and Ridge ClassifierCV. From table IV, the Linear SVC classifier scored the best accuracy score with 75.78% and the best F1-Score with 83.01%.

Table IV: Different experiments of the proposed model

Name	Accuracy	F1-Score	Classifier
168 string + 8 bert + wordnet	**75.78%**	**83.01%**	**Linear SVC**
168 string + 8 bert + spacy + wordnet	75.75%	82.88%	Logistic RegressionCV
168 string + 8 bert + spacy	75.64%	82.97%	Ridge ClassifierCV
168 string + 8 bert	75.59%	82.76%	Logistic RegressionCV
168 string + spacy	74.71%	82.19%	Logistic RegressionCV
168 string + spacy + wordnet	74.66%	82.19%	Logistic RegressionCV
168 string + wordnet	74.38%	81.99%	Logistic RegressionCV
168 string	74.14%	81.96%	Linear SVC

V. DISCUSSION

The combination of the 168 string similarity techniques from the abydos library in python and 8 bert models from the sentence transformers library as the embedding similarity techniques and 3 of wordnet as the semantic similarity techniques. This combination is better than any other combination as it got 75.78%. It is clear from these trials that the more features and similarity algorithms added, the better the outcomes will be. It is concluded from table V that the combination of the different similarity techniques is much better than single text similarity techniques alone and the selected algorithms in table V are the top 20 algorithms with the Gradient Boosting classifier as it is the best classifier for single techniques. It is concluded from table VI that the approach of combining the similarity techniques with the machine learning techniques is giving better results than using the unsupervised approach that

concentrates on using the text similarity techniques alone. And from table IV and table V that the improvement range is 1.26% between the worst combination and the best single text similarity technique and 2.9% between the best combination and the best single text similarity technique and 1.64% between the best combination and the worst combination.

Table V: Single text similarity algorithms bench-marked with the MSRP dataset

Methods	Accuracy	F1-Score
all-MiniLM-L12-v2	**72.88%**	**81.36%**
all-MiniLM-L6-v2	72.83%	81.27%
all-mpnet-base-v2	72.59%	81.30%
Dennis	72.36%	81.05%
Kuhns VII	72.36%	81.05%
Kuhns IX	72.34%	81.22%
ms contingency	72.34%	81.22%
Pearsons Chi-Squared	72.34%	81.22%
Pearson Phi	72.34%	81.22%
Goodman & Kruskals Tau A	72.33%	81.27%
BaroniUrbaniBuserI	72.26%	80.65%
Doolittle	72.24%	81.17%
Pearson II	72.24%	81.17%
Unknown B	72.24%	81.17%
Harris & Lahey	72.19%	81.14%
GilbertWells	72.17%	80.89%
Kuder & Richardson	72.15%	81.13%
Maxwell & Pilliner	72.15%	81.13%
all-distilroberta-v1	72.14%	80.90%
Kuhns XI	72.09%	80.88%

Table VI: Comparing the proposed model with previous models

Ref. No.	Methods	Accuracy	F1-Score
Proposed Model	**Linear SVC**	**75.78%**	**83.01 %**
[6]	Vector based similarity (Baseline) – MCS	65.4% - 70.3%	75.3% - 81.3%
[7]	ESA – LSA – SSA	67.0% - 68.8% - 72.5%	79.3% - 79.9% - 81.4%
[8]	RMLMG	70.6%	80.5%
[9]	STS	72.6%	81.3%
[10]	Vector-based similarity	73.0%	82.0%
[11]	matrixJcn	74.1%	82.4%

VI. CONCLUSION AND FUTURE WORK

Paraphrase detection is one of the most basic and common tasks in the field of Natural language processing. Yet, it is very important as it is involved in many more complex natural language processing tasks like text mining, plagiarism detection, data retrieval and it is used also in academic writing as a lot of people copy from other people's work and research and this is considered as a big crime. Also, it is concluded from the experiments done in this paper that the approach of combining many different similarity techniques is giving much better results than the use of only one type of similarity techniques. it is concluded that the combination of string similarity, semantic similarity and embedding similarity with the help of machine learning classifiers are better than the use of only a single type of similarity technique alone with help of machine learning classifiers and also better than the use of unsupervised methods that use the text similarity techniques with a threshold only and also the proposed model is better and simpler than some previous supervised methods as they used some complex and deep machine learning neural networks.

Our future work is to try different preprocessing techniques like stemming, removing special characters, removing numbers and without any preprocessing and also different combinations of similarity techniques and to add more similarity techniques and to use other different machine learning techniques like neural networks and LSTM

REFERENCES

[1] Meo, S. A., & Talha, M. (2019). Turnitin: Is it a text matching or plagiarism detection tool?. Saudi journal of anaesthesia, 13(Suppl 1), S48.
[2] Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. international journal of Computer Applications, 68(13), 13-18.
[3] Shahmohammadi, H., Dezfoulian, M., & Mansoorizadeh, M. (2021). Paraphrase detection using LSTM networks and handcrafted features. Multimedia Tools and Applications, 80(4), 6479-6492.
[4] Sah, S. (2020). Machine learning: a review of learning types.
[5] El Desouki, M. I., & Gomaa, W. H. (2019). Exploring the recent trends

of paraphrase detection. International Journal of Computer Applications, 975(S 8887).
[6] Mihalcea, R., Corley, C., & Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In Aaai (Vol. 6, No. 2006, pp. 775-780).
[7] Hassan, S. (2011). Measuring semantic relatedness using salient encyclopedic concepts. University of North Texas.
[8] Rus, V., McCarthy, P. M., Lintean, M. C., McNamara, D. S., & Graesser, A. C. (2008, May). Paraphrase Identification with Lexico-Syntactic Graph Subsumption. In FLAIRS conference (pp. 201-206).
[9] Islam, A., & Inkpen, D. (2009). Semantic similarity of short texts. Recent Advances in Natural Language Processing V, 309, 227-236.
[10] Milajevs, D., Kartsaklis, D., Sadrzadeh, M., & Purver, M. (2014). Evaluating neural word representations in tensor-based compositional settings. arXiv preprint arXiv:1408.6179.
[11] Fernando, S., & Stevenson, M. (2008, March). A semantic similarity approach to paraphrase detection. In Proceedings of the 11th annual research colloquium of the UK special interest group for computational linguistics (pp. 45-52).
[12] Kozareva, Z., & Montoyo, A. (2006, August). Paraphrase identification on the basis of supervised machine learning techniques. In International conference on natural language processing (in Finland) (pp. 524-533). Springer, Berlin, Heidelberg.
[13] Qiu, L., Kan, M. Y., & Chua, T. S. (2006, July). Paraphrase recognition via dissimilarity significance classification. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (pp. 18-26).
[14] Ul-Qayyum, Z., & Altaf, W. (2012). Paraphrase identification using semantic heuristic features. Research Journal of Applied Sciences, Engineering and Technology, 4(22), 4894-4904.

[15] Blacoe, W., & Lapata, M. (2012, July). A comparison of vector-based representations for semantic composition. In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning (pp. 546-556).

[16] Ji, Y., & Eisenstein, J. (2013, October). Discriminative improvements to distributional sentence similarity. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 891-896).
[17] El Desouki, M. I., Gomaa, W. H., & Abdalhakim, H. (2019). A hybrid model for paraphrase detection combines pros of text similarity with deep learning. Int. J. Comput. Appl, 975, 8887.

[18] Finch, A., Hwang, Y. S., & Sumita, E. (2005). Using machine translation evaluation techniques to determine sentence-level semantic equivalence. In Proceedings of the third international workshop on paraphrasing (IWP2005).

[19] Wan, S., Dras, M., Dale, R., & Paris, C. (2006, November). Using dependency-based features to take the'para-farce'out of paraphrase. In Proceedings of the Australasian language technology workshop 2006 (pp. 131-138).