**УДК 004.42**
**DOI: 10.18523/2617-3808.2025.8.102-107**

*Дубовик А. В., Волинець Є. А.*

# АВТОМАТИЧНА КЛАСИФІКАЦІЯ ТЕКСТІВ

*У цьому дослідженні здійснено аналіз сучасних підходів до класифікації текстової інформації. Особливу увагу приділено автоматичній класифікації текстів, що передбачає їхній розподіл за визначеними категоріями без використання ручного аналізу. Розглянуто й порівняно ефективність різних методів класифікації з акцентом на гібридні системи, які здатні поєднувати переваги окремих підходів і забезпечувати підвищену точність та продуктивність моделей. Також обґрунтовано вибір інструментальних засобів для подальшої програмної реалізації системи автоматизованої класифікації текстів за категоріями. Для навчання моделей запропоновано використовувати збірку AG News Classification Dataset з платформи kaggle.com. Доцільним вважається обмеження класифікаційного процесу комбінацією трьох моделей — Naive Bayes, Support Vector Machine (SVM) та Recurrent Neural Networks (RNN), які вирізняються невисокими вимогами до обчислювальних ресурсів і часу на тренування.*

**Ключові слова:** класифікація текстів, машинне навчання, оброблення української мови, Naive Bayes, SVM, RNN, попереднє оброблення тексту.

## Вступ

Оброблення текстової інформації має суттєве значення в різних сферах знань сучасного світу, таких як бізнес, наука, соціальні медіа та інші. З розвитком інформаційних технологій і зростанням обсягу доступної інформації стає надзвичайно важливим ефективне управління цими даними та їхній аналіз. Однією з ключових задач у зазначеній сфері є автоматична класифікація текстів, яка полягає в їхньому розподілі за певними категоріями без використання при цьому ручного аналізу (під «категорією» може матися на увазі тема, почуття, мова тексту тощо). Це завдання може бути важливим для багатьох установ під час створення архівів і організації великих обсягів документів. Воно також стає дедалі актуальнішим з розвитком Інтернету, де величезні обсяги різноманітної інформації потребують швидкого та точного аналізу. Зазначимо при цьому, що інформаційні системи, які використовують технології машинного навчання, такі як нейронні мережі та методи оброблення природної мови (NLP), демонструють високу ефективність у розв'язанні цієї задачі завдяки своїй здатності адаптуватися до нових даних і високій швидкості оброблення [1; 4; 8; 20].

Основні складнощі, пов'язані з автоматичною класифікацією текстів, це різноманітність форматів, структур текстів і семантична складність мови, і для їхнього розв'язання потрібні ефективні алгоритми та методи оброблення природної мови.

## Основні поняття

Класифікація є процесом розподілу об'єктів або даних на певні категорії або класи відповідно до їхніх характеристик або властивостей. У контексті оброблення текстів класифікація полягає у призначенні кожному текстовому документу або фрагменту відповідної категорії або класу, що допомагає організувати, проаналізувати та зрозуміти великі обсяги інформації [4]. До XX століття класифікацію текстів здійснювали виключно вручну. У XIX столітті, з поширенням бібліотек та друкарства, було створено системи класифікації книг, такі як десяткова класифікація Дьюї та класифікація Бібліотеки Конгресу.

Потреба у класифікації текстів стала ще більш актуальною з появою комп'ютерів, які дали змогу автоматизувати процес обробки текстів і розробити алгоритми для автоматичної класифікації текстових даних. Перші комп'ютерні системи для класифікації текстів з'явилися ще в другій половині XX століття, але вони були дуже примітивними порівняно з сучасними системами. Технології штуч-

ного інтелекту, зокрема методи машинного навчання, дозволили значно покращити ефективність та точність класифікації текстів.

Автоматична класифікація тексту є одним із фундаментальних завдань оброблення природної мови, і її дослідження матиме важливі наслідки у багатьох сферах життя. Наприклад, у наукових дослідженнях відсутність ефективних засобів для аналізу великих обсягів наукових публікацій може ускладнити синтез знань і виявлення тенденцій розвитку науки. У сфері бізнесу автоматична класифікація текстів може допомогти під час розподілу замовлень або при аналізі ринку, що сприятиме прийняттю обґрунтованих управлінських рішень. Також у сфері соціальних мереж і медіа класифікація текстових повідомлень може допомогти під час аналізу громадської думки та виявлення суспільних тенденцій.

Усі системи автоматичної класифікації текстів можна умовно розділити на такі три групи: системи на основі правил, системи на основі машинного навчання, гібридні системи.

Одним із найпростіших та найбільш зрозумілих методів класифікації текстів є класифікація за допомогою систем на основі правил. Ці системи базуються на використанні набору правил, які визначають спосіб прийняття рішень щодо класифікації текстових даних [19]. Прикладом може бути проста система фільтрації спаму електронної пошти, яка відносить повідомлення, насичені словами «акція», «знижка», «продаж», «виграш», «лотерея», до спаму. Хоча цей підхід може спершу здатися доволі обмеженим порівняно з більш складними методами машинного навчання, він все ще знаходить своє застосування у деяких випадках. Перевагою систем на основі правил є їхня прозорість: ви легко можете зрозуміти, які правила були застосовані до конкретного тексту під час класифікації. Однак недоліком цих систем є їхня обмежена гнучкість, а також відносна складність побудови та підтримки, особливо якщо застосовувати зазначені системи для обробки великих обсягів текстових даних.

У системах класифікації текстів на основі машинного навчання використовуються різноманітні алгоритми та моделі для автоматичного визначення категорій або класів, до яких належать тексти [19]. Ці системи зазвичай навчаються на зразках текстів з уже відомими мітками категорій, які дозволяють зрозуміти певні закономірності у вхідних даних та правильно класифікувати нові зразки. Для класифікації текстів найбільш широко застосовують такі рішення, як Naive Bayes, SVM (Support Vector Machine), RNN (Recurrent Neural Network), BERT (Bidirectional Encoder Representations from Transformers), LLM (Large Language Models).

Naive Bayes — це проста техніка для побудови класифікаторів, що ґрунтується на теоремі Баєса та припущенні про незалежність ознак [16]. Наприклад, якщо фрукт зелений або червоний, круглої форми й має радіус близько п’яти сантиметрів, то його можна вважати яблуком — саме до цього висновку приводить використання наївного баєсового класифікатора, який при визначенні ймовірності того, що цей фрукт є яблуком, розглядає кожну ознаку як незалежну від інших. Перевагами методу Naive Bayes є його ефективність для великих обсягів даних, простота реалізації та швидкість. Недоліком є те, що припущення про незалежність ознак не завжди може бути справедливим, і в цих випадках метод показуватиме нижчу ефективність порівняно з іншими методами.

Для текстової класифікації найчастіше використовується мультиноміальний баєсів класифікатор, в якому розподіл є параметризованим векторами $\theta_y = (\theta_{y1}, \dots, \theta_{yn})$ для кожного класу $y$, де $n$ — це кількість ознак (у цьому випадку розмір словника) та $\theta_{yi}$ — це вірогідність $P(x_i|y)$ ознаки $i$ з’явитись в екземплярі класу $y$. Параметри $\theta_{yi}$ оцінюються підрахунком відносної частоти:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \ ,$$

де $N_{yi} = \sum_{x \in T} x_i$ — це кількість разів, коли ознака $i$ трапляється в класі $y$ в тренувальному наборі даних $T$, і $N_y = \sum_{i=1}^{n} N_{yi}$ — це загальна кількість всіх ознак в класі $y$. Попереднє згладжування $\alpha \geq 0$ враховує функції, яких немає у зразках навчання, і запобігає нульовій ймовірності в подальших обчисленнях. Встановлення $\alpha = 1$ називають згладжуванням Лапласа, $\alpha < 1$ — згладжуванням Лідстоуна [19].

SVM — алгоритм, який шукає в просторі ознак гіперплощину, що найкращим чином розподіляє екземпляри по різних категоріях [16]. Перевагою такого методу є ефективність при роботі з розрідженими даними, а недоліком є те, що за великої кількості зразків для тренування або великої кількості ознак алгоритм може стати обчислювально-вимогливим і також може демонструвати гірші, ніж інші алгоритми, результати при обробленні великих корпусів текстів.

Загалом SVM працює таким чином. Ми маємо набір $n$ точок із тренувального набору даних $(x_1, y_1)$, ... $(x_n, y_n)$, де $y_i$ (1 або –1) вказує на належність точки $x_i$ до певного класу. Кожен $x_i$ — це $p$-розмірний вектор, і SVM шукає гіперплощину, яка з максимальним запасом розділяє групу точок $x_i$, для яких $y_1 = 1$, від тих точок, для яких $y_1 = -1$.

RNN — рекурентна нейронна мережа, що може ефективно працювати з послідовностями даних, такими як рядки тексту, і враховувати контекстуальні зв'язки між словами [4]. Перевагою мереж такого типу є здатність до врахування контексту, що може бути корисним для визначення залежностей між словами та фразами в реченні. Проте оброблення довгих текстових послідовностей для RNN може більш обчислювально-затратною операцією, ніж для Naive Bayes та SVM. Також рекурентні нейронні мережі мають проблему затухання градієнта і можуть погано зв'язувати інформацію при великій відстані між словами. Як вихід із такої ситуації в 1997 р. було запропоновано модель LSTM, якій протягом останніх років удалось набагато покращити можливості RNN. Наприклад, повідомляється, що у 2015 р. розпізнавання мовлення від Google значно покращилось завдяки використанню LSTM [15].

BERT — це одна з найпродуктивніших моделей у сфері оброблення природної мови, що базується на трансформерах [5]. Модель зазвичай попередньо навчається на великому корпусі текстів, а потім підлаштовується для виконання конкретних завдань. Основною перевагою BERT є здатність розуміти контекстне представлення слів і його масштабованість. Недоліками моделі можна вважати потребу в значних обчислювальних ресурсах для тренування та підлаштування, а також складність її налаштування порівняно з більш традиційними методами.

LLM — моделі, які базуються на нейронних мережах із великою кількістю параметрів та навчаються на великих обсягах текстових даних із метою розуміти та генерувати контент [12]. Перевагою таких моделей є їхня гнучкість, адже уже натреновані LLM здатні визначити дуже велику кількість категорій тексту. Проте значним недоліком моделей такого типу є використання потужної обчислювальної машини для роботи та необхідність надзвичайно великого обсягу даних для тренування у випадку, якщо ми створюємо власну модель, а не використовуємо наявну. Деякі відомі LLM — це моделі серії GPT від OpenAI (наприклад, GPT-3.5 і GPT-4, що використовуються в ChatGPT і Microsoft Copilot), PaLM і Gemini від Google (останній сьогодні використовується в однойменному чат-боті), xAI Grok, сімейство моделей LLaMA від Meta. Але саме браузерний ChatGPT 2022 року, орієнтований на споживачів, захопив уяву широких верств населення та викликав ажіотаж у ЗМІ [3].

Гібридні системи класифікації текстів можуть поєднувати переваги різних підходів і методів для створення більш точних та ефективних моделей [2]. Ці системи можуть використовувати комбінації всіх інших методів для досягнення кращих результатів. Вони становлять потужний інструмент, який може бути налаштований та оптимізований для конкретних потреб. Такі моделі досягають високої точності та ефективності у різних сценаріях та сферах застосування. Отже, гібридні системи класифікації текстів є ефективним інструментом, який можна використовувати для різноманітних завдань у багатьох галузях, від аналізу текстових даних до автоматичної обробки природної мови. Проте один з основних недоліків гібридних систем полягає в їхній складності. Поєднання різних методів і підходів може призвести до значного ускладнення самої системи — як її розробки, так і підтримки та масштабування.

Отже, потрібно розробити систему, яка зможе доволі швидко та точно класифікувати тексти. У користувача має бути можливість зручним способом натренувати систему на власних даних та налаштувати її параметри для отримання оптимальних результатів. Система буде попередньо навчена на певному наборі даних (наприклад, AG News Classification Dataset), що дозволить одразу розпізнавати деякі базові категорії текстів.

## Аргументація вибору основних інструментів розробки

Для того щоб ефективно опрацювати вхідні дані та реалізувати алгоритм класифікації, необхідно насамперед визначити мову програмування та основні бібліотеки, які зможуть надати необхідний функціонал.

На нашу думку, тут найбільше підходить мова програмування Python, оскільки саме для цієї мови вже сформована екосистема бібліотек машинного навчання, оброблення даних, матричної

математики тощо. Синтаксис Python є лаконічним і зрозумілим, що в деяких випадках дозволяє реалізовувати більш складні алгоритми швидше та з меншими зусиллями. Великий вибір різноманітних бібліотек, простота та зручність використання, а також всі інші згадані фактори роблять Python стандартним вибором для проєктів, що стосуються машинного навчання та нейронних мереж.

Основними бібліотеками з наявним функціоналом ми обираємо такі. TensorFlow — безкоштовна бібліотека для машинного навчання з відкритим кодом [18]. Вона відома своєю масштабованістю та високою продуктивністю, що дозволяє легко створювати та навчати складні нейронні мережі. Ще одна бібліотека для машинного навчання — scikit-learn, вона надає простий та зрозумілий інтерфейс для реалізації класичних алгоритмів [16]. Ця бібліотека містить великий набір інструментів для класифікації та вирішення інших завдань. Бібліотеку nltk (Natural Language Toolkit) використовують для опрацювання природної мови [10]. Вона надає широкий спектр інструментів для реалізації різноманітних завдань, пов'язаних з обробленням природної мови. Також корисними будуть бібліотеки NumPy [11] — для зручного доступу до великих і багатовимірних масивів, а також функцій високого рівня для роботи з такими масивами; Pandas [13] — для використання швидкої, гнучкої та інтуїтивно зрозумілої табличної структури даних DataFrame; Matplotlib [9] та seaborn [17] — для візуалізації даних за допомогою графіків різних типів.

Цей список засобів не є вичерпним, адже для деяких задач можна використати також інші бібліотеки з аналогічним функціоналом.

## Висновки

У цьому дослідженні проаналізовано задачу класифікації текстів за категоріями з наголосом на оброблення текстів українською мовою. Обґрунтовано інструментарій розроблення майбутньої програмної реалізації автоматичної системи класифікації текстів за категоріями.

Для навчання моделей можна застосувати AG News Classification Dataset з сайту kaggle.com [6]. Всього цей датасет містить 120 тисяч коротких текстів, зібраних з різних новинних джерел. Щоб робота з даними була більш ефективною, потрібно розробити модуль, який виконував би попереднє обробленння текстів.

Для вирішення нашої задачі ми вважаємо недоцільним підхід із використанням систем на основі правил, адже ми не зможемо адаптувати старі правила для нових категорій у випадку, якщо користувач натренує модель на власних текстах. Тому для класифікації ми пропонуємо обмежитися використанням комбінації трьох моделей: Naive Bayes, SVM та RNN. Вони не потребують потужних обчислювальних ресурсів та великої кількості часу для тренування.

Важливим при цьому має бути можливість використовувати будь-яку комбінацію моделей для тренування на користувацьких текстах. Так само і для класифікації повинна бути можливість використовувати будь-яку комбінацію натренованих моделей.

### *Список літератури*


1. Глибовець А. М. Програмна система перевірки на плагіат українських текстів / А. М. Глибовець, М. О. Бікчентаєв // Наукові записки НаУКМА. Комп'ютерні науки. — 2022. — Т. 5. — https://doi.org/10.18523/2617-3808.2022.5.16-25.
2. A Hybrid Text Classification Approach for Analysis of Student Essays [Electronic resource] / C. P. Rosé, A. Roque, D. Bhembe, K. Vanlehn. — Mode of access : https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7f7e133f636308b5f67600ad321335f716a7a14a.
3. ChatGPT a year on: 3 ways the AI chatbot has completely changed the world in 12 months [Electronic resource]. — Mode of access: https://www.euronews.com/next/2023/11/30/chatgpt-a-year-on-3-ways-the-aihttps://www.euronews.com/next/2023/11/30/chatgpt-a-year-on-3-ways-the-ai-chatbot-has-completely-changed-the-world-in-12-monthschatbot-has-completely-changed-the-world-in-12-months.
4. Gasparetto A. A Survey on Text Classification Algorithms: From Text to Predictions [Electronic resource] / A. Gasparetto, M. Marcuzzo, A. Zangari, A. Albarelli. — Mode of access : https://www.mdpi.com/2078-2489/13/2/83.
5. Google voice search: faster and more accurate [Electronic resource]. — Mode of access: https://research.google/blog/google-voice-search-fasterhttps://research.google/blog/google-voice-search-faster-and-more-accurateand-more-accurate.
6. Kaggle [Electronic resource]. — Mode of access: https://www.kaggle.com.
7. Large language model (LLM) [Electronic resource]. — Mode of access: https://www.growthloop.com/university/article/llm.
8. Machine Learning Glossary [Electronic resource]. — Mode of access: https://developers.google.com/machine-learning/glossary.
9. Matplotlib [Electronic resource]. — Mode of access: https://matplotlib.org.
10. Natural Language Toolkit [Electronic resource]. — Mode of access: https://www.nltk.org.
11. NumPy [Electronic resource]. — Mode of access: https://numpy.org.

12. Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing [Electronic resource]. — Mode of access: https://research.google/blog/open-sourcing-bert-state-of-the-art-pre-traininghttps://research.google/blog/open-sourcing-bert-state-of-the-art-pre-training-for-natural-language-processingfor-natural-language-processing.
13. Pandas [Electronic resource]. — Mode of access: https://pandas.pydata.org.
14. Scikit-learn: Naive Bayes [Electronic resource]. — Mode of access: https://scikit-learn.org/stable/modules/naive_bayes.html.
15. Support vector machine [Electronic resource]. — Mode of access: https://en.wikipedia.org/wiki/Support_vector_machine.
16. Scikit-learn [Electronic resource]. — Mode of access: https://scikit-learn.org/stable.
17. Seaborn [Electronic resource]. — Mode of access: https://seaborn.pydata.org.
18. TensorFlow [Electronic resource]. — Mode of access: https://www.tensorflow.org.
19. Understanding Text Classification in Python [Electronic resource]. — Mode of access: https://www.datacamp.com/tutorial/text-classificationhttps://www.datacamp.com/tutorial/text-classification-pythonpython.
20. Zie Eya Ekolle. A Generative Learning Model for Heterogeneous Text Classification Based on Collaborative Partial Classifications [Electronic resource] / Zie Eya Ekolle, Ryuji Kohno. — Mode of access: https://www.mdpi.com/2076-3417/13/14/8211.

***References***

ChatGPT a year on: 3 ways the AI chatbot has completely changed the world in 12 months. (2023). https://www.euronews.com/next/2023/11/30/chatgpt-a-year-on-3-ways-the-aihttps://www.euronews.com/next/2023/11/30/chatgpt-a-year-on-3-ways-the-ai-chatbot-has-completely-changed-the-world-in-12-monthschatbot-has-completely-changed-the-world-in-12-months.

Ekolle, Zie Eya, & Kohno, Ryuji. (2023). *A Generative Learning Model for Heterogeneous Text Classification Based on Collaborative Partial Classifications*. https://www.mdpi.com/2076-3417/13/14/8211.

Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). *A Survey on Text Classification Algorithms: From Text to Predictions*. https://www.mdpi.com/2078-2489/13/2/83.

Google voice search: faster and more accurate. (2015). https://research.google/blog/google-voice-search-faster-and-more-accurate.

Hlybovets, A., & Bikchentaev, M. (2022). Prohramna systema perevirky na plahiat ukrainskykh tekstiv. *NaUKMA Research Papers. Computer Science*, *5*, 16–25. https://doi.org/10.18523/2617-3808.2022.5.16-25 [in Ukrainian].

Kaggle. (n. d.). https://www.kaggle.com.

Large language model (LLM). (2024). https://www.growthloop.com/university/article/llm.

Machine Learning Glossary. (n. d.). https://developers.google.com/machine-learning/glossary.

Matplotlib Documentation. (n. d.). https://matplotlib.org.

Natural Language Toolkit Documentation. (n. d.). https://www.nltk.org.

NumPy Documentation. (n. d.). https://numpy.org.

Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing. (2018). https://research.google/blog/open-sourcing-bert-state-of-the-art-pre-traininghttps://research.google/blog/open-sourcing-bert-state-of-the-art-pre-training-for-natural-language-processingfor-natural-language-processing.

Pandas Documentation. (n. d.). https://pandas.pydata.org.

Rosé, C. P., Roque, A., Bhembe, D., & Vanlehn, K. (2003). *A Hybrid Text Classification Approach for Analysis of Student Essay*. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7f7e133f636308b5f67600ad321335f716a7a14a.

Scikit-learn Documentation. (n. d.). https://scikit-learn.org/stable.

Scikit-learn: Naive Baye. (n. d.). https://scikit-learn.org/stable/modules/naive_bayes.html.

Seaborn Documentation. (n. d.). https://seaborn.pydata.org.

Support vector machine. (n. d.). https://en.wikipedia.org/wiki/Support_vector_machine.

TensorFlow Documentation. (n. d.). https://www.tensorflow.org.

Understanding Text Classification in Python. (2022). https://www.datacamp.com/tutorial/text-classificationhttps://www.datacamp.com/tutorial/text-classification-pythonpython.

*A. Dubovyk, Y. Volynets*

## AUTOMATIC TEXT CLASSIFICATION

*This study explores modern methodologies in the field of automatic text classification, a critical task in natural language processing (NLP) that enables the categorization of unstructured textual data into predefined groups without manual intervention. The rapid growth of digital text across domains such as business, media, science, and social networks has created a pressing need for scalable and accurate classification systems. The research provides an analytical overview of three primary approaches: rule-based systems, machine learning methods, and hybrid models. Particular attention is paid to evaluating the strengths and limitations of several popular machine learning algorithms, including Naive Bayes, Support Vector Machines (SVM), and Recurrent Neural Networks (RNN). While advanced techniques such as BERT and Large Language Models (LLMs) demonstrate high performance, they are not considered optimal for lightweight, user-trainable applications due to their high computational costs.*

*To support practical implementation, the study proposes a system architecture based on the Python programming language and a suite of supporting libraries (e.g., TensorFlow, scikit-learn, NLTK, NumPy,*

*Pandas, Matplotlib, and Seaborn). The AG News Classification Dataset is recommended as the initial training corpus, providing a robust foundation for multi-class categorization tasks.*

*The final system design emphasizes modularity and user configurability. It allows end users to preprocess their own text data, train classification models on domain-specific content, and utilize combinations of models to improve performance. The research recommends a model ensemble consisting of Naive Bayes, SVM, and RNN due to their balance between effectiveness and computational efficiency.*

*This study not only highlights the technical viability of automated text classification systems but also presents a practical, extensible framework suitable for real-world applications, especially for underrepresented languages such as Ukrainian. The resulting system aims to bridge the gap between academic research and deployable technology, offering a customizable platform for tasks ranging from document organization and content filtering to sentiment analysis and market research.*

**Keywords:** text classification, machine learning, Ukrainian language processing, Naive Bayes, SVM, RNN, text preprocessing.

*Матеріал надійшов 14.06.2025*