# Improving sentiment analysis of Ukrainian-language content by applying a rule-based approach

Taras Basyuk*,† and Anton Lomovatskyi†

*Lviv Polytechnic National University, Bandera str.12, Lviv, 79013, Ukraine*

**Abstract**

Sentiment analysis is a fundamental component of natural language processing (NLP), enabling the automated assessment of textual sentiment across different languages. However, widely used sentiment analysis tools, such as VADER, often struggle with language-specific challenges, particularly in morphologically rich and syntactically complex languages like Ukrainian. This study introduces an improved rule-based sentiment analysis algorithm specifically designed for Ukrainian-language texts, addressing the limitations of generic approaches. The proposed algorithm integrates an enhanced lexicon, including the EMOLEX sentiment dictionary, polarity scores, emoji sentiment mapping, and intensity boosters, to refine sentiment classification. Additionally, advanced dependency parsing and position-aware scoring mechanisms are employed to improve contextual understanding, enabling more accurate differentiation between positive, negative, and neutral sentiments. These enhancements are particularly crucial for capturing Ukrainian-specific linguistic structures, which pose difficulties for existing sentiment analysis models. The algorithm's effectiveness was evaluated using Ukrainian-language datasets, comparing its performance against the widely used VADER sentiment analysis tool. The results demonstrate that the custom algorithm significantly outperforms VADER in detecting sentiment polarity, particularly in cases with strong positive or negative sentiment. This confirms the necessity of language-specific sentiment analysis tools for non-English content, as they provide greater accuracy and contextual sensitivity.

Despite the promising results, further improvements remain possible. One key area for future research involves integrating artificial intelligence (AI) techniques, such as machine learning and deep learning, to create a hybrid framework that enhances the accuracy of sentiment classification, especially for ambiguous or nuanced expressions.

**Keywords**

Sentiment analysis, Ukrainian language, rule-based algorithm, natural language processing, dependency parsing, lexicon expansion, sentiment classification

## 1. Introduction

Sentiment analysis is a pivotal function of natural language processing (NLP) that allows one to extract opinions and emotions from texts, as well as attitudes [1]. This is important for market research and social media monitoring, and it is also useful in analyzing customer feedback [2].

Millions of people speak Ukrainian all over the world; it is gaining more and more importance among digital communication, and even more enthusiastic audiences, such as social media, news sites, and customer reviews. However, the very high complexity of the natural language, with rich morphology and flexible syntax, a lot of negations, and idiomatic expressions, makes today existing algorithms for sentiment analysis helpless [3, 4]. Most of the popular sentiment analysis tools, such as VADER, are designed to understand English and do not cover the grammatical structures and lexical properties of Ukrainian, which brings down the accuracy rates of those tools when applied to Ukrainian texts [5].

Sentiment analysis has evolved from traditional rule-based approaches to modern deep learning techniques, enabling more accurate and context-aware classification of emotions in text [6]. While

*CLW-2025: Computational Linguistics Workshop at 9th International Conference on Computational Linguistics and Intelligent Systems (CoLInS-2025), May 15–16, 2025, Kharkiv, Ukraine*

* Corresponding author.

† These authors contributed equally.

Taras.M.Basyuk@lpnu.ua (T. Basyuk); Anton.A.Lomovatskyi@lpnu.ua (A. Lomovatskyi)

0000-0003-0813-0785 (T. Basyuk); 0009-0004-5170-3272 (A. Lomovatskyi)

early methods, such as lexicon-based sentiment scoring, were effective for structured languages, they often fail to capture complex syntactic dependencies in morphologically rich languages like Ukrainian. Recent advancements in natural language processing (NLP) have introduced powerful transformer-based models, such as Ukr-RoBERTa and Multilingual BERT (mBERT), which significantly improve sentiment classification by leveraging deep contextual embeddings [7]. These models are trained on vast multilingual datasets and can generalize well across different languages, including Ukrainian. However, challenges remain, particularly in handling domain-specific sentiment expressions, idiomatic phrases, and negation structures. The following analysis examines key research contributions and existing sentiment analysis tools, emphasizing their methodologies, limitations, and applicability to Ukrainian-language content [8].

Despite these advancements, sentiment analysis for the Ukrainian language remains underdeveloped, with many widely used tools being optimized primarily for English. Existing approaches struggle to fully adapt to the rich morphology, flexible word order, and unique sentiment expressions present in Ukrainian texts [9]. This highlights the need for more specialized research and the development of tailored sentiment analysis models that can accurately interpret sentiment in Ukrainian-language content. Addressing these challenges will not only improve sentiment classification accuracy but also enhance the applicability of NLP techniques for Ukrainian in fields such as social media monitoring, customer feedback analysis, and political discourse evaluation [10].

The current research aims to develop an enhanced rule-based sentiment analysis algorithm specifically tailored for Ukrainian-language content [11]. To address the challenges highlighted in previous studies and overcome the limitations of existing sentiment analysis tools, this work focuses on the following key tasks.

*Expansion of Ukrainian sentiment lexicons.* Integrating a comprehensive sentiment lexicon that accounts for Ukrainian-specific linguistic features, including domain-specific vocabulary, slang, and idiomatic expressions. Incorporating emoji sentiment mappings to improve the classification of informal and digital communication, which is essential for social media and online content analysis.

*Integration of dependency-based syntax parsing.* Utilizing syntactic dependency parsing to capture contextual relationships between words, allowing for more accurate sentiment classification in a morphologically rich and flexible word-order language like Ukrainian [12].

*Refinement of sentiment intensity modifiers.* Developing an improved approach to handling intensity modifiers (e.g., “дуже” – very, “майже” – almost, “надзвичайно” – extremely) to ensure correct sentiment scaling. Enhancing positional effect handling, ensuring that the placement of sentiment-bearing words in a sentence (e.g., at the beginning or end) influences classification outcomes appropriately.

*Comparison with existing sentiment analysis models.* Conducting a quantitative and qualitative comparison of the proposed rule-based model with VADER, a widely used sentiment analysis tool optimized for English [13]. Benchmarking against pre-existing methodologies to demonstrate the strengths and weaknesses of a rule-based approach compared to statistical and deep learning methods for sentiment analysis in non-English languages.

By fulfilling these research objectives, this study aims to enhance sentiment classification accuracy for Ukrainian-language content, contributing to the development of specialized NLP tools for underrepresented languages [14]. The proposed improvements provide a foundation for hybrid sentiment analysis models that can combine rule-based and machine-learning approaches in future studies.

## 2. Related Works

Recent studies have explored hybrid techniques that combine rule-based processing with machine learning to address linguistic nuances and improve classification performance. The following analysis reviews key research contributions, highlighting their methodologies, strengths, and limitations in advancing sentiment analysis:

- "Mining and Summarizing Customer Reviews" [15] authors M. Hu, B. Liu introduce a novel approach for extracting sentiment information from customer reviews. Their method focuses on summarizing reviews by identifying product features and categorizing opinions as positive or negative. The study presents an unsupervised learning model that extracts sentiment-oriented phrases and organizes them into structured summaries. This work laid the foundation for many modern sentiment analysis systems by emphasizing feature-based sentiment extraction.
- In "A Joint Model of Text and Aspect Ratings for Sentiment Summarization" [16] I. Titov, R. McDonald propose a probabilistic model for sentiment summarization, integrating textual reviews with numerical aspect ratings. Their joint modeling approach allows the system to generate more accurate and aspect-specific sentiment summaries. The paper demonstrates how this method improves the interpretability of sentiment classification, making it particularly useful for product review analysis.
- In the work "A Real-time Hand Gesture Recognition System for Human-Computer and Human-Robot Interaction" [17], the proposed gesture recognition system is designed to improve human-computer interaction and human-robot interaction. As the authors of the study assure, such interaction ensures natural and intuitive communication between people and technology using gestures.
- Authors of "Determining the Sentiment of Opinions" [17] explore the challenges of determining the sentiment of user opinions by developing a method that distinguishes between subjectivity and sentiment polarity. Their approach combines machine learning techniques with rule-based linguistic analysis to improve classification accuracy. A key contribution of this work is its focus on contextual sentiment detection, which enhances its applicability to opinion mining.
- In "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews" [18] P. D. Turney introduces an unsupervised method for sentiment classification using semantic orientation. The approach leverages pointwise mutual information (PMI) to measure the association between words and their sentiment polarity. This study is notable for its effectiveness in review classification without requiring labeled training data, making it a significant milestone in early sentiment analysis research.
- In "Thumbs Up? Sentiment Classification Using Machine Learning Techniques" [19] B. Pang, L. Lee, S. Vaithyanathan present one of the first applications of machine learning for sentiment classification. The study compares Naïve Bayes, maximum entropy, and support vector machines (SVM) for sentiment polarity classification on movie reviews. Their findings demonstrate that SVM outperforms other classifiers, establishing it as a dominant technique in early sentiment analysis research.
- Authors of "Learning Extraction Patterns for Subjective Expressions" [20] propose an approach to extract subjective expressions from text. Their method relies on pattern-based learning techniques to identify phrases expressing sentiment. This work is critical in advancing fine-grained sentiment analysis, particularly for detecting implicit opinions that may not contain explicit sentiment words.
- In "Peculiarities of an Information System Development for Studying Ukrainian Language and Carrying out an Emotional and Content Analysis" [21] authors present a study on the development of an information system designed for the analysis of Ukrainian-language content. Their research focuses on integrating emotional and content-based sentiment analysis techniques, addressing the unique linguistic challenges posed by the Ukrainian language.

Building on the challenges identified in previous research, this study continues the development of sentiment analysis tools specifically designed for Ukrainian-language content. Existing sentiment analysis models, including multilingual transformer-based approaches, still struggle with the

morphological complexity, rich syntax, and unique contextual dependencies of Ukrainian [22]. To address these issues, this research proposes an enhanced rule-based sentiment analysis algorithm that leverages expanded lexicons, dependency parsing, and refined rule-based logic to achieve more precise sentiment classification [23]. This will deal with linguistic and contextual problems that Ukrainian is subjected to but is rarely encountered in current frameworks. Rule-based systems offer interpretable and transparent decision-making processes compared to other black-box methods [24].

This research is based on the preliminary work presented in "Naive Rule-Based Method in Sentiment Analysis of Ukrainian Language Content," where a simple rule-based algorithm was introduced to conduct sentiment analysis on Ukrainian text [25]. The baseline method used predefined positive and negative lexicons along with a few grammatical rules for handling negation and modifiers [26]. The study underlined several critical limitations:

- Contextual Insensitivity. The naive method was insensitive to grammatical and syntactic relationships existing between words, hence cutting the accuracy when facing texts comprising complex sentence structures.
- Negation Handling. The basic rules of negations were considered, but they did not capture subtle interactions between negations and word intensities.
- Lexicon Coverage. The small lexicon used resulted in low recall for texts containing slang, idiomatic expressions, or domain-specific terms [27].

## 3. Methods and Materials

In order to present the main aspects of the studied subject area, a scheme was finalized that reflects the main stages that must be implemented in the sentiment analysis system (Fig. 1).

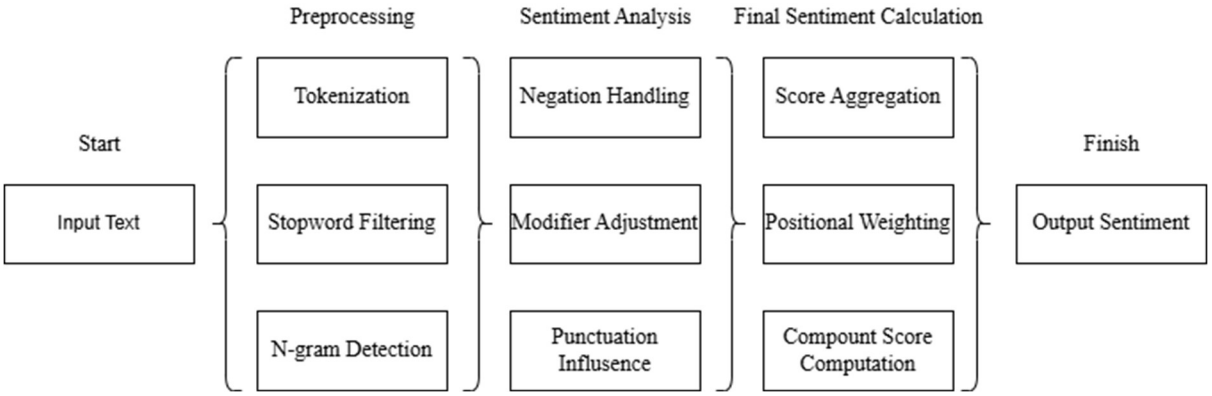


**Figure 1:** The sequence of stages in sentiment analysis

As it is displayed on the picture above the analysis process consists of 3 main parts:

- Preprocessing – removing extra information from the data to analyze.
- Sentiment analysis – main process of evaluating each token.
- Final sentiment calculation – summing up the result of a sentiment analysis and compounding the score into one result.

### 3.1. Description of the New Algorithm: Lexicon

The rule-based sentiment analysis algorithm proposed here will be using the rich lexicon, which is tuned for Ukrainian-language content. These take into account the linguistic and emotional subtleties of sentiment classification [28].

It is using the extended Ukrainian version of the EMOLEX lexicon. It is a great source whereby the set of Ukrainian words carries with it sentimental labels classifying the expressions in categories

of positive, negative, and neutral sentiments, along with categories of joy, anger, trust, and fear. Such a label would allow the algorithm to identify the emotional intensity accurately [29, 30].

Additionally, a supplementary polarity lexicon extends EMOLEX with sentiment scores for less common and domain-specific terms. Words are scored on their polarity, from highly negative to highly positive, allowing a finer granularity and increased coverage for the algorithm [31].

The inclusion of an expanded emoji sentiment mapping allows the algorithm to process informal communication, such as social media texts. Sentiment scores are assigned to commonly used emojis, categorizing them as positive (e.g., "😂", "💔"), negative (e.g., "😢", "😡"), or neutral (e.g., "🙂"). This enhances the algorithm's ability to classify modern digital texts accurately.

The algorithm relies on a sophisticated set of intensity booster words that raise or lower the emotional impact of surrounding words. For example, words like "дуже" ("very") or "абсолютно" ("absolutely") increase the intensity of positive or negative sentiment, while modifiers like "трохи" ("slightly") reduce it [32].

A large phrase sentiment lexicon was used to score multi-word expressions and idiomatic phrases. This resource ensures that the algorithm can capture the sentiment of complex phrases, such as "на межі розпачу" ("on the verge of despair"), which would otherwise be lost in word-by-word analysis.

It then filters out stopwords, or words that occur frequently and do not contribute to the sentiment, such as "і" ("and") or "або" ("or"). It uses a hand-curated list of Ukrainian stopwords so that only meaningful words are looked at for sentiment, thereby improving both accuracy and efficiency.

This combination of resources provides a strong foundation for the algorithm in handling the variety of vocabulary, expressions, and informal modes of communication present in content written in the Ukrainian language.

### 3.2. Description of the New Algorithm: Text Preprocessing

Preprocessing of text accurately is one main step in the rule-based sentiment analysis algorithm proposed herein [33]. This would make sure that input texts are transformed into a structured format for analysis while preserving the nuances of language and context.

The first step in text processing is tokenization, where the input text is divided into smaller units called tokens. Tokens can include words, punctuation marks, and emojis. This process is designed to handle Ukrainian-language content effectively by:

- preservation of the structure of words with rich morphology;
- retention of punctuation marks such as "!" and ".", which later will be analyzed for their influence on sentiment;
- isolating and detecting emojis, since they are treated as independent sentiment-bearing units.

For example, the sentence: "Це було неймовірно красиво, але трохи сумно 😢!" is tokenized into: ["Це", "було", "неймовірно", "красиво", ",", "але", "трохи", "сумно", "😢", "!"].

It captures contextual sentiment and idiomatic expressions by using N-gram analysis. N-grams refer to sequences of N consecutive tokens, which are important for identifying multi-word expressions and phrases that carry sentiment. The algorithm processes unigrams or single tokens for analyzing individual words such as "красиво" ("beautiful"); bigrams or two-word phrases, which capture information in context, for example, "трохи сумно" ("slightly sad"); trigrams or three-word phrases to identify more complex expressions, such as "на межі розпачу" ("on the verge of despair") [34]. N-grams enable the algorithm to bring in phrase-level sentiment, enhancing its capability to deal with subtle language constructs that may not be captured using a purely word-based approach [35].

The integration of tokenization and N-gram analysis ensures that the algorithm captures both the sentiment of individual words and the contextual meaning of phrases, hence making the classification more accurate.

### 3.3. Description of the New Algorithm: Sentiment Analysis

To solve the linguistic complexity of Ukrainian, the suggested algorithm of sentiment analysis will include the component of dependency analysis. Such a module will identify grammatical relations between words, which will enable the algorithm to consider context and interactions within a sentence [36]. Dependency analysis enhances sentiment classification by handling key linguistic phenomena: negations, modifiers, and punctuation.

Negation plays an important role in sentiment polarity. The algorithm recognizes negation words like "не" or "ні" and adjusts the sentiment of the words associated with them.

Example:

- Input. "Це не гарно" ("This is not beautiful").
- Without negation handling. Positive due to "гарно" ("beautiful").
- With negation handling. Negative due to "не" ("not").

The algorithm uses syntactic dependencies to link negations to their target words, ensuring accurate sentiment reversal.

Let:

- $S(w)$ – sentiment score of a word $w_t$.
- $w_{neg}$ negation word (e.g., "не", "ні").
- $dep(w_{neg}, w_t)$ – syntactic dependency linking $w_{neg}$ to its target word $w_t$.
- $S'(w_t)$ – adjusted sentiment score of the target word $w_t$.

The adjusted sentiment score is calculated as (Eq. 1):

$$S'(w_t) = -k \cdot S(w_t) \tag{1}$$

Where:

- $k$ – amplification factor (e.g., $k$ =1.5) to increase the effect of negation.

Modifiers are words such as intensity boosters: "дуже" ("very"), "абсолютно" ("absolutely") or reducers -"трохи" ("slightly") which increase or reduce the emotional weight of words. Example:

- Input. "Це дуже гарно" ("This is very beautiful").
- Sentiment score for "гарно" is increased due to the booster "дуже".

By leveraging dependency relationships, the algorithm ensures that modifiers are correctly associated with their target words.

Let:

- $S(w)$ – sentiment score of a word $w$.
- $w_{mod}$ – modifier word (e.g., "дуже" - "very", "абсолютно" - "absolutely", "трохи" - "slightly").
- $dep(w_{mod}, w_t)$ – syntactic dependency linking $w_{mod}$ to its target word $w_t$.
- $S'(w_t)$ – adjusted sentiment score of the target word $w_t$.
- $\lambda(w_{mod})$ – modifier weight, where $\lambda > 1$ for boosters (e.g., "дуже"), and $0 < \lambda < 1$ for reducers (e.g., "трохи").

The adjusted sentiment score is calculated as (Eq. 2):

$$S'(w_t) = \lambda(w_{mod}) \cdot S(w_t) \quad (2)$$

Where:

- $\lambda(w_{mod})$ – the weight depends on the intensity or reducing effect of the modifier.

For example: $\lambda$ ("дуже") = 1.5 (boosts sentiment), $\lambda$ ("трохи") = 0.8 (reduces sentiment). Punctuation marks, such as exclamation points ("!") and ellipses ("..."), often convey additional emotional context [37]. The algorithm adjusts sentiment scores based on the presence and type of punctuation:

- *Exclamation marks.* Amplify sentiment intensity. Example: "Це чудово!" ("This is wonderful!") has a higher sentiment score due to the exclamation mark.
- *Reduce sentiment intensity,* indicating hesitation or uncertainty. Example: "Це цікаво..." ("This is interesting...") has a lower sentiment score due to the ellipsis.

Let:

- $S(w)$ – sentiment score of a word $w$.
- $P$ – punctuation mark associated with the sentence (e.g., "!" or "...").
- $S'(w)$ – adjusted sentiment score of the word www.
- $\gamma(P)$ – *punctuation multiplier*, where: $\gamma("!") > 1$ (amplifies sentiment intensity), $0 < \gamma("...") < 1$ (reduces sentiment intensity).

The adjusted sentiment score is calculated as (Eq. 3):

$$S'(w) = \gamma(P) * S(w) \quad (3)$$

Where:

- $\gamma(!) = 1.5$ (example amplification factor for exclamation marks).
- $\gamma("...") = 0.8$ (example reduction factor for ellipses).

Using a dependency parser, the algorithm builds a tree structure for each sentence, identifying grammatical relationships between words. For example:

- Input. "Це не дуже гарно!" ("This is not very beautiful!")
- Dependency tree:
  - "не" $\rightarrow$ modifies $\rightarrow$ "гарно".
  - "дуже" $\rightarrow$ modifies $\rightarrow$ "гарно".
  - "!" $\rightarrow$ modifies $\rightarrow$ overall sentiment intensity.

The algorithm processes these relationships to adjust sentiment scores dynamically, improving accuracy in handling complex sentences. By incorporating dependency analysis, the algorithm captures much subtler linguistic interactions that usually elude simpler, rule-based systems. This yields much more detailed and accurate sentiment classification [38].

### 3.4. Description of the New Algorithm: Final Sentiment Calculation

The scoring of sentiment is further enhanced by including multipliers and positional weights to enable even finer levels of sentiment classification by modifying the intensity of emotional words and phrases. The approach ensures that contextually important words and positions of sentences are weighted appropriately [39].

The algorithm explicitly includes boosters-words that increase or decrease the intensity of sentiment-into the scoring mechanism. The algorithm assigns predefined weights to the boosters, based on their strength and direction. As for amplifiers, the words like "дуже" ("very") or "абсолютно" ("absolutely") increase the intensity of the associated sentiment. For example, "Це дуже гарно" ("This is very beautiful") where a sentiment score for "гарно" is multiplied by a factor of 1.5 due to the booster "дуже". Reducers have words like "трохи" ("slightly") or "майже" ("almost") which reduce the intensity. For instance, "Це трохи сумно" ("This is slightly sad") where a sentiment score for "сумно" is multiplied by a factor of 0.7.

The algorithm identifies those words through the dependency parsing applied with the proper multipliers of the corresponding sentiments to dynamically adapt the sentiment score.

The placement of a word or a phrase in the sentence can indeed have a significant impact on the overall sentence. To overcome this, an algorithm assigns a positional weight:

- *Beginning of the sentence.* Words at the start of a sentence often set the tone and are assigned higher weights. For instance, "Чудово, але трохи складно" ("Wonderful, but slightly difficult"), where "Чудово" ("Wonderful") receives a higher weight, emphasizing its influence.
- *End of the sentence.* Words at the end of a sentence often leave a lasting impression and are given slightly higher weights than words in the middle. Example is "Це було добре, але складно" ("It was good, but difficult"), where "Складно" ("Difficult") receives a higher weight due to its sentence-ending position.

Let:

- $S(w)$ – sentiment score of a word $w$.
- $W_{pos}(w)$ – positional weight assigned to the word www, based on its position in the sentence.
- $W_{pos}(w_{start}) > W_{pos}(w_{end}) > W_{pos}(w_{middle})$ – higher weights are assigned to the start and end positions).
- $S'(w)$ – Adjusted sentiment score of the word $w$.

The adjusted sentiment score is calculated as (Eq. 4):

$$S'(w) = W_{pos}(w) * S(w) \tag{4}$$

Where:

- $W_{pos}(w_{start}) = 1.5$ (example weight for words at the start of the sentence).
- $W_{pos}(w_{end}) = 1.2$ (example weight for words at the end of the sentence).
- $W_{pos}(w_{middle}) = 1.0$ (default weight for words in the middle of the sentence).

By having an extended version of lexicon dictionaries, preprocessing tools, sentiment analysis algorithms and composing results utilities the custom sentiment analysis system for Ukrainian language content can be built.

## 4. Experiment

VADER (Valence Aware Dictionary and Sentiment Reasoner) is as a baseline for evaluating the effectiveness of the rule-based sentiment analysis algorithm. This was considered a good benchmark because of its rule-based approach, considering punctuation, emojis, and intensity modifiers. Nevertheless, VADER is mostly optimized for English-language content, and its direct use for Ukrainian texts is challenging [40].

## 4.1. Selection of VADER as the Baseline

The choice of VADER as a baseline for comparison in this study was influenced by several key factors. Firstly, VADER is widely recognized and utilized in sentiment analysis research and industry applications, particularly for analyzing short texts in social media contexts. Its popularity stems from its ability to efficiently process sentiment in informal and digital communication.

Secondly, VADER offers a transparent and interpretable rule-based approach, similar to the proposed algorithm. By relying on predefined lexicons and scoring mechanisms, it allows for a direct comparison of methodologies without the opacity often associated with deep learning models.

Additionally, VADER incorporates various non-lexical features, such as punctuation handling, capitalization detection, and intensity modifiers. These features align with the enhancements introduced in the custom rule-based algorithm, making it a suitable benchmark for evaluating sentiment classification techniques.

However, VADER has significant limitations when applied to the Ukrainian language. It lacks Ukrainian-specific lexicons, dependency parsing, and linguistic resources necessary for accurate sentiment interpretation. As a result, its application to Ukrainian texts requires modifications or adaptations to achieve reliable results, highlighting the need for language-specific sentiment analysis models.

## 4.2. Testing Methodology

Both algorithms are tested with the same dataset of Ukrainian-language texts. The dataset contains pre-labeled samples in three categories: positive, neutral, and negative. It used the same inputs to test both algorithms on identical inputs as a way of checking whose performance will excel.

The steps in the methodology are:

1. *Preprocessing.* The texts were cleaned to remove noise, such as extra spaces and special characters. In the case of VADER, texts were translated into English using Python's translate library. This was necessary because VADER works exclusively with English texts. The custom algorithm processed the original Ukrainian texts without translation.
2. *Sentiment classification.* VADER and the custom algorithm independently classified each text into positive, neutral, or negative categories and assigned a compound score representing sentiment intensity.
3. *Evaluation metrics.*
   - Accuracy – the proportion of correctly classified texts.
   - F1-score – a harmonic mean of precision and recall for each sentiment category.
   - Mean compound score – the average compound score for each sentiment category to align with human annotation.
4. *Error analysis.* Several misclassifications were analyzed to understand the pattern and edge cases where one algorithm performed better than the other.
5. *Visualization.* Comparative sentiment distribution (positive, neutral, negative) through bar charts and mean compound scores for both algorithms in statistical summaries.

A context diagram of the design system (Fig. 2) was built to further explain the process behind comparison of two algorithms.

In the specified model, the input receives raw Ukrainian text data, which serves as the basis for sentiment analysis. This data can originate from various sources, including user-generated content, social media posts, or datasets prepared for research purposes. The output of the system is the Sentiment classification, which reflects whether the analyzed text conveys positive, negative, or neutral sentiment.

The system is influenced by several control components:

- Lexicons and rules. These provide the semantic and syntactic frameworks needed for accurate sentiment detection. Lexicons include emotional word dictionaries, sentiment polarity scores, and rules that define language-specific sentiment cues.

- Preprocessing methods. This refers to the set of techniques used to prepare raw text for analysis, including tokenization, stopword removal, and normalization processes.

- Computational environment. This includes the hardware and software infrastructure that supports the processing and analysis of data, ensuring system performance and scalability.

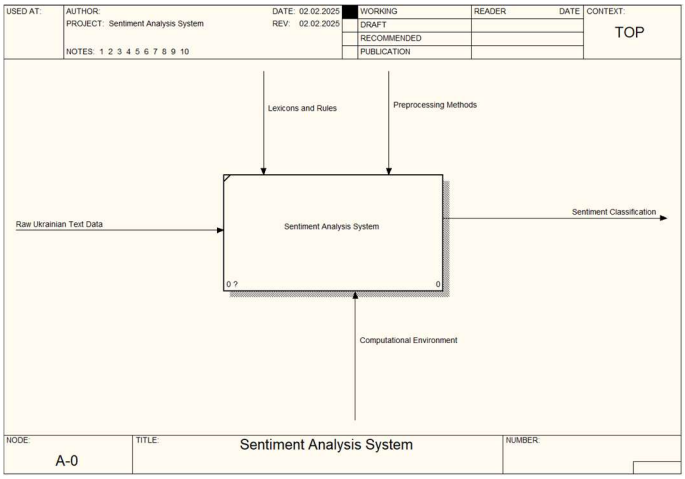


**Figure 2:** Context diagram of the designed system

The Sentiment analysis system operates by integrating these inputs and controls to generate reliable sentiment classifications based on predefined rules and linguistic resources. To gain a deeper understanding of the sentiment analysis workflow, the context diagram was decomposed into several sub-processes (Fig. 3).

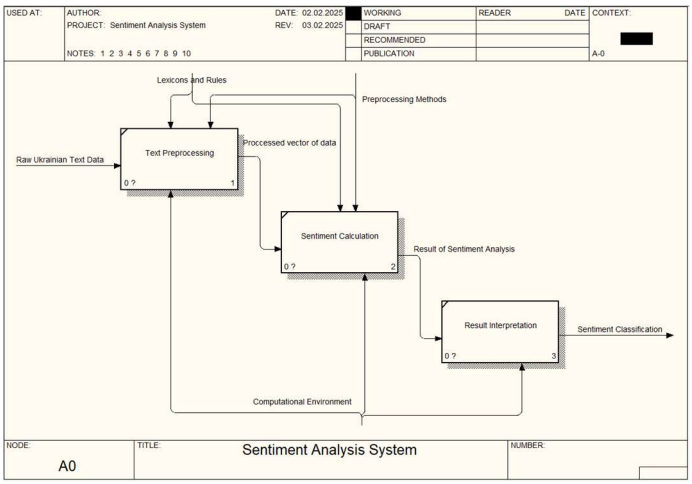


**Figure 3:** Decomposition diagram of the system

The decomposition diagram outlines the system as a series of interconnected stages, each performing a specific role within the sentiment analysis pipeline:

- Text preprocessing. The first sub-process is responsible for preparing raw Ukrainian text data. This step ensures that the input data is clean, structured, and ready for sentiment analysis.

- Sentiment calculation. After preprocessing, the processed text vector is passed to the sentiment calculation module. The output of this stage is the Result of Sentiment Analysis, a numerical or categorical representation of sentiment.

- Result interpretation. The final stage translates the analytical results into human-readable sentiment classifications. The output, Sentiment Classification, is presented to the user or passed to other systems for further use.

To enhance understanding, the sub-process of text preprocessing has been broken down further (Fig. 4) for clarification.

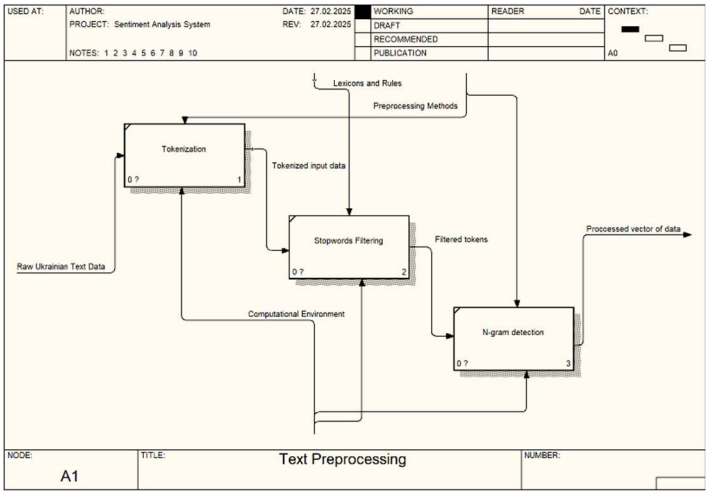


**Figure 4:** Decomposition diagram of the Text Preprocessing process

Text preprocessing involves three main steps. The first step, tokenization, breaks down the input text into individual tokens, such as words, punctuation, and symbols, while preserving meaningful text structures. The output of this process is tokenized input data, which serves as an intermediate representation for further refinement. The next step is stopwords filtering, where common stopwords, such as "і", "та", "або" in Ukrainian, are removed because they do not contribute to sentiment analysis. This results in a filtered set of tokens that are more relevant for sentiment classification. Finally, N-gram detection identifies sequences of words, such as bigrams or trigrams, that may carry contextual sentiment meaning. The output of this step is a processed vector of data, ready for sentiment analysis.

To better explain how different parts of the system work together a class diagram was built (Fig. 5). The *Sentiment Analysis System* consists of key entities that work together to process and analyze Ukrainian-language text. The *TextProcessor* handles initial preprocessing tasks such as tokenization, stopword removal, and n-gram generation. It prepares the raw text data for analysis. The *Lexicon* manages sentiment-related resources, including emotion lexicons, phrase sentiment scores, emoji sentiment mappings, and booster words, which are used to determine the polarity and intensity of sentiments within the text. The *DependencyParser* focuses on the syntactic structure of sentences, identifying elements like negations, modifiers, and punctuation that can influence sentiment. This information is critical for accurate sentiment adjustments. The *SentimentAnalyzer* serves as the core component, combining the outputs from the *TextProcessor, Lexicon*, and *DependencyParser* to calculate sentiment scores. It adjusts these scores based on contextual factors such as negations and modifiers.

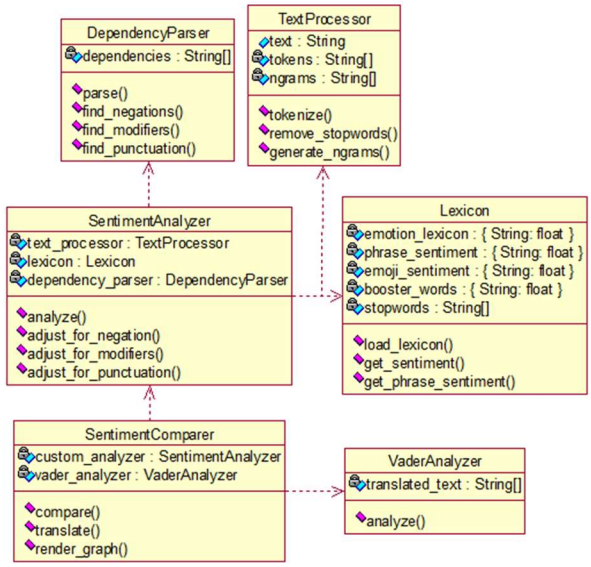


**Figure 5:** Class diagram of the system

For comparative analysis, the system includes a *VaderAnalyzer*, which applies the VADER sentiment analysis method, particularly useful after translating Ukrainian text into English. The *SentimentComparer* brings everything together, comparing results from the custom *SentimentAnalyzer* and the *VaderAnalyzer*. It also handles translation processes and visualizes sentiment distribution through graphs, allowing for an easy comparison of both methods' performance.

In order to show how both algorithms work in comparison, a software tool was built using modern UI and modern frameworks (Fig. 6).

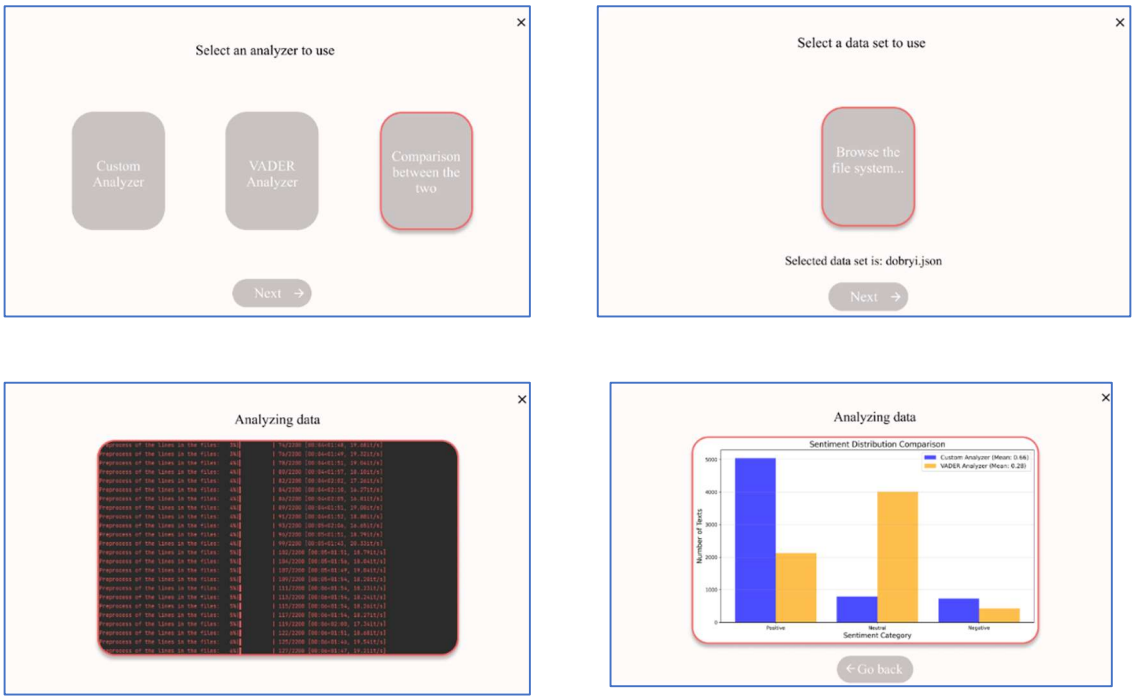


**Figure 6:** Software tool to compare sentiment analysis algorithms

The comparative methodology highlights the flexibility of the custom algorithm for Ukrainian-language content and indicates where VADER, with its reliance on English-specific resources, has

limited accuracy. This dual evaluation provides valuable insights into the strengths and weaknesses of rule-based approaches across different languages.

## 5. Results

This section assesses the performance of the proposed custom rule-based sentiment analysis algorithm compared to the VADER sentiment analysis tool. It evaluates its analysis on Ukrainian-language tweets and their ability to categorize the sentiment into positive, neutral, and negative.

### 5.1. Analysis of Specific Keywords

The subsets of the dataset with certain words with specific sentiments were taken, such as the words "добре" meaning "good", "добрий" meaning "kind", "погано" meaning "badly", and "поганий" meaning "bad".

*"Добре" (Good)*

Figure 7 shows that it classified a considerably higher percentage of tweets containing "добре" as positive at 83% compared with VADER's score at 58%. The compound score average from the custom analyzer was also considerably higher at 0.66 compared to the VADER result at 0.28, which reflects on the ability of the custom analyzer to use contextual positivity, having used its own lexicon with boosted weighting.

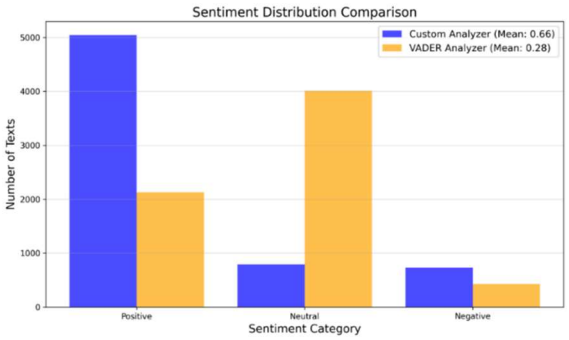


**Figure 7:** Sentiment distribution comparison of the dataset containing word “добре”

*"Добрий" (Kind)*

Figure 8 presents the results for tweets that contain "добрий.". As in "добре," the custom analyzer outperformed VADER in the classification of positive sentiment, 73% compared to 53%, with a mean compound score of 0.53 compared to VADER's 0.50. This further validates the effectiveness of the custom lexicon and the intensity booster words in enhancing the classification of sentiment.

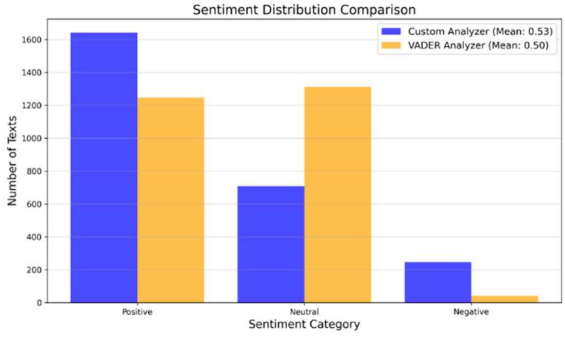


**Figure 8:** Sentiment distribution comparison of the dataset containing word “добрий”

*"Погано" (Badly)*

For the cases of "погано", the custom analyser gave better results as well than the identifications of negatives. Figure 9 depicts 65% vs 49%, where the custom analyst classified the number of tweets based on the determination of the negatives done by VADER. The -0.44 mean compound value of the former was closer compared to the score of the later, which averaged at -0.22 with respect to sentiments.

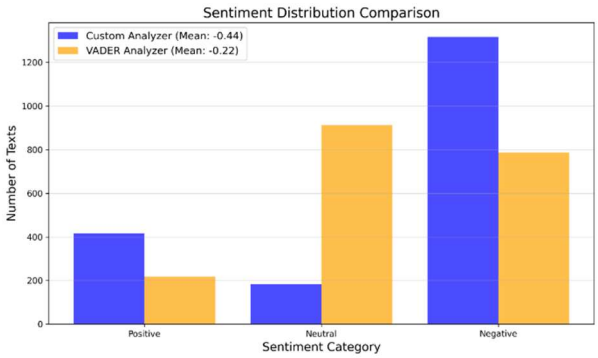


**Figure 9:** Sentiment distribution comparison of the dataset containing word "погано"

*"Поганий" (Bad)*

Figure 10 displays the results for "поганий". The custom analyzer classified 78% of the tweets as negative, while VADER did so for 61%. In addition, the mean compound score of -0.61 for the custom analyzer was much lower than VADER's -0.31, meaning that the former better matched the intensity of negative feeling in the dataset.

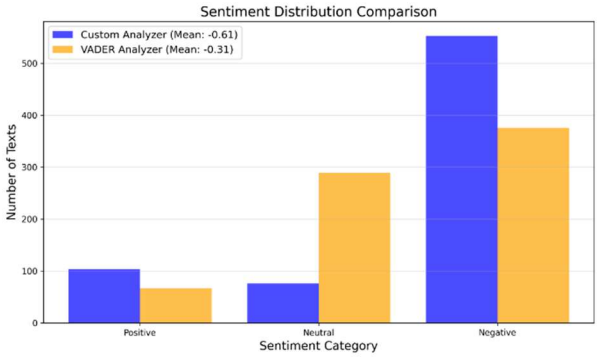


**Figure 10:** Sentiment distribution comparison of the dataset containing word "поганий"

## 6. Discussions

Here is a summary of the experiments results in a table comparing two sentiment analysis methods Vader Analyzer and Custom Analyzer (Table 1).

The results from table 1 and figure 11 have indicated that the custom rule-based analyzer performs superiorly to VADER on all tested datasets. The custom analyzer proves better at both highly positive and highly negative texts classification because of its following strengths:

1. *Expanded lexicon.* EMOLEX, polarity_score.csv, intensity booster words, and large phrase sentiment have included to enhance coverage for sentiment-laden words and phrases.
2. *Context-aware adjustments.* The algorithm makes use of dependency analysis and position-based weighting to account for modifiers, negations, and punctuation, enhancing the accuracy of sentiment classification.

3. *Emoji sentiment recognition.* A carefully curated emoji lexicon enables better handling of modern text features ignored by other algorithms.

**Table 1**
Results of the experiments

| Word | Custom Analyzer (Positive/Negative %) | VADER (Positive/Negative %) | Custom Mean Score | VADER Mean Score |
|---|---|---|---|---|
| добре (good) | 83% (positive) | 58% (positive) | 0.66 | 0.28 |
| добрий (kind) | 73% (positive) | 53% (positive) | 0.53 | 0.50 |
| погано (badly) | 65% (negative) | 49% (negative) | -0.44 | -0.22 |
| поганий (bad) | 78% (negative) | 61% (negative) | -0.61 | -0.31 |

Moreover, text translation for VADER added noise to the dataset, which may impact its results on a Ukrainian-language dataset. This once again points out the need for language-specific SA tools.


**Figure 11:** Comparison of the results

## Conclusions

This study introduced a novel rule-based sentiment analysis algorithm specifically designed for the Ukrainian language. By integrating a diverse set of linguistic resources, including the EMOLEX lexicon, polarity scores, emoji sentiment mapping, and intensity boosters, the proposed approach effectively addresses key challenges in Ukrainian-language sentiment analysis. Additionally, the incorporation of advanced dependency parsing and position-aware scoring enhances the algorithm's ability to process complex linguistic structures.

The evaluation, conducted using datasets from a previous study, demonstrates that the custom algorithm surpasses VADER in identifying sentiment polarity, particularly for texts with clear positive or negative sentiment. However, VADER remains competitive in detecting neutral content due to its generalized optimization for multiple languages. These findings underscore the necessity of language-specific sentiment analysis tools for non-English content.

The comparative analysis further confirms that a domain-specific rule-based algorithm, when supported by a well-structured lexicon and carefully designed linguistic rules, can achieve performance levels comparable to widely used sentiment analysis tools like VADER. The results highlight the potential of tailored linguistic approaches in improving sentiment analysis for underrepresented languages, paving the way for further advancements in this domain.

While the current state of the rule-based algorithm has shown great promise, there is much room for further improvement in several ways:

1. Adding AI Future editions could include integrating machine learning or deep learning models for further enhancement of the rule-based system. Hybrid approaches, that combine

rule-based techniques with neural network models, would likely enhance accuracy on ambiguous and nuanced content further.
2. Extension to other languages. After the success of the Ukrainian language model, the next step would be to extend the approach to other underrepresented languages. Multilingual capabilities would make the algorithm more applicable and increase its impact.
3. Dynamic lexicon updating. Extending the algorithm to automatically adapt its lexicon to emerging trends, slang, and domain-specific terminology through web scraping and natural language processing techniques.
4. Sentiment granularity. Future work could include finer grain analysis, for instance, of the intensity of a sentiment as "somewhat" versus "highly" positive.
5. Benchmarking against AI models. Future studies can be conducted for benchmarking with state-of-the-art AI-based sentiment analysis models, such as BERT or GPT, to position the advantages and disadvantages of the rule-based approach relative to these models.

This study thus constitutes the basis for further explorations of language- and domain-specific sentiment analysis. This approach, though, has its special promise, in that integrating AI models allow for a finally achieved balance between interpretability typical of rule-based methods and adaptability of machine learning techniques.

## Declaration on Generative AI

The authors have not employed any Generative AI tools.

## References


[1] S. Gupta, R. Ranjan, S. N. Singh, Comprehensive Study on Sentiment Analysis: From Rule-based to Modern LLM-based Systems, arXiv preprint arXiv:2409.09989, 2024.

[2] Kotelnikova, D. Paschenko, K. Bochenina, E. Kotelnikov, Lexicon-based Methods vs. BERT for Text Sentiment Analysis, arXiv preprint arXiv:2111.10097, 2021.

[3] D. Vilares, C. Gómez-Rodríguez, M. A. Alonso, Universal, Unsupervised (Rule-Based), Uncovered Sentiment Analysis, arXiv preprint arXiv:1606.05545, 2016.

[4] O. Kellert, M. U. Zaman, N. H. Matlis, C. Gómez-Rodríguez, Experimenting with UD Adaptation of an Unsupervised Rule-based Approach for Sentiment Analysis of Mexican Tourist Texts, arXiv preprint arXiv:2309.05312, 2023.

[5] O. Al-Harbi, Negation Handling in Machine Learning-Based Sentiment Classification for Colloquial Arabic, arXiv preprint arXiv:2107.11597, 2021.

[6] Mediakov O., Basyuk T. Specifics of Designing and Construction of the System for Deep Neural Networks Generation // CEUR Workshop Proceedings. – 2022. – Vol. 3171: Computational Linguistics and Intelligent Systems 2022: Proceedings of the 6th International conference on computational linguistics and intelligent systems (COLINS 2022). Vol. 1: Main conference, Gliwice, Poland, May 12-13, 2022. – P. 1282–1296.

[7] L. Zhang, S. Wang, B. Liu, Deep Learning for Sentiment Analysis: A Survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), 2018.

[8] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, Lexicon-Based Methods for Sentiment Analysis, Computational Linguistics, 37(2), 2011, pp. 267-307.

[9] B. Liu, Sentiment Analysis and Opinion Mining, Synthesis Lectures on Human Language Technologies, 5(1), 2012, pp. 1-167.

[10] E. Cambria, A. Hussain, Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis, Springer, 2015.

[11] E. Cambria, Q. Liu, S. Decherchi, F. Xing, K. Kwok, SenticNet 7: A Commonsense-Based Neurosymbolic AI Framework for Explainable Sentiment Analysis, Proceedings of the 29th ACM International Conference on Information and Knowledge Management, 2020, pp. 105-114.

[12] Basyuk T., Vasyliuk A. Approach to a subject area ontology visualization system creating // CEUR Workshop Proceedings. – 2021. – Vol. 2870: Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Lviv, Ukraine, April 22–23, 2021. Volume I: main conference. – P. 528–540.
[13] Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, Sentiment Analysis of Twitter Data, Proceedings of the Workshop on Languages in Social Media, 2011, pp. 30-38.
[14] Dos Santos, M. Gatti, Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts, Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 69-78.
[15] M. Hu, B. Liu, Mining and Summarizing Customer Reviews, Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 168-177.
[16] R. Titov, McDonald, A Joint Model of Text and Aspect Ratings for Sentiment Summarization, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, 2008, pp. 308-316.
[17] S. M. Kim, E. Hovy, Determining the Sentiment of Opinions, Proceedings of the 20th International Conference on Computational Linguistics, 2004, pp. 1367-1373.
[18] P. D. Turney, Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp. 417-424.
[19] B. Pang, L. Lee, S. Vaithyanathan, Thumbs Up? Sentiment Classification Using Machine Learning Techniques, Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, 2002, pp. 79-86.
[20] E. Riloff, J. Wiebe, Learning Extraction Patterns for Subjective Expressions, Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, 2003, pp. 105-112.
[21] T. Basyuk, A. Vasyliuk, Peculiarities of an Information System Development for Studying Ukrainian Language and Carrying out an Emotional and Content Analysis // CEUR Workshop Proceedings. – 2023. – Vol. 3396: Computational Linguistics and Intelligent Systems 2023: Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems. Volume II: Computational Linguistics Workshop, Kharkiv, Ukraine, April 20-21, 2023. pp. 279–294.
[22] S. Poria, E. Cambria, R. Bajpai, A. Hussain, A Review of Affective Computing: From Unimodal Analysis to Multimodal Fusion, Information Fusion, 37, 2017, pp. 98-125.
[23] E. Cambria, B. Schuller, Y. Xia, C. Havasi, New Avenues in Opinion Mining and Sentiment Analysis, IEEE Intelligent Systems, 28(2), 2013, pp. 15-21.
[24] S. M. Kim, E. Hovy, Identifying and Analyzing Judgment Opinions, Proceedings of the Human Language Technology Conference of the NAACL, 2006, pp. 200-207.
[25] L. Dey, S. K. M. Haque, Opinion Mining from Noisy Text Data, International Journal on Document Analysis and Recognition, 12(3), 2009, pp. 205-226.
[26] E. Cambria, A. Hussain, Sentic Computing: Techniques, Tools, and Applications, Springer, 2012.
[27] C. G. Akcora, M. A. Bayir, M. Demirbas, H. Ferhatosmanoglu, Identifying Breakpoints in Public Opinion, Proceedings of the First Workshop on Social Media Analytics, 2010, pp. 62-66.
[28] Y. Kim, Convolutional Neural Networks for Sentence Classification, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746-1751.
[29] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts, Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631-1642.
[30] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation, 9(8), 1997, pp. 1735-1780.

[31] Graves, N. Jaitly, A. Mohamed, Hybrid Speech Recognition with Deep Bidirectional LSTM, Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, 2013, pp. 273-278.
[32] Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is All You Need, Advances in Neural Information Processing Systems, 30, 2017, pp. 5998-6008.
[33] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171-4186.
[34] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, arXiv preprint arXiv:1301.3781, 2013.
[35] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching Word Vectors with Subword Information, Transactions of the Association for Computational Linguistics, 5, 2017, pp. 135-146.
[36] J. Pennington, R. Socher, C. D. Manning, GloVe: Global Vectors for Word Representation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532-1543.
[37] Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving Language Understanding by Generative Pre-Training, OpenAI preprint, 2018.
[38] Radford, J. Wu, R. Child, D. Luan, D., Language Models are Unsupervised Multitask Learners, OpenAI preprint, 2019.
[39] T. Brown, B. Mann, N. Ryder, M. Subbiah, Language Models are Few-Shot Learners, Advances in Neural Information Processing Systems, 33, 2020, pp. 1877-1901.
[40] Y. Liu, M. Ott, N. Goyal, J. Du, RoBERTa: A Robustly Optimized BERT Pretraining Approach, arXiv preprint arXiv:1907.11692, 2019.