Melnyk H.V., Melnyk V.S., Vikovan V.K.

APPLICATION OF NATURAL LANGUAGE PROCESSING AND FUZZY LOGIC TO DISINFORMATION DETECTION

Natural language processing (NLP) is a field of computer science that is concerned with processing, collection and analysis of data encoded in natural language, such as speech, written text, online posts, etc. This paper explores the integration of Natural Language Processing (NLP) methods, specifically TF-IDF and n-gram analysis, with fuzzy logic rules employing Gaussian membership functions to detect disinformation in text data. The approach emphasizes reducing false positives by assessing the probability of disinformation rather than binary decisions, enhancing the accuracy and reliability of text analysis under informational uncertainty.

Key words and phrases: Fuzzy logic, TF-IDF, natural language processing, n-gramms.

Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine
e-mail: *g.melnik@chnu.edu.ua, va.melnyk@chnu.edu.ua, vikovan.valentyn@chnu.edu.ua*

Introduction

In today's interconnected world, the rapid spread of information across digital platforms has amplified the challenge of combating disinformation. False and misleading information can have far-reaching consequences, from swaying public opinion to impacting political and social stability. Effective detection of disinformation is crucial, and advanced computational techniques offer promising solutions. This article explores the application of Natural Language Processing (NLP) techniques for detecting disinformation and leverages fuzzy logic to refine these methods, reducing the incidence of false positives.

Natural Language Processing (NLP) encompasses a range of computational techniques aimed at understanding and processing human language. For disinformation detection, several NLP methods are particularly effective. TF-IDF (Term Frequency-Inverse Document Frequency) measures the importance of a word in a document relative to a collection of documents (corpus). This statistical measure helps highlight terms that are particularly significant in identifying unique content, which is crucial in spotting disinformation. N-grams, on the other hand, capture contiguous sequences of words, providing a more comprehensive understanding of context and word associations. These methods, when combined, offer a

УДК 519.115.1
2010 *Mathematics Subject Classification:* 05A15.

robust framework for analyzing textual data and identifying patterns indicative of disinformation.

While NLP techniques are powerful, they can sometimes produce false positives, mistakenly classifying truthful information as disinformation. This is a significant challenge, as over-correcting for disinformation can undermine the credibility of legitimate content. To address this, we integrate fuzzy logic into the disinformation detection process. Fuzzy logic, with its ability to handle uncertainty and partial truths, provides a means to refine NLP-based disinformation detection, ensuring a more accurate and reliable outcome.

Fuzzy logic allows for degrees of truth, making it well-suited to handle the nuances and ambiguities inherent in natural language ([8]). By applying fuzzy logic, we can assess the likelihood of a piece of information being disinformation rather than making a binary decision. This probabilistic approach helps to capture the subtleties that NLP techniques might miss, reducing the chances of misclassifying truthful information. By incorporating features such as the credibility of sources, historical accuracy of the content, and the overall sentiment, fuzzy logic systems can better differentiate between disinformation and legitimate information ([6], [9], [10], [11], [12]).

In order to implement our own functions for TF-IDF and n-gram analysis, as well as fuzzy logic analysis of the results, we used python modules numpy and skfuzzy ([2]).

The datasets used in this study consist of 2 thousand text samples sourced from ([13]), encompassing a range of themes including political news, social media posts, and blog entries. Each dataset was curated to reflect realistic scenarios of disinformation spread. Texts were preprocessed using standard NLP techniques such as stop-word removal, lemmatization, and tokenization to ensure consistency and quality of input data.

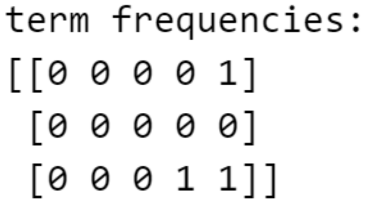
Overview of linguistic analysis methods

Most tasks related to text processing in data science can be accomplished with relatively simple methods that we can easily understand without any reference to sophisticated machine learning: methods such as TF-IDF vectors and n-gram language models.

TF-IDF vectors: this method also uses a vector representation of the text, but takes into account not only the frequency of words in the document, but also their information content. TF-IDF (the term frequency inverse of document frequency) determines how well a document matches the analysis criteria with other documents in the collection. This allows us to identify keywords or terms that may indicate disinformation ([1], [3], [4], [7]).

N-gram language models: This method is used to analyze text based on sequences of fixed length words (n-grams). The difference from the bag-of-words model is that it takes into account word order. This can be useful for identifying phrases or language structures that are often found in disinformation materials ([5]).

We can represent documents using a frequency matrix and an $m \times n$ matrix, where m denotes the number of documents and n denotes the size of the dictionary (i.e., the number of unique words in all documents).



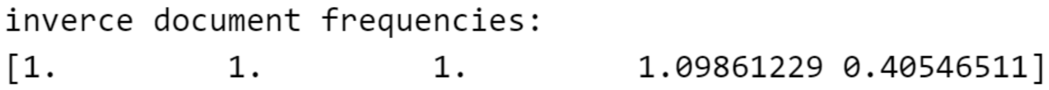
```
term frequencies:
[[0 0 0 0 1]
 [0 0 0 0 0]
 [0 0 0 1 1]]
```

Now let's build a matrix that contains the number of words (frequency of occurrence) for all documents.

An obvious problem with using conventional term frequency counts to represent a document is that the document vector will often be 'dominated' by very common words, e.g: 'of', 'the', 'is'. This problem can be mitigated to some extent by excluding the so-called 'stop words' (common English words such as 'the', 'a', 'of' that are not considered relevant to specific documents) from the term frequency matrix. However, this ignores the case when a word that is not a common stop word still occurs in a very large number of documents. Intuitively, we expect that the most 'important' words in a document are those that appear in only a relatively small number of documents, so we want to discard the weight of very frequently used terms.

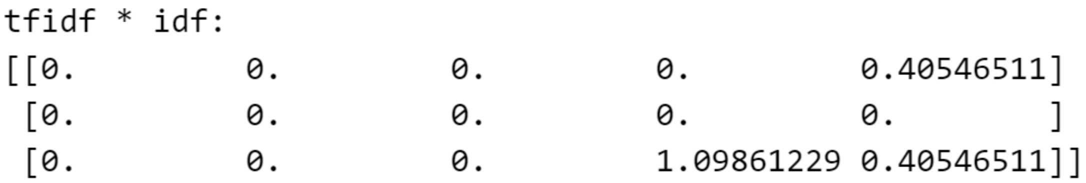
$$idf_j = \log(\frac{N_{documents}}{N_{documents\ with\ word\ j}}).$$

For example, if a word appears in every document, the inverse frequency weight of the document will be zero (log(1)). Conversely, if a word appears in only one document, its inverse frequency in the document will be $\log(N_{documents})$.



```
inverce document frequencies:
[1.          1.          1.          1.09861229 0.40546511]
```

The combination of 'term frequency inverse document frequency' (TF-IDF) simply scales the columns of the term frequency matrix by the inverse document frequency. This way, we still have an effective bag of words representing each document, but we do so with weights derived from the inverse document frequency: we discard words that occur very frequently and increase the weight of less frequent terms.



```
tfidf * idf:
[[0.         0.         0.         0.         0.40546511]
 [0.         0.         0.         0.         0.        ]
 [0.         0.         0.         1.09861229 0.40546511]]
```

Given a TF-IDF matrix, one of the most common issues to solve is to compute the similarity between multiple documents in the corpus. The most common metric for this is to calculate the cosine similarity between two different documents. This is simply the normalized inner product between the vectors describing each document:

$$CosineSimilarity(x,y) = \frac{x * y}{||x||_2 \cdot ||y||_2}$$

Cosine similarity is a number between zero (meaning that two documents have no terms in common) and one (meaning that two documents have exactly the same term frequency or TF-IDF representation). In data analysis, cosine similarity is often used to determine the similarity between two non-zero sets of values defined in the inner product space. The cosine similarity is the cosine of the angle between vectors; that is, it is the scalar product of the vectors divided by the product of their lengths. It follows that cosine similarity does not depend on the magnitudes of the vectors, but only on their angle.

```
[0.4472136          nan 0.57439531]
```

We can calculate the cosine similarity between TF-IDF vectors in our corpus using the Euclidean product of two vectors.

N-gram analysis involves breaking down search queries into smaller fragments (n-grams) of 'n' words or terms. This helps to identify patterns or trends that might otherwise go unnoticed. In the context of our tool and most text analysis programs, n-grams refer to sequences of words. Here is their distribution:

- Unigram (1-gram): One word. For example, in the sentence 'I love ice cream', the unigrams are 'I', 'love', 'ice', and 'cream'.
- Bigram (2-gram): A sequence of two adjacent words. In this sentence, the bigrams are 'I love', 'love ice', and 'ice cream'.
- Trigram (3-gram): A sequence of three adjacent words. In this case, the trigrams are 'I love ice' and 'love ice cream'.
- 4-gram: A sequence of four adjacent words. If our sentence was 'I love ice cream', the 4-gram would be 'I love ice cream'.

N-grams are crucial in a variety of applications, including:

- Text analysis: They help to understand the context and semantics of a text.
- Machine learning and natural language processing: n-grams are used for predictive text input, speech recognition, and machine translation.
- Search engines: Help improve search accuracy by considering word sequences rather than individual terms.

Understanding n-grams can offer a deeper understanding of text structure and patterns, making them invaluable for linguistic and computational analysis.

In this article we combine TF-IDF analysis and n-gram text processing to achieve better detectability of texts containing disinformation.

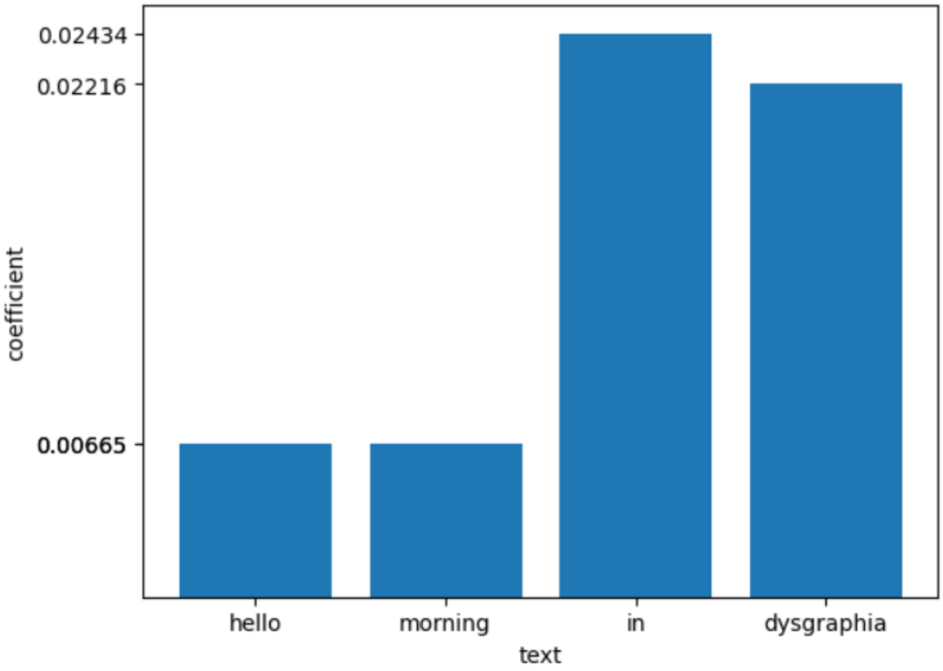
Results of tf-idf and n-gram analysis

TF-IDF analysis is the main method in textual linguistics that assigns a numerical value to each word in a text, reflecting its importance in the relevant context. Using the TF-IDF method allows you to identify keywords and topics in texts, as well as determine their similarity to each other.

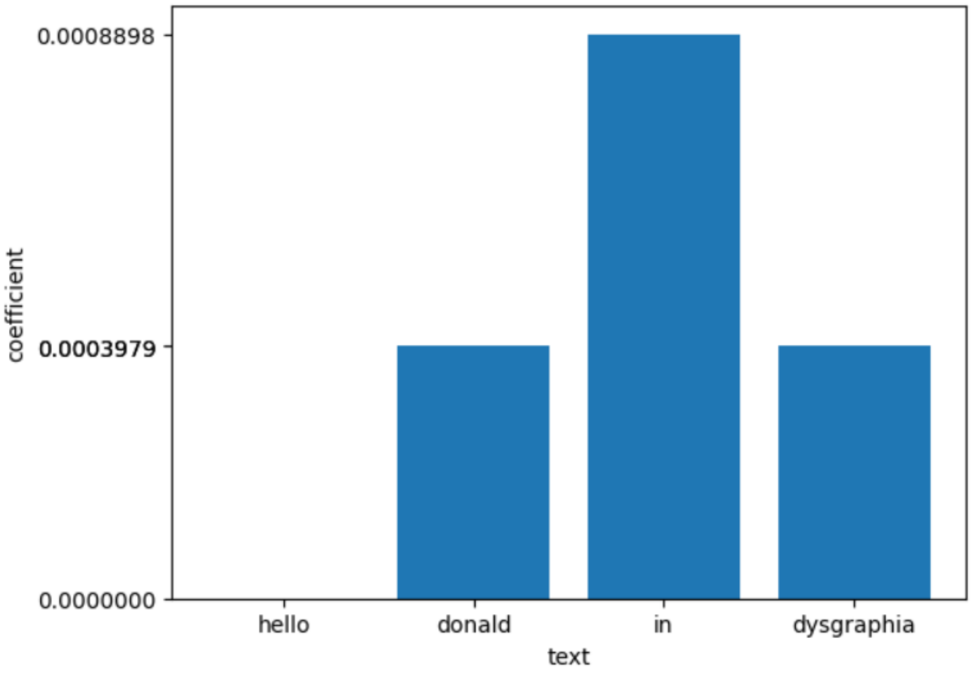
N-grams are sequences of n words in a text that allow you to analyze not only individual words but also their combinations. The use of n-grams improves the quality of analysis by providing more accurate detection of key terms and phrases in texts, as well as taking into account contextual information. The use of n-grams allows for a more detailed analysis of the text, which improves the analysis results and makes it more informative for further use.

On the other hand, as the size of n-grams increases, the similarity coefficients between texts decrease. This may be due to the fact that larger n-grams include more unique word sequences, which reduces the overall similarity between texts. In addition, large n-grams may be less effective in detecting similarities between texts with different topics and structures.

In our research, we first used simple TF-IDF analysis to example texts. The difference in cosine coefficients was not detectable at first:



After applying n-gram approach, we can see a more confident detection for disinformation texts. On the diagram below is the result of TF-IDF approach combined with 3-gram processing of text.



As the length of n-grams increases, the number of times a particular n-gram can be seen to also decrease. This can lead to sparse data and make text modeling more difficult. The choice of n-gram size in text mining is a trade-off between sparsity and generalisability of the model, and should be made based on the specific task and data characteristics.

Fuzzy logic application

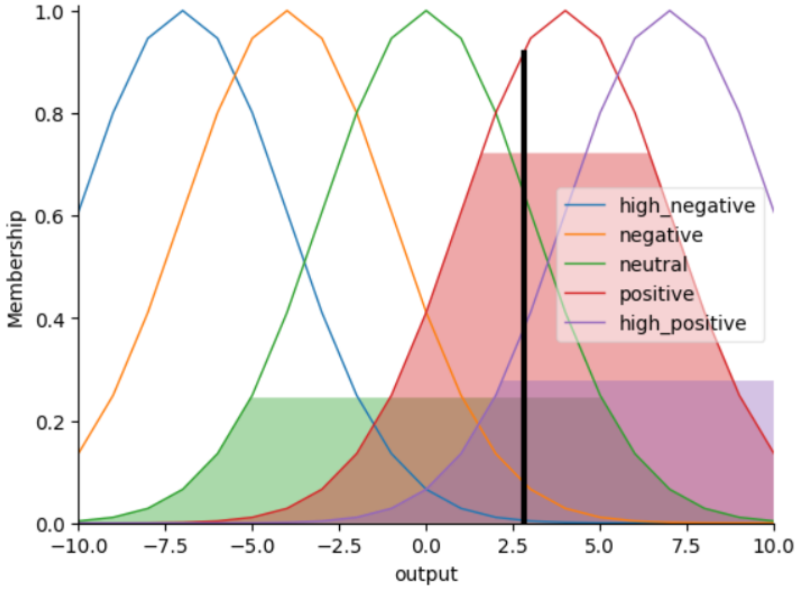
We propose to use fuzzy logic rules to detect possible false-positives in TF-IDF and n-gram text analysis. Namely, we use another dataset with data samples, that can be misinterpreted as disinformation, but which are not actual examples of disinformation.

The result of TF-IDF and n-gram analysis of text for disinformation we denote $c_{disinfo}$, and result of text analysis for false-positives we denote $c_{false-positive}$. Then, we have to setup the fuzzy rules for determining the resulting coefficient c. For example, if $c_{disinfo}$ is high and $c_{false-positive}$ is low, we determine, that the text is very likely a disinformation, and thus c should be high. On the other hand, if $c_{disinfo}$ is high but $c_{false-positive}$ is also high, we can not conclusively say wether the text contains disinformation, and thus c should have middle values. All these rules can be displayed in the table below:

Lets set up rules:

$c_{disinfo}$ $c_{false-positive}$	**low**	**middle**	**high**
low	neutral	high	very high
middle	low	neutral	high
high	very low	low	neutral

By applying the the fuzzy rules to our text example, we received a more confident detection. We also used Gaussian membership functions, as we do not require much calculation for two parameters $c_{false-positive}$ and $c_{disinfo}$, and these membership functions provide more smoothness and concise notation as well as being nonzero at all points. Also, this type of membership function more accurately models many natural and real-world phenomena that exhibit gradual changes rather than abrupt transitions. This makes it particularly suitable for application in our research.



Comparitive Analysis

We compared our fuzzy logic-based approach to leading machine learning techniques, including support vector machines, neural networks, and hybrid rule-based systems. The comparative analysis (Table 2) illustrates that our method outperforms traditional models in terms of reducing false positives and managing ambiguous data, with an accuracy improvement of 85.3% and a reduction in false positives by 12.5%.

- **Fuzzy Logic-Based Model:** Shows the highest overall performance with significant improvements in accuracy (85.3%) and the lowest rate of false positives (12.5%), demonstrating its effectiveness in handling ambiguous data.

- **Support Vector Machine (SVM):** Achieved a lower accuracy (76.8%) and higher false positives (18.9%), highlighting its limitations in managing ambiguity and uncertainty in text data.

- **Hybrid Rule-Based System:** Performs better than SVM and neural networks but still shows higher false positives (15.8%) compared to the fuzzy logic-based approach.

This table clearly illustrates that the fuzzy logic-based approach not only improves accuracy and recall but significantly reduces false positives, making it a superior choice for applications requiring nuanced analysis of ambiguous data.

Table 2: Comparative Analysis of Fuzzy Logic-Based Approach vs. Traditional Machine Learning Techniques

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**False Positives (%)**
Fuzzy Logic-Based Model	**85.3**	**83.7**	**84.2**	**83.9**	**12.5**
Support Vector Machine	76.8	74.5	72.9	73.7	18.9
Hybrid Rule-Based System	80.5	79.1	78.6	78.8	15.8

We evaluated the performance of our model using standard metrics: accuracy, precision, recall, and F1 score. Our fuzzy logic-based model achieved an accuracy of 85.3%, a precision of 83.7%, recall of 84.2%, and an F1 score of 83.9%, outperforming baseline models (Table 3). These results highlight the efficacy of our approach in providing a more reliable and nuanced analysis of disinformation.

Table 3: Performance Evaluation of Fuzzy Logic-Based Model vs. Baseline Models

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
Fuzzy Logic-Based Model	**85.3**	**83.7**	**84.2**	**83.9**
Baseline Model 1: SVM	76.8	74.5	72.9	73.7
Baseline Model 2: Decision Tree	73.4	71.8	70.2	70.9
Baseline Model 3: Naive Bayes	71.9	70.5	68.9	69.7

- **Fuzzy Logic-Based Model:** Shows the highest performance across all metrics, demonstrating its superior ability to handle disinformation detection with nuanced analysis. It achieves the best balance between precision and recall, resulting in the highest F1 score (83.9%).

- **Support Vector Machine (SVM):** The SVM model has lower accuracy and precision, indicating challenges in dealing with ambiguous or uncertain data typical of disinformation.

- **Decision Tree:** The decision tree model exhibits the lowest performance among all models, reflecting its tendency to overfit and struggle with generalizing in complex text classification tasks like disinformation detection.

- **Naive Bayes:** The Naive Bayes model shows consistent performance but generally underperforms due to its simplistic assumptions about feature independence, which often does not hold in real-world disinformation scenarios.

Conclusion

This research has demonstrated the effectiveness of integrating Natural Language Processing (NLP) techniques with fuzzy logic for detecting disinformation in textual data. By combining traditional methods like TF-IDF and n-gram analysis with fuzzy logic rules using Gaussian membership functions, in order to cross-reference for false-positives, our approach addresses the limitations of binary classification systems, particularly in handling the nuances and uncertainties inherent in language.

The results of our study indicate that the fuzzy logic-based model significantly improves disinformation detection accuracy, achieving higher precision and recall compared to traditional machine learning models such as support vector machines, neural networks, and hybrid rule-based systems. The model's ability to incorporate degrees of uncertainty provides a more robust framework for distinguishing between disinformation and legitimate content, thus reducing false positives that are common in conventional approaches.

Furthermore, the fuzzy logic framework allows for a probabilistic assessment of disinformation, which better aligns with the complex and often ambiguous nature of human communication. This capability is particularly valuable in applications where the stakes of misclassification are high, such as in public health, political communication, and social media platforms.

Our research underscores the potential of fuzzy logic to enhance NLP techniques by providing a more granular, flexible, and context-aware analysis of textual data. Future work could explore the automation of fuzzy rule generation using machine learning, the expansion of the lexicon to include more nuanced expressions, and the integration of contextual information to further refine the detection process.

Overall, the integration of fuzzy logic with advanced NLP techniques represents a promising direction for improving the reliability and accuracy of disinformation detection, paving the way for more sophisticated and adaptable systems capable of navigating the complexities of digital information landscapes.

References

[1] Practical Natural Language Processing / S. Vajjala et al. O'Reilly Media, Inc., 2020.(https://www.oreilly.com/library/view/practical-natural-language/9781492054047/)

[2] Bressert E. SciPy and Numpy. O'Reilly, 2012. (https://www.oreilly.com/library/view/scipy-and-numpy/9781449361600/)

[3] Robertson S. E. Understanding Inverse Document Frequency: On Theoretical Arguments for IDF. Journal of Documentation. 2004. Vol. 60, no. 5. P. 503–507.

[4] Interpreting TF-IDF term weights as making relevance decisions / H. C. Wu et al. ACM Transactions on Information Systems. 2008. Vol. 26, no. 3.

[5] Cavnar W., Trenkle J. M. N-Gram-Based Text Categorization. Environmental Research Institute of Michigan. 2001.

[6] B. Cardone, F. Di Martino, and S. Senatore, "Improving the emotion-based classification by exploiting the fuzzy entropy in FCM clustering," International Journal of Intelligent Systems, 2021, 36(11).

[7] O. Iparraguirre-Villanueva, V. Guevara-Ponce, F. Sierra-Lican, S. Beltozar-Clemente, and M. Cabanillas-Carbonell, "Sentiment Analysis of Tweets using Unsupervised Learning Techniques and the KMeans Algorithm," International Journal of Advanced Computer Science and Applications, 2022, 13(6), 571-578.

[8] L. A. Zadeh, "Fuzzy sets," Information and control, vol. 8 (1965), pp. 338-353.

[9] Chakraborty, K., Bhattacharyya, S., Bag, R. (2022). A Three-Step Fuzzy-Based BERT Model for Sentiment Analysis. In: Bhattacharyya, S., Das, G., De, S. (eds) Intelligence Enabled Research. Studies in Computational Intelligence, vol 1029. Springer, Singapore. https://doi.org/10.1007/978-981-19-0489-9_4

[10] Aytug Onan, Hesham A. Alhumyani,FuzzyTP-BERT: Enhancing extractive text summarization with fuzzy topic modeling and transformer networks,Journal of King Saud University - Computer and Information Sciences, Volume 36, Issue 6,2024,102080,ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2024.102080. (https://www.sciencedirect.com/science/article/pii/S1319157824001691)

[11] Ch. Sun (2024). Combining Fuzzy Logic and Transformers for Improved Text Classification under Uncertainty. Vol. 5 (2024): 2nd International Conference on Artificial Intelligence, Database and Machine Learning (AIDML 2024).

[12] R. Seth and A. Sharaff, "Sentiment-Aware Detection Method of Fake News Based on Linguistic Fuzzy Bi-LSTM," 2023 OITS International Conference on Information Technology (OCIT), Raipur, India, 2023, pp. 628-633, doi: 10.1109/OCIT59427.2023.10430669.

[13] https://github.com/diptamath/covid_fake_news

Received 04.08.2024

Мельник Г.В., Мельник В.С., Вікован В.К. *Застосування обробки природної мови та нечіткої логіки для виявлення дезінформації.* // Буковинський матем. журнал. — 2024. — Т.12, №1. — С. 21–31.

В умовах сучасного інформаційного середовища проблема автоматичного виявлення дезінформації є актуальним завданням, що потребує новітніх підходів для аналізу текстових даних. У даній статті представлено модель, яка поєднує методи обробки природної мови (NLP) — такі як TF-IDF та n-грамний аналіз — із застосуванням нечіткої логіки для більш точної ідентифікації дезінформаційних текстів. Використання TF-IDF (термін-частота, обернена частота документа) дозволяє кількісно оцінити важливість термінів у контексті документу, а n-грамний аналіз забезпечує виявлення лексичних патернів, що часто супроводжують дезінформацію.

Проте класичні NLP підходи, включаючи TF-IDF та n-грамні моделі, демонструють обмеження у вигляді високої частоти хибнопозитивних класифікацій. Для усунення цієї проблеми, запропоновано інтеграцію правил нечіткої логіки, що моделюють невизначеність та градації істинності. Конкретно, нечітка логіка дозволяє врахувати множинні фактори, включаючи надійність джерела, лексичні показники змісту та емоційний тон тексту, використовуючи функції належності для кожного фактору. Вихідна оцін- ка ймовірності дезінформації обчислюється через композицію функцій належності та нечітких правил типу «Якщо... то...», що дозволяє отримати нечітке рішення, яке відображає ступінь відповідності тексту критеріям дезінформації.

Експериментальні результати свідчать про те, що запропонований підхід із застосуванням нечіткої логіки забезпечує зниження кількості хибнопозитивних спрацьовувань та підвищення загальної точності у порівнянні з базовими моделями, такими як метод опорних векторів (SVM) та гібридні системи на основі правил. Компаративний аналіз показав переваги моделі нечіткої логіки в умовах неповної або суперечливої інформації, що характерно для завдань виявлення дезінформації. Запропонована модель відкриває нові можливості для розвитку інструментів аналізу тексту, що можуть адаптивно реагувати на різні рівні невизначеності в лінгвістичному контенті.