Comparative Analysis of Text-Based CBR Algorithms for Cybercrime Profiling Investigations

Marc Krüger[1,2,*,1]

[1]Stiftung Universität Hildesheim Universitätsplatz 1 31141 Hildesheim, Germany

Abstract

Cybercrime has emerged as a significant threat in the digital age, necessitating the development of effective investigation methods. This paper focuses on the application of text-based Case-Based Reasoning (CBR) algorithms in cybercrime investigations. CBR is an intelligent problem-solving technique that utilizes past experiences to solve new cases, making it well-suited for analyzing textual data in the context of cybercrime. The primary objective of this study is to compare various text-based CBR algorithms and evaluate their performance in handling cybercrime cases. The algorithms under consideration include traditional CBR, k-nearest neighbors (k-NN), and support vector machines (SVM) as baseline methods. Additionally, advanced techniques such as deep learning-based approaches and natural language processing (NLP) will be explored to enhance the effectiveness of the algorithms. The comparative analysis will involve a comprehensive evaluation of the algorithms based on criteria such as accuracy, efficiency, scalability, and interpretability. Several datasets containing cybercrime-related text documents will be used for training and testing purposes. The experiments will be conducted using a robust evaluation framework to ensure unbiased and reliable results. The findings of this research will provide insights into the strengths and weaknesses of different text-based CBR algorithms in the context of cybercrime investigations. The results will assist cybercrime investigators, law enforcement agencies, and researchers in selecting appropriate techniques for analyzing textual evidence and identifying patterns that can aid in solving cybercrime cases more effectively.

Keywords

Cybercrime, Case-based reasoning, Text-based algorithms, Comparative analysis, Textual evidence, Investigation methods, Profiling, Artifical Intelligence

1. Introduction

In the era of rapid digitalization, the proliferation of cybercrime has become a pressing concern for individuals, organizations, and society as a whole. Cybercriminals exploit the interconnectedness of digital systems to engage in activities such as identity theft, financial fraud, hacking, and data breaches. The complexity and sophistication of cybercrimes necessitate the development of robust investigative techniques to identify and apprehend perpetrators. In this context, the application of intelligent systems and advanced algorithms can greatly enhance the effectiveness of cybercrime investigations. CBR has consistently demonstrated its effectiveness and success in a wide range of domains. It involves utilizing past experiences, represented as cases, to solve new problems or cases. CBR's ability to leverage previous knowledge and

LWDA'23: Lernen, Wissen, Daten, Analysen. October 09–11, 2023, Marburg, Germany

krueger.hannover@t-online.de (M. Krüger)

adapt it to new scenarios makes it an ideal candidate for analyzing textual evidence in the context of cybercrime investigations. Textual evidence, such as chat logs, emails, social media posts, and online forums, often contains valuable information that can aid in identifying the modus operandi, motives, and associations of cybercriminals. This research focuses on the comparative analysis of text-based CBR algorithms for cybercrime investigations. The objective is to evaluate the performance of different algorithms and identify the most effective approach for analyzing textual evidence in the context of cybercrime. By doing so, this study aims to contribute to the development of efficient and reliable investigative techniques that can be utilized by cybercrime investigators, law enforcement agencies, and other stakeholders in the field. The primary motivation for this research stems from the challenges faced by investigators when dealing with large volumes of unstructured textual data. The sheer magnitude of information contained within digital evidence, coupled with its unorganized nature, makes manual analysis a time-consuming and error-prone process. The application of text-based CBR algorithms offers a potential solution by automating the analysis of textual evidence, thereby reducing the investigative burden and improving efficiency. To achieve the research objective, a comparative analysis of various text-based CBR algorithms will be conducted. The algorithms under consideration include traditional CBR, k-nearest neighbors (k-NN), and support vector machines (SVM) as baseline methods. These algorithms will be evaluated based on their ability to extract relevant information, classify textual data, and provide accurate results. Additionally, advanced techniques such as deep learning-based approaches and natural language processing (NLP) will be explored to enhance the performance of the algorithms. The evaluation of the algorithms will be conducted using diverse datasets specifically curated for cybercrime investigations. These datasets will comprise a wide range of textual evidence collected from real-world cybercrime cases, encompassing different types of cybercrimes and linguistic variations. The inclusion of diverse datasets ensures the robustness and generalizability of the findings, enabling the identification of algorithmic strengths and weaknesses in various scenarios. The research methodology will follow a systematic process, including data preprocessing, feature extraction, algorithm training, and performance evaluation. A comprehensive evaluation framework will be employed, considering metrics such as accuracy, efficiency, scalability, and interpretability. The experiments will be designed to provide unbiased and reliable results, enabling a fair comparison among the different algorithms. The significance of this research lies in its potential to advance the field of cybercrime investigations by providing insights into the strengths and limitations of text-based CBR algorithms. The findings will inform investigators and practitioners in selecting appropriate techniques for analyzing textual evidence, thereby facilitating the identification of patterns, associations, and crucial information that can aid in solving cybercrime cases effectively. Furthermore, the research outcomes can guide the development of intelligent systems and tools to support cybercrime investigators in their quest to combat the ever-evolving landscape of cyber threats.

2. Related work

Several studies have been conducted in the field of cybercrime investigations, focusing on the application of intelligent systems and algorithms to analyze textual evidence. These works have contributed to the understanding of the challenges and opportunities in utilizing text-based approaches for cybercrime investigations [1] [2]. In [3], the challenges of Big Data in cybercrime are highlighted.

[4] propose to undertake a classification system in the field of cybercrime as a service. This will be done using a framework for analyzing data. For this purpose, an application has been developed to analyze the information from underground forums.

In a study by Kumari et al. (2018) [5], the authors explored the use of machine learning techniques for classifying and identifying cybercrime-related text documents. They extracted the cybercrime data and according to supervised machine learning, separated the data in two labelled class. So they could get a clean training dataset.

Another relevant work by J. Nicholls, A. Kuppa and N.-A. Le-Khac (2021) [6] focused on how the application of deep learning techniques and artificial intelligence algorithms can be used to detect financial cybercrime. The authors examine various fraud methods used by criminals. In addition, relevant systems, algorithms, drawbacks, limitations, and metrics for fraud detection are presented. Likewise, an identification of the people and actors involved is provided, as well as an explanation of open and emerging issues in the field of cybercrime in the financial sector.

Another approach describes the analysis of posts on social media platforms where personal and confidential information is disseminated. In the approach, an alerting system is developed that uses convolutional neural networks and analyzes Twitter messages [7]. The results showed significant improvements in accuracy and information extraction compared to traditional methods.

In [8] an overview of relevant text-mining technologies is shown by the authors. Furthermore the authors give an overview about related work in the field of text mining research.

While these studies have made significant contributions to the field of text-based cybercrime investigations, there is still room for further exploration and improvement. The rapid advancement of machine learning and NLP techniques offers opportunities to develop more sophisticated and effective algorithms for analyzing textual evidence. Additionally, the integration of domain-specific knowledge and expertise from cybercrime investigators can enhance the performance and interpretability of text-based CBR algorithms in real-world scenarios. In conclusion, previous research has demonstrated the potential of text-based CBR algorithms in cybercrime investigations. These studies have explored various approaches, including traditional CBR, deep learning techniques, forensic linguistics, and ensemble learning, to address the challenges of analyzing textual evidence. The findings of these works lay the foundation for the current study, which aims to conduct a comparative analysis of text-based CBR algorithms in cybercrime investigations, considering factors such as accuracy, efficiency, scalability, and interpretability.

3. Methodology

This chapter outlines the methodology employed in the research for conducting a comparative analysis of text-based CBR algorithms for cybercrime investigations. The methodology encompasses data collection, preprocessing, feature extraction, algorithm selection and training, and performance evaluation.

1. Data Collection: To ensure a comprehensive evaluation of the text-based CBR algorithms, diverse datasets specifically curated for cybercrime investigations are required. These datasets consist of cybercrime-related textual evidence, such as chat logs, emails, social media posts, and online forums. The datasets should cover a wide range of cybercrime types and linguistic variations to ensure the generalizability of the findings. Ethical considerations and privacy protection measures must be adhered to when collecting and utilizing the datasets.
2. Data Preprocessing: Before feeding the data into the algorithms, preprocessing steps are necessary to clean and transform the raw text into a suitable format for analysis. Common preprocessing techniques include removing irrelevant characters and symbols, converting text to lowercase, tokenization (breaking text into individual words or tokens), removing stopwords (common words with little semantic value), and stemming or lemmatization to reduce words to their base or root form. Additionally, techniques like spell checking and normalization may be employed to enhance data quality.
3. Feature Extraction: Textual data requires the extraction of relevant features to represent the information for analysis. Feature extraction techniques aim to capture the semantic and syntactic characteristics of the text. Traditional approaches include bag-of-words representation, where each document is represented by the frequency of occurrence of words or n-grams. More advanced techniques involve the use of word embeddings, such as word2vec or GloVe, which represent words as dense vectors capturing their semantic meaning. Other features such as sentiment analysis, topic modeling, and stylometric features can also be extracted depending on the specific requirements of the investigation.
4. Algorithm Selection and Training: In this stage, various text-based CBR algorithms are considered for evaluation. Traditional CBR algorithms, such as k-nearest neighbors (k-NN) and support vector machines (SVM), can serve as baseline methods. Advanced techniques, including deep learning-based approaches and ensemble learning, can also be explored to improve algorithm performance. The algorithms are implemented using appropriate libraries or frameworks, and training is performed using the preprocessed data and extracted features. Hyperparameter tuning and cross-validation techniques are employed to optimize the models and mitigate overfitting.
5. Performance Evaluation: The performance of the text-based CBR algorithms is assessed using a robust evaluation framework. Evaluation metrics such as accuracy, precision, recall, F1 score, and area under the curve (AUC) are utilized to measure the effectiveness of the algorithms. Comparative analysis is conducted to identify the strengths and weaknesses of each algorithm based on criteria such as accuracy, efficiency, scalability, and interpretability. Statistical tests, such as t-tests or ANOVA, may be applied to assess the significance of differences between the algorithms' performance. Additionally, qualitative

analysis can be conducted to gain insights into the interpretability and explainability of the algorithms' results.

6. Ethical Considerations: Ethical considerations should be a central concern at every stage of the research, necessitating the implementation of data privacy and security measures to safeguard sensitive information. Additionally, compliance with relevant ethical guidelines and regulations, such as obtaining informed consent and ensuring anonymity of participants, must be ensured when working with human-generated textual data.

In summary, the methodology for conducting a comparative analysis of text-based CBR algorithms for cybercrime investigations involves data collection, preprocessing, feature extraction, algorithm selection and training, and performance evaluation. The systematic approach ensures the robustness, reliability, and generalizability of the research findings, enabling the identification of the most effective algorithmic approaches for analyzing textual evidence in the context of cybercrime investigations.

4. Cybercrime Attack Profiling

In this study, publicly available sources were utilized to analyze real cases. At this stage in the paper, briefly introduce the meaning of the attributes. Firstly, an analysis and evaluation of existing software used for cybercrime case management in the market was conducted to determine relevant attributes. Secondly, qualitative evaluation of textual descriptions from law enforcement agency web portals was performed to derive additional attributes.

4.1. Creation of an Appropriate Model

Digital forensics play a crucial role in the investigation of cybercrime cases, encompassing both narrower and broader aspects. It requires a high level of IT expertise and methods, especially in cases of cybercrime in the narrower sense. Law enforcement personnel must have a solid understanding of "cybercrime awareness" to identify cybercrime elements at crime scenes and take appropriate action until IT specialists are available. IT forensics, also known as digital forensics, involves the use of scientific and technological methods to analyze digital objects related to criminal events. The objective is to utilize the seized digital objects as evidence for prosecuting the perpetrators. The entire process involves on-site crime scene work and subsequent steps. Initially, the identification of victims, witnesses, and suspects takes place, aligning with the goals of the investigation. The next step entails preparing measures to document the examination of digital data carriers, which involves selecting established methods and tools. In the case of cybercrime, this process is employed for investigation purposes. Victims, witnesses, and suspects are identified based on the specific objectives of the investigation. Preparations are made to document the evaluation of data carriers in writing, employing established methods and tools to devise a suitable strategy. The management of security-related access restrictions and further processing of the evaluations of digital data carriers is crucial. Subsequently, the actual data collection takes place by gathering, consolidating, and standardizing various data formats into a unified format. This normalized data can then be

converted into a more easily evaluable format. The subsequent analysis involves evaluating events and identifying correlations. Finally, the evaluations are interpreted and presented in a suitable format. Software solutions have emerged in the market to support this process. Case management tools, also known as Law Enforcement Case Management Systems, are utilized to create and manage crime cases, while forensic software solutions aid in providing digital leads and traces as attribute values. Digital forensic tools are purpose-built for analyzing digital traces and have a narrower range of functions, focusing primarily on the search for digital evidence. These tools possess a high degree of specialization, playing a significant role within the investigative process. Most forensic tools are categorized based on specific cases. However, they may lack higher-level aspects such as suspects, witnesses, or victims, thereby performing only a partial task within the context of a cybercrime case. To effectively manage the different aspects of the investigative work, case management tools are employed. Considering this classification, several software solutions available on the market fall into the category of case management tools, including Maltego, Kaseware, goCase, Column Case, OSIRT Browser, and Matrix Investigator. In [1] it is shown how the requirements are first determined in order to then obtain the relevant data fields. The data fields are then also filled with a web crawler for reading in cybercrime cases.

4.2. Text similarity algorithms

In this study, the focus was on determining the similarity between descriptions of cybercrime cases using different text comparison algorithms. A database of 100 cybercrime cases, each containing a text description, was used for the study. From these cases, twelve were selected for consistent comparison of text similarity methods. The descriptions were divided into three length categories: short, medium, and long. The four shortest and four longest descriptions were selected, along with four descriptions from the medium length category that were around the mean of 129 words.

Various similarity measures and algorithms were used to determine text similarity. These included Hamming distance, fuzzy score, Levenshtein distance, Jaro-Winkler similarity, Jaccard distance, cosine similarity, Monge-Elkan similarity, and SoundEx. In addition, artificial neural networks such as Word2Vec, Doc2Vec, GloVe, and fastText were compared in the tests. To ensure comparability, a reference value was created by manually evaluating the similarity of the test dataset to the database. The manual assessment of similarity was based on the identification of keywords related to the modus operandi, attack target, attacker, or victim in the crime description.

In addition, artificial neural networks such as Word2Vec, Doc2Vec, GloVe and fastText were compared in the tests. Models from Apache were used for this purpose.

In addition, Naive Bayes was also investigated. Naive Bayes is a probabilistic algorithm based on Bayes' theorem. It is widely used for text classification tasks such as sentiment analysis, spam filtering, and document categorization. The classifier assumes independence between features (words) and calculates the conditional probabilities for each class based on the input features. However, due to scope issues, the results will be presented in a later paper.

4.3. Procedure for the evaluation

The database was divided into short, medium-length and long texts, of which four crime descriptions each were used for the test data set. In addition, one crime description was grammatically but not content-wise changed and also added to the test data set. In order to compare the different methods for text duplicate detection in a uniform way, 20 cases were selected from the database and used as a test data set. For this purpose, the words of the case descriptions were counted and divided into three categories: Short, Medium and Long. In the first step, the methods for similarity determination in texts were tested in a standalone application. In the further course, a JEE application was deployed so that external users on the web can also use the application. In the course of research work in the field of cyber attack profiling, an external server was set up with the open-source web server Apache Tomcat/9.0.50. Apache Tomcat is a web server and container that can run web applications in the Java programming language on a servlet basis. Apache Tomcat is open-source and the basis for many large web applications in various industries. In order to compare the different methods for text duplicate detection in a uniform way, 25 cases were selected from the database and used as a test data set. For this purpose, the words of the case descriptions were counted and divided into three categories: Short, Medium and Long. The limit for texts in the short category is 205 words or less. Texts of 500 to 700 words are categorized as medium-length. Texts in the ”Long” category have more than 1600 words. Various test cases were formed and these were compared with 25 cases from the data sets. An example of a short manual test case is given below: ”Company was hacked before there was a DDoS attack”.

4.4. Runtime efficiency

To evaluate runtime efficiency, we looked at the computation time each method took per case ID to generate a result. Table 1 shows an average of the calculation time in seconds per method for short, medium, and long case descriptions. It is called ”Inference efficiency”. In addition to training, efficiency in inference, i.e., the use of the learned word embeddings, is also important. If the inference time for processing text or for tasks such as similarity search or classification is too long, this can affect the practical applicability of the models. The following results can be noted in terms of time:

The word embedding methods had significantly higher computation times than the similarity-based methods. These times were increasingly in the range of minutes instead of milliseconds. Word2Vec thus required the highest calculation time in the entire comparison. The alogithm fastText, on the other hand, had the lowest calculation time among the word embedding methods. In order to determine an evaluation scheme, 33.33 percent quantiles are calculated for the average results of the individual text lengths as well as for the overall average results. With the calculation of the quantiles, an evaluation scheme can now be defined(see table 2).

Once the scoring scheme was generated, it was applied to the respective average results of the texts of different lengths and to the results of the overall average. The scores shown were obtained for the medium-length texts and for the overall mean (see table 3).

Method	Short [s]	Medium [s]	Long [s]
Doc2Vec	254.728	251.700	256.625
GloVe	150.425	137.878	139.945
fastText	10.452	9.322	10.678
Word2Vec	720.455	712.145	705.256
SoundEx	0.072	0.162	0.282
Monge-Elkan	0.430	1.170	3.631
Cosine	0.016	0.013	0.021
Jaccard	0.116	0.217	0.545
Jaro-Winkler	0.054	0.042	0.043
Levenshtein	0.063	0.072	0.196
Fuzzy Score	0.035	0.012	0.007
Hamming	0.110	0.051	0.123

Table 1
Calculation times of all methods per text length.

[!h]

Short [s]	Medium [s]	Long [s]	Total [s]	Evaluation Sign
$[0, 0.0152]$	$[0, 0.011]$	$[0, 0.008]$	$[0, 0.016]$	++
$(0.0152, 0.068)$	$(0.011, 0.038)$	$(0.008, 0.094)$	$(0.016, 0.075)$	+
$[0.068, 3.784)$	$[0.038, 3.872)$	$[0.094, 5.820)$	$[0.075, 4.450)$	o
$[3.784, 720.455)$	$[3.872, 712.145)$	$[5.820, 705.256)$	$[4.450, 712.619)$	-
$[720.455, \infty]$	$[712.145, \infty]$	$[705.256, \infty]$	$[712.619, \infty]$	–

Table 2
Runtime efficiency evaluation scheme

Method	Evaluation
Doc2Vec	-
GloVe	-
fastText	-
Word2Vec	–
SoundEx	+
Monge-Elkan	o
Cosine	++
Jaccard	o
Jaro-Winkler	+
Levenshtein	o
Fuzzy Score	+
Hamming	o

Table 3
Evaluation of the runtime efficiency of medium-length texts and on an overall average

4.5. Similarity Score

The time factor plays an important role in text comparison methods, but the focus is on the similarity score when comparing two texts and the accuracy of the particular algorithm is applied.

The success of textual CBR depends largely on the ability to perform similarity evaluations between texts and identify the best matching cases to extract solutions or knowledge. The word similarity criterion evaluates how well the word vectors capture the semantic similarity between pairs of words. This can be verified using a word similarity dataset where human raters evaluate the similarity between words. The evaluation of the results was based on the different scales used by the text comparison algorithms. The evaluation of the results was based on the different scales used by the text comparison algorithms.

4.5.1. Similarity Functions

The chapter discusses the use of mathematical similarity and distance measures for strings to detect the similarity of texts. A distinction is made between similarity functions, which numerically evaluate typographic mismatches between strings to indicate similarity (with normalized results between 0 and 1), and distance functions, which compute the distance between strings, with lower distances indicating higher similarity. In the present, several common measures such as cosine similarity, Jaccard similarity, Levenshtein distance, Jaro-Winkler distance, Hamming distance, and Levenshtein distance are mentioned, each having its own applications and properties. The choice of measure depends on the specific context and requirements of the analysis. Similarity metrics are vital for various applications, including natural language processing, information retrieval, and data analysis, as they enable the quantification of how alike or different objects, such as text strings, are. These metrics are categorized into edit-based, token-based, hybrid, and phonetic similarity measures, each offering distinct functions. The comparison of two strings is done character by character in edit-based similarity measures. A high similarity of the character strings with respect to the character order leads to a high overall similarity. The distance of one character string to another is determined with a so-called distance measure. With the help of edit-based similarity functions, it is not possible to evaluate longer texts, but only individual words, since several words can have a similar meaning with a different order. Edit-based algorithms include Levenshtein, Hamming, Jaro-Winkler and Fuzzy Score.

In token-based similarity functions, strings are decomposed into a set of tokens, as in sequences of related characters. The decomposition can be done using separators like punctuation or spaces or by forming n-grams. Tokens decomposed by n-grams have a uniform length n and overlap. With this method a window of the length n with the step size 1 is moved over the string. The window content is a token, which is supplemented at the beginning and at the end by a special character (for example by a single character). These additions ensure that all characters of the string occur with the same frequency in the set of tokens.

Furthermore, mixed forms of the previously discussed similarity functions exist. The token-based measures have the disadvantage of not checking the strings character by character, while the edit-based methods cannot compensate for word substitutions. Therefore, similarity computations of the token-based and edit-based functions are combined to form hybrid similarity functions.

Similarity between strings can be determined phonetically. In contrast to similarity determination using similarity measures, phonetic similarity determination does not consider the spelling of the strings, but their sounds when pronounced.

4.5.2. Word Embeddings

To analyze texts, the words they contain can be replaced by unique numbers. Afterwards, statistics about the occurrence frequency of individual words can be generated, in order to make statements about the topic of a text, for example. However, since individual words have different meanings depending on the context and the chosen numbers are context-independent, this analysis method can only be used to a limited extent. By representing word meaning as low-dimensional vectors, the context of words to other words can also be included in text analysis. These so-called word embeddings are based on the principle of distributional semantics. Accordingly, the meaning of a word can be determined from its context.

4.5.3. Implementation

A specially developed JEE web application for entering new cases was provided for the evaluation. The different methods for similarity calculation of texts have already been developed. In addition, a database with 100 cases was connected. When a new case was entered, it was automatically assigned a new case number and a case description had to be entered. In addition, a crime, an interface, an attack target, the modus operandi, and the type of contact had to be selected during case entry. Likewise, the method for similarity calculation could be selected in a combo box. Based on this selection, the system showed the user the most similar cases with the corresponding comparison value. In this display, the method for calculating the similarity could be changed dynamically and the system updated the display of the most similar cases based on this selection. To test different duplicate detection methods and compare their results, the individual case descriptions from the test data set were entered as new cases without saving them again. Each method was therefore tested 25 times. This ensured that case descriptions of different lengths were selected for testing to identify any differences in results within methods. In addition, a manual similarity assessment was performed to obtain reference values against which the results of the methods could be compared.

4.5.4. Results

Various methods are used in the area of text similarity measurement. Edit-based methods, such as Hamming distance and Levenshtein distance, focus on comparing strings, character by character. The Hamming distance is only suitable for strings of the same length, while the Levenshtein distance can handle strings of different lengths but requires significant computational resources. The Jaro-Winkler distance is another edit-based approach that takes into account character transpositions and considers prefixes and suffixes of strings. Fuzzy Score is a faster edit-based method, but does not consider the position of characters within strings. Token-based methods, such as Jaccard Similarity and Cosine Similarity, operate at the word or token level. Jaccard similarity evaluates the similarity of groups of words, ignoring word order, but penalizing changes in individual words. Cosine similarity addresses the limitations of Jaccard by converting tokens into vectors, providing a more robust measure of similarity. Hybrid methods, such as Monge-Elkan, combine elements of both edit-based and token-based approaches to improve accuracy. They leverage the strengths of both methods to achieve better results in different applications. Phonetic methods, such as SoundEx, are intended for cases where words sound

similar but are spelled differently. SoundEx takes into account the phonetic representation of words and can be useful in scenarios where typos with accents or diacritics are common. However, it only considers the initial letters of words and is language dependent. Word embeddings are an important component of NLP and machine learning. They help represent words in a way that machines can understand and work with them by capturing semantic and contextual information. Several methods have been used for word embedding. One widely used method is Word2Vec, which is known to give good results even with limited training data, especially when using the skip-gram model. However, it involves a significant computational cost and only represents individual words without considering the context of the entire document. Doc2Vec, on the other hand, considers the contextual information of entire documents and is therefore suitable for tasks that require document-level understanding. However, it still focuses on words within the corpus and does not deal with words outside the corpus. The fastText method is known for its processing speed and its ability to work in multiple languages due to its internal structure-based approach. In particular, it can represent words both inside and outside the training corpus, which makes it valuable for various NLP applications. GloVe strikes a balance by requiring less computation time while taking into account the overall context. However, it also mainly represents words inside the corpus and omits words that do not appear in the training data. In the following the evaluations of the methods with regard to the achieved results and the comparison of the manual similarity evaluation are shown. The cases that were rated worst or most dissimilar were compared to the results of the manual similarity assessment. Correctly negative results included the cases that were rated dissimilar in both the manual and methodological similarity assessments. False negatives included cases that were rated most similar in the manual assessment and rated most dissimilar in the methodological similarity assessment. Table 4 lists the cases that were rated per method and cases. The algorithms use different scales.

Method	Lowest Value	Mean Value	Highest Value	Evaluation
Doc2Vec	0.244	0.831	0.906	+
GloVe	0.047	8.419	24.567	-
fastText	0.508	0.599	0.699	+
Word2Vec	0.821	0.885	0.928	++
SoundEx	0.000	0.480	2.000	+
Monge-Elkan	0.707	0.814	0.872	+
Cosine	0.018	0.093	0.1747	+
Jaccard	0.407	0.493	0.649	+
Jaro-Winkler	0.493	0.528	0.630	o
Levenshtein	106	704.64	1829	o
Fuzzy Score	1.000	4.000	11.000	-
Hamming	133.000	744.680	1879.000	-

Table 4
Calculation of Similarity Score

5. Further research

Currently, the focus of research is on the development of automatic reference types for text comparison methods. This research area aims at making progress in automated text analysis and processing. In this context, various methods such as Rake and Textrank are being intensively tested to evaluate their efficiency and accuracy in text comparison analysis. A particularly promising research project under development is a model based on Bayes that has been successfully tested in conjunction with classification by modus operandi. These successful test results are extremely promising and will soon be presented in a scientific publication. This model has the potential to significantly improve the way we compare and analyze texts. An exciting aspect of this research project is the planned investigation of combinations of the presented text comparison methods in conjunction with the Bayes model. This opens up the possibility of exploring different constellations and configurations to determine how these methods best interact in a complementary manner. This approach promises to expand the power and applications of text comparison methods. Research in this area remains exciting and promising as it continually opens up new possibilities for innovative applications in the world of text processing and data analysis.

References

[1] M. Krüger, An approach to profiler detection of cyber attacks using case-based reasoning, in: P.Reuss, V.Eisenstadt, J.Schönborn, J.Schäfer (Eds.), Proceedings of the LWDA 2022 Workshops: FGWM, FGKD, and FGDB, LWDA, CEUR Workshop Proceedings, 2022, pp. 234–245.

[2] E. Zouave, M. Bruce, K. Colde, M. Jaitner, I. Rodhe, T. Gustafsson, Artificially intelligent cyberattacks, Swedish Defence Research Agency, FOI, Tech. Rep. FOI (2020).

[3] J. Hughes, Y. T. Chua, A. Hutchings, Too Much Data? Opportunities and Challenges of Large Datasets and Cybercrime, Springer International Publishing, Cham, 2021, pp. 191–212. URL: https://doi.org/10.1007/978-3-030-74837-1_10. doi:10.1007/978-3-030-74837-1_10.

[4] J. An, H.-W. Kim, A data analytics approach to the cybercrime underground economy, IEEE Access 6 (2018) 26636–26652. doi:10.1109/ACCESS.2018.2831667.

[5] S. Kumari, Z. Saquib, S. Pawar, Machine learning approach for text classification in cybercrime, in: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), ICCUBEA, IEEE, 2022, pp. 1–6.

[6] J. Nicholls, A. Kuppa, N.-A. Le-Khac, Financial cybercrime: A comprehensive survey of deep learning approaches to tackle the evolving financial crime landscape, IEEE Access 9 (2021) 163965–163986. doi:10.1109/ACCESS.2021.3134076.

[7] I. Ullah, C. Lane, T. S. Buda, B. Drury, M. Mellotte, H. Assem, M. G. Madden, Classification of cybercrime indicators in open social data, in: Information Management and Big Data: 7th Annual International Conference, SIMBig 2020, Lima, Peru, October 1–3, 2020, Proceedings, Springer, 2021, pp. 317–332.

[8] C. Peersman, M. Edwards, E. Williams, A. Rashid, A survey of relevant text mining technology, 2022. arXiv:2211.15784.