A Study on Paraphrase Detection Techniques in NLP

Abstract. Paraphrase detection is related to the domain of Natural Language Processing. The system determines the similarities between multiple sentences semantically provided by the user. If both sentences have the same meaning, then the penalties will denoted as paraphrases with respect to each other, otherwise, they will not be paraphrases to each other. As paraphrasing removes the communication barrier and helps people to express their feelings in an easier way, it plays a very important role in different domains, such as healthcare, teaching, plagiarism detection, research, and many other fields. But still, there are many limitations of paraphrasing, one of the most important challenges of paraphrasing is usually when the writer replaces just one or two words in the source phrases with synonyms. This type of paraphrasing does not show sufficient understanding and engagement with the text. Instead, the writer must try to take ideas and information and put them into his own words. In order to overcome these challenges, the paraphrasing model should be more developed, so there is no over-fitting or under-fitting situation and the dataset should be more enriched with various regional usage of sentences. This article deals with different detection paraphrasing procedures with the main goal of neural methods.

Keywords: Paraphrase · Natural Language Processing · Semantic Similarity Technique · Encoder-Decoder Architecture · Reinforcement Learning

1 Introduction

Natural Language Processing (NLP) is a field that enables computers to understand and analyze human language. It is used in various domains, such as text analysis, chatbots, plagiarism checkers, etc. Numerous NLP-related topics and project concepts exist that can assist people in automating their regular activities. Paraphrase detection is one of these subjects. Plagiarism detection is one of the fundamental NLP operations that is involved in many other complex NLP tasks like information retrieval, text mining, etc. The technique of rewriting a text while keeping the original meaning intact is known as paraphrasing. In many different applications, such as plagiarism detection, authorship attribution, question answering, text summarising, general text mining, etc., automatic paraphrase identification is crucial [23, 17]. It is also crucial to measure how similar texts are to one another on a more general level. There are certain issues with paraphrase detection, although some of the existing paraphrase systems functioned fairly well. Turnitin, a tool for teachers and academics that uses paraphrase detection, is another example of a system from daily life [2]. The most crucial stage in detecting plagiarism is to identify paraphrases, which is where the tremendous utility of paraphrase detection comes in. The model for paraphrase detection will combine machine learning, deep learning, and conventional natural language processing methods. In this survey paper, we are motivated to understand these approaches and datasets related to paraphrase detection. Initially, we have focused on the techniques that assist in identifying the paraphrases within sentences. Thereafter, discussed multiple sentence-based paraphrase detection using machine learning techniques [3]. In order to determine how similar two sentences are, three basic NLP approaches are fundamentally used:

First, string similarity approaches are employed to determine, how similar a sentence's characters are or the entirety of a sentence [4]?

Second, semantic similarity techniques are used to compare two sentences, this comparison typically refers to a dictionary or a resource like Wikipedia or WordNet [4].

Third, similarity embedding algorithms often assess a similarity score by splitting the two phrases into two vectors, comparing the two vectors, and calculating the similarity score [4]. Additionally, employed various machine learning techniques such as classifiers, deep learning with neural networks, and pretrained models to build these embedding algorithms [3].

The objective of this paper is to provide an overview of paraphrase detection with the main goal of neural methods. This article will address a literature study as well as historical background on earlier work on the same topic, methods, and the project's system design. The overall structure of this paper is as follows. The background study done in paraphrase detection is presented in Section 2. Dataset-related discussion has been conducted in Section 3. Sections 4 and 5 describe the employed methodologies and evaluation techniques in detail. Finally, the discussion and conclusions of the study are described in Section 6 and 7.

2 Background Study

These earlier efforts, which mostly fall into the supervised and unsupervised categories, worked using the MSRP dataset. The challenge of paraphrase detection utilizing various similarity algorithms is typically the attention of unsupervised individuals. The second method, which is concerned with the issue of paraphrase detection, always uses machine learning and deep learning approaches, and it makes use of supervised data that has been labeled. The precision score and the F1 score are two often utilized evaluation scores by these two methods [5].

A. Unsupervised Methods

Latent Semantic Analysis (LSA), WORDNET similarity (like WUP, LIN, and JCN), various semantic similarity techniques like Resnik, J&C, and L&C, as well as cosine similarity and Term Frequency–Inverse Document Frequency (TF-IDF) weighting, have been utilized by some researchers to solve the problem of paraphrase detection [6]. By breaking two phrases into two vectors, Milajevs et al. developed a model to evaluate the two vectors using the cosine similarity approach to determine whether the similarity value is greater than a predetermined threshold [7]. This pair of sentences is a paraphrase. Different thresholds, both with and without lemmatization, were tested. In the matrix similarity technique, Fernando and Stevenson used a variety of WORDNET similarity algorithms, including "JCN," "LCH," "LESK," "LIN," "RES," and "WUP [6]."

Ref.No.	Methods	Accuracy	F1-Score
[3]	Maximum Common Substructure (MCS)	65.4%	75.3%
[5]	Explicit Semantic Analysis (ESA)	67.0%	79.3%
[5]	Latent Semantic Analysis (LSA)	68.8%	79.9%
[5]	Similarity Structure Analysis (SSA)	72.5%	81.4%
[7]	RMLMG	70.6%	80.5%
[7]	Semantic Textual Similarity (STS)	72.6%	81.3%
[7]	Vector-based Similarity	73%	82%
[8]	Matrix JCN	74.1%	82.4%

Table I: Different models of unsupervised methods on MSRP data-set

B. Supervised Methods

In order to extract features from these two sentences, Kozareva and Montoyo [9] combined string similarity techniques like "n-grams," "skip-gram," and "Longest Common Subsequence (LCS)" with semantic similarity techniques like "WORDNET similarity methods." A brand-new model of sentence similarity and dissimilarity combinations was presented by Qiu et al. They use the Charniak parser to preprocess the sentences in the first stage, asserting and obtaining the predicates of the sentences. In the second stage, they use a thesaurus to determine how similar the predicates are. The third and final stage uses the predicates to determine whether the two sentences are paraphrased or not. [10] Wan et al. [9] filled numerous classifiers from the Weka tool using a variety of features. Different "N-gram" approaches, "Dependency Techniques", "Dependency Tree Editing Distance Techniques," and Surface Techniques are the characteristics.

Ref.No.	Methods	Accuracy	F1-Score
[7]	KM	76.6%	79.6%
[7]	QKC	72.0%	81.6%
[8]	ParaDetect	74.7%	81.8%
[8]	SDS	73.0%	82.3%
[9]	TF-KLD	80.4%	85.9%
[9]	CombineBest	76.6%	83.5%
[9]	FHS	75.0%	82.7%
[9]	WDDP	75.6%	83.0%

Table II: Different models of supervised methods on MSRP data-set

3 Employed Datasets

In this section, we have discussed datasets that have been used to design various types of paraphrase detection models.

MRPC (Microsoft Research Paraphrase Corpus) This dataset contains 5,801 sentence pairs which have been collected from news-wire items. Thereafter, a group of annotators helps to annotate each pair as a paraphrase or not. The annotated dataset is divided into two subsets training subset (4,076 sentence pairs, of which 2,753 are paraphrases) and test subset (1,725 pairs, of which 1,147 are paraphrases) [10].

PPDB More than 220 million paraphrase pairs are available in the database. In addition, it has 8 million lexical words and 73 million phrasal phrases. Additionally, it has 140 million different sentence patterns. Meaning-preserving and syntactically sound transformations are the inevitable outcomes of these patterns [11].

MSCOCO (Microsoft Common Objects in Context) It is a sizable data set for object detection. More than 120,000 photos from Human Annotated Captions are available, each with five captions contributed by five distinct annotators. There are over 500,000 paraphrase pairs in this data collection [12].

WikiQA (Wikipedia open-domain Question Answering) A collection of question & sentence pairings that have been compiled and annotated for study on open-domain question answering is known as the WikiQA corpus. Bing query logs were employed as the question source to accurately reflect common users' real information needs. Each query contains a link to a possible solution on a Wikipedia page. Sentences from the summary section of a Wikipedia page were utilized as potential answers because they typically contain the most significant and fundamental information about the subject. The corpus consists of 29,258 sentences and 3,047 questions, with 1,473 sentences designated as answers to the relevant questions [13].

TURL (Twitter News URL Corpus) TURL was developed by compiling a large number of Twitter paraphrases. Utilizing shared URLs, it connects tweets. This data collection often consists of two subsets, each of which includes both paraphrases and non-paraphrases. One subset is annotated by human annotators, while the other is automatically annotated [13].

Quora The questions that are posted on the Quora Question Answering website make up the Quora data set. It is the unique data set that simultaneously offers responses at both the sentence and word levels. Additionally, the data set's queries are natural, making them far more realistic for question-answering algorithms. There are duplicate question pairs in more than 400k rows. Annotated 150k pairs are utilized as paraphrases. In both training and testing, these reliable question pairings are employed to build the paraphrase detection model [14].

4 Methodology

In this part, we have discussed various different paraphrase-detection techniques in detail.

Rule-based approach Rule-based paraphrase detection deals with automatically collected paraphrase rules, which are generally created by hand. The limitation of extraction methods results in the generation of long and complex paraphrase patterns, which has a huge impact on performance [15].

Thesaurus-Based Approach In thesaurus-based techniques, every possible synonym for the phrases to be replaced is first extracted from the thesaurus. Generally, by substituting certain words from the source sentences with their thesaurus-extracted synonyms, this method typically produces paraphrases [14]. We have observed that this approach has not provided accurate results due to the lack of varied datasets.

SMT-Based Approach This approach is primarily concerned with statistical machine translation (SMT). This machine translation technique is presented as a paraphrase detection system. Hence, we can employ the SMT approach to create any model of machine translation as well as paraphrase detection. A machine translation model uses the statistical translation model p(f|e) and the language model p(e) to determine the optimal translation of a text from one language into another using the following equation [15].

$$\hat{e} = arg \max_{e \in e^*} p(f|e)p(e)$$

This model will provide the best paraphrase of a text on the source side (s) to a text on the target side (t) by employing this concept to identify paraphrases, which are presented by the following equation.

$$\hat{t} = arg \max_{t \in t^*} p(s|t)p(t)$$

Neural Network Approach In the past, machine translation approaches such as template-based and/or statistical were the main emphasis of paraphrasing. Template matching a statistical translation model modeling are very challenging tasks. Many researchers have begun the widespread use of neural network models for paraphrase detection [20].

Encoder-Decoder Architecture The majority of the currently used models for generating paraphrases are typically sequence-by-sequence models with an encoder and a decoder. The source texts are encoded by the encoder into a context vector representation, which includes a list of vector representations for each word's semantics. The decoder then finds the paraphrases using the vectors provided by the encoder [1].

The fundamental goal of the encoding side is to collect the semantic information that the decoder requires to detect paraphrases. Researchers also have a wide range of options for the encoder after building several neural network models such as the seq2seq, recurrent neural networks (RNN), and long-short-term memory (LSTM) networks [16]. Additionally, on numerous text detection tasks, the transformer model displayed cutting-edge performance [18]. Transformer provides us with lengthy sentence dependencies thanks to its additional ability.

On the other hand, a contextualized representation of paraphrases is used in each decoding step on the decoding side with a vector representation of the previously generated words. The highest probability word is constructed when a vocabulary distribution has been obtained which is known as greedy decoding [19]. Additionally, a more commonly used method called ray tracing is also employed, which identifies the k-best paths up to the current time step during decoding [18].

We have noted that the Encoder-Decoder architecture for paraphrase detection can be categorically separated into two categories such as model-focused and attribute-focused, respectively for better visualization. The model-focused architecture refers to utilizing various strategies to enhance the encoder or decoder. On the contrary, attribute-focused architecture enables the decoder to concentrate on a small number of highly relevant phrases or words using attention mechanism [19].

Additionally, the goal of attribute-focused upgrades is to enable control over certain of the created attribute's paraphrases, such as the amount of syntax and granularity, and to enhance the quality of the generated paraphrases in some specific areas. Some systems control diversity by sending the decoder control signals.

Variational Auto Encoder (VAE) In order to manage the high-dimensional inputs, VAE can learn nonlinear representations. Revision of the original latent code input form is used to facilitate learning [21].

Reinforcement Learning Ramtin Mehdizadeh et al, employed a reinforcement learning approach in the form of exposure bias is a well-known issue with encoder and decoder architecture in their research. During training the current word's decoding is conditioned by gold references, but during testing, it is conditioned by generated output from the previous time step [21].

Generative Adversarial Networks (GAN) The GANs were created by Goodfellow et al. (2014), and they are made up of generators and discriminators. The generators attempt to produce outputs that are realistic and correlate to reality distributions, while the discriminators attempt to tell the difference between samples produced by generators and samples that are actual. Minimax optimization is first introduced to the GAN in (Goodfellow et al., 2014) [22]. However, because creating discrete words is non-differentiable, the usual training procedure cannot be used when GAN is utilized in text generation. The generators are then fixed, and the discriminators are retrained to minimize loss by combining real samples with samples generated by marker generators [22].

Finally, we have observed that the researchers have employed various linguistic expertise, synonym mappings, rule-based approaches, and neural network models to generate word or phrase-level paraphrases [22, 24].

5 Evaluation techniques

In order to determine how good a model is in detecting paraphrases, it must perform well. Some typical evaluation methods for paraphrasing detection are listed below:

Confusion Matrix: A confusion matrix offers a thorough description of true positive, true negative, false positive, and false negative predictions. It's also helpful to know what classes (paraphrase or non-paraphrase) your model may be having trouble with and where it might be making mistakes. Then the researchers employed the standard metrics such as accuracy, precision, recall, and F-measure score to evaluate the effectiveness of the paraphrase detection model. After that, the precision and recall give information on the model's capacity to accurately detect the correct paraphrases or not, while accuracy provides an overall measure of correct predictions.

ROC Curve and AUC: Receiver Operating Characteristic (ROC) curves show how true positive rate and false positive rate are traded off at various levels. The overall effectiveness of the model for detecting paraphrases at various threshold settings is summarised by the Area Under the Curve (AUC) value. The precision-recall curve also shows the trade-off between recall and precision at various classification levels. Dealing with unbalanced datasets can make it more informative.

Mean Average Precision (mAP): This metric is useful when dealing with several positive examples in the dataset and is frequently utilized for ranking-based tasks. It computes the mean across all queries after computing the average precision for each query.

Kappa Score: In order to account for chance agreement, Cohen's Kappa is a statistic that measures the degree of agreement between model predictions and human annotators' assessments. It takes into account the prospect of random agreement.

Spearman's Rank Correlation Coefficient: This metric evaluates the consistency between the paraphrase pair rankings determined by the algorithms and the rankings determined by humans. It can be helpful when the relative order is significant but the absolute similarity score between pairs of phrases is not.

Human Evaluation: Human judgment may ultimately be the best indicator of the effectiveness of a paraphrase detection system. This is important because it provides a precise and quantitative analysis of the measures. On the other hand, the major objective of automatic evaluation is n-gram overlays [13].

We noticed that in order to evaluate the paraphrase detection models, it is crucial to combine these techniques in order to understand both their effectiveness and possible drawbacks. Additionally, take note of the dataset that was used for evaluation because the quality and diversity of the data can have a big impact on how well the model performs.

6 Discussion and Observation

The task of determining whether two sentences or phrases express the same meaning, even if they are formulated differently, is known as paraphrase detection. Generally, we have observed that natural language processing (NLP) classification is difficult for detecting similar phrases. Hence, the researchers have developed paraphrase detection systems to determine the semantic similarity between two sentences given a pair of them. It is noted as a paraphrase if the meaning of both phrases is the same; otherwise, it is marked as a non-paraphrase. On plain text corpora like the Microsoft Paraphrase Corpus (MSRP) most existing paraphrase algorithms have done pretty well. However, due to problems like typos, acronyms, style, and structure, paraphrase identification in noisy user-generated tweets makes it difficult to detect the correct paraphrases. Furthermore, the absence of shared lexical traits makes it exceedingly challenging to compare the semantic similarity of two brief sentences. Although there hasn't been much research on the topic, some preliminary findings using the SemEval 2015 Twitter benchmark dataset have been published. Unfortunately, when compared to different data sets, the best-performing techniques do not appear to perform as well.

Almost all recent research involving the use of language models that have already been trained to create paraphrases is rather basic. As a result, we could combine substantial pre-trained language models with different learning processes including reinforcement learning, VAE, and GAN.

Additionally, we have noted that only word-level and phrase-level paraphrasing is the focus of recent work on multi-level paraphrase production. Multiple granularity levels can be included, though. We think it is beneficial to study at several levels, such as the word level, phrase level, syntactic level, and sentence level. These levels assist in designing text summarising and text simplification systems along with detecting paraphrases of given sentences while maintaining meaning. Therefore, performance could be enhanced by using transfer learning from these three tasks. Producing stylistic phrases at this time, it is not possible to precisely manage word and phrase substitution when creating paraphrases. As a result, it is challenging to manage the generated paraphrases' style.

In terms of evaluated metrics of the paraphrase detection systems, we have observed that the automatically generated assessment metrics are inadequate for evaluating paraphrase generation. Therefore, there is a need for improved automated evaluation techniques. To explicitly determine if the output sentence and the input sentence are paraphrases, one technique is to employ paraphrase identification in automatic assessment metrics [24].

Finally, we can conclude that an efficient paraphrase-detection system is useful for resolving various natural language processing (NLP) activities, including information retrieval, text summarization, question-answering systems, and more.

7 Conclusion

One of the most fundamental and frequent tasks in natural language processing is the recognition of phrases. Nevertheless, it is crucial because it is used in many more challenging NLP tasks, including text mining, plagiarism detection, and data mining. It is also utilized in academic writing because many people plagiarise from the work and research of others, which is a serious offense. The trials done for this research further support the conclusion that employing multiple similarity strategies in combination yields far better results than using a single sort of similarity methodology. It is also superior to using unsupervised methods that use text similarity techniques only with the threshold.

References

1. Chen, W., Tian, J., Xiao, L., He, H., Jin, Y.: A semantically consistent and syntactically variational encoder-decoder framework for paraphrase generation. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 1186–1198 (2020)
2. El Desouki, M.I., Gomaa, W.H., Abdalhakim, H.: A hybrid model for paraphrase detection combines pros of text similarity with deep learning. Int. J. Comput. Appl **975**, 8887 (2019)
3. Fernando, S., Stevenson, M.: 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 (2008)
4. Finch, A., Hwang, Y.S., Sumita, E.: Using machine translation evaluation techniques to determine sentence-level semantic equivalence. In: Proceedings of the third international workshop on paraphrasing (IWP2005) (2005)
5. Gomaa, W.H., Fahmy, A.A., et al.: A survey of text similarity approaches. International Journal of Computer Applications **68**(13), 13–18 (2013)
6. Hany, M., Gomaa, W.H.: A hybrid approach to paraphrase detection based on text similarities and machine learning classifiers. In: 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC). pp. 343–348. IEEE (2022)
7. Hassan, S.: Measuring semantic relatedness using salient encyclopedic concepts. University of North Texas (2011)
8. Hassan, S., Mihalcea, R.: Semantic relatedness using salient semantic analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 25, pp. 884–889 (2011)
9. Ji, Y., Eisenstein, J.: Discriminative improvements to distributional sentence similarity. In: Proceedings of the 2013 conference on empirical methods in natural language processing. pp. 891–896 (2013)
10. Kozareva, Z., Montoyo, A.: Paraphrase identification on the basis of supervised machine learning techniques. In: Advances in Natural Language Processing: 5th International Conference on NLP, FinTAL 2006 Turku, Finland, August 23-25, 2006 Proceedings. pp. 524–533. Springer (2006)

11. Liu, M., Yang, E., Xiong, D., Zhang, Y., Meng, Y., Hu, C., Xu, J., Chen, Y.: A learning-exploring method to generate diverse paraphrases with multi-objective deep reinforcement learning. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 2310–2321 (2020)
12. Ma, S., Sun, X., Li, W., Li, S., Li, W., Ren, X.: Query and output: Generating words by querying distributed word representations for paraphrase generation. arXiv preprint arXiv:1803.01465 (2018)
13. Meo, S.A., Talha, M.: Turnitin: Is it a text matching or plagiarism detection tool? Saudi Journal of Anaesthesia **13**(Suppl 1), S48 (2019)
14. Metzler, D., Hovy, E., Zhang, C.: An empirical evaluation of data-driven paraphrase generation techniques. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 546–551 (2011)
15. Mihalcea, R., Corley, C., Strapparava, C., et al.: Corpus-based and knowledge-based measures of text semantic similarity. In: Aaai. vol. 6, pp. 775–780 (2006)
16. Milajevs, D., Kartsaklis, D., Sadrzadeh, M., Purver, M.: Evaluating neural word representations in tensor-based compositional settings. arXiv preprint arXiv:1408.6179 (2014)
17. Mondal, A., Dey, M., Mahata, S.K., Sarkar, D.: An automatic summarization system to understand the impact of covid-19 on education. In: Applications of Machine Intelligence in Engineering, pp. 379–386. CRC Press (2022)
18. Rus, V., McCarthy, P.M., Lintean, M.C., McNamara, D.S., Graesser, A.C.: Paraphrase identification with lexico-syntactic graph subsumption. In: FLAIRS conference. pp. 201–206 (2008)
19. Sah, S.: Machine learning: a review of learning types (2020)
20. Sarkar, S., Saha, S., Bentham, J., Pakray, P., Das, D., Gelbukh, A.F.: Nlp-nitmz@dpil-fire2016: Language independent paraphrases detection. In: FIRE (Working Notes). pp. 256–259 (2016)
21. Seraj, R.M., Siahbani, M., Sarkar, A.: Improving statistical machine translation with a multilingual paraphrase database. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1379–1390 (2015)
22. Shahmohammadi, H., Dezfoulian, M., Mansoorizadeh, M.: Paraphrase detection using lstm networks and handcrafted features. Multimedia Tools and Applications **80**, 6479–6492 (2021)
23. Sinha, S., Mandal, S., Mondal, A.: Question answering system-based chatbot for health care. In: Proceedings of the Global AI Congress 2019. pp. 71–80. Springer (2020)
24. Zhou, J., Bhat, S.: Paraphrase generation: A survey of the state of the art. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 5075–5086 (2021)