Multilingual analysis for Instagram comments sentiment using transformer models and automatic parsing

Victoria Vysotska[1,†], Andrian Hyriak[1,†], Lyubomyr Chyrun[1,2,†], Rostyslav Fedchuk[1,*,†], Oleksandr Lavrut[3,†], Dmytro Uhryn[4,†], Liubov Kolyasa[1,†], Saule Smailova[5,†] and Mariia Brygadyr[6,†]

[1] *Lviv Polytechnic National University, S. Bandera 12, 79013 Lviv, Ukraine*
[2] *Ivan Franko National University, Universytetska Street 1, Lviv, 79000 Lviv, Ukraine*
[3] *Hetman Petro Sahaidachnyi National Army Academy, Heroes of Maidan 32, 79026 Lviv, Ukraine*
[4] *Yuriy Fedlovyvh Chernivtsi National University, Kotsiubynskoho Street 2, 58012 Chernivtsi, Ukraine*
[5] *D. Serikbayev East Kazakhstan University, D. Serikbayev STR., 19, 070004 Ust-Kamenogorsk, The Republic of Kazakhstan*
[6] *West Ukrainian National University, Lvivska Street 11, 46004 Ternopil, Ukraine*

Abstract

The article presents a comprehensive system for the automated analysis of comments on Instagram, focusing on multilingual content and the unique characteristics of social networks. The system includes a module for automatic parsing of dynamic content, an algorithm for determining the language of comments, and mood analysis modules built on the basis of modern transformer models, in particular XLM-RoBERTa. Particular attention is paid to supporting Ukrainian, Russian, and English, as well as processing texts with informal elements, including slang, abbreviations, emojis, and symbols. An approach to analysing mood dynamics over time by combining models of time series, moving averages, and clustering is proposed. The system is complemented by interactive visualisation of results, which enables researchers and businesses to gain in-depth insights from large amounts of data. The analysis of existing solutions demonstrates the advantages of the proposed approach, particularly its high accuracy for local languages and its adaptation to social media content. The developed tool is crucial for monitoring public sentiment, gathering business intelligence, and enhancing information security, particularly in the Ukrainian context.

Keywords

Sentiment analysis, social media, Instagram, multilingualism, transformer models, XLM-RoBERTa, automatic parsing, language detection, time series, natural language processing (NLP), data visualisation, emotional analysis, comment trends.

1. Introduction

In today's world, social media has become a powerful tool for communication, marketing, opinion analysis, and research into consumer sentiment [1]. Instagram, as one of the most popular platforms, generates millions of comments daily that contain valuable information for businesses, academics, NGOs, and governments. However, it is not possible to process such a volume of data manually, and existing automated solutions have significant limitations, especially in the context of multilingualism, mood specificity, and local contexts [2]. Most modern solutions for analysing comments on social networks focus on English-language content, overlooking multilingualism and

AIT&AIS'2025: International Scientific Workshop on Applied Information Technologies and Artificial Intelligence Systems, December 18–19 2025, Chernivtsi, Ukraine
[1,*] Corresponding author.
[†] These authors contributed equally.
victoria.a.vysotska@lpnu.ua (V. Vysotska); andrian.hyriak.sa.2022@lpnu.ua (A. Hyriak); lyubomyr.v.chyrun@lpnu.ua (L. Chyrun); rostyslav.b.fedchuk@lpnu.ua (R. Fedchuk); alexandrlavrut@gmail.com (O. Lavrut); d.ugryn@chnu.edu.ua (D. Uhryn); liubov.i.koliasa@lpnu.ua (L. Kolyasa); ssmailova@edu.ektu.kz (S. Smailova); m.bryhadyr@wunu.edu.ua (M. Brygadyr)
0000-0001-6417-3689 (V. Vysotska); 0009-0007-4948-4586 (A. Hyriak); 0000-0002-9448-1751 (L. Chyrun); 0009-0002-6669-0369 (R. Fedchuk); 0000-0002-4909-6723 (O. Lavrut); 0000-0003-4858-4511 (D. Uhryn); 0000-0002-9690-8042 (L. Kolyasa); 0000-0002-8411-3584 (S. Smailova); 0000-0002-1101-7479 (M. Brygadyr)

the distinct characteristics of language groups, such as Ukrainian and Russian [3]. Problems with existing solutions [4–6]:

1. Insufficient support for the Ukrainian language.
2. The difficulty of working with multilingual content.
3. Lack of in-depth trend analysis.
4. Limited in specialised analysis.

In the context of Ukraine, there is a growing need to analyse local content, including Ukrainian and Russian. Existing solutions lack high accuracy due to the absence of specialised models for these languages [2–4]. In real-world conditions, comments often contain text in multiple languages, symbols, emojis, and abbreviations, making them challenging to analyse [7]. Existing systems do not account for the dynamics of mood changes over time, linguistic features, or the solidarity of comments with the initial post [8]. Most solutions do not utilise modern advancements in the field of transformers, particularly specialised models that can provide accurate sentiment analysis for various languages [9]. The proposed solution, which includes automatic parsing, multilingual sentiment analysis using specialised transformer models, and the construction of a comprehensive trend analysis, is highly relevant to Ukraine for several reasons [2–4]:

1. Development of the Ukrainian IT sector.
2. Support of public opinion.
3. A tool for business.
4. Promotion of scientific research.

The creation of specialised models for the Ukrainian language will contribute to the development of local technologies and increase the competitiveness of Ukrainian companies in the global market. In times of war and post-conflict reconstruction, sentiment analysis on social media can be helpful in monitoring public opinion, detecting disinformation, and evaluating the effectiveness of information campaigns. Businesses will be able to receive more accurate analytics on reactions to their products or services, taking into account local language features and nuances. The proposed solution opens up new opportunities for research in the field of natural language processing, mood analysis and sociology. Thus, the proposed project is not only relevant but also critically important for the development of text analysis technologies in social networks within a multilingual environment, particularly in the context of Ukraine. It allows you to address existing limitations by providing accurate, contextual and multilingual analysis of comments, which is essential for both business and society as a whole.

The purpose of the study is to develop a comprehensive system of automated analysis of comments on Instagram, which provides multilingual text processing, sentiment identification, trend building and in-depth analysis of the interaction of comments with posts, taking into account the specifics of language groups and the context of social networks, to increase the efficiency of decision-making by businesses, researchers and organisations.

To achieve this goal, it is necessary to solve the following tasks:

1. Development of a mechanism for automatic parsing of comments from Instagram: ensure efficient retrieval of data from posts, including comments, emojis, and symbols; consider the technical limitations of the Instagram API and optimise for high scraping performance.
2. Definition of comment language: implement an algorithm to automatically detect the language of comments, taking into account multilingualism, abbreviations, slang and symbols, and add labelling of comments (e.g. "ua", "ru", "en", "symbol_only") for further analysis.
3. Development of a mechanism for analysing the sentiments of comments: integrate specialised pre-trained transformer models from the Hugging Face platform to analyse the

sentiments of comments in different languages, and to ensure high accuracy of analysis, adapted to the specifics of texts from social networks.
4. Building a comprehensive analysis of the results: develop algorithms to assess the solidarity of comments with posts, determine the general mood, and build trends for maintaining posts over time, and perform an analysis of the sentiment of comments based on the language group.
5. Visualisation of data and analysis results: develop tools for visualising results in the form of graphs, charts, and interactive reports, and to provide the possibility of segmented analysis (by language, time, mood, etc.).
6. Testing and optimisation of the system: test the system on real data from Instagram to check its performance and accuracy, and optimise the algorithm to ensure efficient processing of large datasets.

Thus, the implementation of these tasks will create an innovative solution for the multilingual analysis of comments in social networks, which will be useful for businesses, researchers, and organisations, particularly in Ukraine, and will contribute to the development of natural language processing technologies. The object of this study is the automated processing and analysis of text data generated by users in social networks, specifically Instagram comments that contain multilingual content, symbols, emojis, and other text features. The subject of the study is methods and algorithms for automatic parsing, multilingual text processing, sentiment determination, and analysis of comment trends in social networks, particularly specialised transformer models for analysing texts in different languages, as well as approaches to visualising results to provide a deep understanding of user interaction with content. Within the framework of the study, the following new scientific provisions and solutions were obtained, which differ from the previously known ones and have the subsequent degree of novelty:

1. For the first time, an approach to analysing multilingual comments on Instagram has been developed, taking into account the specifics of the Ukrainian, Russian, and English languages, as well as content consisting only of characters (symbol_only). An algorithm for automatically determining the language of comments is proposed, taking into account the features of texts in social networks, such as slang, abbreviations, emojis and symbols.
2. For the first time, specialised pre-trained transformer models have been integrated to analyse the sentiments of comments in various languages. Adaptation of models from the Hugging Face platform for text analysis in the multilingual environment of Instagram, in particular for the Ukrainian language, which was previously underrepresented in existing solutions.
3. The process of analysing the solidarity of comments with posts has been improved. A new approach has been developed to assess the level of support or criticism of comments regarding the content of the initial post, enabling a more accurate evaluation of user interaction with the content.
4. The method of complex analysis of moods in the dynamics of time was further developed. An algorithm for identifying trends in comment sentiments based on the time of post publication is proposed, enabling the detection of changes in audience reactions to content over time.
5. Improved the approach to analysing comment sentiment based on language group. For the first time, a comparative analysis of the moods of comments in different languages (Ukrainian, Russian, English, and symbol_only) was conducted, taking into account linguistic and cultural features.
6. A new approach to visualising the results of analysing multilingual comments has been developed. Interactive data presentation tools have been developed that enable businesses, researchers, and organisations to quickly gain insights from large datasets, taking into account both linguistic and emotional characteristics.

Thus, the results of the study have a high degree of novelty, as the proposed solutions provide a more accurate, contextual and multilingual analysis of comments on social networks, which has not been implemented in this form before, especially for the Ukrainian context.

The developed project has significant practical value, since its implementation opens up new opportunities for data analysis in social networks, in particular on Instagram, and can be applied in various areas [10–15]:

1. Business Analytics and Marketing – monitoring consumer sentiment. Businesses have the opportunity to analyse audience reactions to their products, services, or advertising campaigns, taking into account the sentiments of commentators in various languages; Evaluating the effectiveness of content. The system enables you to determine the level of solidarity between comments and posts, which helps assess how the content resonates with the audience; Identifying trends. Sentiment analysis, in the context of time dynamics, helps businesses predict consumer behaviour and adapt their strategies.
2. Public opinion and sociological research – monitoring of public sentiment. The project enables you to analyse user reactions to socially significant topics, events, or political decisions, taking into account the multilingual nature of comments; Identification of language features. Analysis of moods across different language groups helps to better understand the cultural and regional characteristics of information perception.
3. Information Security and Countering Disinformation – detection of harmful content. The system can be used to automatically detect toxic or destructive comments, which is essential for content moderation; Monitoring of information campaigns. The tool enables you to evaluate the effectiveness of information campaigns targeting the combat of disinformation, particularly in the context of Ukraine.
4. Development of local technologies: Support for the Ukrainian language – the creation of specialised models for analysing texts in Ukrainian contributes to the development of local natural language processing technologies; Integration into the Ukrainian IT sector. Ukrainian IT companies can utilise the project to develop new products and services centred on social media analysis.
5. Academic Studies: Expansion of research in the field of natural language processing – the integration of pre-trained transformer models for multilingual text analysis opens up new opportunities for scientific research; Study of social interactions. The project can be used to analyse interactions between users and content in social networks, which is relevant to sociology, psychology, and media studies.
6. Tool for state bodies and public organisations – assessment of public support; government agencies and civil society organisations can utilise the system to gauge the public's sentiment on key social or political issues; monitoring social trends. The tool enables you to track changes in societal moods, which is crucial for informed strategic decision-making.
7. Data visualisation for decision-making – interactive reports. The results of the analysis are presented in a visual form (graphs, diagrams, trends), which simplifies the decision-making process for businesses, organisations and researchers; Segmented analysis. The ability to analyse data by language, time, mood, and other factors provides flexibility in working with large amounts of data.

The developed project is of particular importance to Ukraine, as it contributes to the development of local IT solutions, supports the Ukrainian language in the technological environment, helps analyse public sentiment in challenging socio-political conditions, and provides tools to combat disinformation. Thus, the practical value of the project lies in its versatility, adaptability to a multilingual environment, ability to generate deep insights from social media data, and support for the development of technologies focused on local needs.

2. Comparison of the product being developed with analogues, advantages/disadvantages determining and problem formulation

In today's world, social media, particularly Instagram, has become a powerful source of data for analysing user sentiment, public opinion, and behaviour [1–4]. However, processing such data is a challenging task due to its multilingual nature, unstructured format, use of symbols, emojis, and specific slang [5–9]. To solve these challenges, it is necessary to create specialised systems that combine automatic parsing, multilingual text analysis, sentiment determination, and trend building over time. There are numerous solutions on the market today that focus on text analysis. Still, most of them focus on specific aspects of the problem, such as data scraping, language definition, or sentiment analysis [10–12]. Complex systems that integrate all these functions are mostly internal solutions of large companies and are not freely available. It makes it difficult to directly compare the product under development with analogues, since there are no open systems on the market that fully comply with the proposed concept. For objective analysis, this section will compare individual components of the system under development with existing analogues, such as tools for data parsing (Selenium, Instaloader, Instagram API), methods for determining the language of texts (LangDetect, FastText), models for sentiment analysis (BERT, RoBERTa, VADER), and approaches to data visualisation. It will enable you to assess the advantages and disadvantages of each component, justify the choice of methods and technologies for implementing the system, and identify any remaining unresolved problems. Thus, the analytical review will aim to compare the components of the project being developed with existing solutions, critically analysing their effectiveness and adapting them to the specific tasks facing the project. It will enable you to identify key tasks and create a comprehensive system that provides accurate and multilingual analysis of texts from social networks.

2.1. Overview of existing systems for automating text analysis in social networks

Social media is a primary source of textual data for businesses, researchers, and organisations seeking to understand user sentiments, trends, and behaviours [12–15]. To automate the analysis of texts on social networks, several platforms and tools offer various functionalities.

Let's review the most well-known systems, such as Google Cloud Natural Language, Microsoft Azure Text Analytics, Brandwatch, and Sprout Social, with an emphasis on their functionality, multilingual capabilities, sentiment analysis accuracy, integration, and visualisation features.

1. Google Cloud Natural Language API is a service from Google that allows you to analyse text data using artificial intelligence. It supports features such as language detection, sentiment analysis, entity recognition, and parsing. Determines the polarity (positive, negative, neutral) and mood intensity of the text. Automatically identifies the language of the text. Highlights key objects, such as names, places, and organisations. It supports more than 20 languages, including English and Russian, but does not provide full support for the Ukrainian language. The accuracy of sentiment analysis is high for English-language content; however, it decreases for other languages, particularly for texts from social networks that contain slang, emojis, and abbreviations. The API integrates seamlessly into applications thanks to REST APIs and SDKs for different programming languages. It does not provide built-in visualisation tools, but the results can be integrated with other graphing tools. Limited support for the Ukrainian language. High cost for large amounts of data. Google Cloud Natural Language is a powerful tool for text analysis; however, its limitations in local languages and the specific characteristics of texts from social networks reduce its effectiveness for multilingual analysis.
2. Microsoft Azure Text Analytics is a part of Azure Cognitive Services that provides an API for text analysis. It supports sentiment detection, entity recognition, text classification, and language detection. Determines the emotional tone of the text. Highlights entities and

categorises them. Automatically detects the language of the text. Supports more than 120 languages, including Ukrainian, Russian, and English. The accuracy of sentiment analysis is high for major languages; however, it decreases for texts from social networks, particularly those containing symbols and emojis. Easy integration via REST API and SDK for different programming languages. It does not have built-in visualisation tools, but the results can be used in other tools for visual representation. High cost for large amounts of data. Limited adaptation to the specifics of social media texts. Microsoft Azure Text Analytics has broader language support than Google Cloud Natural Language, but its accuracy for texts from social networks remains limited.

3. Brandwatch is a social media monitoring platform that allows you to collect data, analyse sentiment, identify trends, and generate reports. Collects data from various platforms, including Instagram, Twitter, and Facebook. Determining the mood of texts in social networks. Automatic identification of key trends in texts. Supports text analysis in multiple languages, with varying accuracy levels by language. The accuracy of sentiment analysis is high for English-language content, but may be reduced for less popular languages such as Ukrainian. The platform offers APIs for integration with other systems. It has built-in tools for creating graphs, charts, and reports. High cost. Closed access to sentiment analysis algorithms. Brandwatch is an effective tool for monitoring social networks, but its limitations in supporting local languages and high cost make it less accessible for multilingual analysis.
4. Sprout Social is a social media management platform that also includes text analysis features. Data collection from different platforms. Determination of the moods of texts. Automatically generate reports on user interactions. Supports text analysis in multiple languages, but is limited. The accuracy of sentiment analysis is high for English-language content, but limited for other languages. Integration with other platforms via API. It has built-in visualisation tools for generating reports. High cost. Limited support for local languages. Sprout Social is useful for social media management, but its text analysis functionality is basic.

Existing social media text analysis systems offer a wide range of features; however, they have limitations, including support for local languages (such as Ukrainian), the accuracy of text analysis from social networks, and adaptation to the specifics of multilingual content. The system under development aims to solve these problems by integrating specialised models for sentiment analysis, multilingual analysis and adaptation to texts from social networks.

2.2. Comparison of data collection methods

Collecting data from social networks, particularly Instagram, is a crucial step in further analysing comments, sentiments, and trends [15–18]. However, the choice of scraping tool depends on several factors, including the data structure, platform limitations, the tool's performance, and compliance with privacy policies. Let's compare the most common tools for scraping data from Instagram: Selenium, Scrapy, BeautifulSoup, Instaloader, and Instagram Graph API. Particular attention is paid to the advantages of Selenium for dynamic comment parsing.

1. Selenium is a browser automation tool that allows you to interact with web pages the way a real user does. It is ideal for dynamic parsing because it can interact with page elements generated by JavaScript. Selenium allows you to load comments that appear after clicking the "Load more comments" button. It is possible to automate any actions on the page, including scrolling, clicking buttons, and entering data. Using a real-time browser reduces the risk of detecting automated actions – the ability to work with Chrome, Firefox, Edge and other browsers. Selenium consumes a lot of RAM and computing resources due to page rendering in the browser. Less productive compared to tools that work without a browser. Instagram may detect automated activities, so additional measures, such as using proxies,

are required. Selenium is a powerful tool for dynamic scraping, particularly for handling complex pages with dynamically loaded elements, such as those found on Instagram comments.

2. Scrapy is a Python web scraping framework that allows you to quickly scrape data from static web pages. Scrapy is much faster than Selenium because it doesn't use a browser. Suitable for collecting large amounts of data from simple static pages. Easily integrates with other libraries for data processing. Scrapy cannot work with JavaScript-generated elements, such as dynamically loaded comments. To work with Instagram, you need additional tools or workarounds. Scrapy is effective for static pages, but its limitations in working with dynamic content make it less suitable for scraping comments from social media platforms like Instagram.
3. BeautifulSoup is a Python library for analysing HTML and XML documents. It allows you to extract data from web pages using a simple API. Easy to set up for basic parsing. It can be used to work with HTML structures of any complexity. It can be combined with other tools such as Scrapy or Selenium. BeautifulSoup cannot work with dynamically uploaded content. Works slower than Scrapy, especially for large amounts of data. BeautifulSoup is suitable for basic HTML page scraping, but it cannot work with dynamic content, such as Instagram comments.
4. Instaloader is a Python tool for downloading data from Instagram, including posts, profiles, and comments. Easily configurable for basic data collection from public profiles. Supports logging in to the account to access private data. Can extract comments on posts. Using Instaloader may violate Instagram's terms of service, resulting in account suspension or ban. The tool is subject to Instagram's restrictions, which may affect the stability of the work. Cannot interact with elements that JavaScript generates. Instaloader is helpful for basic data collection from Instagram, but its limitations in working with dynamic content make it less effective for scraping comments.
5. Instagram Graph API is an official tool for accessing Instagram data that provides features to collect information about posts, comments, and other metadata. The API complies with Instagram's privacy policy. Allows access to structured data with high accuracy. You can obtain additional information, such as the publication date, the author of the comment, and so on. The API only works with business accounts and creator accounts. There are strict limits on the number of requests, which makes it challenging to work with large amounts of data. The data is only available for public profiles. The Instagram Graph API is a reliable and legal tool, but its limitations make it less flexible for comprehensive comment scraping.

Among all the tools reviewed, Selenium is the best choice for scraping comments from Instagram, as it enables working with dynamically loaded elements, such as the "Load more comments" button. Although Selenium has high resource requirements and is slower, its flexibility and ability to mimic user actions make it ideal for tasks that involve dynamic content. Other tools, such as Scrapy, BeautifulSoup, and Instaloader, are less suitable for this task due to limitations in working with JavaScript-generated elements. At the same time, the Instagram Graph API has strict access restrictions.

Table 1
Comparison of tools

Tool	Dynamic content	Speed	Flexibility	Risk of blocking	Legality
Selenium	Yes	Low	High	Medium	Partially
Scrapy	No	High	Average	Low	Partially
BeautifulSoup	No	Average	Average	Low	Partially

Instaloader	No	Average	Low	High	Partially
Instagram Graph API	No	High	Low	Low	High

2.3. Comparison of methods for determining the language of texts

Determining the language of a text is a key step for multilingual data analysis on social media. In the context of Instagram, comments can be written in different languages, contain mixed language elements, symbols, emojis, and abbreviations, making it difficult to automatically detect the language [18–24]. We will review the most common algorithms and libraries for determining the language of texts, such as LangDetect, FastText, and DeepLang, and their adaptation to the specific needs of texts from social networks. Particular attention is paid to accuracy, performance and the possibilities of integrating these methods into the system being developed.

1. LangDetect is a popular text language detection library based on an algorithm that uses statistical models and Bayesian classification. It supports more than 50 languages. Supports most common languages, including Ukrainian, Russian, and English. Easily integrates into Python projects – high speed of text processing, which allows you to work with large amounts of data. LangDetect has difficulty detecting the language of short texts, such as social media comments. Texts that contain multiple languages or characters may be classified incorrectly. LangDetect is effective for detecting the language of long texts, but its accuracy for short and mixed texts is limited.
2. FastText is a text classification and vectorisation library developed by Facebook AI. It also features a model for detecting the language of the text, which supports over 170 languages. FastText demonstrates high accuracy even for short texts. Supports a greater number of languages than LangDetect. FastText models are designed to be fast, making them well-suited for processing large amounts of data. The model can be additionally trained on specific data from social networks. To use FastText, you must have a sufficient amount of RAM. The integration can be more complicated compared to LangDetect. FastText is one of the best options for determining the language of short texts such as Instagram comments, due to its accuracy and adaptability.
3. DeepLang is a text-language detection library that uses neural networks for classification. It supports more than 100 languages and is one of the most advanced technologies in this field. The use of neural networks ensures high accuracy even for short and mixed texts. Can work with texts that contain symbols, emojis, and abbreviations. You can additionally train the model on specific data. DeepLang requires significant computing resources to operate. Integration into the system requires additional effort due to the complexity of working with neural networks. DeepLang is the best option for accurately determining the language of texts from social networks; however, it requires significant resources to utilise.

Table 2
Comparison of methods

Method	Extensive language support	Accuracy for short texts	Adaptation to mixed texts	Speed	Resources
LangDetect	Yes	Average	Low	High	Low
FastText	Yes	High	Average	High	Average
DeepLang	Yes	High	High	Average	High

For a system that handles comments from Instagram, LangDetect is the best choice. However, FastText offers high accuracy for short texts, broad language support, and the ability to adapt to the specifics of texts from social networks. DeepLang is also a promising option for providing the highest accuracy, but its resource requirements can be a limiting factor. LangDetect can be used for basic tasks; it does not require significant resource expenditures on the part of the system, so it was chosen for the project.

2.4. Comparison of Sentiment Analysis Models

Analysing the sentiments of texts is a crucial task for understanding the emotional reaction of the audience to specific content on social networks [1–24]. This section discusses modern approaches to sentiment analysis, including traditional vocabulary methods and modern transformer-based models. Particular attention is paid to their accuracy for multilingual text analysis, including Ukrainian, Russian and English. The choice of models from the Hugging Face library for implementing the system under development will also be justified.

1. Traditional dictionary approaches to sentiment analysis are based on the use of pre-created dictionaries that contain words with the meanings of their polarity (positive, negative, neutral). The most common tool in this category is VADER (Valence Aware Dictionary and sEntiment Reasoner). Advantages of VADER: ease of use, high accuracy for English-language text, particularly short texts such as tweets or comments and taking into account the intensity of the mood through punctuation, capital letters and emojis. Disadvantages of VADER:

1.1. Limited language support, i.e., VADER is primarily focused on the English language.

1.2. Low accuracy for multilingual analysis or texts from social networks that contain slang, abbreviations, and mixed languages.

1.3. Difficulties with context arise when dictionary methods fail to take into account the context, which can lead to errors in determining mood.

VADER is effective for fundamental sentiment analysis of English-language content; however, its limitations in multilingualism and contextuality render it unsuitable for complex multilingual tasks.

2. Transformers are modern models for natural language processing (NLP) that use a self-attention mechanism to take into account the context of each word in the text. They provide high accuracy for sentiment analysis, even for multilingual content.

2.1. BERT (Bidirectional Encoder Representations from Transformers) is a bidirectional model that takes into account the context of words both to the left and right of the current word. Advantages: high accuracy for sentiment analysis due to contextual consideration and the ability to further train on specific data, which increases accuracy for particular tasks.

2.2. RoBERTa (A Robustly Optimised BERT Pretraining Approach) is an advanced version of BERT that uses better pre-training techniques and larger amounts of data – advantages: higher accuracy compared to BERT, as well as the ability to adapt to specific tasks. The disadvantage is that, like BERT, RoBERTa is based on English-language data in the basic version. RoBERTa is effective for English-language sentiment analysis, but its limitations in multilingualism remain.

2.3. DistilBERT is a simplified version of BERT that provides faster operation and lower resource requirements. Advantages: lower requirements for computing resources and speedier word processing. Disadvantages: reduced accuracy compared to BERT and limited language support. DistilBERT is a compromise between accuracy and speed, but its limitations in multilingualism remain.

2.4. XLM-RoBERTa (Cross-lingual RoBERTa) is a multilingual version of RoBERTa that supports more than 100 languages, including Ukrainian, Russian, and English. Advantages:

High accuracy for multilingual sentiment analysis. Adaptation to texts in different languages, including less common ones. The possibility of additional training on specific data.

The disadvantage is the high requirements for computing resources. XLM-RoBERTa is the best choice for multilingual sentiment analysis because it achieves high accuracy across texts in various languages.

Table 3
Comparison of models

Model	Accuracy	Multilingualism	Resources	Contextuality	Adaptation to social networks
VADER	Average	Low	Low	Low	Low
BERT	High	Low	High	High	Average
RoBERTa	High	Low	High	High	Average
DistilBERT	Average	Low	Average	High	Average
XLM-RoBERTa	High	High	High	High	High

Among the sentiment analysis models considered, XLM-RoBERTa is one of the best choices for multilingual text analysis on social networks due to its accuracy, contextuality, and support for multiple languages. Traditional dictionary approaches, such as VADER, are less effective due to their limited multilingual capabilities and lack of contextual consideration. Transformer-based models, such as BERT, RoBERTa, and DistilBERT, demonstrate high accuracy for English-language content; however, their limitations in multilingualism make them less suitable for the system under development. That is why it was decided to use an individual approach, employing different models for various languages, as well as for symbols and emojis. In the future, with the development of models, there will be a complete transition to specialised models for each language.

2.5. Comparison of approaches to analysing sentiment trends over time

Analysing mood trends over time is a crucial task for understanding the dynamics of changes in users' emotional responses to content on social networks [1–24]. This approach enables you to assess how the perception of posts evolves, identify peaks in activity, and discern long-term trends. This section discusses the primary methods for analysing sentiment trends over time, including time series analysis, moving averages, and sentiment clustering. It compares existing approaches with those proposed in the system under development.

1. Time series is a method of analysing data that changes over time, using forecasting and trend detection models. In the context of sentiment analysis, time series enable you to estimate the change in sentiment of comments on posts during specific time periods, offering advantages such as forecasting, trend detection, and flexibility. Time series models such as ARIMA (AutoRegressive Integrated Moving Average) allow you to predict future mood changes. Will enable you to identify long-term sentiment trends. Suitable for analysing data with different time intervals (hours, days, weeks). Disadvantages: data requirements and complexity of setup. Practical analysis requires large amounts of data that have a regular time structure. Time series models require careful optimisation of their parameters. Time series are a powerful method for analysing sentiment trends over time; however, their effectiveness depends on the quality and regularity of the data.

2. Moving averages are a method that uses averaging sentiment values over a specific time period to smooth out short-term fluctuations and identify general trends. Advantages: simplicity, data smoothing and visualisation. Easy to implement and customise. Allows you to reduce the impact of noise and short-term fluctuations. Moving averages integrate seamlessly with charting tools to build trends – disadvantages include the loss of parts and limited predictive ability. Anti-aliasing can hide significant short-term mood changes. Moving averages do not allow you to predict future mood changes. Moving averages are a simple and effective method for identifying general trends, but their limitations in forecasting make them less suitable for complex tasks.
3. Mood clustering is a method of grouping comments according to similar emotional characteristics, followed by an analysis of their dynamics over time. Advantages: pattern detection, relationship analysis, and flexibility. Allows you to identify comment groups with similar sentiments and their changes over time. You can identify relationships between sentiment and other factors (for example, when posts are posted). Various clustering algorithms, such as K-Means, DBSCAN, or hierarchical clustering, can be used. Disadvantages are complexity and resource requirements. Clustering requires careful adjustment of parameters, such as the number of clusters. Analysing large amounts of data can be a resource-intensive process. Sentiment clustering is an effective method for analysing sentiment dynamics in depth; however, its complexity can be a limiting factor.

Table 4
Comparison of methods

Method	Ease of implementation	Ability to predict	Trend Detection	Noise smoothing	Flexibility
Time series	Average	High	High	Low	High
Moving averages	High	Low	Average	High	Average
Clustering	Low	Low	High	Low	High

In the system being developed, a combined approach will be used to analyse mood trends over time:

1. Time series to predict mood changes based on historical data.
2. Moving averages are used to smooth out short-term fluctuations and identify general trends.
3. Clustering to group comments by similar emotional characteristics and analyse their relationships over time.

This approach allows you to consider the benefits of each method, providing an accurate and in-depth analysis of moods over time.

Analysing mood trends over time is a crucial task for understanding the dynamics of the audience's emotional responses. Among the methods considered, time series provide the best predictive ability, while moving averages allow data to be smoothed to identify general trends, and clustering adds the ability to analyse the relationships between sentiment and other factors in depth. The proposed combined approach in the system under development enables the integration of the advantages of all methods, providing an accurate analysis of mood changes over time.

2.6. Formulation of the problem and justification of the need to develop a system

In today's world, social media, including Instagram, has become a crucial source of data for analysing audience sentiments, trends, and behaviours. User comments contain valuable

information that can be used for business intelligence, sociological research, and public opinion monitoring. However, existing solutions for automating text analysis in social networks have significant drawbacks that limit their effectiveness, especially for multilingual content and local languages such as Ukrainian. This section summarises the main problems that arise during the analysis of texts from social networks, substantiates the need for a new system, and determines how the proposed approach is superior to its analogues. Disadvantages of existing solutions:

1. Low accuracy for local languages. Most modern text analysis systems, such as Google Cloud Natural Language and Microsoft Azure Text Analytics, are focused on English-language content. The Ukrainian language is often underrepresented in educational datasets, resulting in low accuracy in mood analysis and language definition. Many models overlook the slang, abbreviations, and specific grammar of local languages, which compromises the quality of the analysis.
2. Limited support for multilingualism. Existing systems have limited support for multilingual content, especially for texts that contain mixed languages (e.g., Ukrainian and English). Multilingual models, such as mBERT, often exhibit reduced efficiency for less common languages, including English.
3. The complexity of working with the texts of social networks. Texts from social networks have specific features, such as slang, abbreviations, emojis, and symbols, that are not taken into account by many existing models. Dynamic content, such as Instagram comments, uploaded via JavaScript, creates technical challenges for data collection. The lack of adaptation to short texts, which are often found in comments, reduces the accuracy of the analysis.
4. Lack of integration. Most systems do not offer a comprehensive approach that includes parsing, sentiment analysis, language group definition, and trending over time. Existing solutions often require the integration of multiple tools, making them difficult to use.

Based on the analysis of the shortcomings of existing solutions, the following key problems are formulated, which are solved by the developed system:

1. Improving accuracy for local languages. The system under development utilises modern transformer models, such as XLM-RoBERTa, which offer high accuracy for Ukrainian, Russian, and English. Adding models to specific data from social networks enables you to account for the peculiarities of slang, abbreviations, and symbols.
2. Expanding support for multilingualism. The system provides analysis of mixed-language texts, including determining the language group for each comment. The integration of multilingual models enables you to work effectively with texts in multiple languages.
3. Adaptation to texts from social networks. The system takes into account the specific characteristics of texts from social networks, including emojis, symbols, and abbreviated text forms. The use of specialised algorithms to collect data from dynamic content (Selenium) provides access to comments that are loaded via JavaScript.
4. Integrated approach. The system integrates all the key stages of analysis, including parsing comments, determining language, analysing moods, building trends over time, and visualising results. The possibility of interactive data analysis through visualisation (Plotly) is provided.

Justification of the need to develop the system:

1. The uniqueness of the proposed approach. Existing solutions focus on specific aspects of text analysis, such as parsing or sentiment analysis, but do not offer a comprehensive approach. The system under development integrates all stages of analysis, which provides a complete cycle of data processing from social networks.

2. Advantages over analogues: accuracy, flexibility, interactivity and comprehensiveness. The use of XLM-RoBERTa enables you to achieve high accuracy for multilingual content, including Ukrainian. The system is adapted to texts from social networks, taking into account their specifics. Data visualisation through Plotly enables users to interact with the analysis results. The system under development encompasses all stages of data analysis within a single product.
3. Importance for Ukraine. The development of a system that takes into account the specific characteristics of the Ukrainian language contributes to the advancement of local technologies in the field of natural language processing (NLP). The system can be used to monitor public sentiment, assess the effectiveness of information campaigns, and combat disinformation.

Existing social media text analysis solutions have significant drawbacks that limit their effectiveness for multilingual content and local languages. The system under development addresses these problems by integrating modern models for sentiment analysis, multilingual text processing, and adaptation to the specifics of social networks, while providing an integrated approach to data analysis. Its implementation will contribute to improving the accuracy of analysis, expanding support for local languages, and providing interactive data analysis, making it an important tool for businesses, researchers, and organisations, especially in the context of Ukraine.

The analytical review revealed that existing solutions for analysing texts from social networks have significant limitations, making it difficult to utilise them effectively for multilingual content, particularly for local languages such as Ukrainian. The main drawbacks include low accuracy of sentiment analysis for less common languages, limited support for texts from social networks, difficulty working with dynamic content, and a lack of a comprehensive approach to data analysis. The considered tools and methods, such as Google Cloud Natural Language, Microsoft Azure Text Analytics, VADER, XLM-RoBERTa, Selenium, and Plotly, demonstrate strengths in certain aspects of the analysis but do not provide a comprehensive data processing cycle from social networks. They are either focused on English-language content or do not take into account the specifics of texts from social networks, such as slang, emojis, and mixed languages. The developed system has several significant advantages over existing analogues, including multilingualism, adaptation to texts from social networks, comprehensiveness, interactivity, and flexibility. The use of the modern XLM-RoBERTa model yields high accuracy in sentiment analysis for Ukrainian, Russian, and English, taking into account the characteristics of local language groups. The system takes into account the specific characteristics of texts from social networks, such as slang, abbreviations, emojis, and symbols, which enhances the quality of analysis. Integration of all key stages of analysis – parsing comments, determining language, sentiment analysis, building trends over time, and data visualisation – within one product. Using Plotly to visualise results enables interaction with the data, making it easier to analyse and interpret. The system can be further trained on specific data from social networks, which allows it to be adapted to new challenges and tasks.

The developed system addresses the key problems that arise during the analysis of texts from social networks, offering multilingualism, adaptation to local languages, and an integrated approach to data processing. Its implementation will contribute to increasing the efficiency of text analysis, the development of local technologies, and the creation of new opportunities for businesses, researchers, and organisations.

3. System analysis of the product under development

System analysis is a crucial stage in the development of information systems, as it enables a thorough exploration of the subject area, identification of key system requirements, and justification of its architecture and structure. In the context of developing a system for automating the analysis of comments from social networks, system analysis provides an understanding of the complex, multi-level interaction between the system's components, as well as determining the

optimal solutions for addressing the tasks. Social networks, such as Instagram, generate vast amounts of text data that contain valuable insights into user sentiment, trends, and behaviour. However, analysing this data is a challenge due to multilingualism, the use of slang, emojis, symbols, and the dynamic nature of the content. Developing a system that automates the analysis of such data requires a systematic approach that takes into account all aspects, from data collection to processing, analysis, and visualisation. The main goals of the system development include:

1. Automation of data collection from social networks – ensuring effective scraping of comments from dynamic Instagram content.
2. Multilingual analysis of texts – determination of the language group of comments, analysis of moods and taking into account the specifics of local languages (Ukrainian, Russian, English).
3. Building sentiment trends over time – identifying changes in user sentiment depending on time and other factors.
4. Interactive visualisation – creating graphs and reports that allow users to interact with the results of the analysis.

System analysis will not only allow us to determine the main components of the system and their interactions, but also to justify the choice of architecture and implementation methods. It will ensure the creation of an efficient, reliable, and scalable system that addresses the current challenges of analysing texts in social networks.

Social media platforms like Instagram are platforms where users actively interact through text, images, videos, and comments. Instagram allows users to publish posts that other users can comment on and react to with likes and emojis. Comment texts on social networks have several essential characteristics that affect their analysis: content dynamism, multilingualism and a large amount of data. Comments are constantly changing, new ones are added, and old ones can be deleted. Instagram is a global platform, allowing comments to be written in a variety of languages, including English, Ukrainian, Russian, and others. Popular posts can garner thousands of comments, which creates challenges in processing large amounts of information. Multilingual content on Instagram often features comments written in different languages within the same post. It creates challenges for analysis, such as speech recognition (automatic detection of the language of the comment), processing of texts with different grammatical structures (each language has its own rules for constructing sentences that must be taken into account), translation and normalisation (for sentiment analysis, comments can be translated or brought to a single format). The dynamism of comments means that the data is constantly changing. It creates challenges such as the timeline and the relevance of the data. Comments are added in real-time, which is essential for analysing mood changes. Old comments may become irrelevant over time, so the system must take this into account. Social media comment texts often feature several characteristics, including slang and abbreviations, emojis, spelling mistakes, and short phrases. Users frequently use informal words, abbreviations, and acronyms, such as "OMG" and "LOL". Emojis are an integral part of communication on Instagram, as they effectively convey emotions and moods. Comments often contain errors due to the speed of writing or the informality of the platform. The comments are typically brief, making it challenging to comprehensively analyse the context.

The general goal of the system is to develop an automated system for analysing texts in Instagram comments to determine user sentiments, taking into account multilingualism, the dynamism of content, and the specificity of texts. The primary objectives are data collection, text processing, mood analysis, and visualisation of results. Data collection involves parsing comments, processing dynamic data, and utilising web scraping. Text processing encompasses normalisation, pre-processing, and processing of multilingual content. Sentiment analysis involves pinpointing sentiments, analysing emojis, and utilising Transformer models. Visualising the results consists of distributing sentiments by language, displaying mood changes in graphs, and filtering data. The

results of achieving the objectives will define the criteria for the system's functioning, including relevance, accuracy, speed, scalability, and versatility.

To develop a system for analysing texts in Instagram comments, several alternative options are being considered. These options included a selection of models for sentiment analysis, tools for parsing comments, and methods for processing multilingual content. After analysing the advantages and disadvantages of each option, it was decided to use the transformer models for sentiment analysis, since they provide high accuracy and take into account the context of the texts. Alternative options for sentiment analysis:

1. Rule-based models use predefined lexicons (for example, dictionaries of positive and negative words). Advantages: ease of implementation, does not require large amounts of data for training. Disadvantages include low accuracy for texts containing slang, emojis, and abbreviations.
2. Machine learning-based models use algorithms such as Logistic Regression, Random Forest, or SVM. Advantages: high accuracy for texts with well-defined features. Disadvantages: require large amounts of data for training.
3. Deep Learning-based models use neural networks such as LSTM, GRU, or transformers (e.g., BERT, RoBERTa). Advantages: high accuracy for complex texts, takes into account the context. Disadvantages include high computational complexity and the need for significant resources for training.

For sentiment analysis, transformer models (e.g., BERT, RoBERTa, or XLM-R) were chosen, as they provide the highest accuracy for complex texts, take context into account, and support multilingualism. It is vital for analysing texts on Instagram, where comments may contain slang, emojis, and abbreviations. Alternative options for parsing comments:

1. Instagram API – Using the official API to fetch data. Advantages: reliability, access to up-to-date data. Disadvantages include restrictions on the number of requests and the need for authorisation.
2. Web Scraping – using libraries for parsing HTML (for example, BeautifulSoup). Advantages: Ability to bypass API restrictions. Disadvantages: Risk of violating Instagram's terms of service and privacy policy.

Web Scraping was chosen for parsing comments because it provides reliable access to up-to-date data. In the event of API restrictions, alternative approaches can be used as a fallback.

Alternative options for handling multilingual content:

1. Automatic language detection – using libraries such as langdetect or fastText. Advantages: speed and accuracy of language detection. Disadvantages: possible errors for short texts.
2. Multilingual models for sentiment analysis – using transformers such as mBERT or XLM-R. Advantages: Support for multiple languages, taking context into account. Disadvantages: high computational complexity.

Multilingual transformer models (e.g., mBERT or XLM-R) were chosen to handle multilingual content because they support text analysis in multiple languages and take into account context, which is crucial for multilingual content on Instagram.

To select the best option, the analytical hierarchy process (AHP) method was employed. A comparison of alternative options was carried out according to the following criteria:

1. Accuracy (how accurately the model determines the mood of the text).
2. Computational complexity (resources required to process data).
3. Flexibility (the ability to adapt to multilingual content).
4. Data availability (ease of obtaining data for analysis).

Based on the scores obtained using these criteria, it was decided to utilise transformer models for sentiment analysis, Instagram APIs for parsing comments, and multilingual models for processing multilingual content. Thus, the systematic analysis of the object of study and the subject area made it possible to consider all the features of the texts in Instagram comments and select the best options for implementing the system. The system under development is designed to automate the process of analysing comments on social media platforms, particularly on Instagram. Its main functions are:

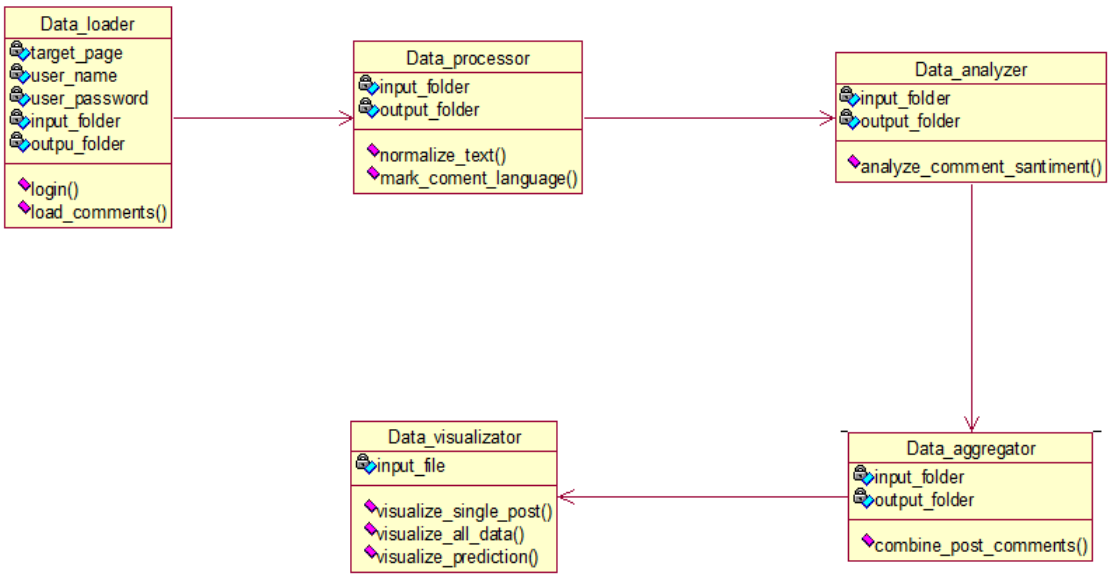


Figure 1: Class diagram.

1. Comment Scraping – Automatically collect comments from social media posts, including dynamic content that is loaded via JavaScript. Parsing is provided using the Selenium tool. Also, collecting metadata such as the time the comment was published, the author, and the related post.
2. Language Definition – Automatically detects the language group of each comment (for example, Ukrainian, Russian, English, or mixed languages). Also, marking comments that consist only of symbols or emojis.
3. Sentiment analysis – the use of modern transformer models, such as XLM-RoBERTa, to determine the polarity of moods (positive, negative, neutral). Additionally, adapting models to the specificities of texts from social networks, including slang, abbreviations, and emojis.
4. Trend building – analysing mood changes over time, creating time series, and identifying long-term trends. Additionally, the construction of graphs that reflect the dynamics of moods over time, language groups, or other factors.
5. Interactive visualisation – using interactive graphs (based on Plotly) to present analysis results that allow users to interact with data, scale graphs, and highlight key segments.

The system is a universal tool for multilingual text analysis, enabling the automation of data collection, processing, and presentation. The system has a wide range of potential users, including:

1. Businesses, in particular, analyse consumer reactions to products, services, or marketing campaigns, as well as identifying trends in customer feedback to make informed decisions.
2. Researchers, such as sociologists, linguists, and analysts, can use the system to study public opinion, audience behaviour, and language characteristics, as well as monitor changes in the population's mood on socially significant topics.
3. Organisations, such as civil society organisations, can utilise the system to monitor public sentiment, evaluate the effectiveness of information campaigns, and detect disinformation.

Government agencies can also use the system to analyse public opinion on political/social initiatives.

The system is helpful for any organisation that works with large amounts of text data on social media platforms and requires automated analysis. Social media text analysis faces several challenges that limit the effectiveness of existing solutions, including low accuracy for local languages, limited support for multilingualism, the complexity of working with social media texts, and a lack of integrated solutions. Most commercial systems, such as Google Cloud Natural Language and Microsoft Azure Text Analytics, are primarily focused on English-language content and exhibit lower accuracy for Ukrainian and Russian. Existing solutions do not cope well with texts that contain mixed languages, slang and symbols. Features of texts from social networks, such as abbreviations, emojis, and dynamic content, are often overlooked. Most systems focus on specific aspects of analysis (for example, only parsing or only sentiment analysis), which makes it challenging to use them for complex analysis. The system under development addresses these problems by integrating modern technologies, including transformers (XLM-RoBERTa), multilingual models for language detection (FastText), and tools for interactive visualisation (Plotly). Expected effects from the implementation of the system: improving the accuracy of sentiment analysis, supporting multilingualism, process automation, interactivity, and the development of local technologies. The use of modern models for multilingual analysis provides high accuracy even for local languages (Ukrainian, Russian). The system supports text analysis in different languages, including mixed languages, making it versatile for working in multilingual environments. Automatic parsing of comments and sentiment analysis significantly reduces labour costs for processing large amounts of data. Interactive graphs allow users to easily analyse the results and gain deeper insights. The system contributes to the development of natural language processing (NLP) technologies for the Ukrainian language, which is essential in the context of Ukraine. Conceptual model of the system:

1. Input: Comments (text data collected from Instagram, including mixed languages, symbols, and emojis) and metadata (time the comment was published, author, related post).
2. Initial data: the results of sentiment analysis (the polarity of each comment as positive, negative, or neutral), trends (graphs of mood changes over time) and reports (interactive reports including sentiment analysis, frequency of language groups, time trends).
3. Functions and structure of the system: parsing module (automatic collection of data from Instagram), language detection module (classification of the language group of each comment), mood analysis module (determination of the emotional polarity of texts) and visualisation module (construction of interactive graphs and reports).
4. System requirements: performance (fast processing of large amounts of data), accuracy (high accuracy of sentiment analysis for multilingual content), relevance (correspondence of data information in real time), and scalability (ability to process data from different sources, not only Instagram).

The system being developed is designed to automate the analysis of comments on social media platforms, particularly Instagram. Its implementation will enhance the accuracy of sentiment analysis, support multilingualism, and interactivity, making it a valuable tool for businesses, researchers, and organisations. The conceptual model of the system defines its key components and requirements, ensuring the effective implementation of tasks.

The system analysis made it possible to thoroughly investigate the problems associated with automating text analysis in social networks and to identify the key aspects of system development. The subject area was studied in detail, and the general goal of the system was formulated. A tree of goals was built, which structures the tasks that must be performed to achieve this goal. The analysis of alternative approaches to building a system made it possible to justify the choice of the optimal architecture, which provides efficiency, flexibility, and scalability. The key aspects of

system development were: justification of the chosen architecture, modelling the system's structure and functions, and determining the system's requirements. The system is based on modern technologies, including transformer models (XLM-RoBERTa) for multilingual sentiment analysis, Selenium for the automatic collection of dynamic content, and seaborn for visualisation. This choice enables you to ensure high accuracy of analysis, adapt to the specifics of texts from social networks, and achieve ease of use. The requirements for performance, accuracy, interactivity and scalability of the system were formulated. It ensures its ability to process large amounts of data, work with multilingual content, and adapt to changes in the subject area. The creation of such a system is a crucial step in addressing the current challenges of analysing texts in social networks. It enables you to automate data collection, processing, and analysis, increase the accuracy of sentiment analysis, maintain multilingual support, and provide interactive visualisation of results. It makes the system useful for businesses, researchers, NGOs, and government agencies that work with large amounts of text data. Thus, the conducted system analysis laid the groundwork for the further implementation of a system that addresses the challenges of analysing texts in social networks and meets the modern requirements for information systems.

4. Selection of methods and means of the product being developed

The choice of methods and means is a key stage in the development of information systems, as the efficiency, accuracy, and productivity of the final product depend on it. For the development of a system to automate the analysis of comments on social networks, particularly Instagram, it is essential to consider the specific characteristics of texts, which are often multilingual, unstructured, and contain slang, emojis, abbreviations, and symbols. It creates additional challenges for data collection, processing, and analysis. The main tasks that the system solves are:

1. Comment parsing – automatic collection of text data from social networks, including dynamic content generated by JavaScript.
2. Language Detection– Automatically detect the language group of texts, including support for mixed languages and texts with symbols.
3. Sentiment analysis – determining the emotional polarity of comments (positive, negative, neutral), taking into account the context.
4. Trend building – analysing mood changes over time and identifying long-term trends.
5. Interactive visualisation – presenting results in the form of graphs and reports that allow users to interact with data.

To address these issues, several key requirements for the methods and technologies employed in the system have been identified: accuracy, performance, multilingualism, adaptability, and interactivity. The methods should provide high accuracy of sentiment analysis, especially for local languages (Ukrainian, Russian) and multilingual content. The system must efficiently process large amounts of data, including thousands of comments from social networks. The methods should support the analysis of texts in various languages, including those that are mixed. The technologies used should take into account the specific characteristics of texts from social networks, such as slang, emojis, and abbreviations. Visualisation tools should provide ease of use and allow for interactive graphing. To achieve the set goals, modern technologies and libraries that meet these requirements were chosen. In particular, Selenium is used for parsing, while transformer models from the Transformers library are utilised for sentiment analysis. The langdetect library is employed for language detection, and the Seaborn and Matplotlib libraries are utilised for data visualisation. Each of these tools has been selected based on its benefits in its respective field, ensuring the system's efficiency and reliability. Thus, the choice of methods and means is a critical stage that determines the success of the system implementation. In the following sections, the choice of each technique and tool will be substantiated in detail, and a comparative analysis with analogous tools will be conducted to confirm their effectiveness.

4.1. Sentiment analysis methods

Sentiment analysis is a key task for the system being developed, as it enables the determination of users' emotional reactions to content on social networks. For this, modern transformer models are used, which provide high accuracy of analysis due to the context. Advantages of transformer models include high accuracy, multilingualism, and adaptability. Transformers consider the context of each word, enabling you to accurately determine the emotional polarity of the text. Models such as XLM-RoBERTa support text analysis in different languages, which is critical for a multilingual social media environment. Models can be further trained on specific data from social networks, which allows slang, abbreviations, and emojis to be taken into account. Comparison with traditional methods:

1. Dictionary Approaches (VADER) is a simple sentiment analysis tool that is based on word polarity dictionaries. Erevaga: Ease of use, high accuracy for English-language content. Disadvantages: low accuracy for multilingual content and a lack of consideration for context.
2. Transformers offer significantly higher accuracy due to their contextual understanding, but require more resources to operate.

Models used:

1. A multilingual version of XLM-RoBERTa that supports more than 100 languages, including Ukrainian, Russian, and English. It is used to analyse the sentiments of comments in Ukrainian. Model is cardiffnlp/twitter-xlm-roberta-base-sentiment.
2. RuBERT is a model designed for analysing texts in Russian, specifically adapted to the language's unique characteristics. It is used to analyse the sentiment of comments in Russian. Model is blanchefort/rubert-base-cased-sentiment.
3. RoBERTa is an English-language model that demonstrates high accuracy for texts from social networks. It is used to analyse the sentiment of comments in the English language. Model is cardiffnlp/twitter-roberta-base-sentiment.
4. RoBERTa for symbols is used to analyse texts consisting only of symbols or emojis. Model is cardiffnlp/twitter-roberta-base-sentiment.

4.2. Methods for determining language

Determining the language of texts is an essential task for multilingual analysis. The system under development uses the langdetect library, which provides automatic detection of the language group of texts. The library uses statistical models to classify texts by language. Advantages: ease of integration into Python projects, high speed and support for more than 50 languages, including Ukrainian, Russian and English. Disadvantages include low accuracy for short texts, which are often found in comments, as well as limited support for mixed languages (for example, when Ukrainian and English are used in the exact text).

Comparison with other methods:

1. FastText is a library that demonstrates high accuracy for short texts and supports more than 170 languages. Advantages: high accuracy, adaptability to texts from social networks. Disadvantages: resource requirements, complexity of integration.
2. Langdetect is a straightforward solution for basic language detection, but it is less effective for complex texts.

4.3. Word processing methods

Pre-processing of texts is a critical step to ensure the accuracy of the analysis. The following approaches are used in the system being developed:

I. Pre-treatment:

1. Normalisation of texts: Remove unnecessary characters, spaces, and punctuation; Convert text to lower case; Using the unicodedata library to remove accents and special characters.
2. Cleaning Texts: Remove URLs, emojis, and other irrelevant items; Using regular expressions (re) to find and remove specific patterns; Tokenisation is the process of splitting text into separate words or tokens for further analysis.

II. Using regular expressions (re):

1. Simplicity – regular expressions allow you to process text data efficiently.
3. Flexibility – the ability to adapt to the specifics of texts from social networks.

4.4. Visualization methods

Data visualisation enables you to present the results of the analysis clearly and understandably. The following libraries are used in the system under development:

1. Seaborn is a tool for creating aesthetic charts. It is used to build bar graphs, heat maps, and line graphs.
2. Matplotlib is a basic library for creating graphs. It is used to adjust the details of graphs (colours, labels, axes).
3. Statsmodels is used to smooth data using the LOWESS (Locally Weighted Scatterplot Smoothing) method. Allows you to identify trends in time series.

Advantages: flexibility (the ability to create both static and interactive graphs), aesthetics (graphs look modern and clear), and interactivity (integration with Plotly to build interactive graphs). The methods for solving the problem were chosen taking into account the specifics of texts from social networks, as well as the requirements for accuracy, productivity, and multilingualism. The use of transformer models for sentiment analysis, language detection libraries such as langdetect, regular expressions for text processing, and modern libraries for visualisation ensures the efficiency and reliability of the system. Each method was chosen based on its advantages in the relevant field, allowing you to solve tasks with high quality.

4.5. Means of solving the problem

Python was chosen as the primary programming language to automate the analysis of comments from social networks. Python is a versatile, user-friendly, and powerful tool for working with text data, analysing large datasets, and developing complex systems. Its popularity in the field of natural language processing (NLP), the presence of numerous libraries, and an active community make Python the optimal choice for this project. The rationale for choosing Python is versatility, simplicity, community, and scalability. Python supports a wide range of libraries for working with texts, data analysis, and visualisation. Easy to digest and allows you to quickly develop complex systems – a large user base and an active community that provides access to documentation and support. Python is well-suited for working with both small datasets and large amounts of information. Overview of the libraries used:

1. Selenium is used to automatically scrape comments from dynamic Instagram content that JavaScript generates. Selenium enables you to interact with web pages in a manner similar to a real user, including clicking buttons, scrolling through pages, and filling out forms. Advantages: the ability to work with dynamic content and flexibility in setting up parsing for specific tasks. Disadvantages: high resource requirements compared to other tools.
2. Pandas is used to process and analyse structured data. Pandas allows you to work conveniently with data tables (DataFrames). Support for filtering, aggregation,

transformation and saving operations of data in various formats (CSV, Excel, JSON). Advantages: fast processing of large amounts of data and flexibility in working with text data. Disadvantages: Can be resource-intensive for massive datasets.

4.6. Parsing Tools Comparison

Scraping comments from social media is a challenging task due to the dynamic nature of the content generated by JavaScript. In this context, Selenium is compared to other tools such as BeautifulSoup, Instaloader, and Instagram Graph APIs.

1. BeautifulSoup is a library for analysing HTML and XML documents. Advantages: ease of use and flexibility in working with static web pages. Disadvantages include not supporting dynamic content generated by JavaScript and being less effective for complex pages.
2. Instaloader – A tool for downloading data from Instagram, including comments, posts, and profiles. Advantages: ease of setup and the ability to get data from public profiles without using the API. Disadvantages include instability when working with large amounts of data, a lack of support for dynamic content, and the risk of account blocking due to policy violations.
3. Instagram Graph API – The official tool for accessing Instagram data. Advantages: legality and compliance with Instagram's privacy policy and access to structured data. Disadvantages: rate limits, only works with business and creator accounts, and does not support access to private profiles.
4. Selenium is a browser automation tool that allows you to work with dynamic content. Advantages: The ability to interact with page elements that JavaScript generates. Flexibility in setting up parsing for specific tasks. Realistic behaviour that reduces the risk of blocking.

Disadvantages include high resource requirements and a relatively low work speed.

Among the tools reviewed, Selenium is the best choice for scraping comments from Instagram due to its ability to work with dynamic content. BeautifulSoup and Instaloader are suitable for static pages, but their limitations in working with JavaScript make them less effective for the task. The Instagram Graph API is a legal and stable tool, but its access restrictions reduce flexibility.

4.7. Comparison of visualisation tools

Data visualisation is an essential step in presenting the results of the analysis in a clear and understandable form. In this context, Plotly, Seaborn, Matplotlib and Tableau are compared.

1. Seaborn and Matplotlib are libraries for creating static graphs in Python. Advantages: ease of use, a large set of graph types (bar graphs, heatmaps, line graphs) and aesthetics of graphs (especially Seaborn). Disadvantages: lack of interactivity.
2. Plotly – a library for creating interactive graphs. Advantages: Interactivity: the ability to zoom, select individual segments, and interact with graphs; Support for complex types of graphs, including time series and 3D visualisations; Easy integration with web applications. Disadvantages: Higher resource requirements compared to static libraries.
3. Tableau is a commercial tool for creating interactive dashboards. Advantages: intuitive interface and powerful capabilities for interactive data analysis. Disadvantages: high license cost and less flexibility in configuring graphs compared to Python libraries.

Seaborn was chosen to create visualisations in the system under development due to its ease of use, aesthetically pleasing static graphs, and integration with Matplotlib. While Plotly offers extensive options for interactive graphs, the project's needs do not require a high level of interactivity, making Seaborn an optimal choice. Matplotlib remains an essential tool for basic

rendering, but its syntax is less user-friendly compared to Seaborn. Tableau, despite its capabilities, was rejected due to its high cost, which does not align with the project's budget.

4.8. Justification for the choice of methods and means

The development of a system to automate the analysis of comments in social networks requires the use of methods that provide high accuracy, performance, and adaptability to the specific characteristics of texts from social networks. The methods chosen have been carefully selected to meet these requirements.

1. Transformer models such as XLM-RoBERTa, RuBERTa, and RoBERTa are the modern standard in natural language processing (NLP). They achieve high accuracy in sentiment analysis by considering the context of each word within the text. Unlike traditional dictionary methods, Transformers can effectively work with multilingual content, including Ukrainian, Russian, and English. The ability to learn from specific data from social networks allows you to adapt models to slang, abbreviations and emojis that are often found in comments.
2. Langdetect is a quick and easy tool for determining the language of texts. It supports over 50 languages and integrates seamlessly with Python projects. Despite its limitations for short texts, its speed and simplicity make it an optimal choice for basic language definition. Alternatives like FastText provide higher accuracy but require more resources and are more Python is a versatile programming language widely used for tasks such as word processing, data analysis, and machine learning. Its rich set of libraries, such as Pandas, Transformers, Seaborn, Matplotlib, and Selenium, provides flexibility and performance in implementing complex systems. The simplicity of Python's syntax enables you to quickly develop and test new features, which is especially important for projects that require adaptability to change.

To address the system's problems, tools were selected that provide efficiency, flexibility, and accuracy at each stage of data processing.

1. Selenium for parsing dynamic content. Instagram generates comments dynamically using JavaScript, making them difficult to collect using traditional tools like BeautifulSoup or Instaloader. Selenium allows you to interact with web pages the way a real user does: click buttons, scroll pages, and upload new comments. It makes Selenium the best choice for working with dynamic content despite its resource requirements.
2. Seaborn to visualise results. Visualising the results is a crucial part of the system, as it enables users to easily analyse the data obtained. Seaborn is built on top of the Matplotlib library, allowing you to create informative and aesthetically pleasing graphs with minimal code. Unlike Plotly, which provides interactivity, Seaborn is easier to use for building standard visualisations that are sufficient for this system. It integrates well with Pandas DataFrame and allows you to quickly build distribution charts, heatmaps, boxplots, and other typical analytics graphs.

Specialised models for each language:

1. The use of separate models for Ukrainian, Russian, and English (XLM-RoBERTa, RuBERT, RoBERTa) yields high accuracy in sentiment analysis for each language group.
2. Additionally, the RoBERTa model is utilised to analyse texts that consist solely of symbols or emojis, enabling consideration of the unique characteristics of texts from social networks.
3. Unlike universal models such as mBERT, specialised models are better adapted to the grammatical and semantic features of each language.

The methods and tools chosen have been carefully selected to ensure the system's efficiency, accuracy, and adaptability. The use of transformer models ensures high accuracy in sentiment analysis for multilingual content, while langdetect provides rapid detection of the language of texts. Python and its libraries offer flexibility in system implementation, and Selenium enables efficient interaction with dynamic Instagram content. To visualise the results, seaborn was chosen. All these tools were selected with consideration for the specific tasks that the system addresses, ensuring its reliability and performance.

A detailed analysis of the methods and tools used to implement a system for automating the analysis of comments from social networks has been conducted. The chosen approaches were carefully selected, taking into account the specifics of the tasks that the system solves, such as multilingual text analysis, working with dynamic content, and interactive visualisation of results. The comparative analysis with analogues confirmed that the selected methods and tools provide: high accuracy, flexibility, performance and interactivity. Transformer models surpass dictionary approaches in terms of contextual analysis and multilingual capabilities. The use of Python and its libraries enables you to adapt the system to new tasks and the specific requirements of texts from social networks. Selenium works efficiently with dynamic content, while Plotly provides convenient data visualisation. Visualisation tools like Plotly enable you to create graphs that facilitate easier analysis of the results. The selected methods and means give solutions to the tasks of automation, accuracy, and interactivity. The system automatically collects, processes, and analyses comments, which significantly reduces labour costs. The use of modern transformer models provides accurate sentiment analysis for multilingual content. Visualisation tools enable users to easily interact with the analysis results and gain valuable insights.

Thus, the chosen methods and tools enable the effective implementation of a system that meets the modern requirements for analysing texts in social networks, ensuring accuracy, performance, and ease of use.

5. Software Development

Software development is a crucial stage in creating a system to automate the analysis of comments from social networks. The primary purpose of the software is to automate the collection, processing, analysis, and visualisation of text data generated by Instagram users. The system is designed to address problems related to multilingual text analysis, mood detection, trend analysis over time, and the interactive presentation of results. The developed software allows:

1. Automatically Collect Comments from Instagram – Using dynamic content scraping tools like Selenium ensures efficient text data collection.
2. Process text data – Text preprocessing involves cleaning, normalising, and determining the language of each comment.
3. Perform sentiment analysis – integration of modern transformer models (XLM-RoBERTa, RuBERT, RoBERTa) provides accurate analysis of comment sentiment for Ukrainian, Russian and English.
4. Build mood trends over time – the system enables you to identify the dynamics of mood changes over time.
5. Visualise Results – The use of libraries for interactive visualisation (Plotly) allows you to create graphs that facilitate the analysis of the data obtained.

The main tasks that the system solves:

1. Automation of the collection and processing of large amounts of text data.
2. Providing support for multilingual text analysis, including Ukrainian, Russian and English.
3. Improving the accuracy of sentiment analysis using modern machine learning models.
4. Creation of interactive tools for the visualisation and analysis of results.

5. Building an all-in-one system that can be adapted to other social media platforms.

In the process of designing and implementing the software, a modular approach was used, which involves dividing the system into separate functional modules:

1. Data upload module (Load_Data.ipynb) – collecting data from sources and writing to the database.
2. Data Processing Module (Process_Data.ipynb) – Text Cleaning, Language Detection, and Normalisation.
3. Mood Analysis Module (Analyze_Data.ipynb) – the use of transformer models to determine the emotional polarity of texts.
4. Data Aggregation Module (Aggregate_Data.ipynb): Formation of the final dataset for visualisation.
5. Visualisation module (Visualize_Data.ipynb) – construction of graphs, trends, and interactive reports.

Each module implements a separate function, providing flexibility, scalability, and convenience in system development and maintenance. To store data, a relational database is used, which consists of several tables corresponding to different stages of data processing. The developed software is a universal tool for automating the analysis of texts from social networks. It provides efficient processing of large amounts of data, accurate sentiment analysis, and interactive visualisation of results. The use of modern standards for the development and execution of documentation ensures the clarity and structure of all system components, facilitating their use and further improvement. A database is a key component of a developed system, as it provides storage, organisation, and access to data at various stages of processing. The primary purpose of the database is to store text comments, sentiment analysis results, and their metadata. It is designed to support all stages of the system, from collecting raw comments to forming the final dataset for visualisation. The database consists of four main tables:

1. Unprocessed saves raw comments that have been collected from Instagram.
2. Language_Marked_Comments saves comments with a defined language and clears unnecessary characters.
3. Sentiment_Analysis stores sentiment analysis results for each comment.
4. The Final Dataset forms the basis for visualisation and analysis.

This structure allows you to organise the data processing process in a precise sequence, providing transparency and flexibility in working with data. Description of database tables:

1. The Unprocessed table stores raw comments collected from Instagram at the initial stage of system operation. The Comment field (text) is the text of the comment that was collected from the social network. This table serves as the entry point for data in the system. It contains raw text data that has not been processed yet.
2. Table Language_Marked_Comments stores comments for which the primary language is defined, as well as cleaned texts. This table is used to store data after the preprocessing step. It contains cleaned comments and a defined language group. Field: Main_Language (text) – the primary language of the commentary (for example, uk, ru, en); Filtered_Comment (text) – cleaned comment text after normalisation (without unnecessary characters, emojis, URLs, etc.).
3. Table Sentiment_Analysis stores the sentiment analysis results for each comment. The table presents the results of transformer models analysing the emotional polarity of texts. Field: Main_Language (text) – the primary language of the commentary; Filtered_Comment (text) – cleaned text of the comment; Sentiment (text) – the result of sentiment analysis (for example, positive, negative, neutral).

4. Table Final_Dataset stores the final set of data that is used for graphing, trend analysis, and visualisation. This table is the result of data processing and is used for visualisation and further study. Field: Id (integer) – the unique identifier of the record; sub_id (integer) – identifier of a subgroup or category; Main_Language (text) – the primary language of the commentary; Sentiment (text) is the result of sentiment analysis; Filtered_Comment (text) – cleaned text of the comment.

The database diagram is presented in the form of an ER diagram (Entity-Relationship Diagram), which displays the structure of tables and their relationships between tables:

1. Unprocessed ⟶ Language_Marked_Comments – Data from the Unprocessed table is passed to the table Language_Marked_Comments after a preprocessing step that includes text scraping and language detection.
2. Language_Marked_Comments ⟶ Sentiment_Analysis – Table Language_Marked_Comments is a source of data for sentiment analysis. The cleaned comments are sent to the sentiment analysis module, and the results are recorded in the Sentiment_Analysis table.
3. Sentiment_Analysis ⟶ Final_Dataset – Data from table Sentiment_Analysis is combined with other metadata (e.g., id, sub_id) to form the final dataset in table Final_Dataset. This kit is used for visualisation and analysis.

The database is designed to meet the needs of the automation system for analysing comments in social networks. Its structure provides a logical sequence of data processing: from collecting raw comments to forming a final set for visualisation. Using tables with well-defined fields and establishing relationships between them allows you to ensure transparency and efficiency of the system. An ER diagram clearly demonstrates the relationships between tables, making it easier to understand the database's structure. Let's describe the software tool in detail.

I. The load_data.ipynb module is designed to automate the process of collecting data from the social network Instagram. The primary function of the module is to download raw comments from the posts of a specific Instagram profile and store them in the Unprocessed table of the database. The module is implemented based on the Selenium library, which provides access to dynamic content generated by JavaScript. The main components of the module are:

1. Functions for working with files and saving data: get_next_filename generates a unique filename to save comments; save_comments collects comments from the post and saves them to a CSV file.
2. Functions for working with the browser: setup_driver configures the browser driver (Chrome) to work with Instagram; login_to_instagram performs user authorisation on Instagram; save_cookies and load_cookies store and download cookies to automate re-logins.
3. Functions for data collection: scroll_and_load_comments loads all comments using scrolling; collect_comments collects comment texts from the page; load_all_posts downloads all posts of the selected profile; collect_all_hrefs collects links to profile posts.
4. The main script performs authorisation on Instagram, uploads profile posts, and saves the comments of each post to a file.

Actual description of functions:

1. get_next_filename generates a unique filename to save comments – checks for a folder to save files. If the folder does not exist, create it. It also analyses the files in the folder, determines the maximum file number, and generates a new name accordingly. Returns the path to a new file in the format base_filename_<number>.csv.

2. setup_driver sets up the Chrome browser to work with Instagram. Sets browser settings, such as language and window size, and turns off automation. Uses webdriver. Chrome to launch the browser. Returns a browser driver object.
3. login_to_instagram performs user authorisation on Instagram. Opens the Instagram login page. Fills in the login and password fields, presses the login button. Checks if authorisation has been completed. The result is a successful login or an error message.
4. save_cookies stores cookies in a file for reuse, and uses Pickle to save cookies to a file. load_cookies downloads cookies from a file to automate logins and downloads cookies from the file, adding them to the browser. The result is the successful storage or download of cookies.
5. scroll_and_load_comments loads all comments by scrolling. Uses the comment container scrolling to load new data. Re-scrolls until all comments are loaded. The result is that all comments from the page are loaded.
6. collect_comments collects the texts of comments from the page. Uses XPath to search for comment items. Extracts the text of each comment and adds it to the list. The result is a list of comment texts.
7. load_all_posts downloads all profile posts. Uses profile page scrolling to load posts. Collects links to all posts using the collect_all_hrefs function. The result is a list of links to publications.
8. save_comments collects comments from the post and saves them to a file. Opens the post by URL. Loads all comments using scroll_and_load_comments and collect_comments functions. Saves comments to a CSV file. The result is that the data is saved to a file.

The load_data.ipynb module works as an input stage in the system. Main processes of functioning:

1. Authorisation – downloading cookies or logging in to Instagram.
2. Post Downloads – uses load_all_posts to get a list of links to posts.
3. Collect comments – for each post, a save_comments is performed that collects comments and saves them to the Unprocessed table.

The load_data.ipynb module is a critical component of the system, as it provides automated collection of raw comments from Instagram. Its features, such as authorisation, scrolling, collecting comments, and saving data, ensure the system's efficiency and reliability. The module integrates with other system components through the Unprocessed table, ensuring consistency in data processing.

II. The process_data.ipynb module is designed to process comments that have been collected using the load_data.ipynb module. The primary function of the module is to clean up texts, identify the primary language of comments, filter texts according to the defined language, and save the results in a Language_Marked_Comments database. The main components of the module are:

1. Functions for cleaning texts: Normalize_unicode normalises text to NFC form; Clean_text removes unnecessary characters, leaving only the letters of the Ukrainian, Russian and English alphabets.
2. The function for detecting the language is detect_main_language. specifies the primary language of the comment (uk, ru, en, symbols_only).
3. The function for filtering texts is filter_comment_by_main_language. leaves only words in the text that correspond to the alphabet of the defined primary language.
4. The function for processing CSV files is process_csv. processes the input CSV file, determines the language of each comment, cleans the text, and creates a new CSV file.
5. The function for working with files is get_next_filename. generates a unique filename to save the results.

6. The main script processes all CSV files from the input_folder folder, saving the results in the output_folder folder.

Description of functions:

1. normalize_unicode normalises the text to the NFC form to combine the combined characters. Uses the unicodedata.normalise function to normalise text. Returns normalised text.
2. clean_text cleans up the text, leaving only the characters of the Ukrainian, Russian and English alphabets. Uses regular expressions to remove all characters that do not belong to the specified alphabet. Returns the cleaned text.
3. detect_main_language determines the primary language of the commentary. Uses the langdetect library to parse text and choose the most likely language. If the text consists only of characters, it returns "symbols_only". If the language cannot be detected, it returns "unknown". The result is the code of the primary language (en, ru, en, symbols_only or unknown).
4. filter_comment_by_main_language leaves in the text only words that correspond to the alphabet of the defined primary language. Uses regular expressions to check every word in the text. Filters words that match the alphabet of the primary language. Returns text cleaned according to the main language.
5. process_csv processes the CSV file, determines the language of each comment, cleans the text, and creates a new CSV file. Loads data from the input CSV file. For each comment, it detects the primary language and cleans up the text. Saves the results in a new CSV file. The result is a new CSV file with two columns: Main Language and Filtered Comment.
6. get_next_filename generates a unique filename to save the results. Analyses files in a folder, determines the maximum file number, and generates a new name accordingly. Returns the path to the new file.

III. The process_data.ipynb module works at the second stage of data processing after collecting comments by the load_data.ipynb module.

1. Data Loading – loads raw comments from CSV files that were created by the load_data.ipynb module.
2. Text Cleanup – normalises the text, removes unnecessary characters, and leaves only the letters of the corresponding language.
3. Language Definition – analyses each comment to determine the primary language.
4. Text filtering – leaves words that match the alphabet of the primary language.
5. Save Results – creates a new CSV file with cleaned comments and languages defined.

Algorithms used:

1. The text normalisation algorithm utilises unicodedata.normalise function to combine characters.
2. The text cleaning algorithm removes all characters that do not belong to the Ukrainian, Russian, or English alphabets.
3. The language detection algorithm utilises the langdetect library to analyse the text and identify the primary language.
4. The text filtering algorithm leaves only words in the text that match the alphabet of the defined primary language.
5. The CSV file processing algorithm loads data from the input file, processes each comment, and stores the results in the output file.

The process_data.ipynb module is a crucial component of the system, providing text cleaning, determining the primary language of comments, and filtering texts based on a defined language. Its functions, such as text normalisation, language detection, and filtering, enable you to prepare data for the next stages of analysis, ensuring high-quality processing. The module integrates with other system components through the Language_Marked_Comments table, which provides a sequence of data processing.

IV. The analyze_data.ipynb module is designed to analyse the sentiments of comments that have been cleaned and classified by language by the process_data.ipynb module. The main Sentiment_Analysis components of the module:

1. The function for uploading models is load_models. It downloads transformer models for sentiment analysis.
2. The function for sentiment analysis is analyze_sentiment. Performs comment sentiment analysis using the appropriate model.
3. The function for processing CSV files is process_sentiment_analysis. Processes the input CSV file, analyses the sentiment of each comment, and generates a new CSV file containing the results.
4. The function for working with files is get_next_filename. Generates a unique filename to save the results.
5. The main script performs sentiment analysis for all CSV files in the input_folder folder, saving the results in the output_folder folder.

Description of functions:

1. load_models downloads transformer models to analyse moods based on language. Loads models from the Hugging Face (pipeline) library for each language group (en, ru, en, symbols_only). Returns a dictionary with loaded models. Model: Ukrainian (uk): cardiffnlp/twitter-xlm-roberta-base-sentiment; Russian (ru): blanchefort/rubert-base-cased-sentiment; English: cardiffnlp/twitter-roberta-base-sentiment; Symbols and emojis (symbols_only): cardiffnlp/twitter-roberta-base-sentiment.
2. analyze_sentiment performs an analysis of the tone of a comment depending on its language. Uses the appropriate model depending on the language of the comment (en, ru, en, symbols_only). Analyses the text of the comment and returns the result in the form of a tone (positive, neutral, negative). In the event of an error, returns the value "neutral". The result is the tone of the comment.
3. process_sentiment_analysis processes the CSV file, analyses the sentiment of the comments, and creates a new CSV file with the results. Loads data from the input CSV file. Checks for the required columns (Filtered Comment, Main Language). Uses the analyze_sentiment function to analyse the sentiment of each comment. Adds a new Sentiment column with analysis results. Saves the results to a new CSV file. The result is a new CSV file with columns Filtered_Comment, Main_Language, and Sentiment.
4. get_next_filename generates a unique filename to save the results. Analyses files in a folder, determines the maximum file number, and generates a new name. Returns the path to the new file.

The analyze_data.ipynb module operates in the third stage of data processing, following the cleaning and classification of comments by the process_data module.ipynb module. Main processes of functioning:

1. Data Upload – Loads cleaned comments from CSV files that were created by the process_data.ipynb module.
2. Sentiment Analysis – Uses transformer models to determine the tone of each comment.

3. Save Results – Creates a new CSV file with a Sentiment column that contains the results of the analysis.

Algorithms used:

1. Model Upload Algorithm – Loads transformer models through the Hugging Face (pipeline) library for each language group.
2. Sentiment analysis algorithm – uses the appropriate model depending on the language of the comment to determine the sentiment of the text. Converts model labels (LABEL_0, LABEL_1, LABEL_2) to text evaluations (negative, neutral, positive).
3. CSV file processing algorithm – downloads data from the input file, performs sentiment analysis for each comment and saves the results in the output file.

The analyze_data.ipynb module is a crucial component of the system, providing comment sentiment analysis using modern transformer models. Its features, such as uploading models, analysing sentiments, and processing CSV files, enable you to obtain accurate results for each comment. The module integrates with other system components through the Sentiment_Analysis table, ensuring the sequence of data processing and information preparation for subsequent analysis and visualisation stages.

V. The aggregate_data.ipynb module is designed to combine data from several CSV files that were created at the previous stages of processing (modules process_data.ipynb and analyze_data.ipynb). The primary function of the module is to collect all the data, assign unique identifiers to each record (id and sub_id), organise the data into predefined columns, and save the results to the Final_Dataset database table. The main components of the module:

1. The function for data processing is collect_and_prepare_data. Collects all files with input data, adds identifiers (id, sub_id) and combines them into a single dataset.
2. The function for working with files is get_next_filename. Generates a unique filename to save the merged data.
3. The main script merges all CSV files from the input_folder folder, adds IDs, and stores the result in the output_folder folder.

Description of functions:

1. collect_and_prepare_data collects all files with comments, adds unique identifiers (ID and sub_id) and combines all data into one dataset. Finds all CSV files in the input_folder. Sorts files by number in the name (e.g., comments_with_languages_filtered_1.csv, comments_with_languages_filtered_2.csv, etc.). For each file, it loads the data into a DataFrame format, adds an ID column that corresponds to the file number, and adds a sub_ID column that numbers each entry in the file. Combines all DataFrames into a single dataset. Orders the columns in a given order: id, sub_id, Main_Language, Sentiment, Filtered_Comment. Saves the result to the original CSV file. The result is a single combined dataset with organised data.
2. get_next_filename generates a unique filename to save the merged data. Analyses the files in the source folder, determines the maximum file number, and generates a new name accordingly. Returns the path to a new file in the format combined_comments_<number>.csv.

The aggregate_data.ipynb module works at the final stage of data processing. Main processes of functioning:

1. Data Collection – Downloads data from CSV files that were created by the process_data.ipynb and analyze_data.ipynb modules.

2. Add IDs – adds unique identifiers for each record: id (corresponds to the file number from which the data was obtained) and sub_id (numbering records within the same file).
3. Data Merge – Merges all files into a single dataset, arranges columns, and saves the result to the output file.

Algorithms used:

1. File collection algorithm – finds all files in the input folder, sorts them by number in the name.
2. Algorithm for adding identifiers – adds an ID column that corresponds to the file number. Adds a column sub_id that numbers each entry in the file.
3. Data Aggregation Algorithm – Combines all DataFrames into a single dataset. Arranges columns in a given order.
4. Data saving algorithm – generates a unique file name to save the results. Saves the merged dataset to the original CSV file.

The aggregate_data.ipynb module is the final component of the system, which combines data from different stages of processing into a single dataset. Its functions, such as adding identifiers and organising data, enable you to create a structured dataset for further visualisation and analysis. The module integrates with other system components through a table named Final_Dataset, ensuring the integrity of data processing and the preparation of information for the end user.

VI. The visualize_data.ipynb module is designed to analyse and visualise the results of processing comments from Table Final_Dataset. The primary function of the module is to create graphs, analyse sentiment trends, construct solidarity indices, and identify patterns in the data. The module enables users to gain a deep understanding of the sentiments expressed in comments, their distribution by language, length, and the relationships between different characteristics. The main components of the module:

1. Data analysis – uploading and preliminary analysis of data (checking for missing values, descriptive statistics). Calculation of additional metrics such as comment length, solidarity index, and sentiment positivity.
2. Data visualisation is the construction of graphs to display comments by language, mood, and length. Analysis of sentiment trends over time. Visualisation of solidarity indices and their distribution.
3. Interactive analysis – plotting for each post, comparing the sentiments of comments with the description of the post.

Description of functions:

1. Loading and preliminary analysis of data – loads the combined dataset from table Final_Dataset, checks it for missing values, and performs descriptive statistics. Loads data from a CSV file. Uses the info(), head(), and describe() methods to analyse the data structure. Checks for missing values in columns. Output of basic information about the dataset, and detection of missing values.
2. Visualisation of comment distribution – plotting to analyse the distribution of comments by post ID, languages, and sentiments. The result is graphs that show the distribution of comments according to different characteristics. Sets axis labels, titles, and legends for graphs. Uses the Seaborn library to create graphs: Countplot to distribute comments by post IDs; Barplot to distribute comments by language; Countplot to parse the sentiment of comments for each language and each ID.
3. Analysis of the length of comments – calculation of the length of comments (number of words) and analysis of their distribution. Adds a new comment_length column that calculates the number of words in each comment. Uses histplot to visualise the length

distribution of comments. Uses boxplot to analyse the relationships between comment length, language, and sentiment. The result is a graph of the distribution of comment lengths and their relationships with other characteristics.

4. Building a solidarity index – analyses how the tone of comments coincides with the tone of the post description. Determines the sentiment of the post description (first comment with sub_id = 1). Compares the tone of other comments to the tone of the post description. Calculates the percentage of matches for each post and language (percent_matches). Uses bar plots and box plots to visualise the solidarity index. The result is graphs that show the level of solidarity for each post and each language.
5. Sentiment Positivity Analysis – analyses the proportion of positive comments in the total number for each post and language. Adds a new is_positive column that determines whether the sentiment of the comment is positive. Calculates the proportion of positive comments for each post and language. Uses lowess (smoothing) to build trends of sentiment positivity over time. The result is graphs that show the change in sentiment positivity over time.

The visualize_data.ipynb module works at the final stage of data analysis. Main processes of functioning:

1. Data Loading – Loads the merged dataset from the Final_Dataset table created by the aggregate_data.ipynb module.
2. Data analysis – calculates additional metrics (length of comments, solidarity index, positivity of sentiment).
3. Visualisation – creates graphs to analyse the distribution of comments, sentiment trends, and solidarity indices.

Algorithms used:

1. Distribution analysis algorithm – uses countplot and barplot graphs to analyse the distribution of comments by language, sentiment, and post ID.
2. Comment length analysis algorithm – adds a comment_length column for each comment. Analyses the distribution of comment length using histplot and boxplot.
3. Solidarity index algorithm – calculates the percentage of matches between the tone of comments and the sentiment of the post description. Visualises the results using bar plots and box plots.
4. Sentiment Positivity Analysis Algorithm – calculates the proportion of positive comments for each post and language. Smoothes out sentiment positivity trends over time with lowess.

The visualize_data.ipynb module is the final component of the system, providing analysis and visualisation of the data processing results. Its features allow users to gain a deep understanding of the sentiment of comments, their distribution by language, length, and solidarity indexes. The use of modern visualisation libraries, such as Seaborn and Matplotlib, enables you to create informative graphs that simplify the interpretation of results. The developed software is a comprehensive system that automates the analysis of comments from social networks, including Instagram. The system consists of five main modules, each of which performs a well-defined function: data collection, word processing, sentiment analysis, result aggregation, and visualisation. This modular structure enables easy scaling of the system, adaptation to other platforms, and provides ease of use. Structure and functions of the developed software:

1. The load_data.ipynb module automates the process of collecting comments from Instagram posts and ensures that raw data is saved in a structured way.

2. The process_data.ipynb module cleans up the comment texts, determines the primary language of each comment and filters the text according to the language group.
3. The analyze_data.ipynb module utilises modern transformer models to analyse sentiments and assigns a sentiment score (positive, neutral, or negative) to each comment.
4. The aggregate_data.ipynb module combines data from different stages of processing into a single dataset and adds unique identifiers for each record.
5. The visualize_data.ipynb module analyses the results and creates graphs to visualise the distribution of sentiments, trends, solidarity indices, and other key characteristics.

The system addresses several urgent tasks related to text analysis in social networks, including automation of processes, multilingual support, accuracy of analysis, and data visualisation. Instead of manually collecting and analysing comments, the system provides a fully automated process, saving a significant amount of time and resources. The system supports the analysis of texts in Ukrainian, Russian, and English, making it universally applicable for use in different regions. The use of transformer models ensures high accuracy in sentiment determination, even for texts from social networks that contain slang, abbreviations, and emojis. Interactive graphs and reports enable users to quickly gain insights from large datasets.

The developed software provides efficient data collection, high-quality word processing, accurate sentiment analysis, and interactivity. Using Selenium allows you to automate the collection of comments from dynamic Instagram content. Normalisation, cleaning, and filtering of texts ensure high-quality input data for analysis. Transformer models enable you to consider the context of texts, thereby enhancing the analysis results. Visualising the results in the form of graphs makes the system user-friendly.

The developed software is a powerful tool for automating the analysis of social media comments. Its modular structure, support for multilingualism, and interactive visualisation of results make the system versatile and effective for use in various fields, including business intelligence, sociological research, and public opinion monitoring. The system fully meets its tasks and provides high-quality processing of large amounts of data.

6. Control example of program execution

This section presents a control example, designed to demonstrate the performance of the developed software and verify its compliance with the intended tasks. The proposed system comprises five main modules: data loading, processing, sentiment analysis, aggregation, and visualisation. Each of these modules performs a unique function in the overall process of data analysis, allowing you to provide an integrated approach to solving the problem.

A control example involves running all modules in a sequence determined by the system's logic. At each stage, the correctness of the modules will be verified, along with the analysis of intermediate and final results. The primary goal is to confirm whether the results align with expectations and to assess the effectiveness and accuracy of the assigned tasks. This section describes the process of executing a control case, starting from downloading data from external sources and ending with visualising the results obtained. In addition, a detailed analysis of the operation of each module will be conducted, and how well the system performs in relation to the tasks set at the design stage will be assessed. Particular attention will be paid to verifying the correctness of the modules' functioning, their interaction with each other, and the compliance of the results with expectations. Thus, conducting a control case will not only confirm the performance of the developed software but also identify possible weaknesses or shortcomings that may require further improvement. The results of the analysis will serve as a basis for conclusions about the effectiveness of the proposed approach and its practical value. Since the system is designed on a modular principle, let's start with the first module, which downloads data from the Instagram environment. The main module, Load_Data.ipynb, utilises an automated browser client

to log in to the Instagram page and post comments on a predefined page (Fig. 2). In our case, we access my account and use posts from the President's page to analyse comments (Fig. 3).

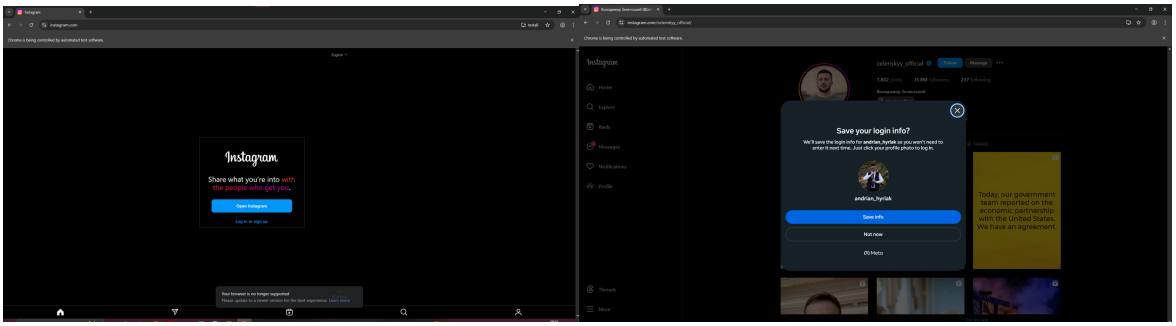


Figure 2: The welcome page of Instagram after running the script, and the appearance of the page after downloading cookies and going to the target page.

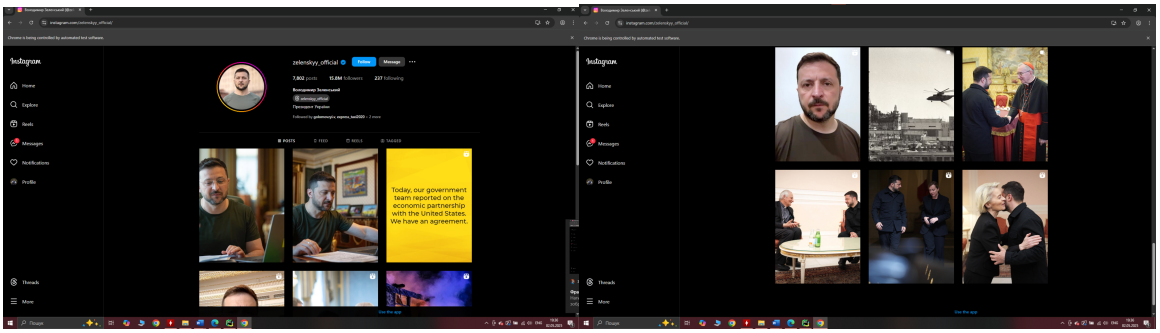


Figure 3: Page view after automatic closing of the extra wreath and automatic scrolling of the target page.

Here, you can see the process of opening each post and collecting comments directly (Fig. 4-5).

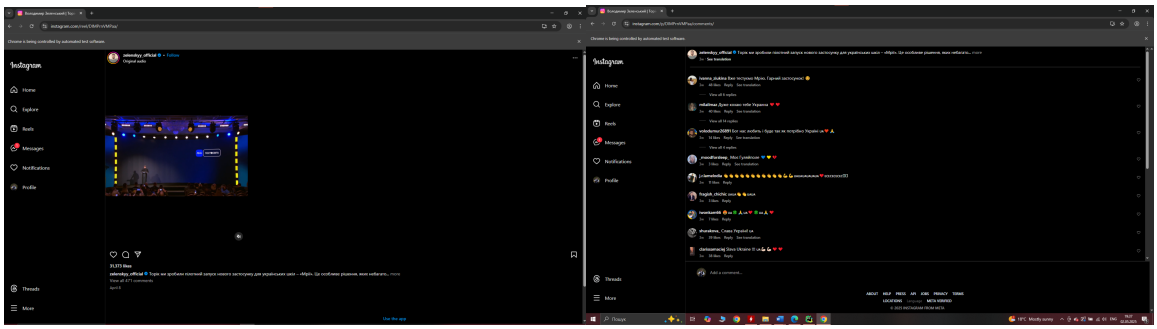


Figure 4: Opening the post link and opening the comments on the post.

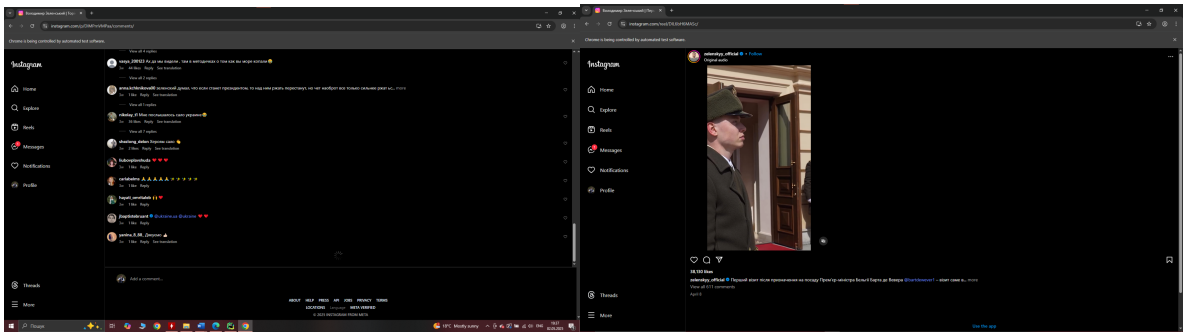


Figure 5: Automatic scrolling of comments and opening the next post.

In this picture, you can see the progress of the module inside the runtime (Fig. 6).

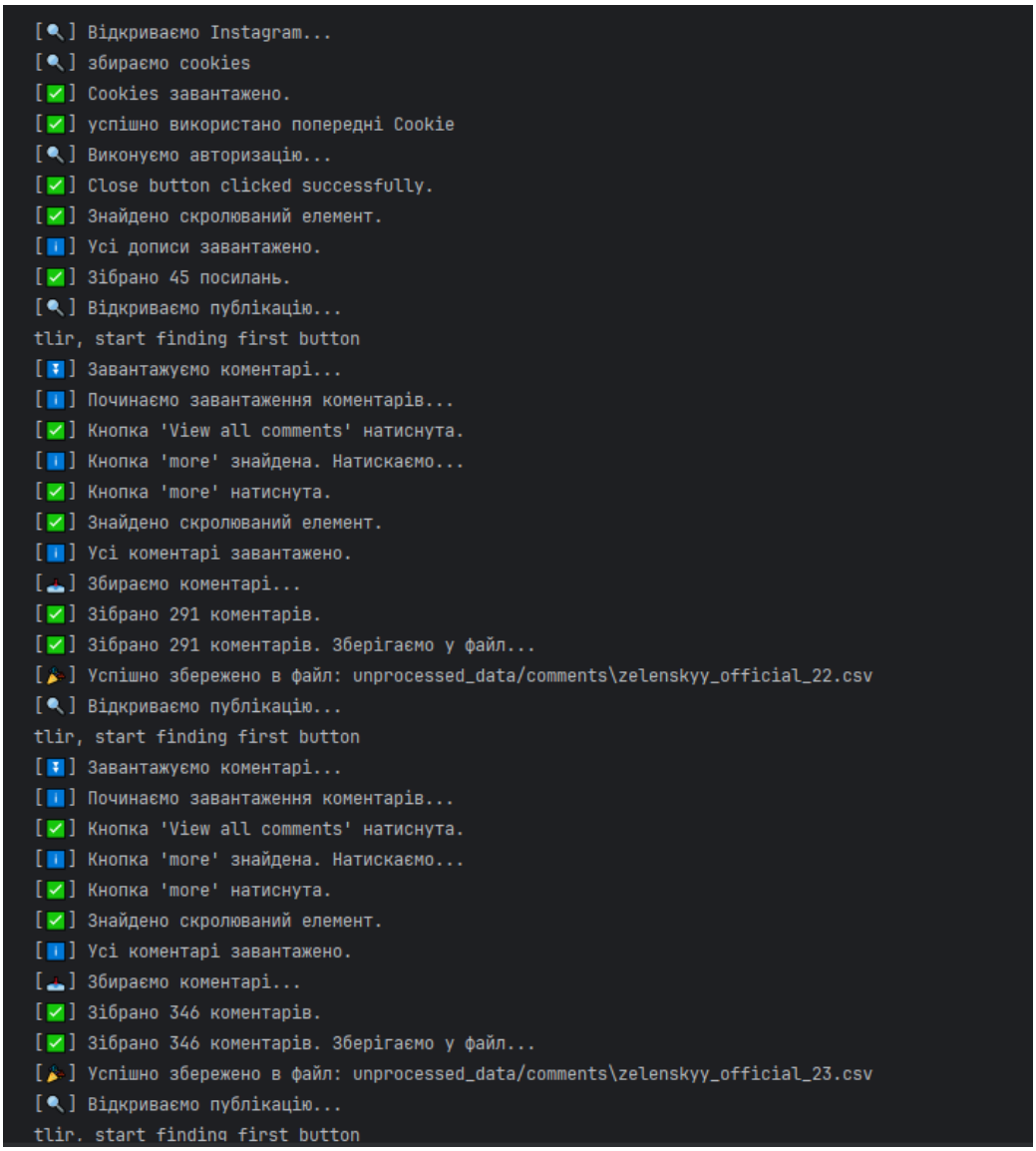


Figure 6: The progress of the module in the Jupiter notebook environment.

In the Load_comments module, you can see the database after the first module is executed. Here, the first comment is always a description of the post being analysed, and the subsequent ones are user comments. At this stage, each post is saved in a separate file.

Comment "Today, I presented state awards – the Crosses of Military Merit and Orders of the Golden Star – to our warriors and handed the orders to the family members of the Heroes who were posthumously awarded this title. Everyone who fights. Everyone who works for defence and truly for the state, not just for themselves. Everyone who helps. I thank you. I thank the families of our warriors – for such heroes, for such strength. We will undoubtedly endure this war. And we will certainly ensure a dignified life for Ukraine."

GLORY TO THE HEROES ❤❤❤

Glory to 💛💙❤️ Ukraine

God bless Ukraine👏👏💛💙🙏

No one will care as much as the President. He is a true blessingUA✌️🌹

Glory to Ukraine 💙💛❤️❤️

A great President to be respected, and thank you to the Warriors of Ukraine🙏💛💙

Ukraine will win ❤

In Process_data, all comments are divided into one of 4 languages, Ukrainian, Russian, English and symbolic (Fig. 7). If the language could not be detected, then the comment is skipped.

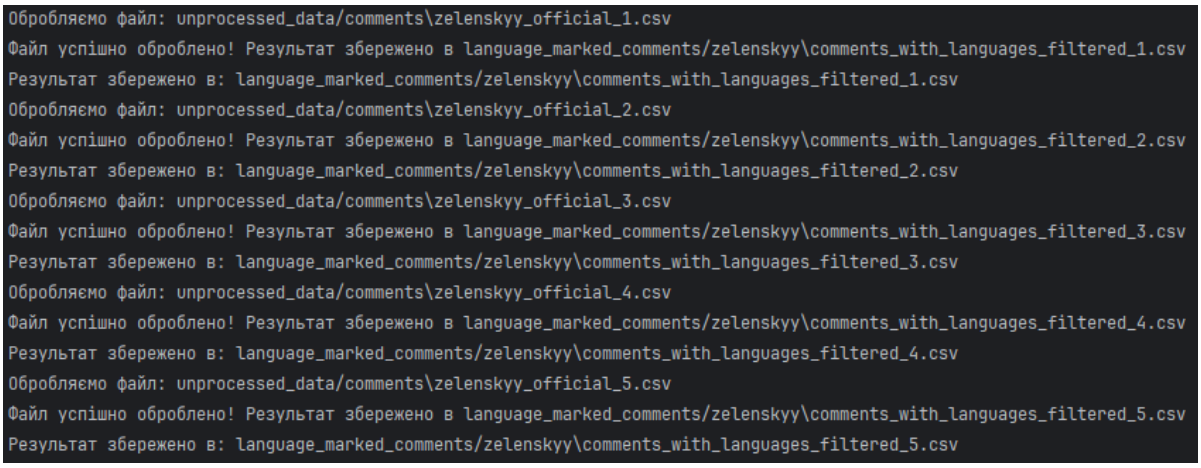


Figure 7: View of the module execution in the Jupiter environment.

Database view after executing the module Process_data:

Main_Language,Filtered_Comment
en, I presented state awards Crosses of Military Merit and Orders of the Golden Star to our and handed the orders to the family members of the Heroes who were posthumously awarded this Everyone who Everyone who works for defense and truly for the state not just for Everyone who I thank I thank the families of our warriors for such for such We will undoubtedly endure this And we will certainly ensure a dignified life for
ru, HEROES GLORY
symbols_only,symbols_only,symbols_only,en,No ❤
UAUA👏👏UAUA💛💙🙏
one will care as much as He is true
en, Glory to Ukraine
symbols_only,en,A ❤❤
great President to be respected and thank you to the Warriors of
uk,Win
en,is with
symbols_only, 🔥💪UAUAUA💪🔥
symbols_only, UK,take ❤❤❤❤❤❤❤❤❤❤❤
care of all and symbols_only,symbols_only,UK,MPs 💙💛PT❤️UA
' salaries and then there will be

The Analyze_data module uses transformer models that add a sentiment column to the database, where one of three values (positive, neutral, or negative) is possible (Fig. 8).

Database view after executing the module Process_data:

Main_Language,Filtered_Comment,Sentiment
en,I presented state awards Crosses of Military Merit and Orders of the Golden Star to our and handed the orders to the family members of the Heroes who were posthumously awarded this Everyone who Everyone who works for defense and truly for the state not just for Everyone who I thank I thank the families of our warriors for such for such We will undoubtedly endure this And we will certainly ensure a dignified life for,positive
ru,HEROES GLORY,neutral
symbols_only,💛💙❤️,positive
symbols_only,UAUA👏👏UAUA,positive
symbols_only,💛💙🙏,positive
en,No one will care as much as He is true,neutral
en,Glory to Ukraine,positive
symbols_only,❤❤,positive
en,A great President to be respected and thankyou to the Warriors of,positive
uk,Win,positive
en,is with,neutral
symbols_only,🔥💪UAUAUA💪🔥,neutral

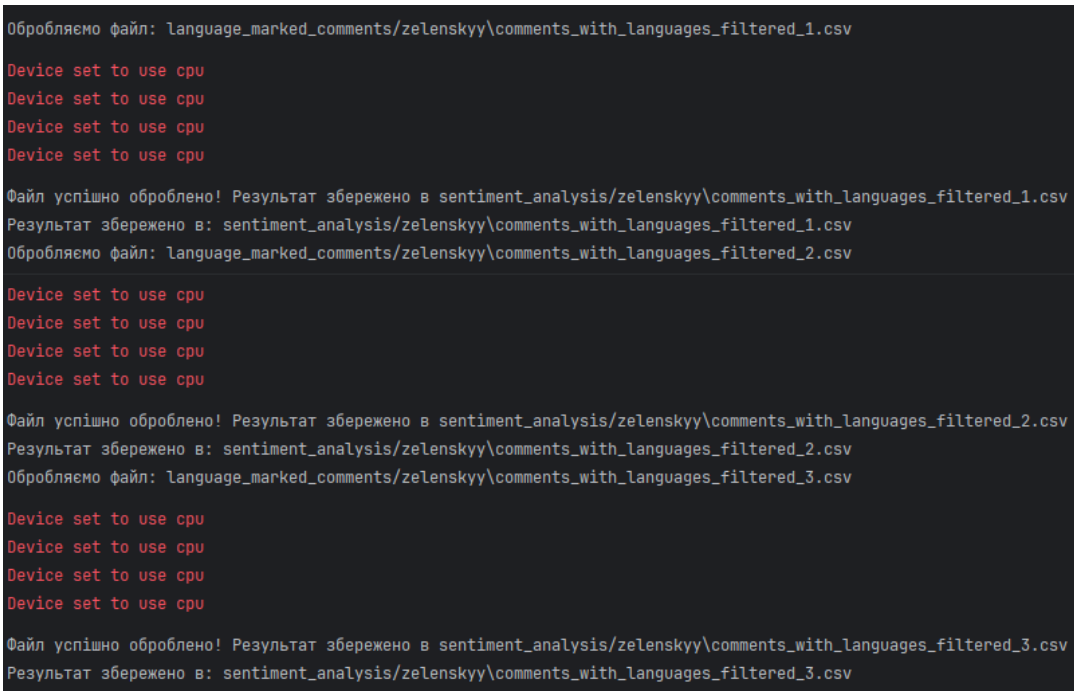


Figure 8: View of the module execution in the jupiter environment.

In the Aggregate_data module, all files with posts are numbered and combined into one file (Fig. 9), where the id column indicates the post number, with 1 being the newest and each subsequent number representing an older post. And the sub_id column means the order in which the comment is displayed, which is formed by Instagram algorithms based on likes and other indicators.

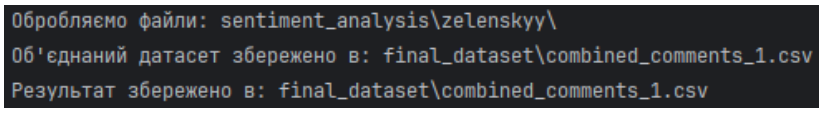


Figure 9: View of the module execution in the jupiter environment.

Database view after executing the aggregate_data module:

id,sub_id,Main_Language,Sentiment,Filtered_Comment1,1,en,positive,I presented state awards Crosses of Military Merit and Orders of the Golden Star to our and handed the orders to the family members of the Heroes who were posthumously awarded this Everyone who Everyone who works for defense and truly for the state not just for Everyone who I thank I thank the families of our warriors for such for such We will undoubtedly endure this And we will certainly ensure a dignified life for1,2,ru,neutral,HEROES GLORY1,3,symbols_only,positive,1,5,symbols_only,positive,1,6,symbols_only,positive,1,7,en,neutral ,No 💛💙❤️

UAUA👏👏UAUA💛💙🙏one will care as much as He is true1,8,en,positive,Glory to Ukraine1,9,symbols_only,positive,1 ❤❤, 10,en,positive,A great President to be respected and thankyou to the Warriors of1,11,uk,positive,Win1,12,en,neutral,is with1,13,symbols_only,neutral,1,14,symbols_only,positive,1,15,uk,neutral,save 🔥💪UAUAUA💪🔥 ❤❤❤❤❤❤❤❤❤❤❤all

and1,16,symbols_only,positive,1,17,symbols_only,positive,1,18,uk,neutral,MPs 💙💛PT❤️UA' salaries and then it will be1, 19,symbols_only,positive,1,20,en,positive,I 💙💛am continuously so impressed by President He seems to work so tirelessly for his country and God1,21,uk,neutral,Eternal1,22,uk,negative,Eternal and Light1,23,uk,neutral,how

The visualize_data module is the final module, in which the dataset is evaluated, and all the data within it is visualised. Additionally, the solidarity index is calculated, which represents the percentage of comments that match the author's description of the post. Here are some common characteristics of the dataset, shown in Fig. 10.

count	3887.000000	3887.000000
mean	11.407769	127.652174
std	5.514420	97.615940
min	1.000000	1.000000
25%	7.000000	49.000000
50%	12.000000	106.000000
75%	16.000000	186.500000
max	20.000000	438.000000

Пропущені значення:

id	0
sub_id	0
Main_Language	0
Sentiment	0
Filtered_Comment	0

dtype: int64

Figure 10: Descriptive statistics of the created dataset.

In the following visualisation, you can see the number of comments for each post, where the first is the most recent, the twentieth is the oldest (Fig. 11a). Next, you can see how many comments were written by each of the language categories (Fig. 11b).

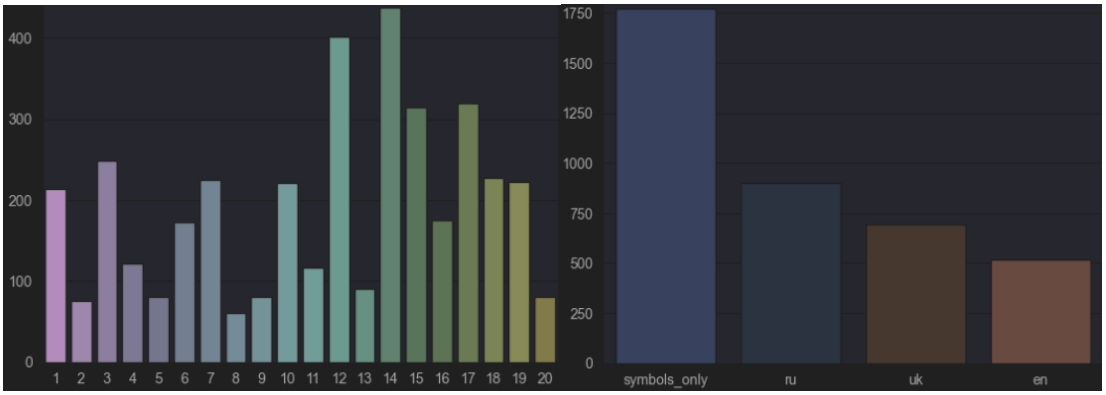


Figure 11: a) Number of comments on each post (distribution of comments by id), where X is the id of the post, Y is the number of comments, and b) Distribution of comments by languages, where X is the language, Y is the number of comments.

In the following images, you can see the tonality distribution for each language, as well as for each post (Fig. 12). Fig. 13 shows the distribution of comment lengths. Fig. 14 shows how the length of a comment depends on the key, as well as on the language in which it was written.

Figure 12: General distribution of comments by tone for: a) each language and b) each post by id, where X is the language, Y is the number of comments.

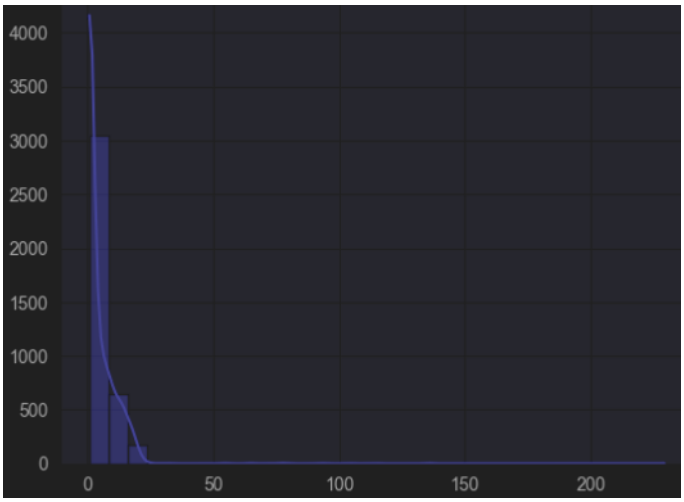


Figure 13: Distribution of comment length after filtering, where X is the length of the comment (number of words), and Y is the number of comments.

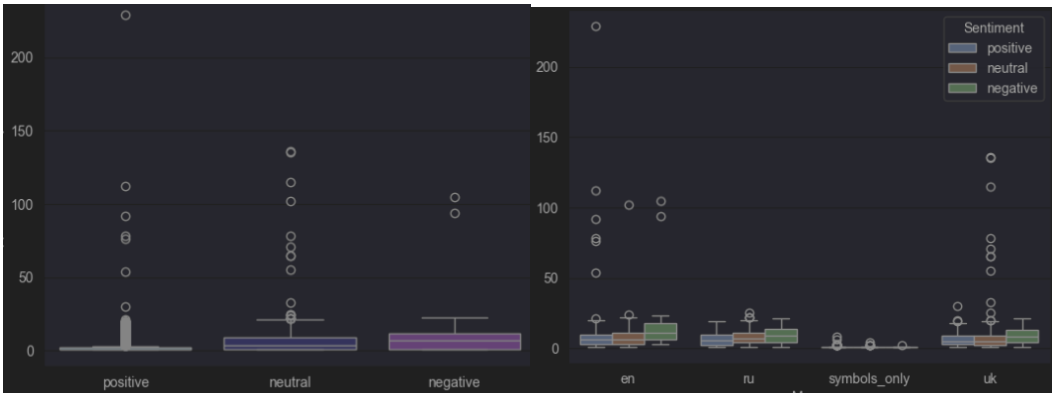


Figure 14: The relationship between a) the length of the comment and the tonality, b) the language, the length of the comment, and the tonality, where X is the tonality/language, Y is the length of the comment.

Fig. 15a analysed the longest comments. Fig. 15b presents the solidarity index in tabular form. Fig. 16a and Fig. 17a show the distribution of moods within the comments of each language. Next, the solidarity index for each language was analysed, as well as the sentiments expressed in the comments for each language across all posts (Fig. 16b and Fig. 17b).

	id	sub_id	Main_Language	Sentiment
0	1	1	en	positive
24	1	25	en	neutral
110	1	111	en	negative
214	2	1	uk	neutral
290	3	1	en	positive
500	3	211	en	negative
538	4	1	uk	neutral
659	5	1	uk	neutral
739	6	1	en	positive
818	6	80	ru	neutral
823	6	85	en	neutral
897	6	159	ru	neutral
912	7	1	en	positive

	id	Main_Language	percent_matches
0	1	en	45.161290
1	1	ru	4.255319
2	1	symbols_only	79.347826
3	1	uk	48.837209
4	2	en	33.333333
..	..	...	...
75	19	uk	40.909091
76	20	en	22.727273
77	20	ru	59.523810
78	20	symbols_only	50.000000
79	20	uk	37.500000

Figure 15: a) List of longest comments and b) solidarity index for each language in each comment.

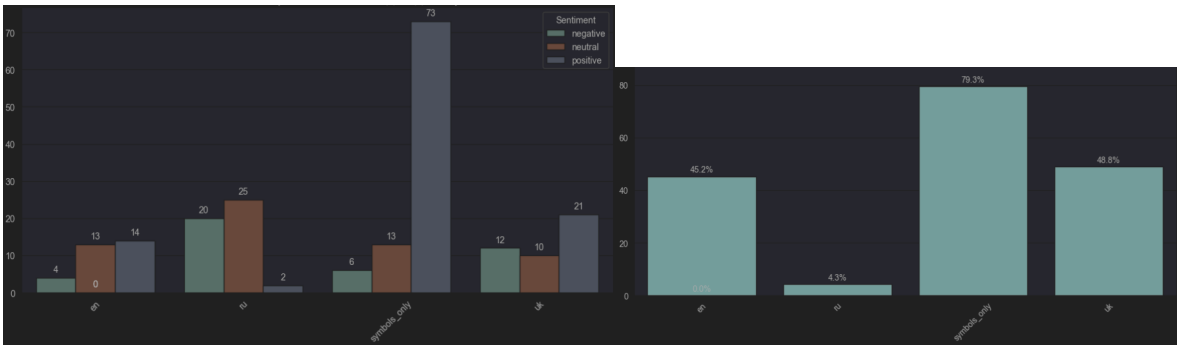


Figure 16: a) Detailed analysis of the sentiment of the comments on the first post, where X is the language, Y is the number of comments, b) the analysis of the solidarity index for the first post, where X is the language, Y is the percentage of matches.

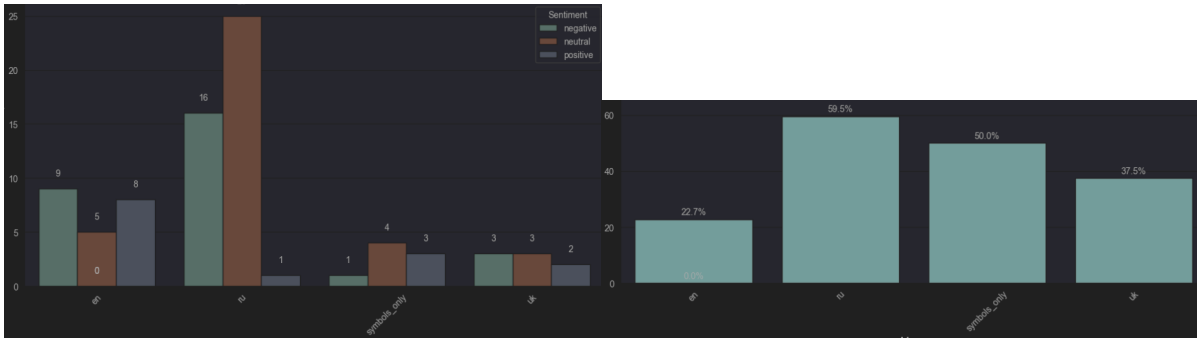
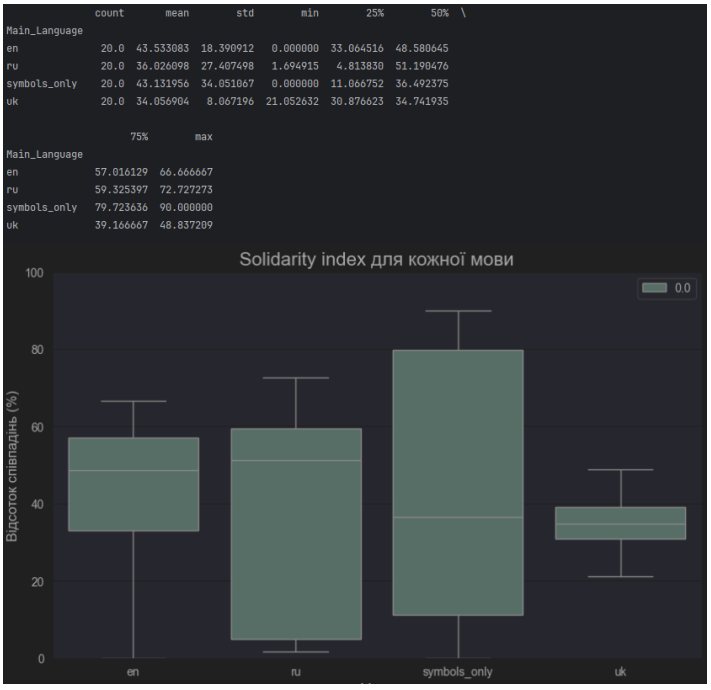


Figure 17: a) Detailed analysis of the sentiment of the comments to the last post, where X is the language, Y is the number of comments, b) the analysis of the solidarity index to the last post, where X is the language, Y is the percentage of matches.

In the following Figs. In Figures 18–20, you can see the characteristics of the distribution of the solidarity index for each of the languages.



Main_Language	count	mean	std	min	25%	50%	75%	max
en	20.0	43.533083	18.390912	0.000000	33.064516	48.580645	57.016129	66.666667
ru	20.0	36.026098	27.407498	1.694915	4.813830	51.190476	59.325397	72.727273
symbols_only	20.0	43.131956	34.051067	0.000000	11.066752	36.492375	79.723636	90.000000
uk	20.0	34.056904	8.067196	21.052632	30.876623	34.741935	39.166667	48.837209

Figure 18: Distribution of the solidarity index for all posts for each language, where X is the language, and Y is the percentage of matches.

Figure 19: Average solidarity index for each language, where X is the language, and Y is the average percentage of matches.

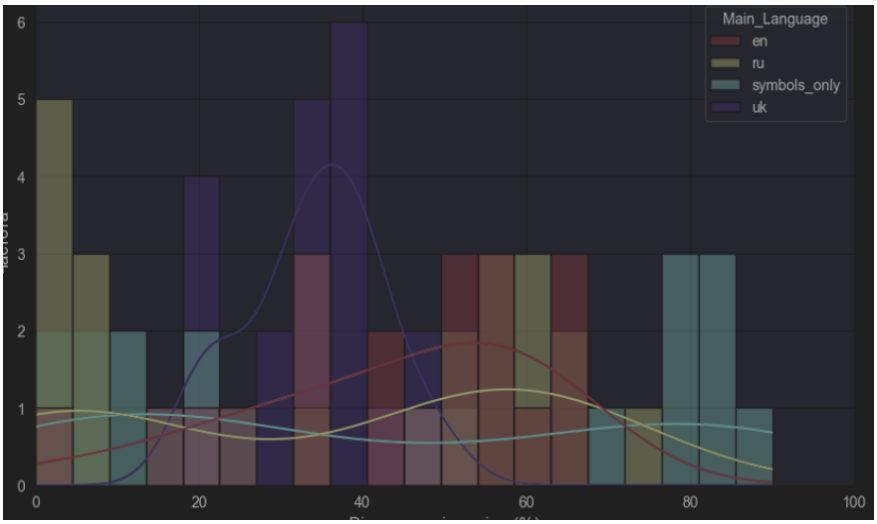


Figure 20: Distribution of the solidarity index for all languages, where X is the percentage of coincidences, and Y is the frequency.

Graphs in Figs. 21–25 show a change in the positivity of sentiments in the comments for each language over time. You need to read them from right to left, because 1 is the most recent post and 20 is the oldest. You can see that comments in Russian are very rarely positive, for apparent reasons (Fig. 22). Fig. 23 shows that comments from symbols are most often positive. The remarks in Ukrainian have the most stable positivity index, which fluctuates approximately within the range of 0.2-0.4 (Fig. 24). The positivity graph in Fig. 25, which combines comments from all languages, shows a rather volatile trend towards change.

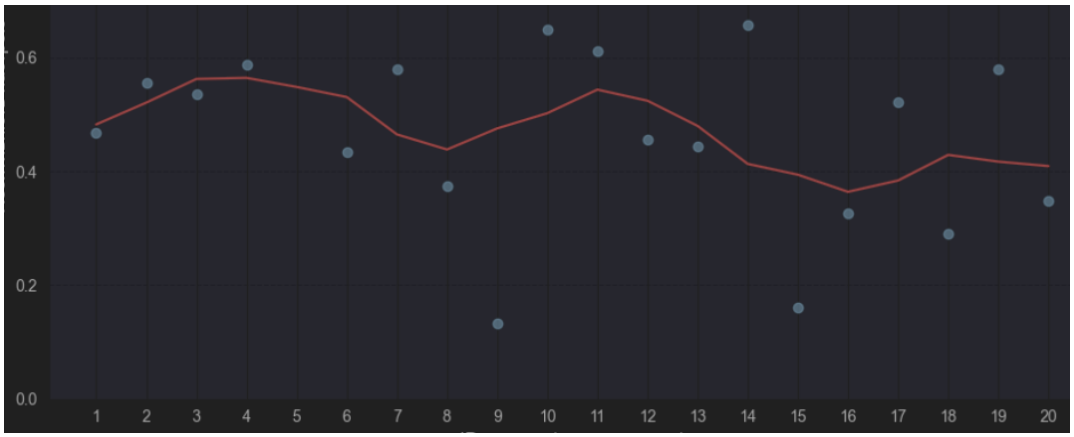


Figure 21: Trend of change in the positivity of the mood of comments over time for English, where blue dots are data, red line is a smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of the mood.

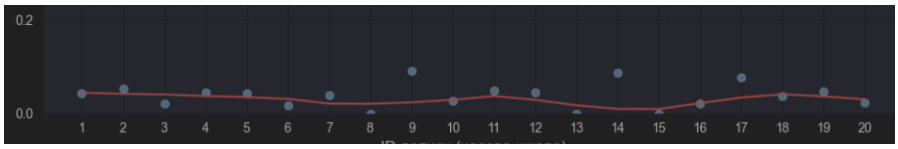


Figure 22: Trend of change in the positivity of the mood of comments over time for the Russian language, where blue dots are data, red line is a smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of the mood.

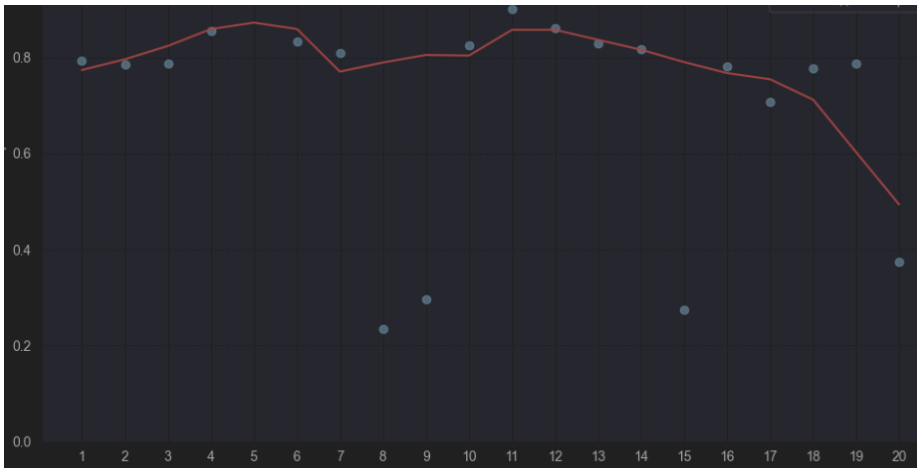


Figure 23: Trend of change in the positivity of the sentiment of comments over time for symbols, where blue dots are data, red line is a smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of mood.

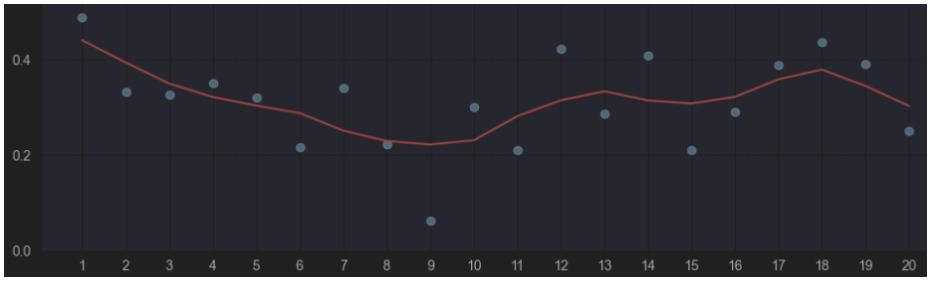


Figure 24: Trend of change in the positivity of the mood of comments over time for the Ukrainian language, where blue dots are data, red line is a smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of mood.

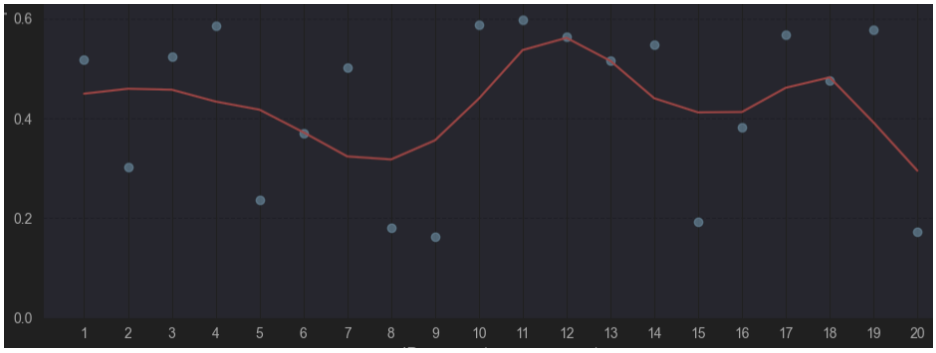


Figure 25: The general trend of change in the positivity of the mood of comments over time, where the blue dots are the data, the red line is the smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of the mood.

So, following this control example, with the help of existing modules, you can download and analyse comments on posts from any Instagram page.

7. Program Execution Statistics

In modern information systems, the efficiency of software is a key factor in determining its practical value. Performance analysis enables you to assess how efficiently the system performs tasks, identify potential bottlenecks, and determine ways to optimise. This section presents a statistical analysis of the system's implementation for automating comment analysis in social networks. The main attention is paid to the following indicators:

1. The execution time of the main modules is a measurement of the time required to complete each stage of data processing (collection, processing, sentiment analysis, aggregation, and visualisation).
2. Data processing – analysis of the number of processed comments at each stage.
3. Resources – An estimate of the use of computing resources such as RAM and CPU.

It is necessary to evaluate the system's efficiency, identify its strengths and weaknesses, and provide recommendations for enhancing performance. The statistics will be presented in the form of tables, graphs, and charts, allowing you to visualise the results of the analysis and make them more understandable to users. The results obtained will not only confirm the effectiveness of the developed system, but also provide valuable information for its further improvement. Let's proceed to the analysis of the program's main modules. Since selenium and an automatic browser client are used to load (Load_data) all comments as the basis of the dataset, the downloading process takes a significant amount of time. After all, to make the process as secure as possible and the script fault-tolerant, various approaches, such as WebDriverWait and Time.sleep(), are integrated into it (Fig. 26). The combination of these approaches avoids blocking the IP address due to excessive requests. Also, it ensures the fault tolerance of the script (Fig. 27a). Additionally, in guaranteeing the script's speed, storing browser cookies after the first authorisation plays a crucial role. It avoids a delay in logging into your account. Also, the performance of the script is affected by the number of posts uploaded to the target page, as well as the number of comments in each of them (Fig. 27b). In our case, when uploading 20 posts with an average of 180 comments in each of them, the execution time reached 36 minutes.

```
view_all_button = WebDriverWait(driver, 10).until(
    EC.element_to_be_clickable((By.XPATH, "//a[.//span[contains(text(), 'View all')]]"))
)
```

```
time.sleep(random.uniform(2, 4))
try_press_cancel_button(driver)
time.sleep(random.uniform(2, 4))
```

Figure 26: Waiting for the appearance of the View all element and artificially adding a delay between actions.

```
if load_cookie_success:
    for i in load_all_posts(driver):
        save_comments(driver, i)
else:
    login_to_instagram(driver, USERNAME, PASSWORD)
    save_cookies(driver)
    if "/accounts/login/" not in driver.current_url:
        for i in load_all_posts(driver):
            save_comments(driver, i)
    else:
        print(f"[✗] ./accounts/login/ in {driver.current_url}")
```

```
 "source": "#### основний скрипт",
 "id": "e5d2574525a8cae2"
},
{
 "metadata": {
  "ExecuteTime": {
   "end_time": "2025-04-28T07:38:32.571818Z",
   "start_time": "2025-04-28T07:02:03.650480Z"
```

Figure 27: a) Downloading Cookies and b) Execution Time of the Main Script.

The primary function of the Process_data module is to filter comments and determine the primary language in each of them (Fig. 28). In order to save time and resources of the system during scaling, a highly efficient Langdetect library was chosen, which made it possible to reduce the module's execution time to about 8 seconds on a dataset of almost 4000 entries. The primary function of the Analyze_data module is to utilise transformer models specific to each language to assess the mood of each comment and add the corresponding designation to the dataset (Fig. 29). This analysis, which involves processing comments through models, is the most time-consuming step. The longer the comment, the longer it will take to process. Since all the models were pre-loaded and most comments on social media are pretty brief, the execution time was only 4 minutes.

```
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

input_files = [f for f in os.listdir(input_folder) if f.endswith('.csv')]
input_files = sorted(input_files, key=lambda x: int(re.search(r'\d+', x).group()))
for input_file in input_files:
    input_csv_path = os.path.join(input_folder, input_file)
    output_path = get_next_filename("comments_with_languages_filtered", output_folder)

    print(f"Обробляємо файл: {input_csv_path}")
    process_csv(input_csv_path, output_path)
    print(f"Результат збережено в: {output_path}")
Executed at 2025.04.29 08:25:00 in 8s 812ms
```

Figure 28: Main script and execution time.

```
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

# Отримуємо список усіх файлів у вхідній папці
input_files = [f for f in os.listdir(input_folder) if f.endswith('.csv')]
input_files = sorted(input_files, key=lambda x: int(re.search(r'\d+', x).group()))
# Обробляємо кожен файл
for input_file in input_files:
    input_csv_path = os.path.join(input_folder, input_file)
    output_path = get_next_filename("comments_with_languages_filtered", output_folder)

    print(f"Обробляємо файл: {input_csv_path}")
    process_sentiment_analysis(input_csv_path, output_path)
    print(f"Результат збережено в: {output_path}")
Executed at 2025.04.29 08:34:02 in 3m 59s 925ms
```

Figure 29: Main script and execution time.

The Aggregate_data module aggregates all received files into a single dataset using the pandas library, so it does not require much time. It took less than a second to execute (Fig. 30).

```
output_path = get_next_filename("combined_comments", output_folder)

print(f"Обробляємо файли: {input_folder}\\")
collect_and_prepare_data(input_folder, output_path)
print(f"Результат збережено в: {output_path}")
Executed at 2025.04.29 08:57:25 in 38ms
```

Figure 30: Main script and execution time.

The Visualize_data module does not manipulate data; its primary function is to visualise an already created dataset (Fig. 31–32). Those graphs and statistics, which are built for the entire dataset, do not depend significantly on its size, and in general, each took less than a second. The only exception is the visualisation of mood trends, as we used the lowess function from the statsmodels library, which is picky about resources. However, the section for analysing individual posts took significantly longer – 6.2 seconds - and the time of its execution is directly proportional to the number of posts and comments within them.

```
file_path = "final_dataset/combined_comments_1.csv"
df = pd.read_csv(file_path)
print(df.info())
print(df.head())
print(df.describe())

print("Пропущені значення:\n", df.isnull().sum())
```

Executed at 2025.05.01 07:45:12 in 47ms

```
sns.countplot(x='id', data=df, hue='id', palette="viridis", dodge=False, legend=False)
plt.title('Розподіл коментарів за id')
plt.xlabel('ID допису')
plt.ylabel('Кількість коментарів')
plt.show()
```

Executed at 2025.05.01 07:45:12 in 285ms

Figure 31: Time for performing the analysis of the dataset and building one of the graphs on the entire dataset.

Executed at 2025.05.01 07:45:19 in 6s 235ms

	id	Main_Language	percent_matches
0	1	en	45.161290
1	1	ru	4.255319
2	1	symbols_only	79.347826
3	1	uk	48.837209
4	2	en	33.333333
..	..	...	...
75	19	uk	40.909091
76	20	en	22.727273
77	20	ru	59.523810
78	20	symbols_only	50.000000
79	20	uk	37.500000

[80 rows x 3 columns]

Executed at 2025.05.01 07:45:20 in 474ms

```
Описова статистика для кожної мови:
```

Main_Language	count	mean	std	min	25%	50%	75%	max
en	20.0	43.533083	18.390912	0.000000	33.064516	48.580645	57.016129	66.666667
ru	20.0	36.026098	27.407498	1.694915	4.813830	51.190476	59.325397	72.727273
symbols_only	20.0	43.131956	34.051067	0.000000	11.066752	36.492375	79.723636	90.000000
uk	20.0	34.056904	8.067196	21.052632	30.876623	34.741935	39.166667	48.837209

```
plt.tight_layout()
plt.show()
```

Executed at 2025.05.01 07:45:21 in 988ms

Figure 32: The cumulative time to perform the analysis of each post, the time to plot solidarity graphs for all languages, and the time to plot using the lowess function.

The analysis of the system's implementation statistics enabled an assessment of its effectiveness, performance, and compliance with the assigned tasks. The results confirmed that the system successfully handles the functions of collecting, processing, analysing, and visualising large amounts of text data. Key takeaways:

1. Module execution time. The stage of mood analysis using transformer models takes the most time, which is due to the high computational requirements of these models. Other modules, such as data collection, word processing, and aggregation, are relatively fast due to the use of efficient algorithms and streamlined approaches.
2. System performance. The system is capable of processing large amounts of data, maintaining high accuracy at every stage. The use of multilingual transformer models provides a qualitative analysis of moods for Ukrainian, Russian and English.
3. Use of resources. The main load falls on the stage of mood analysis, where transformer models are used. It requires significant computing resources, including RAM and CPU time. Other stages, such as text scraping, filtering, and data aggregation, have low resource requirements.
4. Analysis of processed data. The system demonstrates high efficiency in working with texts of varying lengths and complexities, including comments with symbols, slang, and mixed languages. Visualising the results makes it easy to interpret the data obtained, which is essential for end users.

Recommendations for improvement:

1. Optimisation of transformer models – use of less resource-intensive models (e.g. DistilBERT) for texts with low complexity, as well as integration of methods of preliminary classification of texts to determine which comments require detailed analysis.
2. Data collection optimisation – using Instagram's official API (subject to availability) to reduce comment collection time and avoid possible restrictions from the platform.
3. System scaling – the implementation of multithreading or distributed computing for parallel processing of large amounts of data, as well as the use of cloud services for the analysis of large data sets.

The developed system demonstrates high efficiency and productivity in solving the problems of analysing comments in social networks. It meets modern requirements for automating the processing of large amounts of text data, providing accurate sentiment analysis, multilingual capabilities, and convenient visualisation of results. However, further optimisation of computing resources and integration with other platforms will make the system even more versatile and productive.

8. Comprehensive analysis of different categories of Instagram accounts

As an additional task, a new dataset was created, which included more than 20000 comments. Let's conduct a comprehensive analysis of various categories of Instagram accounts. The first category we will consider is the business page (Fig. 33). This dataset includes 1600 comments and three different pages (Fig. 34).

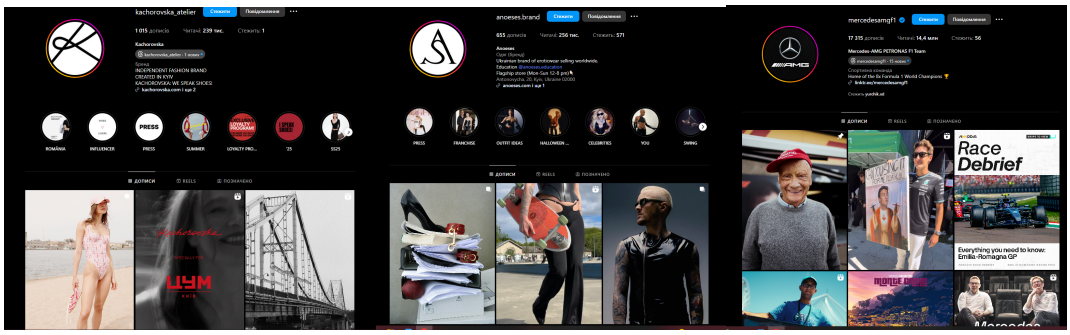


Figure 33: Business Page Example.

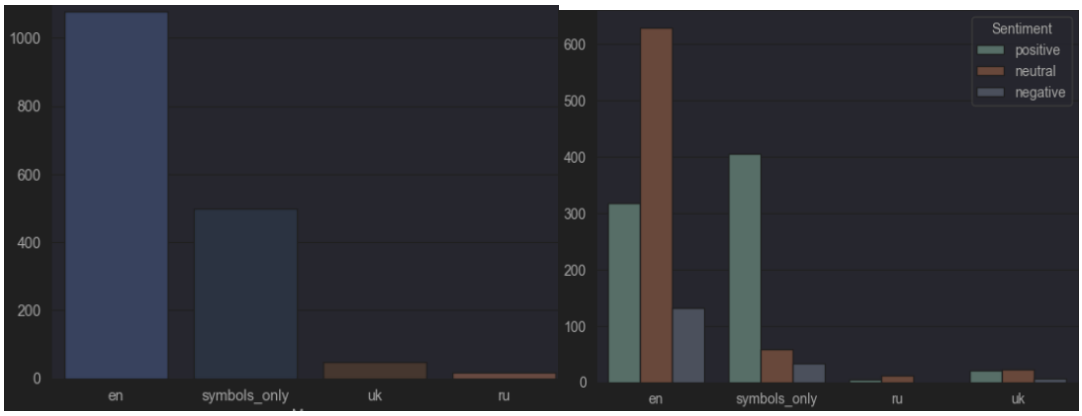


Figure 34: a) Distribution of comments by language and b) tone of comments for each language, where X is the language, Y is the number of comments.

Therefore, even Ukrainian business accounts primarily focus on the English language. As expected, according to Fig. 35, symbolic comments contain the most positive dynamics, while in other languages, neutral comments prevail. On average, comments for business accounts are not very long, but they are nevertheless longer than those for personal pages (Fig. 36–37). In these images, you can see that the comments in business accounts for the most part coincide in tone with the description from the author of the page.

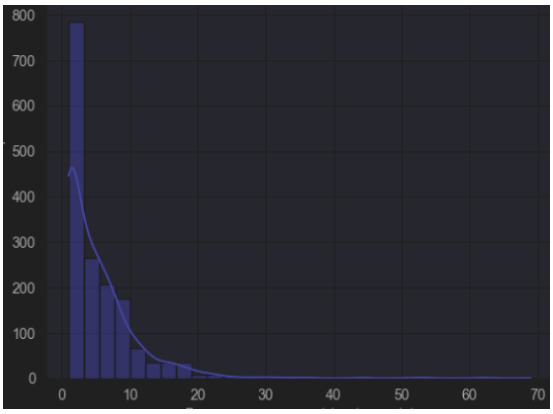


Figure 35: Distribution of comment length after filtering, where X is the length of the comment (number of words), and Y is the number of comments.

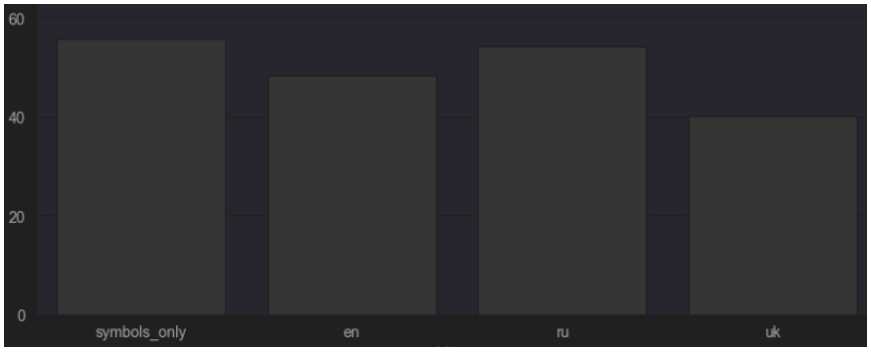


Figure 36: Average solidarity index for each language, where X is the language, and Y is the average percentage of matches.

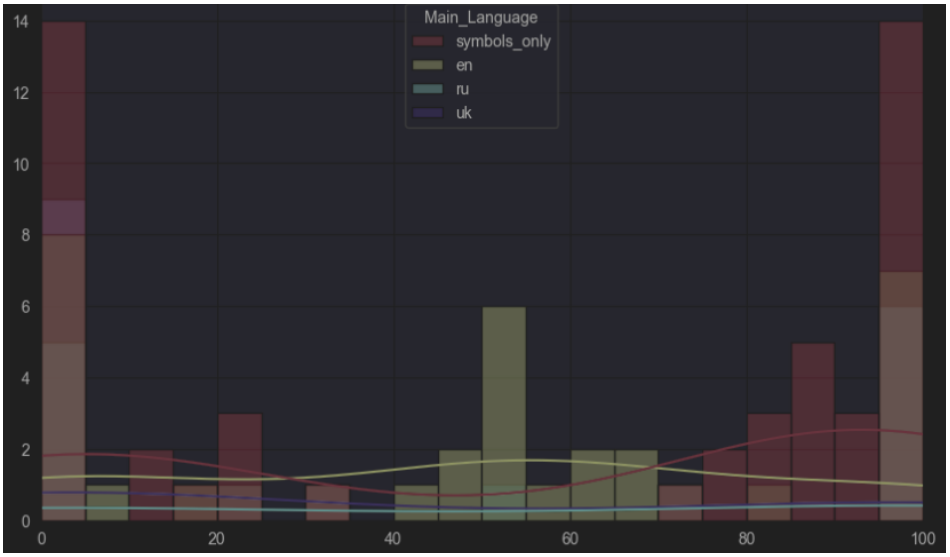


Figure 37: Distribution of the solidarity index for all languages, where X is the percentage of coincidences, and Y is the frequency.

Let's proceed to the analysis of the positivity of comments (Figs. 38-41). You can see that most comments in this topic are neutral or positive, with a clear trend towards improving the sentiment of comments over time.

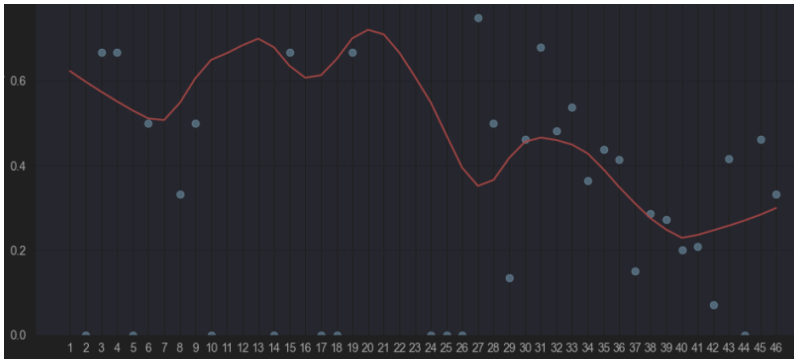


Figure 38: Trend of change in the positivity of the mood of comments over time for English, where blue dots are data, red line is a smoothed trend, where X is the id of the post (timeline), and Y is the positivity of the mood.

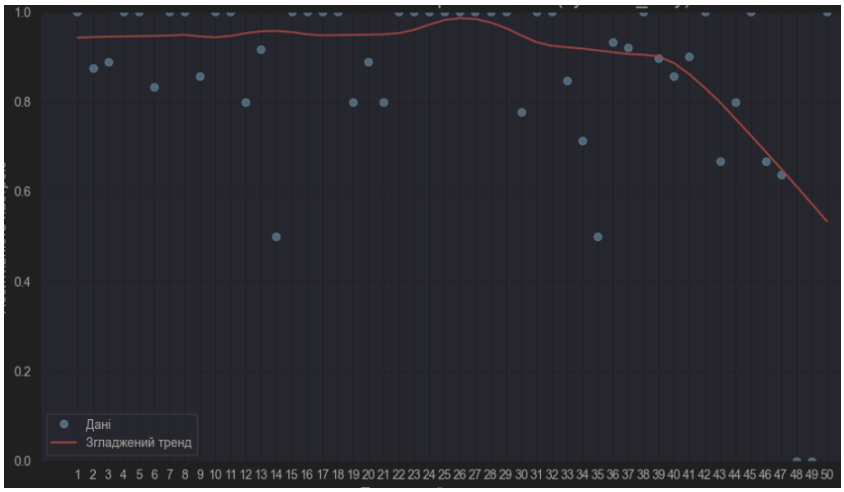


Figure 39: Trend of change in the positivity of the sentiment of comments over time for symbols, where blue dots are data, red line is a smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of sentiment.

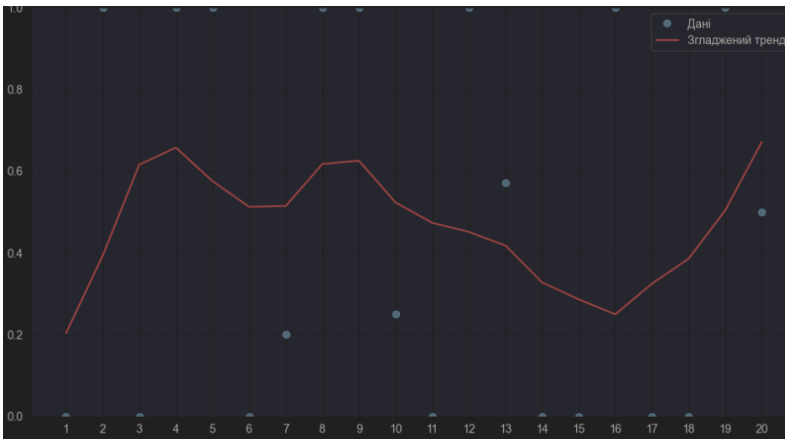


Figure 40: Trend of change in the positivity of the mood of comments over time for the Ukrainian language, where blue dots are data, red line is a smoothed trend, where X is the id of the post (timescale), and Y is the positivity of mood.

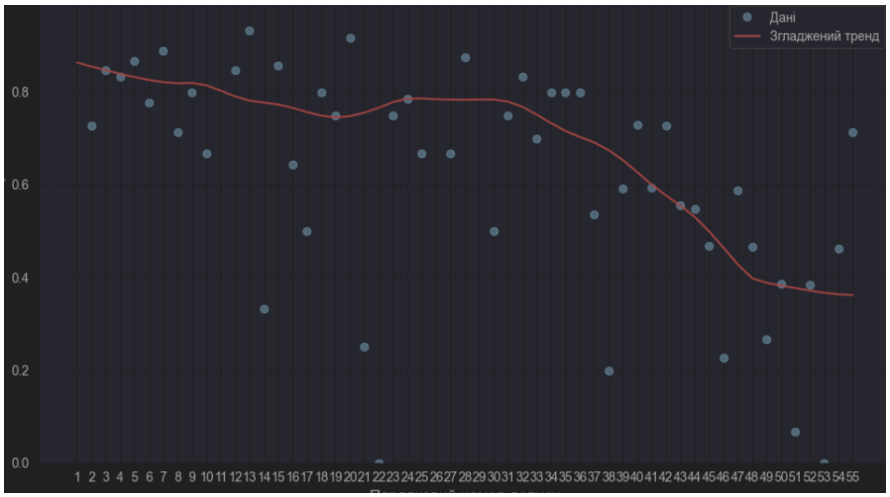


Figure 41: The general trend of change in the positivity of the sentiment of comments over time, where the blue dots are the data, the red line is the smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of the mood.

The entertainment category contains 1766 comments from 3 different pages (Fig. 42). In category discussions, most of the comments turned out to be neutral (Fig. 43a), even in the symbol_only category. Comments turned out to be significantly longer than those in the category of business pages (Fig. 43b). The solidarity index was, on average, the same as that of business pages, but with a different distribution by language (Fig. 44–45).

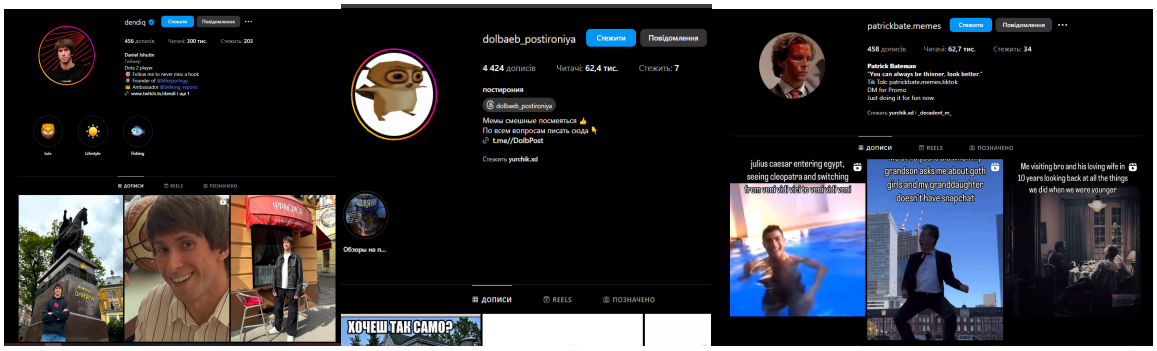


Figure 42: Example of an entertainment page.

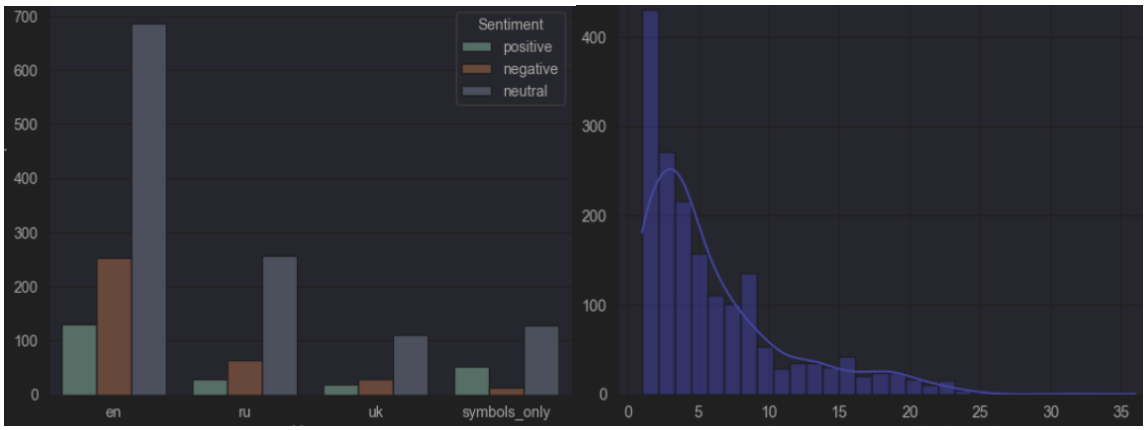


Figure 43: a) The tone of comments for each language, where X is the language, Y is the number of comments; b) the distribution of the length of comments after filtering, where X is the length of the comment (number of words), Y is the number of comments.

Figure 44: Average solidarity index for each language, where X is the language, and Y is the average percentage of matches.

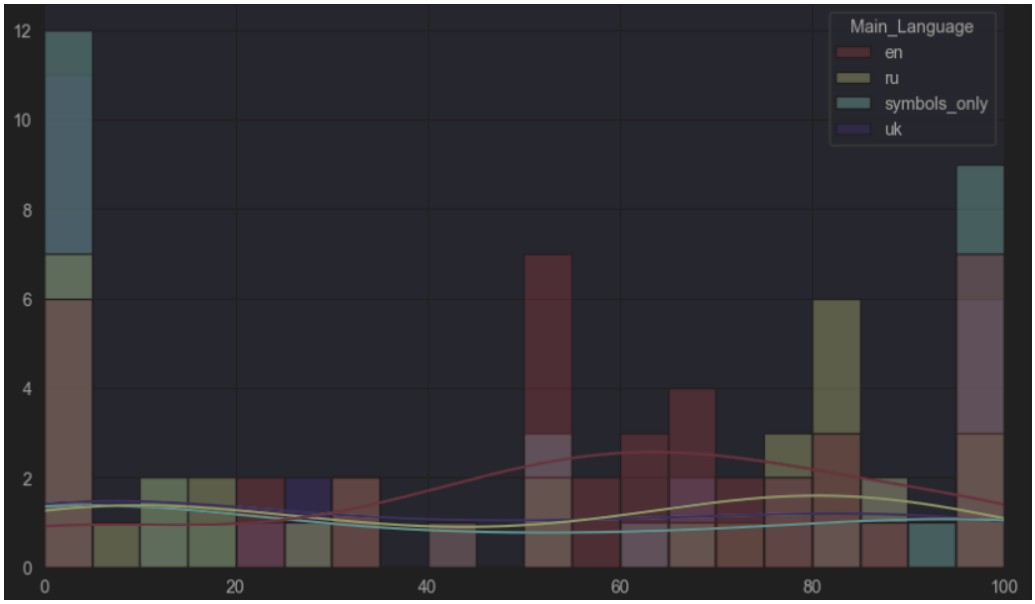


Figure 45: Distribution of the solidarity index for all languages, where X is the percentage of coincidences, and Y is the frequency.

Let's proceed to the analysis of moods (Figs. 46–50). You can see that pages from the entertainment category have a significantly lower number of positive comments compared to business accounts. However, a positive trend is also evident over time.

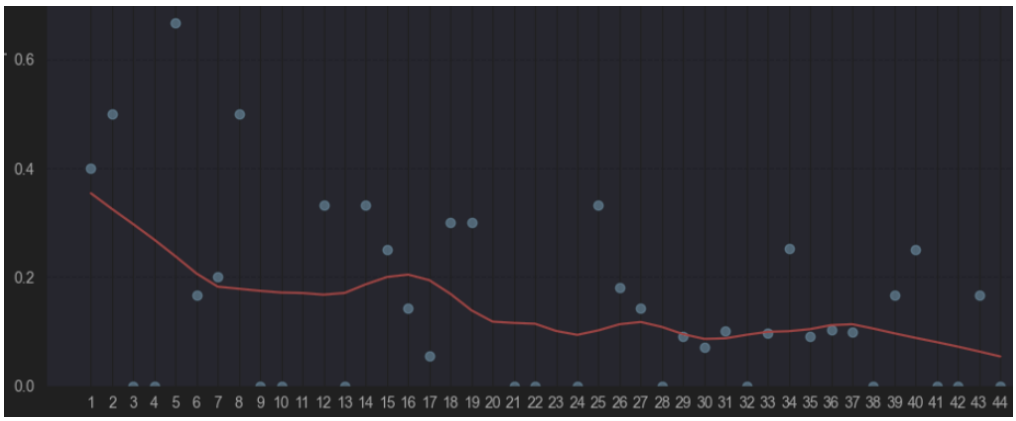


Figure 46: Trend of change in the positivity of the mood of comments over time for English, where blue dots are data, red line is a smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of the mood.

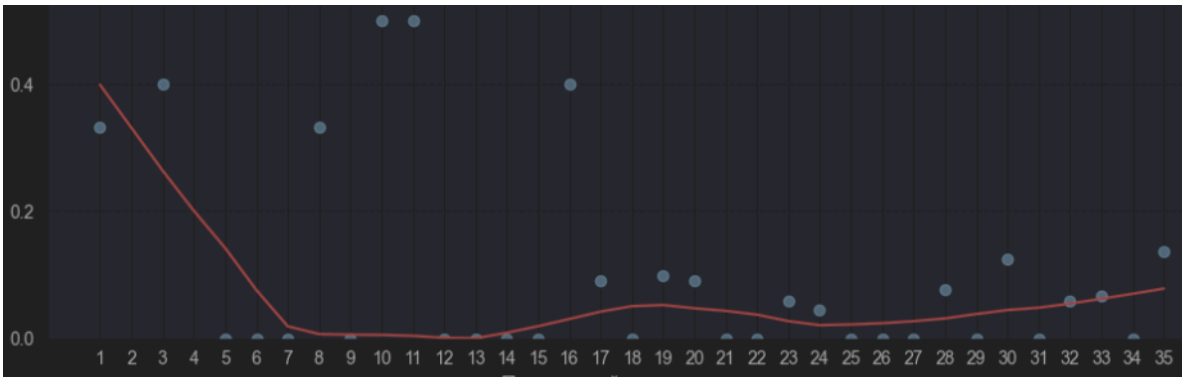


Figure 47: Trend of change in the positivity of the mood of comments over time for the Russian language, where blue dots are data, red line is a smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of the mood.

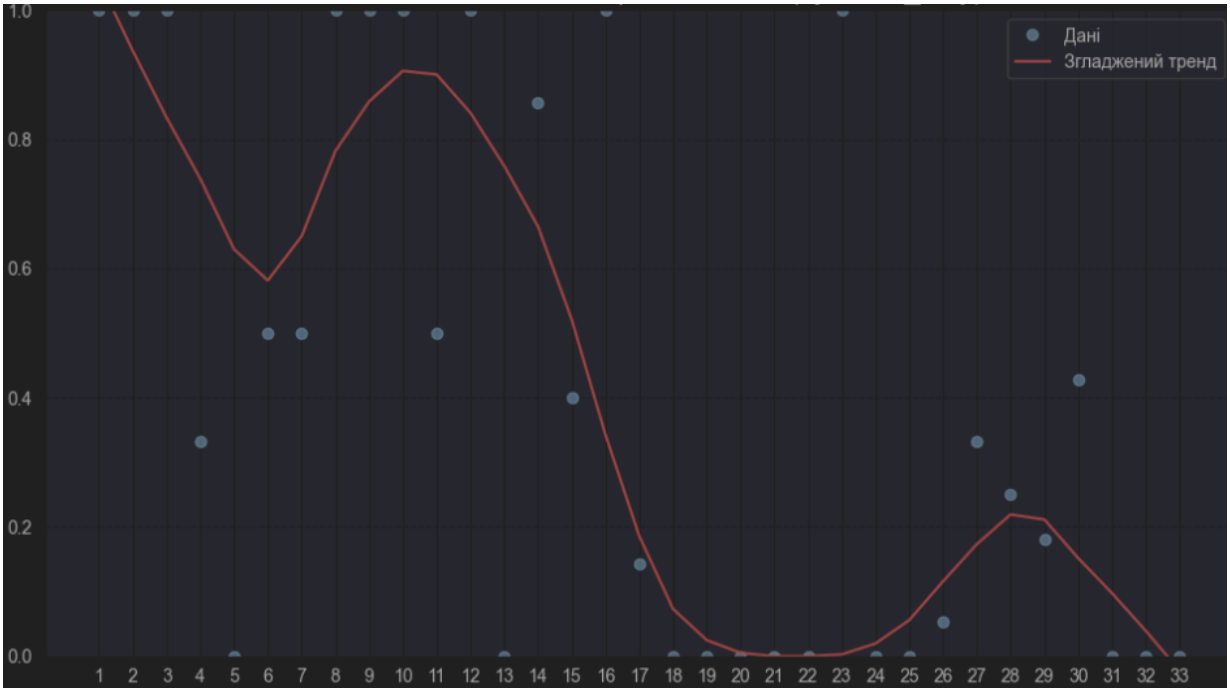


Figure 48: Comment sentiment positivity trend over time for characters, where blue dots are data, red line is a smoothed trend, where X is post id (timeline), and Y is sentiment positivity.

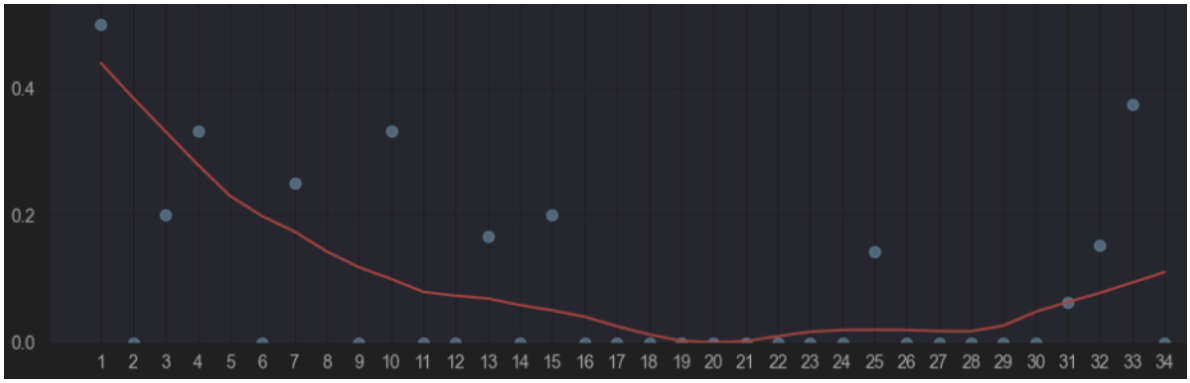


Figure 49: Trend of change in the positivity of comments' moods over time for the Ukrainian language, where blue dots represent the data, and the red line represents a smoothed trend. The X-axis represents the post ID (timescale), and the Y-axis represents the positivity of mood.

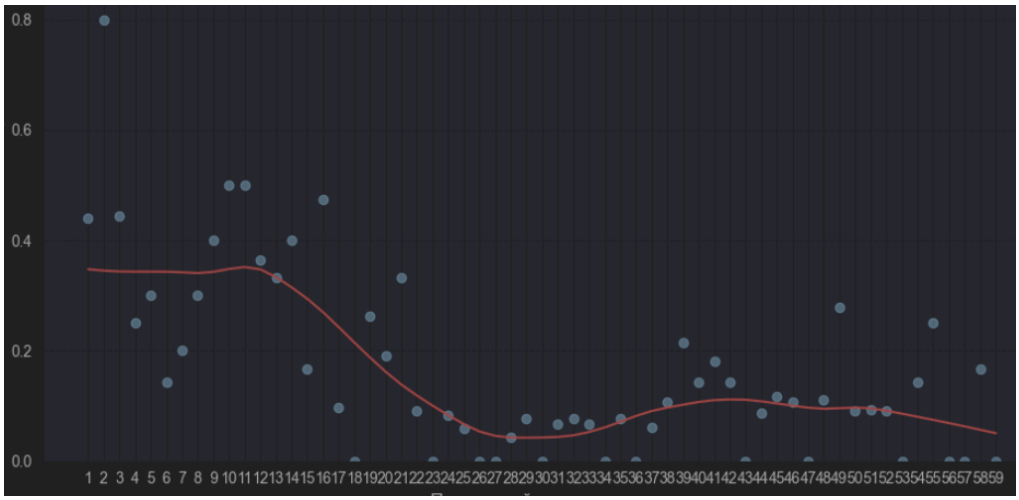


Figure 50: The general trend of change in the positivity of the sentiment of comments over time, where the blue dots are the data, the red line is the smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of the mood.

The last and largest analysed category is political accounts (Fig. 51). Three accounts, with a total of 17,000, were analysed. As in the category of business accounts, the largest categories of mood in languages are symbolic and English positive comments (Fig. 52a). It can be seen that in the sample of political commentaries there were significantly longer comments than in the previous categories (Fig. 52b). The solidarity index for political posts was considerably lower than for other categories (Fig. 53). The graph also shows that most of the solidarity comments are symbolic (Fig. 54).

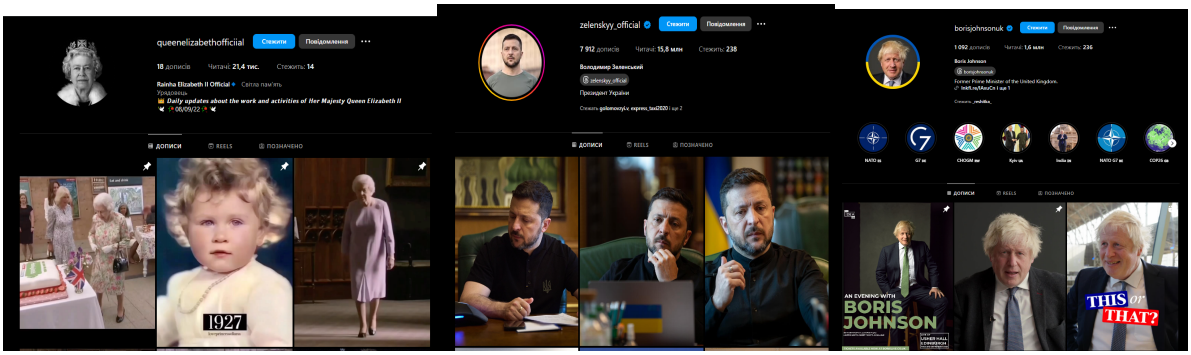


Figure 51: Example of a political news page.

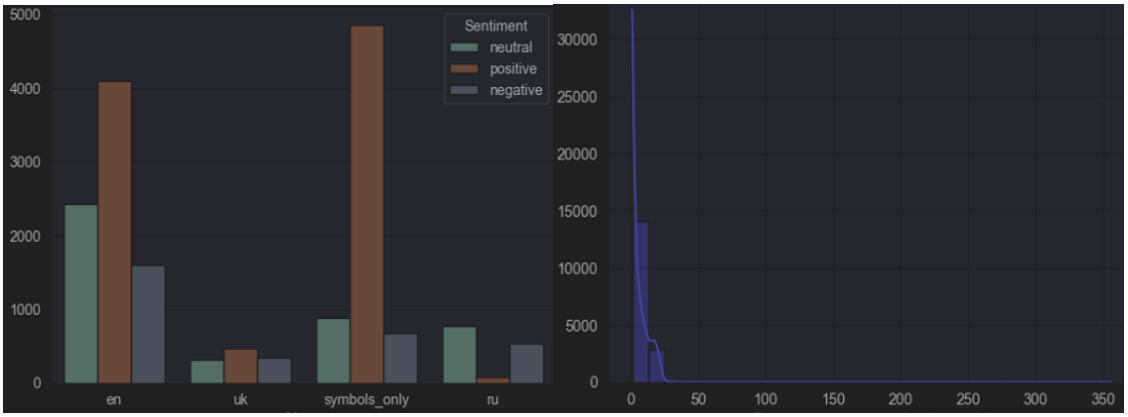


Figure 52: a) Sentiment of comments for each language, where X is the language, Y is the number of comments; b) the distribution of the length of comments after filtering, where X is the length of the comment (number of words), Y is the number of comments.

Figure 53: Average solidarity index for each language, where X is the language, and Y is the average percentage of matches.

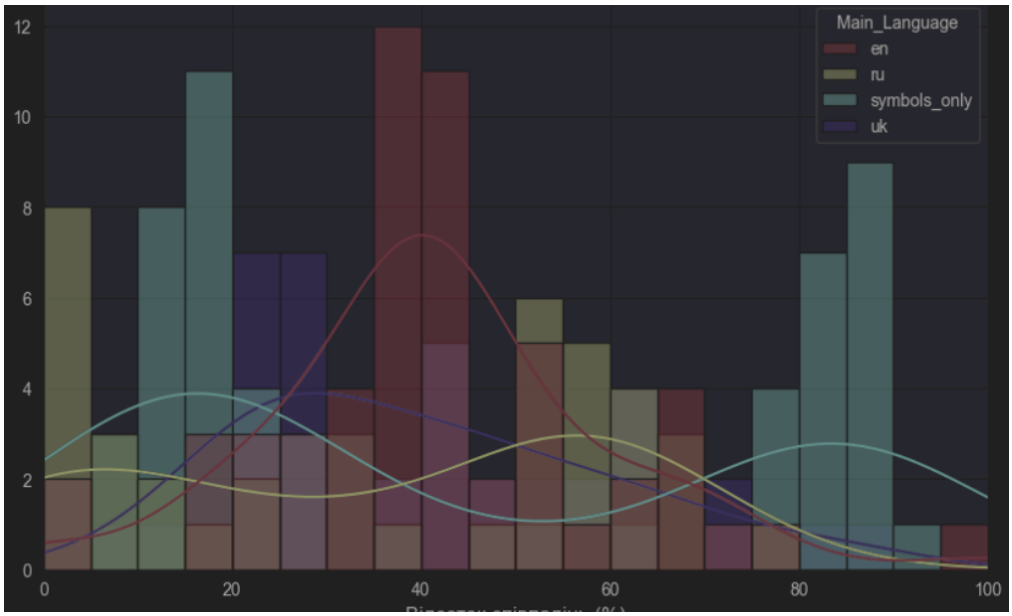


Figure 54: Distribution of the solidarity index for all languages, where X is the percentage of matches, and Y is the frequency.

Let's move on to sentiment analysis (Fig. 55–59). You can see that here, unlike other categories, there are clear peaks and troughs on the chart, indicating a high correlation between political events and the mood of comments. It is also worth noting that for comment schedule in Ukrainian and all languages, there is a strong trend towards improving the comments mood over time.

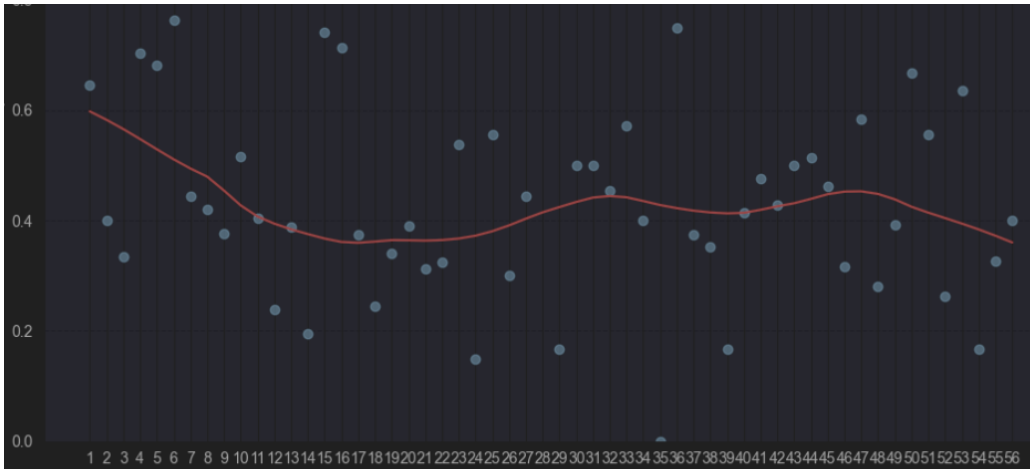


Figure 55: Trend of change in the positivity of the mood of comments over time for English, where blue dots are data, red line is a smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of the mood.

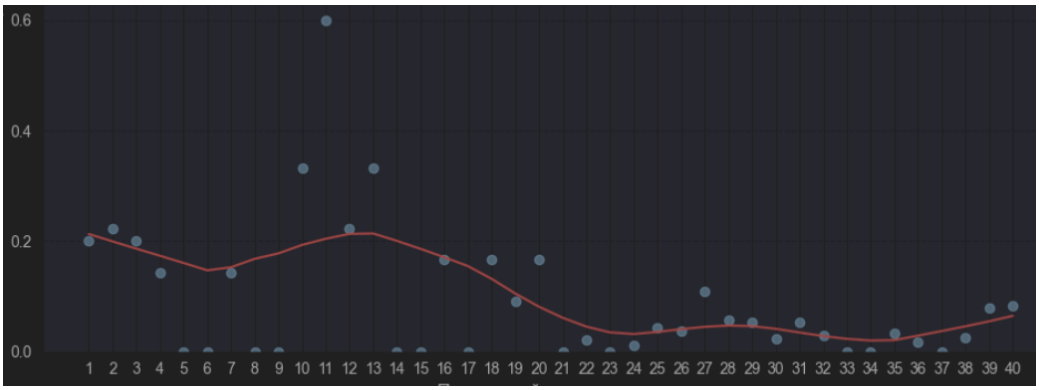


Figure 56: Trend of change in the positivity of the mood of comments over time for the Russian language, where blue dots are data, red line is a smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of the mood.

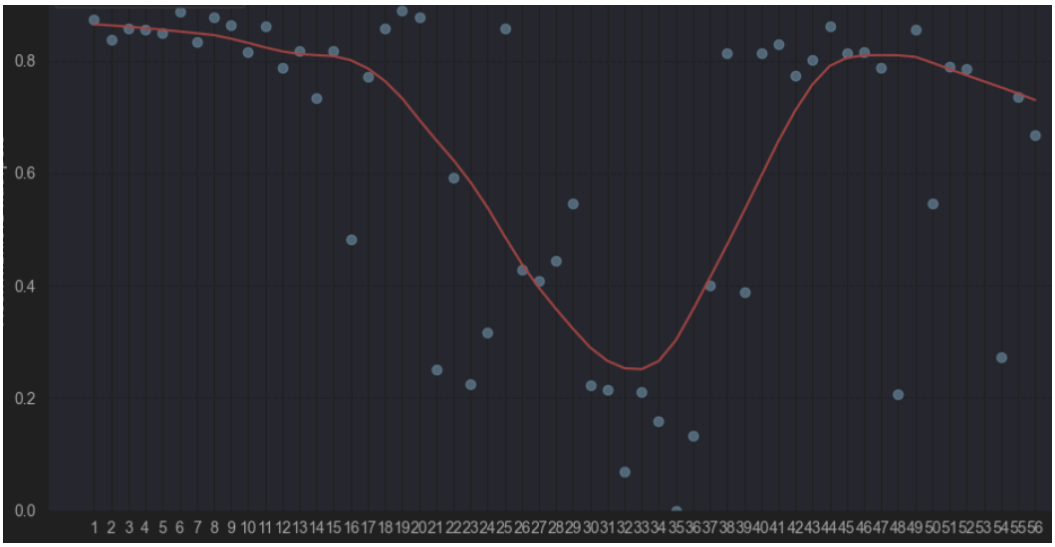


Figure 57: Comment sentiment positivity trend over time for characters, where blue dots are data, red line is a smoothed trend, where X is post ID (timeline), and Y is sentiment positivity.

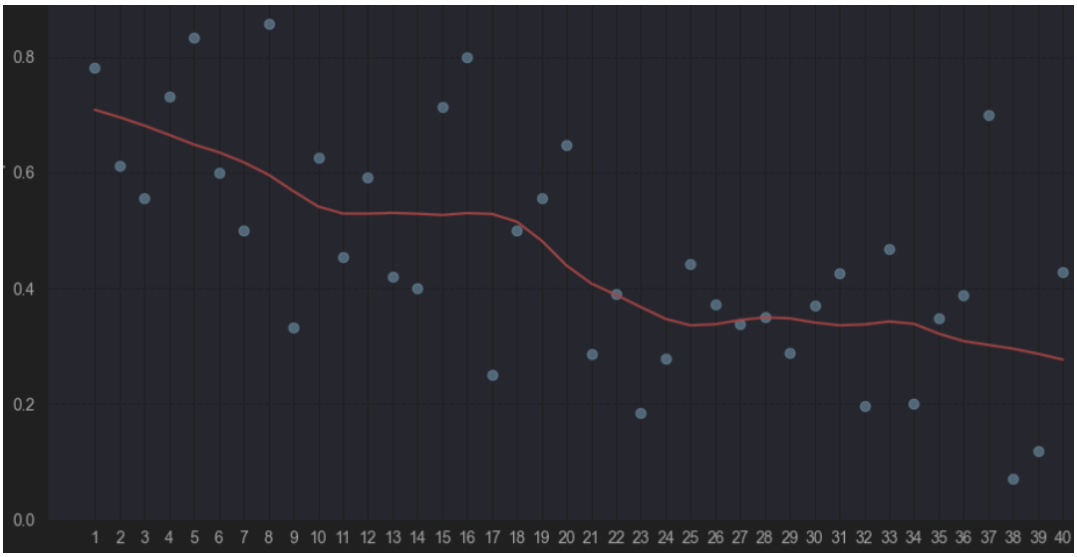


Figure 58: Trend of change in the positivity of the mood of comments over time for the Ukrainian language, where blue dots are data, red line is a smoothed trend, where X is the id of the post (timescale), and Y is the positivity of mood.

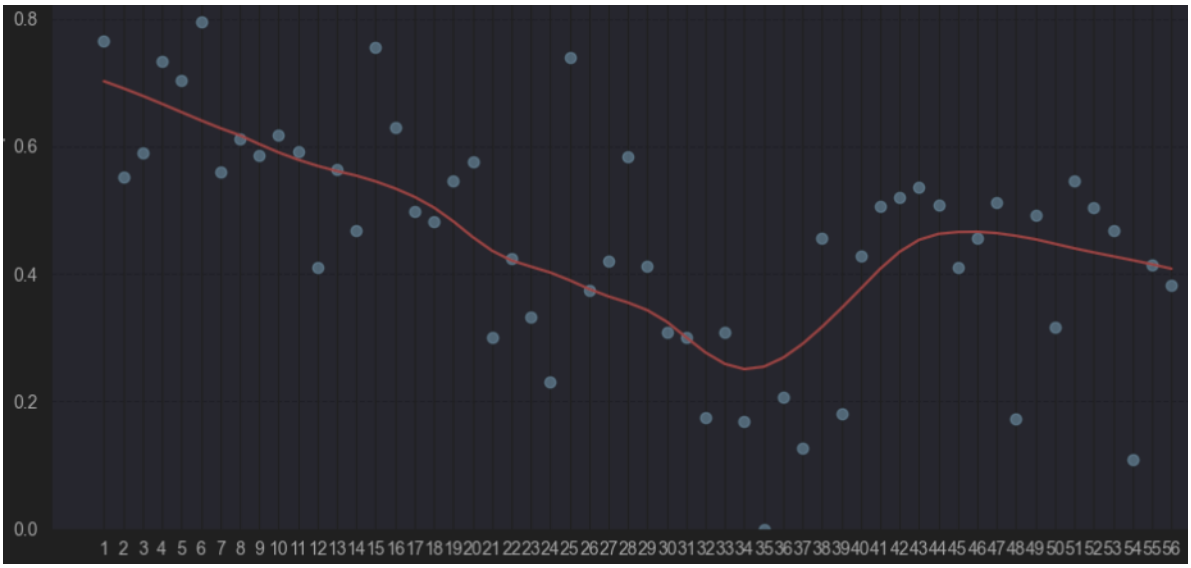


Figure 59: The general trend of change in the positivity of the sentiment of comments over time, where the blue dots are the data, the red line is the smoothed trend, where X is the ID of the post (timeline), and Y is the positivity of the mood.

9. Conclusions

As a result of the study, an integrated approach to the automated analysis of Instagram comments was developed, which considers the multilingual nature of content, the dynamism of social networks, and the characteristics of the informal online communication environment. The proposed system provides a comprehensive cycle of data processing, encompassing automatic parsing of comments, language detection, sentiment analysis, time trend analysis, and interactive visualisation of results. Based on a critical analysis of existing solutions, it was found that most commercial and open source tools demonstrate low accuracy for local languages, in particular Ukrainian, work with short and mixed texts to a limited extent, and do not take into account the specifics of social networks (emojis, slang, symbols). It confirmed the need to create a specialised system capable of adapting to the real-world conditions of data processing on social platforms.

As part of the study, a method for multilingual analysis of Instagram comments using specialised transformer models, specifically XLM-RoBERTa and individual models for specific language groups, was proposed for the first time. It enabled the achievement of high accuracy in determining moods for Ukrainian, Russian, and English, as well as in processing content consisting only of symbols or emojis. The developed algorithms for identifying the language group and assessing the solidarity of comments with posts provide a deeper contextual analysis of user reactions. An additional scientific result is a technique for analysing mood dynamics over time, which combines time series models, moving averages, and clustering. It enables you to identify changes in the audience's emotional response, predict trends, and assess long-term engagement with content. Visualising results using interactive tools enhances the practical value of the system for businesses, researchers, analysts, and organisations. The created system is relevant and significant for Ukraine, as it contributes to the development of local NLP solutions, supports the analysis of Ukrainian-language content, enables the monitoring of public sentiment, counters disinformation, and provides new opportunities for business intelligence.

Thus, the developed system demonstrates high efficiency in the multilingual analysis of texts from social networks, opening up prospects for further improvement, including expanding language support, integrating new transformer models, and enhancing the accuracy of analysing emotional and semantic characteristics of comments.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] V. Vysotska, M. Nazarkevych, S. Vladov, S. Chyrun, O. Lozynska, T. Lavrut, I. Budz, O. Muzychuk, O. Nagachevska, S. Diakun, Information technology for promoting Instagram accounts, in: Proceedings of the Computational Intelligence Application Workshop, CIAW '2024, CEUR Workshop Proceedings, Aachen, Germany, 2024, pp. 237–266.

[2] V. Vysotska, A. Starchenko, L. Chyrun, Z. Hu, Y. Ushenko, D. Uhryn, Sentiment analysing and visualising public opinion on political figures across YouTube and Twitter using NLP and machine learning, IJIGSP 17 (5) (2025) 117–164. doi:10.5815/ijigsp.2025.05.08.

[3] D. Uhryn, V. Vysotska, L. Chyrun, S. Chyrun, C. Hu, Y. Ushenko, Intelligent application for textual content authorship identification based on machine learning and sentiment analysis, IJISA 17 (2) (2025) 56–100. doi:10.5815/ijisa.2025.02.05.

[4] D. Holubinka, V. Vysotska, S. Vladov, Y. Ushenko, M. Talakh, Y. Tomka, Intelligent system for recognizing tone and categorizing text in media news at an electronic business based on sentiment and sarcasm analysis, IJIEEB 17 (1) (2025) 90–139. doi:10.5815/ijieeb.2025.01.06.

[5] M. N. Wa Nkongolo, News classification and categorization with smart function sentiment analysis, Int. J. Intell. Syst. 2023 (1) (2023). doi:10.1155/2023/1784394.

[6] M. Hasan, T. Ahmed, M. R. Islam, M. P. Uddin, Leveraging textual information for social media news categorization and sentiment analysis, PLOS ONE 19 (7) (2024). doi:10.1371/journal.pone.0307027.

[7] M. Shrivastava, S. Kumar, A pragmatic and intelligent model for sarcasm detection in social media text, Technol. Soc. 64 (2021). doi:10.1016/j.techsoc.2020.101489.

[8] U. Ahmed, J. C. W. Lin, G. Srivastava, Emotional intelligence attention unsupervised learning using lexicon analysis for irony-based advertising, TALLIP 23 (1) (2024) 1–19. doi:10.1145/3580496.

[9] A. M. Iddrisu, S. Mensah, F. Boafo, G. R. Yeluripati, P. Kudjo, A sentiment analysis framework to classify instances of sarcastic sentiments within the aviation sector, Int. J. Inf. Manag. Data Insights 3 (2) (2023). doi:10.1016/j.jjimei.2023.100180.

[10] A. D. Yacoub, S. Slim, A. Aboutabl, A survey of sentiment analysis and sarcasm detection: Challenges, techniques, and trends, IJECES 15 (1) (2024) 69–78. doi:10.32985/ijeces.15.1.7.

[11] J. Ahmed, M. Ahmed, Classification, detection and sentiment analysis using machine learning over next generation communication platforms, Microprocessors and Microsystems 98 (2023). doi:10.1016/j.micpro.2023.104795.

[12] C. I. Eke, A. A. Norman, L. Shuib, H. F. Nweke, Sarcasm identification in textual data: Systematic review, research challenges and open directions, Artif. Intell. Rev. 53 (6) (2020) 4215–4258. doi:10.1007/s10462-019-09791-8.

[13] A. Mansoori, K. Tahat, O. Al Zoubi, D. N. Tahat, M. Habes, H. Himdi, S. A. Salloum, Detection of sarcasm in news headlines using NLP and machine learning, in: A. Al-Marzouqi, S. Salloum, K. Shaalan, T. Gaber, R. Masa'deh (Eds.), Generative AI in Creative Industries, Springer, Cham, Switzerland, 2025, pp. 503–517. doi:10.1007/978-3-031-89175-5_31.

[14] D. K. Sharma, B. Singh, S. Agarwal, N. Pachauri, A. A. Alhussan, H. A. Abdallah, Sarcasm detection over social media platforms using hybrid ensemble model with fuzzy logic, Electronics 12 (4) (2023). doi:10.3390/electronics12040937.

[15] A. R. W. Sait, M. K. Ishak, Deep learning with natural language processing enabled sentimental analysis on sarcasm classification, Computer Systems Science and Engineering 44 (3) (2023) 2553–2567. doi:10.32604/csse.2023.029603.

[16] P. Tungthamthiti, K. Shirai, M. Mohd, Recognition of sarcasm in microblogging based on sentiment analysis and coherence identification, J. Nat. Lang. Process. 23 (5) (2016) 383–405. doi:10.5715/jnlp.23.383.

[17] M. N. Lahaji, T. R. Razak, M. H. Ismail, Unveiling sarcastic intent: Web-based detection of sarcasm in news headlines, JCRINN 8 (2) (2023) 215–225. doi:10.24191/jcrinn.v8i2.365.

[18] V. Vysotska, O. Markiv, S. Tchynetskyi, B. Polishchuk, O. Bratasyuk, V. Panasyuk, Sentiment analysis of information space as feedback of target audience for regional e-business support in Ukraine, in: Proceedings of the Modern Machine Learning Technologies and Data Science Workshop, MoMLeT&DS '2023, CEUR Workshop Proceedings, Aachen, Germany, 2023, pp. 488–513.

[19] D. K. Sharma, B. Singh, S. Agarwal, H. Kim, R. Sharma, Sarcasm detection over social media platforms using hybrid auto-encoder-based model, Electronics 11 (18) (2022). doi:10.3390/electronics11182844.

[20] A. Reyes, P. Rosso, D. Buscaldi, From humor recognition to irony detection: The figurative language of social media, Data Knowl. Eng. 74 (2012) 1–12. doi:10.1016/j.datak.2012.02.005.

[21] S. Voloshyn, V. Vysotska, O. Markiv, I. Dyyak, I. Budz, V. Schuchmann, Sentiment analysis technology of English newspapers quotes based on neural network as public opinion influences identification tool, in: Proceedings of the 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies, CSIT '2022, IEEE, New York, NY, 2022, pp. 83–88. doi:10.1109/CSIT56902.2022.10000627.

[22] V. Vysotska, S. Voloshyn, O. Markiv, O. Brodyak, N. Sokulska, V. Panasyuk, Tone analysis of regional articles in English-language newspapers based on recurrent neural network Bi-LSTM, in: Proceedings of the 2023 IEEE 5th International Conference on Advanced Information and Communication Technologies, AICT '2023, IEEE, New York, NY, 2023, pp. 1–6. doi:10.1109/AICT61584.2023.10452700.

[23] M. Zanchak, V. Vysotska, S. Albota, The sarcasm detection in news headlines based on machine learning technology, in: Proceedings of the 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies, CSIT '2021, IEEE, New York, NY, 2021, pp. 131–137. doi:10.1109/CSIT52700.2021.9648710.

[24] O. Tverdokhlib, V. Vysotska, O. Nagachevska, Y. Ushenko, D. Uhryn, Y. Tomka, Intelligent processing censoring inappropriate content in images, news, messages and articles on web pages based on machine learning, IJIGSP 17 (1) (2025) 107–164. doi:10.5815/ijigsp.2025.01.08.