I. J. Computer Network and Information Security, **2025, 1, 92-127**
Published Online on February 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2025.01.08

Recognizing Fakes, Propaganda and Disinformation in Ukrainian Content based on NLP and Machine-learning Technology

Victoria Vysotska
Information Systems and Networks Department, Lviv Polytechnic National University, Lviv, 79013, Ukraine
E-mail: Victoria.A.Vysotska@lpnu.ua
ORCID iD: https://orcid.org/0000-0001-6417-3689

Krzysztof Przystupa
Department of Automation, Lublin University of Technology, Poland
Vilnius Gediminas Technical University, Sauletekio al. 11, LT-10223 Vilnius, Lithuania
E-mail: k.przystupa@pollub.pl
ORCID iD: https://orcid.org/0000-0003-4361-2763

Yurii Kulikov
Information Systems and Networks Department, Lviv Polytechnic National University, Lviv, 79013, Ukraine
E-mail: yurii.kulikov.sa.2020@lpnu.ua
ORCID iD: https://orcid.org/0009-0008-6680-7313

Sofiia Chyrun
Telecommunication Department, Lviv Polytechnic National University, Lviv, 79013, Ukraine
E-mail: sofiia.chyrun.sa.2022@lpnu.ua
ORCID iD: https://orcid.org/0000-0002-2829-0164

Yuriy Ushenko*
Department of Computer Science of the Yuriy Fedkovych Chernivtsi National University, Chernivtsi, 58012, Ukraine
E-mail: y.ushenko@chnu.edu.ua
ORCID iD: https://orcid.org/0000-0003-1767-1882
*Corresponding Author

Zhengbing Hu
School of Computer Science, Hubei University of Technology, Wuhan, China
E-mail: drzbhu@gmail.com
ORCID iD: https://orcid.org/0000-0002-6140-3351

Dmytro Uhryn
Department of Computer Science of the Yuriy Fedkovych Chernivtsi National University, Chernivtsi, 58012, Ukraine
E-mail: d.ugryn@chnu.edu.ua
ORCID iD: https://orcid.org/ 0000-0003-4858-4511

Received: 14 January 2024; Revised: 25 March 2023; Accepted: 11 May 2023; Published: 08 February 2025

Abstract: The project envisages the creation of a complex system that integrates advanced technologies of machine learning and natural language processing for media content analysis. The main goal is to provide means for quick and accurate verification of information, reduce the impact of disinformation campaigns and increase media literacy of the population. Research tasks included the development of algorithms for the analysis of textual information, the creation of a database of fakes, and the development of an interface for convenient access to analytical tools. The object of the study was the process of spreading information in the media space, and the subject was methods and means for identifying disinformation. The scientific novelty of the project consists of the development of algorithms adapted to the peculiarities of the Ukrainian language, which allows for more effective work with local content and ensures higher accuracy in identifying fake news. Also, the significance of the project is enhanced by its practical value, as the developed tools can be used by government structures, media organizations, educational institutions and the public to increase the level of

This work is open access and licensed under the Creative Commons CC BY 4.0.

information security. Thus, the development of this project is of great importance for increasing Ukraine's resilience to information threats and forming an open, transparent information society.

Index Terms: Information Security, Cybersecurity, Content, NLP, Propaganda, Disinformation, Fake News, Message, Text, Linguistic Analysis, Artificial Intelligence, Cyber Warfare, Machine Learning, Information Technology.

1. Introduction

In the digital age, information security is one of the critical challenges for many societies, especially for countries in a state of political change or conflict. As a country heavily influenced by information operations, Ukraine faces the need to combat disinformation, fake news and propaganda [1]. Accordingly, the development of tools for identifying and analysing such information threats is an urgent task that is important for ensuring the country's information security [2].

The relevance of this project cannot be overestimated in the conditions of the modern information space, where the struggle for the truth becomes almost synonymous with the preservation of national security. Information wars, in which the truth becomes the first victim, mercilessly bombard the public consciousness of millions of people, distorting reality and forming an artificial reality that serves the interests of external and internal antagonists [1].

In Ukraine, on the front lines of the confrontation with hybrid threats, the lack of reliable tools for recognizing disinformation can lead to systemic failures in public trust, erosion of fundamental democratic values and stability of state institutions. It is not simply a matter of media literacy but a matter of strategic national security defence [1-3].

The flywheel of disinformation can have a destructive effect on domestic political stability and the international reputation of Ukraine, affecting the investment climate and bilateral relations with other states. A qualitatively new level of aggression in the information field requires an adequate response in the form of developing and implementing advanced technological solutions.

The project, which aims to create comprehensive tools for identifying fakes and disinformation, not only meets the critical need of Ukrainian society for reliable means of information verification but also increases the general culture of information consumption, strengthening the information resilience of the nation. This project will become a buffer that will protect Ukrainian society from false narratives and hostile information interventions, ensuring the stable development of democratic institutions and values in the country.

*This research aim*s to create an innovative software solution – a digital shield capable of identifying and analysing flows of disinformation, propaganda, and fake news that are spread in the Ukrainian language. This tool will serve as a strategic resource for rapid verification of information, allowing users to distinguish fact from fiction at a critical moment. For this, we must follow the solutions of the tasks:

- *Creation of algorithms for automated recognition of disinformation*. Using advanced technologies of machine learning and natural language processing, develop intelligent systems that can analyse large volumes of text and detect false information with high accuracy and speed.
- *Creation of an extensive database of disinformation campaigns*. Collect and systematize a large database that includes historical and current examples of fakes identified and verified by experts. This database will be the basis for training algorithms and ensuring their relevance.
- *Development of an intuitive user interface*. Create a convenient and understandable interface accessible to a wide range of users – from scientists to ordinary citizens. This interface will allow users to easily navigate and analyse information and receive reliable reports on the reliability of content.

This project is designed not only to increase the level of information literacy among the population but also to create a powerful tool in the fight against information chaos, ensuring the stability and security of the information space of Ukraine. The basis of this study is the process of dissemination and consumption of information in global media spaces - this is a vast and multifaceted canvas on which dynamic and often contradictory information flows unfold. In the age of digital technologies, this process has become especially significant due to its influence on the formation of public opinion, political attitudes, and socio-cultural trends. The speed and volume of information dissemination create unique challenges for information verification and analysis. Our research focuses on the methods and tools used to identify, analyse and neutralize disinformation, fake news and propaganda messages in the media space. We examine modern technological approaches, such as machine learning and natural language processing algorithms, to determine their effectiveness in detecting distorted content. We also analyse how these techniques can be integrated into everyday media consumption, providing users with a powerful means to assess the veracity of the information they consume independently. This two-dimensional approach allows you to dive deeply into the mechanisms of information influence and determine strategies for developing practical tools that could resist manipulation and distortion in the media, thereby ensuring a higher level of information transparency and trust in society. This project opens new horizons in machine learning by adapting advanced algorithms to the peculiarities of the Ukrainian language. The uniqueness lies in the creation of the latest methods of deep semantic analysis, which allow us to understand the structural and contextual features of Ukrainian

vocabulary and syntax. These developments provide more accurate and efficient detection of information distortions, offering algorithmic innovations that can become the foundation for future research in natural language processing. The project has a significant practical impact, providing Ukrainian society with reliable and affordable tools for identifying and analysing disinformation. These tools allow users to identify false content and understand its sources and potential targets, thereby strengthening society's information immunity. As a result, the project strengthens Ukraine's information independence and sovereignty, increasing information transparency and trust, which is critical to the stability of democratic institutions and national security. The development of digital technologies and the growing amount of information generated and distributed on the Internet daily creates significant challenges for information security, particularly in the context of disinformation and manipulation of public consciousness. In this project, we focused on the importance and necessity of developing practical tools for identifying and analysing disinformation in the Ukrainian language, which is critically important for protecting Ukraine's national security.

2. Related Works

The problem of the ineffectiveness of existing tools for identification, analysis and response to disinformation, fake news and propaganda in the Ukrainian information space consists of several components, each requiring special attention and an individual approach [4-6].

Multilingualism and localization. Many modern tools and solutions for detecting disinformation are optimized for English, leaving a significant gap for other languages, including Ukrainian. It creates a barrier, as algorithms that work effectively with the English language may not consider linguistic features and context, which are critical for correctly analysing Ukrainian textual content. Lack of localization and adaptation of technologies can lead to misinterpretation of information or underestimation of threats.

The difficulty of detecting misinformation. Disinformation often uses sophisticated masking techniques, including mixing true and false information, making it difficult to detect with traditional fact-checking methods. Misinformation can be embedded in logically plausible narratives, which requires deeper contextual analysis and the involvement of additional sources for fact-checking.

Dynamics of information campaigns. Information campaigns, especially those involving disinformation, are rapidly changing their forms and means of distribution, which requires disinformation detection tools to be flexible and adapt to new challenges. The spread of misinformation can escalate quickly, making it urgent to implement operational responses.

Limited resources for monitoring. The large volume of information constantly generated on the Internet makes it impossible to monitor all potential sources of disinformation manually. Existing tools are often unable or ineffective at detecting misinformation in real time due to limitations in computing and human resources [7-9].

To solve the problems outlined above, our approach to creating a system includes several key areas of development:

Developing *deep learning algorithms for the Ukrainian language* is necessary to create specialized algorithms that use natural language processing technologies to recognize Ukrainian content. These algorithms must be able to analyze verbal content and understand the context, idioms, slang and stylistic features characteristic of the Ukrainian language. An important aspect will be training data that includes enough examples of fakes to train the models to recognize fake information.

- *Creating a modular media monitoring platform* that integrates various media monitoring tools and modules. This platform should be able to quickly track the spread of potentially dangerous narratives, using algorithms to determine the virality of the content and its impact on the audience. A critical function will be automatic analysis and reporting, which will help quickly respond to critical threats.
- *Integration with media platforms and social networks*. APIs or plugins must be developed to integrate our system with popular media platforms and social networks. It will ensure the possibility of early detection and blocking of disinformation at the initial stages of its spread. The integration should allow the analysis of large data streams in real-time, revealing patterns in spreading misinformation.
- *Development of educational tools for users* that help people critically evaluate information to strengthen society's information literacy. It may include developing training courses, interactive workshops, and other resources that teach users to recognize the signs of fakes and manipulative techniques.

With the help of these measures, it is possible to create an effective system capable of adequately responding to modern challenges in the information space, providing reliable protection against information threats in the Ukrainian context. Implementation is cooperation with scientific institutes, technology firms, government organizations and global partners [1]:

- Cooperation with scientific institutes is vital to establish partnerships with universities and research centres that have expertise in natural language processing and machine learning. It will provide access to the latest research and development and attract students and teachers to participate in the project. Cooperation may include joint research projects and developing courses to train personnel who will work with the system.
- Technology companies can provide the necessary software, hardware and technical expertise. They can also help integrate the system into existing media platforms and social networks. It will facilitate the implementation of solutions based on artificial intelligence for analysing large volumes of data in real-time.
- Government support is critical to providing the legal framework necessary for the implementation and operation of the system. Government agencies can also help fund the project and provide the necessary regulations so that the system can effectively interact with other national initiatives in the fight against disinformation.
- Establishing relations with international organizations and foreign partners will help share knowledge, experience, and best practices. It can also provide access to international funding and expand the potential of the system to be applied internationally by ensuring the implementation of globally recognized standards and methodologies.

These measures will help launch the project effectively and ensure its sustainable development and scaling in the future, responding to changes in the information space and technological environment. Several existing platforms and projects can be considered analogues or competitors of your product to create comprehensive software for identifying and analysing fakes, propaganda, and disinformation. Here are some of them:

NewsGuard is an innovative project focused on increasing the transparency and reliability of information on the Internet, particularly on news sites. The service works by assigning trustworthiness ratings to various media resources, allowing users to understand how credible and reliable the sources of their news content are. NewsGuard uses a team of experienced journalists who analyze and evaluate news sites according to journalistic criteria. The team reviews sites and determines whether they meet some standards, such as honesty in reporting, accountability, transparency about funding and ownership, and clarity of advertising and editorial standards. The site is assigned a trustworthiness rating based on these criteria, which becomes available to users through browser plugins or other partner platforms.

The advantages of NewsGuard are the direct involvement of experts and broad coverage. Using professional journalists to evaluate news sources adds weight and credibility to their ratings. Media experts have the necessary knowledge and experience to assess the quality of content adequately. NewsGuard analyzes thousands of sites covering a wide range of the media landscape, providing users with information about the reliability of many sources.

The disadvantages of NewsGuard are the high cost and potential subjectivity of ratings. The analysis process involving qualified journalists is resource-intensive and may require significant financial costs. It may limit the ability to rapidly expand coverage of new sites or the frequency of rating updates. Despite the standardized approach, personal preferences and judgments of journalists may influence the rating. Although NewsGuard tries to minimize subjectivity, it is impossible to eliminate it, which may raise questions about the objectivity and independence of ratings.

Overall, NewsGuard represents a significant step forward in media transparency by providing users with tools to better understand the reliability of online news sources.

Hoaxy is a platform developed by Indiana University's Center for Network Research that specializes in visualizing the dynamics of information dissemination and fact-checking in social networks. This platform will allow experienced researchers, journalists and the general public to see how to amplify specific assertions and news stories online and track the fact-checking results. Hoaxy analyzes and visualizes the distribution chains of articles and applications in social networks. Users can enter specific queries or keywords to track the spread of related information. Your platform uses algorithms to image and visualize information's paths between users and media nodes. It allows you to understand how information and current misinformation is spread through the network.

The advantages of Hoaxy are real-time analysis and visualization of the spread of disinformation. Hoaxy allows users to see how information spreads in real time, which is critical for understanding the dynamics of information campaigns. Clear visual graphics will enable you to quickly identify the primary sources and vectors of the spread of disinformation and the interaction between different actors in the network.

The disadvantages of Hoaxy are the limited ability to detect disinformation and the fact that it does not consider the linguistic features of individual languages. While Hoaxy effectively visualises and analyses how information is spread, the platform cannot always accurately identify whether the content is disinformation. It requires additional verification and critical analysis by users. Hoaxy is mainly focused on English-language content and may not consider the features of other languages, which limits its use in non-English-speaking regions or for language communities that require a specific approach to information analysis.

Overall, Hoaxy is a powerful tool for analysing the spread of information in social networks that can serve as an essential resource for researchers, journalists, and policy analysts interested in studying and combating disinformation.

Factmata is an innovative platform that uses artificial intelligence to detect and analyze malicious content and misinformation online. This system can identify fake news, propaganda, biased statements, and other manipulative content. Factmata uses sophisticated machine learning and natural language processing algorithms to analyze texts. The

platform analyzes large volumes of information from various sources, including news, social networks, and blogs. It allows the detection of potentially harmful content in its early distribution stages. Factmata also provides feedback from users and experts, which helps to constantly improve and adjust the platform's algorithms.

The advantages of Factmata are an automated approach and continuous learning. Artificial intelligence allows you to automate detecting malicious content, reducing the need for manual verification and analysis of large volumes of data. Factmata is constantly learning from new data, which improves its ability to detect various forms of misinformation and adapt to changing methods of information manipulation.

The disadvantages of Factmata are high learning resources and accuracy and Errors. Developing and maintaining the algorithms used by Factmata requires significant computing and financial resources. Training artificial intelligence on large data sets can be costly and complex. Like any AI-based system, Factmata can have accuracy issues, especially regarding content that contains subjective judgments, irony, or jargon that could mislead the system.

Overall, Factmata represents a significant step forward in developing anti-disinformation tools, offering a robust platform that can help various organizations, from news agencies to corporate clients, control the quality and reliability of information.

Snopes is one of the first and most recognizable fact-checking platforms on the Internet. Founded in 1994, Snopes specializes in debunking urban legends, popular myths, famous stories, and misinformation spread across the Internet and social media. The Snopes platform uses a research approach involving a team of experienced fact-checkers and journalists who analyse and verify the reliability of information. They investigate each request in-depth based on primary data sources, scientific studies, official statements and other verifiable facts. Snopes is also known for its detailed explanations and conclusions that help users understand the context and nature of the information.

The advantages of Snopes are strong user trust and authority. With its long history and consistency in fact-checking, Snopes has earned a reputation as a reliable source for distinguishing true from false information. Their work has earned them trust and recognition among users around the world.

The disadvantage of Snopes is its dependence on manual fact-checking. The main drawback of Snopes is that the platform is highly dependent on the manual labour of their fact-checkers. It can slow down the verification process, especially in today's news cycle, where information changes quickly. This approach may also prove insufficiently effective in situations where it is necessary to respond quickly to a large amount of fake news on social media.

Snopes continues to be a valuable resource in the fight against disinformation, providing users with a tool to verify the authenticity of the content they consume. However, the growing need for faster and more automated fact-checking may require Snopes and similar platforms to adapt to new technological solutions.

Bellingcat is an independent international initiative specializing in open-source intelligence (OSINT) investigations. Founded in 2014, Bellingcat uses data from open sources, such as social media, satellite images, videos, photos, and even advertising databases, to investigate and verify information often related to international conflicts, geopolitical crises, and rights violations of a person. Bellingcat uses modern data analysis techniques, including geolocation analysis, image and video analysis, and social media research, to uncover and document facts often overlooked by traditional media. It allows Bellingcat to conduct investigations with high independence and objectivity.

The advantages of Bellingcat are its high accuracy and depth of analysis as well as its advanced data collection and analysis techniques. Thanks to an interdisciplinary approach and the use of various sources of information, Bellingcat can conduct in-depth analytical investigations. They are known for their ability to reveal complex networks of relationships and events that take place behind closed doors. Leveraging advanced technology and analysis techniques, Bellingcat leverages large volumes of open data to uncover insights often unavailable through traditional sources.

The disadvantages of Bellingcat are that it focuses primarily on international events and needs specialized knowledge to interpret data. Because Bellingcat often focuses on global events and international conflicts, its activities may be less relevant to local or regional issues that require specific knowledge and resources. Bellingcat's work requires high expertise in data analysis, satellite mapping, and video and image verification. It creates barriers to attracting new team members and requires constant professional development.

Bellingcat remains one of the leading open data investigative organizations, making significant contributions to developing investigative journalism and data analysis.

3. Material and Methods

For an analytical review of the literature and related works on the identification of disinformation, it is essential to research the various methods and tools used in this area. Scientific works often focus on machine learning technologies, particularly deep learning methods for analysing textual data [10-12]. For example, algorithms based on natural language processing (NLP) models can detect patterns and inconsistencies in text that may indicate misinformation [12-15]. These techniques include using transformers such as BERT or GPT, which can generate and understand text at a very high level. However, beyond the technical aspects, many studies emphasize the importance of contextual analysis. Detecting disinformation often requires not only text analysis but also the inclusion of contextual information, such as determining the source of the information and its reliability [16-18]. The work shows that integrating data from different sources and

their cross-analysis can significantly improve the accuracy of detecting fake news.

Another aspect often discussed in scientific works is the development of algorithms for automated detection of changes in the behaviour of social network users, which can be an indicator of manipulative information campaigns. Studying information dissemination patterns allows for identifying potentially harmful content before it reaches a large audience. Given the dynamics of modern media, there is a growing need to develop more effective tools for real-time data monitoring and analysis [18-21]. Many works propose using complex systems that integrate artificial intelligence, machine learning and automated monitoring tools to strengthen the ability to quickly respond to information threats [21-24]. Finally, it is essential to note that no tool or method can be completely efficacious without considering the ethical aspects and possible consequences of implementing such technologies. A critical review of current research shows that continued development in algorithmic transparency, privacy and data protection is vital to maintaining trust and acceptance among users. The development of our software for the identification and analysis of disinformation offers significant advantages, in particular, the provision of a high level of customization, which allows you to precisely adjust the system to the specific needs of the Ukrainian media space [1]. Given the linguistic and cultural differences often ignored by one-size-fits-all solutions, it is essential. Automation of analysis processes allows the processing of large volumes of data, reducing dependence on the human factor and increasing the speed of response to information threats.

On the other hand, developing your system requires significant initial investment in research and development and ongoing costs for software maintenance and updates. Because such a system depends on skilled professionals for its growth and maintenance, there is a risk if access to skilled labour or technological resources is limited [24-28]. In addition, the constant challenge of ensuring security and data protection can be a significant burden for a start-up [29-33].

Goal tree for the fake, propaganda and disinformation identification project:

Data collection and preparation (Fig. 1-2) are as data loading (automating the process of collecting texts through the API of social media and news platforms) and data cleaning (removal of non-informative parts of the text (advertising, spam), standardization of data formats).

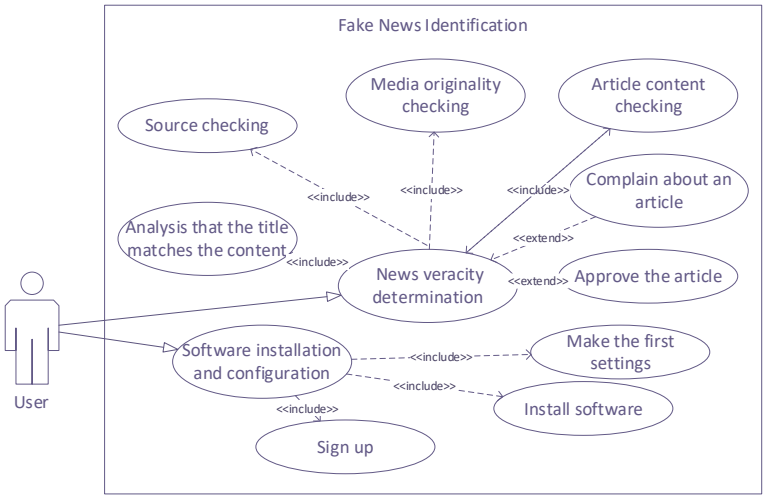


Fig.1. The use case diagram for the system of recognising propaganda, fake news and disinformation

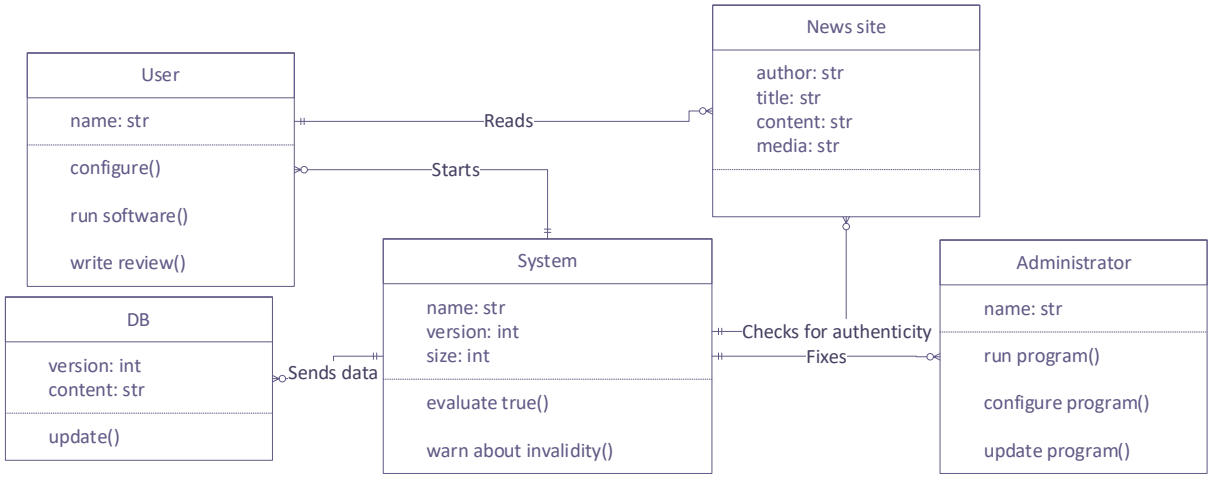


Fig.2. The class diagram for the system of recognising propaganda, fake news and disinformation

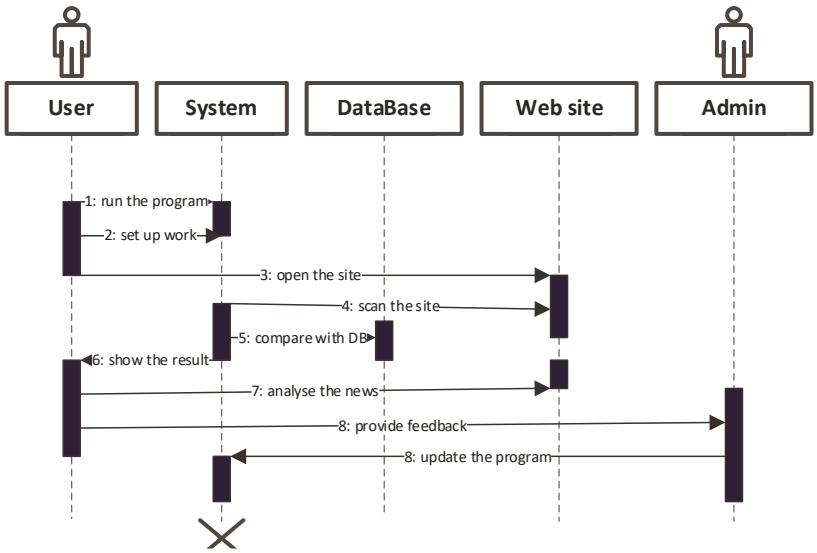


Fig.3. The flow diagram for the system of recognising propaganda, fake news and disinformation

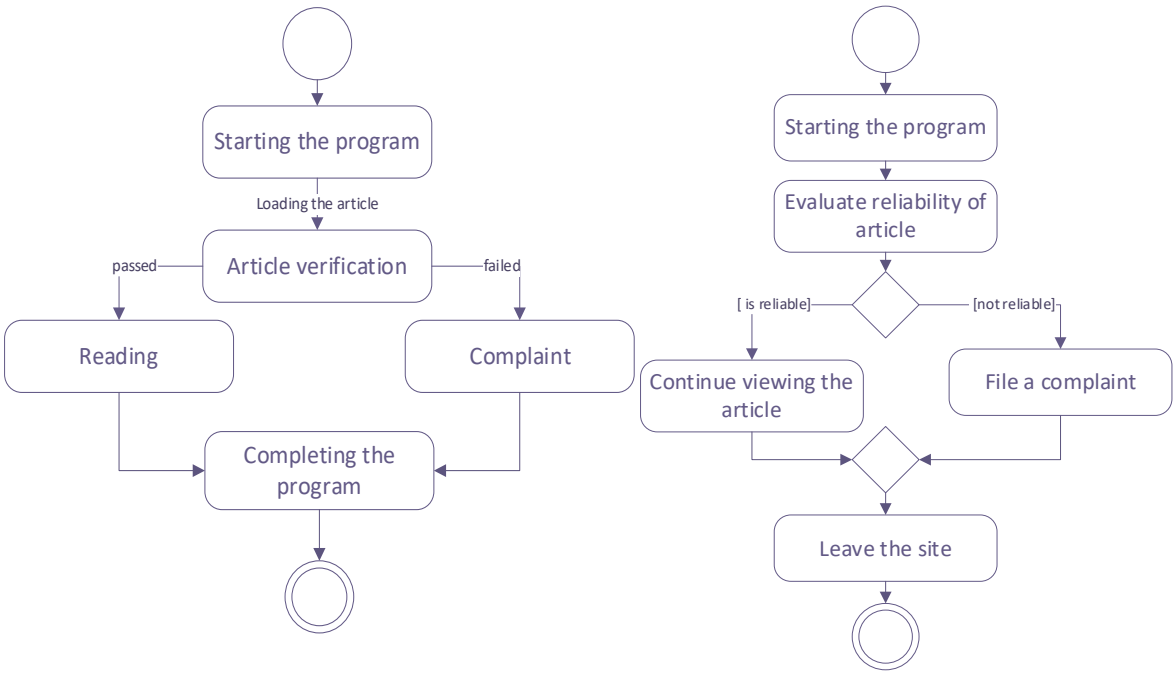


Fig.4. State and activity diagrams for the system of recognising propaganda, fake news and disinformation

The development of models to detect misinformation (Fig. 3-4) is as machine learning for text classification (developing models that can classify text as "true" or "false," based on text characteristics and context) and deep sentiment analysis (development of specialized NLP models to analyse the emotional colouring of text and identify language markers often associated with manipulation and propaganda).

The development of fact-checking algorithms is as knowledge Base Building (creating and maintaining a fact-checked knowledge base that can be accessed to automatically compare statements from text) and development of verification algorithms (using combinations of NLP and artificial intelligence algorithms to analyse mentions of facts and automatically verify them in real-time).

Model optimization is tuning hyperparameters (such as Grid Search and Random Search to determine optimal model parameters) and fine-tuning models (micro-tuning models to specific types of misinformation that change over time).

System testing and validation are manual testing with experts (using media and information security experts to validate model results) and error analysis (detailed examination of model errors to improve their accuracy, including analysis of false positives and false negatives).

Identifying fakes, propaganda, and disinformation involves input data, output data, mechanisms, and management. Input data is text data sets that potentially contain fakes, propaganda disinformation and a list of known reliable and questionable sources of information. Output data is information reliability assessment reports and confirmed or denied disinformation cases.

Mechanisms:

- choosing a model type is deciding whether to use certain types of analytical models (e.g., classification models, sentiment analysis models);
- tuning hyperparameters is fine-tuning model parameters to increase their efficiency and adapt to the specific requirements of a given task;
- Fact-checking algorithms use specialized tools and methods for information verification.

Management:

- quality assessment methods are validation of information accuracy through comparison with a database of verified data and application of machine learning methods;
- accuracy and reliability requirements set high accuracy standards for the identification system to ensure its effectiveness in real-world conditions.

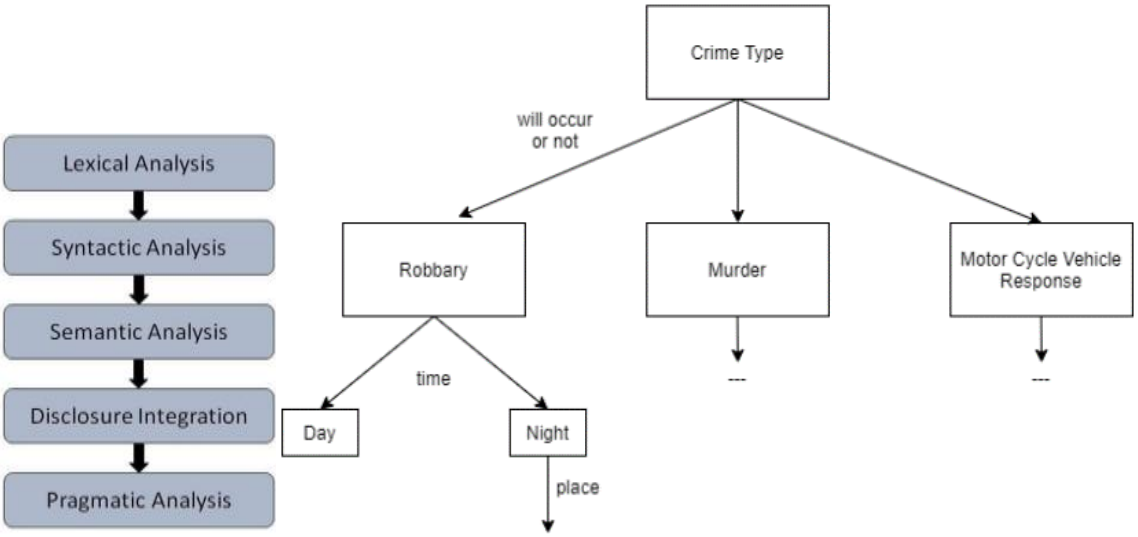


Fig.5. NLP techniques application sequence and using the solution tree to classify crime types

Table 1. Fulfilment of requirements by the linguistic system for identification of fakes/propaganda or analysis of criminal reports

Class	Non-functional	Functional	Business	User
Fulfilment of requirements in the developed information technology	- The functionality does not require the user to have specialized skills. - Accuracy of the neural network should be greater than 92%.	- Semantic analysis of any textual content/reports. - Semantic analysis of content from text files.	- Reduce the time needed to analyse arbitrary text/reports. - Provide the user truthful information about emotions/keywords in the text/reports. - Provide developers with quick access to ready-made models designed for content analysis.	- System authorization. - Text interface. - Interface for working with files. - Results display interface.
Requirements content example	1. Users are divided into roles to which part of the functionality is attached. 2. System quality attributes: efficiency, reliability, convenience. 3. Requirement for interfaces: users without specialized knowledge must be able to use them.	1. Executes processes of user authorization; analyses the entered text, text from files, and text from websites of natural language newspapers/reports. 2. Textual content downloading, text input, parsing of web sites according to the selected markup element. 3. A single flow carries out all actions.	1. Product based on semantic analysis of texts, websites of newspapers or criminal reports. Relevant to businesses that conduct field marketing surveys. 2. Works with typed text, text from text files and specified websites of newspapers. 3. Provided an API for working with neural networks for external systems. 4. Effects: time, economic, technological.	1. Visitor, user with subscription, marketer. 2. Semantic analysis of text and automated analysis of a sample of newspapers/reports. 3. Effects: time 4. the user must register and subscribe to the service to use all functions. 5. API access must be confirmed for developers.

The following features of software solutions that work in the field of NLP (Fig. 5a) are highlighted:

- Ability to use as an API;
- Possibility of use for users who do not know the NLP area (for example, to classify crime types as in Fig. 5b);
- Working with text involves additional functionality (for example, to criminal reports analysis as in Table 1);
- Several methods of working with the text are provided;
- Running the service in the cloud (Fig. 6);
- Use of private datasets (for example, to classify crime types as in Fig. 7).

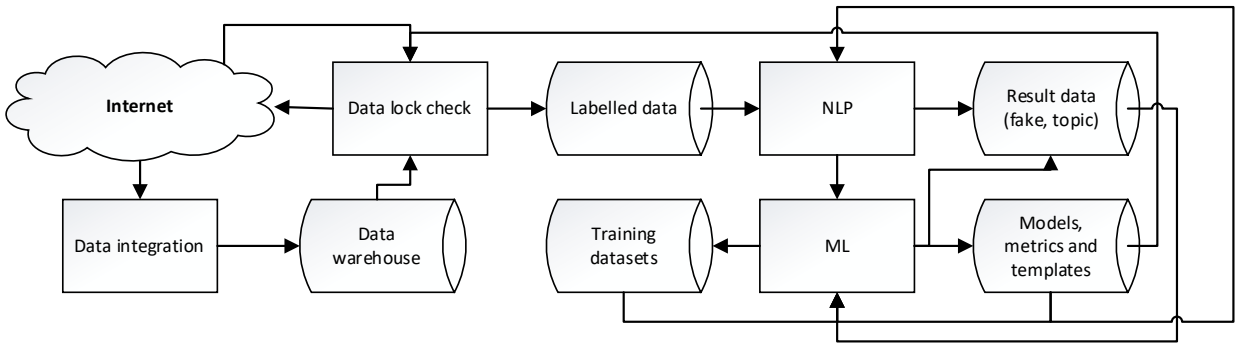


Fig.6. Running the service of the system in the cloud

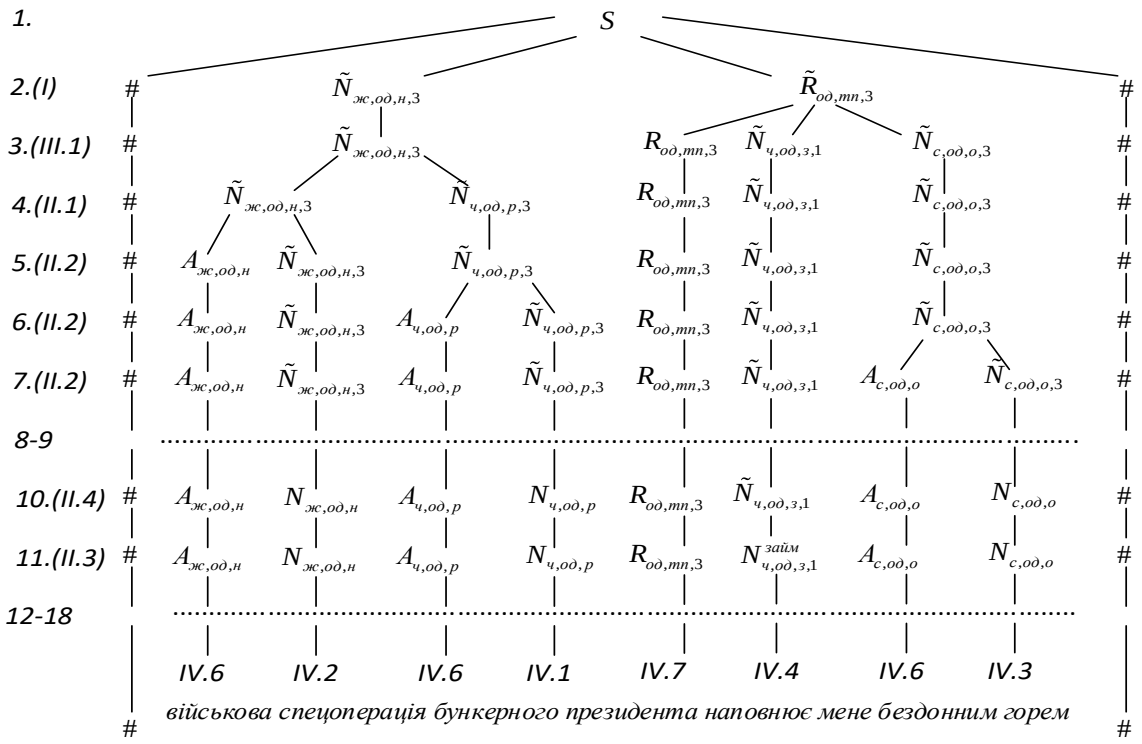


Fig.7. Work results on private test dataset in the system of recognising propaganda, fake news and disinformation

The beginning is the start of the Workflow process (Fig. 8-9).

- Download datasets:
  - – From a dataset of news articles and social media for analysis.
  - – With test sets from proven sources to evaluate the model.
- Text cleaning and normalization - preparation of texts for further processing:
  - – Remove non-informative elements (advertising, spam).
  - – Normalization of language expressions and format standardization.
- Choosing an algorithm for disinformation analysis - defining methods for detection and analysis:

– Text classification based on machine learning.
– Sentiment analysis to identify emotional colouring.

- Tuning model hyperparameters - tuning the model for optimal performance:

  – Selection of parameters that provide the best recognition of disinformation.
  – Adapt the model to specific types of disinformation.

- Model testing and validation - checking the quality of the trained model:

  – Using validation techniques such as cross-validation.
  – Evaluation of the effectiveness of the model in detecting misinformation.

- Integration of the model into information systems - use of the model in systems for monitoring the media space and information portals.

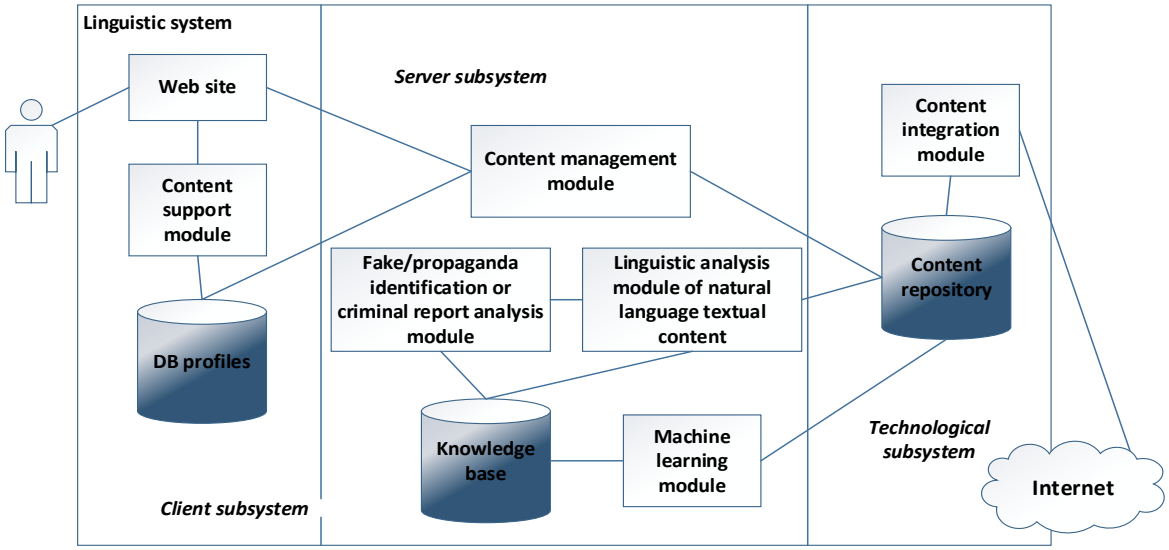


Fig.8. Architecture of a linguistic system for identification of fakes/propaganda or analysis of criminal reports

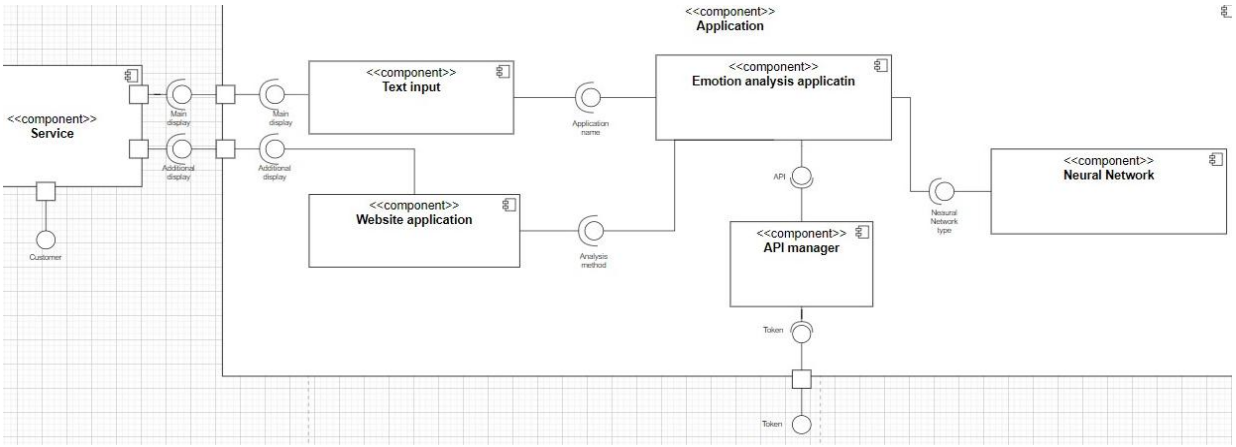


Fig.9. Components diagram for the system of recognising propaganda, fake news and disinformation

Intellectual component for the process of recognising propaganda, fake news and disinformation. Development of algorithms for analysing texts to detect disinformation and unreliable information based on stylistic and contextual features [24-28].

- Input data is news texts from various media sources, social networks and information sites.
- Output data is scores of credibility of the news, determined based on analytical models, which classify the text as fake, accurate, or questionable.
- Data submission form clean, normalized, annotated or unannotated texts used for training and testing models.
- Business processes [24-28]:

– Data collection and processing are automated collections of texts from news resources and social media and further cleaning and normalization of texts in preparation for analysis.

– Model training and testing is developing and optimising machine learning models for classifying texts according to their reliability.

– Implementing the model into practical applications is the integration of models into media monitoring systems and content analysis platforms, where models can automatically recognize fake news and disinformation.

- Architecture for disinformation analysis [24-28]:

– Machine Learning based on SVM, Naive Bayes classifier, and deep neural networks.

– Natural language processing is based on NLP algorithms to analyse text data, such as BERT, and detect semantic anomalies and stylistic deviations often associated with fakes.

– Sentiment Analysis to detect emotional colouring that may indicate propagandistic or manipulative text materials.

- Evaluation of the quality of models:

– The confusion matrix uses the error matrix to evaluate the model's accuracy, recovery, and F1 score.

– Conducting A/B testing with real users to evaluate the practical effectiveness of algorithms in actual conditions.

- Place of application:

– Automate news and social media monitoring systems are used to identify potentially falsified information.

– Use educational programs to teach students to recognize fakes and think critically.

- Expected effects:

– Awareness raising promotes education and public awareness of the risks of misinformation and fakes.

– Protection of the information space: Improving the protection by timely detection and neutralising disinformation.

– Enriching scientific research is developing scientific research in information security and machine learning.

Training the embedding model. The following Python libraries can be used to analyse and identify disinformation:

- Scikit-learn is a well-known machine learning library that supports a variety of classification algorithms such as logistic regression, naive Bayes, and SVM. These algorithms can be used to develop classifiers that distinguish authentic news from fake news.
- NLTK (Natural Language Toolkit) is a feature-rich library for text processing, including sentiment analysis and tokenization tools that are useful in analysing text data for manipulative language patterns. NLTK has essential support for the Ukrainian language with means for tokenization and partial language tagging but may require additional resources for more complex tasks such as parsing.
- SpaCy is a high-performance natural language processing library containing pre-trained sentiment analysis and entity detection models. SpaCy can be used for advanced study of phrase structure and context, which is critical when developing algorithms to detect disinformation. SpaCy supports the Ukrainian language with the help of external packages. For example, community-developed models for the Ukrainian language can be integrated with SpaCy for NLP tasks.
- Transformers (Hugging Face) - A library that provides access to pre-trained models based on transformers such as BERT, GPT, and RoBERTa. These models can be used for deep contextual analysis of texts, which allows the detection of complex cases of misinformation and manipulation. This library supports the Ukrainian language thanks to multilingual models such as BERT and GPT, which are already trained on data from different languages, including Ukrainian. For specific tasks, these models can be adjusted on the corpora of the Ukrainian language.
- TensorFlow and PyTorch are advanced deep learning frameworks that allow you to design and configure complex neural networks for disinformation identification tasks. They are handy for creating customized solutions that consider data specificity and identification accuracy requirements.

In sequence-to-sequence problems, such as neural machine translation, initial proposals for solutions were based on the use of RNN in the encoder-decoder architecture. These architectures had significant limitations when dealing with long sequences. Their ability to retain information from the first elements was lost when new elements were incorporated into the sequence. In an encoder, the hidden state at each step is associated with a specific word in the input sentence, usually one of the last. Therefore, if the decoder only accesses the previous hidden state of the decoder, it will lose the relevant information about the first elements of the sequence. For example, in the sentence "There are dolphins in the …", the next word is obviously "sea" since there is a connection with the word "dolphins". If the distance between the dolphins and the intended word is short, RNN can easily predict it. Consider another example: "I grew up in Ukraine with my

parents, spent many years there, and know their culture well. That's why I speak fluently … ." The expected word is "German", which is directly related to "Ukraine". However, in this case, the distance between "Ukraine" and the intended word is more significant, making it more difficult for RNN to predict the correct word.

Consequently, as distance increases, RNNs cannot find connections due to low memory power. So, to eliminate RNN's problems, a new concept was introduced – the attention mechanism. Instead of paying attention to the last state of the encoder, as is usually done with RNN, at each step of the decoder, we view all the states of the encoder, being able to access information about all the elements of the input sequence. It is what "attention" does. It extracts information from the entire sequence, the weighted sum of all the past states of the encoder. It allows the decoder to assign more weight or importance to each output element's occurrence element. However, this approach has a significant limitation: a component must process each sequence. The encoder and the decoder must wait until the t-1 steps are complete to process the step. Therefore, large volumes of occurrences take a long time.

Transformers are a neural network, the architecture aimed at solving sequential tasks while efficiently processing long-term dependencies. The Transformers model extracts characteristics for each word using a self-attention mechanism to determine how important all the other words in the sentence are about the previous words. No duplicate units are used to get these features; they're just weighted sums and activations to be parallelized and efficient.

When choosing the BERT model, we took into account several key factors.

Firstly, BERT can effectively model the context and dependencies between words in the text thanks to the self-attention mechanism used in Transformers. It allows BERT to identify semantic connections and understand the meaning of the text at a deeper level.

Secondly, BERT is a pre-trained model, meaning it has been pre-trained on a large amount of textual data. It allows the model to learn common dependencies and language properties, resulting in better performance when refining for specific tasks.

One of the critical factors that led to the choice of transformers to work with BERT was their ability to effectively model context in text. BERT uses transformer architecture to account for the broad context and interactions between words in a text. It allows BERT to identify semantic dependencies and understand the meaning of the text at a deeper level. Therefore, the choice of transformers for BERT was due to their power in context modelling and the ability to work with large amounts of text data.

Torch (and its PyTorch shell) is one of the most popular packages for developing neural networks and implementing optimization algorithms. Torch is a high-performance library optimized for working with neural networks. It uses accelerators such as graphics processing units (GPUs) to quickly compute and train models. It provides flexible tools for developing neural networks. It makes it easy to create and configure complex architectures, such as transformers, and it has many built-in features for working with text data. Torch can easily integrate with other popular libraries, such as NumPy and Pandas, allowing for convenient data processing and analysis before using it in neural networks.

Hugging Face, Inc. is an American company that develops tools for creating applications using machine learning. She is best known for creating the Transformers library for natural language processing and a platform that allows users to share machine learning models and datasets. Hugging Face provides a wide variety of pre-made models, including BERTs, that can be used directly. It significantly reduces the time needed to develop and train the model. It provides a simple interface for downloading, configuring, and using models. Hugging Face offers additional tools and libraries for working with neural networks, including various modules for processing text data, evaluating models, and conveniently visualizing results. Hugging Face has a large and active community of developers and researchers, which contributes to the constant updating and improvement of the platform. Many resources are also available to help with questions and problems that arise when working with BERT models.

The Jupyter project is an interactive computing platform that works with all programming languages through free software and open standards. This platform, which supports over 40 programming languages, is viral for developing Python, R, Julia, and Scala code. The main element of Jupyter is "notebooks" - a file format that contains code and its executed results to support machine learning and data analysis projects.

Critical Benefits of Jupyter Notebook:

- code editing in the browser, with automatic syntax highlighting;
- the ability to execute code from the browser with the results of the calculations generated by the code;
- displaying the result of the calculation using multimedia formats such as HTML, LaTeX, PNG, SVG, etc.;
- editing rich text in the browser using the Markdown markup language, which allows you to add comments to the code without being limited to plain text;
- the ability to easily include mathematical notation in commenting cells using LaTeX and be rendered by MathJax.

Statistical evaluation of embedding models. A library specially developed for these purposes can be used to evaluate models of misinformation identification. Like genism evaluations, it provides methods for assessing models but focuses on analysing texts and determining their reliability. Currently, there is no standard, generally accepted library in Python that specializes exclusively in model evaluation for misinformation identification with the same functionality as genism-evaluations for word embeddings. However, for the development and testing of disinformation detection systems, libraries such as:

- Scikit-learn for machine learning,
- NLTK and SpaCy for natural language processing,
- Hugging Face's Transformers for working with advanced transformer-based models that can be adapted for specific disinformation detection tasks.
- A library for evaluating disinformation detection algorithms implements methods designed to assess the performance of models in detecting fake news and propaganda, particularly in data-limited language resources. It allows users to automatically generate specialized test sets for any supported languages, using resources such as Wikidata to create sets of articles or texts often associated with disinformation.
- Special test sets. Each test set may contain articles or text messages classified (e.g., politics and health) that have already been identified as targets of disinformation. Users can select items from Wikidata or other databases to form these categories.

Similar to Topk and OddOneOut, evaluation functions in the library may include:

- Precision@K estimates the accuracy of disinformation identification by counting the number of times fake news was correctly identified among the first *K* results.
- Recall Rate is a score that measures the ability of a system to detect all possible instances of misinformation in a dataset.

The evaluation results can be represented as a tuple of five values that include:

- general accuracy of the model;
- accuracy by category;
- a list of categories where the model showed the worst results;
- total score for all tests (number of correctly identified cases);
- category score (number of correctly identified cases in each category).

Vector database. To create a system that identifies disinformation, you can consider using a vector database to store articles, news and other textual records for further analysis of fakes. A vector database for such a system could include the following components [24-28]:

- Embedded vector represents text derived from advanced NLP models such as BERT or GPT to preserve deep contextual knowledge about the text.
- The full text of an article or news story analysed for misinformation.
- Additional metadata is information about the source, author, date of publication, and any other metadata that can help verify the authenticity of the information.

When a user receives a query, the query text is transformed into a vector using the embedding model, after which a database search is performed to find the most similar vectors and corresponding texts that can answer the query or refute misinformation.

Web interface. Streamlit was chosen to develop the web interface for several reasons:

- Speed of development. Streamlit lets you quickly create interactive web interfaces with minimal code, which is ideal for displaying research results and real-time models.
- Ease of use and adoption among data analysts. Streamlit's interface is easy to use and configure, allowing non-web developers to integrate their models quickly.
- Integration with Python. Since most embedding models are developed in Python, Streamlit provides seamless integration with the Python ecosystem, simplifying interaction with the developed models.
- Community and ecosystem. Streamlit has a growing community and ecosystem of extensions and custom components that can enhance its functionality. You can find various resources and support from the Streamlit community.

4. Experiments, Results and Discussion

Data collection for the project. You need to choose the correct data for the neural network to work efficiently and train it. Since my topic is related to the search for fake news about the Russian-Ukrainian war, this is exactly what was needed, along with true news. However, we faced the problem of no ready-made datasets on this topic. Accordingly, it was necessary to collect data independently from the beginning. As a data source, we chose various Ukrainian and Russian news sites. It allowed us to get a variety of data on the russian-Ukrainian war. An additional analysis of each news item was carried out during the data collection. We checked the source, the date of publication, and the author of the news and looked for whether this news was mentioned anywhere else. Since we manually entered everything from the beginning, we immediately marked this news in the Label column, where it marked 0 for true news and 1 for fake news. Later, based

on these labels, the neural network is learned and trained. The development and content of a dataset for training and testing a machine-learning model is based on the following steps:

Step 1. Research the information base on web resources for the period after the full-scale invasion of Ukraine on February 24, 2022.

Step 2. Form a database of social media chats to identify disinformation, namely, fakes, propaganda and manipulation.

Step 3. Form a database of fact-checking resources to trace the refutation of fake information.

Step 4. A selection of the central narratives of hostile language is needed for the reliability of information sampling and better coverage.

Step 5. Search for at least two posts for each fake: true-false and fake-true pairs.

Step 6. Search for a proportional number of posts in both Ukrainian and Russian, true and false, for more excellent sampling reliability.

Step 7. Formation of criteria for adding fakes to the dataset table.

Step 8. Filling the dataset with the necessary information.

The paper has developed and filled a dataset of disinformation, taking into account the above steps.

Following these steps allows you to solve the following research tasks:

- identify the characteristics of primary sources and distribution routes, as well as the characteristics of inauthentic behaviour of chat users;
- analyze the statistical characteristics of the indicators of the intensity of the development of information threats;
- describe the existing sources of detecting and refuting fakes, propaganda and disinformation;
- analyze the existing methods for determining the characteristics of primary sources and distribution routes, as well as the attributes of inauthentic behaviour of chat users;
- determine the method of developing and filling disinformation datasets;
- to carry out experimental studies based on dataset data to find criteria and parameters for the distribution and change of the dynamics of participants' behaviour.

For the study, a dataset (Fig. 10) was collected, which consists of various news and posts on social networks, such as Facebook, Telegram, etc. (true-false). Narratives based on analysing open web resources in the context of a full-scale invasion are studied. The data with fakes and sources of their refutation were investigated to further fill the dataset and identify the primary sources of disinformation. The developed dataset demonstrates:

- variability in time of posts, i.e. that posts can be both edited and deleted; For example, this post in the picture no longer exists;
- the possibility of self-propagation for different purposes;
- the presence of reposts and the possibility of writing both fake and truthful information in the same language, in this case in Ukrainian (also paragraph 1 and paragraph 2 in Fig. 10 in Russian, the language of the aggressor);
- Not being able to view some posts due to privacy settings.

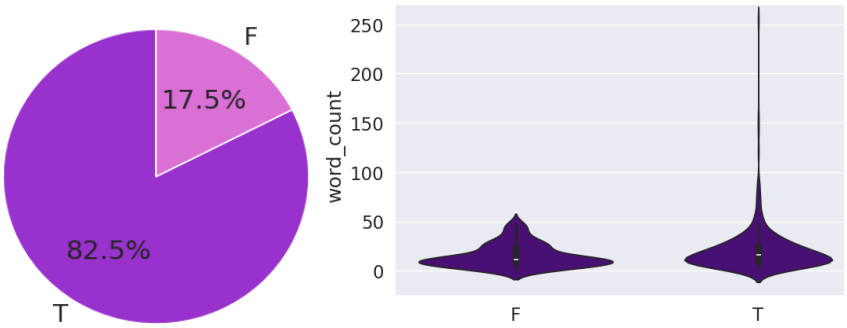


Fig.10. 'Class' pie chart for distribution and 'Text' length

Neural network architecture. To accomplish this task, we use the BERT neural network model. As described in the previous paragraphs, it is suitable for natural language processing tasks and does not require large data.

Sentiment analysis. The General Goal is to create a system for pragmatic analysis of the Ukrainian-language text, which allows you to analyse what the given text means. The goal is to conclude the given text. Decisions at the next level involve developing an analysis method. The diagram shows the types of methods. The initial step is to analyse a small text and use a method of complete analysis. The following goals are to determine the context in texts that confuse concepts and add uncertainty to the text. An important step is determining the text's emotions and the user's intent (Fig. 11). First, you

must choose an artist whose lyrics must be analysed (on these texts, it is better to check the methods of studying the emotionality of the text, because usually the poems are overloaded with them). Next, you need to enter the data for analysis and perform validation (for example, if you entered only text or symbols). You can enter the text yourself or find it on the Internet. After the input data, that is, the text has been entered, it is necessary to preprocess the text and tokenise it. After that, we send the text for analysis and collect statistics, after which we can determine the mood of the object described in the text after the user receives the study result. NLTK has been called "a great tool for teaching and working with Computational Linguistics with Python" and "a great library for playing with natural language". Libraries such as Pandas and NumPy will also be used for data manipulation.

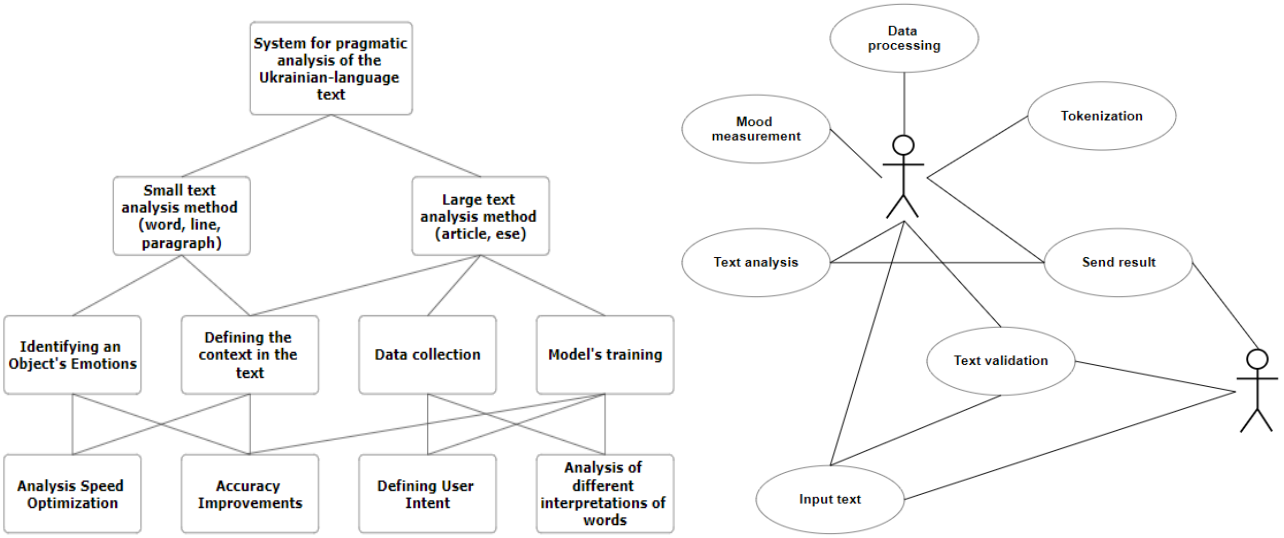


Fig.11. Tree of goals for sentiment analysis of Ukrainian text and activity diagram

As a result, a script was created that analyses emotions in the text and highlights positive and negative sentences. First, you must select and write the text in the TEXT variable in the text.py file. For a test version, let's take an excerpt from Ivan Kotlyarevsky's work "Aeneid" (Fig.12). Before the analysis, the text was tokenised using the NLTK library and the sent_tokenize method. Tokenisation took place by sentences. The next step is to analyse each sentence and identify the emotions that prevail in the text (Fig 12c). You can understand that in the first sentence, only joy is manifested; for example, in the 3rd sentence, it is sadness and anger.

```
Еней був парубок моторний
І хлопець хоть куди козак,
Удавсь на всеє зле проворний,
Завзятіший од всіх бурлак.

Но греки, як спаливши Трою,
Зробили з неї скирту гною,
Він, взявши торбу, тягу дав;
Забравши деяких троянців.

Він, швидко поробивши човни,
На синє море поспускав,
Троянців насажавши повні,
І куди очі почухрав.

Но зла Юнона, суча дочка,
Розкудкудакалась, як квочка, —
Енея не любила — страх.

Давно уже вона хотіла,
Його щоб душка полетіла
К чортам і щоб і дух не пах.
```

```
>>> Токенізований текст:
['\n'
 'Еней був парубок моторний\n'
 'І хлопець хоть куди козак,\n'
 'Удавсь на всеє зле проворний,\n'
 'Завзятіший од всіх бурлак.',
 'Но греки, як спаливши Трою,\n'
 'Зробили з неї скирту гною,\n'
 'Він, взявши торбу, тягу дав;\n'
 'Забравши деяких троянців.',
 'Він, швидко поробивши човни,\n'
 'На синє море поспускав,\n'
 'Троянців насажавши повні,\n'
 'І куди очі почухрав.',
 'Но зла Юнона, суча дочка,\n'
 'Розкудкудакалась, як квочка, —\n'
 'Енея не любила — страх.',
 'Давно уже вона хотіла,\n'
 'Його щоб душка полетіла\n'
 'К чортам і щоб і дух не пах.']
```

```
>>> Emotions:
   Happy  Angry  Surprise   Sad  Fear
0    1.0   0.00      0.00  0.00  0.00
1    0.0   0.14      0.29  0.29  0.29
2    0.0   0.00      0.00  1.00  0.00
3    0.0   0.50      0.00  0.50  0.00
4    0.0   0.00      0.20  0.40  0.40
```

Fig.12. Test data for analysis, text tokenisation results, and five-sentence analysis results from the text

In the following Fig. 13a, you can see the average values of the analysis. A method was added for statistics to determine whether sentences are positive or negative. As you can see in the picture, this text has two positive sentences (Fig. 13b) and three negative (Fig. 13c).

	Emotion	Value
0	Happy	0.200
1	Angry	0.128
2	Surprise	0.098
3	Sad	0.438
4	Fear	0.138

```
>>> Positive Sentences Count : 2
He quickly made boats,
He descended on the blue sea,
Trojans planted full,
And where the eyes scratched.

She's wanted for a long time,
Let his soul fly
To hell with the spirit and not smell.
```

```
>>> Negative Sentences Count : 3
Aeneas was a young man
And the guy at least somewhere Cossack,
Succeeded in all evil agile,
Zealous of all burlaks.

But the Greeks, as burning Troy,
They made a pile of manure out of it,
He took the bag and gave it a try;
Taking away some Trojans.

But evil Juno, sucking daughter,
Razkudkudakalas like a hen -
Aeneas did not like - fear.
```

Fig.13. Average values of the analysis, sentences with a positive character and negative sentences

For statistics, three favourite texts were found on the Internet:

- Ivan Kotlyarevsky's poem "Aeneid" (Fig. 14). You can immediately understand that the Aeneid is a literary text in which many emotions are manifested. Fear is the most common emotion in the text. This text is difficult to analyse because many literary transitions are unclear immediately.

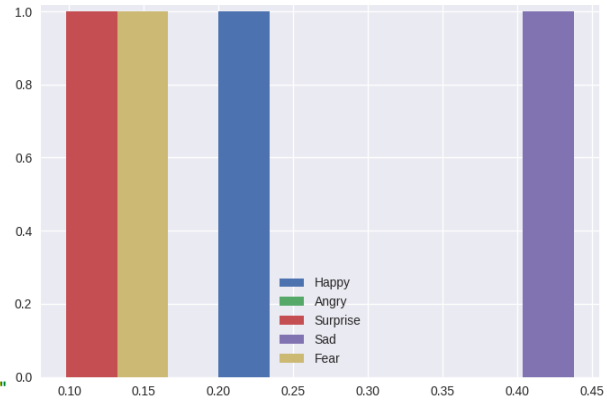
```
TEXT1 = """
Еней був парубок моторний
І хлопець хоть куди козак,
Удавсь на всеє зле проворний,
Завзятіший од всіх бурлак.

Но греки, як спаливши Трою,
Зробили з неї скирту гною,
Він, взявши торбу, тягу дав;
Забравши деяких троянців.

Він, швидко поробивши човни,
На синє море поспускав,
Троянців насажавши повні,
І куди очі почухрав.

Но зла Юнона, суча дочка,
Розкудкудакалась, як квочка, —
Енея не любила — страх.

Давно уже вона хотіла,
Його щоб душка полетіла
К чортам і щоб і дух не пах."""
```



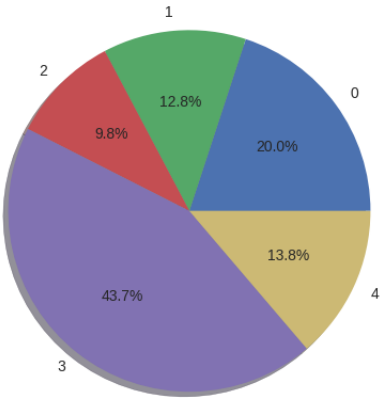


Fig.14. The text of the poem and histograms

- Text from Ukrainian news about the football match of Ukraine against Wales (Fig. 15-16). Based on the results of the analysis of the football article, you can understand that the text is about something that a person does not care about because the text contains emotions of joy and surprise. The emotion of fear is probably the understanding that the national team can lose the match.

```
TEXT2 = """
Збірній України з футболу залишилося зробити один крок для виходу на ЧС-2022.
У фіналі плей-оф українцям потрібно здолати Уельс на полі суперника.
Матч Україна - Уельс визначить останнього учасника ЧС-2022 з футболу серед європейських збірних.
Обидві країни потрапили у плей-оф за підсумками групового етапу, де фінішували другими у своїх групах.
А в першому раунді плей-оф здобули перемоги: Україна обіграла Шотландію, Уельс - Австрію.

Футбол Україна - Уельс відбудеться в Кардіффі на полі валлійців.
 Дата матчу з футболу: Україна - Уельс 5 червня.
  Це буде четверта гра в історії між цими збірними: Україна ще жодного разу не програвала валлійцям.
"""
```

Fig.15. The text of the article about football

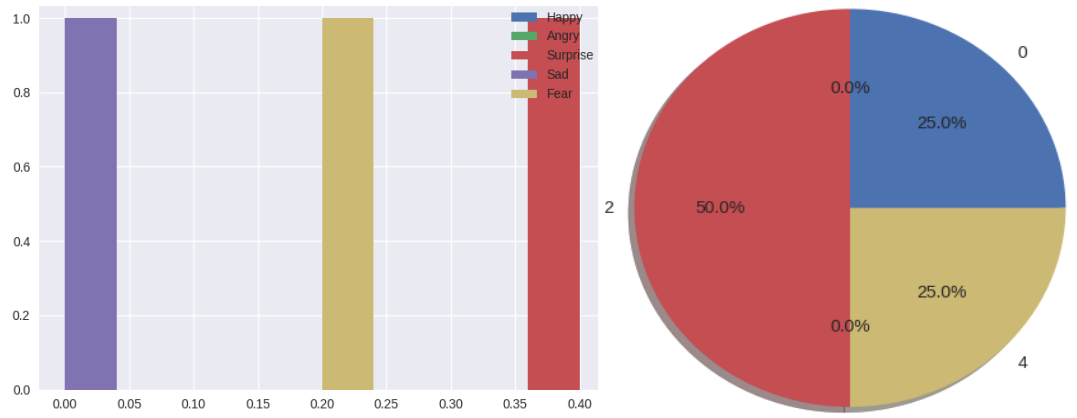


Fig.16. Histograms

- Text with news about the total losses from the russian federation in Ukraine (Fig. 17-18). After analysing the news about the war, it becomes possible to immediately see that the emotion of fear here prevails over others at times, which is entirely true.

```
TEXT3 = """Загальні збитки від російського воєнного нападу в сільському господарстві України сягнули 4,3 млрд доларів,
найбільші втрати – внаслідок знищення чи пошкодження угідь.
Про це йдеться в Огляді збитків від війни в сільському господарстві України від Центру досліджень продовольства та
землекористування KSE Institute спільно з Міністерством аграрної політики.
Так, у структурі пошкоджень найбільші втрати фіксуються внаслідок знищення або часткового пошкодження сільськогосподарських
угідь та незбору врожаю – 2,1 млрд доларів.
Крім прямого пошкодження земель, окупація, військові дії та мінне забруднення обмежують доступ фермерів до полів і
можливості збору врожаю.
Орієнтовно, 2,4 млн га озимих культур вартістю у 1,4 млрд дол залишаться незібраними внаслідок агресії РФ.
```

Fig.17. Text from the news about the losses from the russian federation

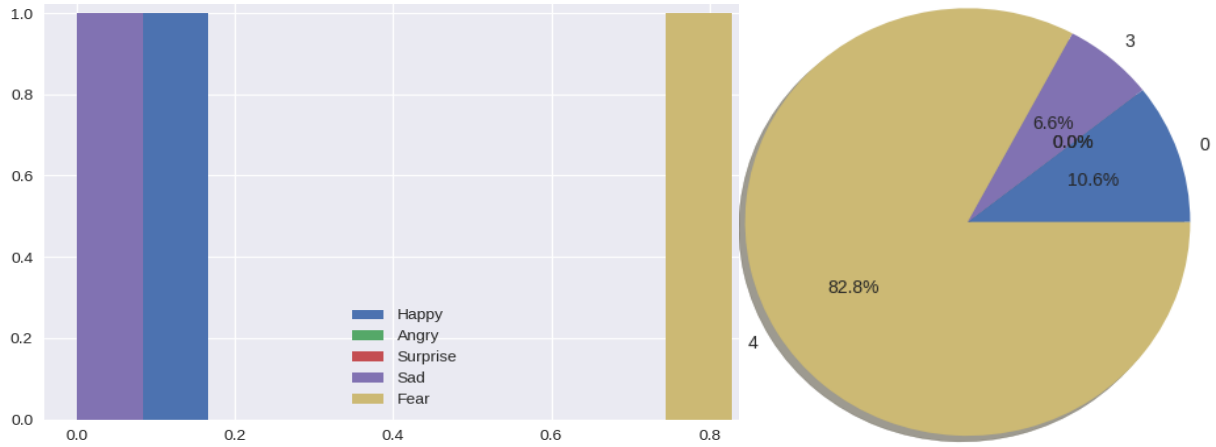
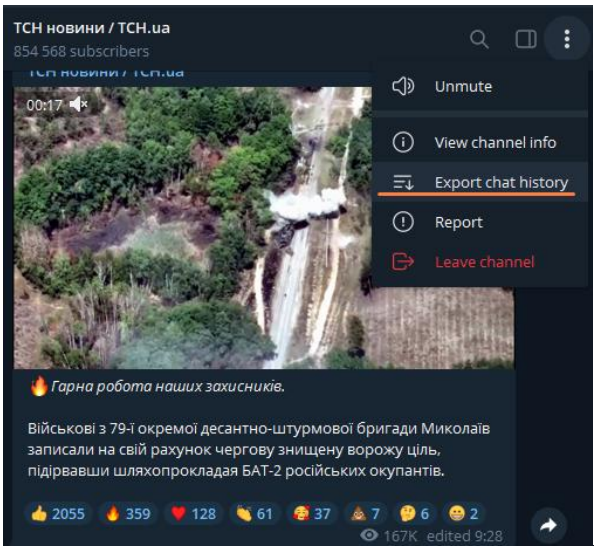


Fig.18. Histograms

Based on sentiment analysis, we used data from Telegram channels (Fig. 19a) for fake news research (such as mass media or political news). Telegram has a function that allows you to export all text (and not only) data for the entire period from any channel. It is the "Export chat history" function. Here, you can download data in video, photo, files, text and other formats. We care only about the text. We can also choose the format of the downloaded text. We will select json, as it is more convenient for manipulating the data frame. To begin with, open the file with data about the channel's telegram messages. Next, we filter the data frame from {'type': 'service'} since we are only interested in messages. We also open the sheet to extract the text from it. And we run the code for Zipf's law analysis (Fig. 19b-20). The program results show that the most popular words are {in, on, in, and, with, and, that, for, to, not} (Fig. 19b). In the program, data from the Ukrainian telegrams of the channel "TSN Novyy" were used (Fig. 21-22). Since the structure of telegram messages in all groups is the same, the user can successfully use any other news telegram channel. Now, the program has been launched on a channel with fewer publications. In the previous news, the top 10 words {in, on, in, and, with, and, what, for, to, not}. In this: {on, in, with, in, and, for, to, from, for}. 70% match.



Rank	Word	Frequency
1	у	4653
2	–	4133
3	на	3959
4	в	3566
5	та	2532
6	з	2338
7	і	1971
8	що	1765
9	за	1617
10	до	1554
11	не	1485
12	про	1322
13	україни	1212
14	україні	1149
15	від	889
16	які	753
17	-	694
18	це	668
19	а	665
20	як	655
21	для	620
22	під	600
23	із	597
24	👉	559
25	❗	536
26	сша	477
27	через	457
28	час	444
29	який	411
30	добу	397
31	президент	393
32	області	387
33	після	379
34	вже	372
35	росії	371
36	рф	362
37	києві	357
38	але	353

Rank	Word	Frequency
1	на	706
2	в	601
3	з	469
4	у	425
5	та	418
6	за	250
7	до	237
8	–	213
9	від	196
10	для	192
11	не	186
12	і	183
13	україні	158
14	про	150
15	києві	146
16	україни	138
17	що	121
18	із	120
19	під	89
20	області	89
21	які	88
22	по	83
23	як	79
24	росії	76
25	через	74
26	рф	74
27	російських	73
28	людей	72
29	ще	71
30	окупанти	67

Fig.19. An example of news, word analysis results for a total of 1419532 words and example for a total word count of 301120

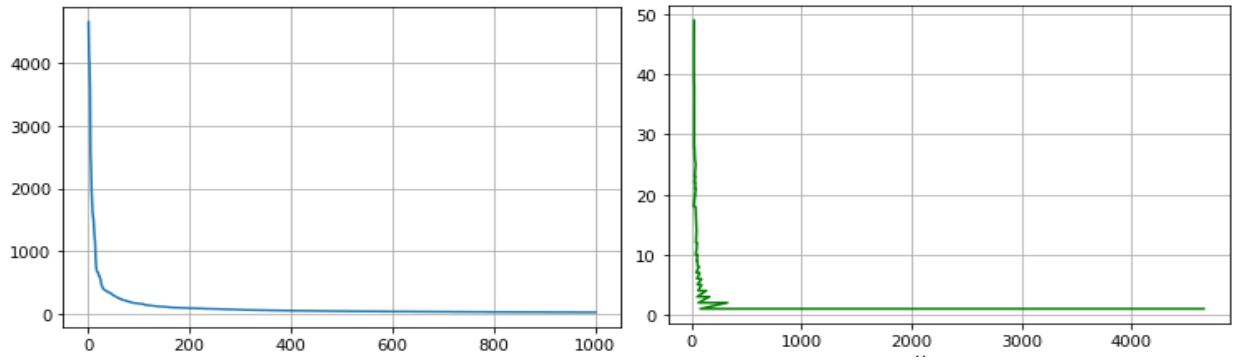


Fig.20. Zipf's first and second law

Named Entity Recognition (NER) is part of Natural Language Processing. The main goal of NER is to process structured and unstructured data and classify these named entities into predefined categories. Some common categories include name, location, company, time, monetary value, events, etc. (Fig. 22). In a nutshell, NER deals with:

- Named Object Recognition/Detection – Identify a word or series of words in a document.
- Named Object Classification – Classifies each detected object into predefined categories.

Named entity recognition makes your machine-learning models more efficient and reliable. However, it would help if you had high-quality training datasets to ensure your models perform optimally and achieve their goals. All you need is an experienced service partner who can provide you with quality datasets ready to use. If that's the case, Shaip is your best bet. Contact us for comprehensive NER datasets to help you develop practical and advanced machine-learning solutions for your AI models.

```
"id": 35703,
"type": "message",
"date": "2022-06-15T09:28:42",
"edited": "2022-06-15T09:34:48",
"from": "ТСН новини / TCH.ua",
"from_id": "channel1305722586",
"file": "(File not included. Change data exporting settings to download.)",
"thumbnail": "(File not included. Change data exporting settings to download.)",
"media_type": "video_file",
"mime_type": "video/mp4",
"duration_seconds": 25,
"width": 1280,
"height": 720,
"text": [
 "🔥",
 {
  "type": "italic",
  "text": "Гарна робота наших захисників.\n\n"
 },
 "Військові з 79-ї окремої десантно-штурмової бригади Миколаїв записали на св
]
},
{
 "id": 35704,
 "type": "message",
 "date": "2022-06-15T09:39:20",
 "edited": "2022-06-15T09:41:01",
 "from": "ТСН новини / TCH.ua",
 "from_id": "channel1305722586",
 "text": [
  "🔵",
  {
   "type": "bold",
   "text": "Основні зусилля росіян "
  },
  {
   "type": "text_link",
   "text": "зосереджені",
   "href": "https://tsn.ua/ato/v-isw-rozpovili-pro-novu-ataku-okupantiv-na-har
  },
  {
   "type": "bold",
   "text": " на південній частині Харківської області, Донецькій, Луганській о
  },
  ". Таким чином окупанти мають намір оточити українські сили на сході України
```

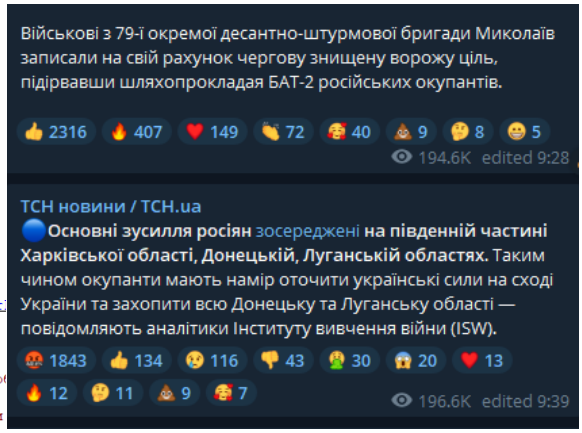



Fig.21. Code for user interface and examples of news

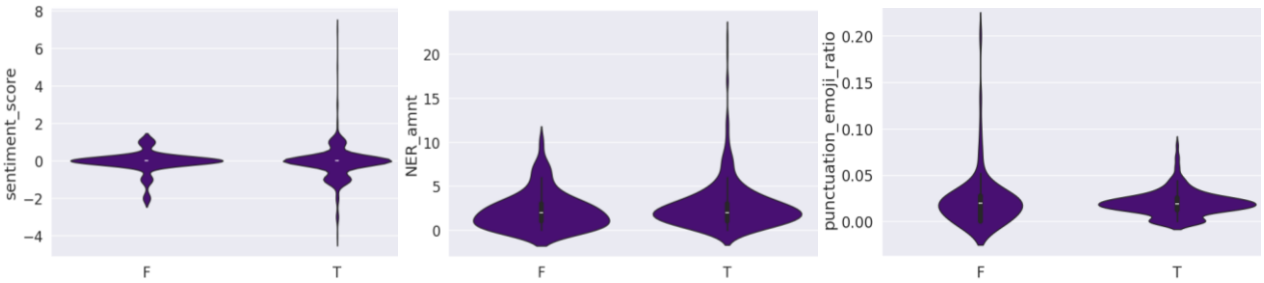


Fig.22. Sentiment analysis for news (fate and true) from our dataset and NER

Most popular NERs in class T: [('України', 55), ('Україні', 52), ('РФ', 27), ('Росія', 26), ('Україна', 24), ('США', 23), ('ЗСУ', 23), ('Росії', 12), ('Україну', 11), ('Києві', 11), ('Володимир Зеленський', 8), ('АЕС', 8), ('Зеленський', 7), ('Донеччині', 7), ('Україною', 7), ('НАТО', 7), ('росії', 6), ('ООН', 6), ('рф', 6), ('ГУР', 5), ('Охматдит', 5), ('Львові', 5), ('Харкові', 5), ('Китаю', 5), ('Курщині', 4), ('Києва', 4), ('Київ', 4), ('Голодомор', 4), ('Польщі', 4), ('Нью-Йорку', 4), ('Зеленського', 4), ('Вугледар', 4), ('Володимира Зеленського', 4), ('Чехії', 4), ('Трамп', 4), ('Африки', 3), ('ОВА', 3), ('Харкова', 3), ('Швейцарії', 3), ('СБУ', 3), ('Японії', 3), ('Тернопільщині', 3), ('Румунії', 3), ('StopFake', 3), ('Вовчанський агрегатний завод', 3), ('Львова', 3), ('МЗС', 3), ('ВОЛЯ', 3), ('Китай', 3), ('Байден', 3), ('Дмитрука', 3), ('Лондоні', 3), ('Захід', 3), ('Bloomberg', 3), ('Монтре', 3), ('Запоріжжя', 3), ('Іван Федоров', 3), ('Вугледара', 3), ('Збройні сили', 2), ('Донецькій області', 2), ('Донбас', 2), ('Африка', 2), ('росія', 2), ('Дніпра', 2), ('Харкову', 2), ('Таджикистану', 2), ('Херсонщині', 2), ('Крокус Сіті Хол', 2), ('УПЦ МП.', 2), ('Закарпаття', 2), ('Росію', 2), ('Олег Синєгубов', 2), ('кремль', 2), ('Краматорська', 2), ('Курській області', 2), ('Франції', 2), ('Сил оборони', 2), ('Єврокомісії', 2), ('Путін', 2), ('DeepState', 2), ('Туреччина', 2), ('Естонії', 2), ('Верховна Рада України', 2), ('Гітлера', 2), ('Курської області', 2), ('ТУ САМУ СІЛЬ', 2), ('Умєров', 2), ('Гнезділова', 2), ('Суспільному', 2), ('Ради Безпеки ООН', 2), ('Генасамблеї ООН', 2), ('Харківщині', 2), ('Грузії', 2), ('Великої Британії', 2), ('ЄС', 2), ('Анатолій Тимощук', 2), ('Байдена', 2), ('Будапешті', 2), ('Олександра Усика', 2), ('Кракова', 2), ('ВР', 2), ('Politico', 2), ('МЗС України', 2), ('Павела', 2), ('Путіна', 2), ('Ердоган', 2), ('Запорізької ОВА', 2), ('Байдену', 2), ('Конгресу', 2), ('Ван Ї.', 2), ('Дональд Трамп', 2), ('Reuters', 2), ('Джо Байдена', 2), ('Запоріжжі', 2), ('Нацбанк', 2), ('Львів', 2), ('Покровська', 2), ('Кремлі', 2), ('The New York Times', 2), ('Anonymous', 2), ('Тихому океані', 2), ('Міністерство оборони Китаю', 2), ('Словаччини', 2), ('Фіцо', 2), ('Заходу', 2), ('USKO MFU', 2), ('Криму', 2)]

Most popular NERs in class F: [('ЗСУ', 8), ('США', 7), ('рф', 5), ('Україні', 5), ('Донецькій області', 4), ('Донбасі', 4), ('України', 4), ('Курщину', 3), ('Курщині', 3), ('Україну', 3), ('Дональд Трамп', 3), ('Курській області', 2), ('Донеччині', 2), ('Курському напрямку', 2), ('Донеччини', 2), ('The Financial Times', 2), ('Дніпра', 2), ('Кремль', 2), ('Львові', 2), ('Харкові', 2), ('Росії', 2), ('Зеленського', 2), ('Польщі', 2), ('Чарльз III', 2), ('Республіканської партії', 2)]

Our program aims to detect disinformation in text documents. Misinformation can come in many forms, such as fake news, manipulative articles, or distorted information that misleads readers. The program's name, "Disinformation Detector", clearly reflects the primary purpose and function of the software. The program offers a tool for detecting disinformation in text documents [29-33], which can be helpful for a variety of purposes, including analysing texts for

truthfulness, detecting manipulation or fake news, and for research in media literacy and information security.

Purpose of the program:

- *Detection of misinformation in the text*. The program's primary function is to analyse the text to identify fragments containing misinformation. It may include texts that spread false or distorted information, manipulate facts or create a misleading impression.
- *Choice of model and parameters.* The program was developed considering the possibility of analysing different formats of text documents. It supports formats such as CSV, which allows you to conveniently download and analyse text data. The CSV format is one of the most common formats for storing tabular data, which makes it versatile for software processing.
- *The model Training module* of the app allows it to create a machine learning model that can "learn" from previously collected data. The process of training the model is for the program to automatically analyse the data set, identify patterns in it, and establish connections between various text characteristics that may indicate the presence of misinformation. For this purpose, a proprietary dataset focused on the Ukrainian language, including multiple examples of disinformation and truthful texts. This dataset contains texts from various sources, including news, social media, blogs, and other publications. Data was collected to create the most accurate and representative set for training the model.
- *Display Results.* This program feature provides a user-friendly interface for displaying analysis results to the user via the command line. The program outputs the results, including the predicted label (true or false) and probabilities for each class, simplifying interpreting and using the information. In addition, the program can generate reports in CSV format, which allows users to save and analyze the results later.

The functional purpose of the program is to analyze text documents to identify fragments that may contain misinformation. The main aspects of this functional purpose include:

- *Analysis of text documents.* The program can accept text documents in CSV format and analyze them for the presence of misinformation.
- *Detection of fragments with disinformation*. During the analysis, the program identifies parts of the text that indicate the possible use of disinformation. These can be fragments that look like fake news, manipulative articles or other signs that indicate distortion of information.
- *Model training.* The program uses its dataset to train the model, which allows it to consider the specifics of the Ukrainian language and context. The model learns to recognize different types of misinformation based on input data.
- *Notification of results.* After the analysis, the program notifies the user of the results, indicating the fragments of text that contain misinformation and providing a rating or other information about the detected misinformation. The program can also generate reports for further analysis and storage of results.

Structure of the program. This program is divided into modules and components to organize and structure the code better. They perform various functions and tasks within the program. Components and modules:

- *The text analysis module contains a set of methods* responsible for processing the text, identifying fragments that may contain misinformation, and calculating their percentage relative to the total text volume. These methods provide analysis of textual data to detect and assess the presence of misinformation, helping the user to understand how susceptible the text is to false information or manipulation.
- *The document Upload Module allows the user to upload text documents* to the program for further analysis. It provides an interface through which the user can select a file or files from their device and pass them to the application for processing. Once downloaded, the program may analyse the content of these documents to detect misinformation or other specific characteristics.
- *Model Training Module includes* a set of techniques for training a machine learning model based on a dataset. It enables the program to use machine learning algorithms to analyse and learn the structure and features of input data. These methods may include data preparation, selection and tuning of learning algorithms, execution of the learning process, and evaluation of the resulting models. The output of this component is a trained model that can be used for further analysis and prediction based on new input data.

Main functions of the program.

- *Training a machine learning model based on a dataset*.

```
def load_training_data(file_path):
    df = pd.read_csv(file_path)
    return df
def train_model(training_data):
```

```
    vectorizer = TfidfVectorizer()
    X_text = vectorizer.fit_transform(training_data["text"]).toarray()
    y = np.array(training_data["label"])
    X_train, X_test, y_train, y_test = train_test_split(X_text, y, test_size=0.2, random_state=0)
    trained_model = MultinomialNB()
    trained_model.fit(X_train, y_train)
    y_prediction = trained_model.predict(X_test)
    acc = accuracy_score(y_test, y_prediction)
    f1 = f1_score(y_test, y_prediction)
    confusion = confusion_matrix(y_test, y_prediction)
    print(f"Accuracy: {acc}")
    print(f"F1: {f1}")
    print(f"Confusion Matrix: ")
    print(confusion)
    return trained_model, vectorizer
```

- *Downloading text documents for analysis.*

```
def upload_document():
    file_path = filedialog.askopenfilename(filetypes=[("CSV files", "*.csv")])
    if file_path:
        df = pd.read_csv(file_path)
        content = df["text"].tolist()
        return content
    else:
        messagebox.showwarning("Invalid File", "Upload a valid CSV document...")
        return None
```

- *Text analysis with detection of fragments containing misinformation.*

```
def analyse_text(input_text, model, vectorizer):
    input_vector = vectorizer.transform([input_text]).toarray()
    prediction = model.predict(input_vector)
    prediction_proba = model.predict_proba(input_vector)
    if prediction[0] == 1:
        result = " Misinformation "
        probability = prediction_proba[0][1] * 100
    else:
        result = " True information "
        probability = prediction_proba[0][0] * 100
    print(f"Результат: {result}")
    print(f" Probability: {probability:.2f}%")
    return result, probability
```

- *Display of analysis results in the command interface.*

```
def display_results(result, probability):
    print("\nРезультати аналізу:")
    print(f" Result: {result}")
    print(f" Probability: {probability:.2f}%")
```

Description of the technologies used.

- *Python* is a general-purpose programming language used to develop various applications, from web applications to data mining and artificial intelligence. In our project, this is the primary programming language.
- *NumPy* is a Python library for calculating data. It supports large arrays and matrices, along with high-level math functions for working with these arrays. In our project, this library is used for operations with arrays and matrices.
- *Pandas* is a Python library for data processing and analysis. It provides data structures such as DataFrames that simplify working with tabular data and a set of functions for their processing and analysis. In the project, it is used for data processing and analysis.
- *scikit-learn* is a Python library for machine learning and data analysis. It contains a variety of machine learning algorithms, including classification, regression, clustering, and others, as well as tools for building and evaluating models. In the project, it is used for model training and text vectorization.
- *Faiss is a library developed by Facebook to find similar vectors in large datasets efficiently*. It supports a fast search of similar vectors, which makes it useful for clustering and similarity search tasks. The project uses it to find the nearest neighbours in the vector space.
- *imbalanced-learn* is a Python library that provides tools for working with unbalanced data. It contains various methods for balancing classes, such as oversampling and undersampling. In the project, it is used to balance training data.
- *tkinter* is a standard Python library for creating graphical user interfaces (GUIs). It provides various tools and widgets for creating various user interfaces. Although there is no GUI in this project, tkinter can be used for future updates if there is a need for an interactive interface.

- *CSV* (Comma Separated Values) is a format for storing tabular data. It allows data to be stored as text, where each line is a record, and commas separate each field. In the project, it is used to store and load text data for analysis.

User manual. Disinformation detection software has a wide range of applications. It can be helpful in scientific research to analyse the texts of scientific articles, reports or publications to detect misinformation. Information platforms such as libraries, databases, or online resources can use this program to identify materials that contain misinformation so that users can find reliable information faster. In the medical field, the program can help identify articles or documents where studies contain inaccurate information. In the technology industry, the program can be used to analyse technical documentation, patents, or manuals to detect misinformation. In the educational sector, the program can serve as a tool for analysing the texts of scientific articles or educational materials to identify inaccurate information. The program can analyse textual details in reports, analytical materials, or news in business and finance to detect misinformation in various industries. The program is designed for users with different levels of training, from beginners to advanced users. It has an intuitive interface allows you to easily interact with the program. Even users without experience working with programs can quickly master their essential functions. Users with a certain level of technical training can use the program more effectively, using additional capabilities for text analysis and working with results. For example, they can understand the meaning of the accuracy metrics used in the text analysis process or adjust the parameters of a machine learning model to get better results.

Level of user training. This program is intended for users of various training levels, from beginners to advanced users. It has an intuitive interface allows you to easily interact with the program. Thus, even users without previous experience with programs can quickly master their primary functions. Users with a certain level of technical training can use the program more effectively, using additional capabilities for text analysis and working with results. For example, they can understand the meaning of the accuracy metrics used in the text analysis process or adjust the parameters of a machine learning model to get better results.

Starting the program. When starting the program, the user sees the interface, which consists of various elements designed to interact with the program. This interface is divided into two main zones or areas. Each of these zones has its purpose and functionality. The first zone is intended for text input. In this area, the user can directly type or insert text the program will analyse. After analysing the text, the program outputs a result that shows whether the entered text is misinformation or accurate information. The second zone contains all the program functionality. The buttons and other interface elements allow the user to interact with the program. For example, these can be buttons for starting text analysis, loading documents, selecting analysis parameters, etc. The user can add text to the program (Fig. 23).

Performing the analysis. After the user has added some additional features to the text analysis, we are finally ready to start using our text analysis application (Fig. 24).

```
true_text = "Мобілізація в Україні розпочинається з 18 років. Це передбачено чинним законодавством, яке регулює
classify_new_text(true_text)
false_text = "Україна дозволила мобілізувати дітей з 10 років для участі у військових діях. Всі учні шкіл тепер
classify_new_text(false_text)
```

Fig. 23. Added text to the program

```
New text: Мобілізація в Україні розпочинається з 18 років. Це передбачено чинним законодавством, яке регулює питання військової служби для громадян. Чоловіки,
Predicted label: 0
Prediction probabilities: [0.64867946 0.35132054]
Nearest neighbors indices: [1 4 2 3 5]
Nearest neighbors distances: [0.        1.9147311 1.9163284 2.        2.       ]
Nearest neighbors labels: 0    0
4    1
8    1
2    1
3    0
Name: label, dtype: int64
New text: Україна дозволила мобілізувати дітей з 10 років для участі у військових діях. Всі учні шкіл тепер зобов'язані проходити військову підготовку і можут
Predicted label: 1
Prediction probabilities: [0.40557381 0.59442619]
Nearest neighbors indices: [1 3 2 4 0]
Nearest neighbors distances: [1.3344624 1.3442578 1.6647459 1.8291727 1.8353997]
Nearest neighbors labels: 0    0
2    1
8    1
4    1
5    0
Name: label, dtype: int64

Process finished with exit code 0
```

Fig.24. Performing the analysis

Analysis of results. After the program finished parsing the text, it provided the following information to the user:

- for the text

Мобілізація в Україні розпочинається з 18 років. Це передбачено чинним законодавством, яке регулює питання військової служби для громадян. Чоловіки, які досягли 18 років, підлягають військовому обліку і можуть бути призвані на службу.
[Mobilization in Ukraine begins at the age of 18. It is stipulated by the current legislation that regulates the issue of military service for citizens. Men who have reached the age of 18 are subject to military registration and can be called up for service.]

The program determined the label "0", which means true information. The forecast probabilities were 64.87% in favour of true information and 35.13% in favour of misinformation. The nearest neighbours of this text vector indicated that they were also classified as true information.

- for the text

Україна дозволила мобілізувати дітей з 10 років для участі у військових діях. Всі учні шкіл тепер зобов'язані проходити військову підготовку і можуть бути відправлені на передову."
[Ukraine allowed the mobilization of children from the age of 10 to participate in military operations. All schoolchildren must now undergo military training and can be sent to the front.]

The program identified a label of "1", which means misinformation. The forecast probabilities were 40.56% in favour of true information and 59.44% in favour of misinformation. The nearest neighbours of this text vector confirmed the classification as misinformation.

This functionality helps the user to recognize and classify texts as true or misinformation, which is vital for further analysis and decision-making. The program provides detailed information about the vector's prediction probabilities and nearest neighbours, which helps the user better understand the basics of classification.

During the work, a text analysis program was developed to detect disinformation. The program works through the command line, allowing users to interact with it without needing a graphical interface. It will enable downloading text documents in CSV format for further analysis. In addition, the program includes the functionality of adding additional elements, such as sources and references, to improve the accuracy of the analysis. After loading the text, the user can start the analysis process using the appropriate command on the command line. After the analysis, the program provides results, including each class's predicted label (true or false) and probabilities.

In conclusion, we can say that this program allows users to quickly and efficiently analyse texts from the point of view of the presence of misinformation. Its ease of use via the command line makes it an essential tool for research in the field of text analysis for disinformation. The program helps users determine the reliability of information, which is especially important in today's information environment.

Running the test case. Example of using the function for new text:

```
# True text
true_text1 = "Мобілізація в Україні розпочинається з 18 років. Це передбачено чинним законодавством, яке регулює питання військової служби для громадян. Чоловіки, які досягли 18 років, підлягають військовому обліку і можуть бути призвані на службу."#"Mobilization in Ukraine begins at the age of 18. It is stipulated by the current legislation that regulates the issue of military service for citizens. Men who have reached the age of 18 are subject to military registration and can be called up for service."
classify_new_text(true_text1)
true_text2 = "В Україні прийняли новий закон, який дозволяє громадянам зберігати зброю вдома для самозахисту."#"Ukraine adopted a new law that allows citizens to keep weapons at home for self-defence."
classify_new_text(true_text2)
true_text3 = "Міністерство охорони здоров'я України повідомило про зростання кількості вакцинацій проти коронавірусу."#"The Ministry of Health of Ukraine reported an increase in vaccinations against the coronavirus."
classify_new_text(true_text3)
true_text4 = "Україна отримала нову партію гуманітарної допомоги від Європейського Союзу."#"Ukraine received a new batch of humanitarian aid from the European Union."
classify_new_text(true_text4)
true_text5 = "В Україні проводять широкомасштабні військові навчання з участю НАТО."#"Large-scale military exercises with the participation of NATO are being conducted in Ukraine."
classify_new_text(true_text5)
true_text6 = "Президент України оголосив про нову економічну реформу, яка спрямована на зниження податкового тиску на малий бізнес."#"The President of Ukraine announced a new economic reform aimed at reducing the tax burden on small businesses."
classify_new_text(true_text6)
true_text7 = "Усі школи в Україні зобов'язані перейти на дистанційне навчання через збільшення випадків захворювань на COVID-19.""All schools in Ukraine are obliged to switch to distance learning due to the increase in cases of COVID-19."
classify_new_text(true_text7)
true_text8 = "Міністерство фінансів України запровадило нові правила для ведення бухгалтерського обліку для малих підприємств."#"The Ministry of Finance of Ukraine introduced new rules for accounting for small businesses."
classify_new_text(true_text8)
true_text9 ="Уряд України запланував будівництво нової швидкісної залізниці між Києвом та Одесою." #"The Ukrainian government has planned the construction of a new high-speed railway between Kyiv and Odesa."
classify_new_text(true_text9)
true_text10 = "Міністерство освіти України оголосило про нові стипендії для студентів, які досягли високих результатів у навчанні."#"The Ministry of Education of Ukraine has announced new scholarships for students who have achieved high academic results."
```

```
classify_new_text(true_text10)
# False text
false_text1 = "Україна дозволила мобілізувати дітей з 554 років для участі у військових діях. Всі учні шкіл тепер зобов'язані проходити військову підготовку і можуть бути відправлені на передову."#"Ukraine allowed the mobilization of children from the age of 554 to participate in military operations. All school students must now undergo military training and can be sent to the front."
classify_new_text(false_text1)
false_text2 = "Всі громадяни України зобов'язані здати свої паспорти до кінця року."#"All citizens of Ukraine are required to hand in their passports by the end of the year."
classify_new_text(false_text2)
false_text3 = "Вакцини проти коронавірусу містять мікрочіпи для стеження за громадянами."#"Coronavirus vaccines contain microchips to track citizens."
classify_new_text(false_text3)
false_text4 ="Європейський Союз заборонив українцям в'їзд до своїх країн без пояснення причин." #"The European Union has banned Ukrainians from entering their countries without explaining the reasons."
classify_new_text(false_text4)
false_text5 = "Україна стала першою країною, що заснувала колонію на Марсі."#"Ukraine became the first country to establish a colony on Mars."
classify_new_text(false_text5)
false_text6 = "В Україні введено комендантську годину для всіх громадян з 20:00 до 06:00."#" In Ukraine, a curfew has been introduced for all citizens from 20:00 to 06:00."
classify_new_text(false_text6)
false_text7 = "Україна стала членом Європейського Союзу з правом голосу у всіх рішеннях."#"Ukraine became a member of the European Union with the right to vote in all decisions."
classify_new_text(false_text7)
false_text8 = "Українська влада заборонила всім громадянам мати більш ніж один мобільний телефон."#"Ukrainian authorities have banned all citizens from having more than one mobile phone."
classify_new_text(false_text8)
false_text9 = "Усі громадяни України зобов'язані щомісяця здавати аналізи крові на вимогу уряду."#"All citizens of Ukraine are obliged to take blood tests every month at the government's request."
classify_new_text(false_text9)
false_text10 = "Усі автомобілі в Україні мають бути обладнані пристроями для стеження за місцем розташування."#"All cars in Ukraine must be equipped with location tracking devices."
classify_new_text(false_text10)
```

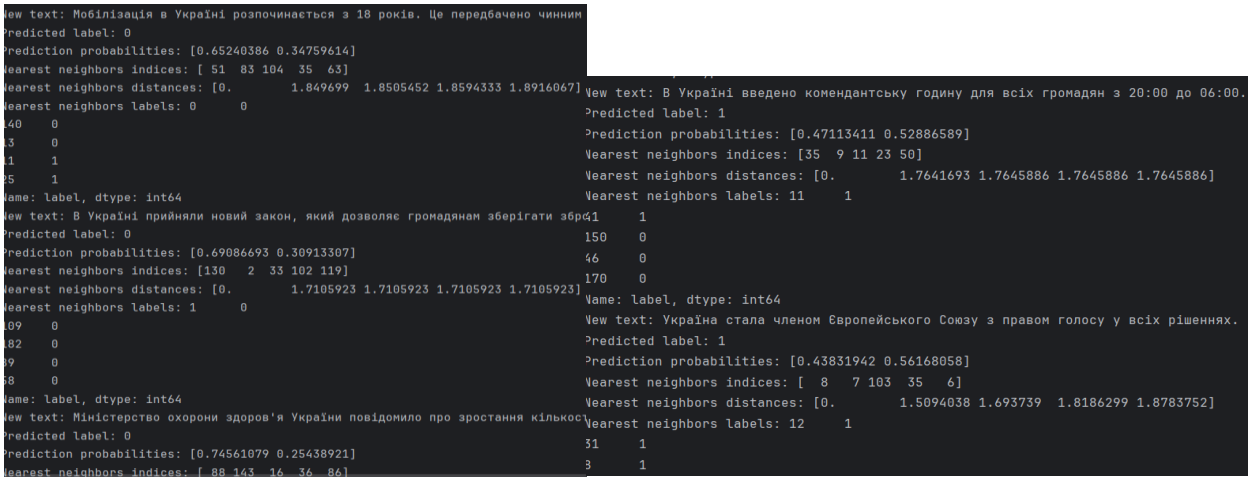


Fig.25. Example of using the function for new text

Text analysis. After the user has entered text or uploaded a file, he can start the analysis by clicking the appropriate button in the command interface. It initiates the processing process in which the program analyses the text. First, the text is converted into vector input data for machine learning models. Vectorization helps to present the text in a numerical format, which allows the program to efficiently process it and detect characteristic patterns. During vectorization, the program may consider additional options if selected by the user. For example, if the "References" or "Sources" options were selected, these parameters were added to the vectors and considered during the analysis. Thus, the program can process various types of information and adapt to additional requirements. After the program converts the text into vectors using the vectorization process, these vectors are fed into a machine-learning model for further analysis. The model uses these vectors to predict the text's probability of containing misinformation. The machine learning model used in the program has been pre-trained on a large data set that includes texts that contain both true information and misinformation. Thanks to this, it can recognize characteristic patterns found in texts that contain misinformation. If the program detects such patterns or features in the analysed text, it affects the analysis results, making it more likely that the text includes misinformation. The program will analyse these texts and receive results showing whether each text contains misinformation. Thus, running text analysis is a central step involving converting the text into an analysis-friendly format and running machine learning to determine its likely origin.

Analysis of results. When the text analysis is complete, the program displays the results in a particular output field where the user can quickly review and interpret them. This field is usually found in the command interface or displayed as text output after completing the analysis. In the output field, the user receives information about the percentage of text likely to contain misinformation. This percentage is calculated based on analysis performed by a machine learning model. The model reveals characteristic features that may indicate the presence of misinformation. The percentage may vary depending on the complexity and features of the text, as well as the model used. The program can highlight fragments in red text that are identified as misinformation. To facilitate visual recognition, the user can see which parts of the text most likely contain false or manipulative information. Such underlining helps to quickly identify suspicious areas of the text. The program can also provide a rating based on the percentage of text identified as misinformation. The rating lets the user get a general idea of the text's credibility level. If the percentage is deficient, the rating may be "Reliable", meaning the text is reliable. If the percentage is average, the rating can be "Caution", indicating the possible presence of misinformation. If the percentage is high, the "Unreliable" rating means that the text contains significant misinformation.

Various researchers collected the dataset for the study, namely the first part – by linguistic researchers and the second part – by researchers in IT technologies. Fig. Figure 26a shows the relationship between truth and fakes and the number of articles, and in Fig. 26b, the ratio between the number of articles and the language (the first part of the dataset).

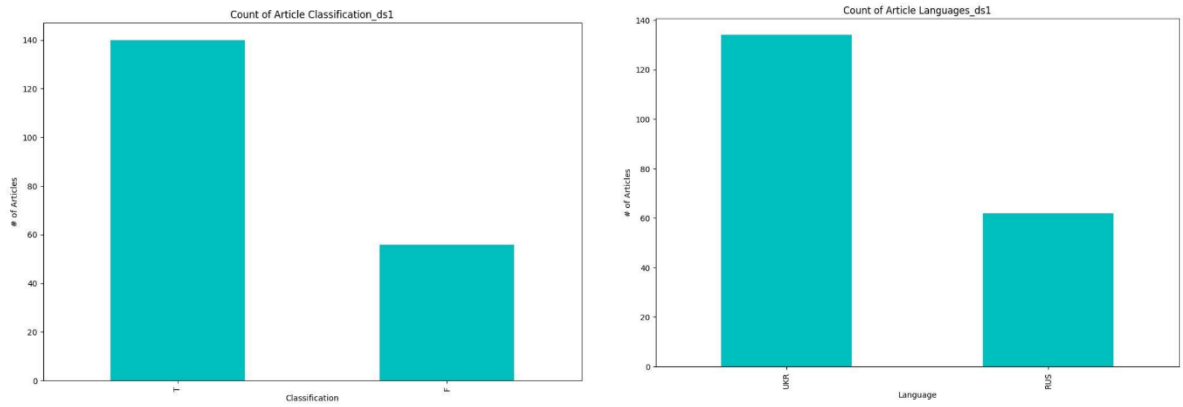


Fig.26. Diagram of the correlation (the first part of the dataset) between a) – truth-fakes and the number of articles; b) – the number of articles and the language

Fig. 27a and 27b show the same ratios for the second part of the dataset, respectively. After combining the first and second parts of the dataset, the correlation between truth and fakes and the number of articles is shown in Fig. 28a, and the ratio between the number of articles and the language is shown in Fig. 28b.

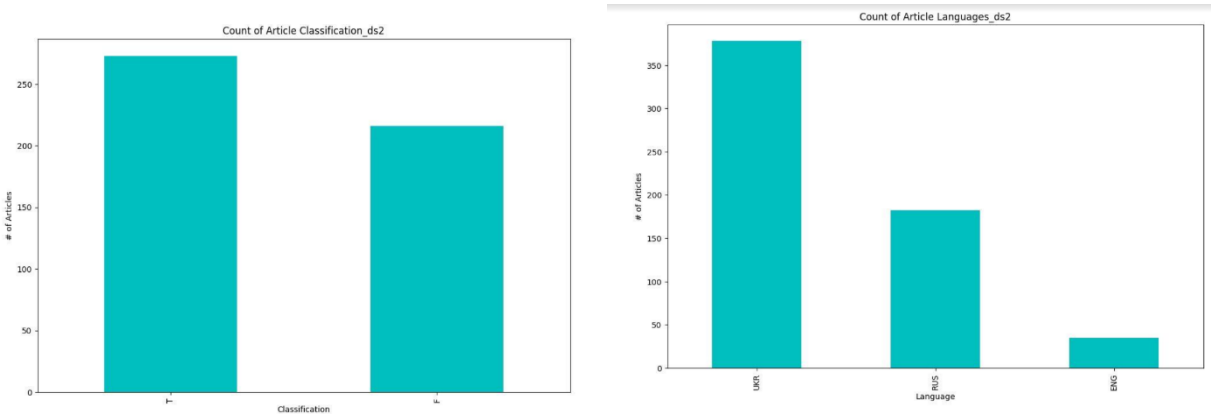


Fig.27. Diagram of the correlation (second part of the dataset) between a) – truth-fakes and the number of articles; b) – the number of articles and the language

Grammatically and qualitatively constructed fakes, propaganda, and disinformation datasets improve their use in systems for recognizing and identifying fake new ones in the Internet space based on NLP methods, machine learning and semantic and linguistic analysis of big data, including lemmatization and tokenization.

When applying classical machine learning classifiers, such as logistic regression, for English-language texts and using balanced datasets for training, the accuracy of identifying and recognizing fakes is often greater than 90%. For the forecast model, the BoW and Logistic Regression functions were used. The model results for the classification of English text are shown in Fig. 29. The F1 score is 0.98 for both classes (0-fake, 1-non-fake). Such good results can be attributed to the "laboratory" quality of the dataset.

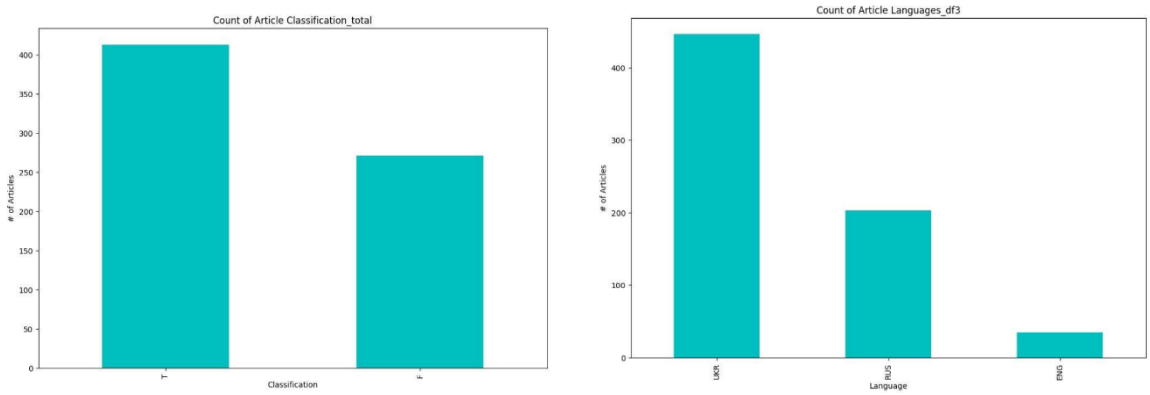


Fig.28. Diagram of the correlation (combined dataset) between a) – truth-fakes and the number of articles; b) – the number of articles and the language

```
print(classification_report(y_test, y_pred_lr))
```

	precision	recall	f1-score	support
0	0.98	0.99	0.98	2971
1	0.99	0.98	0.98	3029
accuracy			0.98	6000
macro avg	0.98	0.98	0.98	6000
weighted avg	0.98	0.98	0.98	6000

Fig.29. Model results

In further experiments, we want to test the model on real-time news. It is a problem for Ukrainian-language content. A model based on TF-IDF (Term Frequency-Inverse Document Frequency) and a model based on BERT (Bidirectional Encoder Representations from Transformers) were used to train and test news in Ukrainian.

The TF-IDF-based model allows you to determine the importance of words in text about the entire collection of documents. After the TF-IDF was calculated, each text was compared in cosine similarity with the database's (DB) texts. It allows you to determine the similarity between the entered text and texts from trusted or untrusted sources. The problem lies in the poor understanding of semantics, the lack of support for multilingualism, and the low accuracy of identifying the Ukrainian language fake. After training the TF-IDF-based model, testing it on new data, and adding it to the dataset, we obtained the following results in Fig. 30a. The model achieves an overall accuracy of 0.846. It means that 84.6% of the model's predictions were correct. For Class 0 (true), the model has a high accuracy (0.78) and an ideal recall score (1.00), indicating that the model is good at detecting truthful texts. For Class 1 (disinformation), the model has a high accuracy (1.00) but a lower recall score (0.67), indicating that the model may miss some instances of disinformation. However, it accurately identifies those instances and classifies them as disinformation. F1-Score is a harmonic mean between precision and recall. For Class 0, this score is 0.88; for Class 1, it is 0.80, indicating a balanced model performance between these two metrics.

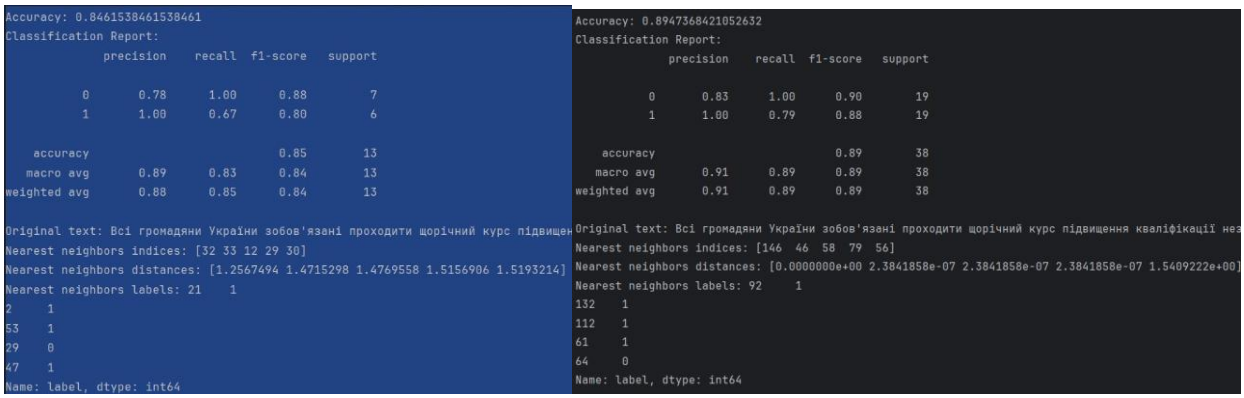


Fig.30. Model results based on TF-IDF and BERT of Ukrainian-language news

The BERT-based model makes it possible to better analyze the semantics of the text. Using BERT made it possible to convert texts to vector representations, after which a linear search took place for the nearest vectors in the database. The problem is low search efficiency, high resource consumption, and insufficient support for the Ukrainian language. Linear vector search takes too much time when processing large databases of texts, which reduces the speed of analysis. Using the model is computationally expensive and requires significant resources to run on conventional machines. The model is not optimized for multilingual processing and does not give the best results for Ukrainian-language texts. It leads to a loss of accuracy in analyzing such texts, especially in cases with complex language constructions and contexts. After

training based on BERT (Fig. 30b) and testing it on the same data and supplementing the dataset, the following results were obtained for different samples from the dataset: Accuracy – 0.895, Precision for the true/fake class – 0.83/1.00, respectively, Recall – 1.00/0.79 and F1-Score – 0.90/0.88, indices of the nearest neighbours – [146, 46, 58, 79, 56], and all distances to the nearest neighbours go to 0.

After the training and testing model on new data and supplementing the dataset, we got the following results:

- Accuracy of the model (Accuracy) is 0.846;
- Precision for class 0 (true) is 0.78;
- Recall for class 0 (true) is 1.00;
- F1-Score for class 0 (true) is 0.88;
- Precision for class 1 (misinformation) is 1.00;
- Recall for class 1 (misinformation) is 0.67;
- F1-Score for class 1 (misinformation) is 0.80.

Now, let's analyse the results of our control testing.

- In text containing truthful information, it was determined that only 10% of the content could contain misinformation. This text has been rated "Reliable", indicating that most content is reliable and high-quality.
- In this case, 85% of all content was disinformation. This text has received an "Unreliable" rating, which indicates that almost all of the content contains false information, and its credibility is much lower.

From this analysis, we can conclude that texts that mostly contain true information receive higher ratings, while significant content of misinformation in the text leads to lower ratings. Thus, the user gets a complete picture of the analysed text when displaying the results. The program provides a percentage, rating, and recommendations for further actions. This information allows the user to quickly and efficiently determine how suspicious the text may be and make appropriate decisions about its authenticity or the need for further analysis.

The model achieves an overall accuracy of 0.846 (this means that 84.6% of all model predictions were correct). For class 0 (true), the model has high precision (0.78) and perfect recall (1.00), indicating that the model is good at detecting true texts. For class 1 (misinformation), the model has a high precision (1.00) but a lower recall (0.67), indicating that the model may miss some cases of misinformation. However, it accurately identifies those cases and classifies them as misinformation. F1-Score is a harmonic mean between precision and recall. For class 0, this indicator is 0.88, and for class 1 - 0.80, which indicates a balanced performance of the model between these two metrics.

Correspondence of the functioning of the system to the assigned task. A program developed to detect texts containing disinformation must meet specific criteria to ensure its effectiveness, reliability and practicality. These criteria are the basis for determining the quality and effectiveness of the program, allowing it to perform the task with high accuracy. Criteria that the system must meet: Mixed text analysis, Percentage display, Text underlining, Text rating and High accuracy. However, to have a more complete picture of how effective our implemented program is, we must consider these criteria in more detail and analyse how exactly our program meets each of them:

- *Analysis of mixed text*. The program we developed can analyse texts containing fragments of both true information and disinformation. Its ability to correctly identify mixed texts helps users get accurate results about the origin of different text parts. It is beneficial when a single document includes multiple sources of information or uses different writing styles. The program uses modern machine learning technologies to identify specific patterns and features characteristic of disinformation. It analyses each sentence from mixed texts, turning them into vectors and using a model that recognizes which parts of the text contain false or manipulative information.
- *Percentage display*. The program displays the percentage of text that is likely to contain misinformation. It allows users to quickly assess how much of the text is false information. The percentage value helps to understand the text's overall reliability level.
- *Text underlining*. The program highlights those parts of the text identified as misinformation in red to facilitate visual recognition. It gives the user a visual way to see exactly which pieces of text are suspicious. Underlining helps quickly identify the text's problematic areas and take the necessary measures.
- *Rating of the text*. The program also provides a text rating based on the percentage of misinformation. The rating gives users a general idea of the text's credibility. For example, a low rate of misinformation may correspond to a "Reliable" rating, a medium percentage to "Caution", and a high percentage to "Unreliable". It helps users make informed decisions about using or validating text.
- *High accuracy*. The program achieves high accuracy due to modern machine learning algorithms and a large amount of training data, including texts with true information and disinformation. The model's accuracy is confirmed by the results of testing on control texts, which provides users with reliable results.

In general, our program meets the task of detecting misinformation in texts, providing analysis of mixed text, showing the percentage of misinformation, visually highlighting suspicious fragments, and providing a rating and high

accuracy of the analysis. It makes the program effective for combating disinformation and improving information security. In conclusion, creating a program for reliable detection of misinformation in texts is a significant achievement in natural language processing and machine learning. Thanks to advanced algorithms and large amounts of data for training, the program has demonstrated high accuracy in recognizing texts with true information and misinformation, reducing the risk of false conclusions. The model successfully identifies the characteristic features inherent in disinformation texts, providing accurate analysis results. It is an essential tool for users who seek to distinguish true information from manipulative or false texts. The program allows you to analyse texts of various origins, identify suspicious fragments and provide recommendations regarding their authenticity. As a result, this program can be a valuable tool in many industries where it is essential to distinguish misinformation from true information. It also highlights the potential of machine learning in solving complex misinformation detection problems and opens up new perspectives for further research in this area. The program can be used in journalism, education, governmental and non-governmental organizations dealing with the fight against fake news and media manipulation. Thanks to its functionality and accuracy, this program provides users with an effective tool for text analysis, contributing to improving information security and media literacy in society.

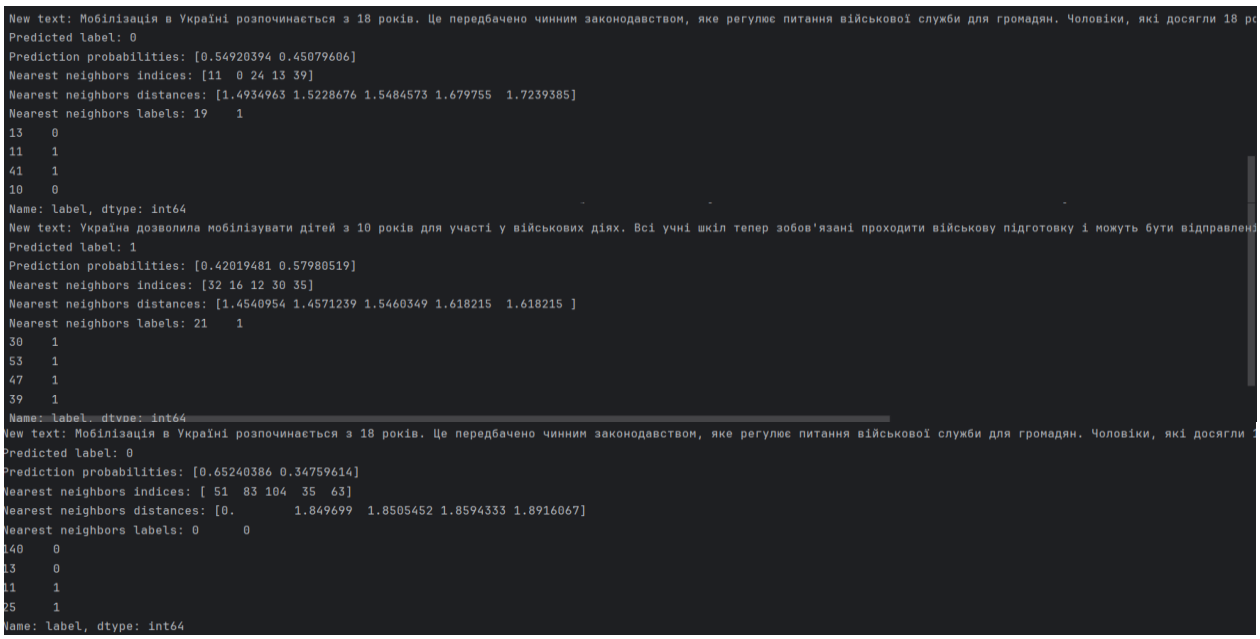
Program execution statistics. Fig. 15 shows results for new text as Text 1 and Text 2.

- For Text 1 "Мобілізація в Україні розпочинається з 18 років. Це передбачено чинним законодавством, яке регулює питання військової служби для громадян. Чоловіки, які досягли 18 років, підлягають військовому обліку і можуть бути призвані на службу." ["Mobilization in Ukraine begins at the age of 18. It is stipulated by the current legislation, which regulates the issue of military service for citizens. Men who have reached the age of 18 are subject to military registration and can be called up for service."]

The predicted label is 0 (true);
Forecast probabilities are [0.54920394, 0.45079606];
Indices of nearest neighbours are [11, 0, 24, 13, 39];
Distances to nearest neighbours are [1.4934963, 1.5228676, 1.5484573, 1.679755, 1.7239385];
The nearest neighbour labels are [1, 0, 0, 0, 1].

- For Text 2 "Україна дозволила мобілізувати дітей з 10 років для участі у військових діях. Всі учні шкіл тепер зобов'язані проходити військову підготовку і можуть бути відправлені на передову." ["Ukraine has allowed the mobilization of children from the age of 10 to participate in military operations. All schoolchildren are now required to undergo military training and can be sent to the front."]

The predicted label is 1 (misinformation);
Forecast probabilities are [0.42019481, 0.57980519];
Indices of nearest neighbours are [32, 16, 12, 30, 35];
Distances to nearest neighbours are [1.4540945, 1.4571239, 1.5460349, 1.618215, 1.618215];
The nearest neighbour labels are [1, 1, 0, 1, 1].



```
New text: Україна дозволила мобілізувати дітей з 554 років для участі у військових діях. Всі учні шкіл тепер зобов'язані проходити військову підготовку і можуть
Predicted label: 1
Prediction probabilities: [0.40281983 0.59718017]
Nearest neighbors indices: [26 52 76 82 51]
Nearest neighbors distances: [1.2956223 1.2956223 1.2956223 1.2956223 1.3899834]
Nearest neighbors labels: 114    1
167    1
94     1
43     1
0      0
Name: label, dtype: int64
```

Fig.31. Results for new text as Text 1 and Text 2

For Text 1"Mobilization in Ukraine begins at 18...", the program correctly identified it as true (label 0). The predicted probability for true information was 54.92%, which indicates a reasonably high confidence in the model.

For Text 2 "Ukraine allowed to mobilize children from the age of 10...", the program correctly classified it as misinformation (label 1). The predicted probability for misinformation was 57.98%, indicating sufficient confidence in the model's decision.

After conducting a statistical analysis, it can be concluded that the program generally works and fulfils its primary task of detecting disinformation. However, sometimes, the program may give false results, which can affect the accuracy of the conclusions. Because of this, we recommend that users looking to check the results obtained use additional methods or compare them with the results of similar programs. To describe the model's effectiveness, we derive a matrix of inconsistencies. With the help of a matrix of inconsistencies, the following metrics can be obtained to evaluate the results of the model:

- Accuracy. The percentage of correctly classified instances is relative to the total number of cases.
- Recall. The percentage of correctly detected positive instances relative to all actual positive instances.
- Positive Predictive Value or Precision. Percentage of correctly classified positive instances relative to all predicted positive instances.
- F-measure. The harmonic means between the sensitivity score and the implied positive value indicator. It is used to evaluate the balance between the sensitivity and accuracy of the model.

Recommendations. After conducting a statistical analysis, it can be concluded that the program generally works and fulfils its primary task of detecting disinformation. However, sometimes, the program can give false results, which can affect the accuracy of the conclusions. Because of this, we recommend that users who want to check the obtained results use additional methods or compare them with the results of other similar programs. Using different programs for comparison will help identify potential errors and provide an extra level of confidence in the correctness of the analysis. Therefore, despite the program's overall effectiveness, users should be careful and pay attention to cases where the results may be inaccurate. Comparison with other sources or programs will allow for more informed decisions and reduce the risk of false conclusions. Fig. 32 shows the nearest neighbour distances for the new text. It helps to understand how close the texts from the training sample are to the latest text to be classified. Fig. 33 shows the main model evaluation metrics: precision, recall, precision and F1-Score. The model demonstrates high accuracy and precision, but recall for misinformation needs improvement.

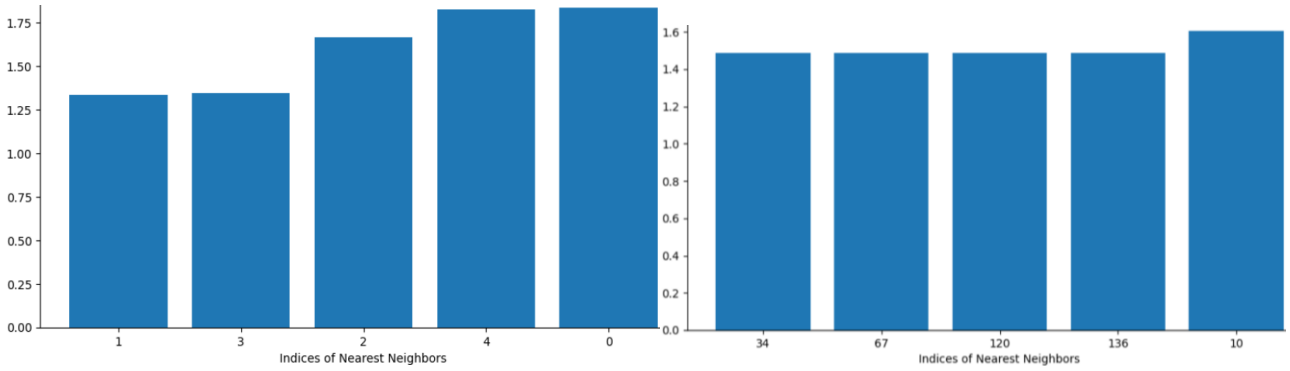


Fig.32. Distances to nearest neighbours for new text

Using other comparison programs will help identify potential errors and provide an additional level of confidence in the correctness of the analysis. Thus, despite the overall effectiveness of the app, users should exercise caution and pay attention to cases where the results obtained may not be accurate. Comparison with other sources or programs will allow you to make more informed decisions and reduce the risk of erroneous conclusions (Fig. 34-36). Analysis of Ukrainian-language fake names by classical classifiers showed much lower results than based on the BERT model (Fig. 33 and Fig. 36). First experiment (Fig. 34):

- Model MultinomialNB implements Naïve Bayes' algorithm for polynomially distributed data and is one of the two classic Naïve Bayesian variants used in text classification (where the data is usually represented as the number of word vectors, although TF-IDF vectors are also known to work well in practice).
- Random Search is a technique that selects random combinations of parameters from a given parameter space and evaluates them to find the most optimal solution. The best alpha for Random Search value is: {'alpha': 0.010597059705970597}. The best accuracy on training data is 0.68666666666666668.

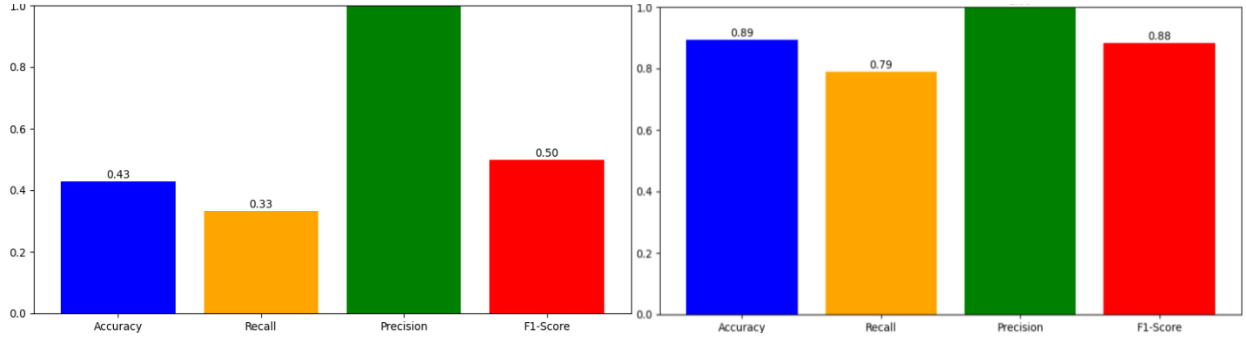


Fig.33. Visualization of results for model metrics

Second experiment (Fig. 35):

- Model GloVe is one of the models for distributed word representation. This model is an unsupervised learning algorithm that derives vector representations of words.
- Random forest is an ensemble machine learning method for classification, regression, and other tasks that works by constructing numerous decision trees during model training and produces a mode for classes (classifications) or an average prediction (regression) of constructed trees. The disadvantage is the tendency to retrain. Best hyperparameters: {'n_estimators': 400, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': 50, 'bootstrap': False}.

	precision	recall	f1-score	support
F	0.38	0.11	0.17	27
T	0.84	0.96	0.90	130
accuracy			0.82	157
macro avg	0.61	0.54	0.53	157
weighted avg	0.76	0.82	0.77	157

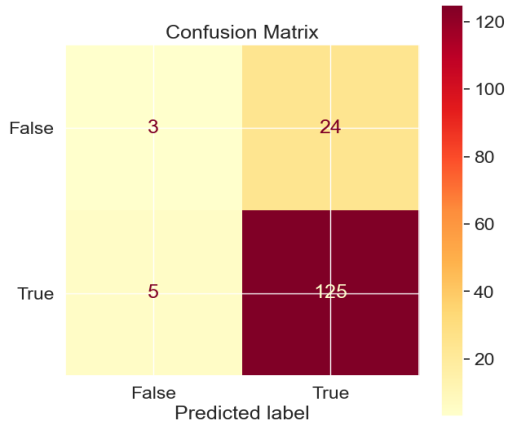


Fig.34. Visualization of results for model metrics

	precision	recall	f1-score	support
F	0.56	0.19	0.28	27
T	0.85	0.97	0.91	130
accuracy			0.83	157
macro avg	0.70	0.58	0.59	157
weighted avg	0.80	0.83	0.80	157

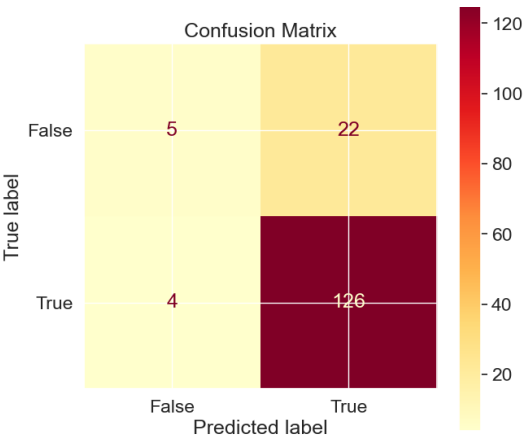


Fig.35. Visualization of results for model metrics

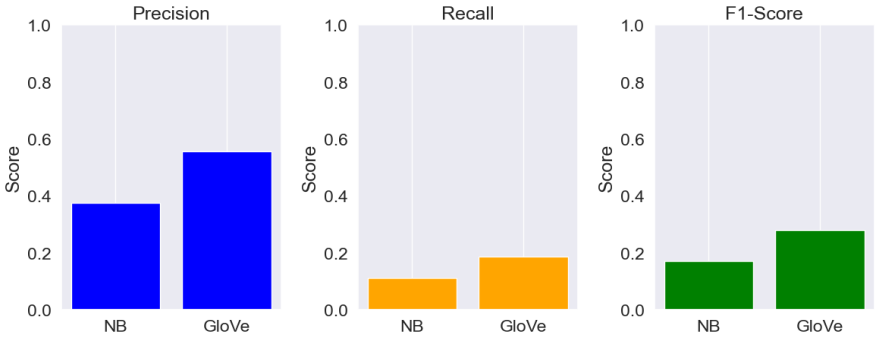


Fig.36. Models' comparison

The model was trained on two datasets of 400 records and 1000 records. Since it was difficult to find the necessary data at the beginning of the study, it was tried to train on a small amount of data and later conduct a comparative analysis of the dependence of the model's quality depending on the dataset's volume.

Since we calculated losses on training data and validation data during training, we can now see how losses have changed with each training epoch. With a larger dataset, the losses decreased gradually, and the difference in value decreased markedly, as can be seen in Fig. 37a. However, on the smaller dataset (Fig. 37b) it can be seen that the value decreased quite a bit, and the loss on the validation data almost does not decrease at all.

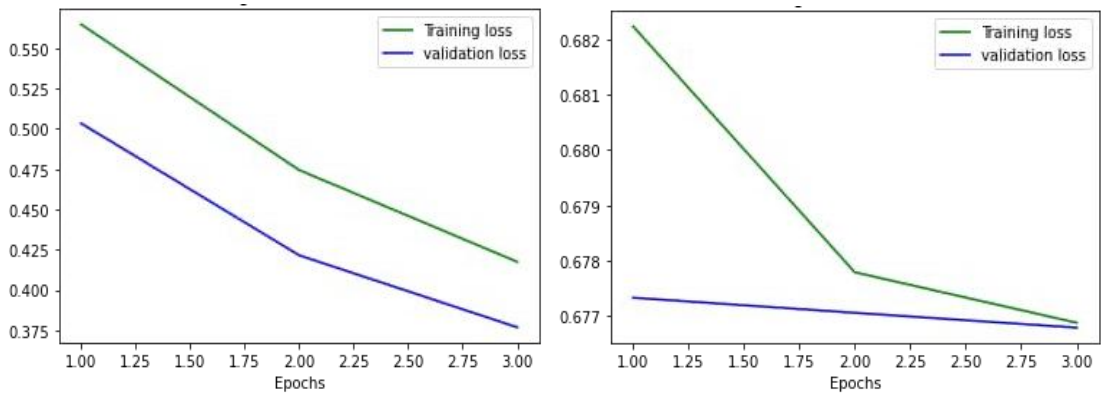


Fig.37. Losses when training a model on 1000 and 400 data quantities

For a model trained on a larger dataset (Fig. 38a), it can be seen that the accuracy of the model for fake news (label - 1) is 0.92, and for true news (label - 0) – 0.79. The sensitivity of the model is 0.78 and 0.92, respectively. So, as you can see, the metrics are very high, which means the model has learned well and can predict correctly.

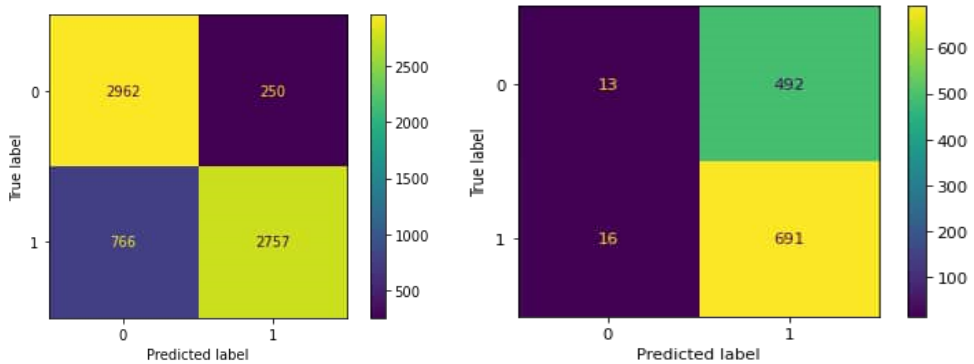


Fig.38. Matrix of inconsistencies (1000 and 400 records)

Для of the model trained on a smaller dataset (Fig. 38b), it can be seen that the accuracy of the model for fake news (label - 1) is 0.58, and for true news (label - 0) - 0.45. The sensitivity of the model is 0.98 and 0.03, respectively. We can see that there was too little data for quality training, and as a result, the model can identify fake news well but does not cope well with true news. Therefore, this model labels all news as false. Also, another option for assessing the quality of classification, the ROC curve, is a graph showing the ratio between the proportion of objects from the total number of trait carriers correctly classified to the total number of non-trait bearers erroneously classified as having a trait. In general,

a good result can be evaluated based on the following characteristics of the ROC curve:

- The larger the area under the curve, the better the model result.
- If the ROC curve deviates from the diagonal, the model performs the classification better than a simple random classification. It can be seen that the first model (Fig. 39a) meets these characteristics, which again proves that it is well-trained. In contrast, we see the second model (Fig. 39b), which acts as a random classification.

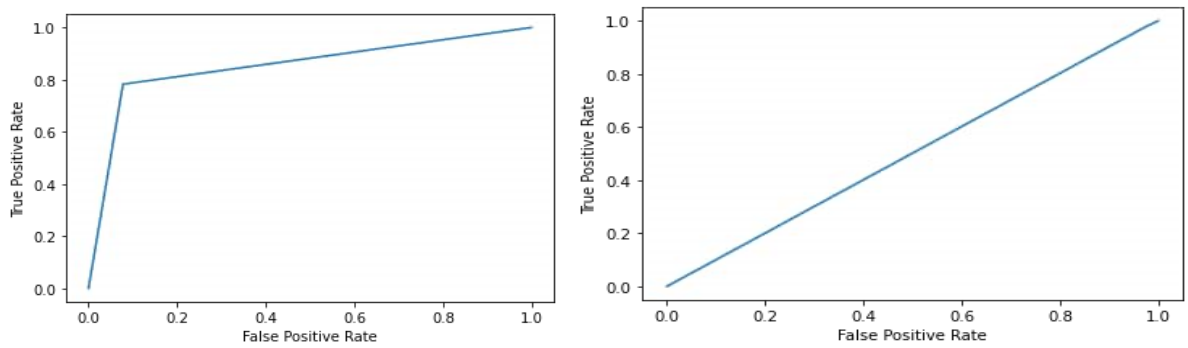


Fig.39. ROC curve (1000 and 400 records)

Comparing the metrics of the model, we can draw the following conclusion. If a model trained on more data performs better metrics (e.g., higher accuracy, sensitivity, or specificity), it can be argued that more data helps improve model performance. It may be due to the greater representativeness and diversity of the data, which contributes to better training of the model.

After conducting a statistical analysis, it can be concluded that the program generally works and fulfils its primary task of detecting disinformation. However, sometimes, the program can give false results, which can affect the accuracy of the conclusions. Because of this, we recommend that users who want to check the obtained results use additional methods or compare them with the results of other similar programs. Using different programs for comparison will help identify potential errors and provide an extra level of confidence in the correctness of the analysis. Therefore, despite the program's overall effectiveness, users should be careful and pay attention to cases where the results may be inaccurate. Comparison with other sources or programs will allow for more informed decisions and reduce the risk of false conclusions.

5. Discussion and Conclusions

In the course of various works, a program was developed to identify disinformation in texts. The program was tested on different datasets containing both true and false information and showed high accuracy in identifying the source of the text. One of the critical aspects of the work was distinguishing between texts containing true information and texts containing misinformation. These differences included style, lexical variety, sentence structure, and other linguistic characteristics. Based on these differences, models were developed to determine whether a text contains misinformation. The work results confirmed the proposed methods' effectiveness but indicated the need for further research. Other techniques like deep learning should be explored to improve the model's performance.

In the future, the program can be used in various fields, such as journalism, education, and the probable detection of fake news, as well as other areas where it is vital to know the text's authenticity. Further research may also focus on the ethical aspects of disinformation detection, privacy issues, and possible workarounds to ensure the safety and reliability of the technology. As for the program's future development, further development involves regular updates to maintain a high level of accuracy and adapt to changes in the methods of creating disinformation. Given the rapid growth of technology and the increasing variety of ways misinformation can be spread, constant updates will be critical to keeping the program relevant. New program versions should include improvements to algorithms, adding new languages, as currently only Ukrainian is available for analysis, and expanding the data set for model training. The first upcoming app update aims to improve usability and design updates for better aesthetics and user experience. This update adds a new feature - a button to clear the entered text. This button will allow users to quickly delete the contents of a text box to enter new text for analysis without using other cleaning methods. It will ensure speed, convenience, and efficiency when working with the program. This update is the first step in modernizing the app to make it more attractive and user-friendly. As a result, the update will help increase the efficiency of working with the program and attract a wider audience. As for more global updates, a web version of the program that will allow users to analyse the text online is envisaged. Such a web version will open the program to a broader audience, providing a convenient and fast way to detect text misinformation. The web version can include an intuitive interface to make the application more accessible for users without technical experience. It will also be able to provide advanced features such as reports and statistics to help users better understand the analysis results.

Thus, future updates and the web version of the application will ensure the tool's relevance and effectiveness and promote its wider use. It, in turn, can contribute to the fight against disinformation, improving the quality of content and raising awareness of the role of disinformation in today's world.

As a result of the work carried out, an effective program for detecting disinformation in texts was created, which was tested on various data sets and demonstrated high accuracy. This program can be a helpful tool for many industries where it is essential to recognize text's authenticity, such as in journalism, education and detecting fake news. The program's future development is planned with an emphasis on regular updates to maintain high accuracy and adapt to the latest technologies. In particular, it is intended to improve algorithms, add new languages for analysis and expand the data set for training models. The work results and further development plans indicate the program's significant potential to contribute to the fight against disinformation and raise awareness of its impact on today's world. Implementing updates and a web version will facilitate the broader use of the program, providing a reliable tool for detecting false information. The program has every chance of becoming an essential tool for improving the quality of content and increasing the level of information security. It, in turn, will contribute to greater public awareness of disinformation and help users better navigate the flow of information.

In times of war, it is essential to consume truthful information. Therefore, my work aimed to develop a neural network to classify news into true and fake.

I apply emotional analysis of the text's tone to achieve the goal. It helps to understand natural language patterns and learn the computer to perceive it at a level close to human perception.

Neural networks are a powerful tool for solving the problem of analyzing text sentiments. Since this industry is actively developing, there are many different options for different tasks, not only sentiment analysis. Despite the choice of neural network training (with or without a teacher), it is imperative to have appropriately crafted inputs. This part is one of the most costly when analyzing the emotional colouring of the text.

After collecting the data, I created the BERT model, which was trained on this data to conduct emotional analysis and identify fake news. The model has been trained and optimized to achieve optimal results. Studies have shown that the BERT model achieves better quality metrics when using a larger dataset for training.

It highlights the importance of having a large dataset when training neural networks. A more significant amount of data provides the model with more information to learn the language's general patterns and features, improving its ability to accurately analyze texts. With a larger dataset, the model can learn more diverse relationships and contexts, which positively impacts its ability to accurately classify and understand text sentiment.

So, the study's results showed that the BERT model shows good efficiency in emotional analysis and fake news detection. Its ability to analyze text and distinguish between true and fake news has become valuable for combating disinformation, especially in the Russia-Ukraine war.

Modern models, even the most powerful ones like GPT and BERT, can effectively analyze language and identify potentially manipulative or false content. But they still achieve complete accuracy. One of the biggest obstacles is the difficulty in determining the context and intentions of the author. BERT can find implicit hints but doesn't always understand sarcasm or satire and sometimes doesn't pick up when lyrics are just personal opinions.

Researchers are trying to improve the accuracy of BERT through a combination of different approaches: language models, user behaviour analysis, and content fact-checking. In addition, mechanisms for source verification and cross-platform analysis are added, allowing you to consider information from various social networks and media.

The task is complicated because there is a difference between outright fake news and those that partially contain the truth presented for manipulative purposes. The developers admit that creating a truly accurate tool takes more years of research and testing. However, according to them, these systems can already help identify the most obvious fakes, and this is a significant step forward in the fight against disinformation. So, while fully automated and accurate recognition of fake news remains out of reach, progress in this area gives reason for optimism.

Acknowledgement

The research was carried out with the grant support of the National Research Fund of Ukraine, "Information system development for automatic detection of misinformation sources and inauthentic behaviour of chat users", project registration number 187/0012 from 1/08/2024 (2023.04/0012). Also, we would like to thank the reviewers for their precise and concise recommendations that improved the presentation of the results obtained.

References

[1] N. Khairova, A. Galassi, F. Lo Scudo, B. Ivasiuk, I. Redozub, “Unsupervised approach for misinformation detection in Russia-Ukraine war news,” CEUR Workshop Proceedings, Vol-3722, 2024, pp. 21-36.

[2] V. Vysotska, D. Shavaiev, M. Greguš, Y. Ushenko, Z. Hu, D. Uhryn, "Information Technology for Gender Voice Recognition Based on Machine Learning Methods", International Journal of Modern Education and Computer Science, Vol.16, No.5, pp. 65-87, 2024.

[3] Victoria Vysotska, Krzysztof Przystupa, Lyubomyr Chyrun, Serhii Vladov, Yuriy Ushenko, Dmytro Uhryn, Zhengbing Hu, "Disinformation, Fakes and Propaganda Identifying Methods in Online Messages Based on NLP and Machine Learning Methods", International Journal of Computer Network and Information Security, Vol.16, No.5, pp.57-85, 2024.

[4] O. Prokipchuk, et. al., "Intelligent Analysis of Ukrainian-language Tweets for Public Opinion Research based on NLP Methods and Machine Learning Technology", International Journal of Modern Education and Computer Science, Vol.15, No.3, pp. 70-93, 2023.
[5] X. Men, V. Y. Mariano, "Explainable Fake News Detection Based on BERT and SHAP Applied to COVID-19", International Journal of Modern Education and Computer Science, Vol.16, No.1, pp. 11-22, 2024.
[6] M. Hartmann, Y. Golovchenko, I. Augenstein, "Mapping (dis-)information flow about the MH17 plane crash,"/ arXiv. https://arxiv.org/abs/1910.01363.
[7] B. Akinyemi, O. Adewusi, A. Oyebade, "An Improved Classification Model for Fake News Detection in Social Media", International Journal of Information Technology and Computer Science, Vol.12, No.1, pp.34-43, 2020.
[8] S. K. Kiran, M. Shashi, K. B. Madhuri, "Multi-stage Transfer Learning for Fake News Detection Using AWD-LSTM Network", International Journal of Information Technology and Computer Science, Vol.14, No.5, pp. 58-69, 2022.
[9] A. S. Noah, N. E. Ghannam, G. A. Elsharawy, A. S. Desuky, "An Intelligent System for Detecting Fake Materials on the Internet", International Journal of Modern Education and Computer Science, Vol.15, No.5, pp. 42-59, 2023.
[10] V. Vysotska, et al., "NLP tool for extracting relevant information from criminal reports or fakes/propaganda content." 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT) (pp. 93-98). IEEE, 2022. DOI: 10.1109/CSIT56902.2022.10000563
[11] Afeez Ayomide Olagunju, Iyabo Olukemi Awoyelu, "Performance Evaluation of Fake News Detection Models", International Journal of Information Technology and Computer Science, Vol.16, No.6, pp.89-100, 2024.
[12] S. Bauskar, V. Badole, P. Jain, M. Chawla, "Natural Language Processing based Hybrid Model for Detecting Fake News Using Content-Based Features and Social Features", International Journal of Information Engineering and Electronic Business, Vol.11, No.4, pp. 1-10, 2019.
[13] S. Mazepa, et al., "Relationships Knowledge Graphs Construction Between Evidence Based on Crime Reports." 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT) (pp. 165-171). IEEE, 2022. DOI: 10.1109/CSIT56902.2022.10000587
[14] A. M. Meligy, H. M. Ibrahim, M. F. Torky, "Identity Verification Mechanism for Detecting Fake Profiles in Online Social Networks," International Journal of Computer Network and Information Security, Vol.9(1), 2017, pp.31-39.
[15] Ogunsuyi Opeyemi J., Adebola K. OJO, "K-Nearest Neighbors Bayesian Approach to False News Detection from Text on Social Media", International Journal of Education and Management Engineering, Vol.12, No.4, pp. 22-32, 2022.
[16] V. A. Oliinyk, et. al., "Propaganda Detection in Text Data Based on NLP and Machine Learning," CEUR Workshop Proceedings, Vol. 2631. 2020, pp. 132-144.
[17] Dharmaraj R. Patil, Rajnikant B. Wagh, Vipul D. Punjabi, Shailendra M. Pardeshi, "Enhanced Phishing URLs Detection using Feature Selection and Machine Learning Approaches", International Journal of Wireless and Microwave Technologies, Vol.14, No.6, pp. 48-67, 2024.
[18] Aya S. Noah, Naglaa E. Ghannam, Gaber A. Elsharawy, Abeer S. Desuky, "An Intelligent System for Detecting Fake Materials on the Internet", International Journal of Modern Education and Computer Science, Vol.15, No.5, pp. 42-59, 2023.
[19] A. Mykytiuk, et. al., "Technology of Fake News Recognition Based on Machine Learning Methods," CEUR Workshop Proceedings, Vol-3387, 2023, pp. 311-330.
[20] I. Afanasieva, N. Golian, V. Golian, A. Khovrat, K. Onyshchenko, "Application of Neural Networks to Identify of Fake News," CEUR Workshop Proceedings, Vol-3396, 2023, pp. 346-358.
[21] A. Wierzbicki, A. Shupta, O. Barmak, "Synthesis of model features for fake news detection using large language models," CEUR Workshop Proceedings, Vol. 3722, 2024, pp. 50-65.
[22] Y. Burov, et al., "Intelligent Network Architecture Development for E-Business Processes Based on Ontological Models", International Journal of Information Engineering and Electronic Business, Vol.16, No.5, pp. 1-54, 2024.
[23] A. Shupta, O. Barmak, A. Wierzbicki, T. Skrypnyk, "An Adaptive Approach to Detecting Fake News Based on Generalized Text Features," CEUR Workshop Proceedings, Vol-3387, 2023, pp. 300-310.
[24] J. Garcia-Marín, A. Calatrava, "The Use of Supervised Learning Algorithms in Political Communication and Media Studies: Locating Frames in the Press," Pamplona, Vol. 31(3), 2018, pp. 175-188. DOI: 10.15581/003.31.3.175-188.
[25] Victoria Vysotska, Andrii Berko, Yevhen Burov, Dmytro Uhryn, Zhengbing Hu, Valentyna Dvorzhak, "Information Technology for the Data Integration in Intelligent Systems of Business Analytics", International Journal of Information Engineering and Electronic Business, Vol.16, No.4, pp. 66-92, 2024.
[26] V. Vysotska, A. Berko, Y. Burov, D. Uhryn, Z. Hu, V. Dvorzhak, "Information Technology for the Data Integration in Intelligent Systems of Business Analytics", International Journal of Information Engineering and Electronic Business, Vol.16, No.4, pp. 66-92, 2024.
[27] C. Bjola "Propaganda in the digital age," Global Affairs, Vol. 3(3), 2017, pp. 189-191. DOI: 10.1080/23340460.2017.1427694.
[28] R. A. Dar, Dr. R. Hashmy, "A Survey on COVID-19 related Fake News Detection using Machine Learning Models," CEUR Workshop Proceedings, Vol-3426, 2023, pp. 36-46.
[29] Propaganda detection. https://www.kaggle.com/datasets/vladimirsydor/propaganda-detection-our-data.
[30] Fake News Detection, https://www.kaggle.com/code/ilaydadu/fake-news-detection-with-nlp-and-lstm.
[31] Fake News Detection, https://www.kaggle.com/code/superrajdoor/fake-news-detection-with-lstm-and-nlp-prorew1/input%20//
[32] Propaganda Definitions. https://propaganda.qcri.org/annotations/definitions.html.
[33] texty.org.ua. How Texty detects and makes sense of manipulative news. https://medium.com/@texty.org.ua/how-texty-detects-and-makes-sense-of-manipulative-news-1f43d33936eb.

Authors' Profiles

Victoria Vysotska is a Professor in the Department of Information Systems and Networks at Lviv Polytechnic National University. In 2023, she completed her Doctoral degree in Technical Sciences, specializing in "Structural, Applied, and Mathematical Linguistics," with a dissertation titled "Analysis and Synthesis of Computational Linguistic Systems for Processing Ukrainian Textual Content." She also holds a PhD in Information Technologies from Lviv Polytechnic National University, awarded in 2014. Victoria has authored over 500 publications, and her primary research interests focus on the identification of disinformation, fake news, and propaganda, as well as detecting disinformation sources and inauthentic behavior (e.g., bots) in coordinated groups through the use of artificial intelligence. Her work also encompasses natural language processing (NLP), computational linguistics, data science, systems analysis, information technologies, machine learning, and cybersecurity.

Dr. Krzysztof Przystupa, Professor of Department of Automation, Lublin University of Technology, Poland. Chief Research Fellow, Vilnius Gediminas Technical University, Sauletekio al. 11, LT-10223 Vilnius, Lithuania. Research Interests: Automation and Control, Electrical Engineering, Electronic Engineering, Mechanical Engineering

Yurii Kulikov is a student at the Department of Computer Science and Information Technology at Lviv Polytechnic National University. He is interested in research in the field of artificial intelligence, in particular, machine learning and deep learning. His interests also include web development.

Sofiia Chyrun is a student at the Telecommunication Department at Lviv Polytechnic National University. Her main research interests are focused on the fake identification, natural language processing, computer linguistics, data science, web technologies, data science, system analysis, information technologies, machine learning, Cisco Systems, information systems and networks, cyber security.

Yuriy Ushenko: Prof., Computer Science Department, Chernivtsi National University, Chernivtsi, Ukraine. Research Interests: Data Mining and Analysis, Computer Vision and Pattern Recognition, Optics & Photonics, Biophysics.

Zhengbing Hu: Prof., Deputy Director, International Center of Informatics and Computer Science, Faculty of Applied Mathematics, National Technical University of Ukraine "Kyiv Polytechnic Institute", Ukraine. Adjunct Professor, School of Computer Science, Hubei University of Technology, China. Visiting Prof., DSc Candidate in National Aviation University (Ukraine) from 2019. Major research interests: Computer Science and Technology Applications, Artificial Intelligence, Network Security, Communications, Data Processing, Cloud Computing, Education Technology.

Dmytro Uhryn graduated from Yuriy Fedkovych Chernivtsi National University, Chernivtsi. Currently, he is a Doctor of Technical Sciences and associate professor at Yuriy Fedkovych Chernivtsi National University. He has currently published more than 140 publications. His research interests are data mining, information technologies for decision support, swarm intelligence systems, and industry-specific geographic information systems.

How to cite this paper: Victoria Vysotska, Krzysztof Przystupa, Yurii Kulikov, Sofiia Chyrun, Yuriy Ushenko, Zhengbing Hu, Dmytro Uhryn, "Recognizing Fakes, Propaganda and Disinformation in Ukrainian Content based on NLP and Machine-learning Technology", International Journal of Computer Network and Information Security(IJCNIS), Vol.17, No.1, pp.92-127, 2025. DOI:10.5815/ijcnis.2025.01.08