I.J. Modern Education and Computer Science, **2025, 2, 15-66**
Published Online on April 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2025.02.02

Agile Intelligent Software Solution for Textual Content Authorship Identification Based on NLP, Artificial Intelligence and Machine Learning

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

Victoria Vysotska
Department of Information Systems and Networks, Lviv Polytechnic National University, Lviv, 79013, Ukraine
Osnabrück University, Osnabrück, 49076, Germany
E-mail: Victoria.A.Vysotska@lpnu.ua, victoria.vysotska@uni-osnabrueck.de
ORCID iD: https://orcid.org/0000-0001-6417-3689

Lyubomyr Chyrun
Ivan Franko National University of Lviv, Lviv, 79000, Ukraine
E-mail: Lyubomyr.Chyrun@lnu.edu.ua
ORCID iD: https://orcid.org/0000-0002-9448-1751

Roman Romanchuk
Department of Information Systems and Networks, Lviv Polytechnic National University, Lviv, 79013, Ukraine
E-mail: roman.v.romanchuk@lpnu.ua
ORCID iD: https://orcid.org/0009-0004-4352-1073

Yuriy Ushenko*
Department of Physics, Shaoxing University, Shaoxing, Zhejiang Province 312000, China
Department of Computer Science, 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

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

Cennuo Hu
Department of Computer Science, College of Science, Purdue University, West Lafayette, IN 47907, USA
E-mail: hu945@purdue.edu
ORCID iD: https://orcid.org/0009-0008-9589-9253

Received: 16 January, 2025; Revised: 22 February, 2025; Accepted: 18 March, 2025; Published: 08 April, 2025

Abstract: The main goal of the work is to create an intelligent system that uses NLP methods and machine learning algorithms to analyse and classify textual content authorship. The following machine learning models for English and Ukrainian publications were tested and trained on the dataset: Support Vector Classifier, Random Forest, Naive Bayes, Logistic Regression and Neuron Networks. For English, the accuracy of the models was higher due to the more significant amount of text data available. The results for English fiction publication show that the Neuron Networks classifier outperforms the other models in all evaluated metrics, achieving the highest accuracy (0.97), recall (0.96), F1 score (0.98), and precision (0.96). It shows that Neuron Networks is particularly effective in capturing distinctive features of the writing styles of different English authors in scientific and technical texts. For the Ukrainian language,

there is a drop in accuracy by 5-10% due to the smaller number of corpora of texts for teaching. The results for scientific and technical Ukrainian publications show that the Random Forest classifier outperforms the other models in all evaluated metrics, achieving the highest accuracy (0.88), recall (0.87), F1 score (0.87), and precision (0.87). It shows that Random Forest is particularly effective in capturing distinctive features of the writing styles of different Ukrainian authors in scientific and technical texts. Much worse accuracy results were shown by other models such as Support Vector Classifier (77%), Logistic Regression (73%) and Naive Bayes (70%). The results for the Ukrainian fiction publication show that the Random Forest classifier outperforms the other models in all evaluated metrics, achieving the highest accuracy (0.85), recall (0.84), F1 score (0.84), and precision (0.84). Much worse accuracy results were shown by other models such as Support Vector Classifier (77%), Logistic Regression (73%) and Naive Bayes (70%)

Index Terms: Author's Style, Machine Learning, Authorship Identification, Stylometry, NLP, Artificial Intelligence, Information Technology

1. Introduction

Identifying the author of a scientific or technical publication based on a writing style is a challenging and vital area of computational linguistics and data science [1-3]. As the volume of scientific literature grows exponentially, author identification tools and methods are becoming increasingly important to maintain research integrity and detect plagiarism [6-9]. This paper explores the intersection of machine learning (ML) and natural language processing (NLP) to develop a Python program capable of identifying the authorship of scientific and technical publications through stylistic analysis [10-12]. The identification of the author's style, often called "stylometry", involves the analysis of various linguistic features characteristic of an individual's writing [15-18]. These features can range from simple metrics such as word frequency and sentence length to more complex attributes such as syntactic structures, use of specific jargon, and unique punctuation patterns [19-21]. The goal is to identify subtle but consistent differences in writing that are distinctive enough to distinguish one author from another [1]. In scientific and technical publications, the problem is compounded by the formal and often rigid structure of writing, which can mask individual stylistic tendencies [22-25]. Unlike literary works, where authors have more freedom to express their unique voices, academic papers follow strict guidelines that prioritise clarity, precision, and objectivity. Despite these limitations, authors often retain stylistic imprints in their choice of vocabulary, sentence construction, and structure of their arguments and present data. Identifying the author's style is relevant and essential because it plays a significant role in academic integrity, plagiarism detection and intellectual property protection [26-29]. In addition, developing a practical algorithm for identifying the author's style can contribute to determining authorship in historical documents, improving natural language processing algorithms, and improving personalised content recommendations to users on the Internet. The main goal of the work is to create an intelligent system that uses NLP methods and ML algorithms to analyse and classify the authorship of scientific texts. As a result, the developed program should be able to reliably attribute authorship to scientific and technical texts based on stylistic features. This work contributes to computational linguistics and provides practical knowledge about the application of machine learning in text analysis. On a practical level, the development of effective tools for analysing authorship can contribute to the fight against literary piracy, a significant problem in modern education and computer science. Such tools allow publishing houses, libraries and educational institutes to protect writers' rights and promote the literary market's development. The prospects of developing a product for text analysis for the original authorship of works of literature, books, student works, or science and mass media articles open up new opportunities for developing the science of literature, cultural preservation, education and technologies. This product can have a significant impact in several key areas:

- **Scientific research.** The product can become an essential tool for academic research, allowing scholars to analyse literary texts more deeply and identify stylistic features of different authors. It can contribute to a better understanding of literary trends and the evolution of language and help resolve academic disputes about the authorship of works.
- **Education.** In education, such a product can teach students to analyse literary works, develop critical thinking and improve literacy. Text analysis tools can become part of the learning process in Ukrainian literature classes, allowing students to experiment and learn through hands-on experience.
- **Protection of intellectual property.** In the context of the growing challenges of plagiarism and copyright, development can be essential for publishers, legal authorities and independent authors to protect their intellectual property. The system can help identify and document cases of copyright infringement, as well as provide an evidentiary basis in legal cases.
- **Expansion of technological capabilities.** Technological innovations based on artificial intelligence (AI) and ML continue to develop, and such a product can contribute to the further development of NLP and deep learning algorithms. It opens the door to new tools capable of analysing text at a much deeper level, including emotional content, subtext, and contextual meanings.

- **Cultural preservation.** The product can be necessary for the preservation of Ukrainian culture, especially in the context of modern globalization and cultural homogenization. Preserving unique linguistic and stylistic features through accurate authorship analysis will help future generations understand and appreciate the contribution of previous generations of writers.

- **Development at the international level.** Since Ukrainian literature is becoming increasingly famous and popular abroad, there is a significant potential for using such a product globally, especially in areas where it is necessary to determine the authorship of translations or verify the text within the framework of literary studies. Such a tool can contribute to greater visibility of Ukrainian literature and its analysis in world academic circles.

- **Analysis of fakes, propaganda and disinformation** to identify the sources of disinformation and inauthentic behaviour of users of social media chats.

- **Forensics**. Forensic auto research is a sub-branch of the forensic study of writing, which studies the patterns of the formation of written speech and develops, on their basis, methods for identifying a particular author or his data (gender, age, education, profession, etc.). The subject of forensic auto investigation is the establishment of factual data on the identity of the document's author. The direct object of the expert research is written speech and general (lexical and phraseological, syntactic, stylistic, spelling, punctuation) and individual (persistent speech disorders, use of specific language means, etc.) language skills that are manifested in it.

Written speech is understood as a set of linguistic means characteristic of each person who writes for the written expression of their thoughts, expression of human thoughts with the help of various linguistic means (syntax, spelling, vocabulary, punctuation, stylistics). Signs of written speech are divided into general and separate. General features characterize written speech as a whole (possessing punctuation, spelling, lexical and phraseological, syntactic, stylistic and other skills). The general ones include the following: the degree of proficiency in written speech (high, medium, low), the general level of literacy, the degree of development of linguistic skills, and the degree of development of stylistic skills. Certain features of written speech characterize individual aspects, elements of written speech of a particular person, manifested in stable lexical, stylistic features and grammatical errors. Such features are the basis for identifying the author of a specific text. Individual signs include spelling and stylistic errors, peculiar execution of punctuation marks, construction of sentences, formal and logical skills of written speech, in particular, intellectual skills of perception surrounding reality, argumentation, evaluation, accentuation, etc.

Based on the current state of development of a software product for text analysis for the original authorship of works of Ukrainian literature, it can be concluded that this project is vital for both the academic community and the cultural sphere. The use of advanced technologies of NLP and ML allows for an increase in the accuracy and speed of text analysis, which opens up new opportunities for research in literary studies and the fight against plagiarism. Convenient functionality and the availability of a specialized Ukrainian dataset ensure the availability and effectiveness of the product for a wide range of users, including researchers, educators and law enforcement agencies. It contributes to the in-depth analysis of literary works and the preservation of the national cultural heritage. However, the project has some drawbacks, such as limited data and high requirements for technical resources, which may limit its application in certain conditions. It is also important to consider ethical and legal issues related to the use of literary works.

2. Related Works

2.1 Main tasks and applications of the science of authorship determination

The art and science of distinguishing authors by the author's style of writing by identifying the characteristics of the authors' personalities based on the study of the author's articles is called authorship analysis [1-3]. The author's style is the author's writing style, which is subconsciously used by the author when writing texts. Stylometry is the study of language style, usually written language, based on the fact that each author has irreplaceable written features that cannot be imitated. The main tasks of determining authorship are [1-6]:

1. Author Attribution or Identification is the task of finding the most likely author of an article, text or document. In terms of machine learning, this is a multi-classification problem designed to answer the question of which of a given set of possible authors wrote a given text.
2. Author Verification answers the question of whether a particular author wrote a given text or not. In fact, this is a binary classification problem that is solved in order to determine how high the probability of writing a given text by a particular author is.
3. The definition of Author Profiling is the identification of a person's parameters (for example, gender, age, native language, cognitive psychological characteristics, socio-cultural level) by studying the "available information" in their texts. It is a problem of multi-class classification, as well as clustering.
4. Plagiarism detection or Similarity detection between two texts is the task of determining the similarity between two texts, and not necessarily with the definition and indication of authorship.
5. Identification of stylistic inconsistencies – refers to the search for differences in writing a joint work.

Despite the fact that determining authorship is a highly specialized task, it has a number of practical applications:

- Civil law (resolution of disputes regarding the authorship of works, program code);
- Intelligence (e.g., assigning messages to known terrorists);
- Criminal law (for example, identifying the authors of abusive or threatening messages);
- Computer forensics (fake social media profiles, fake bot reviews, malware developers;
- Literary research (for example, attributing anonymous literary works to famous authors);
- Consumer market research. Authorship analysis tasks help to create a consumer profile, identify fake reviews, and conduct customer segmentation);
- Mental health examination.

Identification of authorship of scientific and technical publications by writing style faces several serious challenges and limitations [7-12]. One of the main problems is the availability of a sufficient number of texts for analysis, as well as their quality [9-15]. The scientific style differs from other writing styles, which creates difficulties for algorithms for analyzing authorship [12-18]. Scholarly texts often go through several stages of editing or even translation, which can distort the original style [15-21]. Let's consider them in more detail (Table 1).

Table 1. Main potential limitations or challenges for identifying the authorship of scientific and technical publications by writing style [21-30]

N	Name	Characteristic	Explanation
1	Availability and quality of data for analysis	Limited access to scientific publications	- Many scientific journals are paid or closed, which makes it challenging to collect a corpus of texts for analysis. - Open sources, such as arXiv or DOAJ, may not be sufficient to compare the author's style accurately. - Authors can be published in various journals that have different rules of style and formatting.
		Lack of previous texts by a particular author	- For practical analysis, you need to have a sufficient number of texts written by the author. - For practical analysis, you need to have a sufficient number of texts written by the author.
		Co-authorship and collective writing of articles	- Most scientific and technical articles are co-authored, which blurs the individual style. - It is often difficult to determine which co-author wrote a specific part of the article. - In many cases, texts are edited by several people or reviewers, which changes the writing style.
2	The inherent complexity of the scientific writing style	Formal and standardized style	- Scientific publications use a standardized style that includes technical terms, structured sentences, and formal expressions. - Due to standardization, texts can be similar between different authors, which makes their attribution difficult. - Template phrases are often used ("In this study, we consider...", "The results demonstrate..."), which reduces the uniqueness of the style.
		Use of technical and mathematical language	- Many texts contain a large number of mathematical expressions, formulas, tables and graphs, which are not taken into account during the analysis of style. - Specific terminology is often dictated by the field of study rather than the individual style of the author.
		Use of standard bibliographic sources	- Scholars often cite the same sources, which can create false matches between texts by different authors. - Automated systems may perceive the bibliography as part of the author's style, even though it is standardized.
3	Impact of editorial editing and translation	The intervention of editors and reviewers	- Editors and reviewers often change the wording, paragraphs, and overall structure of an article. - Edits can unify the style, making it difficult to determine authorship.
		Using templates for writing articles	- Many scientific journals have strict requirements for the structure of an article, which reduces the variability of style. - For example, the introduction, methodology, results, and conclusions often have a very similar structure.
		English publication and machine translation	- For an international audience, many authors write in English even though their native language is another. - Using translators or editors can change the author's style, making it less unique. - If the article was written in another language and translated, the style may differ significantly from the author's other works.
4	Limitations of authorship analysis algorithms	Using stylometric methods that do not take context into account	- Most style analysis algorithms are based on word frequency, sentence length, and other statistical parameters. - They may not take into account semantic context or scientific terminology, which makes the analysis less accurate.
		False positives and negatives	- Unifying style through editing can lead to the system incorrectly identifying the author. - If two researchers work in the same field, their style may be similar, creating false positives.
		Difficulty in processing multilingual texts	- If authors publish in different languages, the algorithm may mistakenly perceive their texts as written by various people. - Translation may change the frequency characteristics of word usage and sentence structure.

Identification of authorship of scientific and technical publications is a difficult task due to a number of factors [15-21]:

- Limited access to data and lack of texts from a single author.
- Co-authorship and editorial editing influences that change individual style.
- A formalized and standardized style of scientific writing that reduces variability.
- A formalized and standardized style of scientific writing that reduces variability.
- Using translation and editing can distort the author's style.

Despite these challenges, style analysis can still be helpful when combined with other methods (e.g., metadata analysis, citations, and meaningful text analysis) to improve the accuracy of authorship.

2.2 The history of the science of stylometry

For the first time, the study of the style of the text for attribution was made in the XV century. Italian philologist Lorenzo Valla published a treatise, "Considerations on the falsity of the so-called Constantine's deed of gift", in which, on the basis of various, including stylistic criteria, it was proved that this text was a forgery.

The history of modern statistical stylistics began in the middle of the XIX century when the English mathematician Augustus de Morgan in 1851 suggested that different authors could be determined using hidden statistical characteristics. Considering the problems of Greek prose, Morgan argued that the average length of words in the author's work can be a characteristic feature of the author's style. However, as far as we know, de Morgan himself did not make any calculations.

In the middle of the XIX century, there was also a group of scientists developing the so-called method of "stylemetrics" (F.G. Friari, J.K. Ingram, F.W. Furnival). They counted the number of repetitions of a particular word and the change in size in verse. The main result of their work was the discovery of a slow but steady change in Shakespeare's style for 22 years. The term "stylemetry" was invented by the German philologist Wilhelm Dittenberger (1880), who attempted to solve the problem of attribution and chronology of Plato's dialogues. He investigated the frequency of use of words, especially service words, in Plato's texts, the implementation of which does not depend on the subject of the text. Later, his research on various materials was continued by E. Zeller (1887), F. Chadoux (1901), and C. Ritter (1903).

In the 20s Twentieth century, only a few serious researchers on style statistics can be named, such as R.E. Parker (1925), Z.E. Chandler (1928), M. Perry (1928), and especially A. Busman (1925), the author of the so-called verb-adjective ratio. In the 30s Twentieth century, a new step was taken in the application of statistical methods in style by linguists such as J. S. Miller. W. Fletcher (1934), who considered the development of Spencer's style, G.M. Bolling (1937), with a critical essay on the statistical study of Homer's language, J. S. Spencer. B. Carroll (1938), who raised the problem of vocabulary diversity, and W.G. Yule (1938), the first to investigate the distribution of sentence length as a statistical characteristic of style. It is from him that the application of modern statistical methods in style begins. Since this period, the application of statistical methods in style research has spread throughout the world. Interest in statistical linguistics increased sharply, especially in the 1960s-70s. (J. B. Carroll, G. Herdan, H. H. Somers, C. Muller, B. Kelman, L. T. Mylik, J. Mitrick, L. Dolezela, K. B. Williams, B. N. Golovin, J. Kraus, M. N. Kozhina, etc.). It was during this period that various ideas for analyzing the author's style arose and developed. With the advent of computers in the 60s. Twentieth century. It became possible to use them in linguistic research and, therefore, new methods for solving the problem of authorship.

2.3 Overview of the main approaches

The task of author attribution of texts arose a long time ago. People copy other people's works due to their ignorance or to achieve valuable goals. With the development of society, new problems began to arise. In particular, it became clear that not only money or other material possessions are the property of a person but also the intellectual achievement of a person. In particular, his works or other textual information is his property [30-37]. Also, one of the reasons for this problem is the historical aspect of such research. In particular, the determination of the era of writing of this or that work, the number of people who wrote it, etc. It is necessary to find out to whom exactly. Based on the results of experimental verification of the method proposed by D.V. Khmelov in his work [50], it is claimed that this is possible with a specific, reasonably high degree of probability. The tradition of a formal approach to the search for methods of determining authorship should be counted from the work [38]. Markov [31] responded to this work, which testifies to the interest of the creator of the Markov chain apparatus in this topic. We should also note that the first application of his method was described by Markov in the work [32], devoted to the analysis of the distribution of vowels and consonants among the first 20,000 letters of "Eugene Onegin". In computer lexicography [21], an electronic dictionary is "a dictionary whose compilation procedures are carried out with the help of a computer"[27]. Undoubtedly, electronic dictionaries have many advantages over printed counterparts [37-51]:

1. Storing a large amount of information: Electronic dictionaries can include different types of dictionaries and have hyperlinks, which allows you to store much more information compared to the limited amount of printed publications.
2. Convenient search system: Electronic dictionaries have an effective search system that allows you to quickly find the necessary information in the entire dictionary, conduct a simultaneous search in several dictionaries and speed up the search process.
3. Use of multimedia: Electronic dictionaries can use multimedia tools, such as title voiceovers, illustrations, photos, animations and videos, which contribute to better understanding and semanticisation of vocabulary.
4. Network availability: Electronic dictionaries can be used locally and in global networks, where many users can work with them simultaneously, ensuring availability and sharing.
5. Saving time and resources: when compiling computer dictionaries, you can save time and material costs since editing, updating and searching for information in electronic dictionaries are faster and more efficient.

These advantages make electronic dictionaries essential for language research, learning and use [52-60]. The current state of authorship determination methods is reflected in [39], and a good overview of foreign authors is given in [54].

Lexicography (from the Greek graph as written and lexicon is a dictionary) studies compiling various language dictionaries with the theory and practice [42]. Theoretical lexicography studies the dictionary macrostructure and microstructure development, the dictionary typology development, and the history of the development of lexicography. The dictionary macrostructure determines the main aspects, such as vocabulary selection, vocabulary volume and nature, and the material's organising principles. It defines the general structure of the dictionary and its internal logical order. The microstructure of the dictionary defines the detailed structure of the dictionary article. Such a structure includes types of definitions and relationships between different types of word information and types of language illustrations. The microstructure determines which elements are contained in each dictionary entry and how they are organised. Macrostructure and microstructure are interrelated and jointly affect the functionality and effectiveness of the dictionary. They help users find the information they need quickly and efficiently by providing logical and organised access to vocabulary material. Practical lexicography can be defined as the process of compiling dictionaries of various types based on theoretical developments [45]. Lexicography is a) the collecting words process of a specific language, arranging them, and describing vocabulary material [44]; b) the linguistics branch, the subject of which is the compiling dictionaries theory and practice, dictionary studies types [58,59]; c) a specific language and scientific works dictionaries set in this field [59]. So, lexicography, or lexicography, is a linguistics branch that focuses on creating dictionaries and their theoretical foundations study. This direction includes practical lexicography, which deals with the specific arrangement of dictionaries, and theoretical lexicography, which examines the general principles and concepts underlying dictionary works. Lexicography arose from the need to explain obscure words, initially by placing explanations in the margins and the text of manuscript books. According to Yushchuk I. [48], in modern understanding, a dictionary is a collection of words (sometimes word combinations) that are systematised and ordered with appropriate explanations according to their purpose. Dictionaries are a systematically arranged collection of linguistic forms used in the speech practice of a specific language community. They contain descriptions of the meanings of individual forms and provide information about their functioning according to the specifics of the community. The organisation of vocabulary units and the amount of information supplied depend on the type and purpose of the lexicographic work. Dictionaries can explain the meaning of words, reveal their origins, provide guidance on correct spelling or pronunciation, and translate them into other languages. Some dictionaries also provide encyclopedias about subjects, phenomena, scientific concepts and other word-related aspects. Dictionaries can vary in size, from short (up to 30,000 words) to medium (70,000–80,000 words) or full (over 80,000 words). Compiling and publishing dictionaries is an essential social activity and one of the signs of society's progress. Dictionaries perform informative and normative functions, contributing to language understanding and use [48]. Dictionary development stages:

1. A requirements system development related to the purpose and range of users;
2. A requirements system development related to such parameters of the dictionary as description units, volume, structure, and type dictionary information;
3. Texts selection, context description, grammatical forms characteristics, preliminary dictionaries compilation;
4. Texts distributive analysis and tests with native speakers;
5. Experimental data generalisation;
6. Definitions construction at the appropriate level of metalanguage and their verification in the new experiments course;
7. Additional information collection and systematisation about each language unit;
8. Designing dictionary articles;
9. Dictionary articles system analysis and arrangement;
10. Dictionary designing [47].

Currently, lexicography is under the strong influence of new methods of processing information. The change in tools led to new vocabulary technologies (modern information technology of lexicography), such as computer lexicography. A significant part of intellectual operations is classified as routine. At the same time, there is a transition process for lexicographers who learn new professions, move away from "pure" lexicographic activity, and begin to engage in publishing activities or organisation of lexicographic research and publication of their results. On the other hand, some specialists in the field of informatics are actively involved in lexicographic activities. Computer lexicography is an applied scientific discipline in linguistics that investigates methods, technology and specific approaches to computer technology in the theory and practice of compiling dictionaries. It includes various methods and software tools for processing text information to create dictionaries. Computer lexicography provides processes support and automation of updating, compiling, searching and using dictionary information with computer technologies [31].

In the papers [61-66], the authors investigated the effectiveness of the Support Vector Classifier (SVC), Random Forest, Naive Bayes, and Logistic Regression algorithms for identifying text authorship by writing style. SVC and Random Forest have been shown to achieve high accuracy (about 85-97%) when classifying authorship on large datasets. However, issues related to the limited interpretation of the Random Forest and SVC models, as well as their sensitivity to high-dimensional data, remain unresolved. The reason for this may be the objective difficulties associated with the processing of a large number of stylistic features and the need to balance between the accuracy and interpretation of the models.

An option to overcome the relevant difficulties can be the use of ensemble methods or a combination of several algorithms to improve the accuracy and interpretation of the results. It is the approach used in the work [67], where the authors combined the results of several models to achieve better performance. However, even when using ensemble methods, the problem of choosing optimal parameters and computational complexity of models remains.

All this gives grounds to assert that it is expedient to conduct a study dedicated to the development of hybrid approaches that combine traditional linguistic features with modern methods of machine learning to improve the accuracy and interpretation of models for identifying the authorship of the text.

In the paper [68], the authors reviewed modern methods for identifying text authorship, focusing on machine learning and deep learning algorithms. The use of deep learning models, such as recurrent neural networks and transformers, has been shown to achieve high accuracy in the attribution of authorship. However, issues related to the need for large amounts of data to train such models and their computational complexity remain unresolved. The reason for this may be objective difficulties associated with the limited availability of large corpora of texts by well-known authors and high requirements for computing resources, which makes the relevant research costly.

An option to overcome these difficulties may be the use of transfer learning methods, which allow adapting models trained on large general corpora to specific tasks of authorship identification with smaller amounts of data. It is the approach used in the work [68-69], where the authors applied pre-trained models, such as BERT, for the attribution problem of authorship. However, even when using transfer learning, the issue of adapting models to specific stylistic features of individual authors is also a problem.

All this gives grounds to assert that it is advisable to conduct a study dedicated to the development of hybrid methods of authorship identification, which combine the advantages of deep learning and traditional linguistic approaches, which will increase the accuracy of attribution while reducing the need for large amounts of data and computing resources.

2.4 Ukrainian lexicography development

The formation of lexicography in Ukraine took place by considering specific conditions. It was related to the political situation in Ukraine, the people's social and cultural needs, and the state of the literary language. There were other needs for lexicographic publications in different historical periods in Ukraine. For example, during the national liberation struggle of Ukrainians in the 19th century, there was a great need for dictionaries to help Ukrainians learn their language. During the Soviet occupation of Ukraine, there was a great need for dictionaries to help Ukrainians learn Russian. Today, there is an excellent need in Ukraine for dictionaries to help Ukrainians learn their language and culture. It is necessary because reviving the Ukrainian language and culture is taking place in Ukraine.

2.5 Ancient Rus' period

On the territory of Ukraine, education began its development even during the existence of Kyivan Rus in the XI-XIII centuries. Much attention was paid to language during this period since language and linguistics became the basis for other disciplines' development. Dictionaries originated in Ukraine in the 13th century. The main centre of writing became the Kyiv-Pechersk Lavra. There, lists of obscure words borrowed from Church Slavonic books are published. However, the education development, including lexicography, was stopped due to the Tatars' invasion of the territory of Ukraine and the destruction of Kyiv. The cultural centre moved to Halychyna (Lviv). Unfortunately, no lexicographical works have survived from this period. These events significantly influenced the development of lexicography in Ukraine. Still, historical changes did not stop the process of formation and development of Ukrainian vocabulary, which continued in other periods of history.

2.6 The lexicographic practice of the period of the new literary language

Starting from the end of the 18th century, a new stage in the development of Ukrainian lexicography began, caused by a significant historical event - the annexation of the right-bank Ukrainian lands to Russia. At this stage, the work of the dictionary focused on the new Ukrainian literary language. The lexicographers aimed to collect as much Ukrainian vocabulary as possible. Based on the created lists and registers, the first Ukrainian-Russian dictionaries began to be compiled. The first complete Ukrainian dictionary also appeared. In the 1930s, lexicographic work spread to western Ukraine, where German-Ukrainian and Polish-Ukrainian dictionaries were developed. This period can be considered a productive and significant stage in forming Ukrainian lexicography. However, the further development of Ukrainian lexicography again became dependent on that time's political and social conditions. From the middle of the 19th century until 1917, Ukraine was under the oppression of the tsarist regime. The development of Ukrainian culture and science was stopped, and even if dictionaries appeared in the Ukrainian language, it was often impossible to publish them.

2.7 Modern lexicography

The end of the 20th century was a period of significant changes for Ukraine as the country became an independent state. It led to the consolidation of the Ukrainian language as the only state language of Ukraine. In the context of modernity, Ukrainian lexicography develops without national restrictions and becomes more diverse, aimed at meeting the needs of social life. Thanks to independence and societal changes, Ukrainian lexicographers and linguists gained great freedom in developing and publishing dictionaries. Modern Ukrainian lexicography covers various fields and specialised areas, considering modern society's needs. It actively uses the latest technologies and information resources to create high-quality and accessible dictionary publications. Dictionaries published in modern times consider new words, terms and phraseology that have appeared in the Ukrainian language. They also reflect societal changes and cultural and technological advances. Ukrainian lexicography aims to provide users with high-quality and relevant lexicographic information that meets the modern needs of society [53].

2.8 An Overview of the Most Known Authorship Issues

Before presenting the issue's history, it is necessary to pay attention to the terminology used. In particular works related to the definition of authorship, the broad concept of attribution of the text is used [21]. The attribution of texts is a complex and multifaceted process that includes the analysis of historical, cultural, stylistic, and linguistic aspects [21]. It can use comparative literary examination, statistical analysis, computer linguistics, and other approaches to establish authorship or other attributes of the text [21]. There are several well-known examples of uncertainty of authorship. In particular, among the widely known examples of disputed authorship, we can mention the "Sholokhov question", which has been discussed in recent decades - how the epic novel "Silent Don" was written by Sholokhov or by Sholokhov using the texts of another author (or other authors), for example, Kryukov. Or, for example, whether "A Novel with Cocaine" was written by Ageev (a personality who has not revealed himself in any way in the literary world) or perhaps by V. Nabokov [41].

Also known as the "Homeric question" [25], it is a complex of problems concerning the authorship of the epic works "Iliad" and "Odyssey". Issues related to both the presence of the historical figure of Homer and possible co-authorship in these works are investigated. "Shakespearean question" - the problem of the authorship of the corpus of texts attributed to the authorship of William Shakespeare, discussed by some researchers. This problem was considered and is being considered in many scientific works. There are whole theories about the origin of these works. For non-professionals, the most natural way to identify the author's features is by recording the external details of the author's style, characteristics of this or that person and, in particular, favourite words, terms, and phraseological turns and sayings. This technique is sometimes used even today. However, the choice of such details is inevitably subjective and, in addition, does not guarantee infallibility in the case of using the external information of the author's language. In addition, not all authors express themselves in this way. After all, there are often no pronounced words and phrases in the author's language. In this case, all hopes are connected with some information from the text itself, which can provide information about the author's political, ideological, aesthetic, religious and other views of the text or some information about his life path, details of his biography, etc. Sometimes, it serves as excellent material for text attribution. Unfortunately, not all anonymous texts contain such information. In the case of plagiarism, such details may be deliberately added by the plagiarist. Thus, identifying subconscious features of the author's language is an essential step in text attribution. It means identifying unique stylistic, grammatical, lexical, syntactic and semantic features that characterise the author's writing style. Such features are tried to be detected using non-traditional formal and quantitative methods.

Morozov N. A. took the first tentative steps on this path at the beginning of the 20th century [38] (However, Markov [31] judiciously criticised Morozov's approach). The need to search for new ways and to abandon the "subjective-attributive" practice became most clearly felt in the 1950s and 1960s of the 20th century. At the beginning of the 1960s, prominent Russian philologists V.V. Vinogradov and D.S. Likhachev almost simultaneously noted the fading trend of the traditional method. At this time, the first work developing a quantitative direction in studying language and language style began to appear. In the 1960s and 1970s, the number of studies that developed statistical methods for vocabulary and grammar gradually increased. Attribution methods, which focus on analysing syntactic structures of language, have made significant progress in recent years. The author [39] singles out two directions here.

One is related to constructing and analysing graphs of syntactic connections within the framework of typical phrases and sentences (the most important here are the works of I.P. Sevbo [40]).

In the second direction of research on attribution, attention is paid to identifying dependencies and regularities in the relationships between different syntactic structures. The problem is that researchers working in this direction cannot find any parameter or characteristic characterising the author's style. "In modern literature devoted to the attribution of anonymous and pseudonymous works, although the task of minimising parameters as material for processing and study is set, the tendency to maximise texts and their parameters and characteristics prevails (in the most successful works). Moreover, along lexical, morphological, grammatical, etc. parameters can also be used with higher level parameters (syntactic constructions) "[39]. From this point of view, M.A. Marusenko's monograph on determining the authorship of several pseudonymous works attributed to the early V. V. Mayakovsky [33] is a particularly revealing work of this plan. In this work, 56 main parameters of the text are used, along with the same number of derived (aggregated) indicators.

In 1984, the work of a group of Norwegian and Swedish scientists was published, which was devoted to one of the most acute problems of literary studies of the 20th century and, at times, took on an acute political character [29]. We are talking about many years of periodic outbursts of suspicion about the authorship of a famous work of Russian literature of the 20th century. - "Silent Don". The book [29] contains several works that formally substantiate the authorship of M. Sholokhov and refute any relation of F. Kryukov to the first parts of "Silent Don". The authors study the distribution of word classes, the use of some combinations of grammatical classes, the length of the sentence, the length of the words, the vocabulary profile of the text and the so-called saturation of the dictionary. By many parameters, they repeatedly confirm Sholokhov's authorship. Despite the optimism of the mentioned authors, the transition of researchers from one parameter to another (or even a complete review of these parameters) indicates a low level of development of methods for determining authorship rather than a significant success. A common weakness of these studies is that they have never been conducted on a large number of authors. Meanwhile, suppose the method claims to be objective in recognising authorship. In that case, it should be tested on at least dozens of authors, and only after such justification should it be used to investigate cases of disputed authorship. Otherwise, no arguments about using "sound assumptions" should be taken on faith. When sorting out many parameters, all the methods highlighted in [39] were limited to a few studied authors. In the review [39], only one verification of the techniques used by two authors is mentioned: M. Marusenko verified his methodology on the stories of A.I. Kuprin and I.A. Bunin. Although all these methods claim the correctness of the initial assumptions, it is perilous to rely on their conclusions. The approach in work [49] is fundamentally different from the above. As a characteristic of the author in the work [49], the proportion of all official words (prepositions, conjunctions and particles) in a sequential fragment of 16,000-word usages was used. The simplicity of the characterisation made it possible to conduct a comprehensive statistical experiment on the works of more than 20 classical writers. It was found that during the entire period of the writer's creativity, the proportion of official words (two decimal places) remained constant. At the same time, this share varies significantly from writer to writer, taking values from 0.20 to 0.30. It allowed the authors [49] to seriously substantiate Sholokhov's plagiarism since his author's invariance is significantly different from the value of the parameter found in the first parts of "Silent Don". Thus, the study [49] allows us to make a reasonable judgment that in the first parts of his epic, Sholokhov "used" some "additional source", i.e., committed plagiarism, which follows from such an analysis. It is another matter of what the "source" was: the texts of F. Kryukov or the texts of some other "co-author" of M. Sholokhov. In the future, the development of scientific thought in this problem will go towards selecting other necessary statistical parameters of the text, which can indicate the work's authorship with a certain degree of probability. In particular, work [51] uses a method based on the characteristics of word frequency and the "distance" between words. In work [36], the authors followed the path of improving the existing Hetso methodology and chose seven statistical characteristics of the text as a basis:

1. Lexical spectrum of the text at the text level;
2. The linguistic spectrum of the text at the dictionary level;
3. General distribution of sentence/word length;
4. Average sentence length in words, calculated based on samples of 30 sentences;
5. Average word length in letters, which is calculated based on samples of 500 text words;
6. Index of vocabulary diversity.

Numerous studies and conference presentations are devoted to various issues of linguistic statistics and primarily stylistic statistics - N.D. Andreev, O.S. Akhmanova, R.G. Piotrovskyi, G.A. Lesskiss, M.P. Kulhav, P.S. Kuznetsov, T.P. Lomtev, A.Ya. Shaykevich, R.M. Frumkina, B. Havranek, Y. Mistryk, Y. Kraus and many others. Other artificial intelligence methods are also being used to determine authorship. In particular, the work [46] uses the Hamming neural network and the pattern recognition method. The author of this method divides the text into three sublevels - the level of words, the level of sentences, and the level of paragraphs. The most significant parts of each level are found, and patterns are recognised through Hamming's neural network. In all the listed works on the definition of authorship, the researchers focused on syntactic constructions, ignoring morphological constructions. Meanwhile, it is straightforward to set up an experiment that shows that frequency analysis distinguishes fragments of 5-6 reference authors well. Such accuracy is probably related to the stability of such morphological features as affixal-root constructions (sequences of

prefixes, roots, suffixes and endings). Careful research led to a methodology for determining authorship based on a mathematical model that takes into account such formal characteristics of the author's language as:

a) number of operative words (prepositions, conjunctions and particles),
b) morphemes (prefix, root, suffix, inflectional) and their sequences are used;
c) the complexity of the grammatical constructions used;
d) Actually, the author uses a dictionary.

2.9 Description of known lexicographic analysis systems

The beginning of the development of applied computer lexicography fell on the 2nd half of the 20th century. Thus, initially, experimental text analysis systems were created. In the 50s and 60s, linguists tested the correctness and suitability of linguistic theories for automation. The first developed systems used data from the theory of morpheme analysis, morphonology and lexicogrammatical classes of words. Automatic analysis in such systems includes three components:

- Automatic selection of the base of the word form from the text;
- Search for the base of the word in the base dictionary;
- Comparison of the structure of the word form with data about its basis, which are stored in the dictionary of bases.

Thus, each text's word form was analysed using previously prepared dictionaries of bases, roots, prefixes, suffixes and inflexions. However, the systems of this period did not distinguish between homonyms - words that have the same form but different meanings. During work on experimental systems, new principles of interaction with the computer appeared over time. In systems of automatic morphological analysis, the possibility of simplifying the procedure for determining the morphological characteristics of word forms with the help of "intelligent" computer capabilities was introduced. In addition, the use of textual regularities in word forms, the combination of morphological analysis with syntactic, and in some cases - with semantic and lexicographic analysis [6] became evident.

In the 70s and 80s of the 20th century, industrial systems for processing text information appeared. The following functions were included in the developed systems:

- Automatic translation from one language to another allowed the system to convert text from the source language to the target language using different translation methods and algorithms.
- Literature information retrieval systems help users find the needed literature using search queries, keywords, or other parameters. They provided access to large databases and helped navigate the volume of available literature.
- Automated and automated text editing system functions allow text editing to be performed automatically by checking spelling, grammar, punctuation, and other aspects. They helped to improve the quality and style of texts.
- Automated annotation and abstracting of literature functions allowed systems to automatically create abstracts and annotations for scientific articles, helping users quickly familiarise themselves with the content of documents without reading them in full.

As Y. Marchuk notes, these systems played an important role in providing various text-processing processes and expanding the possibilities of working with text information [34]. One of the developed tools for document text indexing is the SMART system, described by G. Selton [26]. This system is based on a developed system of dictionaries and tools for their processing. The main components of the SMART system include:

- The system of dividing the words of the text into inflexions and the base allows you to break the words into the central part and inflexions, simplifying the processing of the text.
- A dictionary of equivalences or thesaurus allows you to replace equivalent words with one or more concepts. A thesaurus is used to generate content identifiers instead of word bases.
- The hierarchical thesaurus, which is presented as a hierarchy of concepts, allows you to search by more general or specific ideas and concepts associated with them.
- Dictionaries of statistical and syntactic phrases are used to analyse the relationships between words in the text.
- A dictionary service system ensures practical work with dictionaries and access to necessary lexicographic information.
- The SMART system provides ample opportunities for indexing and processing textual information using developed dictionary components and algorithms.

Other systems have also been developed and continue to be developed, particularly the system developed by the Ukrainian Language and Information Fund of the National Academy of Sciences of Ukraine headed by the famous Ukrainian linguist V. A. Shirokov. under the name "MONDILEX"[53]. The project "Conceptual modelling of the

network of professional research centres in the field of Slavic lexicography and their electronic (digital) resources" (MONDILEX) is an initiative that unites lexicography professionals from various countries of Eastern Europe. This network includes the following institutions: the Institute of Mathematics and Informatics in Bulgaria, the Institute of Slavic Studies in Poland, the Institute of Linguistics named after L. Shtura in Slovakia, the Institute named after Jozef Stefan in Slovenia, the Institute of Information Transfer Problems in the Russian Federation and the Ukrainian Language and Information Foundation of the National Academy of Sciences of Ukraine. This system provides an opportunity:

- to automate the work of each lexicographer at his workplace;
- provide the possibility of online communication between territorially separated researchers and whole creative teams working on the implementation of fundamental joint linguistic projects;
- ensure creative interaction of linguists of all MONDILEX universities;
- easily create bilingual, grammatical, and explanatory dictionaries.

An unsolved problem is the identification of the authorship of short texts. Existing methods work with texts of more than 30,000-40,000 characters and many training examples (5-100 or more). Therefore, finding solutions to reduce the required volume of samples and their number is an urgent task. Currently existing software complexes for identifying the author, including "Stampomer" (JI.JI. Delitsyn), "Linguoanalyzator" (Moscow, D.V. Khmelov), "Attributor" (Moscow, Moscow State University, Polikarpov AA, etc.), "Linguistic Analyzer" (Samara, A. Lviv), "SMALT" (Petrozavodsk, PetrGU, AA Rogov, etc.), "Stile Analyzer" (Tomsk, 11 U, O.G. Shevelyov), "JGAAP" (USA, R. Juola), "Author" (Moscow, VNIISE, EKC UVS of Russia) are implemented based on authorship identification methods, the mathematical and linguistic apparatus of which does not always guarantee an accurate result. Most of the programs are of a demonstration nature or are not intended for solving real practical problems. The amount of text required for the programs to work is, at best, 30,000 characters, which also calls into question the possibility of their use in attributing real disputed texts. Existing programs are not oriented to work with short texts that have their specifics. Peculiarities of natural languages are not taken into account when analysing authorship. Modern machine methods of intelligent data analysis are only partially implemented in them.

3. Problem Formulation

Over the past five years, the combination of NLP, AI, and ML has advanced the field of authorship identification significantly. While significant progress has been made, further research is needed to overcome existing challenges and expand the applications of authorship analysis in different contexts.

Identifying the authorship of texts is an essential task in the fields of natural language processing, artificial intelligence, and machine learning. It finds applications in areas such as plagiarism detection, forensic linguistics, and social media analysis. In recent years, there has been significant progress in this area due to the development of new methods and algorithms. Stylometry deals with the quantitative analysis of the author's writing style. Modern techniques use statistical and machine approaches to analyze the lexical characteristics of a text, such as the frequency of use of functional words. It allows you to identify the unique stylistic features of the author.

The application of machine learning algorithms, such as supporting vector machines and naïve Bayesian classifiers, is a common approach to identifying authorship. In addition, neural networks, specifically deep neural networks, are used to analyse texts in order to determine authorship. They are able to learn from the texts of well-known authors and classify new texts with a certain degree of confidence. Various ML algorithms are applied to identify authorship. **Supporting vector engines (SVMs)** are effective for classifying texts based on highlighted features. **Naïve Bayesian classifiers** use probabilistic models to analyse text. **Deep neural networks** of the architecture, such as long-term and short-term memory (LSTM), are applied to detect sequential patterns in text, which improves the accuracy of authorship identification.

Personality analytics tools such as NetOwl use AI approaches, including NLP and ML, to extract entities, relationships, and events from textual data. They are applied for semantic search, geospatial analysis, and other tasks related to text analysis. Despite the achievements, there are specific challenges in identifying authorship:

- Analyzing **short texts** such as tweets is difficult due to the limited number of stylistic features.
- **Cross-genre attribution.** Models can have difficulty generalizing between different genres or domains, which requires adapting models to the specifics of the genre.
- **Multilingualism.** Processing texts in different languages requires models that are able to take into account linguistic features and interlingual patterns.

An author's style is a unique set of linguistic and stylistic features that distinguish one author's texts from another's. This section aims to formulate the problem of identifying such styles in scientific and technical publications, taking into account both the theoretical basis and the practical aspect. The main task considered in this work is the identification and classification of unique stylistic features of authors in scientific and technical publications. The goal is to develop a

reliable method for distinguishing and using these features to accurately determine authorship or analyse writing styles. This task includes several subtasks.

- Identification of features is the identification of relevant linguistic and stylistic features that can be used to characterise the author's style. These features may include syntax, vocabulary, punctuation, sentence length, and complexity.
- Data collection comprises significant and representative scientific and technical works from various authors.
- Modelling is developing or using computational models to learn and distinguish between different authoring styles based on selected features.
- Verification of accuracy and stability of models for determining authorship within various disciplines of scientific and technical works. [2], [3]

The main steps that must be taken to develop and implement an algorithm for solving the given problem:

- Identify and distinguish distinctive language features from scientific and technical texts.
- Use existing machine learning models capable of distinguishing between different authoring styles.
- Evaluate the performance of machine learning models using well-known and widely used metrics.

To conclude, there is the potential of using machine learning models to identify the styles of different authors.

4. Selection and Justification of Problem-Solving Methods

4.1 Classification of methods for researching the style of writing texts

Since the task of determining the authorship of a document is found in various fields and is of interest not only to philologists or literary critics but also to historians, criminologists, and lawyers, the study of existing and the creation of new methods for determining authorship is an urgent scientific task. The study of the "author's style" can be carried out at different levels: punctuation, spelling, syntactic, lexicophraseological and stylistic. The most significant interest of researchers is the analysis of the last three levels. There are quite a few methods of style analysis. In general, they can be divided into two groups: expert and formal. Experts can identify the authors of an unknown text or determine the belonging of a work to another author using characteristic linguistic features and stylistic techniques. However, expert methods of analysis are pretty time-consuming. Taking into account the possibilities of analysis using information technologies, we will consider in more detail the formal techniques of text analysis that can be used to create automated systems for determining authorship. The modern problem of determining a document's authorship lies at the intersection of AI, ML, deep learning, psychology, linguistics, and other sciences (Fig. 1).

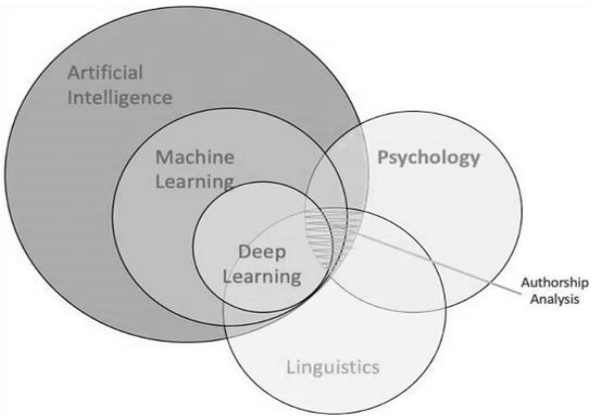


Fig. 1. The place of the problem of identifying the author in the context of other related sciences

Formal methods are based on algorithms of statistical analysis, machine learning, neural networks, Text mining algorithms, etc. Recently, the popularity of embedded network methods has been growing. Let us consider in more detail the main ideas and tasks of algorithms of each of the industries on which modern methods of identifying the author are based on the classification of textual documents.

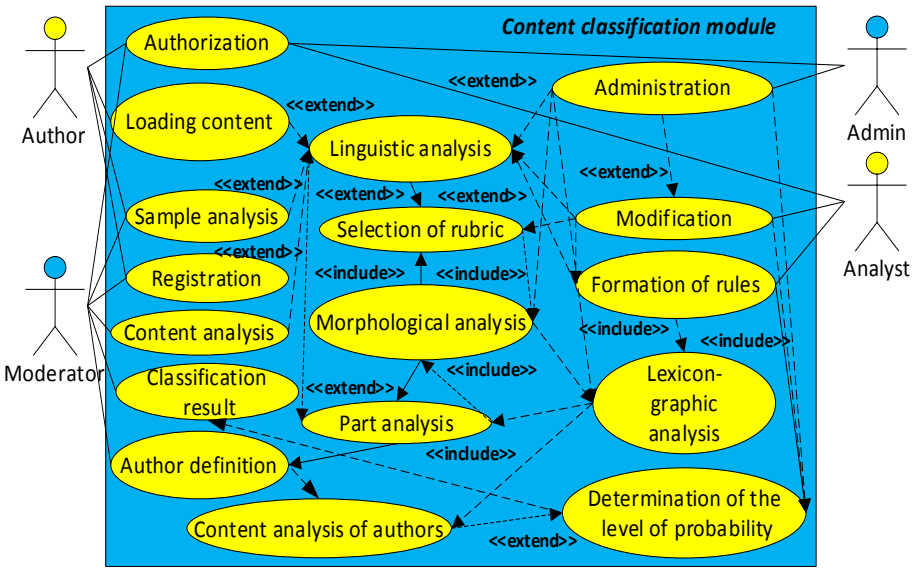


Fig. 2. Diagram of text classification use cases

5. The Mathematical Model for the Primary Linguistic Process of Processing Textual Content

The NLP process in an AI system is presented as a mandatory operators sequence:

The textual content source → *input content* → *grapheme analysis* α → *morphological analysis* β → *lexical analysis* γ → *syntactic analysis* δ → *semantic analysis* λ → *structured text content*

The primary linguistic analysis process is presented:

$$Y = \mu(D_\mu, C_\mu, R_\mu, o(D_o, C_o, R_o, \varsigma(D_\varsigma, C_\varsigma, R_\varsigma, \iota(D_\iota, C_\iota, R_\iota, , \lambda(D_\lambda, C_\lambda, R_\lambda, \delta(D_\delta, C_\delta, R_\delta, \gamma(D_\gamma, C_\gamma, R_\gamma, \beta(D_\beta, C_\beta, R_\beta, \alpha(D_\alpha, C_\alpha, R_\alpha, X)))))))))), Y = \mu \circ o \circ \varsigma \circ \iota \circ \lambda \circ \delta \circ \gamma \circ \beta \circ \alpha, \quad (1)$$

where sets of production/association rules $R = \{R_\alpha, R_\beta, R_\gamma, R_\delta, R_\lambda, R_\varsigma, R_o, R_\iota, R_\mu\}$, linguistic dictionaries $D = \{D_\beta, D_\mu, D_\delta, D_\alpha, D_o, D_\varsigma, D_\gamma, D_\iota, D_\lambda\}$ and multiple content $C = \{C_\mu, C_\varsigma, C_\iota, C_\delta, C_\gamma, C_\beta, C_o, C_\lambda, C_\alpha\}$.

Algorithm for the primary linguistic process of processing content:

Stage 1. Grapheme analysis:

$$C_\alpha = \alpha_7 \circ \alpha_6 \circ \alpha_5 \circ \alpha_4 \circ \alpha_3 \circ \alpha_2 \circ \alpha_1, \text{ or } C_\alpha = \alpha(X, D_\alpha, R_\alpha), \quad (2)$$

where α_7 is generating a marked linear sequence of words C_α with official signs and connections; α_6 is marking non-text strings as special symbols, formulas, figures, tables, etc.; α_5 is identification and marking of unrecognized chains as numbers, dates, constant returns, abbreviations, proper and geographical names, etc.; α_4 is forming a set of unrecognized chains; α_3 is grapheme analysis of linguistic chains into separate words; α_2 is grapheme parsing of the input text X into sections, paragraphs and sentences; α_1 is OCR operator; R_α is grapheme analysis rules; D_α is grapheme dictionaries and libraries; C_α is grapheme structure of the input text; α is the GA operator; X is the input text data array.

Stage 2. Morphological analysis:

$$C_\beta = \beta_3 \circ \beta_2 \circ \beta_1 \text{ or } C_\beta = \beta_3 \circ \beta_4 \circ \beta_1, \text{ or } C_\beta = \beta(C_\alpha, D_\beta, R_\beta), \quad (3)$$

where β_4 is stemming operator of words; β_3 is POST operator for segmented words; β_2 is lemmatization of lexemes; β_1 is the morphological segmentation of a graphemically recognized chain of words/tokens.

Stage 3. Lexical analysis:

$$C'_\gamma = \gamma_5 \circ \gamma_4 \circ \gamma_3, C'_\gamma = \gamma_2 \circ \gamma_1, \text{ or } C'_\gamma = \gamma_5 \circ \gamma_4, \text{ or } C_\gamma = \gamma(C_\beta, D_\gamma, R_\gamma), \quad (4)$$

whereγ_5 is a Text-To-Speech operator (TTS); γ_4 is a word tokenization/segmentation operator as data preparation for building a parsing tree in SYA; γ_3 is Optical Character Recognition operator (OCR) as the second part after GA and MA for clarifying incorrect moments of recognition, taking into account the recognized neighbouring tokens; γ_2 is speech recognition operator (SR) or speech-to-text (STT) depending on the content of the NLP task; γ_1 is the Speech

Segmentation operator for identification/clarification of words/phrases/tokens.

Stage 4. The syntactic analysis:

$$C_\delta = \delta_3 \circ \delta_2 \circ \delta_1, C_\delta = \delta(C_\gamma, D_\delta, R_\delta), \tag{5}$$

where δ_3 is syntactic parsing of phrases/sentences for a SYA tree creation; δ_2 is a boundary ambiguity identification/elimination or sentence violation; δ_1 is the Grammar induction;

A language consisting of strings of the form $abcd\ldots d'c'b'a'$ (composed of symbols a_1, a_2, a_3, a_1', a_2', a_3') is generated by a grammar of 6 rules:

$$\left.\begin{matrix} I \to a_i I a_i^{'} \\ I \to a_i a^{'} \end{matrix}\right\} i = 1,2,3. \tag{6}$$

Such grammar does not provide, for example, a natural description for the so-called non-project constructions with breaks, crossing ( , ) or framing ( , ) directions of syntactic dependence (Fig. 3).

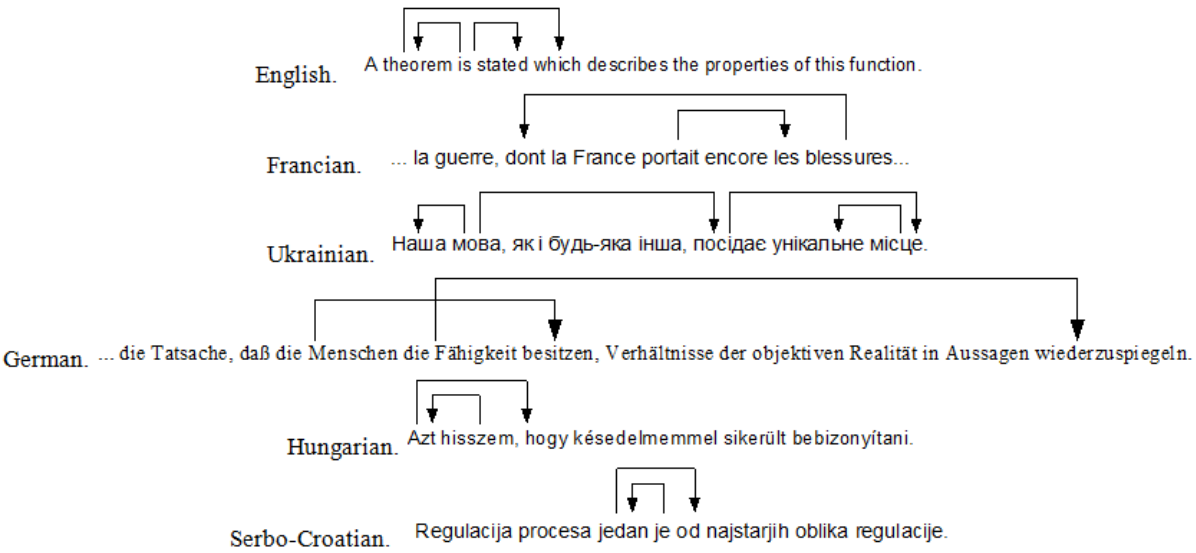


Fig. 3. Examples of natural descriptions for so-called non-design constructions

The grammar $G' = \langle D', D_1', I', R' \rangle$ has a basic dictionary $D' = N_1, N_2, \ldots, N_n$ symbols and rules of the form $R' = \{Y \to ZN_i, X \to N_i\}$, where $Y \in D_1'$ and $Z \in D_1'$.

$$D_1 = D' \cup D_1' \cup D_1^1 \cup D_1^2 \cup \ldots \cup D_1^n,\ R = R' \cup R_1 \cup R_2 \cup \ldots \cup R_n,\ G = G' \cup G_1' \cup G_2' \cup \ldots \cup G_n', \tag{7}$$

where the leading dictionary D in all G_i', and the auxiliary additional dictionary and scheme.

$$R_1 = \begin{cases} N_1 \to bP_1 \\ P_1 \to aQ_1 \\ Q_1 \to aQ_1 \\ Q_1 \to c \end{cases}, R_2 = \{N_2 \to d, R_3 = \begin{cases} N_3 \to aP_3 \\ N_3 \to bQ_3 \\ N_3 \to cW_3 \\ P_3 \to a \\ Q_3 \to b \\ W_3 \to dW_3 \\ W_3 \to eW_3 \\ W_3 \to d \end{cases}, R_4 = \begin{cases} N_4 \to cP_4 \\ P_4 \to b \end{cases}, R' = \begin{cases} I \to BN_1 \\ B \to CN_1 \\ C \to BN_2 \\ C \to EN_3 \\ E \to EN_4 \\ E \to N_2 \end{cases} \tag{8}$$

Step 1. An unconstrained generated sequence is generated to the right by N_i as a syntactic group or sentence based on the rules of R'.

Step 2. Any of N_i based on R_i is expanded indefinitely in the form of a tree (Fig. 4-10) from right to left - into a chain of terminal symbols as words.

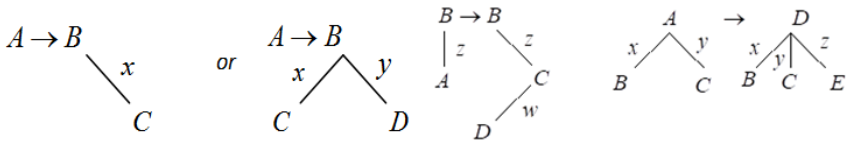


Fig. 4. Rules for building a tree

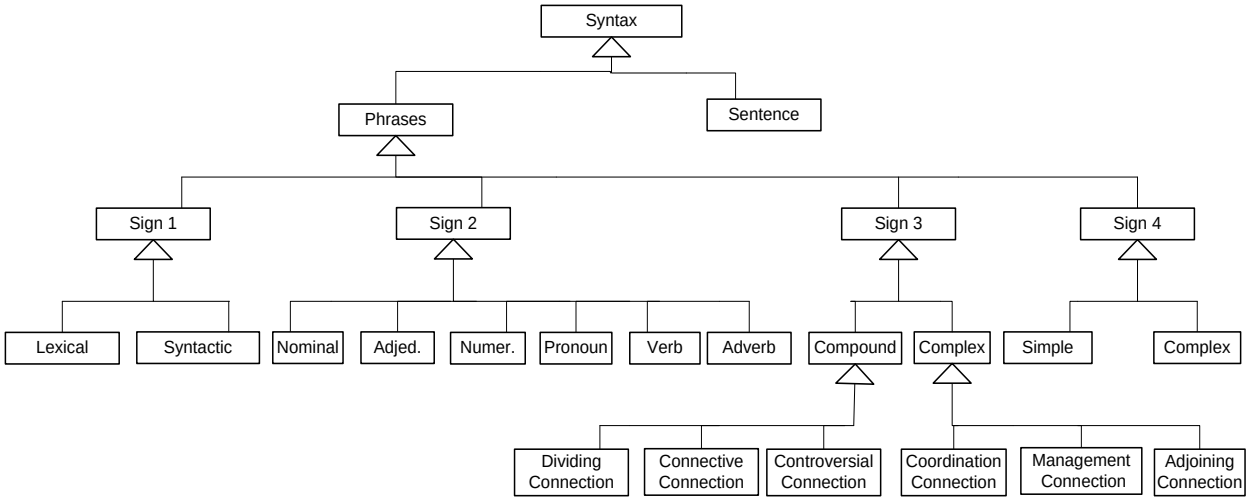


Fig. 5. Class diagram for the Syntax Phrase type hierarchy

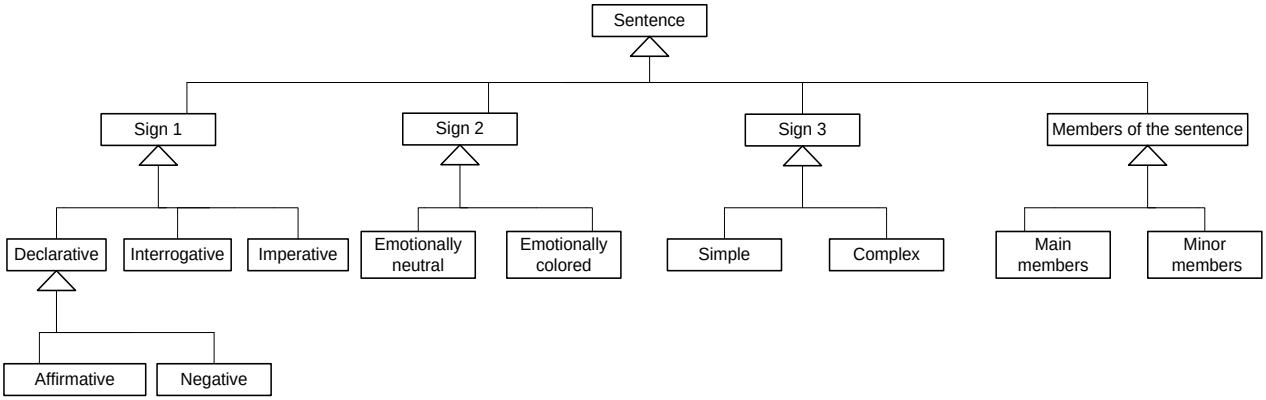


Fig. 6. Class diagram for the sentence type hierarchy

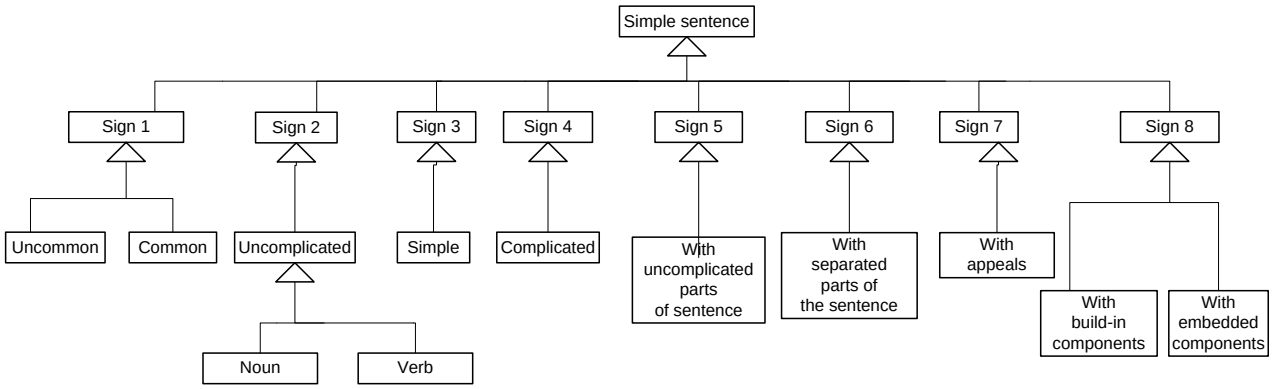


Fig. 7. Class diagram for a hierarchy of the type Simple sentence

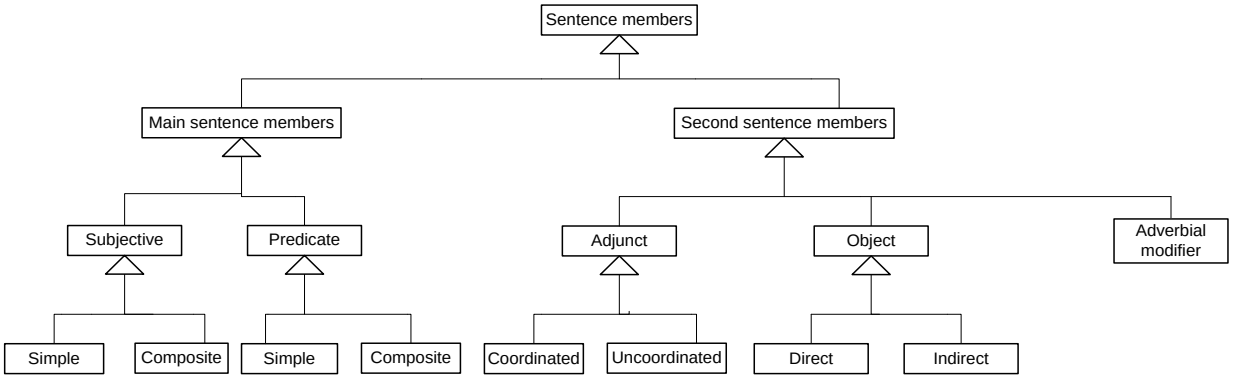


Fig. 8. Class diagram for a hierarchy of the type the Sentence Members

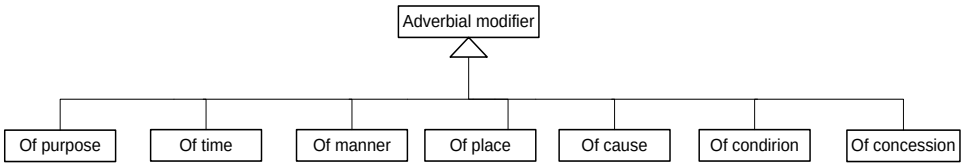


Fig. 9. Class diagram for the Circumstance type hierarchy

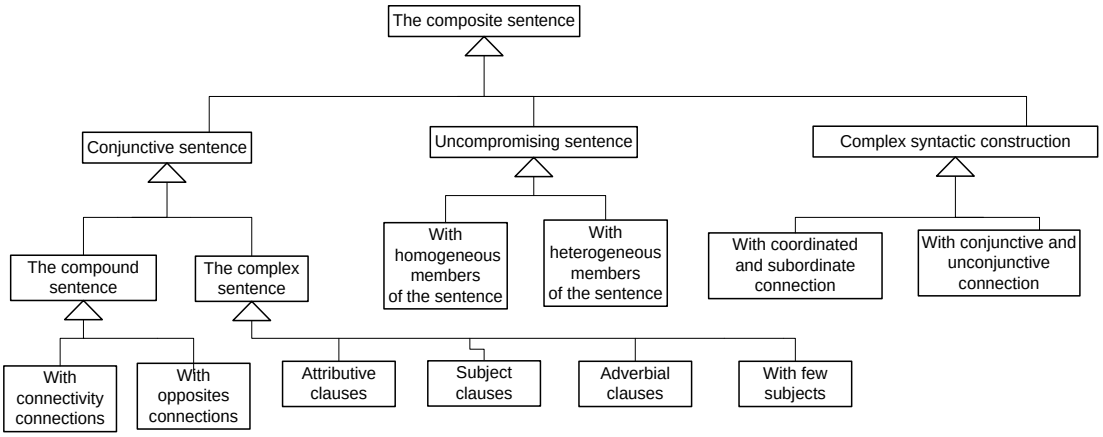


Fig. 10. Class diagram for the Complex Sentence type hierarchy

Stage 5. Semantic analysis:

$$C_\lambda = \lambda_2 \circ \lambda_1, C_\lambda = \lambda(D_\lambda, C_\delta, R_\lambda), \tag{9}$$

where λ_2 is the relational semantics identification of the interdependencies of the lexeme; λ_1 is the lexical semantics identification with each lexeme values collection generation.

Stage 6. Referential analysis ι formation of interphrase units C_ι.

$$C_\iota = \iota(D_\iota, C_\lambda, R_\iota) \tag{10}$$

Stage 7. Structural analysis:

$$C_\varsigma = \varsigma(D_\varsigma, C_\lambda, R_\varsigma) \text{ or } C_\varsigma = \varsigma\big(D_\varsigma, C_\iota, R_\varsigma\big). \tag{11}$$

Stage 8. Ontological analysis:

$$\mathrm{C_o} = \mathrm{o}(D_\mathrm{o}, C_\lambda, R_\mathrm{o}), \mathrm{C_o} = \mathrm{o}(D_\mathrm{o}, C_\iota, R_\mathrm{o}) \text{ or } \mathrm{C_o} = \mathrm{o}(D_\mathrm{o}, C_\varsigma, R_\mathrm{o}). \tag{12}$$

Stage 9. Pragmatic analysis:

$$Y = \mu_2 \circ \mu_1, \text{ or } Y = \mu(D_\mu, C_\mu, R_\mu, C_\lambda, [\mathrm{C_o}, C_\varsigma, C_\iota],), \tag{13}$$

where μ_2 is text processing through higher-level NLP applications; μ_1 is the semantics identification outside individual sentences/phrases.

5.1 Stemming

Stemming is the reduction of several common root words to their common root (Fig. 11). The root of a word is not identical to the morphological root based on the dictionary. It is simply an equal or lesser form of the word.

MA consists of identifying, analysing and determining the form and structure of words in a natural language text:

$$C'_\beta = \beta_3 \circ \beta_2 \circ \beta_4 \circ \beta_1, \text{ or } C'_\beta = \beta(D_\beta, C_\beta, R_\beta), \tag{14}$$

where $C'_\beta = \beta_3 \circ \beta_4 \circ \beta_1$ (for most cases of various topics messages) or $C'_\beta \subseteq C_\beta$, $C'_\beta = \beta_3 \circ \beta_2 \circ \beta_1$ (enough for English-language short texts on a specific topic) using methods such as Stemming β_4, POST β_3 (marking of parts of speech), Lemmatization β_2, and Morphological Segmentation β_1.

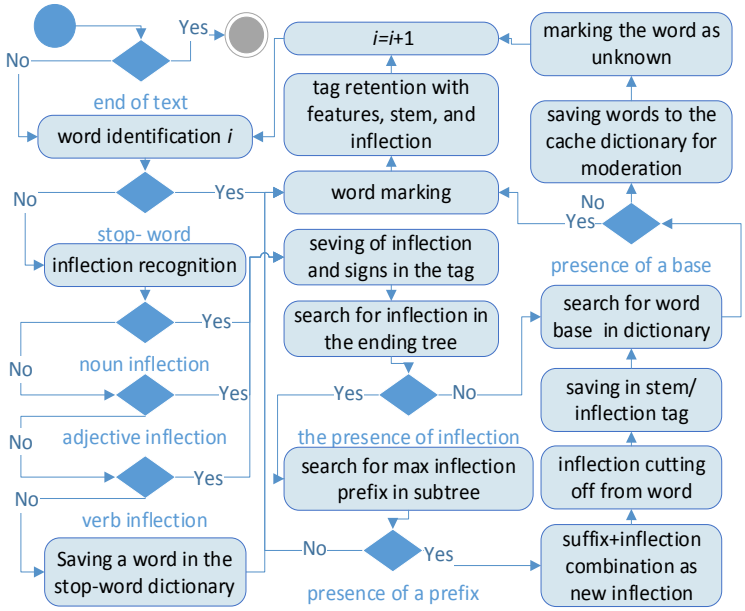


Fig. 11. Modified stemming algorithm

Stemming algorithms are usually rule-based. It can be thought of as a heuristic process that seems to tear off the ends and beginnings of words. The word is revised and goes through a series of conditions that determine how to abbreviate it. For example, we might have a suffix rule that relies on a list of known suffixes. In English, we have suffixes like "-ed" and "ing" that can be useful for cutting off in order to match the words "cook," "cooking," and "cooked" all to the same root "cook.

The purpose of the definition is to recognize sets of words such as "calculation" and "calculation" or "applied", "apply", "apply" and "apply", which can be interpreted as equivalent. Many algorithms create basic principles that have been developed to reduce a word to its root form, with which it is replaced. For example, "computing" and "computing" can be related to "computer" and "apply", and so on to "application".

However, since stemming is generally based on heuristics, it is far from perfect. In fact, this method usually suffers from two disadvantages: overfulfillment and underfulfillment. Overfulfillment occurs when too many words are abbreviated. It can lead to meaningless roots, where all the meaning of the word is lost or confused. Or it can cause words to become attached to the same roots, although they probably shouldn't be.

Take the four words "universal", "university", "Universe", and "universities". The stemming algorithm that shortens these four words to "univers" is overstemmed. While it might be nice for "universal" and "Universe" to tie together and "university" and "universities" to come together, all four don't fit. The first two solutions can have "univers" and the last two "universi" solutions. However, the creation of rules that will lead to such a result is a separate scientific task.

Understanding is the opposite question. It comes when we have several words that are actually forms of each other. It would be nice if everyone solved the same root, but unfortunately, they don't. It would be nice to subdue them to "dat". But what then to do with "date"? And is there a good general rule? Or do we apply a particular rule for a specific example? These issues quickly become problems when it comes to stemming.

The application of new rules and heuristics can quickly get out of control. Solving one or two problems that arise as a result of underperformance or overperformance of stemming can lead to the appearance of two more! Composing a good stemming algorithm is hard work.

Using stemming can be a very effective way to reduce the number of words in a dictionary to a relatively manageable number. However, as with stop words, there is no standard algorithm, which is ideal and can lead to the extraction of essential words. For example, the word "applique´e" in a document can be a necessary attribute in terms of its classification. Still, it can be reduced by switching to "application", which has the same root.

Known stemming algorithms are:

- The porter stemmer algorithm is the oldest. It dates back to the 1980s, and its main task was to eliminate commonly known endings to words so that they could be reduced to a standard form. It is not too tricky, and development has frozen on it. It is generally a good initial base stemmer, but it is not recommended for use in any application. Instead, it takes its place in research as a good, basic algorithm that can guarantee reproducibility. It is also a very mild algorithm compared to others.
- The snowball stemmer algorithm is also known as the Porter2 algorithm. It is almost universally recognized as better than its predecessor. This algorithm is also more aggressive than Porter's stemmer. A lot of the things that have been added to the definition of root in Snowball have been related to the problems seen in the Porter stemmer. There is approximately a 5% difference in how Snowball is applied against Porter's algorithm.
- Lancaster stemmer is the most aggressive stemming algorithm in groups. However, if you use NLTK stemming, you can easily add your own custom rules to this algorithm. It is a good choice for this. One of the complaints about this algorithm is that it is sometimes too aggressive and can turn words into strange roots.

5.2 Lemmatization

Lemmatization usually refers to the correct operation using a dictionary and morphological analysis of words, generally aimed at extracting only unimportant endings and returning the main or dictionary form of the word, which is known as the lemma. To bring a word to its original form, it is necessary, first of all, to determine which part of speech it belongs to. It requires additional computational linguistics, such as the part of speech. It allows us to make better decisions (such as solving "is" and "am" to the word "be").

Another thing to note about lemmatization is that it is more challenging to create a lemmatizer in a new language than a stemming algorithm. Since lemmatizers need much more knowledge about the structure of language, this is a much longer process than just trying to set up a heuristic stemming algorithm.

Fortunately, if you work with English, you can quickly use lemmatization through NLTK, as in the case of stemming. However, for best results, you will have to submit a portion of the language tags to the lemmatizer. Otherwise, it may not reduce all the words to the desired lemmas. Moreover, it is based on the WordNet database (which is similar to a network of synonyms or thesaurus), so if there is no good link there, then we still will not get the correct lemma. To demonstrate the difference between stemming and lemmatization, consider the following example for the word " caring" Stemming should take away the standard ending "ing". In contrast, lemmatization should lead to the initial form of the verb " care".

‘Caring’ -> Lemmatization -> ‘Care’ and ‘Caring’ -> Stemming -> ‘Car’

It is advisable to carry out lemmatization before stemming.

5.3 Lexical methods of keyword extraction

The previous paragraph describes a statistical approach to keyword allocation. In contrast to statistical methods, there are also lexical methods for constructing a feature space. As with statistical methods, there is no perfect way to highlight keywords. Attempts to create a universal linguistic method for highlighting keywords were unsuccessful. The classification of lexical approaches can be carried out in specific linguistic directions. Linguistic methods based on the meanings of words use the ontology and semantics of the word. The disadvantage of these methods is their resource consumption and labour intensity: the development of ontologies requires high costs. In addition, the human factor in performing the operations of linguistic analysis of texts is a source of a significant number of errors and inaccuracies. Therefore, it is necessary to automate the process of analyzing the texts of documents, and this is a far from trivial scientific task. There is a method for automated definition of key terms from text documents based on the measure of semantic proximity of words, calculated using the Wikipedia database, construction of a semantic graph, and selection of keywords using the Girvan-Newman algorithm. The advantage of this method is the absence of the need for prior training, as it works directly with the database. The effectiveness of the technique has been confirmed by experimental estimates of the accuracy and completeness of extracting key terms from the text.

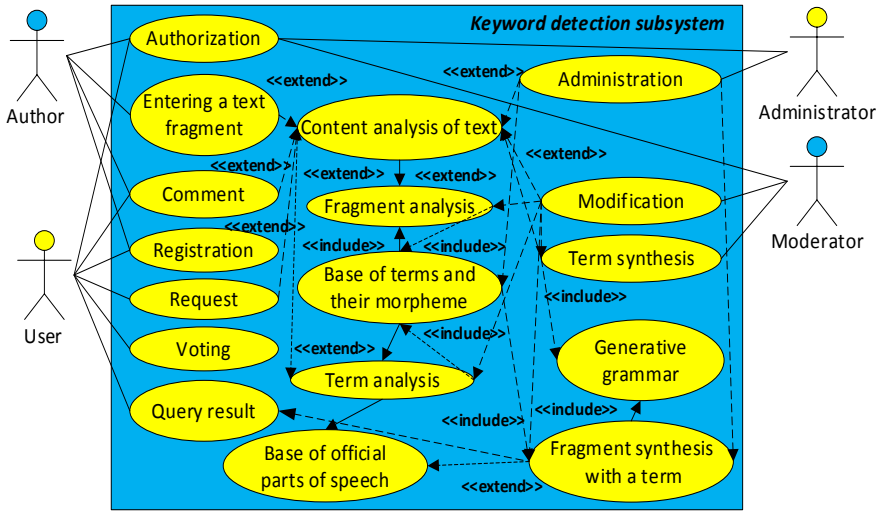


Fig. 12. Keyword Identification Use Case Diagram

5.4 Text Classifier Evaluation

After we have converted the training documents into a normalized vector form, we can build a training set for each category C_k. In Figure 13, the values *a*, *b*, *c*, and *d* are the quantities of True Positive, False Positive, False Negative, and True Negative classifications, respectively. For a perfect classifier, both b and c would have to be zero.

Actual classes	Predicted classes	
	C_k	not C_k
C_k	*a*	*c*
not C_k	*b*	*d*

Fig. 13. Configuration matrix for category C_k

5.5 Hierarchical clustering method

The hierarchical clustering method allows you to identify clusters in a data set and understand their interpretation. For this, different metrics of the distance between objects are used. For example, you can use the Euclidean and the Chebyshev (Kolmogorov) measures $\rho(x,y) = \max_{i \in 1...n} |x_i - y_i|$. When using the hierarchical clustering method, the question arises of determining the distance between clusters when they are linked together.

6. Stages of Software Implementation of the Algorithm

This section presents a detailed algorithm for determining the author's style in scientific and technical publications with all the main stages. The methodology for determining the author's style includes several vital stages: data collection and pre-processing, feature selection, training of machine learning models, evaluation and comparison of the effectiveness of models.

6.1 Data collection and pre-processing

Data collection. Collecting a set of scientific and technical English publications from various authors is necessary. It is also worth ensuring that the data set is balanced, with approximately the same number of publications from each author, and sufficiently diverse. Data collection sources include open digital libraries such as IEEE Xplore, PubMed, and arXiv. The arXiv digital library was chosen as a source of scientific and technical publications to implement this task. A total of 5,943 texts from 124 different authors were collected. Some of the scientific and technical publications were provided to us by the editorial board of the Bulletin of Lviv Polytechnic University of the "Information Systems and Networks" series for the period 2001–2021.

Preliminary processing. The next step is converting the collected publications into a single format (for example, plain text in .txt format). Tokenisation is also required to break the text into words or sentences and remove stop words, punctuation marks, and non-alphabetic characters. Lemmatisation is also used to reduce words to their root forms. [4]

6.2 Identification of features

TF-IDF (Term Frequency-Inverse Document Frequency). An important step is to apply the TF-IDF technique to weigh the importance of words based on their frequency in a document and the number of documents containing a

particular word relative to the entire set of documents. The primary motivation for using TF-IDF in text analysis, in particular for determining the author's style, includes the following [4-5]:

- TF-IDF de-weights frequently occurring terms and highlights terms more specific to particular texts or authors.
- TF-IDF helps to capture the unique vocabulary of an author by assigning more weight to terms that frequently occur in the texts of each particular author but rarely occur in other texts.
- TF-IDF provides a better representation of textual data, improving machine learning models' performance.

6.3 Selection and training of ML models

Selection of machine learning models. You must select the appropriate machine-learning algorithms for further application and comparison at this step. The following four models were chosen for this task: Support Vector Classifier (SVC), Logistic Regression, 97 and Naive Bayes. This research explores the intersection of machine learning (ML) and natural language processing (NLP) to develop a Python program capable of identifying the authorship of scientific and technical publications through stylistic analysis.

Training machine learning models. Using a specific part of the data set for learning (training) the model is necessary. An essential and valuable practice in training models is to adjust the available hyperparameters for each model, selecting their most optimal values to achieve better accuracy rates.

6.4 Performance evaluation and comparison of machine learning models

Testing machine learning models. It is necessary to apply the trained models to the part of the data set that was not used in training. In this way, it is possible to check the ability of models to generalise and their operation on previously unknown data.

Performance evaluation and comparison of ML models. To evaluate the performance of machine learning models, the metrics known are precision, sensitivity, recall, F1-score, and accuracy. Calculating these metrics allows you to compare models based on their performance metrics. Also, in the comparison, you can take into account values such as the training time of the model and the time the model spent identifying the authorship of the texts. Based on these data, a conclusion can be drawn about which of the considered models was the best and its further application potential. [19].

7. Used Tools

This section describes the tools and libraries used to complete the task steps. These stages include data collection and processing, vectorisation of the data set, application of some existing machine learning models to the processed data, and evaluation of model performance.

7.1 Data Collection Tools

A generalised data collection process involves retrieving publication data from arXiv, downloading publications in .pdf format, and converting these PDF files to text in .txt format. Three libraries are used: requests, arxiv and PyMuPDF.

1) Library requests

The requests library sends HTTP requests to interact with web APIs. Using this library for this task consists of several steps:

- Setting parameters for the API request (search query, maximum number of results per query, sorting order, etc.);
- Sending an HTTP request uses the requests.get tool to send a request to the arXiv API endpoint.
- Receive Process Response checks the response status code to ensure the request was successful. Parse the XML response to get the body of the post.
- Extracting authors: selection of authors' names from the received texts. [6]

2) The arxiv library

The arxiv library was used to search the arXiv and download articles. Python tools using the arxiv API configured a search query to find articles by a specific author and used the arXiv client to upload the resulting PDFs to the specified folder. [7]

3) PyMuPDF library

At the final stage of data collection, it is necessary to convert the content of the downloaded PDF files into plain text for further analysis. The PyMuPDF library (imported into the development environment as Fitz) was used to process PDF files. By opening each downloaded PDF article, extracting the text from each page and concatenating it into a single line, we got each post in .txt format. [8]

7.2 Data processing tools

4) NLTK (Natural Language Toolkit) library

Text tokenisation. The word_tokenize function divides texts into separate words (tokens). It is part of the nltk. tokenise module. [9]

Filtering of non-alphabetic markers. Any tokens that are not purely alphabetic should be filtered out. Punctuation marks, numbers and any other non-alphabetic characters are removed.

Removing stop words. Stopwords are a list of common words (for example, in English, it is "the" and "in" and others) that are removed from tokens during text pre-processing to reduce noise in the data. These words add little meaningful information to the analysis of the text. It is part of the nltk.corpus module. [9]

Filtering of non-alphabetic markers. Any tokens that are not purely alphabetic should be filtered out. Punctuation marks, numbers and any other non-alphabetic characters are removed.

Removing stop words. Stopwords are a list of common words (for example, in English, it is "the" and "in" and others) that are removed from tokens during text pre-processing to reduce noise in the data. These words add little meaningful information to the analysis of the text. It is part of the nltk.corpus module. [9]

Lemmatisation of lexemes. Word lemmatisation reduces words to their base or root form. It helps to perceive different forms of a word (for example, "big "bigla "runs") as the same word ("run"). WordNetLemmatizer is a class used to lemmatise words. It is part of the nltk.stem module. [9]

Token pooling. At the end of the processing of each publication text, the processed tokens are combined back into one line with words separated by a space.

7.3 Tools for character selection

1) TF-IDF Vectorizer

The TF-IDF (Term Frequency-Inverse Document Frequency) vectoriser is a popular natural language processing and information retrieval technique for converting a collection of raw documents into a matrix of TF-IDF functions. It aims to reflect the importance of a single word in a document relative to a set of documents. Term Frequency (TF): Measures how often a term (word) appears in a document. The term frequency for term t in document d is calculated as:

$$TF(t,d) = \frac{N_d}{N_t} \tag{15}$$

where N_d is the total number of words in document *d*, N_t is the number of occurrences of the word *t* in document *d.*

Inverse Document Frequency (IDF): Measures how important a term is in the entire set of documents. It is the logarithmically scaled inverse proportion of documents that contain the term. The IDF for the term t is calculated as:

$$IDF(t) = \log\left(\frac{N}{N_{dt}}\right) \tag{16}$$

where *N* is the total number of documents, N_{dt} is several documents containing the word *t.*

If a term occurs in many documents, its IDF value will be low.

TF – IDF: Combines TF and IDF scores to assign a weight to each term in each document. The TF – IDF estimate for term t in document d is calculated as: [10]

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \tag{17}$$

TfidfVectorizer from the sklearn library was used in the software implementation. It divides documents into words (tokens), creates a dictionary of all unique terms in a set of documents, and computes the TF-IDF value for each term. TfidfVectorizer in sklearn returns a sparse matrix, where rows represent documents, columns represent terms (features) in the dictionary, and the value in position (i,j) is the TF − IDF weight of term j in document i. [11]

7.4 Machine learning models

Classification models are essential machine learning tools for classifying data into predefined classes. In this task, machine learning models were used to classify the authorship of texts (attribute the text to the author). The following classification models were used: Support Vector Classifier (SVC), Naive Bayes, Logistic Regression and Random Forest.

Support Vector Classifier (SVC) is a vector machine (SVM) for classification tasks. The basic idea of SVC is to find a hyperplane in an N-dimensional space (N is the number of features) that classifies the data points [12-13].

- **Hyperplane and fields.** SVC creates a hyperplane or set of hyperplanes in a high-dimensional space. The best hyperplane is the one that maximises the distance between the data points of both classes. Longer distances are considered better in this regard because they provide better generalizability.

- **Support vectors.** The data points closest to the hyperplane are called support vectors. These points are critical for determining the position and orientation of the hyperplane. Removing the reference vector would change the position of the hyperplane.
- **Kernel conversion.** Suppose the data is not linearly separable in its original feature space. In that case, SVC uses kernel transformation to transform the data into a higher dimensional space that is linearly separable. The most widely used kernels include the linear, polynomial, and radial basis functions (RBF).
- **Optimisation.** SVC solves the optimisation problem to maximise the distance between class data points. It involves minimising the classification error [12-13].

Naive Bayes is a probabilistic classifier based on Bayes theorem with the "naive" assumption of independence between each pair of features given by the class label [14-15].

- **Bayes theorem.** The posterior probability $P(C|X)$ of class C with a given feature X is calculated by the formula:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \tag{18}$$

where $P(C|X)$ is the posterior probability of the class with the given characteristic, $P(X|C)$ is the probability of the characteristic taking into account the class, $P(C)$ is the a priori probability of the class, $P(X)$ is the a priori probability of the characteristic.

- **Assumption of independence.** The Naive Bayes model assumes that the presence of a particular feature in a class is independent of the presence of any other feature.
- **Model options** are several types of naive Bayesian classifiers based on feature distributions [14-15]:
- Gaussian Na ïve Bayes assumes that the functions follow a normal distribution;
- Multinomial Na ïve Bayes used for discrete data where the features represent a count;
- Na ïve Bayes Bernoulli is used for binary/Boolean functions.

Logistic regression is a linear model for binary classification that estimates the probability that the input belongs to a specific class. The model uses a logistic function (sigmoid) to display predicted values with a probability from 0 to 1. [16]

- **The logistic function** is defined as

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{19}$$

where $z = \beta_0+\beta_1x_1+\beta_2x_2+\cdots+\beta_nx_n$. The coefficients β_i, $i = 0,\ldots,n$ are determined using the maximum likelihood estimate.

- The output of the logistic function is a probability value between 0 and 1. Based on a threshold value (usually 0.5), the probability is displayed on a binary class label.
- **Loss function** or Logistic regression uses the logarithm of the loss (or binary cross-entropy) as a loss function that is minimised during training:

$$L(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i log\left(h_\theta(x_i)\right) + (1-y_i) log\left(1-h_\theta(x_i)\right)\right]$$

where $h_\theta(x_i)$ is the predicted probability, y_i is the actual classification label, and m is the number of samples.

- **Optimisation.** The model parameters (θ) are optimised using techniques such as gradient descent to minimise the loss function and fit the model to the training data. [16]

7.5 Random Forest

Random Forest is a machine learning technique that combines multiple decision trees to produce a more accurate and stable prediction. It creates a "forest" of random decision trees, each trained on a loaded data sample and a random subset of features. During class prediction, each tree votes for a class, and the class with the most votes is selected as the final prediction. This method reduces overfitting compared to using a single decision tree, increases reliability, and provides less sensitivity to noise in the data. [17-18]. Each decision tree is a process that makes predictions based on a series of binary decisions based on feature values.

- Random Forest uses a technique called **bagging (Bootstrap Aggregating)**, where multiple subsets of data are created by random sampling with replacement. Each subgroup is used to train a different decision tree.
- **Randomness of functions.** Random Forest selects a random subset of features when splitting nodes while building trees. This randomness helps make the trees less correlated and more diverse.

- **Aggregation.** For classification tasks, Random Forest takes the majority vote of all trees to make the final prediction. [17-18]

7.6 Performance metrics of machine learning models

This research explores the intersection of ML and NLP methods to develop a Python program capable of identifying the authorship of scientific and technical publications through stylistic analysis. When evaluating the machine learning model's performance in a classification task, several metrics can be used to determine how well the model performs. These are precision, recall, F1-score, and accuracy. To describe and understand these metrics, we introduce the following notations:

- The negative state (N) is the data's number of actual negative cases.
- Positive status (P) is the data's number of actual positive cases.
- A negative predicted state (NP) is the number of predicted negative instances in the data.
- Predicted positive status (PP) is the data's number of predicted positive cases.
- True negative (TN) is a result that indeed indicates the absence of the condition.
- True positive (TP) is a result that genuinely shows the presence of the condition.
- False negative (FN) is a result that falsely indicates the absence of the condition.
- False positive (FP) is a result that falsely indicates the presence of a condition.

The total number of cases = P + N.

The precision or predictive significance of a positive result (PP+) is the ratio of correctly predicted positive observations to the total number of expected positive results. This value shows how many of all texts predicted by the model to be written by a particular author were written by that author $Precision = TP/(TP + FP)$. Precision measures the accuracy of optimistic predictions. High precision means that there are few false predictions [19]. A discrepancy matrix is a table that allows you to visualise the performance of a machine-learning algorithm. It has the following form:

Table 2. Matrix of inconformities for evaluating the productivity of classification models

P + N		Predicted state	
		PP	NP
Real state	P	TP	FN
	N	FP	TN

Recall is the ratio of correctly predicted positive results to all results in the actual class. Its value shows how many texts written by a particular author correctly identified by the model.

$$Recall = TP/(TP + FN)$$

A high recall means there are few false negatives. [19]. F1-score is the average harmonic value of precision and recall. It is an indicator that balances the two previous metrics. [19]

$$F1 - score = 2 \times (Precision \times Recall)/(Precision + Recall)$$

Accuracy is the ratio of correctly predicted observations to total observations. This metric shows how often the model makes correct predictions.

$$Accuracy = (TP + FN)/(TP + TN + FP + FN)$$

High accuracy means that the model's predictions are correct in many cases. [19].

8. Experiments

Fig. 14-16 shows the primary goals of an Agile intelligent software solution for textual content authorship identification based on NLP, artificial intelligence, and machine learning (to identify the author of the text as a Ukrainian-language artistic work) [21].

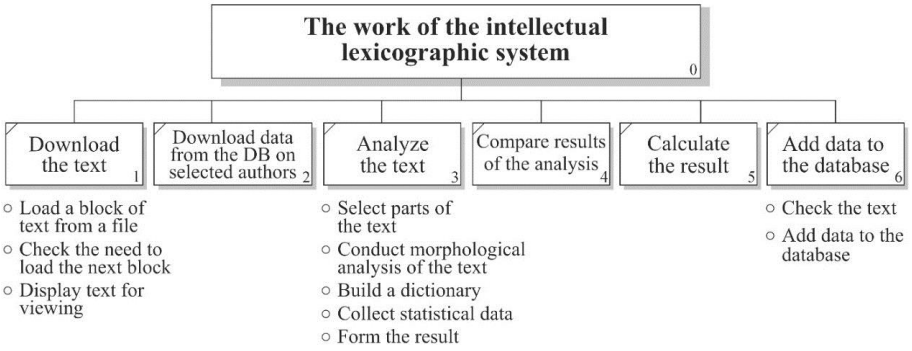


Fig. 14. Tree of goals for an Agile intelligent software solution for textual content authorship identification based on NLP, artificial intelligence and machine learning

The "Load text" process is divided into three sub-processes (Fig. 17): load the current text block from the file, check the need to load the next block and display the text for viewing. The process "Conduct text analysis for analysis" is divided into five sub-processes (Fig. 18): select parts of the text, conduct morphological analysis of the text for analysis, build a dictionary, collect statistical data, and form a result.

Fig. 15. Context diagram of Agile intelligent software solution for textual content authorship identification based on NLP, artificial intelligence and machine learning

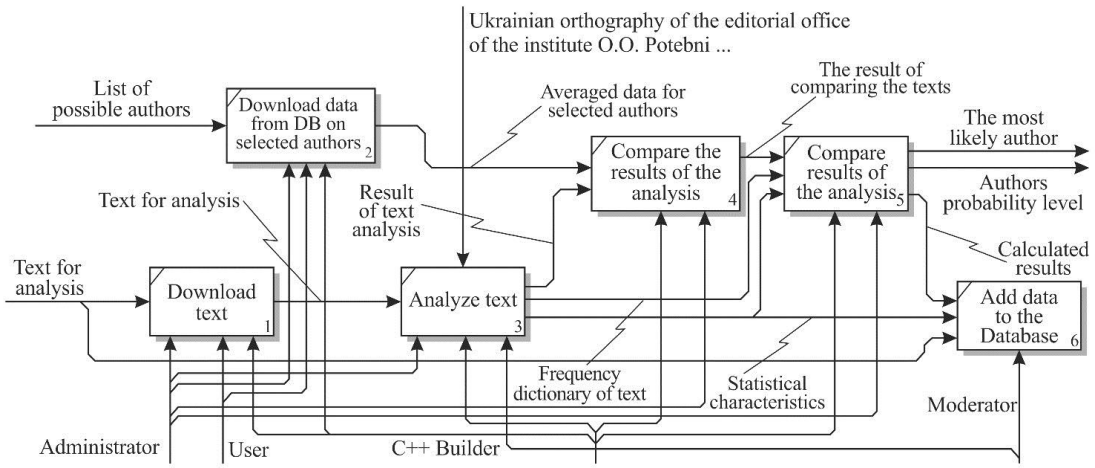


Fig. 16. Context diagram decomposition for Agile intelligent software solution for textual content authorship identification based on NLP, artificial intelligence and machine learning

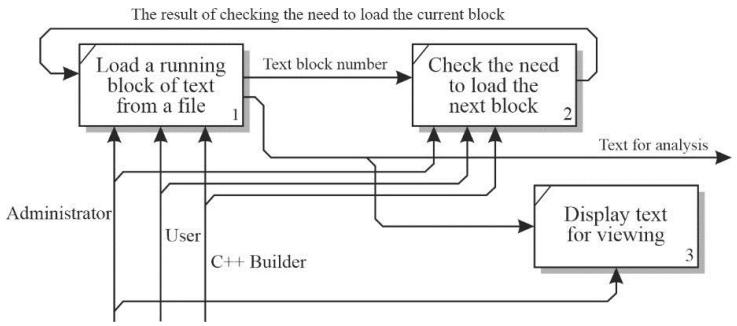


Fig. 17. Load text process decomposition of Agile intelligent software solution for textual content authorship identification based on NLP, artificial intelligence and machine learning

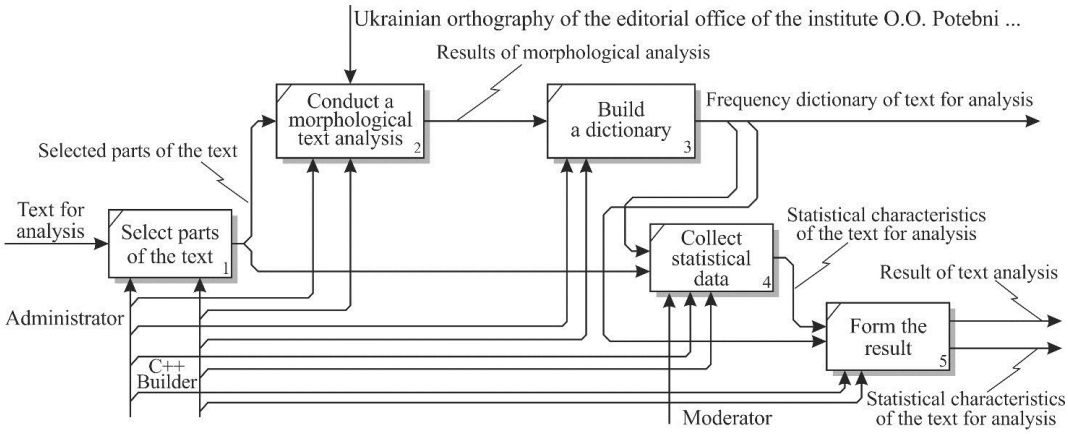


Fig. 18. Decomposition of the process "Perform text analysis for analysis."

The process of "Conducting morphological analysis of the text for analysis" is divided into five tasks (Fig. 19). The "Collect statistical data" process in Fig. 20 is divided into nine tasks: count the number of words, count the number of sentences, count the number of letters, count words of length 1, count words of length 10, perform stylometric analysis, find vowels and consonants, generate statistical data, and supplement the database. The "Add data to the database" process is divided into two tasks (Fig. 21).

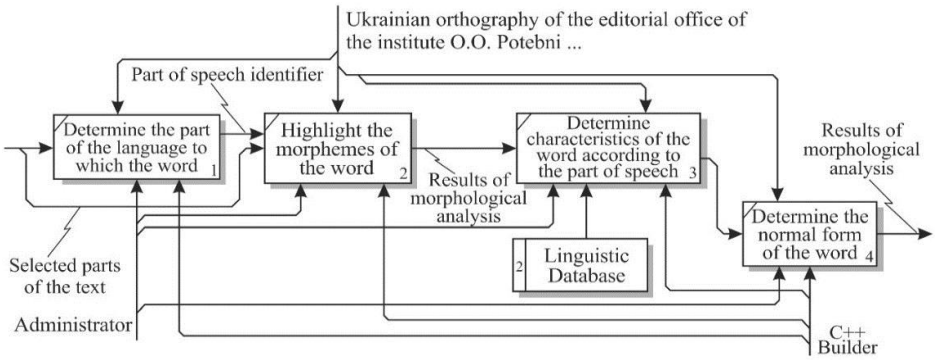


Fig. 19. Decomposition of the process: Carry out a morphological analysis of the text for analysis

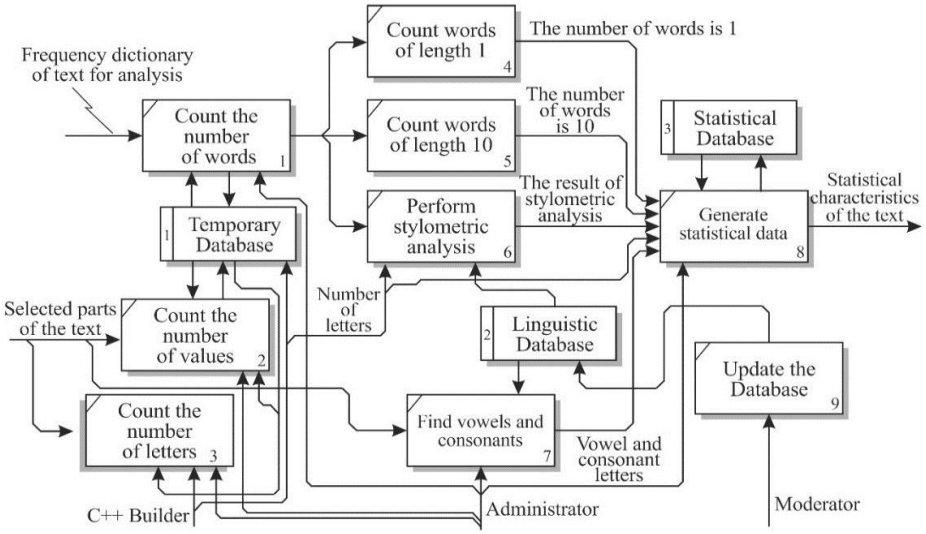


Fig. 20. Decomposition of the "Collect statistical data" process

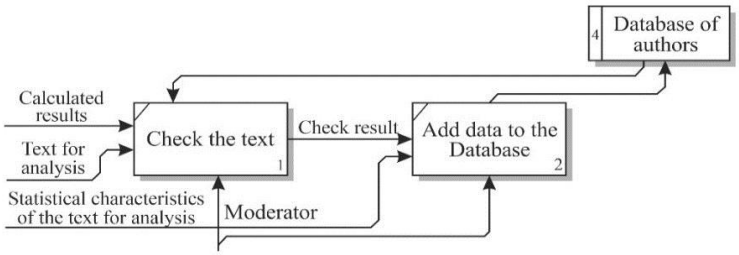


Fig. 21. Decomposition of the "Add data to the database" process

Input data: a fragment of the work "Cross Paths" by I.Y. Franko selected from the list of all possible Ukrainian authors (Ivan Franko, Lesya Ukrayinka, Ivan Nechuy-Levyts′kyy, Panas Myrnyy, Pavlo Zahrebel′nyy, Borys Hrinchenko, Marko Vovchok, Ulas Samchuk) of approximately the same period. The result of the program is shown in Fig. 22.

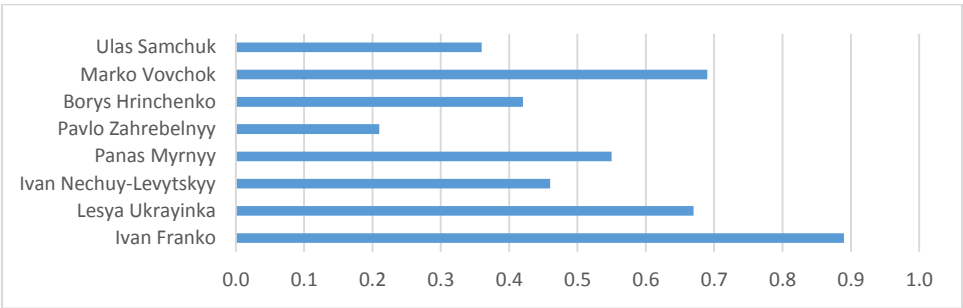


Fig. 22. Histogram of probabilities belonging to the analysis text to all selected authors

The proposed solution is a flexible (Agile) intelligent platform for the automatic determination of text authorship. It combines modern methods of natural language processing (NLP), machine learning (ML), and artificial intelligence (AI) to analyze the stylistic, lexical, and semantic features of the text and identify its author.

The solution provides **an adaptive approach** to text analysis, which allows for continuous improvement of the system with the help of new data and methods. The main features are:

- **Flexibility and scalability** are the ability to work with different languages and types of texts.
- **Modularity** – the use of independent components for data collection, processing, analysis and classification.
- **Automated Learning** – Using Self-Learning Models to Improve Identification Accuracy.
- **API integration** – the ability to integrate with other systems, such as content review platforms or cybersecurity services.

The key technologies of NLP for automatic determination of text authorship are morphological and syntactic analysis, writing style analysis, text vectorization (TF-IDF, word embeddings) and identification of stylistic features (n-gram analysis, frequency analysis of words, punctuation analysis).

The key ML and AI technologies for automatic determination of text authorship are classification algorithms (SVM, Random Forest, Naïve Bayes, Logistic Regression), Deep Learning, RNN (Recurrent Neural Networks) and Transformers (BERT, RoBERTa)), as well as hybrid approaches (combining classical ML algorithms with neural networks to improve accuracy).

Models for identifying authorship are stylometry (analysis of linguistic features), graph methods for identifying relationships between texts and analysis of latent semantic structures.

Architectural solution for a flexible (Agile) intelligent platform for automatic determination of text authorship:

Collection and Pre-Processing Module → NLP Analysis Module → Machine Learning Module → Forecasting and Decision-Making Module → API & Integration

1. The collection and pre-processing module consists of the processes of receiving text data, removing redundant characters, tokenization, lemmatization, language detection, and text cleaning.
2. The NLP analysis module consists of the processes of highlighting stylistic features, structural and syntactic analysis, and determining unique patterns of authorship.
3. The machine learning module consists of data processing processes using ML and AI algorithms, using pre-trained models, and training and updating models on new data.
4. The Prediction and Decision-Making module consists of processes for assessing the likelihood of a text being relevant to a particular author, visualizing the results, and making recommendations (e.g., whether the text is suspiciously similar to others).
5. API and integration consist of the processes of providing access to the model through REST API and integration with other services (search engines, anti-fraud systems, legal services).

Agile development is based on **Scrum or Kanban** to quickly adapt the solution to new challenges. Main stages:

1. Research and development (1-2 sprints) consists of analyzing requirements, building an MVP (Minimum Viable Product), and choosing an initial technology stack.
2. Prototyping (3-4 sprints) consists of creating the first version of the NLP module, training and testing ML models, and developing basic stylometric analysis algorithms.
3. Iterative improvement (5-6 sprints) consists of optimizing algorithms, introducing new AI models, and adding APIs to integrate with other services.
4. Deployment and support consists of deploying in a cloud environment, testing on actual data, and continuously updating models.

Expected results:

- **Identification accuracy** at the level of 85-95% due to a combination of classical and deep ML methods.
- **Flexibility** – the system can adapt to different genres of texts (news, journalism, technical documents).
- **Automatic learning** – continuous improvement of models through feedback.
- **Scalability** – support for different languages and integration with analytics platforms.

The proposed Agile solution allows you to effectively identify the authorship of text based on NLP and ML. It provides flexibility, scalability, and high accuracy, which makes it relevant for the field of digital security, academic integrity, and fake news investigations. Experiential testing of text analysis for effective and more accurate determination of authorship attribution led to the understanding that the system project should have the following characteristics: using the lexicographic analysis method, the characteristics of the entire text should be determined, not a separate sentence/fragment of the text; the description of the text should cover different levels of the language system, and in addition to the linguistic composition of the text, its structure should also be considered; the use of multidimensional classifications is necessary. In the future, there is a need to continue the research, mainly to increase the set of reference texts, implement multi-platform ILS, and increase the number of supported languages. There is also a need to implement other methods of author attribution and, if necessary, implement a combination of such techniques to increase accuracy. Fig. 23-25 demonstrates the features of generating a set of probable keywords compared to an author set. During the experimental testing, an analysis of more than 300 excerpts of texts of one-person (≥100 authors) scientific and technical publications of the Bulletin of Lviv Polytechnic University of the "Information Systems and Networks" series for the period 2001–2021 was carried out (Table 4 and Fig. 26).

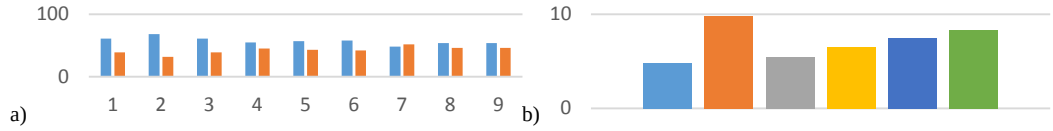


Fig. 23. Analysis of verification of 100 articles (explanation in Table 3): a) blue– less than the average value, orange – more than the average value; b) blue - author's keywords, orange - quantity, grey - stage 1 of step 1, yellow - stage 1 of step 2, light blue - stage 2 of step 1, green - stage 2 of step 2

Table 3. Statistical data as an explanation for Fig. 23

Chart column name	Marking	Arithmetic average number of keywords	
		Explanation	Value
Author's keywords	A_1	defined by the author	4.77
Number of words	A_2	contain author's	9.82
Stage 1, Step 1	A_3	probable keywords found by the module at stage X and step Y (Fig. 5.9-Fig. 5.10)	5.46
Stage 1, Step 2	A_4		6.51
Stage 2, Step 1	A_5		7.43
Stage 2, Step 2	A_6		8.35

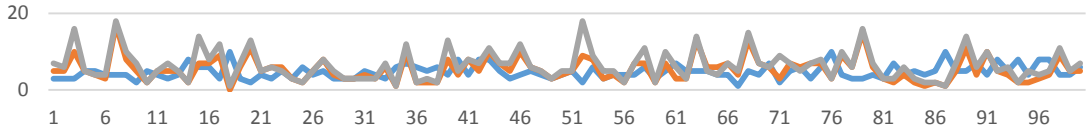


Fig. 24. Obtaining meaningful words during text processing at a) stage 1, step 1, b) stage 1, step 2, c) stage 2, step 1 and d) stage 2, step 2, where blue – all words, orange – meaningful words, grey – match with author's, yellow – accuracy of the match, light blue – additional words

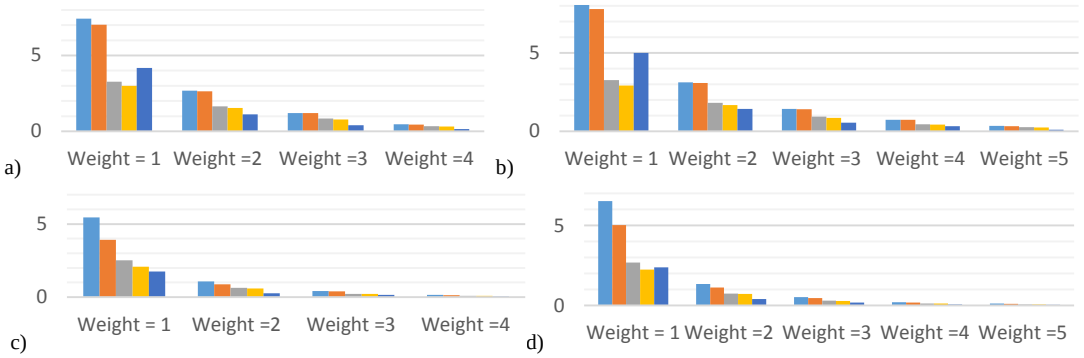


Fig. 25. Obtaining meaningful words during text processing at a) stage 2, step 1 and b) stage 2, step 2, c) stage 1, step 1, d) stage 1, step 2, where blue – all words, light blue – additional words, yellow – accuracy of the match, grey – match with author's, orange – meaningful words

Table 4. An example of analysing the author's style of speech

Result	Degree	Calculation
W_{10}=2; W=184	concentration: $I_{kt}=W_{10}/W$	I_{kt}=0.01
W_1=141; W=184	exclusivity: $I_{wt}=W_1/W$	I_{wt}=0.7663
P=18; W=184	syntactic complexity: $K_s=1–P/W$	K_s=0.902
W=184; N=295	lexical diversity: $K_l=W/N$	K_l=0.6237
Z=20; S=28; P=18	coherence of speech: $K_z=(Z+S)/(3*P)$	K_z=0.889

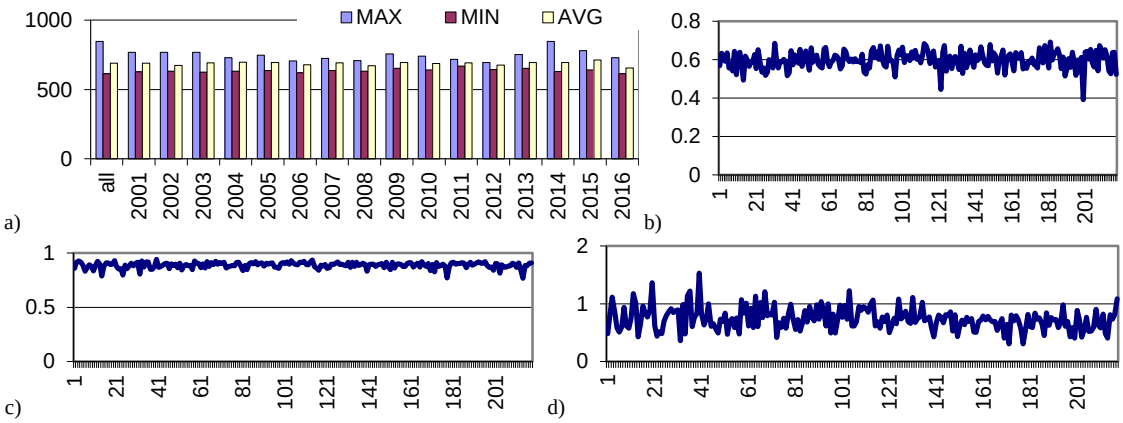


Fig. 26. Distribution: a – words and speech parameters for texts of the same volume in the range [2001; 2021] year: b – K_l; c – K_s; d – K_z

The degree of speech connectivity K_z does not decrease significantly. In 2001, it changes within [0.5; 1.2], and in 2021 – within [0.4; 0.9] (Fig. 27). The distribution does not change significantly over time for the exclusivity parameter I_{wt}, and there are significant changes for the concentration parameter I_{kt} (Fig. 28). For example, over time, authors for a specific topic of research use some service words more often in their publications (Fig. 29-34).

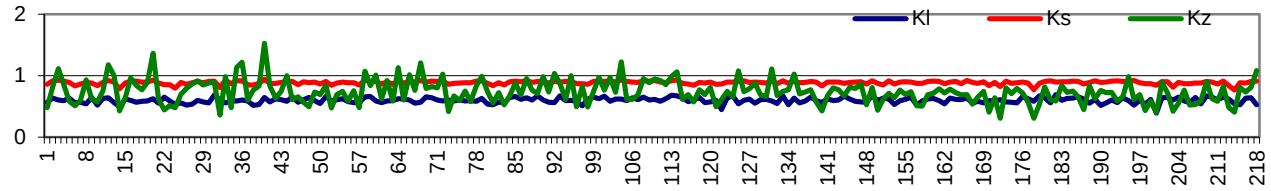


Fig. 27. Analysis of the distribution of speech style parameters K_l, K_s and K_z

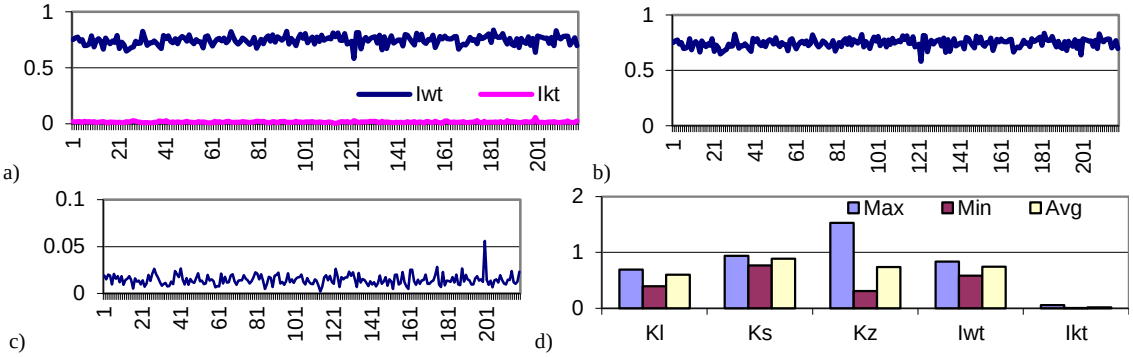


Fig. 28. Distribution of the degree of speech for a – both parameters; b – I_{wt}; c – I_{kt}; d is the minimum, maximum and average value for all parameters

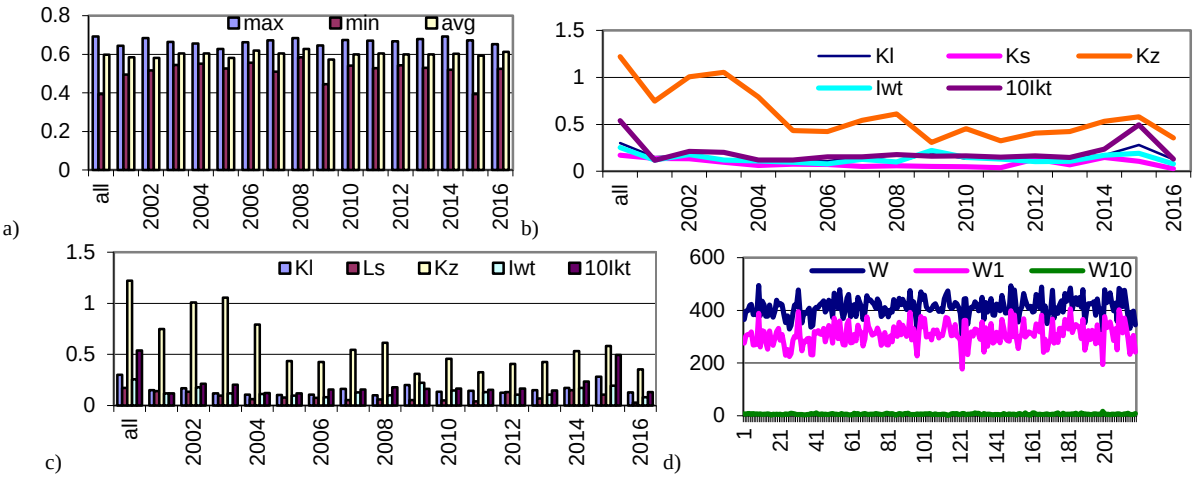


Fig. 29. Distribution of speech parameters for texts of equal volume in the range of 2001–2017 years: a – maximum, minimum and average value K_l; b,c – change of parameter values; d – the appearance of word forms (all, only 1 time and ≥10)

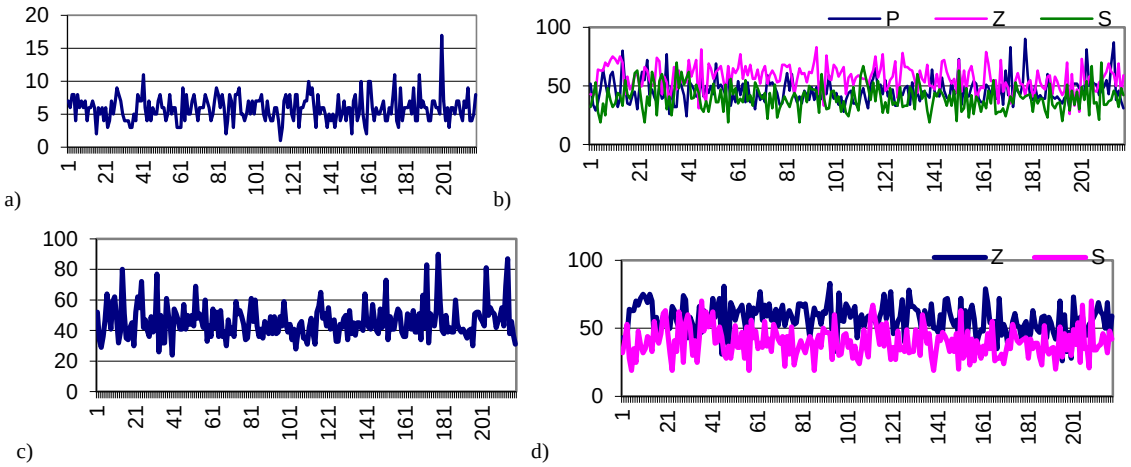


Fig. 30. Distribution of occurrence of words: a – ≥10 (W_{10}); b – the degree of speech connectivity; c – a sentence; d – prepositions and conjunctions

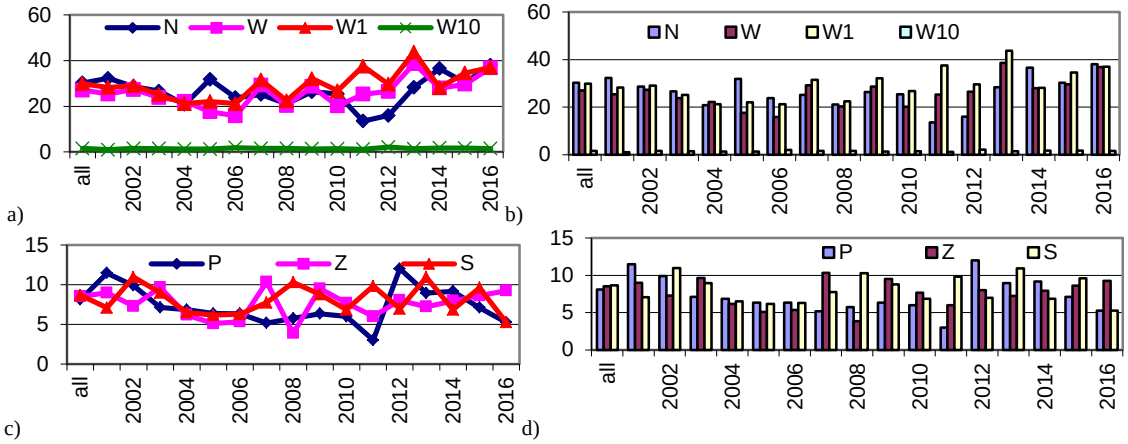


Fig. 31. Changes in the distribution of features of the author's speech style over time

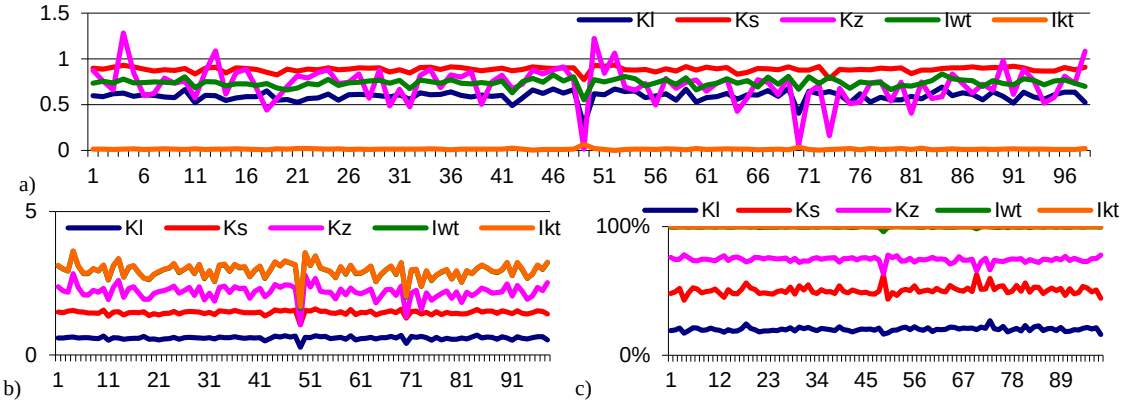


Fig. 32. Study of change over time according to the features of speech: a – identification of the author's style; b – total amount; c – value embedding based on normalization

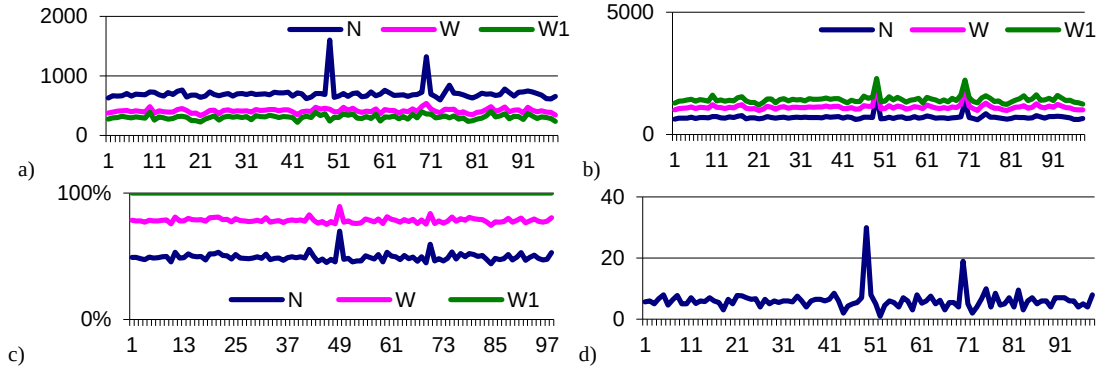


Fig. 33. Changes study in time according to speech features: a – the author's style identification; b – total amount; c – value embedding; d – W_{10} changes

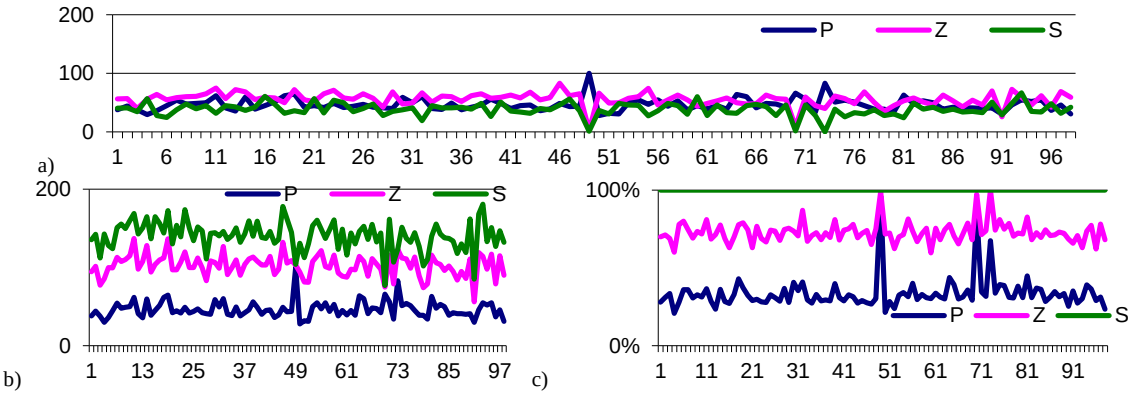


Fig. 34. Study of changes in time according to the speech features: a – identification of the author's style; b – total amount; c –the nesting of each value

The study implemented and tested several machine learning models for classifying the authorship of texts, in particular **SVC, Random Forest, Naive Bayes, and Logistic Regression.** To prepare the data, the **TF-IDF technique was used**, which allows you to weigh words according to their significance in the text.

The SVC **classifier** uses a hyperplane to separate data in a multidimensional space. The selection of the optimal hyperplane is performed using the margin maximization method.: $min_{w,b}\frac{1}{2}||w||^2 w, b$ with restrictions: $y_i(w \cdot x_i + b) \geq 1,\ \forall i$, where w is the vector of weights, x_i is signs of input text, y_i is a class label, b is offset. For non-linearly resolution data, kernel tricks are used, in particular the radial-basis function (RBF):

$$K(x_i, x_j) = \exp\left(-\gamma \left||x_i - x_j|\right|^2\right).$$

Random Forest builds an ensemble of decision trees, combining their predictions through voting. Main stages of the algorithm:

1. Creating **bootstrapped** subsamples from a training set.
2. Building **decision trees** using random feature matching.
3. Voting among trees: $\hat{y} = \arg\max_k \sum_{i=1}^{N} 1(y_i = k)$, where N is the number of trees, k is class.

Na ïve Bayes is based on **Bayes' theorem**:

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)},$$

where C_k is class (author of the text), X is features vector (TF-IDF), $P(C_k)$ is a priori probability of a class, $P(X|C_k)$ is plausibility. Multinomial Naive Bayes was used for text data.

Logistic Regression simulates the probability of belonging to a class using **a sigmoid function**:

$$P(y = 1|x) = \frac{1}{1 + e^{-z}}$$

where $z = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$. Parameters are optimized by w**gradient descent**.

9. Results

This research explores the intersection of ML and NLP methods to develop a Python program capable of identifying the authorship of scientific and technical publications through stylistic analysis. This section analyses the results obtained after applying ML models to determine the author's style in scientific and technical publications. Various pre-processing steps were used to classify the texts to ensure the data was in a suitable format for ML algorithms. These steps included tokenisation, removal of stop words, basic form, and converting the text into a numerical representation using the TF-IDF method. The dataset was then split into training and testing sets to evaluate the performance of the models. The following machine learning models were trained and tested on the dataset: Support Vector Classifier (SVC), Naive Bayes (NB), Logistic Regression (LR) and Random Forest (RF). The performance of each model was measured using accuracy, recall, precision, and F1-score. The results of these metrics are summarised below for the models (Fig. 35).

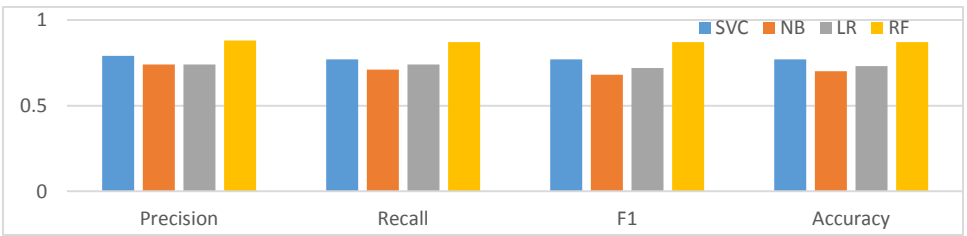


Fig. 35. Evaluation metrics results for machine learning models

The results show that the Random Forest classifier outperforms the other models in all evaluated metrics, achieving the highest accuracy (0.88), recall (0.87), F1 measure (0.87), and precision (0.87). It shows that Random Forest is particularly effective in capturing distinctive features of the writing styles of different authors in scientific and technical texts. The SVC model demonstrated quite good performance with an accuracy of 0.77. The accuracy and sensitivity scores suggest that while SVM is generally good at identifying correct classes (authors) for texts, there is room for improvement in reducing false negatives. Naive Bayes showed the lowest performance among the models,

with an accuracy of 0.70. The lowest F1 measure (0.68) indicates that the Naive Bayes model has more difficulty balancing accuracy and sensitivity than the other models, as the F1 measure is the harmonic mean of the two. This model makes the most assumptions about the distribution of classes that do not fit the input data set. Logistic regression performed moderately well with a precision of 0.73 and a balanced F1 measure of 0.72. Although logistic regression is a reliable model for this classification task, it is not as reliable as random forest or SVC. The higher performance of the Random Forest classifier can be attributed to its complex nature, which combines the prediction of multiple decision trees to improve overall performance and reduce model overfitting. It allows you to capture complex patterns in the data that simpler models like Naive Bayes might miss. Although simpler models such as Naive Bayes are faster to learn, they may not always provide the best accuracy in text classification. On the other hand, more sophisticated models such as Random Forest and SVC can significantly improve accuracy and other metrics, although they may require more computing resources.

An important task in determining the author of the text is the collection and processing of data for training. For the final training and comparison of the results of the previous study, it was decided to analyze the English-language dataset of popular Twitter accounts. Text snippets for 10 different authors were uploaded and processed, and the eleventh grade is a set of random Twitter entries as large as all other classes.

Table 5. Description of the collected data

Author	Short description	Number of specimens
Donald Trump	Politician, businessman, 45th President of the United States	1000+100
NASA	Information related to space, planets	1000+100
Kim Kardashian	Model, actor	1000+100
FiveThirtyEight	Site with news about politics, economics	1000+100
Barack Obama	Politician, 44th President of the United States	1000+100
Richard Dawkins	Biologist, popularizer of science	1000+100
Adam Savage	Engineer and teacher, host of "Mythbusters"	1000+100
Hillary Clinton	American politician	1000+100
Neil deGrasse Tyson	Astrophizic, popularizer of science	1000+100
Scott Kelly	Astronaut, Capitan USA	1000+100
Random Tweets	Random tweets from popular personalities	1000+100

Let's visualize the most popular tokens for some authors: NASA channel, Donald Trump, and Adam Savage. We can see that the most popular words are related to space, various specific terms (Fig. 36a), and the hashtags "NASA" and "gov". Since the data has not been cleaned beforehand, the most popular terms include various tags, such as pic and twitter. As expected, the politician uses (Fig. 36b) many hashtags and political slogans such as "realDonaldTrump" and "Make America great again" to spread as much as possible. As in the previous example, all kinds of technical lexemes are also present here. Inherent in the hashtags of the host of the television show "MythBusters" are "MythBuster" and "Jamie" - his partner, engineer, and singing host of the show. Since the data is not cleared, and only the frequency is used, not the ratio of frequency to the inverse frequency of the document, the same tokens popular for Twitter are found in all three examples. In order to clearly show the lexemes that are inherent in one author (let's choose Donald Trump for this), let's take two critical steps:

1. Let's add to the process of removing stop words, that is, those that are used to build a sentence and do not carry much meaning.
2. Instead of passing text to the cloud-building function, we will pass the text along with the TF-IDF indices.

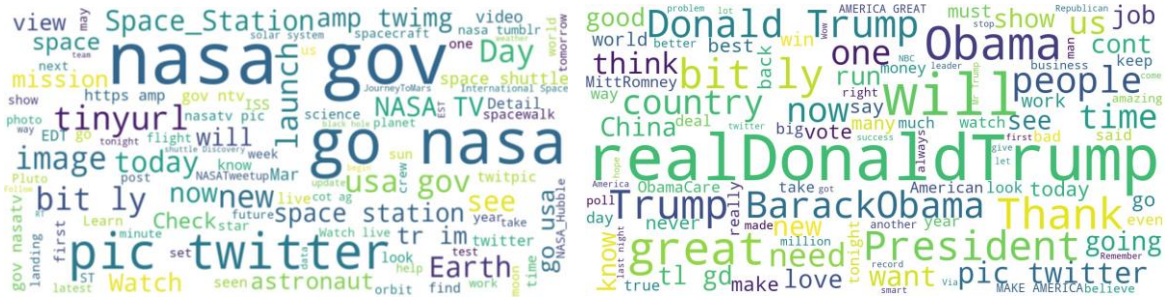


Fig. 36. The most popular tokens from the NASA dataset and the most popular tokens from Trump's account

Fig. 37. The most popular tokens from Adam Savage's account and lexemes inherent in Donald Trump's texts after a preliminary study of the text

After preliminary processing of text data, the author's writing style is drawn much more clearly, which once again proves the need for correct work with the text in order to prepare it for use in the task of determining the author of the text. Pre-collected data is placed in a table that looks like Table 6. To use this type of data to train a deep learning model, using the attention mechanism, the data must go through the pre-processing stage and be fed to the model input in a particular form. There are two things to keep in mind when preparing:

1. Pre-processing of data should not reduce the size of information about the author's style and the manner of his posts.
2. The data must be formatted so that it can be submitted for elaboration of the deep learning model. In the future, we will consider what exactly is needed and how to achieve it.

Table 6. Example of collected data

Author	Text
DonaldTrump	Is this really America? Terrible!
NASA	New software on the @Space_Station will make data communications faster and easier for hundreds of scientists: http://go.nasa.gov/2dQrLto
KimKardashian	The Yeezy Show Room in Paris is heaven.
various	Top Data Scientist @Claudia_Perlich on Biggest Issues in #DataScience via @kdnuggets

Since the deep learning model with the BERT attention mechanism was chosen for the final training, we will rely on the preparation for data training for this particular model.

First, import the model and see what the input text will look like after passing through the tokenizer. The text we will test is "This is a simple test for my diploma work. Another sentence also here». Tokenizer output: *['This', 'is', 'simple', 'test', 'for', 'our', 'article', '.', 'Another', 'sentence', 'here', '.']*. Details such as turning off the automatic conversion of uppercase letters to lowercase letters are important to us, as well as the fact that the tokenizer leaves word order and punctuation. To convert the basic tokenizer into one that can be effectively used for the task of determining the author of the text, you need to perform several essential steps:

1. Add unique tokens that will indicate the beginning and end of a piece of text. Add the [CLS] token at the beginning and the end of each sentence to the [SEP] token. Now our example will look like this: *[[CLS], 'This', 'is', 'simple', 'test', 'for', 'our', 'article', '.',[SEP], 'Another', 'sentence', 'here', '.',[SEP]]*.
2. Since all sentences and text fragments have different lengths, and a vector of the same size must be fed to the input of the model, we have two options. Either to cut the sentence to a specific size, which is unacceptable in the problem of determining the author of the text. Or fill all empty values with [PAD] tokens, which will allow us to avoid losing information about the author's style and will not affect the model's training process. The fragment contains 15 tokens, and the length of the vector fed to the input of the model is 512. So, starting with 16 tokens, each will be equal to [PAD]. This token also belongs to the special class and will have an id equal to zero. *3.* Attention mask. An attention mask is simply a vector of length 512, the values of which are equal to one if the token corresponding to this index is not equal to [PAD] and zero if it is.
3. Sequence coding. Sequence encoding is an operation that performs the three previous steps and converts text tokens into their corresponding identifiers. Text sequence, after encoding, are [101, 1188, 1110, 3014, 2774, 1111, 1139, 14985, 119, 2543, 5650, 1303, 119, 102], where the identifiers 101 and 102 correspond to the [CLS] and [SEP] tokens.
4. Data is divided into training, validation, and testing. One thousand copies of each of the 11 classes were allocated for training, 1000 for validation, and 1000 copies for final testing.

Thus, we have a complete algorithm for converting data to feed them in the desired form into a neural network. The next step is to configure the parameters of the neural network and the entire training process. For training, the BERT architecture model, bert-based-cased, was chosen. For convenience, we will display the model parameters for three different layers: the transformation layer, the "first transformer", and the final layer. Transformation layer:

bert.embeddings.word_embeddings.weight	(28996, 768)
bert.embeddings.position_embeddings.weight	(512, 768)
bert.embeddings.token_type_embeddings.weight	(2, 768)
bert.embeddings.LayerNorm.weight	(768,)
bert.embeddings.LayerNorm.bias	(768,)

The layer of the "first transformer" that accepts data from the previous layer as input:

bert.encoder.layer.0.attention.self.query.weight	(768, 768)
bert.encoder.layer.0.attention.self.query.bias	(768,)
bert.encoder.layer.0.attention.self.key.weight	(768, 768)
bert.encoder.layer.0.attention.self.key.bias	(768,)
bert.encoder.layer.0.attention.self.value.weight	(768, 768)
bert.encoder.layer.0.attention.self.value.bias	(768,)
bert.encoder.layer.0.attention.output.dense.weight	(768, 768)
bert.encoder.layer.0.attention.output.dense.bias	(768,)
bert.encoder.layer.0.attention.output.LayerNorm.weight	(768,)
bert.encoder.layer.0.attention.output.LayerNorm.bias	(768,)
bert.encoder.layer.0.intermediate.dense.weight	(3072, 768)
bert.encoder.layer.0.intermediate.dense.bias	(3072,)
bert.encoder.layer.0.output.dense.weight	(768, 3072)
bert.encoder.layer.0.output.dense.bias	(768,)
bert.encoder.layer.0.output.LayerNorm.weight	(768,)
bert.encoder.layer.0.output.LayerNorm.bias	(768,)

The last layer, which takes data from the previous one as input and gives the probability of distributing 11 classes from the training set:

bert.pooler.dense.weight	(768, 768)
bert.pooler.dense.bias	(768,)
classifier.weight	(11, 768)
classifier.bias	(11,)

All experiments described below are divided into the following groups:

1. The complete experiment is described in the previous section. That is, all 11 classes were used, and 1000 copies for each. There will be two options for preliminary elaboration: regular and improved for our task.
2. Reduced the number of classes to 5 compared to the first experiment. The number of specimens remained unchanged, and five classes for training were chosen randomly.
3. Binary classification experiment: DonaldTrump class is used, which is compared to the class various, i.e. random tweets.
4. Experiment number 1, with a reduced number of specimens for each class - 500 and 100 classes for each workout. To display the change in the loss function during training, we will use the tensorboard library.

For training, we perform 120 thousand steps, which is equal to several training eras, depending on the size of the batch. On the *X* axis is the number of epochs, and on the *Y* axis is the average value of the loss function. The translucent graph shows the value of the loss function without smoothing (Fig. 39a). From the results of the training, it can be seen that the most pronounced author's style and specific topics for tweets are used by the site FiveThirtyEight, NASA and Donald Trump. Also, the model recognizes with high accuracy the class that contains random tweets because they differ in writing style and content from the classes we have chosen. Next, we will look at training on the same dataset but with a different pre-processing, improved for the task of determining the author of the text (Fig. 39b).

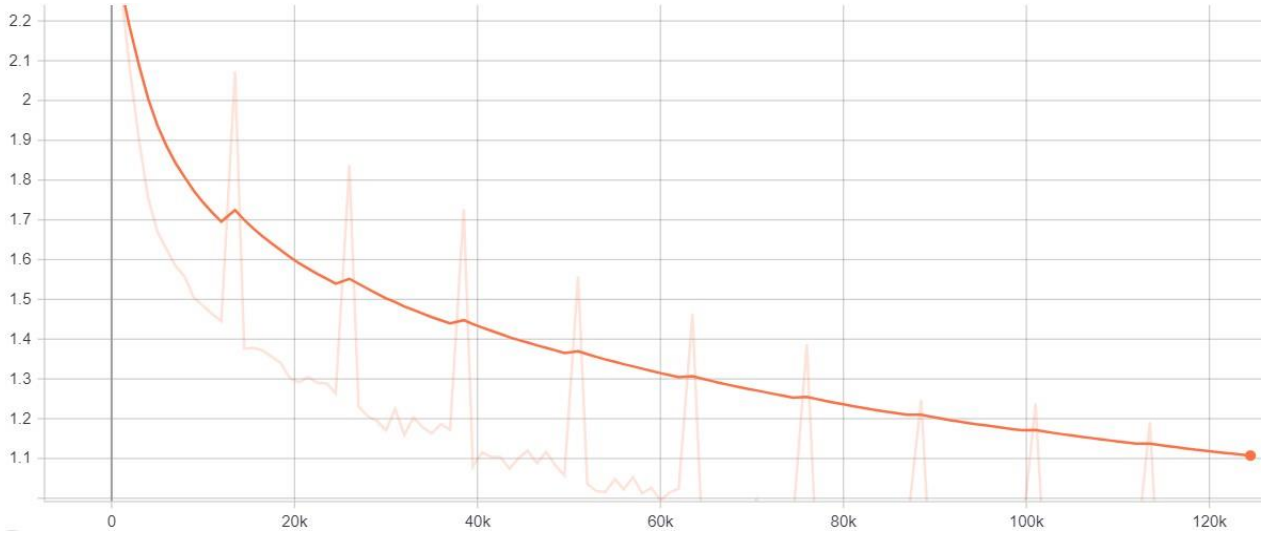


Fig. 38. Change in loss function during exercise

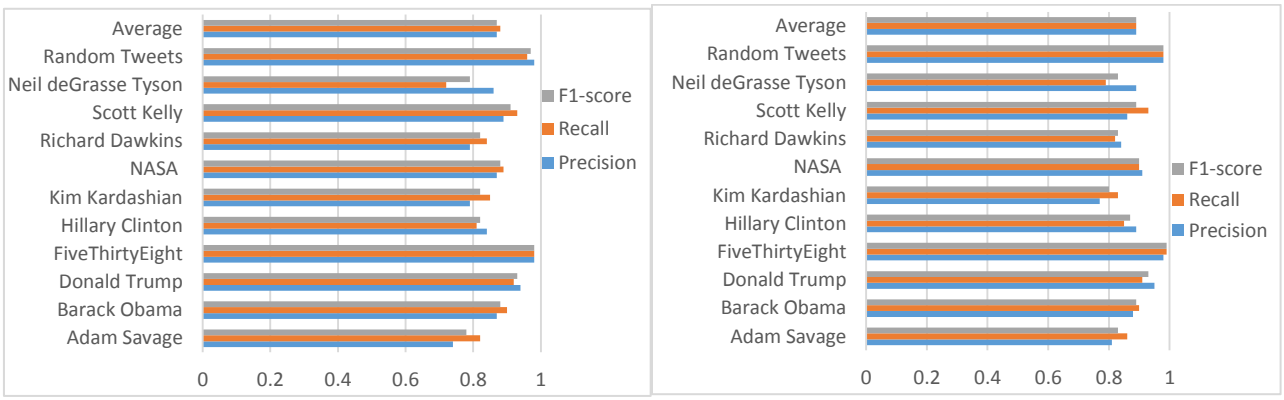


Fig. 39. Results of the first experiment and results of the first experiment, with improved data preparation

Improved pre-processing techniques have made the model more accurate. Both accuracy and completeness increased by about 2% on average. The classes with the most accurate forecasts remained Donald Trump, NASA and FiveThirtyEight. However, the classes for which the results were the lowest began to be predicted on average 0.04 more accurately.

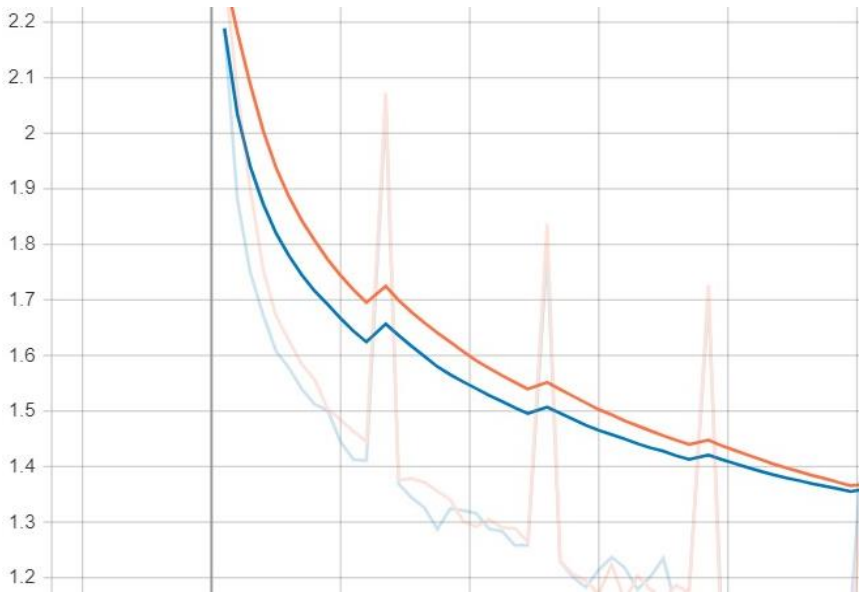


Fig. 40. Comparison of the mean value of the loss function of two experiments

During training (Fig. 41), the model using pre-processing of data converges faster and has a lower final loss value. It is because, thanks to this data processing, more information about the author and his style is stored.

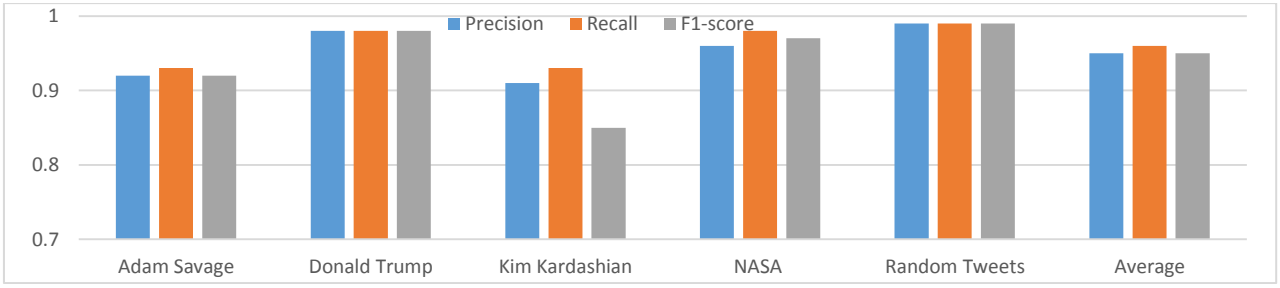


Fig. 41. Training in five grades

The following accounts were chosen for the experiment with five classes: DonaldTrump, NASA, AdamSavage, KimKardashian, various. These accounts contain entries related to different areas and perform well on the 11th-grade classification problem. We can see that the results have greatly improved due to the decrease in the number of classes and due to the fact that the authors of these instances relate to different spheres of life, often far from each other. At first, the model that trained in five grades converged worse due to the relatively high learning rate for this task (learning rate = 0.005). But starting from one hundred thousand steps, the learning outcomes of a model that studied in more classes improved.

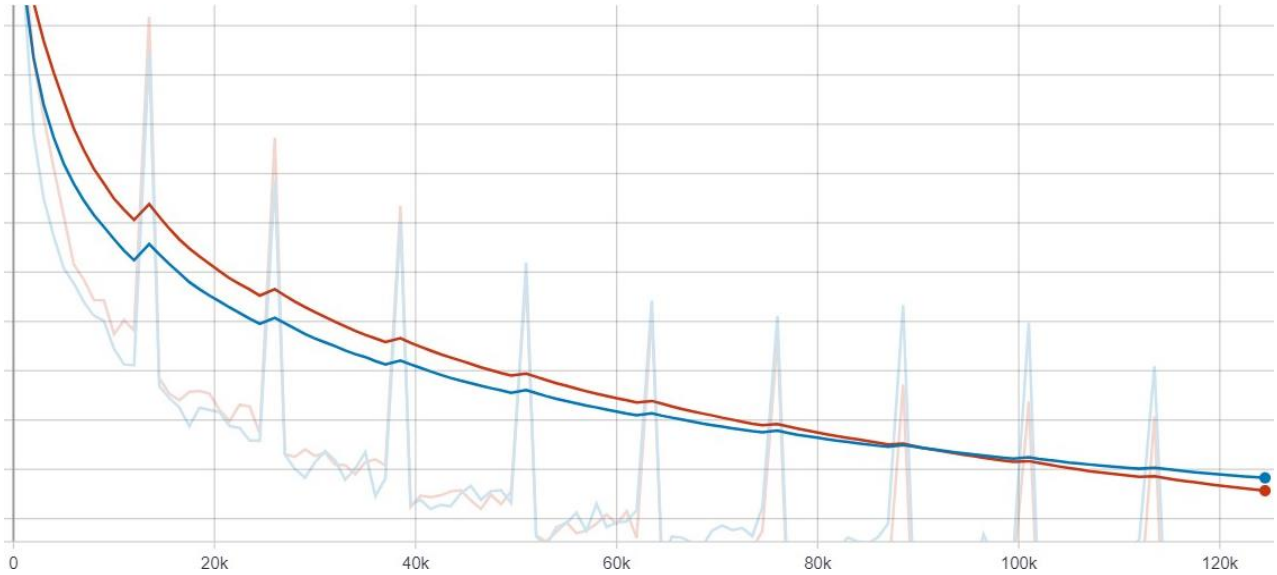


Fig. 42. Comparison of training in grades 11 and 5

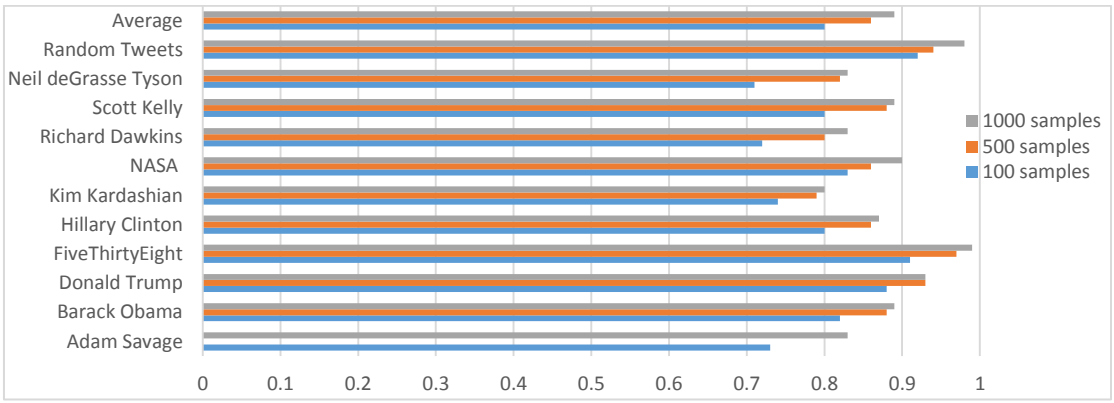


Fig. 43. Comparison of F1-score training results for 100,500 and 1000 instances per class for the classification model in 11 grades

From the results of comparing training with a different number of copies, it can be seen that the neural network still has millions of parameters. With a decrease in the data set, the results deteriorate pretty strongly. Still, even a hundred copies are enough to distinguish authors with the most pronounced writing style from each other.

To test the developed methodology, a scenario was carried out that considers a dataset with a limited number of authors, where each author is presented with a sufficient amount of training and test data. The dataset used is made up of 20 different authors with a maximum of 20 works for each and with a mixed selection of syntactic and lexical textual features. The validation of the datasets took place with a different number of articles per author, from five to twenty. The reason for this was to check how the classification for each set was performed with a different size of data for training and testing.

Four iterations of classifiers were performed for the data. The first iteration used only five scientific papers per author as training and test data. In comparison, the second iteration used ten research papers, the third iteration used 15 papers per author, and the fourth iteration used 20 texts for each author. For the scenario carried out, three different metrics were used: the success rate, which is the number of instances that are classified correctly. The second metric we used was the F measure, and the last metric that was considered was the Kappa Cohen statistic, which measures the internal agreement of the classifier. For the convenience of presenting the data graphically, the results of the classifications for each metric were presented as a percentage. The result of the accuracy of the classification with five works for each author is shown in Figure 44a. The total number of training texts is 100. The result of the accuracy of the classification with 10 works for each author is shown in Fig. 44b. The total number of training texts is 200.

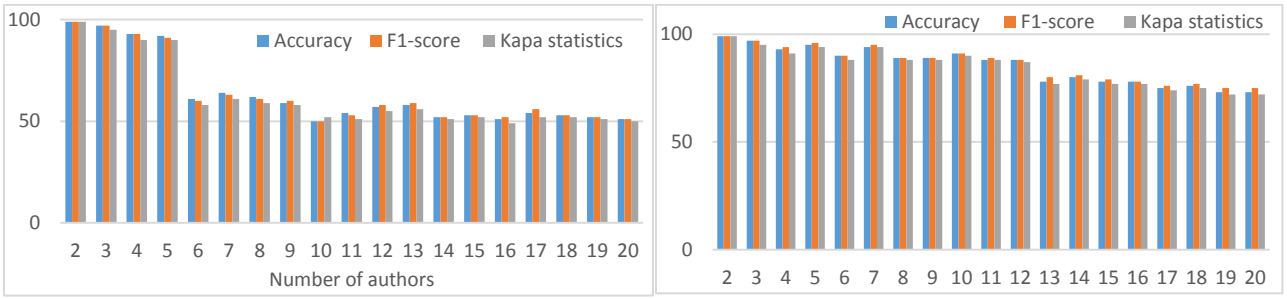


Fig. 44. Classification accuracy for five works per author and for 10 works per author

The result of the accuracy of the classification with 15 works for each author is shown in Fig. 45a. The total number of training texts is 300. The result of the accuracy of the classification with 20 works for each author is shown in Fig. 45b. The total number of training texts is 400.

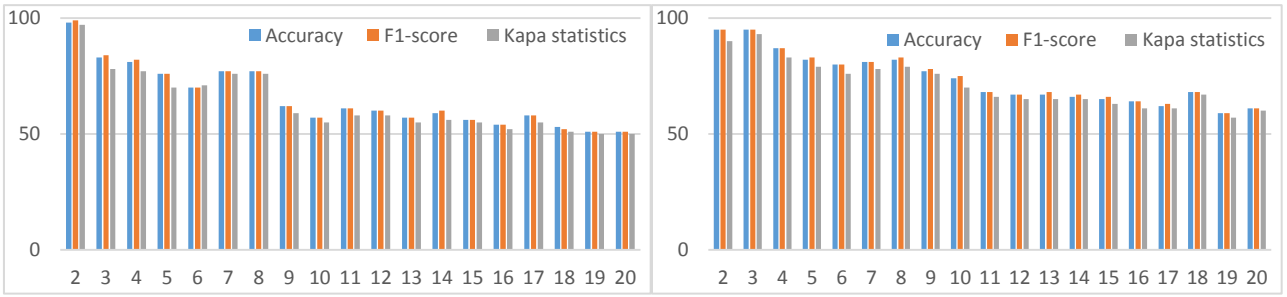


Fig. 45. Classification accuracy for 15 works per author and for 20 works per author

From the results given in Figures 44 – 45, it is observed that the algorithm of an artificial neural network with a multilayer perceptron performed well. Still, each time we added a new author, the accuracy of the classification decreased.

For a task with two authors, where the attribution of the work to one of the two authors is considered, the classifier achieves a maximum accuracy of 97% with a Kappa indicator and an F1 indicator equal to 1. As can be seen from the above graphs of results, for a situation of twenty authors, a maximum accuracy of 67% is achieved with a kappa index of 0.66 and an F1 indicator of 0.68. The main idea behind using a set of twenty authors is that this was the most challenging problem for classification algorithms, and therefore, it gives the most critical result. The results show that increasing the amount of training data for the classifier makes the classroom more efficient.

For the first dataset, we started testing with five research papers per author for our tenfold cross-validation and increased the size of the dataset to ten research papers per author. When looking at all twenty authors in the dataset, we saw an apparent increase in performance, which can be traced from the graphs. In terms of accuracy, the number of correctly classified texts improved by 61%, the F-measure by 64%, and the Kappa indicator by 72%. In general, this confirms that the performance of classifiers increases when the number of training and test cases increases.

Since the accuracy of the module's performance indicators at this stage is not high enough, it was decided to find additional solutions for this kind of system that can potentially improve the recognition results. As was already mentioned at the beginning of this section in the subsection on the result of the study of the subject area, the input text was subjected only to basic processing, which included the conversion of all letters to lower case. The first thing that should have a good effect on the result is the use of stemming in word processing. After processing the word with the stemming algorithm, the word may differ from the morphological root of the word. It will help to discard unnecessary symbols from which n-grams will actually be formed, and such symbols practically do not carry any semantic load. The next step was to remove unnecessary service characters and random characters since the input set of texts has certain hash tags, blotting, etc. The Regex (regular expression) approach was used to perform phrase parsing and the eradication of special characters. After the implementation of the proposed options that should potentially improve the result, new measurements were made according to the selected indicators to assess the quality of the model. Fig. 46 shows the results of the classification of texts after additional processing.

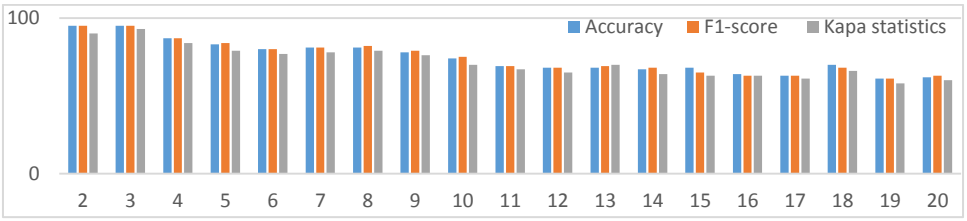


Fig. 46. Result of classification (accuracy) after additional processing of texts

For comparison, when detecting the influence of the selected descriptors on the result of detecting the author of the Ukrainian-language text, an experiment was carried out, which included the sequential addition of new descriptors to the input of the neural network. To investigate the influence of descriptors on classification accuracy, they were divided into the following sets:

- set 1, which will then be designated as N1 and will contain a set of four punctuation marks: ",", ".", "?", "!";
- set 2, which will then be designated as N2 and will contain a set of the other four punctuation marks: ":", ";", "' ", "" ";
- set 3, which will be further designated as N3 and will contain a set of sets N1 and N2: ",", ".", "?", "!", ":", ";", "' ", "" ";
- set 4, which will then be designated as N4 and will contain a set of functional words.

The result of the accuracy of the classification for each of the sets of descriptors is given in Fig. 47. The greater the number of training works used, the higher the accuracy of the classification. Therefore, the maximum number of works per author was used for this application (Ivan Franko, Ostap Vyshnya, Volodymyr Vynychenko, Marko Vovchok, Pavlo Zahrebelnyy, Hryhir Tyutyunyk, Mykola Khvylovyyy, Ivan Nechuy-Levytskyy, Anatoliy Dimarov, Lesya Ukrayinka, Mykhaylo Kotsyubynskyy, Oleksandr Dovzhenko, Lina Kostenko, Yuriy Andrukhovych, Ivan Drach, Valeriy Shevchuk, Taras Shevchenko, Valerian Pidmohylnyy, Oleh Honchar, Anatoliy Kostetskyy), namely 20 works, and the identification of the author among 20 authors. A total of 400 works were used for training. The accuracy metric was used as a metric for the classification results. Having carried out the results of the sets from Fig. 47, we can conclude that the use of syntactic textual features provides the highest classification coefficient and the lowest classification coefficient is provided by the use of lexical text features.

The writing style can be so specific and distinctive that it requires the use of less typical descriptors, such as different function words. After all the stages passed, a corpus of data was obtained, which can be used in machine learning models. But before proceeding with the training of models based on this corpus, you need to pay attention to one more small but important detail, namely that the authors (Ivan Bahryanyy, Volodymyr Vynychenko, Panas Myrnyy, Ivan Nechuy-Levytskyy, Olha Kobylyanska, Taras Shevchenko) that are in this corpus have a different number of sentences in their works, thus causing an imbalance in the corpus of data. An imbalance of data in the corpus can lead to a deterioration in the quality of model learning. There are many methods to solve the problem of imbalance, from increasing the volume of text data to reducing the number of classification objects. In this case, a data normalization technique called downsampling was used to solve this problem. It is a special technique for data processing that can be used to balance, and it is done by randomly selecting some subset of the master data. To apply this method to the data corpus, the script that converts the data into a tabular view was changed, and now it uses the downsampling technique before saving the data to the table.

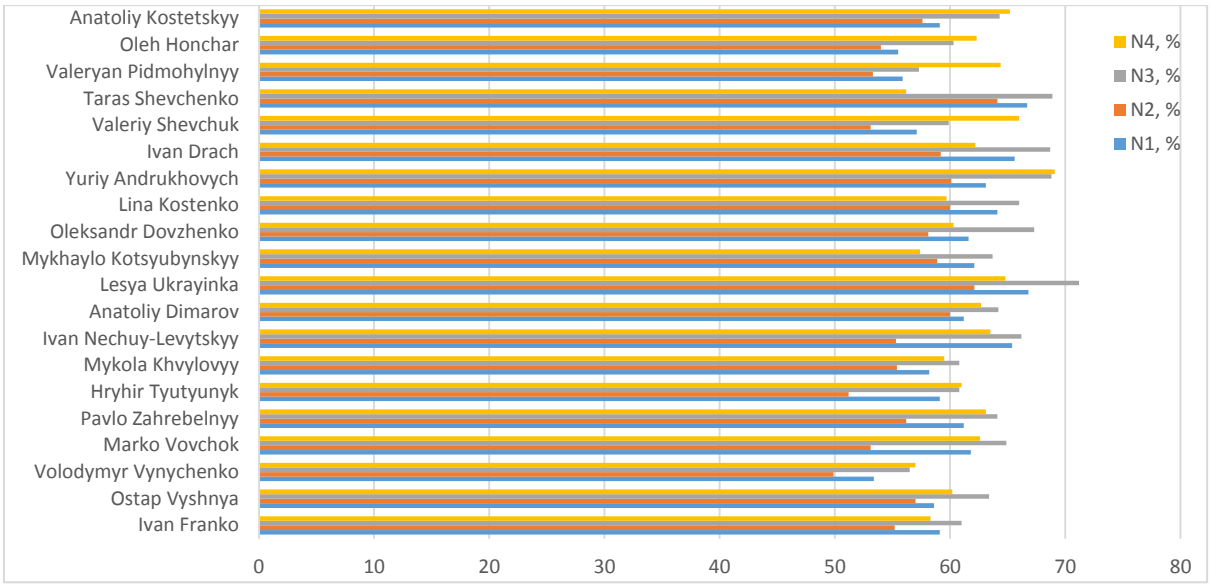


Fig. 47. Results of the accuracy of classification of texts with different sets of input models

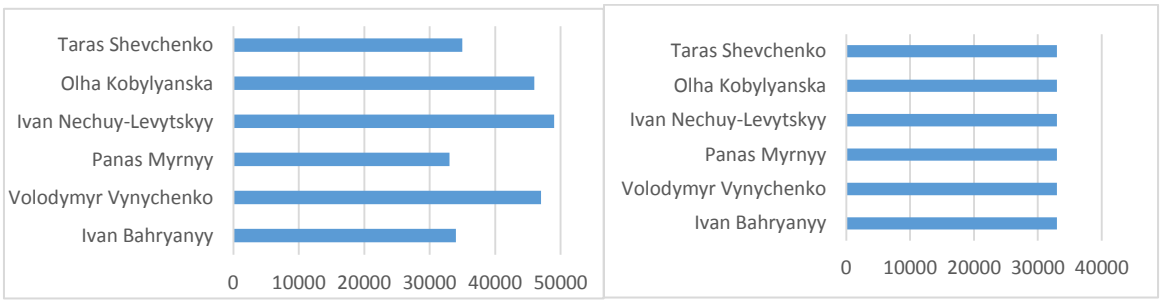


Fig. 48. Graphical display of data imbalance in the enclosure and graphical display of a balanced data corpus

After completing this process, the case can be considered entirely ready for further use in machine learning models.

The effectiveness of four classical machine learning models suitable for solving the problem of identifying text authorship was analysed. In NLP terms, this task can be classified as a multi-cluster classification problem - assigning two or more categories to a text. In this case, the category can be considered the author of the text. The effectiveness of the following models was investigated: Logistic Regression, Decision Tree, Multinominal NB and Multi-layer Perceptron. Vienna data: Corpus of authors of Ukrainian literature. Embedding methods: Bag of words and TF-IDF. Toning methods: 1-grams and 1-grams + 2-grams. The model algorithms were implemented using the Scikit-learn library. Research results:

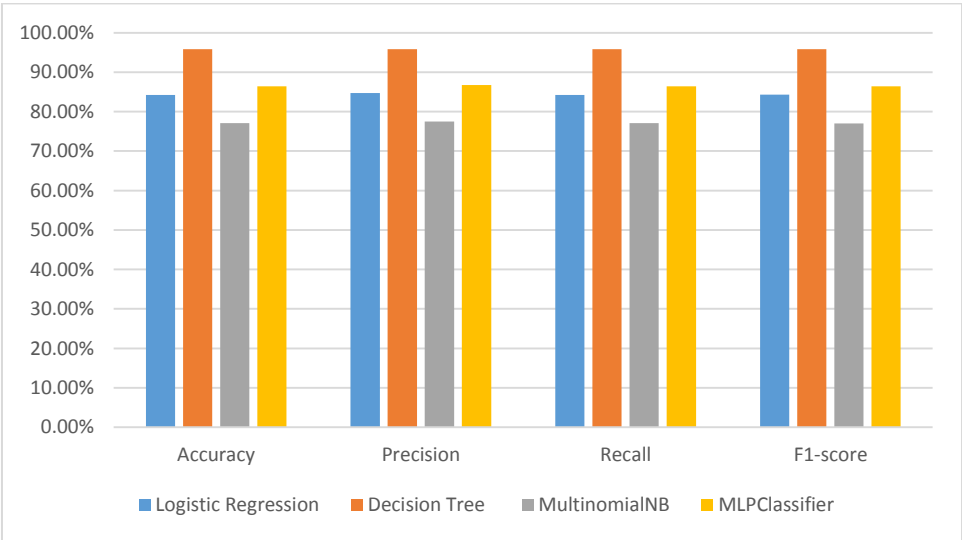


Fig. 49. Results of the Word Bag and 1-Gram Models

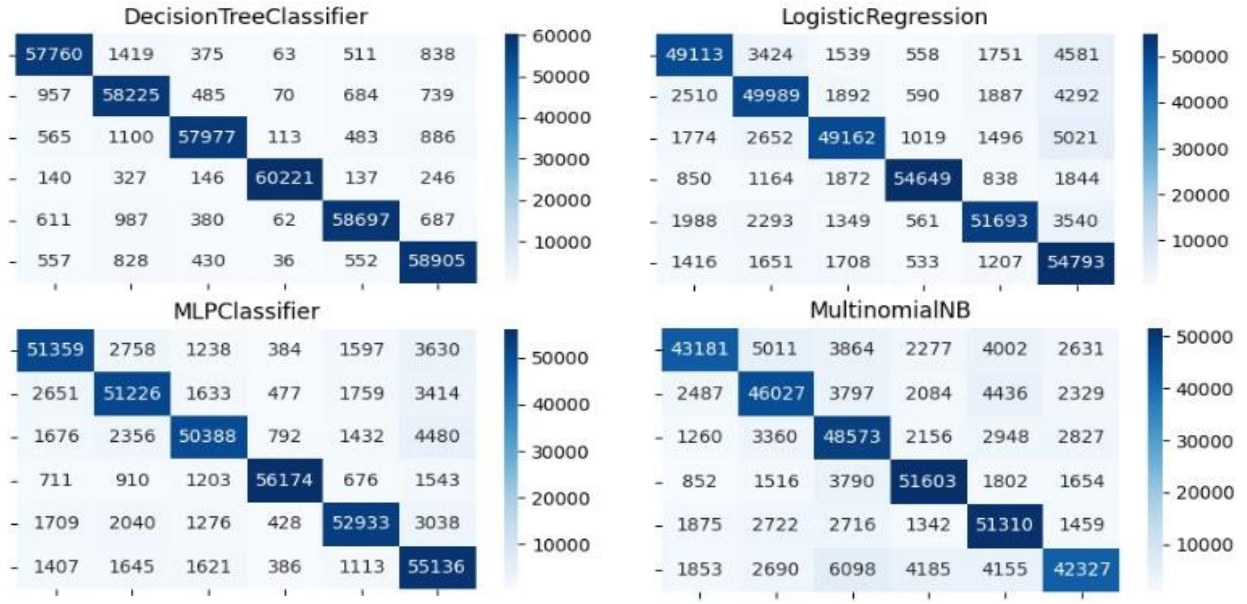


Fig. 50. Error matrix of the bag of words and 1-gram models

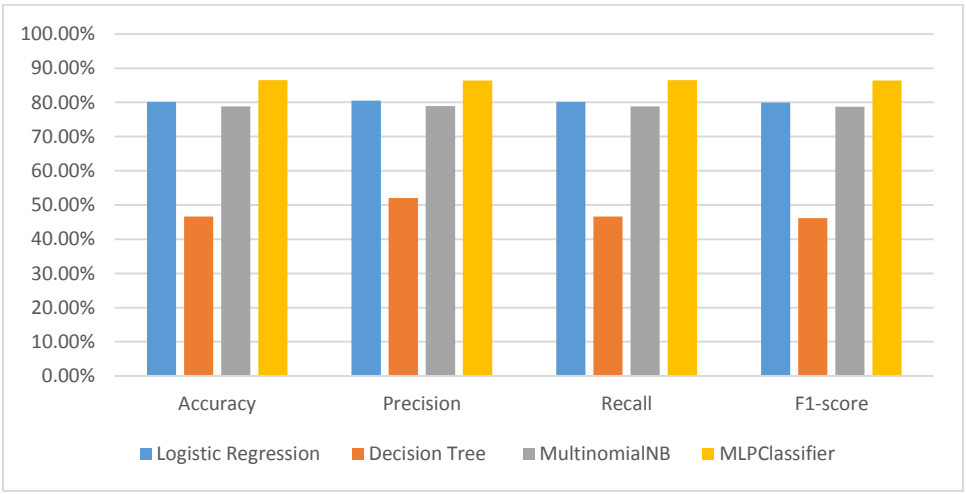


Fig. 51. Results of the Word Bag and 1-gram and 2-gram models

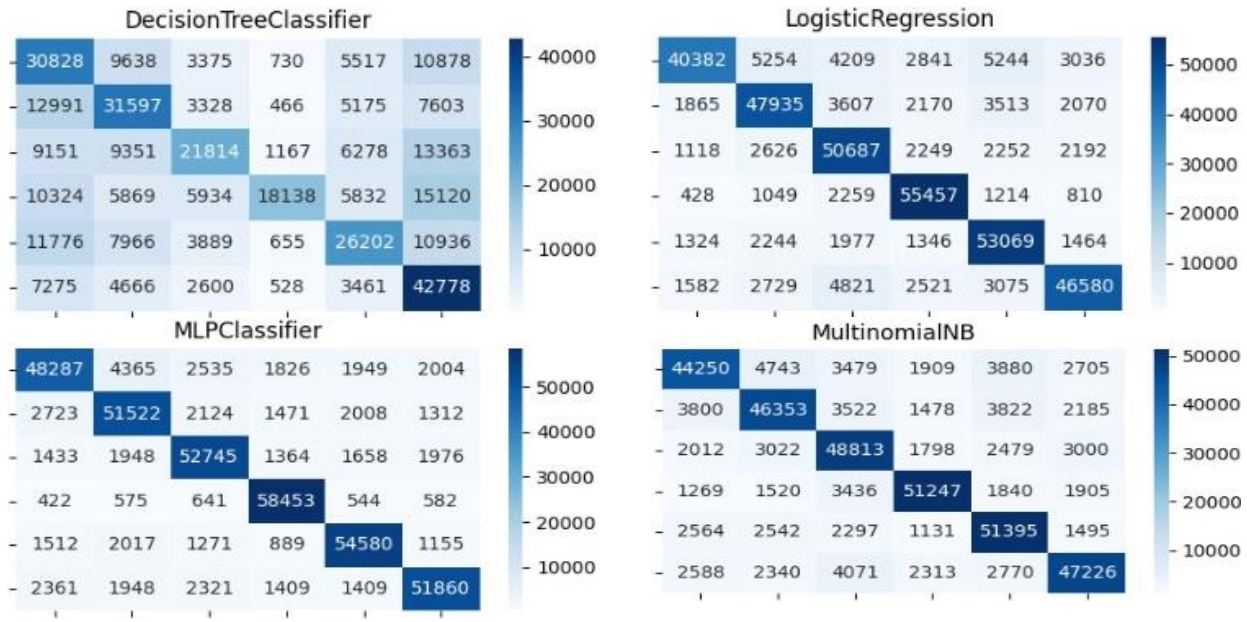


Fig. 52. Error matrix of the word bag and 1-gram and 2-gram models

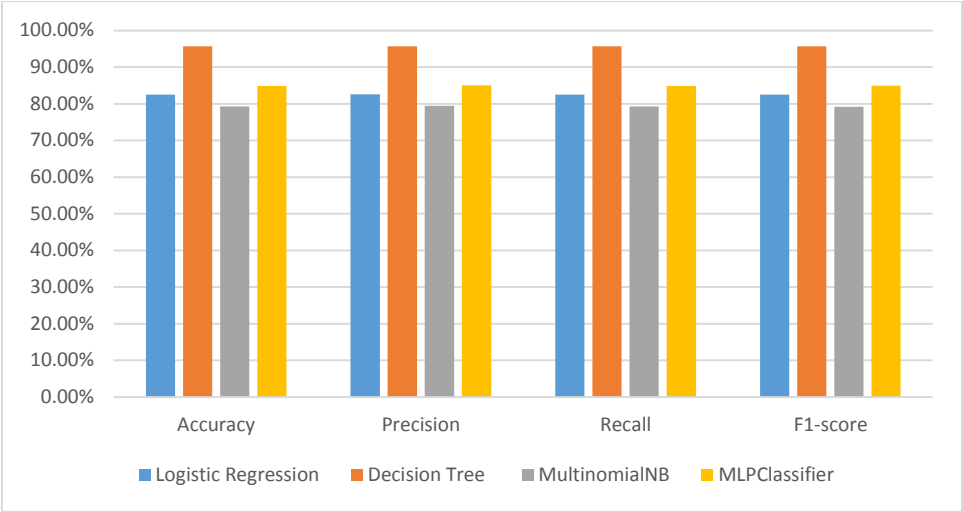


Fig. 53. Results of TF-IDF and 1-gram models

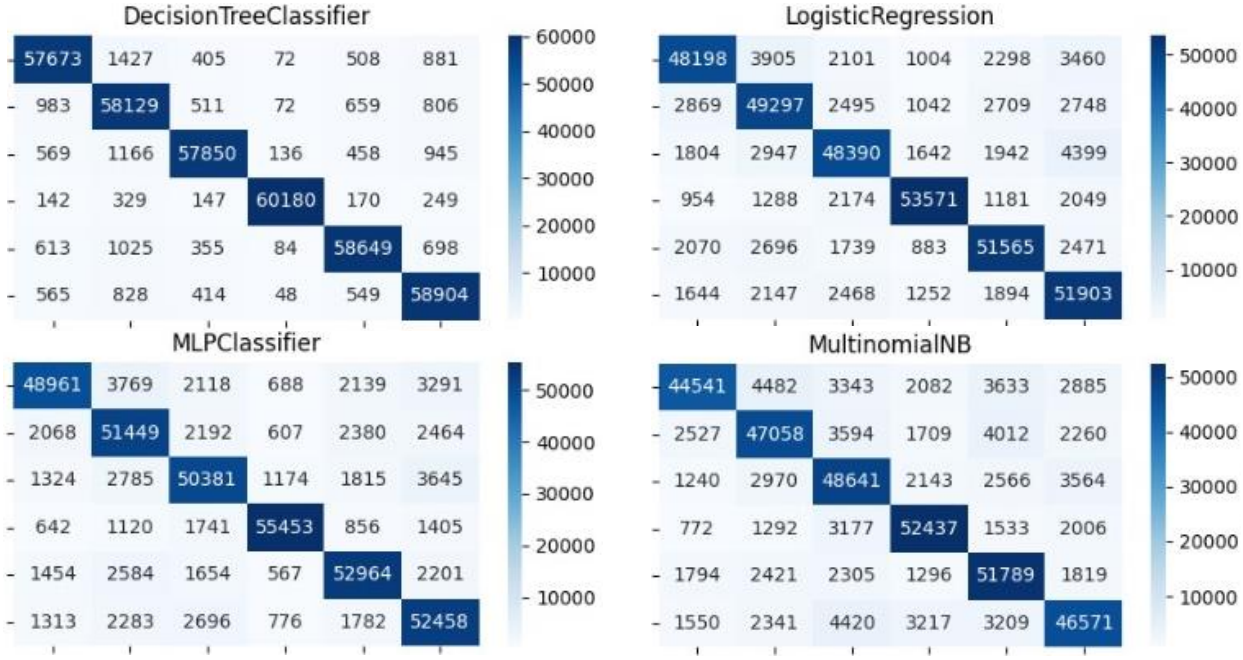


Fig. 54. TF-IDF and 1-gram model error matrix

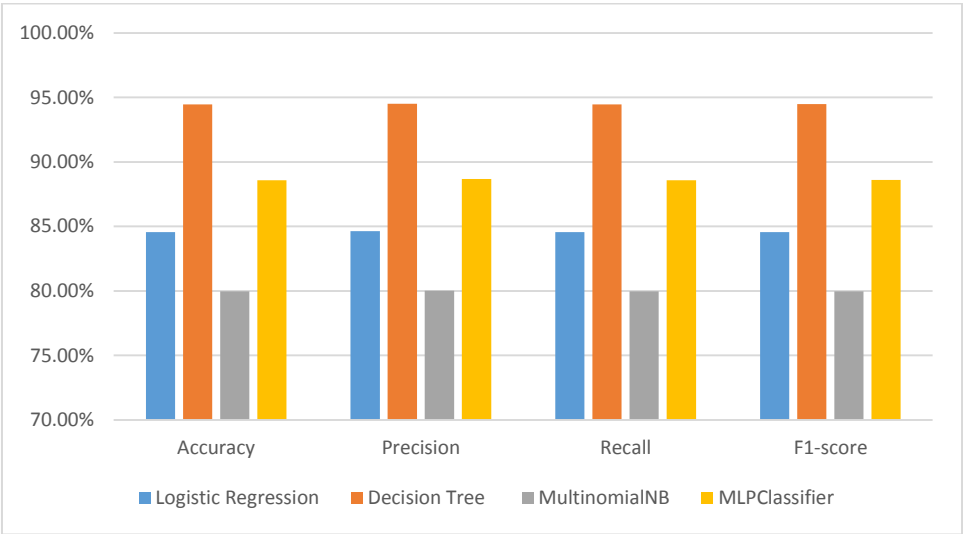


Fig. 55. Results of TF-IDF and 1-gram models with 2-grams

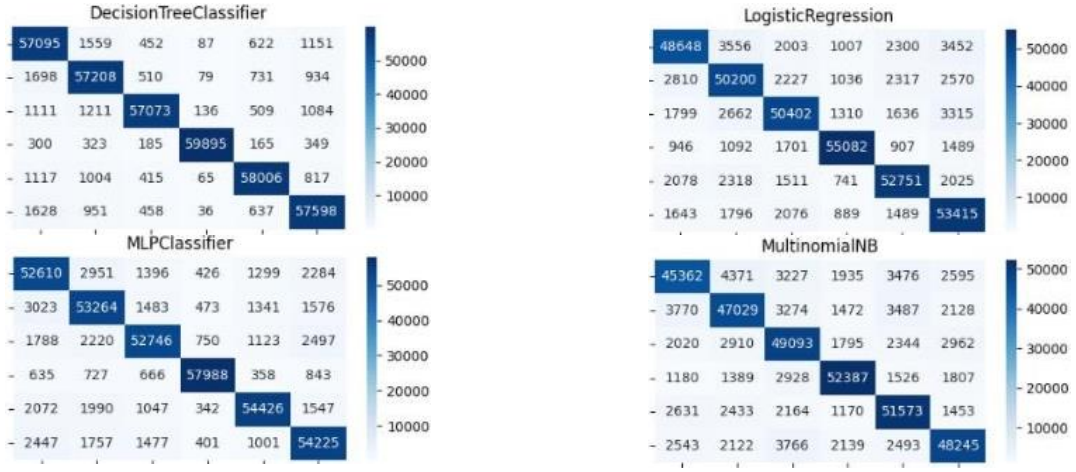


Fig. 56. TF-IDF and 1-gram model error matrix with 2-gram models

An analysis of the effectiveness of two deep learning models, namely based on the GRU and LSTM architectures, is carried out. The architecture of the developed models:

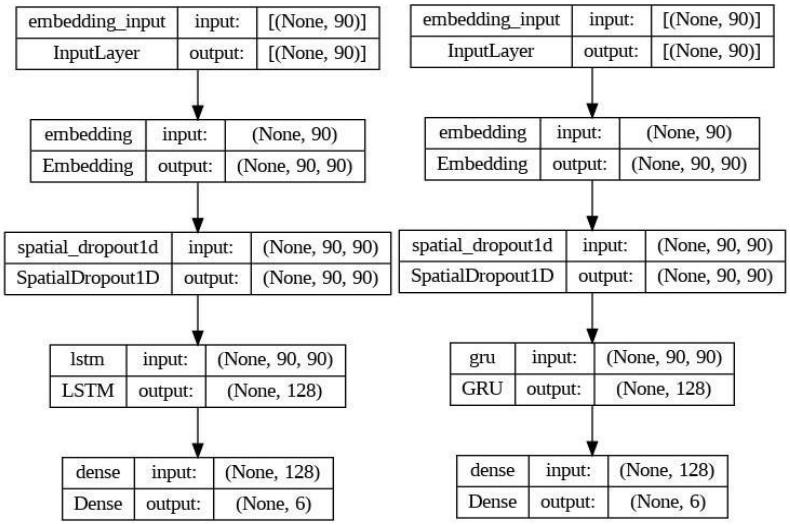


Fig. 57. LSTM and GRU Deep Model Architecture

```
Epoch 1/10
3347/3347 [==============================] - 1014s 303ms/step - loss: 0.6286 - accuracy: 0.7707 - val_loss: 0.4265 - val_accuracy: 0.8446
Epoch 2/10
3347/3347 [==============================] - 962s 288ms/step - loss: 0.3779 - accuracy: 0.8598 - val_loss: 0.3433 - val_accuracy: 0.8723
Epoch 3/10
3347/3347 [==============================] - 952s 284ms/step - loss: 0.3147 - accuracy: 0.8812 - val_loss: 0.2956 - val_accuracy: 0.8887
Epoch 4/10
3347/3347 [==============================] - 962s 287ms/step - loss: 0.2714 - accuracy: 0.8969 - val_loss: 0.2574 - val_accuracy: 0.9028
Epoch 5/10
3347/3347 [==============================] - 958s 286ms/step - loss: 0.2380 - accuracy: 0.9092 - val_loss: 0.2274 - val_accuracy: 0.9142
Epoch 6/10
3347/3347 [==============================] - 944s 282ms/step - loss: 0.2122 - accuracy: 0.9188 - val_loss: 0.2026 - val_accuracy: 0.9232
Epoch 7/10
3347/3347 [==============================] - 939s 280ms/step - loss: 0.1918 - accuracy: 0.9264 - val_loss: 0.1835 - val_accuracy: 0.9309
Epoch 8/10
3347/3347 [==============================] - 934s 279ms/step - loss: 0.1755 - accuracy: 0.9323 - val_loss: 0.1681 - val_accuracy: 0.9367
Epoch 9/10
3347/3347 [==============================] - 934s 279ms/step - loss: 0.1625 - accuracy: 0.9369 - val_loss: 0.1546 - val_accuracy: 0.9416
Epoch 10/10
3347/3347 [==============================] - 940s 281ms/step - loss: 0.1521 - accuracy: 0.9408 - val_loss: 0.1430 - val_accuracy: 0.9459
```

Fig. 58. LSTM Learning Process

The models are implemented using the Keras library. They have a similar architecture. Each of them consists of a model layer and helper layers. Auxiliary layers:

- The embedding layer is used to convert the input text into a numerical sequence using vector embedding. In this case, each embedding has a dimension of 90 words.
- The spatialDropout1D layer randomly turns off the original functions of neurons. It helps to avoid overtraining the model.

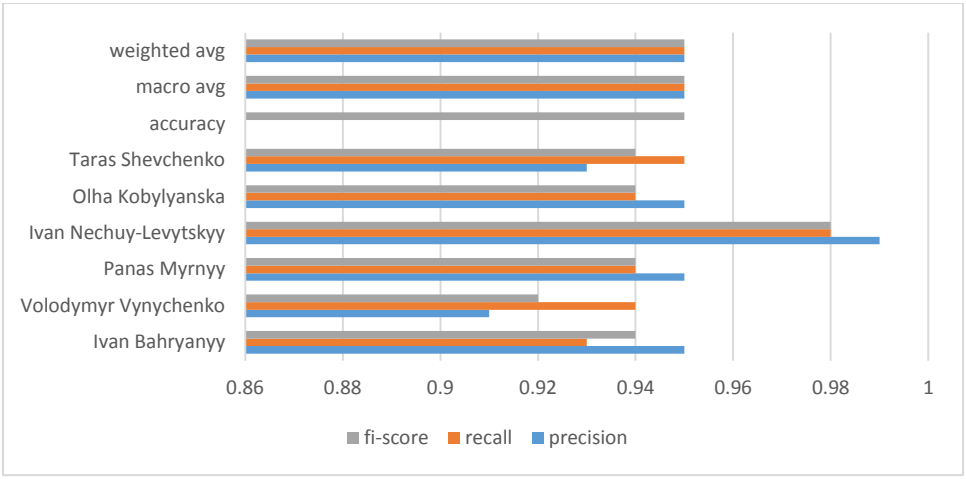
- Dense uses the activation function to solve the classification problem, which, in this case, is divided into six classes.

Model layers are

- The LSTM layer uses LSTM neurons for analysis.
- The GRU layer uses GRU neurons for analysis.

```
Epoch 1/10
3347/3347 [==============================] - 882s 261ms/step - loss: 0.6177 - accuracy: 0.7738 - val_loss: 0.4087 - val_accuracy: 0.8507
Epoch 2/10
3347/3347 [==============================] - 812s 242ms/step - loss: 0.3638 - accuracy: 0.8659 - val_loss: 0.3310 - val_accuracy: 0.8772
Epoch 3/10
3347/3347 [==============================] - 810s 242ms/step - loss: 0.3051 - accuracy: 0.8858 - val_loss: 0.2832 - val_accuracy: 0.8948
Epoch 4/10
3347/3347 [==============================] - 807s 241ms/step - loss: 0.2623 - accuracy: 0.9014 - val_loss: 0.2458 - val_accuracy: 0.9083
Epoch 5/10
3347/3347 [==============================] - 816s 244ms/step - loss: 0.2303 - accuracy: 0.9129 - val_loss: 0.2174 - val_accuracy: 0.9180
Epoch 6/10
3347/3347 [==============================] - 814s 243ms/step - loss: 0.2065 - accuracy: 0.9211 - val_loss: 0.1957 - val_accuracy: 0.9268
Epoch 7/10
3347/3347 [==============================] - 806s 241ms/step - loss: 0.1874 - accuracy: 0.9283 - val_loss: 0.1762 - val_accuracy: 0.9333
Epoch 8/10
3347/3347 [==============================] - 802s 240ms/step - loss: 0.1722 - accuracy: 0.9336 - val_loss: 0.1634 - val_accuracy: 0.9382
Epoch 9/10
3347/3347 [==============================] - 799s 239ms/step - loss: 0.1615 - accuracy: 0.9376 - val_loss: 0.1511 - val_accuracy: 0.9430
Epoch 10/10
3347/3347 [==============================] - 794s 237ms/step - loss: 0.1522 - accuracy: 0.9410 - val_loss: 0.1426 - val_accuracy: 0.9463
```

Fig. 59. GRU Learning Process



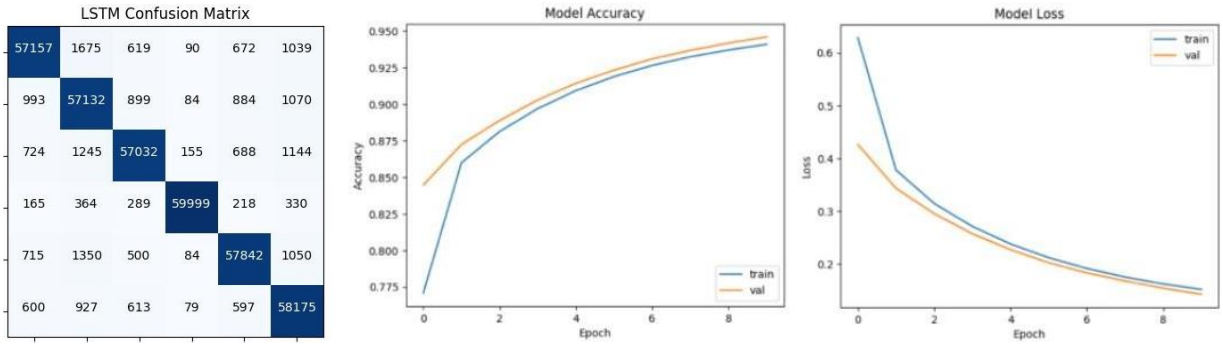


Fig. 60. LSTM Train results

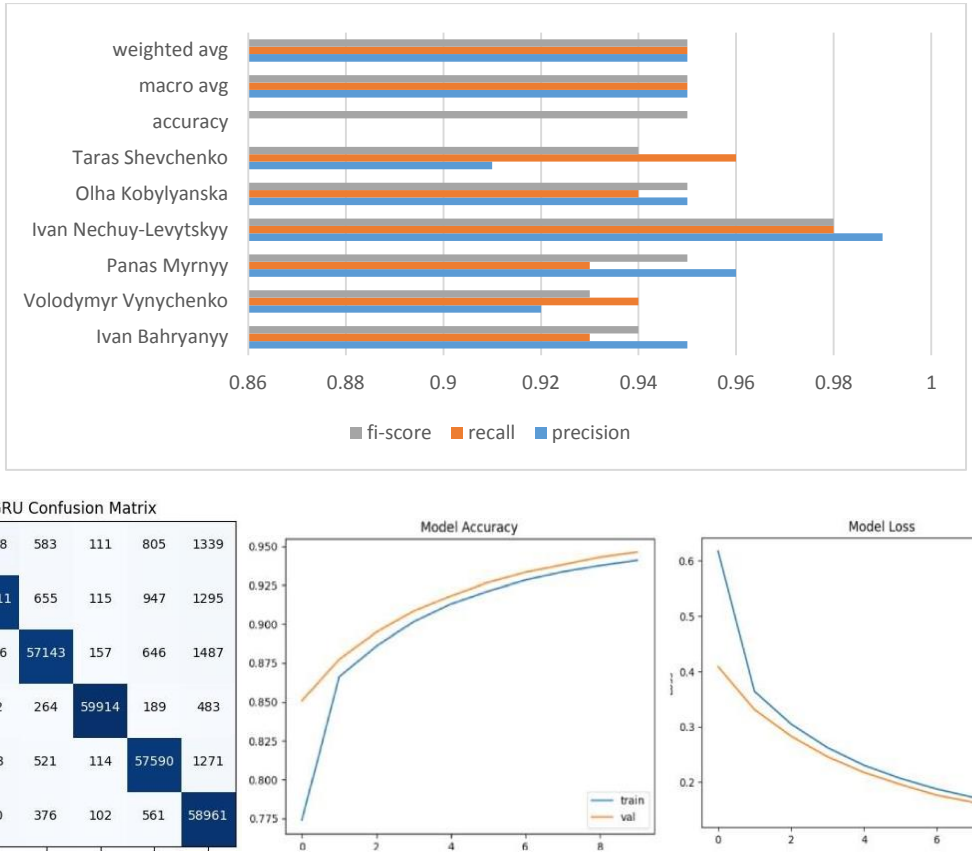


Fig. 61. GRU Train results

As a result of the computational experiment, four popular classical learning models for multi-class classification were tested, and work was carried out with deep learning models by creating and testing two models of the LSTM and GRU types. All studies were conducted using different types of data embedding. In order to find the best, during the study, the methods Word Bag and TF-IDF were used with two types of n-grams: 1-grams and a combination of 1-gram and 2-gram.

Fig. 62. General List of Texts for Analysis and Mining Results - Authorship Determination by Alternative Machine Learning Models

Also, two sets of data were used to train ML models that are able to detect the authorship of texts by writing style: "Netflix Reviews" and "Spotify Reviews".

- Netflix Reviews [DAILY UPDATED]. URL: https://www.kaggle.com/datasets/ashishkumarak/netflix-reviews-playstore-dailyupdated/data
- Spotify Reviews. URL: https://www.kaggle.com/datasets/ashishkumarak/spotify-reviews-playstore-dailyupdate

The "Netflix Reviews" dataset contains information about Netflix user reviews on the Google Play Store. In addition to reviews, it also includes information regarding the rating and date of viewing, likes for each review, and user ID. The dataset contains a semi-structured format for each user's reviews and ratings, and a total of 111,836 reviews are compiled.

The "Spotify Reviews" dataset contains user reviews of the Spotify Music app on the Google Play Store. It also includes information about ratings and dates of views. The file serves as a real-time record of user reviews and ratings and provides additional information about the relevance of reviews and when they were published, as well as the ID of the user who left the review. The number of reviews in this set is 84,165.

The two described data sets underwent software processing and were grouped by users. Only those entries with more than 50 author's comments were included in the working dataset. Thus, the data of 8 authors were obtained, which were divided into training (50 samples each) and testing (remaining samples). Accordingly, the distribution of texts by the received authors who satisfied the quantitative condition is shown in Fig. 63a.

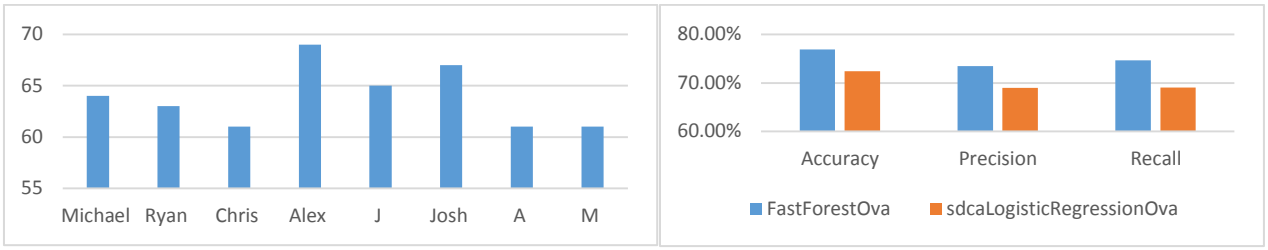


Fig. 63. Author's texts distribution in a working dataset and performance evaluation macro metrics importance for alternative machine learning models

To study the effectiveness of the method of intelligent authorship of texts by writing style, the created and tested software will be used. Two alternative machine learning models, FastForestOva and sdcaLogisticRegressionOva, will be used, and a comparison will be made between them. A pre-marked dataset will be used to compare the author's texts, on which the main metrics, accuracy, and completeness, will be calculated.

During the experiment, data from 8 authors were involved, whose texts were pre-marked but did not participate in training and testing. Based on the test data on the accuracy macro metrics, a value of 76.86 % for FastForestOva and 72.38 % for sdcaLogisticRegressionOva was obtained. The data of the rest of the metrics are shown in Fig. 63b.

Since the FastForestOva machine learning model performed better on all three metrics, it was also separately investigated by metrics for each of the eight authors. The study data for the FastForestOva machine-learning model is shown in Fig. 64.

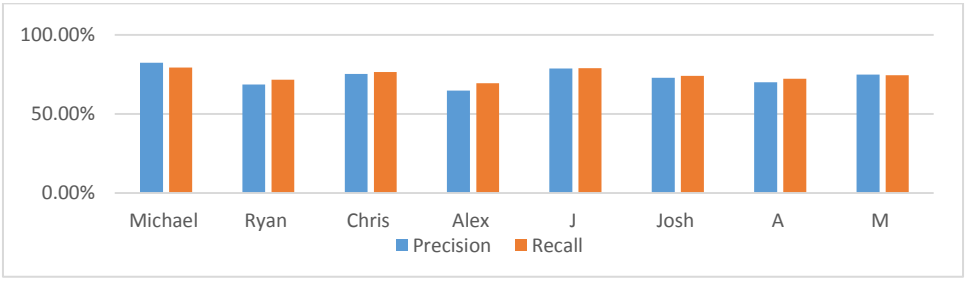


Fig. 64. Significance of FastForestOva Micro-Performance Evaluation Metrics for Each Author

The author with the ID Michael has the highest accuracy scores (82.46%) and completeness (79.30%), indicating that the FastForestOva model recognizes this author's texts best. The high values of both metrics suggest that the model makes almost no errors in predicting Michael's texts. An author with an Alex ID has the lowest accuracy (64.85%) and completeness (69.49%) scores, indicating that FastForestOva's machine learning model has the most significant difficulty identifying this author's texts. It shows a substantial number of false positive and false pessimistic predictions for Alex's texts. Authors with Ryan and A IDs also have lower-than-average accuracy and completeness scores. It suggests that the model has some difficulty correctly identifying the texts of these authors, but not as much as with Alex. Authors with identifiers J, Chris, Josh, and M have near-average accuracy and completeness scores. It suggests that the machine learning model has moderate accuracy in recognizing the texts of these authors, making a certain number of errors. Taking into account the subject area and the fact that the texts are pretty short, the results obtained can be assessed as suitable. It was possible to achieve an accuracy of 76.86% for the experiment with eight authors.

To improve the results obtained, it is necessary to increase the amount of data for training, and you can also try other neural network architectures, such as BERT. It is also planned to expand the number of classes in further research.

Metrics were used to assess the accuracy of the constructed classifier for arbitrary English-language texts to determine the authorship of documents, namely, reliability, accuracy, and completeness of the model. The model was

evaluated for different values of the proportion of keywords from the total feature space used for classification. Fig. 65 shows a comparison of model quality criteria for a different number of coefficients.

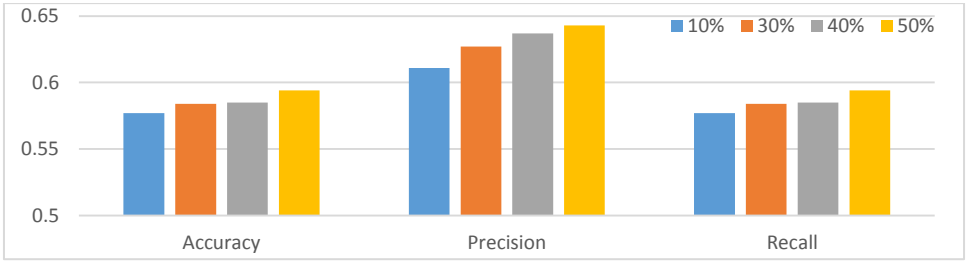


Fig. 65. Comparative Analysis of the Accuracy of the Classification Model for Different Quantiles of the Number of Terms in the Feature Dictionary

The overall score was calculated as a weighted average according to the number of valid cases for each class. The results shown in the table confirm the assumption that the accuracy of the model increases with the number of features. However, the increase in accuracy is so negligible compared to the amount of resources required that it does not seem appropriate. Fig. 66 shows a comparison of accuracy indicators for different classification model settings to determine the authorship of English-language documents.

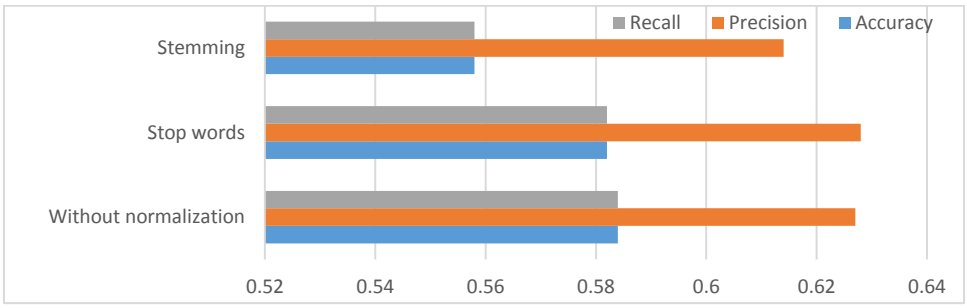


Fig. 66. Comparison of model accuracy measures for different model settings for the 30% quantile of traits used for classification

Contrary to the generally accepted belief that additional normalization of the dataset leads to improved results in this case, the extraction of noise words did not actually affect the accuracy, and after stemming, the accuracy even deteriorated somewhat. It once again proves the fact that the problem of identifying the author by a text document is a highly complex scientific task that requires further research.

10. Discussion

The paper analyses studies devoted to the identification of the authorship of text content based on natural language processing (NLP), artificial intelligence (AI) and machine learning (ML). The results of research demonstrating the effectiveness of modern methods in this area are presented. It is shown that, despite the achievements, issues related to the processing of short texts, cross-genre attribution and multilingualism remain unresolved. The reason for this may be objective difficulties associated with the limited number of stylistic features in short texts, the variability of style between genres, and the complexity of modelling multilingual data. An option to overcome the relevant difficulties can be the use of hybrid models that combine different approaches and take into account the contextual features of the text. It is the approach used in the works [61-69], but further research is needed to improve it. All this gives grounds to assert that it is advisable to conduct a study on the development of adaptive models for identifying authorship in conditions of limited data and high variability of texts. An analysis of modern research shows that, despite significant progress, the identification of the authorship of texts faces a number of challenges.

The models were tested on an English-language dataset with texts from 10 popular personalities on Twitter, as well as scientific texts from arXiv. Main metrics (Table 7-8): Accuracy (classification accuracy), Precision (class prediction accuracy), Recall (class prediction completeness), and F1-score (harmonic average of Precision and Recall). For English, the accuracy of the models was higher due to the more significant amount of available text data. For Ukrainian, a 5-10% drop in accuracy was observed due to the smaller number of text corpora for training.

Table 7. Results of scientific and technical texts analysis

Model	Accuracy	Precision	Recall	F1-score
Random Forest	**88%**	**87%**	**87%**	**87%**
Support Vector Classifier (SVC)	77%	75%	74%	74%
Logistic Regression	73%	72%	72%	72%
Naive Bayes	70%	69%	68%	68%

Table 8. Fiction analysis results

Model	Accuracy	Precision	Recall	F1-score
Random Forest	85%	84%	84%	84%
Support Vector Classifier (SVC)	73%	72%	71%	71%
Logistic Regression	70%	69%	68%	68%
Naive Bayes	65%	63%	62%	62%

Random Forest showed the best results in all tests, especially on scientific and technical texts. **SVC** also performed well but had worse accuracy compared to Random Forest. **The naïve Bayes** showed the worst results due to simplistic assumptions about the independence of features. **Logistic regression** worked well but was inferior to more complex models. Using **deep learning (e.g., BERT)** can improve outcomes but requires significantly more resources.

Further research aimed at developing adaptive and sustainable models is necessary to overcome existing limitations and expand the possibilities of applying these technologies. To overcome these difficulties, it is proposed:

- **Hybrid models,** as a combination of different approaches, such as stylometry and deep learning, can improve the accuracy of authorship identification.
- **Adaptive algorithms**. The development of models that can adapt to different genres and languages is a promising direction.
- **Expanding datasets.** The creation of large and varied corpora of texts will contribute to the training of more stable models.

The research fails to discuss potential limitations or challenges in greater detail, such as data availability and the inherent complexity of scientific writing styles. It is the next goal of further our research. As part of this study, we faced the problem of collecting Ukrainian-language data and their accessibility. Datasets were generated manually from various sources. In particular, some of the scientific and technical publications were provided to us by the editorial board of the Bulletin of Lviv Polytechnic University of the "Information Systems and Networks" series for the period 2001–2021. For English-speaking publications, free online resources from the Internet were used, particularly for the collection of English-language scientific and technical papers/articles/thesis, such as Google Scholar (only your publications, to which there is free access). Additional research is needed in the direction of the complexity of writing texts in a scientific and technical style to improve the accuracy of authorship identification using a scientific writing style.

11. Conclusions

In this work, a Python program was developed to identify the author's style in scientific and technical publications. Using machine learning capabilities, the program effectively distinguishes stylistic patterns unique to individual authors. The complete program code is available in the repository [20]. The process included data collection, pre-processing, feature extraction, model training and evaluation of the results. Using ML algorithms, the program has demonstrated the ability to analyse and classify text data with a significant degree of accuracy. The program's performance was evaluated through thorough testing and validation, which showed promising results regarding standard metrics used to assess ML models: precision, accuracy, and recall. The use of developed tools and well-known libraries, such as scikit-learn, NLTK, and NumPy, contributed to the reliability and efficiency of the program. In addition, experiments with the adjustment of the hyperparameters of the models made it possible to identify the most effective combinations of model parameters for determining the author's style in scientific and technical text. Among the ML models for Ukrainian text tested, the Random Forest classifier performed best on all metrics, making it the best choice for this task. These findings suggest that further experiments with the Random Forest model may yield even better performance for future work.

The document presents the results of testing various machine-learning models for identifying text authorship. The leading indicators for Random Forest Accuracy are 88%, Recall - 87%, Precision - 87%, F1 Score -87%. The Accuracy value for SVC is much lower and equals 77%. For Naive Bayes, Accuracy was 70% and F1 Score was 68%. Accordingly, for Logistic Regression, Precision was 73%, and F1 Score was 72%. The best results were demonstrated by the Random Forest model, which shows its effectiveness in identifying stylistic features of different authors in scientific and technical texts.

Datasets and input data used for the developed authorship identification system based on writing style:

1. The corpus of Ukrainian authors contains the works of 6 Ukrainian authors. The training was carried out using classical learning and deep learning models (LSTM, GRU). The best accuracy among classical models is 95.80% (Decision Tree, Word Bag + 1 gram). Deep models (LSTM, GRU) achieved an accuracy of 94%. Then, the works of 20 Ukrainian authors were analyzed. Four hundred works were used for training. The use of syntactic text characteristics gave the highest accuracy, and lexical ones gave the lowest.

2. The English-language datasets are "Netflix Reviews" and "Spotify Reviews" (Kaggle). They were used to analyze the author's style based on stylistic characteristics. Two methods (FastForestOva and sdcaLogisticRegressionOva) showed an accuracy of 76.86% and 72.38%, respectively.
3. More than 300 excerpts of texts of one-person (≥100 authors) scientific and technical publications of the Bulletin of Lviv Polytechnic University of the "Information Systems and Networks" series for the period 2001–2021 were carried out.

Comparison of analysis results by text areas:

1. Fiction (Ukrainian) contains high style variability, which complicates identification. The highest accuracy was achieved when using syntactic characteristics.
2. Scientific and technical texts (English) are more structured, which facilitates analysis. The best results were demonstrated by Random Forest (88%).

Comparison of analysis results by language

1. English has more available datasets and high accuracy of Random Forest in scientific and technical texts.
2. Ukrainian has fewer resources, which complicates model training. For Ukrainian, a 5-10% drop in accuracy was observed due to the smaller number of text corpora for training. The GRU and LSTM models demonstrated an accuracy of 94% in the corpus of Ukrainian authors.

This work emphasises the potential of ML in the field of authorship determination and text analysis. The developed program serves as a practical tool for determining author's styles and lays the foundation for future research and improvements. Potential improvements could include expanding the dataset, adding more complex linguistic features, and exploring deep learning approaches for more accurate analysis. In conclusion, successfully implementing a Python program for identifying the author's style in scientific and technical texts emphasises the feasibility and value of machine learning methods for determining the author's style in scientific and technical publications. This project demonstrates the intersection of computational linguistics and artificial intelligence, demonstrating their application in academic and research contexts. The highest accuracy was shown by deep learning models (GRU, LSTM) for Ukrainian (94%) and Random Forest for English (88%). Scientific and technical texts are easier for authors to identify than fiction. The limited availability of Ukrainian language corpora remains a challenge for improving the models. English-language datasets with reviews (Netflix, Spotify) yielded an accuracy of 76-78%. To increase the accuracy of authorship identification, it is necessary to expand the data corpora, improve the pre-processing algorithms, and use more powerful neural network architectures. The integration of NLP, AI, and ML into authorship identification has led to a variety of applications:

- **Forensic linguistics -** authorship analysis helps in criminal investigations by attributing anonymous texts to suspects.
- **Plagiarism Detection –** Advanced Models Detect Hidden Forms of Plagiarism by Analysing Stylistic Similarities Between Texts.
- **Cybersecurity –** identifying the authorship of malicious communications, such as phishing emails, increases security.

Such intelligent systems for identifying authorship can be used in many areas (Table 9). All of these applications demonstrate that authorship identification systems have a wide range of uses, ranging from science and journalism to forensics and anti-fraud.

Table 9. The main directions of application of intelligent systems for authorship identification

N	Direction	Features of functioning	Application
1	Detection of plagiarism in scientific papers and academic texts	The system analyzes writing style, sentence structure, and vocabulary to determine if the text matches the author's style.	Universities and publishers can use such systems to check academic articles, dissertations, and student papers for compliance with the author's style or possible borrowing from other sources.
2	Attribution of authorship in historical documents	The system analyzes handwritten or printed texts, comparing them with the style of famous authors.	Researchers can use these systems to determine the authorship of anonymous works, medieval manuscripts, or forged documents.
3	Investigation of journalistic and journalistic materials	If an article is published anonymously or under a pseudonym, the system can identify a potential author based on their previous work.	Identifying authors of fake news or exposés, verifying fake reports.
4	Forensic Science and Crime Investigation	Analysis of letters, messages or notes left by criminals, comparison with existing texts from suspects	It is used by police and intelligence agencies to determine whether the same person wrote different documents or messages.

5	Fake Document Analysis and Cyber Fraud	The system checks texts in emails, reports, or contracts for compliance with the style of the real author.	Banks and law firms can use such systems to detect fake contracts or fraudulent financial statements.
6	Identification of authorship in literature	Algorithms analyze writing style, sentence length, expressions used, and grammar.	Identifying the actual author of an anonymous or controversial work, such as in the case of works by William Shakespeare or Miguel de Cervantes.
7	Copyright protection in the field of digital content	The system analyzes the texts of blogs, articles, social networks or video scripts for correspondence with well-known authors.	Platforms for writers, journalists, or bloggers can check whether their texts have been used without permission.

Table 10. Main Directions of Application of Intelligent Systems for Authorship Identification in Modern Education and Computer Science

N	Direction	Features of functioning	Application
1	Automatic grading of written work and feedback	Algorithms analyze text structure, speech complexity, and relevance to the topic to automatically evaluate works.	- Tools that help teachers quickly analyze large volumes of student texts. - Systems that highlight atypical style changes that may indicate the use of artificial intelligence (e.g. ChatGPT) or someone else's authorship.
2	Detection of academic plagiarism and fraud	The system analyzes the student's writing style and compares it with previous work or publicly available sources.	- Automated systems for checking homework, coursework and diploma projects for plagiarism (Turnitin, Grammarly, Unicheck). - Detection of commissioned work when a student submits a text written by another person.
3	Code Authorship Attribution	The code writing style, frequency of use of certain constructs, variable identifiers, etc., are analyzed.	- Detecting plagiarism in programming (for example, when students copy code from each other or online resources). - Cybersecurity assistance to identify authors of malicious code or attacks (Malware Attribution). - Programming style analysis for recommendations for code improvement.
4	Recognizing the use of AI-generated texts and code	The system analyzes typical writing patterns of text or code to detect whether they were created by AI (e.g. ChatGPT or Copilot).	- Teachers can determine whether a student completed the assignment independently or used a text/code generator. - Automatic checks of homework for uniqueness and originality.
5	Adaptive learning and personalized approach	The system analyzes the student's writing and programming style and offers personalized recommendations.	- Online education platforms can tailor materials to the student's style and level of knowledge. - Teachers can receive information about which aspects of writing or programming each student needs to improve.
6	Automatic creation of study materials and tests	The system analyzes the teacher's teaching materials and generates additional tasks or questions in the appropriate style.	- Generating tests based on textbooks and lectures. - Automatic creation of explanations for code examples.
7	Identification of authorship in scientific articles and peer review	The system analyzes the style of scientific articles and compares them with other publications by the author.	- Journals can verify that the author of an article is indeed its creator. - Automatically identify reviewers whose style and expertise best match the article.

Authorship identification systems can significantly improve education and computer science by automating assessments, helping to combat academic fraud, and improving personalized learning. It makes the learning process more honest, efficient, and transparent.

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 33/0012 from 3/03/2025 (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] Stamatatos E. A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology. - 2009. -60(3). - pp. 538-556.
[2] Koppel M., Schler J. Authorship verification as a one-class classification problem. Proceedings of the twenty-first international conference on Machine learning. - Banff, Alberta, Canada, 2004.
[3] Daelemans W. Explanation in Computational Stylometry. Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing. - 2013. -.2. - pp. 451-462.
[4] Manning C.D., Raghavan P., Schu ̈tze H. Introduction to Information Retrieval. Cambridge University Press, 2008.
[5] Jurafsky D., Martin J. H. Speech and Language Processing. Pearson Education, 2014. - 1024 p.
[6] Requests in Python. URL: https://docs.python-requests.org/en/v2.9.1/.
[7] Arxiv API. URL: https://info.arxiv.org/help/api/user-manual.html.
[8] PyMuPDF 1.24.4. URL: https://pymupdf.readthedocs.io/en/latest/the-basics.html.

[9] nltk in Python. URL: https://www.nltk.org/api/nltk.html.
[10] Sammut C. (ed). Encyclopedia of Machine Learning. Springer, Boston, MA, 2011. - 1031 c.
[11] sklearn. URL: https://scikit-learn.org/stable/user_guide.html.
[12] Bishop C. M. Pattern Recognition and Machine Learning. Springer, 2006. - 738 c.
[13] Cortes C., Vapnik V. Support-vector networks. Machine Learning. - 1995. - Вип.20. - С. 273-297.
[14] Murphy K. P. Machine Learning: A Probabilistic Perspective. - MIT Press, 2012. - 1104 c.
[15] A comparison of event models for Naive Bayes text classification.: AAAI Technical Report WS-98-05 URL: https://cdn.aaai.org/Workshops/1998/WS-98-05/WS98-05-007.pdf.
[16] Hosmer D. W., Lemeshow S., Sturdivant R. X. Applied Logistic Regression. John Wiley & Sons, 2013. - 528 c.
[17] Breiman L. Random forests. Machine Learning, 2001. Вип.45. - С. 5-32.
[18] Liaw A., Wiener M. Classification and regression by RandomForest. Forest. - 2002. - Вип.23. - pp. 18-22.
[19] Powers D. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies. - 2011. -.2. - С. 37-63.
[20] student-applications/2023-2024/PMP-42/bachelor-thesis/ Melenevych Roksoliana. URL: https://github.com/kpm-lnu/student-applications/tree/master/2023-2024/PMP-42/bachelorthesis/Melenevych%20Roksoliana..
[21] R. Romanchuk et al., Intellectual Analysis System Project for Ukrainian-language Artistic Works to Determine the Text Authorship Attribution Probability, in: Proceedings of the 18th International Conference on Computer Sciences and Information Technologies, Lviv, 19-21 October 2023.
[22] S.N. Buk, Fundamentals of statistical linguistics. The Publishing Center of LNU was named after I. Franko, 2008.
[23] M. Bublyk et al., The Decision Tree Usage for the Results Analysis of the Psychophysiological Testing, CEUR workshop proceedings, Vol-2753, 2020, pp. 458-472.
[24] M. Kestemont, et al. "Overview of the cross-domain authorship attribution task at {PAN}," Working Notes of CLEF Conference and Labs of the Evaluation Forum, Switzerland, September 9-12, 2019.
[25] Homer's question. URL: http://febweb.ru/feb/litenc/encyclop/le2/le2-5991.htm
[26] N. Darchuk, K. Sergii, V. Sorokin, Logico-Linguistic Model of Ukrainian Text, CEUR Workshop Proceedings, Vol-3396, 2023, 20-31.
[27] S. Buk, V. Zhukovska, O. Mosiiuk, Multiparametric profiling of a linguistic construction: linguoquantitative and machine-learning aspects, CEUR Workshop Proceedings 3722 (2024) 236-250.
[28] S. Kubinska, et al. "Ukrainian Language Chatbot for Sentiment Analysis and User Interests Recognition based on Data Mining," CEUR Workshop Proceedings 3171 (2022): 315-327.
[29] Who wrote "Silent Don"? (The problem of the authorship of "Silent Don") / Hyetso H., Gustavsson S., Beckman B., Gil S. – M.: Book, 1989.
[30] N. Darchuk, and V. Sorokin, "Parameterisation of the Ukrainian Text Corpus Based on Parsing Results," CEUR Workshop Proceedings 3171 (2022): 256-265.
[31] I. Khomytska et al., "Machine Learning and Classical Methods Combined for Text Differentiation," CEUR Workshop Proceedings 3171 (2022): 1107-1116.
[32] I. Khomytska et al., "The statistical parameters of Ivan Franko's authorial style determined by the chi-square test," International Conference on Computer Sciences and Information Technologies (CSIT), 2022, pp.73-76.
[33] I. Khomytska, et al., "Automated Identification of Authorial Styles," CEUR Workshop Proceedings 3396 (2023): 323-333.
[34] Z. Kunch, O. Lytvyn, and I. Mentynska, "Modern Ukrainian Electronic Dictionaries: the Problem of Implementing Spelling Changes," CEUR Workshop Proceedings 3396 (2023): 32-47.
[35] V. Lytvyn et al. "Development of the quantitative method for automated text content authorship attribution based on the statistical analysis of N-grams distribution," Eastern-European Journal of Enterprise Technologies 6(2-102) (2019): 28-51. DOI: 10.15587/1729-4061.2019.186834.
[36] V. Motyka, et al., "Lexical Diversity Parameters Analysis for Author's Styles in Scientific and Technical Publications," CEUR Workshop Proceedings 3403 (2023): 595-617. URL: https://ceur-ws.org/Vol-3403/paper45.pdf.
[37] I. Khomytska et al. "The Multifactor Method Applied for Authorship Attribution on the Phonological Level," CEUR workshop proceedings 2604 (2020): 189-198.
[38] I. Khomytska et al., "Approach for Minimization Of Phoneme Groups In Authorship Attribution. International Journal of Computing," 19(1), (2020): 55-62.
[39] Z. Haladzhun, et al. "Anti-vaccinationists&Anti-vax": Linguistic Means of Actualizing Assessment in the Headlines and Leads of Ukrainian Text Media, CEUR Workshop Proceedings 3396 (2023): 118-129.
[40] K. Datsyshyn, et al. "Neologisms with the Prefix Anti- in the Ukrainian Online Media in the Covid-19 Pandemic Period," CEUR Workshop Proceedings 3171 (2022): 192-211.
[41] Serkov A. I. "Sorbonneists" and "archivists", or Once again about the authorship of "Romana with cocaine", New Literary Review, 1997, vol. 24, pp. 260–266.
[42] Ancient Ukrainian lexicography. URL: http://litopys.org.ua/ukrmova/um38.htm
[43] V. Shyrokov et al. Terminology Dictionary Digitalisation, CEUR Workshop Proceedings, Vol-3171, 2022, pp. 3-15
[44] M. Vakulenko, V. Slyusar, Automatic smart subword segmentation for the reverse Ukrainian physical dictionary task, CEUR Workshop Proceedings 3723 (2024) 59-73.
[45] A. Dmytriv, et al. "Comparative Analysis of Using Different Parts of Speech in the Ukrainian Texts Based on Stylistic Approach," CEUR Workshop Proceedings 3171 (2022): 546-560.
[46] Text analyser: recognition of authorship. URL: http://habrahabr.ru/post/114187/
[47] N. Borysova, et al. "Gender Classification of Surnames: Ukrainian aspect," CEUR Workshop Proceedings 3171 (2022): 354-364.
[48] V. Starko, and A. Rysin, "VESUM: A Large Morphological Dictionary of Ukrainian As a Dynamic Tool," CEUR Workshop Proceedings 3171 (2022): 61-70.
[49] Y. Hlavcheva et al., "Language-independent features for authorship attribution on Ukrainian texts," CEUR Workshop Proceedings 2833 (2021). URL: https://ceur-ws.org/Vol-2833/Paper_13.pdf.

[50] V. Shynkarenko, et al. "Natural Language Texts Authorship Establishing Based on the Sentences Structure," Idea 18 (2022): 19.
[51] F.A. Acheampong, W. Chen, and H. Nunoo‐Mensah, "Text‐based emotion detection: Advances, challenges, and opportunities," Engineering Reports 2.7 (2020): e12189.
[52] M. Shvedova, N. Prydvorova, and I. Skibina, "Normalisation of Early Modern Ukrainian in GRAC: the Case of Lesia Ukrainka's Works," CEUR Workshop Proceedings 3171 (2022): 71-80.
[53] V. Shynkarenko, and I. Demidovich, "Natural Language Texts Authorship Establishing Based on the Sentences Structure," CEUR Workshop Proceedings 3171 (2022): 328-337
[54] V. Shynkarenko, and I. Demidovich, "Authorship Determination of Natural Language Texts by Several Classes of Indicators with Customizable Weights," CEUR Workshop Proceedings 2870 (2021): 832-844.
[55] H. Antonyuk, L. Chernysh, Diachronical Analysis of Lexicographic Sources of Military Terminology, CEUR Workshop Proceedings, Vol-3171, 2022, pp. 664-676
[56] V. Vysotska, Linguistic intellectual analysis methods for Ukrainian textual content processing, CEUR Workshop Proceedings 3722 (2024) 490-552.
[57] O. Tverdokhlib, et al., Information technology for identifying hate speech in online communication based on machine learning, Lecture Notes on Data Engineering and Communications Technologies 195, 2024, pp. 339–369.
[58] P. Zdebskyi et al., Framework for Improving the Effectiveness of Discussions at English-Language Articles Analysis, in: Proceedings of the IEEE 5th International Conference on Advanced Information and Communication Technologies (AICT), 2023, pp. 127-132.
[59] N. Khairova, et al., Preface: computational linguistics workshop, CEUR Workshop Proceedings, Vol. 3722, https://ceur-ws.org/Vol-3722/
[60] O. Mediakov et al., Information technology for generating lyrics for song extensions based on transformers, International Journal of Modern Education and Computer Science 16(1), 2024, pp. 23–36.
[61] Yülüce, İbrahim, and Ferişţah Dalkılıç. "Author identification with machine learning algorithms." International Journal of Multidisciplinary Studies and Innovative Technologies 6.1 (2022): 45-50.
[62] Misini, Arta, Arbana Kadriu, and Ercan Canhasi. "Authorship classification techniques: Bridging textual domains and languages." International Journal on Information Technologies and Security 16.1 (2024): 27-38.
[63] Hassan, Sayar Ul, Jameel Ahamed, and Khaleel Ahmad. "Analytics of machine learning-based algorithms for text classification." Sustainable Operations and Computers 3 (2022): 238-248.
[64] Alsanoosy, Tawfeeq, Bodor Shalbi, and Ayman Noor. "Authorship Attribution for English Short Texts." Engineering, Technology & Applied Science Research 14.5 (2024): 16419-16426.
[65] Gupta, Sumit, Swarupa Das, and Jyotish Ranjan Mallik. "Machine learning-based authorship attribution using token n-grams and other time tested features." International Journal of Hybrid Intelligent Systems 18.1-2 (2022): 37-51.
[66] Berriche, Lamia, and Souad Larabi-Marie-Sainte. "Unveiling ChatGPT text using writing style." Heliyon 10.12 (2024).
[67] Misini, Arta, Arbana Kadriu, and Ercan Canhasi. "Authorship classification techniques: Bridging textual domains and languages." International Journal on Information Technologies and Security 16.1 (2024): 27-38.
[68] Huang, Baixiang, Canyu Chen, and Kai Shu. "Can large language models identify authorship?." arXiv preprint arXiv:2403.08213 (2024).
[69] Zamir, Muhammad Tayyab, et al. "Stylometry analysis of multi-authored documents for authorship and author style change detection." arXiv preprint arXiv:2401.06752 (2024)

Authors' Profiles

Zhengbing Hu, Professor, 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. Primary research interests: Computer Science and Technology Applications, Artificial Intelligence, Network Security, Communications, Data Processing, Cloud Computing, Education Technology.

Victoria Vysotska is a Professor at the Information Systems and Networks Department of Lviv Polytechnic National University. She defended her Doctoral degree in Technical Science, speciality in «structural, applied and mathematical linguistics», in 2023 on the topic "Analysis and synthesis of computational linguistic systems for processing Ukrainian textual content" Also, she received a PhD degree in Information Technologies from Lviv Polytechnic National University in 2014. She has currently published more than 50 publications. Her main research interests are focused on the identification of disinformation/fakes/propaganda, detection of sources of disinformation and inauthentic behaviour (bots) of coordinated groups based on artificial intelligence, NLP, computer linguistics, data science, system analysis, information technologies, machine learning, cyber security, modern education and distance learning system development.

Lyubomyr Chyrun is an Associate Professor at the Ivan Franko National University of Lviv. Since 2001, he worked at the Lviv Polytechnic University at the Institute of Computer Sciences and Information Technologies. as an associate professor of the Department of Information Systems and Networks. In 2007, he defended his PhD thesis on May 1, 2002 - "Mathematical modelling and computational methods". He co-authors the monograph "Continuous Fractions and Complex Numbers". He is the author of more than 120 scientific publications. His areas of scientific interest are pattern recognition, application of numerical methods in information technologies, object-oriented programming, web technologies, fake identification, natural language processing, computer linguistics, data science, system analysis, information technologies, and machine learning.

Roman Romanchuk is a postgraduate student at the Department of Information Systems and Networks at Lviv Polytechnic National University and a project manager at TietoEvry LLC, Lviv, Ukraine. My research interests are natural language processing, fake detection, computer vision, and telecommunication.

Yuriy Ushenko is a professor at the Computer Science Department of Chernivtsi National University, Chernivtsi, Ukraine. Research Interests are Data Mining and Analysis, Computer Vision and Pattern Recognition, Optics & Photonics, Biophysics. His research interests include information systems design, data mining, pattern recognition and digital image processing, artificial neural networks, laser polarimetry and interferometry.

Dmytro Uhryn graduated from Yuriy Fedkovych Chernivtsi National University, Chernivtsi. He is a Doctor of Technical Sciences and associate professor at Yuriy Fedkovych Chernivtsi National University. He has currently published more than 170 publications. His research interests include data mining, information technology for decision support, swarm intelligence systems, and industry-specific geographic information systems.

Cennuo Hu was born on September 16, 2005. Now she is a student in the Department of Computer Science of the College of Science at Purdue University, West Lafayette, IN 47907, USA. Her main research interests are software engineering and computer security. She is an IEEE student member and has three invention patents.

How to cite this paper: Zhengbing Hu, Victoria Vysotska, Lyubomyr Chyrun, Roman Romanchuk, Yuriy Ushenko, Dmytro Uhryn, Cennuo Hu, "Agile Intelligent Software Solution for Textual Content Authorship Identification Based on NLP, Artificial Intelligence and Machine Learning", International Journal of Modern Education and Computer Science(IJMECS), Vol.17, No.2, pp. 15-66, 2025. DOI:10.5815/ijmecs.2025.02.02