Revisiting the challenges and surveys in text similarity matching and detection methods

Alva Hendi Muhammad [a,1,*], Kusrini [a,2], Irwan Oyong [a,3]

[a] Department of Informatics Engineering, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia
[1] alva@amikom.ac.id; [2] kusrini@amikom.ac.id; [3] oyong@amikom.ac.id
* Corresponding Author

Received 25 May 2022; accepted 13 August 2022; published 20 September 2022

ABSTRACT

The massive amount of information from the internet has revolutionized the field of natural language processing. One of the challenges was estimating the similarity between texts. This has been an open research problem although various studies have proposed new methods over the years. This paper surveyed and traced the primary studies in the field of text similarity. The aim was to give a broad overview of existing issues, applications, and methods of text similarity research. This paper identified four issues and several applications of text similarity matching. It classified current studies based on intrinsic, extrinsic, and hybrid approaches. Then, we identified the methods and classified them into lexical-similarity, syntactic-similarity, semantic-similarity, structural-similarity, and hybrid. Furthermore, this study also analyzed and discussed method improvement, current limitations, and open challenges on this topic for future research directions. As the results, this paper highlighted the importance of selecting the appropriate preprocessing algorithms to reduce data dimensionality and also combining several algorithms to enhance the overall matching and detection process.

KEYWORDS
Text similarity
Similarity detection
Document similarity
Text matching
Natural language processing

1. Introduction

Finding text similarity between documents has been the dominant issue for decades. Two texts are similar when both have closeness in terms of character or meaning. Take an example, the phrase "the cat sits under the tree" with "there is a dog sit under the tree". In order to determine whether or not these two phrases are similar, first, we can observe on the surface can consider the similar word. Both phrases have similar word levels with exactly four unique words: "the", "sit", "under", "tree". The type of similarity by considering the surface closeness without considering the word's actual meaning is known as *lexical similarity*. Instead of comparing word by word between sentences, another approach determines similarity by focusing on the context. Let us consider another example: "The elephant eats mostly grass" and "Grass is the main food for elephant". By just looking at the words, we understand that both sentences have different forms on the surface. However, considering the context, the phrases share similar meanings and information. The similarity on the basis of meaning or context rather than character is called *semantic similarity*.

The evidence has shown the earlier similarity detection has been utilized to investigate plagiarism code in software by identifying the pattern of structure and syntax. Since the internet era in the 1990s, there has been a growing trend of text similarity detection for determining digital documents similarity. Previous studies have shown that similarity detection in a programming language has a slightly different method with digital documents, which is based on natural language. Similarity detection in programming language mainly focused on a developed metric based on several features, such as variables, expressions, statements, number of lines, and subprograms. However, recent trends in text similarity detection have been influenced by the growth of artificial intelligence, data mining, information retrieval, natural language processing, and soft computing. Hence, the present study in document similarity detection focuses not only on identifying copying text but also on determining the ideas presented in different words.

10.26555/jifo.v16i3.a23471

In this paper, we present an advanced survey in text similarity detection from articles paper started in 2015. Investigation from an earlier study was reviewed by [1], but the papers limited the focus only to review plagiarism. This paper investigates text similarity detection in digital documents and highlights the significance of similarity patterns together with methods for detecting the similarity. Thus, the contributions of this paper are twofold. First, we presented a review of existing works that encompass the text similarity by various patterns, regardless of the purpose for finding similarity. Second, we described the methods for detecting similarity based on recent techniques. The review methodology of this paper will be explained in the following section. Then, the results and discussion of the are presented, followed by a conclusion section.

2. Method

This paper carried out a systematic literature review and investigated all available research evidence to better understand further challenges in text similarity detection. The design was based on the systematic literature review framework proposed by [2]. It begins with identifying initial research questions addressed in the study, followed by the searching process with inclusion and exclusion criteria. Then, the review protocol was developed to collect, analyze, and synthesize the data. Furthermore, the research questions and motivation to find challenges and opportunities addressed in this literature review are as follows:

1) RQ1. What are the main issues addressed in text similarity research?
2) RQ2. What are the available methods proposed in the existing literature?
3) RQ3. What are the current limitation and future challenges for text similarity research?

Since the research questions were developed to focus the review, the following description, such as population, intervention, outcomes, and context, are required to sharpen the results. The population in this paper is document, system, application, and software. The intervention includes text similarity, document similarity matching, similarity detection, methods, techniques, and datasets. Successful finding text similarity and accuracy detection are the outcomes, while the studies in industry and academia will be chosen as the context for the review. Based on the research questions above, we exclude the discussions regarding the best methods or performance of accuracy metric from the existing methods. However, we discuss the improvement of existing methods solved the issues raised.

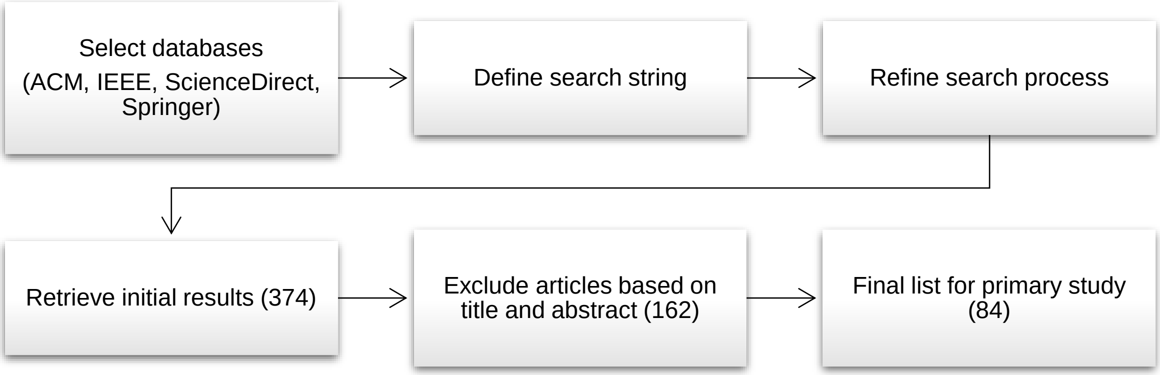

Select databases
(ACM, IEEE, ScienceDirect, Springer)
Define search string
Refine search process
Retrieve initial results (374)
Exclude articles based on title and abstract (162)
Final list for primary study (84)

Fig. 1. Systematic literature review process

As shown in Fig. 1, we performed some activities to collect the research papers included in this review. The first activity is selecting an appropriate set of databases. In order to find the relevant articles, the following digital databases are used: ACM Digital Library, IEEE eXplore, ScienceDirect, and Springer. Afterward, we queried from the selected digital libraries using a defined search string. Construction of search strings is identified from research questions, population, and intervention. The following search string was used for querying the results:

```
(document OR text OR system OR software) AND (lexical OR semantic) AND (similar* OR match* OR detect* OR plagiarism)
```

The search string was adjusted according to the specific requirement of databases. Then, a refinement was conducted using the selection criteria by searching on title, keyword, and abstract. The period of searching was limited to 2015 until 2021. We only included full research articles published in journals to ensure our survey covers primary sources or original materials. Hence, we excluded all articles that appeared in conferences, proceedings, book chapters, case reports, and short communications. The articles were limited only to English.

The initial results after refinement were 374 papers. The first screening from title and abstract had resulted in 162 articles. Further screening from the full text was conducted by searching for studies without a robust methodology, validation, experimental results, and context other than text similarity matching. Finally, 84 articles are collected for the final study in this paper.

3. Results

This section summarizes the results of the study along with answers to the problem questions in this study. Using the searching procedure above, we have identified 84 articles as the final listed papers for further investigation. The annual distribution of the papers in our literature searching is presented in Fig. 2. The trend was declining in 2018, but more papers were published after that. The overall trend over time of the publishing frequency had shown that the publication numbers are growing over time. Up to now, the studies in text similarity detection are still prevailing within the academic community. The most active journals covering text similarity topics are IEEE Access (9) and Expert System with Applications (6). They were followed by Empirical Software Engineering, Information Processing and Management, Journal of Biomedical Informatics, and Knowledge-Based Systems with four articles respectively. From the selected studies, articles that contributed the most citation on text similarity detection are Al-Anzi [3], Pilehvar [4], and Al-Smadi [5]. It should be noted that in terms of the most active researchers, we recognized Ragkhitwetsagul [6], [7] and Vani [8], [9], who contributed very well in our listing.

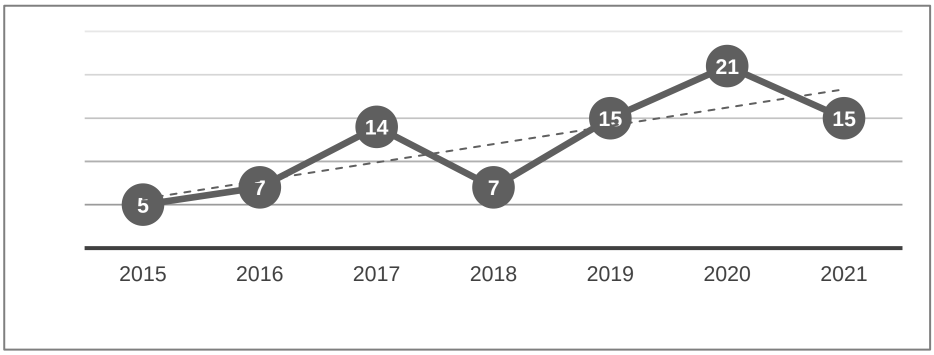


Fig. 2. Annual distribution of primary articles

3.1. Research Issues

There are many issues covered in the selected studies papers. We identified issues that need to be addressed in each paper carefully. Based on their appearance and classified into six following research topics:

1) *Similarity detection methods.* By far, this is the most widely investigated issue from the selected papers. The studies in this group challenge the problem of finding the best method to identify the part of the text that is similar within a context [10], [11]. Another problem that needs to be addressed is related to finding in what way precisely a part of the text can be recognized as same as others [12]. The researchers commonly used filtering to reduce the number of comparisons by skipping the potentially similar parts in the document [13].

2) *Discover similarity patterns.* The focus of this area is to find the similarity structures, rules, or patterns from the suspicious document. Finding the similarity pattern is advantageous for early recognition before conducting a deeper similarity checking. The tasks in this area include explaining the pattern of similarity, finding the rules, and recognizing the content structure based on the document. The simple method to discover the pattern is by brute force the suspected text and compare the patterns

with already known patterns [14]. Another critical issue in this area is constructing the pattern in a cluster or a semantic network structure [15].

3) *Develop corpus or knowledge base.* A corpus is an extensive collection of structured text that is usually stored in a database. The presence of a corpus in text similarity research is a consequence, as presented in the previous issue. There are several free and publicly available online corpus in English, like WordNet [9], Microsoft [16], Reuters [13], Standford [17], and PAN [11]. The problem in this group includes a corpus development for specific needs [18]–[20] or corpus for non-English languages [3], [21]. As external knowledge, some studies combined several sources of corpus for their research [22].

4) *Performance evaluation.* The main issue in this criteria relates to measuring the proposed method's performance for a presented problem. The answer for this problem is complicated and often needs comparison with previous results as the baseline for comparison. Many evaluation methods are presented in the literature, but the majority evaluate the method in terms of robustness, accuracy, and time. The common methods among performance metrics are matching accuracy, precision, recall, granularity, and F1-measure [8], [3], [23], [24]. Contrary to accuracy, several studies analyze the performance with respect to the error metrics by employing root mean square error (RMSE) and failed detection ratio (FDR) [11], [25]. In addition, the processing time was measured by counting the execution time when running the proposed methods with various inputs. However, the result of execution time is often limited by the resources, hardware, and runtime environment [26]. Another reported study compared the accuracy results with an external system like Turnitin [27].

3.2. Fields of Applications

The increasing amount of data and information has rapidly expanded the application of text similarity matching. As shown in Fig. 3, the text similarity has been applied in many areas. For many years, the *plagiarism detection system* has been the most popular application among research papers [28], [29]. Although the plagiarism detection system has been available for many years, the rapid development of the internet to access a vast number of electronic documents has facilitated plagiarism. The forms of plagiarism have varied over time. The research areas that receive increasing attention are cross-language plagiarism detection [18], applying heuristic and machine learning methods [28], [30], multiple data sources corpus [12], adaptive and automatic plagiarism detection [24], [31].

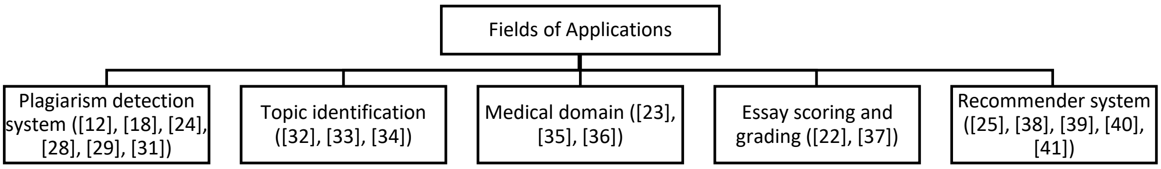

Fields of Applications
Plagiarism detection system ([12], [18], [24], [28], [29], [31])
Topic identification ([32], [33], [34])
Medical domain ([23], [35], [36])
Essay scoring and grading ([22], [37])
Recommender system ([25], [38], [39], [40], [41])

Fig. 3. The application areas of text similarity matching

Text similarity is also useful to *identify the relevant topic* from a given document. Relevant document retrieval is essential for some domains. An example of this application is legal information retrieval to find legal document similarity [32], [33]. The source of legal information is derived from various documents, such as court transcripts, legislation documents, verdicts, and judgment generated from the legal process. These documents are required for developing argumentation and decision making for a legal professional. Another example is in scientific research for patent application [34]. In order to issue a patent, examiners compare all previous patents and articles that are relevant to the application. Due to the long span period of collected documents, some language changes and has new meaning or concepts. Hence, the process of document retrieving has become more complex.

In the *medical domain*, text similarity can be applied matching patients to clinical trials [35] or drugs [36]. The existing approach for matching is often time-consuming for medics and patients. The source of unstructured information can be found from progress notes, discharge summaries, pathology, or radiology reports. Another application in the medical domain is the development of automatic Q&A systems using text matching [23]. The syntax and meaning of medical words are more complex, with many complicated medical terms to extract. Meanwhile, the text in Q&A is short and increases the difficulty of retrieving the correct information.

The practice of *automatic essay scoring and grading* using text similarity matching is widely used in massive learning systems nowadays. Essay scoring or grading is often operated as an e-learning extension for student assignments. The utilization saves instructors time and provides immense feedback to students. The reference collection in this approach is the collection of academic documents. Some studies developed their own collection and separated the reference from the internet to focus only on giving detailed analysis related to the query document [22]. The challenge in this area is not only for detecting plagiarism but also for advising the students' work and recommending the appropriate score [37].

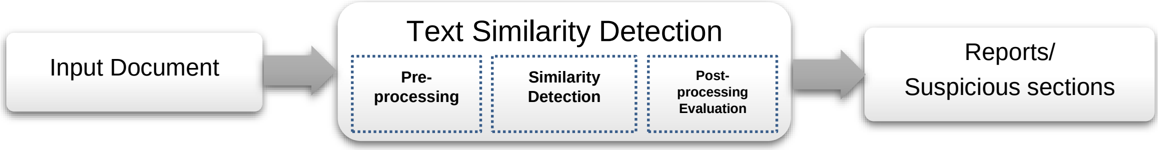
Text similarity also has a considerable impact on *recommender systems* to guide the user with appropriate content based on their preferences. By using references from user ratings and reviews, the recommender system uses text similarity for matching the user profile. Recommender system can be implemented for e-commerce [25], [38], online repositories [39], [40], or part of the specific system like in [41] for supporting emergency response plan.

3.3. Approaches

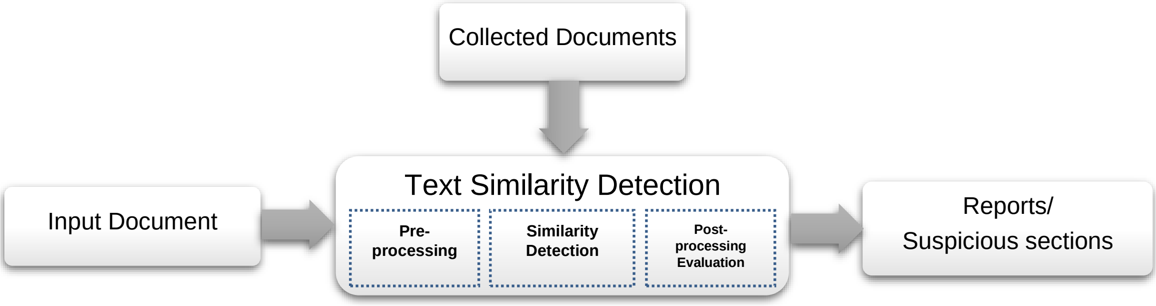
There are two general approaches to determine text similarity: *intrinsic detection* and *extrinsic detection* [1], [42]. Intrinsic detection examines dissimilarities within a document, while extrinsic detection examines similarities across documents [11]. The generic process of the intrinsic detection approach analyzes the input documents without comparing them to a collection of documents. This approach employs a complete analysis known as stylometry. The purpose is to examine the changes in writing style and thus identify the author of the document [43]. Extrinsic detection differs from intrinsic detection in terms of source of the document collection. The reference collection in extrinsic detection consists of documents or other resources used for contrasting suspicious documents. Apart from these approaches, text similarity detection can be viewed as an information retrieval task with the basis of the query from small keywords to large documents. Fig. 4 (a) shows that the process of intrinsic detection only requires an input document without referencing collected documents. Then, text segmentation operates by dividing the document into smaller sections, paragraphs, sentences, or words. The stylometric inspect various aspect of the segmented text including frequency of words, characters types, semantic features, and context-specific keywords.

The process of extrinsic detection is shown in Fig. 4 (b). It typically has two core inputs: suspected documents and collected documents. Before the system started to work on detecting a suspected document, the reference collection must be available in the system. The source of reference can be built from collected documents, web pages, or available corpus. The initial detection process is preprocessing, which aim to remove unwanted object in the document and keep only important information for further analysis. Then, similarity detection can be conducted in several ways, depending on the purpose of finding similarity, techniques, and application. Some studies proposed a hybrid approach by combining intrinsic and extrinsic to improve the accuracy of the results [6], [44].

Regarding the research method, the tasks of text similarity usually include measuring the similarity of the words, sentences, or segmented documents based on several features. The post-processing or evaluation phase is basically representing the comparison results that have been obtained before. The comparison can be presented in several ways, including showing the similar part, similarity form or pattern, degree of plagiarism, presenting possible sources, and ranking the results [7], [9], [19].



(a) Intrinsic approach


Collected Documents
Input Document
Text Similarity Detection
Pre-processing
Similarity Detection
Post-processing Evaluation
Reports/ Suspicious sections

(b) Extrinsic approach

Fig. 4. Generic approach for text similarity detection

3.4. Preprocessing

Preprocessing is an essential step in text similarity detection. The process aims to transform the text into a predictable and analyzable form for further analysis. The operation is a combination of removing unwanted objects and keeping required information in the domain of analysis. There are several types of preprocessing techniques available in the literature depending on the approach and domain. A detailed description of the typical preprocessing techniques is given in the following.

1) *Lowercasing.* This process is one of the simplest but most effective forms for preprocessing [45]. For example, the words "Jakarta" and "jakarta" have similar meanings, but they represent two different words in the vector space model when not converted to the lower case. The implementation can reduce the dimensions and significantly improve the consistency of the output [20]. This operation is applicable to most text similarity approaches where the dataset is not very large.

2) *Tokenization & segmentation.* Given a text sequence, tokenization aims to segment each written text into meaningful units, such as independent words, sentences, or topics. A simple step to segment text is to map each character into a set of individual keywords [46]. The implementation depends on the language, such as NLTK for English [47], UCAS for Chinese [48], [49], or HAC for Arabic [21]. It can also be mplemented to converted documents into a string of tokens [50] or chop paragraphs into unigrams and bigrams [24].

3) *Stopwords removal.* Stopwords are a set of commonly used words that have no concrete meanings in a language. Examples of stopwords in English are prepositions ("in", "at"), pronouns ("she", "he"), and articles ("an", "the"). The rationale behind removing the stopwords is to help focus on the important words and improving accuracy. Hence, this process could reduce time and memory space for comparisons and increase the speed of processing. Among other operations, this operation is widely used for preprocessing as reported in [3], [12], [18], [20], [48].

4) *Stemming.* The aim of stemming is to change the word into its base or root form. The word can be presented in various forms. For instance, "connection", "connected", and "connecting" stemmed into the word "connect". The stemming helps to focus the language information and reduce the calculation for subsequent steps. The stemming process includes eliminating the prefixes and suffixes, changing words in different tenses (past, continuous), changing alternative spelling and error, and replacing multiple constructions of words (noun, verb). There are various algorithms for stemming in NLP, but they might not be applicable for all languages. The typical algorithm for English is the Porters and Lovins algorithm [47], [20]. The Indo-European language uses the Snowball algorithm [51], [12].

5) *Text cleaning.* This preprocessing technique is rather complex, depending on the requirement. The operation typically applied for non-ASCII characters [5] or part of source code documents [50] which has unidentified text. The step is typically modified by discovering unwanted words, hyperlinks, header, hash-tags, unimportant identifiers, comments, and excess whitespace [39], [52].

6) *Characters removal.* This is also a widely used operation in preprocessing [3], [18], [20], [24]. A sentence often contains special characters or symbols, such as dot, colon, dash, or non-ASCII characters [45].

Those characters do not contribute to measuring the similarity. Hence, this step removes all unnecessary characters with still maintains the structure of original texts [9], [17], [35].

7) *Characters replacement.* Some articles use different texts to share similar concepts. Depending on the context, this process will replace the symbols with their names and vice versa. For example, both "AI" and "Artificial Intelligence" have similar meanings, though they are different in words [16]. Some symbols need to be replaced with their names, such as "%" with percent, '1000' with thousand [45], [46].

8) *Format conversion.* The early studies in text similarity matching often described format conversion from documents like PDF, DOC, and HTML into a structured format [42]. However, format conversion in a recent study is more related to converting the text into a customized data structure for further analysis. For instance, the conversion task in [46], [52] is to transform the whole document into some structured paragraphs with a unique IDE on the document's physical structure.

The preprocessing approaches above are typically employed for extrinsic detection. In contrast, the intrinsic detection often limits or avoid preprocessing step to maintain useful information within the document. For instance, intrinsic detection retains punctuation and keeps format differentiation to not lose the needed information for further analysis.

3.5. Similarity Detection Method

Regardless of intrinsic and extrinsic approaches, there are several different ways of detecting textual similarity. As shown in Fig. 5, the method for finding text similarity typically will fall into one of the following categories: lexical, syntactic, semantic, structural, and hybrid.

1) *Lexical-similarity method*

Lexical matching is a straightforward yet one of the most widely used methods for measuring text similarity. It encompasses character or word matching for providing similarity. The lexical similarity of two texts is measured by the degree of similarity between two sets of the same vocabulary. The two sentences with a complete overlap between the words will have a lexical similarity score of 1 (high). In contrast, a score of 0 (low) means no intersection of similar words between the two texts. String-based operates on character composition and string sequences, while term-based measures similarity or dissimilarity of shared terms. The common algorithms for term-based matching are Longest Common Subsequence (LCS), Levenshtein, and N-gram. LCS focuses on the length of the contiguous chain between strings. LCS measures the longest total length where appears in a similar order as in other strings [45]. Levenshtein also uses the distance factor to measure the similarity between two strings. The algorithm computes the edit distance when a part of the string is moved to another string instead of detecting a single change in the string. Previous studies have confirmed its effectiveness [22], [23], [26]. N-gram used a sequence of strings and measured the similarity of sub-sequence of *n* words from a given sequence text. The similarity in n-gram is measured on the basis of the distance between each character [21]. In [20] employs character-level n-gram for identifying paraphrase plagiarism on the passage level.

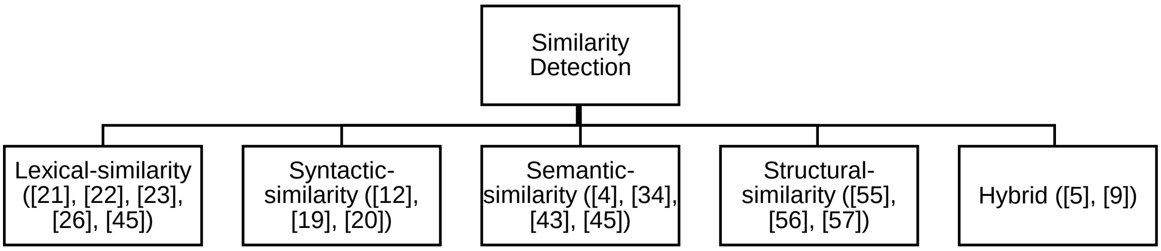

Similarity Detection
Lexical-similarity ([21], [22], [23], [26], [45])
Syntactic-similarity ([12], [19], [20])
Semantic-similarity ([4], [34], [43], [45])
Structural-similarity ([55], [56], [57])
Hybrid ([5], [9])

Fig. 5. Classification of text similarity detection methods

2) *Syntactic-similarity method*

Syntactic is typically manifested as part of speech (POS) and operates at the word level. POS in English includes verbs, nouns, pronouns, adjectives, adverbs, prepositions, conjunction, and interjections. The aim is to measure the similarity words but ignore some common words to reduce the percentage similarity.

Hence, the information from the syntactic structure can be used to reduce the workload of subsequent semantic analysis [19]. The operation in syntactic matching considers the structure and relationship of the sentence for the analysis. This method works on sentence-level by utilizing PoS tagging to determine the syntactic structure of sentences [12], [20].

3) *Semantic-similarity method*

The similarity in semantic matching is measured by the word meaning rather than the character similarity. In order to determine the degree of similarity meaning between words, the semantic information network was established through the collection of a knowledge base, corpus, or a combination of both [12], [44]. The challenge in the knowledge-based method is the way of calculating the similarity between two terms based on the information derived from knowledge sources. Several studies utilize WordNet and NLTK to measure common semantic similarity [4], [34]. Semantic similarity in the corpus-based method also measured the similarity, but the knowledge is derived from large corpora. The principle of distributional hypothesis is applied in semantic similarity. It means the similar words occurred together and frequently are expected to appear in similar contexts [53]. In earlier studies of semantic similarity, LDA, LSA, NGD, and vector are among the most popular method to compute semantic similarity [22], [43], [46]. However, the trend in recent studies exploits the neural networks to enhance the performance of semantic measurement. The commonly used techniques are CNN, LSTM, and transformer-based model [23], [45], [54].

4) *Structural-similarity method*

Unlike previous methods that consider similarity in terms of words and meaning, the structural similarity uses the structure of information in the document as the point of analysis. With the growth of electronic documents, the structure of sentences tends to leave a fingerprint and pattern that can be used to find the similarity. Structural can be identified from the block and content of the document. Block can be gathered specifically from webpage documents, while the content is recognized in any document. Research in [55] has shown potential results when exploiting structural information to measure word order similarity. Their experiment reported that the structural information achieved good results and outperformed the string-based approach in a benchmark test. The structural similarity is also useful for detecting source code similarity. As reported in [56], [57], the structural information of source code could show the part of plagiarism modification.

5) *Hybrid method.*

Since each of the above methods has its advantage and disadvantage, researchers often proposed a hybrid model by combining two or more methods to improve the performance of the current text similarity method. For instance, the syntactic method has a basic limitation to identify plagiarism. Thus, the study in [9] proposed a combined syntactic and semantic method to identify fragmented plagiarism. The evaluation results show the potential use of the method for actual plagiarism cases. Furthermore, the challenge for non-English language is quite complex in terms of their recognized characters. The hybrid methods provide an opportunity to solve the problems. As reported in [5], the combination of lexical, syntactic, and semantic features effectively identifies paraphrases in Arabic.

4. Discussions

We have presented the issues, current application, and methods in the previous section. Furthermore, this section highlights several efforts to improve the existing problem and discusses the results focusing on answering the latest research question.

4.1. Method Improvements

There are a number of important method improvements in our investigation. We highlight the key methods used in many studies to improve the accuracy of their model. First, *selecting the appropriate preprocessing algorithm* to enhance the matching process. The proper preprocessing algorithm is a key step in this field. By manipulating or dropping unessential objects in the dataset, it will gradually reduce the problems in the following process. The preprocessing can be applied by *reducing data dimensionality* or *choosing the best feature* in the data set. Different preprocessing methods have been proposed to reduce data dimensionality, but the task is typically classified into cleansing, editing, and reduction. This paper has presented several preprocessing methods for identifying and removing unneeded data. However, not all methods can contribute to improving the accuracy of the result. It is important to understand the dataset as the data from the real-world always have noise or missing values.

Furthermore, an important step in preprocessing is a feature selection process. Machine learning uses feature selection to reduce the input variables to improve the performance and reduce computational cost of the model. Comparing to fine-tuning parameters or boosting algorithms, there is nothing similar to a good feature selection [58]. The feature selection is performed after removing unwanted information in the dataset to improve the efficiency of the algorithm. In [59], text filtering, segmentation, tagging, and stop word removing were used to clean the data. Then, feature selection was carried out, and resulted in nouns and verbs were retained in the experimental data. Some researchers have developed their own algorithms for this purpose. A study in [8] proposed two-phase feature selection by implementing rank-based feature and heuristic approach. Classification is then performed once the best feature is selected.

Another key method is combining several algorithms that have become a common trend in this field. Since the task of finding similarities has become sophisticated and more complex, a simple paraphrasing or idea adoption will need more effort to be recognized. Thus, a hybrid method is one of the most common procedures to resolve the issue. The combination of the methods can be implemented since preprocessing. The feature selection and extraction are generally exploited to reduce data complexity and improve the similarity measurement. The study in [59] has shown that using feature selection after preprocessing short text could improve the accuracy of the proposed model.

As reported in [12], the hybrid approach combined between lexical, semantic, and syntactic similarity methods. The lexical corpus contains linguistic information between text documents was developed. Then, knowledge of semantic and syntactic was computed. The results have shown the capability to detect idea plagiarism, such as rewording, paraphrasing, verbatim copy, and sentence transformation. The practice of combining several methods is also common for detecting similarities in Arabic. Previous study has reported that the cosine similarity performed well in the Arabic language [3]. Combining this method with latent semantic indexing for providing semantic information could improve the classifier's performance. Another proposed method by [8] has reported an effective plagiarism matching when integrating syntax-semantic text analysis along with structural and citation-based analysis.

4.2. Future Challenges

The focus of this section is presenting the limitation of the current study and further research opportunities. In our investigation, most research in this area dominates by proposing new methods to measure the text similarity problem. However, the remaining unsettled problems for a long time are (1) "*How to determine the similarity of text?*" and (2) "*Which method is used to detect text similarity?*".

In our analysis, the first challenge of text similarity is not about the way we find identical notation between texts. Instead, the challenge is how to define in what way the text are considered similar. With the increase of the information, the ways to judge similarity are increased too. For this reason, future studies need to specify the task and context before measuring the similarity. The task of detecting literal plagiarism from copying and pasting text can be measured using lexical similarity. When the task is finding idea plagiarism, the synonyms and varieties should be included in the judgment too. For example, the words "villa", "mansion", "barrack", and "flat" are similar for the context *house*. Other words such as "laptop", "server", "mobile", "desktop", and "tablet", although not being synonyms, still can be classified as the *device*. Without providing the context, the task of similarity measure will produce an uncertain judgment. When particular words have a strong correlation, they can be formed in a feature set which helps to improve understanding of the context.

Furthermore, the second challenge is which method can be used for detecting a specific case of similarity. Regardless of the potential contribution, each method has own limitation and drawback. For instance, the strategy to combine several methods had the potential use to solve the problem. However, this approach will increase processing time and may not be suitable for all cases with time issues. Another problem is combining the heuristic and machine learning to improve processing performance. In reality, the required data to improve performance is often not available. This cold start problem in the early stage of implementation has been a challenge for many researchers. The lexical similarity is powerful enough to detect similarity on the surface. Thus, the small fraction similarity from copy-paste activity will be recognized, but not the similarity from paraphrasing or idea adoption. In fact, most of the existing similarity detection systems fail to detect plagiarism from summary or paraphrasing. If the semantic and structural methods are included in the process, it will increase the time to find the similarity. Therefore, the above question will always be relevant, and understanding how the methods solve the problem domain is essential in this research domain. We realized various modern methods in data science have been available that might be out of our findings. The methods, such as logic-based, intelligent-based, and learning-based, were proven to deal with the finding pattern from the collection of information. More research is needed to explore the insights and establish a greater degree of accuracy on this matter.

5. Conclusion

Measuring text similarity has been one of the most challenging tasks in NLP. Many studies have proposed various methods for text similarity matching since the tasks have become increasingly more sophisticated. This paper attempts to present the discussion about related issues and advantages of available methods from the survey of the recent literature. Our analysis found that most studies in this field have attempted to investigate the appropriate similarity detection methods, discover similarity patterns, develop a corpus or knowledge base, or evaluate the proposed method's performance. The text similarity also has broader application, mainly focused on plagiarism detector, to identify the topic from a document, for automatic essay scoring, and become integrated part of other systems, such as recommender systems.

We continued the investigation by analyzing the most commonly used approaches and methods in this field. We recognized that each method has its benefit and drawback. Hence, selecting the best method is not possible without understanding the context of the document. The focus of recent research is shifted towards building a hybrid method with regards to compensate for the computational resource and performance. The future challenge in this area needs to address the necessity of understanding the context in order to determine the similarity. Another challenge is related to select the best method that worked in a certain problem. Finally, this survey should contribute to a better understanding of the foundation of text similarity research for the scientific community.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding statement. The research activity leading to the publication has been funded by the Ministry of Research, Technology and Higher Education Indonesia with contract number 311/E4.1/AK.04.PT/2021, 3281.11/LL5/PG/2021, 191/KONTRAK-LPPM/AMIKOM/VII/2021.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

[1] S. M. Alzahrani, N. Salim, and A. Abraham, "Understanding plagiarism linguistic patterns, textual features, and detection methods," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 42, no. 2, pp. 133–149, 2012.

[2] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering–a systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2009.

[3] F. S. Al-Anzi and D. AbuZeina, "Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 29, no. 2, pp. 189–195, 2017, doi: 10.1016/j.jksuci.2016.04.001.

[4] M. T. Pilehvar and R. Navigli, "From senses to texts: An all-in-one graph-based approach for measuring semantic similarity," *Artif. Intell.*, vol. 228, pp. 95–128, 2015, doi: 10.1016/j.artint.2015.07.005.

[5] M. AL-Smadi, Z. Jaradat, M. AL-Ayyoub, and Y. Jararweh, "Paraphrase identification and semantic text similarity analysis in Arabic news tweets using lexical, syntactic, and semantic features," *Inf. Process. Manag.*, vol. 53, no. 3, pp. 640–652, 2017, doi: 10.1016/j.ipm.2017.01.002.

[6] C. Ragkhitwetsagul, J. Krinke, and D. Clark, *A comparison of code similarity analysers*, vol. 23, no. 4. Empirical Software Engineering, 2018.

[7] C. Ragkhitwetsagul and J. Krinke, “Siamese: scalable and incremental code clone search via multiple code representations,” *Empir. Softw. Eng.*, vol. 24, no. 4, pp. 2236–2284, 2019, doi: 10.1007/s10664-019-09697-7.

[8] K. Vani and D. Gupta, “Text plagiarism classification using syntax based linguistic features,” *Expert Syst. Appl.*, vol. 88, pp. 448–464, 2017, doi: 10.1016/j.eswa.2017.07.006.

[9] K. Vani and D. Gupta, “Unmasking text plagiarism using syntactic-semantic based natural language processing techniques: Comparisons, analysis and challenges,” *Inf. Process. Manag.*, vol. 54, no. 3, pp. 408–432, 2018, doi: 10.1016/j.ipm.2018.01.008.

[10] E. Gharavi, H. Veisi, and P. Rosso, “Scalable and language-independent embedding-based approach for plagiarism detection considering obfuscation type: no training phase,” *Neural Comput. Appl.*, vol. 32, no. 14, pp. 10593–10607, 2020, doi: 10.1007/s00521-019-04594-y.

[11] C. Yang, C. Shie, J. Lee, C. Hung, W. Chih, and F. Liu, “Using word semantic concepts for plagiarism detection in text documents,” *Inf. Retr. J.*, no. 0123456789, 2021, doi: 10.1007/s10791-021-09394-4.

[12] M. Sahi and V. Gupta, “A Novel Technique for Detecting Plagiarism in Documents Exploiting Information Sources,” *Cognit. Comput.*, vol. 9, no. 6, pp. 852–867, 2017, doi: 10.1007/s12559-017-9502-4.

[13] M. Alewiwi, C. Orencik, and E. Savaş, “Efficient top-k similarity document search utilizing distributed file systems and cosine similarity,” *Cluster Comput.*, vol. 19, no. 1, pp. 109–126, 2016, doi: 10.1007/s10586-015-0506-0.

[14] N. M. Tien and C. Labbé, “Detecting automatically generated sentences with grammatical structure similarity,” *Scientometrics*, vol. 116, no. 2, pp. 1247–1271, 2018, doi: 10.1007/s11192-018-2789-4.

[15] T. Botsis, J. Scott, E. J. Woo, and R. Ball, “Identifying similar cases in document networks using cross-reference structures,” *IEEE J. Biomed. Heal. Informatics*, vol. 19, no. 6, pp. 1906–1917, 2015, doi: 10.1109/JBHI.2014.2345873.

[16] H. T. Nguyen, P. H. Duong, and E. Cambria, “Learning short-text semantic similarity with word embeddings and external knowledge sources,” *Knowledge-Based Syst.*, vol. 182, p. 104842, 2019, doi: 10.1016/j.knosys.2019.07.013.

[17] M. Atabuzzaman, M. Shajalal, M. E. Ahmed, M. I. Afjal, and M. Aono, “Leveraging Grammatical Roles for Measuring Semantic Similarity between Texts,” *IEEE Access*, vol. 9, pp. 62972–62983, 2021, doi: 10.1109/ACCESS.2021.3074747.

[18] N. Ehsan and A. Shakery, “Candidate document retrieval for cross-lingual plagiarism detection using two-level proximity information,” *Inf. Process. Manag.*, vol. 52, no. 6, pp. 1004–1017, 2016, doi: 10.1016/j.ipm.2016.04.006.

[19] F. Ullah, J. Wang, M. Farhan, S. Jabbar, Z. Wu, and S. Khalid, “Plagiarism detection in students’ programming assignments based on semantics: multimedia e-learning based smart assessment methodology,” *Multimed. Tools Appl.*, vol. 79, no. 13–14, pp. 8581–8598, 2020, doi: 10.1007/s11042-018-5827-6.

[20] F. Sánchez-Vega, E. Villatoro-Tello, M. Montes-y-Gómez, P. Rosso, E. Stamatatos, and L. Villaseñor-Pineda, “Paraphrase plagiarism identification with character-level features,” *Pattern Anal. Appl.*, vol. 22, no. 2, pp. 669–681, 2019, doi: 10.1007/s10044-017-0674-z.

[21] E. Al-Thwaib, B. H. Hammo, and S. Yagi, “An academic Arabic corpus for plagiarism detection: design, construction and experimentation,” *Int. J. Educ. Technol. High. Educ.*, vol. 17, no. 1, pp. 1–26, 2020, doi: 10.1186/s41239-019-0174-x.

[22] Z. Rahimi, D. Litman, R. Correnti, E. Wang, and L. C. Matsumura, *Assessing Students’ Use of Evidence and Organization in Response-to-Text Writing: Using Natural Language Processing for Rubric-Based Automated Scoring*, vol. 27, no. 4. International Journal of Artificial Intelligence in Education, 2017.

[23] Z. Wu, J. Liang, Z. Zhang, and J. Lei, “Exploration of text matching methods in Chinese disease Q&A systems: A method using ensemble based on BERT and boosted tree models,” *J. Biomed. Inform.*, vol. 115, no. January, p. 103683, 2021, doi: 10.1016/j.jbi.2021.103683.

[24] A. S. Altheneyan and M. E. B. Menai, “Automatic plagiarism detection in obfuscated text,” *Pattern Anal. Appl.*, vol. 23, no. 4, pp. 1627–1650, 2020, doi: 10.1007/s10044-020-00882-9.

[25] N. Ghasemi and S. Momtazi, “Neural text similarity of user reviews for improving collaborative filtering recommender systems,” *Electron. Commer. Res. Appl.*, vol. 45, no. November 2020, p. 101019, 2021, doi: 10.1016/j.elerap.2020.101019.

[26] H. Kaur and R. Maini, “Assessing lexical similarity between short sentences of source code based on granularity,” *Int. J. Inf. Technol.*, vol. 11, no. 3, pp. 599–614, 2019, doi: 10.1007/s41870-018-0213-1.

[27] S. F. Hussain and A. Suryani, “On retrieving intelligently plagiarized documents using semantic similarity,” *Eng. Appl. Artif. Intell.*, vol. 45, pp. 246–258, 2015, doi: 10.1016/j.engappai.2015.07.011.

[28] V. Kuppili, M. Biswas, D. R. Edla, K. J. R. Prasad, and J. S. Suri, “A Mechanics-Based Similarity Measure for Text Classification in Machine Learning Paradigm,” *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 2, pp. 180–200, 2020, doi: 10.1109/TETCI.2018.2863728.

[29] W. Wali, B. Gargouri, and A. Ben Hamadou, *An Enhanced Plagiarism Detection Based on Syntactico-Semantic Knowledge*, vol. 941. Springer International Publishing, 2020.

[30] L. lei Kong, Z. mao Lu, H. liang Qi, and Z. yuan Han, “A machine learning approach to query generation in plagiarism source retrieval,” *Front. Inf. Technol. Electron. Eng.*, vol. 18, no. 10, pp. 1556–1572, 2017, doi:

10.1631/FITEE.1601344.

[31] S. M. Darwish and M. M. Moawad, *An Adaptive Plagiarism Detection System Based on Semantic Concept and Hierarchical Genetic Algorithm*, vol. 1058. Springer International Publishing, 2020.

[32] R. S. Wagh and D. Anand, “Legal document similarity: a multi-criteria decision-making perspective,” *PeerJ Comput. Sci.*, vol. 6, p. e262, 2020.

[33] A. Mandal, R. Chaki, S. Saha, K. Ghosh, A. Pal, and S. Ghosh, “Measuring similarity among legal court case documents,” *ACM Int. Conf. Proceeding Ser.*, pp. 1–9, 2017, doi: 10.1145/3140107.3140119.

[34] S. Morsy and G. Karypis, “Accounting for language changes over time in document similarity search,” *ACM Trans. Inf. Syst.*, vol. 35, no. 1, 2016, doi: 10.1145/2934671.

[35] H. Hassanzadeh, S. Karimi, and A. Nguyen, “Matching patients to clinical trials using semantically enriched document representation,” *J. Biomed. Inform.*, vol. 105, no. June 2019, p. 103406, 2020, doi: 10.1016/j.jbi.2020.103406.

[36] S. Fathiamini *et al.*, “Rapamycin − mTOR + BRAF = ? Using relational similarity to find therapeutically relevant drug-gene relationships in unstructured text,” *J. Biomed. Inform.*, vol. 90, no. November 2018, p. 103094, 2019, doi: 10.1016/j.jbi.2019.103094.

[37] J. Wiley *et al.*, *Different Approaches to Assessing the Quality of Explanations Following a Multiple-Document Inquiry Activity in Science*, vol. 27, no. 4. International Journal of Artificial Intelligence in Education, 2017.

[38] J. Chambua, Z. Niu, A. Yousif, and J. Mbelwa, “Tensor factorization method based on review text semantic similarity for rating prediction,” *Expert Syst. Appl.*, vol. 114, pp. 629–638, 2018, doi: 10.1016/j.eswa.2018.07.059.

[39] P. T. Nguyen, J. Di Rocco, R. Rubei, and D. Di Ruscio, “An automated approach to assess the similarity of GitHub repositories,” *Softw. Qual. J.*, vol. 28, no. 2, pp. 595–631, 2020, doi: 10.1007/s11219-019-09483-0.

[40] M. Oppermann, R. Kincaid, and T. Munzner, “VizCommender: Computing text-based similarity in visualization repositories for content-based recommendations,” *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 495–505, 2021, doi: 10.1109/TVCG.2020.3030387.

[41] W. Guo, Q. Zeng, H. Duan, W. Ni, and C. Liu, “Process-extraction-based text similarity measure for emergency response plans,” *Expert Syst. Appl.*, vol. 183, no. May, p. 115301, 2021, doi: 10.1016/j.eswa.2021.115301.

[42] T. Foltýnek, N. Meuschke, and B. Gipp, “Academic Plagiarism Detection: A Systematic Literature Review,” *ACM Comput. Surv.*, vol. 52, no. 6, p. Article 112, 2019, doi: 10.1145/3345317.

[43] J. D. Velásquez, Y. Covacevich, F. Molina, E. Marrese-Taylor, C. Rodríguez, and F. Bravo-Marquez, “DOCODE 3.0 (DOcument COpy DEtector): A system for plagiarism detection by applying an information fusion process from multiple documental data sources,” *Inf. Fusion*, vol. 27, pp. 64–75, 2016, doi: 10.1016/j.inffus.2015.05.006.

[44] A. Abdi, S. M. Shamsuddin, N. Idris, R. M. Alguliyev, and R. M. Aliguliyev, “A linguistic treatment for automatic external plagiarism detection,” *Knowledge-Based Syst.*, vol. 135, pp. 135–146, 2017, doi: 10.1016/j.knosys.2017.08.008.

[45] H. Shahmohammadi, M. H. Dezfoulian, and M. Mansoorizadeh, “Paraphrase detection using LSTM networks and handcrafted features,” *Multimed. Tools Appl.*, vol. 80, no. 4, pp. 6479–6492, 2021, doi: 10.1007/s11042-020-09996-y.

[46] T. Zhang, B. Lee, and Q. Zhu, *Semantic measure of plagiarism using a hierarchical graph model*, vol. 121, no. 1. Springer International Publishing, 2019.

[47] Z. Wu *et al.*, “An efficient Wikipedia semantic matching approach to text document classification,” *Inf. Sci. (Ny).*, vol. 393, pp. 15–28, 2017, doi: 10.1016/j.ins.2017.02.009.

[48] S. Zhou, X. Xu, Y. Liu, R. Chang, and Y. Xiao, “Text Similarity Measurement of Semantic Cognition Based on Word Vector Distance Decentralization with Clustering Analysis,” *IEEE Access*, vol. 7, pp. 107247–107258, 2019, doi: 10.1109/ACCESS.2019.2932334.

[49] S. Yang, R. Wei, J. Guo, and H. Tan, “Chinese semantic document classification based on strategies of semantic similarity computation and correlation analysis,” *J. Web Semant.*, vol. 63, p. 100578, 2020, doi: 10.1016/j.websem.2020.100578.

[50] T. Ohmann and I. Rahal, “Efficient clustering-based source code plagiarism detection using PIY,” *Knowl. Inf. Syst.*, vol. 43, no. 2, pp. 445–472, 2015, doi: 10.1007/s10115-014-0742-2.

[51] W. A. Mohotti and R. Nayak, “Efficient Outlier Detection in Text Corpus Using Rare Frequency and Ranking,” *ACM Trans. Knowl. Discov. Data*, vol. 14, no. 6, 2020, doi: 10.1145/3399712.

[52] I. Alonso and D. Contreras, “Evaluation of semantic similarity metrics applied to the automatic retrieval of medical documents: An UMLS approach,” *Expert Syst. Appl.*, vol. 44, pp. 386–399, 2016, doi: 10.1016/j.eswa.2015.09.028.

[53] J. Yang, Y. Li, C. Gao, and Y. Zhang, “Measuring the short text similarity based on semantic and syntactic information,” *Futur. Gener. Comput. Syst.*, vol. 114, pp. 169–180, 2021, doi: 10.1016/j.future.2020.07.043.

[54] D. Peng, J. Yang, and J. Lu, “Similar case matching with explicit knowledge-enhanced text representation,” *Appl. Soft Comput. J.*, vol. 95, p. 106514, 2020, doi: 10.1016/j.asoc.2020.106514.

[55] M. Farouk, “Measuring text similarity based on structure and word embedding,” *Cogn. Syst. Res.*, vol. 63, pp. 1–10, 2020, doi: 10.1016/j.cogsys.2020.04.002.

[56] C. Oprişa, D. Gavriluţ, and G. Cabău, “A scalable approach for detecting plagiarized mobile applications,” *Knowl. Inf.*

Syst., vol. 49, no. 1, pp. 143–169, 2016, doi: 10.1007/s10115-015-0903-y.

[57] H. Cheers, Y. Lin, and S. P. Smith, “Evaluating the robustness of source code plagiarism detection tools to pervasive plagiarism-hiding modifications,” *Empir. Softw. Eng.*, vol. 26, no. 5, pp. 1–62, 2021, doi: 10.1007/s10664-021-09990-4.

[58] H. T. Nguyen, P. H. Duong, and E. Cambria, “Knowledge-Based Systems Learning short-text semantic similarity with word embeddings and external knowledge sources,” *Knowledge-Based Syst.*, vol. 182, p. 104842, 2019, doi: 10.1016/j.knosys.2019.07.013.

[59] D. Wu, M. Zhang, C. Shen, Z. Huang, and M. Gu, “BTM and GloVe Similarity Linear Fusion-Based Short Text Clustering Algorithm for Microblog Hot Topic Discovery,” *IEEE Access*, vol. 8, pp. 32215–32225, 2020, doi: 10.1109/ACCESS.2020.2973430.