Одержано редакцією 19.09.2025 р.
Прийнято до публікації 17.12.2025 р.

УДК 004.85+025.4

DOI 10.31651/2076-5886-2025-1-72-85

PACS 07.05.Mh

КРАСНОШЛИК Наталія Олександрівна
кандидат технічних наук, доцент, доцент кафедри прикладної математики та інформатики Черкаського національного університету імені Богдана Хмельницького
e-mail: krasnoshlyk@vu.cdu.edu.ua
ORCID 0000-0003-4661-6997

БОГАТИРЕНКО Павло Русланович
студент спеціальності «Інформаційні системи та технології» Черкаського національного університету імені Богдана Хмельницького
e-mail: bogatyrenko.pavlo@vu.cdu.edu.ua

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ АВТОМАТИЧНОЇ КЛАСИФІКАЦІЇ ТЕКСТУ У БІБЛІОГРАФІЧНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

У статті представлено результати комплексного дослідження методів автоматичної класифікації тексту стосовно їх застосування у бібліографічних інформаційних системах. Розглянуто широкий спектр підходів – від класичних статистичних методів машинного навчання до сучасних архітектур глибокого навчання на основі трансформерів. Для кожного із досліджуваних методів проведено аналіз особливостей попередньої обробки бібліографічних текстів, методів векторизації та параметрів налаштування моделей. Розроблено модульну систему класифікації бібліографічних записів мовою Python з використанням фреймворків scikit-learn, PyTorch та FastAPI. Проведено порівняльне оцінювання шести моделей за метриками точності (Accuracy, Precision, Recall, F1-Score), швидкодії та ресурсоємності. Встановлено, що модель BERT досягає найвищої якості класифікації (F1-Score = 0.912), тоді як метод опорних векторів (SVM) забезпечує оптимальне співвідношення між точністю та продуктивністю для систем реального часу. Сформульовано практичні рекомендації щодо вибору методу класифікації залежно від вимог конкретного застосування.

Ключові слова: *автоматична класифікація тексту, бібліографічні інформаційні системи, машинне навчання, BERT, метод опорних векторів, глибоке навчання, обробка природної мови, векторизація тексту.*

Вступ

Стрімкий розвиток цифрових технологій та інтенсивність науково-технічних комунікацій зумовлюють постійне зростання обсягів текстових даних, що накопичуються у бібліографічних інформаційних системах (БІС), електронних репозитаріях та наукометричних базах даних. За оцінками дослідників, щорічно у світі публікується понад 2,5 мільйона наукових статей [1], що формує критичну потребу в ефективних засобах тематичної організації та пошуку наукової інформації. Без автоматизованих інструментів навігація в такому інформаційному просторі була б практично неможливою, а пошук релевантних публікацій займав би у дослідників надмірно багато часу.

Традиційні підходи до каталогізації, що ґрунтуються на ручному або

напівавтоматичному індексуванні, вже не відповідають вимогам масштабованості та оперативності сучасних інформаційних середовищ. Досвідчений каталогізатор спроможний опрацювати лише 20–30 документів на день [1], що є абсолютно недостатнім в умовах безперервного та стрімкого приросту наукової продукції. Це робить автоматичну класифікацію не просто зручною опцією, а критичною необхідністю для забезпечення своєчасної обробки нових публікацій. Дослідження показують, що коректна тематична класифікація здатна підвищити релевантність результатів пошуку на 30–40% порівняно із системами, що покладаються виключно на повнотекстовий пошук без тематичного структурування [1].

Питання застосування різних підходів до класифікації текстів у контексті інформаційного пошуку розглядалось у ряді фундаментальних робіт. Зокрема, Манінг з колегами [10] детально описав математичні основи пошуку інформації та класифікації, включаючи метрики оцінки якості. Комплексний огляд алгоритмів класифікації текстів запропонований у роботі Ковсарі та ін. [5], де систематизовано понад 40 методів. Специфіку застосування SVM для класифікації текстів вперше детально описав Йоахімс [6], довівши перевагу лінійних ядер для текстових даних. З появою трансформерів BERT [8] та наступних моделей парадигма класифікації текстів зазнала революційних змін: для ряду задач точність зросла на 5–10 відсоткових пунктів. Спеціалізована модель SciBERT [15], навчена на корпусі наукових публікацій, демонструє ще кращі результати для класифікації наукових текстів порівняно із загальновживаними моделями. Огляд літератури засвідчує, що попри значну кількість досліджень у галузі класифікації текстів, систематичні порівняльні дослідження саме для задачі тематичної категоризації бібліографічних записів з урахуванням вимог до продуктивності та ресурсоємності залишаються відносно нечисленними.

Особливу актуальність набуває застосування методів автоматичної класифікації саме до бібліографічних текстів, оскільки вони характеризуються специфічною структурною неоднорідністю, обмеженою довжиною та надзвичайно високою термінологічною насиченістю. Це потребує адаптації загальних алгоритмів машинного навчання та інтелектуального аналізу даних до специфіки предметної галузі, що робить дану тематику науково значущою та практично важливою. Наявні огляди алгоритмів класифікації текстів [5, 10] засвідчують широкий спектр доступних підходів, проте систематичне порівняння їх ефективності саме для бібліографічних даних у науковій літературі представлене недостатньо повно.

Таким чином, дослідження та порівняльна оцінка методів автоматичної класифікації тексту стосовно задачі організації бібліографічних записів є актуальним завданням, вирішення якого має як теоретичне, так і прикладне значення.

Метою статті є порівняльний аналіз та експериментальна оцінка методів автоматичної класифікації тексту – від класичних алгоритмів машинного навчання до нейромережевих архітектур на основі трансформерів – з точки зору їх ефективності, продуктивності та практичної придатності для застосування у бібліографічних інформаційних системах. Для досягнення цієї мети вирішувались такі завдання: аналіз сучасних підходів до побудови БІС у контексті обробки текстових даних; дослідження та систематизація методів автоматичної класифікації тексту; реалізація та налаштування моделей класифікації; проведення експериментальних досліджень і порівняльної оцінки ефективності обраних методів; формування практичних рекомендацій щодо вибору методу залежно від вимог конкретного застосування.

Виклад основного матеріалу

1. Огляд бібліографічних інформаційних систем і класифікаційних схем

Сучасні БІС є спеціалізованими програмними комплексами, що забезпечують повний цикл роботи з науковою інформацією: каталогізацію та індексування публікацій, тематичну класифікацію, пошук і фільтрацію документів, управління посиланнями та інтеграцію із зовнішніми сервісами [4]. Серед найбільш відомих комерційних платформ – Scopus від Elsevier, що охоплює понад 25 тисяч рецензованих журналів з 27 тематичними категоріями та численними підкатегоріями [3]; Web of Science від Clarivate Analytics з понад 250 предметними категоріями та глибоким ретроспективним архівом [3]; Google Scholar, що охоплює широкий спектр академічних джерел, але не надає детальної класифікації за тематичними категоріями. Відкриті бібліотечні системи Koha та Evergreen підтримують класифікацію за УДК, ББК та Десятковою класифікацією Дьюї і широко використовуються бібліотеками по всьому світу [4].

Для систематизації документів у БІС застосовуються спеціалізовані класифікаційні схеми. Універсальна десяткова класифікація (УДК) є однією з найпоширеніших міжнародних систем, що використовується у понад 130 країнах і охоплює всі галузі знань, розподіляючи їх за десятковим принципом на десять основних класів від 0 до 9 [3]. УДК дозволяє створювати складні індекси шляхом комбінування кодів за допомогою спеціальних знаків, що забезпечує високу специфічність класифікації та можливість відображення міждисциплінарних зв'язків. Стандарти опису бібліографічних записів MARC (Machine-Readable Cataloging) та Dublin Core забезпечують уніфікований підхід до структурування метаданих. MARC містить понад 200 полів для опису різних аспектів документа і широко використовується в бібліотечних системах, тоді як Dublin Core є більш простим стандартом з 15 базовими елементами, що часто застосовується для цифрових ресурсів та електронних публікацій [2].

Ключовим елементом бібліографічного запису з погляду автоматичної класифікації є анотація або реферат, що містить стислий виклад основного змісту роботи обсягом від 100 до 300 слів. Анотація та ключові слова є найбільш інформативними джерелами для визначення тематичної приналежності публікації, оскільки вони концентрують основну семантику предметної галузі дослідження. Якісна анотація включає опис проблеми, що розглядається, методів дослідження, основних отриманих результатів та висновків, що робить її надзвичайно інформативною для задач автоматичної категоризації документів.

2. Огляд методів класифікації тексту

Традиційні підходи. Перші підходи до автоматичної класифікації текстів ґрунтувалися на системах правил та онтологіях предметної галузі. Класифікація здійснювалась шляхом зіставлення термінів з тексту з концепціями онтології та використання правил виведення. Такі системи потребують значних експертних зусиль для створення й підтримки актуальності, є погано масштабованими і демонструють обмежену здатність обробляти складні або неоднозначні випадки та тексти міждисциплінарного характеру [5].

Класичні методи машинного навчання. З розвитком методів машинного навчання з'явилася можливість автоматичного виявлення закономірностей у даних без явного програмування правил класифікації, що суттєво спростило процес створення систем категоризації. Наївний байєсівський класифікатор (Naive Bayes) ґрунтується на теоремі Байєса з припущенням про умовну незалежність ознак [5]. Для текстової класифікації часто використовується варіант Multinomial Naive Bayes, що моделює розподіл частот термінів у документах. Незважаючи на «наївне» припущення про незалежність, яке рідко виконується для реальних текстів, метод демонструє задовільні результати,

особливо при обмежених обсягах навчальних даних, та має низьку схильність до перенавчання. Метод опорних векторів (Support Vector Machines, SVM) будує оптимальну розділяючу гіперплощину у багатовимірному просторі ознак, максимізуючи відстань до найближчих точок різних класів [6]. SVM є ефективним при великій кількості ознак, що характерно для текстових даних, де розмірність простору може становити десятки тисяч. Ансамблевий метод Random Forest комбінує множину дерев рішень, навчених на різних підвибірках даних, і забезпечує стійкість до перенавчання, а також надає можливість оцінювання важливості ознак [5]. Порівняльні характеристики класичних методів за очікуваними значеннями F1 та іншими показниками наведено в Таблиці 1.

Таблиця 1

Порівняння класичних методів машинного навчання (за даними огляду літератури)

Метод	Переваги	Недоліки	Очікувана точність (F1)	Швидкість
Naive Bayes	Швидке навчання, малі дані	Припущення незалежності	0.75–0.85	Дуже висока
SVM	Висока точність, теоретична обґрунтованість	Повільніше навчання	0.82–0.90	Середня
Random Forest	Стійкість, оцінка ознак	Великий розмір моделі	0.80–0.88	Середня

Класичні методи машинного навчання вимагають ручного конструювання ознак: TF-IDF представлень, n-грам слів і символів, статистичних характеристик тексту та лінгвістичних ознак. TF-IDF (Term Frequency-Inverse Document Frequency) є базовим методом оцінювання важливості терміна: компонента TF вимірює частоту терміна в документі, а IDF знижує вагу термінів, що зустрічаються у багатьох документах [10]. Терміни з високими TF-IDF вагами є найбільш характерними для конкретного документа.

Методи глибокого навчання. Методи глибокого навчання автоматично виявляють складні ієрархічні представлення даних, що особливо корисно для обробки природної мови. Рекурентні нейронні мережі (RNN) здатні обробляти послідовності змінної довжини, зберігаючи інформацію про попередній контекст. Мережі типу LSTM (Long Short-Term Memory) вирішують проблему зникаючого градієнта завдяки спеціальній архітектурі з вентилями: вентиль забування, вхідний вентиль та вихідний вентиль контролюють потік інформації через комірку пам'яті [7]. Двонаправлені LSTM обробляють текст у обох напрямках, що покращує розуміння контексту. Згорткові нейронні мережі (CNN) для текстів застосовують фільтри різних розмірів для виявлення локальних паттернів – характерних n-грам, що індикують приналежність до певної категорії [7]; вони швидше навчаються порівняно з RNN завдяки можливості паралельних обчислень.

Архітектура трансформерів, запропонована у 2017 році, революціонізувала область обробки природної мови завдяки механізму самоуваги (self-attention), що дозволяє моделювати залежності між усіма елементами послідовності незалежно від відстані між ними [8]. Механізм уваги обчислює для кожного слова контекстуалізоване представлення, враховуючи всі інші слова з різними вагами. BERT (Bidirectional Encoder Representations from Transformers) є попередньо навченою моделлю на корпусах

Wikipedia та BookCorpus через Masked Language Modeling та Next Sentence Prediction [8]. Ключовою інновацією є одночасне врахування як лівого, так і правого контексту слова. Для конкретних задач класифікації BERT донавчається на спеціалізованих датасетах, що дозволяє досягти високої точності навіть при обмежених обсягах специфічних навчальних даних. Спеціалізована версія SciBERT, навчена на корпусі наукових публікацій, демонструє ще кращі результати для класифікації наукових текстів [15].

Багатомовні моделі (Multilingual BERT, XLM-RoBERTa) навчені на текстах багатьма мовами одночасно, що дозволяє використовувати їх для обробки текстів різними мовами без окремого навчання [9]. mBERT навчений на Wikipedia 104 мовами, XLM-RoBERTa – на корпусі CommonCrawl 100 мовами. Ці моделі виявляють здатність до міжмовного перенесення навчання: zero-shot класифікація може досягати 70–85% точності моделі, навченої на цільовій мові, а few-shot навчання – 85–95% точності [9]. Для виконання задачі класифікації для бібліографічних систем, що обслуговують переважно україномовну наукову літературу, ця властивість є особливо важливою.

3. Інструментальна база та датасети

Для реалізації системи обрано екосистему Python, що є домінуючою платформою для задач машинного навчання та обробки природної мови [5]. Бібліотека scikit-learn [12] надає реалізації класичних алгоритмів машинного навчання з уніфікованим інтерфейсом, зручним для порівняльних досліджень. PyTorch [13] використовується для реалізації нейромережевих моделей завдяки зручності налагодження та широкій підтримці спільноти. Бібліотека Transformers від Hugging Face [14] надає попередньо навчені трансформерні моделі та зручний інтерфейс для тонкого налаштування. Для обробки тексту використовуються NLTK та spaCy – дві провідні бібліотеки NLP для Python.

Для навчання та тестування моделей використовувалися датасети бібліографічних записів із публічних наукових баз даних. PubMed через MEDLINE API надає мільйони анотацій біомедичних публікацій з класифікацією за контрольованим словником MeSH. Semantic Scholar надає датасет з понад двохсот мільйонів наукових публікацій з різних дисциплін. ACM Digital Library та IEEE Xplore пропонують публікації комп'ютерних наук та інженерії з детальною класифікацією за ACM Computing Classification System та IEEE taxonomy відповідно. Дослідження показало, що розмічені датасети з анотаціями наукових публікацій є найбільш інформативним джерелом для навчання спеціалізованих класифікаційних моделей порівняно із загальними текстовими корпусами. Платформа Jupyter Notebook [17] використовувалась для документування експериментів та відтворюваності результатів. Датасети розбивалися у пропорції 70/15/15 на навчальну, валідаційну та тестову вибірки із забезпеченням рівномірного представлення класів у кожному розбитті.

4. Постановка задачі класифікації бібліографічних записів

Задача автоматичної класифікації бібліографічних записів полягає у визначенні тематичної категорії публікації на основі її метаданих, насамперед заголовку, анотації та ключових слів, відповідно до заздалегідь визначеної класифікаційної схеми. Формально: нехай задано множину бібліографічних записів $\boldsymbol{D} = \{d_1, d_2, ..., d_n\}$ та множину тематичних категорій $\boldsymbol{C} = \{c_1, c_2, ..., c_k\}$; потрібно побудувати функцію $f: D \rightarrow C$, що відображає кожен запис d_i у відповідну категорію c_j з максимальною точністю.

Специфіка бібліографічних текстів визначає особливі вимоги до системи класифікації. По-перше, короткий обсяг: анотація типово містить лише 100-300 слів, що суттєво обмежує кількість статистичних ознак. По-друге, висока термінологічна

насиченість: наукові тексти містять велику концентрацію спеціалізованих термінів, абревіатур та назв методів [10], які є ключовими індикаторами предметної галузі і мають бути збережені в незміненому вигляді при обробці. По-третє, структурованість: більшість наукових публікацій має структуру IMRAD (вступ, методи, результати, обговорення), де різні розділи мають різну інформативність для задачі класифікації. По-четверте, багатомовність: міжнародні бібліографічні системи містять публікації різними мовами, що потребує або підтримки мовозалежної обробки, або застосування мовонезалежних методів.

Для оцінювання якості класифікації застосовуються стандартні метрики [10]. Precision (Точність) визначає частку правильних серед усіх передбачених позитивів: *Precision = TP / (TP + FP)*. Recall (Повнота) – частку виявлених дійсних позитивів: *Recall = TP / (TP + FN)*. F1-Score є гармонійним середнім цих двох метрик: *F1 = 2 · (Precision · Recall) / (Precision + Recall)*, що забезпечує збалансовану оцінку якості. Accuracy вимірює загальну частку коректно класифікованих документів, але може бути оманливою при незбалансованих класах. Macro-F1 обчислює F1 для кожної категорії окремо та усереднює, даючи однакову вагу всім категоріям, що особливо важливо при нерівномірному розподілі документів за тематиками. Додатково оцінювались практичні характеристики: швидкість класифікації (кількість документів за секунду), вимоги до оперативної пам'яті та дискового простору для зберігання моделі.

5. Методи та реалізація системи класифікації

Попередня обробка текстів

Розроблена система реалізована у вигляді послідовного pipeline, де кожен етап обробки виконується у строгому порядку. Модуль попередньої обробки відповідає за трансформацію вхідних бібліографічних записів у формат, придатний для подальшого аналізу. Компонент парсингу забезпечує завантаження даних із форматів BibTeX, JSON та XML – найпоширеніших у бібліографічних системах. Парсер автоматично визначає тип вхідного файлу й екстрагує необхідні поля: назву публікації, анотацію, ключові слова та метадані авторів. Для кожного запису формується словник з нормалізованими назвами полів, що спрощує подальшу обробку незалежно від вихідного формату.

Компонент очищення тексту виконує видалення HTML-тегів та XML-розмітки, нормалізацію пробільних символів та зайвих розривів рядків, обробку математичних формул, грецьких літер і спеціальної нотації. Особливу увагу приділено збереженню структури складних термінів та абревіатур – для цього створено спеціальний словник термінів наукової літератури. Такі елементи або видаляються, або замінюються на текстові еквіваленти залежно від контексту, але структура термінів зберігається.

Нормалізація тексту включає лематизацію за допомогою бібліотеки spaCy з підтримкою морфологічно багатих мов [11] та pymorphy2 для морфологічного аналізу україномовних текстів. Лематизація зводить різні граматичні форми слова до єдиної канонічної форми, що суттєво скорочує словниковий запас і підвищує якість класифікації. Дослідження показують, що для класифікації текстів українською мовою лематизація дає на 2–5% вищу точність порівняно зі стемінгом завдяки врахуванню багатої морфологічної варіативності флективної мови [11]. Додатково виконується видалення стоп-слів, приведення до нижнього регістру, обробка цифр та нормалізація спеціальних символів.

Методи векторизації

Для класичних методів застосовується TF-IDF векторизація з урахуванням уніграм та біграм (діапазон n-грам 1–2). Використання біграм дозволяє захопити контекстні зв'язки між словами, що покращує якість класифікації: наприклад, «машинне навчання» несе більше семантичного навантаження, ніж два окремих слова. Застосування

сублінійного масштабування частот зменшує вплив дуже частих термінів та робить представлення більш стабільним. Для нейромережевих методів вкладений шар ініціалізується попередньо навченими векторами Word2Vec з розмірністю 300 вимірів. Word2Vec навчається передбачати слово за контекстом (CBOW) або контекст за словом (Skip-gram), формуючи векторні представлення, що захоплюють семантичні та синтаксичні зв'язки між словами. Семантично близькі слова мають близькі вектори, що можна вимірювати косинусною подібністю.

Для BERT застосовується власний підхід до токенізації: WordPiece токенізатор розбиває слова на субслівні одиниці, що дозволяє ефективно обробляти нові слова та складну термінологію. BERT генерує контекстуалізовані ембедінги довжиною 768 вимірів для кожного токена, враховуючи весь контекст речення. На відміну від статичних ембедингів, де кожне слово має фіксоване представлення, контекстуалізовані ембедінги трансформерів створюють різні вектори для одного слова у різних контекстах, що є надзвичайно важливим для наукових текстів із багатозначними термінами.

Реалізація LSTM та CNN класифікаторів

Архітектура на основі LSTM побудована з використанням двошарової двонаправленої мережі. Перший шар LSTM має розмірність прихованого стану 256 та обробляє текстову послідовність у прямому та зворотному напрямках одночасно, формуючи 512-вимірне представлення кожного кроку. Другий шар LSTM агрегує контекстні зв'язки вищого рівня абстракції. Вкладений шар ініціалізується попередньо навченими векторами Word2Vec з розмірністю 300 вимірів, що забезпечує краще початкове представлення слів порівняно з випадковою ініціалізацією. Dropout з ймовірністю 0.3 між LSTM шарами та 0.5 перед фінальним класифікатором запобігають перенавчанню. Глобальне усереднення виходів LSTM дозволяє сформувати фіксований вектор представлення для послідовностей довільної довжини.

Згорткова нейронна мережа для текстів застосовує фільтри розмірів 3, 4 та 5, що дозволяє виявляти паттерни від триграм до п'ятиграм – типовий діапазон для наукової термінології. Кожен тип фільтра представлений 128 окремими фільтрами, що забезпечує достатню виразність для захоплення різноманітних текстових паттернів. Після згорткових шарів застосовується операція max-pooling, що виділяє найбільш значущу ознаку з кожної карти активації. Всього нейронна мережа отримує $128 \times 3 = 384$ ознаки, що конкатенуються перед фінальним класифікаційним шаром. Нормалізація пакету після згорткових шарів прискорює навчання та покращує стабільність моделі.

Архітектура системи

Загальна архітектура системи реалізована у вигляді модульного pipeline з чотирьох компонентів: модуль попередньої обробки даних, модуль векторизації та створення ознак, модуль класифікації, модуль оцінки результатів. Такий підхід забезпечує гнучкість у виборі методів класифікації та можливість легкого розширення функціоналу. Система реалізована мовою Python з використанням scikit-learn для класичних методів машинного навчання [12], PyTorch для глибокого навчання [13] та FastAPI для створення веб-інтерфейсу. Вхідними даними є бібліографічні записи у форматі JSON або XML, що містять метадані публікацій; на виході система надає передбачену категорію з відповідним рівнем впевненості.

На основі FastAPI розроблено RESTful API, що підтримує синхронну та асинхронну обробку запитів. Основний маршрут (endpoint) приймає бібліографічний запис у форматі JSON та повертає передбачену категорію з рівнем впевненості. Додатково реалізовано маршрут для пакетної обробки множини документів з підтримкою прогресу виконання. Для обробки великих обсягів даних реалізовано систему пакетної класифікації з використанням черг завдань на основі Celery, що

автоматично розподіляє навантаження між доступними обчислювальними ресурсами та забезпечує відновлення після збоїв. Веб-інтерфейс розроблено з використанням React, що забезпечує зручний доступ до функціоналу класифікації: користувачі можуть завантажувати як окремі документи, так і цілі файли з бібліографічними записами для автоматичної категоризації.

Модуль класифікації представлений набором реалізацій різних алгоритмів: від традиційних методів машинного навчання до сучасних архітектур глибокого навчання. Кожен метод реалізовано як окремий клас з уніфікованим інтерфейсом, що дозволяє легко змінювати та комбінувати різні підходи. Для класичних методів створено класи-обгортки, що інкапсулюють логіку векторизації та навчання моделей. Реалізовано можливість тонкого налаштування гіперпараметрів через конфігураційні файли, що спрощує процес експериментування з різними налаштуваннями.

6. Результати порівняльного оцінювання

Результати класичних методів машинного навчання

Результати порівняльного тестування класичних методів наведено в Таблиці 2. SVM досягла найвищої якості серед класичних алгоритмів із показником F1-Score 0.8847, що на 7% краще за Naive Bayes та на 4% краще за Random Forest. Висока ефективність SVM пояснюється ефективністю при роботі з розрідженими TF-IDF матрицями великої розмірності – саме в такому просторі ознак представлено бібліографічні тексти. При цьому Naive Bayes виявилась найшвидшою у навчанні (0.52 сек) та інференсі (0.002 сек на зразок), що робить її оптимальним вибором для швидкого прототипування чи систем із жорсткими обмеженнями за часом відгуку. Random Forest, попри тривалий час навчання (12.45 сек), забезпечує збалансовану точність та унікальну можливість аналізу важливості ознак для пояснення рішень класифікатора.

Таблиця 2

Метрики якості класичних методів машинного навчання

Модель	Accuracy	Precision	Recall	F1-Score	Час навчання	Час інференсу (1 зразок)
Naive Bayes	0.8247	0.8156	0.8247	0.8135	0.52 сек	0.002 сек
SVM	**0.8921**	**0.8854**	**0.8921**	**0.8847**	3.18 сек	0.005 сек
Random Forest	0.8536	0.8472	0.8536	0.8468	12.45 сек	0.010 сек

Аналіз швидкодії демонструє значні відмінності між моделями: Naive Bayes обробляє 487 зразків за секунду – найшвидший результат серед усіх класичних методів; SVM забезпечує 198.5 зр/с, залишаючись прийнятною для більшості практичних задач реального часу. Random Forest є найповільнішою (96.2 зр/с) через необхідність агрегації прогнозів від 200 дерев рішень. Вимоги до RAM також корелюють зі складністю моделі: Naive Bayes потребує лише 150 MB, тоді як Random Forest вимагає 320 MB для зберігання ансамблю дерев.

Результати методів глибокого навчання

Моделі глибокого навчання демонструють вищу точність порівняно з класичними підходами (Таблиця 3). BERT досягає найкращого результату (F1-Score 0.9124) завдяки попередньому навчанню на величезних текстових корпусах, однак вимагає 18.5 годин навчання через 110 мільйонів параметрів, а також GPU та 8 GB оперативної пам'яті. CNN показує оптимальний баланс між точністю (F1-Score 0.8761) та швидкістю навчання (3.2 год), використовуючи паралельні згортки для ефективного виділення ознак без великих вимог до обчислювальних ресурсів. LSTM, маючи 2.1M параметрів, забезпечує проміжну точність (F1-Score 0.8638) та враховує послідовний характер текстових даних через двонаправлену архітектуру.

Таблиця 3

Метрики якості методів глибокого навчання

Модель	Accuracy	Precision	Recall	F1-Score	Час навчання (5 епох)	Кількість параметрів
LSTM	0.8674	0.8621	0.8674	0.8638	4.8 год	~2.1M
CNN	0.8792	0.8745	0.8792	0.8761	3.2 год	~1.8M
BERT*	**0.9156**	**0.9102**	**0.9156**	**0.9124**	18.5 год	~110M

Примітка: Модель bert-base-multilingual-cased з тонким налаштуванням

Швидкість виводу суттєво залежить від наявності GPU-прискорення. На CPU найшвидшою є CNN (0.062 сек на зразок), що в 1.4 рази швидше за LSTM та в 4 рази швидше за BERT. Використання GPU радикально покращує продуктивність: CNN обробляє зразок за 0.008 сек (майже в 8 разів швидше), LSTM – за 0.012 сек (у 7 разів швидше). Пакетна обробка (batch size 32) значно ефективніша: на GPU CNN обробляє 32 зразки за 0.09 сек, що еквівалентно 355 зразкам за секунду порівняно з лише 36 зр/с на CPU.

7. Комплексне порівняння всіх шести моделей

У Таблиці 4 наведено зведені показники всіх досліджуваних моделей за сукупністю критеріїв якості та ресурсних вимог.

Комплексний аналіз виявляє чіткі компроміси між точністю та ефективністю. BERT лідирує за точністю (F1-Score 0.912), але є найповільнішою (4.1 зр/с) та найресурсомісткішою (3500 МБ RAM). SVM забезпечує найкращий баланс із високою точністю (0.885) та прийнятною швидкістю (199 зр/с), що робить її оптимальним вибором для виробничих систем реального часу. Naive Bayes є найшвидшою (487 зр/с) та найекономнішою (150 МБ), ідеальною для обмежених ресурсів. Складність налаштування та інтерпретованість обернено корелюють із потужністю моделей: прості моделі легше налаштовувати та пояснювати кінцевому користувачеві.

Важливим аспектом аналізу є поведінка моделей на незбалансованому датасеті, де деякі тематичні категорії представлені значно більшою кількістю документів, ніж інші. Метрика Macro-F1, що надає однакову вагу всім категоріям незалежно від їх розміру, виявилась дещо нижчою за Micro-F1 для всіх моделей, що свідчить про відносно гірше опрацювання малочисельних категорій. Найбільш критичною ця проблема виявилась для Naive Bayes, де різниця між Macro- та Micro-F1 сягала 3.5%. SVM продемонструвала найрівномірніший розподіл якості по категоріях завдяки механізму

регуляризації. BERT показав найкращі результати і для малих, і для великих категорій, що свідчить про загальнішу здатність до узагальнення.

Таблиця 4

Підсумкове порівняння всіх методів класифікації

Критерій	Naive Bayes	SVM	Random Forest	LSTM	CNN	BERT
F1-Score	0.814	0.885	0.847	0.864	0.876	**0.912**
Швидкість (зр/с)	**487**	199	96	11.5	16.1	4.1
Час навчання	0.5 с	3.2 с	12.5 с	288 с	192 с	1110 с
Використання RAM	**150 МБ**	180 МБ	320 МБ	800 МБ	600 МБ	3500 МБ
Розмір моделі	2 МБ	5 МБ	28 МБ	25 МБ	18 МБ	450 МБ
Складність налаштування	Низька	Середня	Середня	Висока	Висока	Дуже висока
Інтерпретованість	Висока	Середня	Висока	Низька	Низька	Дуже низька

Аналіз похибок класифікації виявив, що найчастіше неправильна категоризація трапляється для міждисциплінарних публікацій, що однаково відносяться до кількох тематичних областей. Наприклад, публікації на перетині комп'ютерних наук та біоінформатики нерідко відносились то до однієї, то до іншої категорії залежно від переважаючої термінології. BERT виявляє меншу частку таких помилок (близько 8%) порівняно із SVM (15%) та Naive Bayes (22%), що свідчить про кращу здатність трансформерів враховувати семантичний контекст.

8. Наукова новизна та практичні рекомендації

Наукова цінність проведеного дослідження визначається кількома аспектами.

По-перше, систематизовано структурні особливості бібліографічних текстів з позиції вимог до автоматизованої обробки в інформаційних системах. Доведено, що комбіноване використання заголовку, анотації та ключових слів дає на 8-12% кращі результати порівняно з використанням лише одного поля бібліографічного запису. Це підтверджує важливість урахування структури запису при побудові системи класифікації та є важливим орієнтиром для практиків.

По-друге, здійснено комплексний порівняльний аналіз шести методів із урахуванням не лише точності, а й практичних характеристик – швидкодії, ресурсоємності та складності налаштування й розгортання. Більшість наявних у літературі порівняльних досліджень [5] не враховує практичних аспектів розгортання систем у виробничих умовах, обмежуючись лише метриками точності.

По-третє, підтверджено та кількісно оцінено ефективність лематизації для морфологічно багатої української мови: приріст точності на 2-5% порівняно зі стемінгом є статистично значущим для задачі класифікації наукових текстів [11]. Це є цінним практичним результатом для розробників україномовних систем.

По-четверте, продемонстровано потенціал багатомовних трансформерних моделей для роботи з україномовними та іншими слаборесурсними мовами через

механізм крос-лінгвістичного перенесення знань [9]. Zero-shot класифікація дозволяє досягати прийнятної якості навіть без специфічних навчальних даних для цільової мови, що особливо цінно для невеликих бібліографічних систем.

Практичні рекомендації щодо вибору методу можна сформулювати таким чином. Для виробничих систем реального часу з помірними ресурсами рекомендується SVM (F1-Score 0.885, 199 зр/с, 180 МБ RAM) – оптимальне поєднання точності та продуктивності. Для задач, де першочерговим є максимальна точність та доступний GPU, рекомендується BERT (F1-Score 0.912); якщо GPU відсутній, хорошим компромісом є CNN (F1-Score 0.876, 16.1 зр/с). Для задач швидкого прототипування, де важлива передусім швидкість розробки, достатньо Naive Bayes (F1-Score 0.814, 0.5 сек навчання). Для систем, де важлива інтерпретованість рішень, рекомендується Naive Bayes або Random Forest, що надають зрозумілі пояснення через ймовірності та важливість ознак. Для міжнародних бібліографічних систем із підтримкою множини мов рекомендується XLM-RoBERTa, ефективна для крос-лінгвістичної класифікації [9].

Висновки

У статті проведено комплексний порівняльний аналіз методів автоматичної класифікації тексту для задачі тематичної категоризації бібліографічних записів. Розроблено та апробовано модульну систему класифікації мовою Python, що охоплює повний цикл обробки: завантаження та парсинг записів у форматах JSON, XML та BibTeX, багатоетапне очищення і нормалізацію тексту з урахуванням специфіки наукових публікацій, векторизацію та навчання моделей, оцінку результатів за стандартними метриками якості. Отримані результати дозволяють зробити такі висновки.

Порівняльне оцінювання шести методів підтвердило суттєвий компроміс між точністю класифікації та обчислювальними вимогами. BERT досягає найвищої точності (F1-Score = 0.912), але потребує значних ресурсів. SVM забезпечує найкращий баланс для виробничих систем (F1-Score = 0.885, 199 зр/с, 180 МБ RAM). Naive Bayes є оптимальною для систем з обмеженими ресурсами та вимогами до швидкого прототипування (F1-Score = 0.814, 487 зр/с).

Встановлено, що поєднання заголовку, анотації та ключових слів бібліографічного запису дає на 8–12% кращі результати порівняно з обробкою лише одного поля. Для морфологічно багатих мов, зокрема української, лематизація із застосуванням pymorphy2 забезпечує приріст точності на 2–5% порівняно зі стемінгом.

Практичні рекомендації щодо вибору методу класифікації, сформульовані за результатами дослідження, дозволяють розробникам бібліографічних систем обґрунтовано обирати підхід відповідно до конкретних вимог до точності, швидкодії, ресурсоємності та інтерпретованості результатів.

Слід відзначити перспективи подальших досліджень у цьому напрямку. Зокрема, доцільним є дослідження ансамблевих підходів, що комбінують переваги класичних та нейромережевих методів: наприклад, використання BERT-ембедингів як вхідних ознак для SVM може забезпечити кращий результат, ніж кожен метод окремо. Актуальним є також дослідження активного навчання (active learning) для зменшення вимог до розміченого датасету та методів дистиляції знань для стиснення великих моделей типу BERT до більш компактних аналогів із незначною втратою точності. Важливим напрямком є також врахування динамічного характеру наукового словника – появи нових термінів та понять – через механізми адаптивного оновлення моделей.

Отримані результати підтверджують доцільність впровадження методів автоматичної класифікації тексту для підвищення ефективності обробки бібліографічної інформації та можуть бути використані як основа для подальших

досліджень у сфері інтелектуальних інформаційних систем.

Список використаної літератури

1. Бахтурин С. В. Інформаційні системи наукових бібліотек: сучасний стан та перспективи розвитку / С. В. Бахтурин // Вісник Книжкової палати. – 2019. – № 5. – С. 12-18.
2. ДСТУ ГОСТ 7.1:2006. Бібліографічний запис. Бібліографічний опис [Текст]. – К.: Держспоживстандарт України, 2007. – 47 с.
3. Scopus Content Coverage Guide [Електронний ресурс] // Elsevier. – Режим доступу: https://www.elsevier.com/solutions/scopus. – Назва з екрану.
4. Breeding M. Library Services Platforms: A Maturing Genre of Products / M. Breeding // Library Technology Reports. – 2015. – Vol. 51, No. 4. – P. 5-38.
5. Kowsari K. Text Classification Algorithms: A Survey / K. Kowsari et al. // Information. – 2019. – Vol. 10, No. 4. – P. 150.
6. Joachims T. Text Categorization with Support Vector Machines / T. Joachims // Proceedings of ECML. – 1998. – P. 137-142.
7. Chollet F. Deep Learning with Python / F. Chollet. – Manning Publications, 2021. – 504 p.
8. Devlin J. BERT: Pre-training of Deep Bidirectional Transformers / J. Devlin et al. // Proceedings of NAACL-HLT. – 2019. – P. 4171-4186.
9. Conneau A. Unsupervised Cross-lingual Representation Learning at Scale / A. Conneau et al. // Proceedings of ACL. – 2020. – P. 8440-8451.
10. Manning C. D. Introduction to Information Retrieval / C. D. Manning, P. Raghavan, H. Schütze. – Cambridge University Press, 2008. – 506 p.
11. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian / M. Korobov // Analysis of Images, Social Networks and Texts. – 2015. – P. 320-332.
12. Pedregosa F. Scikit-learn: Machine Learning in Python / F. Pedregosa et al. // Journal of Machine Learning Research. – 2011. – Vol. 12. – P. 2825-2830.
13. Paszke A. PyTorch: An imperative style, high-performance deep learning library / A. Paszke et al. // Advances in Neural Information Processing Systems. – 2019. – Vol. 32. – P. 8024-8035.
14. Wolf T. Transformers: State-of-the-art natural language processing / T. Wolf et al. // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. – 2020. – P. 38-45.
15. Beltagy I. SciBERT: A Pretrained Language Model for Scientific Text / I. Beltagy, K. Lo, A. Cohan // Proceedings of EMNLP-IJCNLP. – 2019. – P. 3615-3620.
16. Gusenbauer M. Google Scholar to overshadow them all? / M. Gusenbauer // Scientometrics. – 2019. – Vol. 118, No. 1. – P. 177-214.
17. Kluyver T. Jupyter Notebooks – a publishing format for reproducible computational workflows / T. Kluyver et al. // Positioning and Power in Academic Publishing. – 2016. – P. 87-90.

References:

1. Bakhturyn S. V. (2019) Information systems of scientific libraries: current state and development prospects. Bulletin of the Book Chamber, No. 5, pp. 12-18. (in Ukr.)
2. DSTU GOST 7.1:2006. Bibliographic record. Bibliographic description. Kyiv: Derzhspozhyvstandart Ukrainy, 2007. 47 p. (in Ukr.)
3. Scopus Content Coverage Guide [Electronic resource] // Elsevier. Available at: https://www.elsevier.com/solutions/scopus.
4. Breeding M. (2015) Library Services Platforms: A Maturing Genre of Products. Library Technology Reports, Vol. 51, No. 4, pp. 5-38.
5. Kowsari K. et al. (2019) Text Classification Algorithms: A Survey. Information, Vol. 10, No. 4, p. 150.
6. Joachims T. (1998) Text Categorization with Support Vector Machines. Proceedings of ECML, pp. 137-142.
7. Chollet F. (2021) Deep Learning with Python. Manning Publications. 504 p.
8. Devlin J. et al. (2019) BERT: Pre-training of Deep Bidirectional Transformers. Proceedings of NAACL-HLT, pp. 4171-4186.
9. Conneau A. et al. (2020) Unsupervised Cross-lingual Representation Learning at Scale. Proceedings of ACL, pp. 8440-8451.
10. Manning C. D., Raghavan P., Schütze H. (2008) Introduction to Information Retrieval. Cambridge University Press. 506 p.
11. Korobov M. (2015) Morphological Analyzer and Generator for Russian and Ukrainian. Analysis of Images, Social Networks and Texts, pp. 320-332.

12. Pedregosa F. et al. (2011) Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, Vol. 12, pp. 2825-2830.
13. Paszke A. et al. (2019) PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, Vol. 32, pp. 8024-8035.
14. Wolf T. et al. (2020) Transformers: State-of-the-art natural language processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38-45.
15. Beltagy I., Lo K., Cohan A. (2019) SciBERT: A Pretrained Language Model for Scientific Text. Proceedings of EMNLP-IJCNLP, pp. 3615-3620.
16. Gusenbauer M. (2019) Google Scholar to overshadow them all? Scientometrics, Vol. 118, No. 1, pp. 177-214.
17. Kluyver T. et al. (2016) Jupyter Notebooks – a publishing format for reproducible computational workflows. Positioning and Power in Academic Publishing, pp. 87-90.

KRASNOSHLYK Nataliya,
Candidate of Technical Sciences, Associate Professor, Department of Applied Mathematics and Informatics, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine
BOHATYRENKO Pavlo,
Student, Department of Applied Mathematics and Informatics, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

COMPARATIVE ANALYSIS OF AUTOMATIC TEXT CLASSIFICATION METHODS IN BIBLIOGRAPHIC INFORMATION SYSTEMS

Summary. Introduction. *The rapid growth of digital technologies and the increasing intensity of scientific communication result in a constant expansion of textual data volumes accumulated in bibliographic information systems (BIS), electronic repositories, and scientometric databases. Traditional cataloguing approaches based on manual or semi-automatic indexing no longer meet the scalability and timeliness requirements of modern information environments. An experienced cataloguer can process only 20–30 documents per day, which is wholly insufficient given the continuous and rapid growth of scientific output. Research shows that accurate thematic classification can improve the relevance of search results by 30–40% compared to systems relying solely on full-text search. Despite a substantial body of work on text classification algorithms, systematic comparative studies specifically addressing the thematic categorisation of bibliographic records — with consideration of performance and resource requirements — remain relatively scarce in the literature.*

Purpose. *The aim of this article is a comparative analysis and experimental evaluation of automatic text classification methods — from classical machine learning algorithms to neural network architectures based on transformers — with respect to their effectiveness, performance, and practical suitability for use in bibliographic information systems.*

Results. *A modular classification system for bibliographic records was developed in Python using the scikit-learn, PyTorch, and FastAPI frameworks. Six models were evaluated using accuracy metrics (Accuracy, Precision, Recall, F1-Score), processing speed, and resource consumption. Among classical machine learning methods, Support Vector Machines (SVM) achieved the highest quality (F1-Score = 0.885) while Naive Bayes demonstrated the fastest processing speed (487 samples/sec). Among deep learning methods, BERT achieved the best classification quality (F1-Score = 0.912) but requires substantial computational resources (18.5 hours of training, GPU, 8 GB RAM). CNN provides a good balance between accuracy (F1-Score = 0.876) and training speed (3.2 hours). It was established that combining title, abstract, and keywords yields 8–12% better results than using any single field. For the morphologically rich Ukrainian language, lemmatisation using pymorphy2 provides a 2–5% accuracy improvement over stemming.*

Conclusion. *BERT achieves the highest classification quality (F1-Score = 0.912) but demands significant resources. SVM provides the best balance for production real-time systems (F1-Score = 0.885, 199 samples/sec, 180 MB RAM). Naive Bayes is optimal for resource-constrained environments and rapid prototyping. Practical recommendations for choosing a classification method according to specific requirements for accuracy, speed, resource consumption, and interpretability are formulated. Multilingual transformer models (XLM-RoBERTa) are recommended for international systems supporting multiple languages due to their cross-lingual transfer capabilities.*

Keywords: *automatic text classification, bibliographic information systems, machine learning, BERT, support vector machines, deep learning, natural language processing, text vectorisation.*

Одержано редакцією 15.11.2025 р.
Прийнято до публікації 17.12.2025 р.

УДК 004.415.2

DOI 10.31651/2076-5886-2025-1-85-95

PACS 07.05.Tp, 89.20.Ff

ТКАЧЕНКО Олексій Олексійович
студент спеціальності «Інформаційні системи та технології» Черкаського національного університету імені Богдана Хмельницького

ДІДКОВСЬКИЙ Руслан Михайлович,
доктор технічних наук, доцент, доцент кафедри прикладної математики та інформатики Черкаського національного університету імені Богдана Хмельницького
e-mail: didkovskyirm@vu.cdu.edu.ua
ORCID 0000-0002-5166-7564

ХОВАЙБА Дарина Євгенівна
викладач кафедри прикладної математики та інформатики Черкаського національного університету імені Богдана Хмельницького
e-mail:
tovstopiat.daryna1618@vu.cdu.edu.ua
ORCID 0009-0001-4202-0451

ДОСЛІДЖЕННЯ МІКРОФРОНТЕНД АРХІТЕКТУРИ ДЛЯ ПОБУДОВИ МАСШТАБОВАНИХ ВЕБ-СИСТЕМ

У роботі досліджено підходи до проектування та реалізації масштабованих веб-систем на основі мікрофронтенд архітектури з використанням технологій Module Federation та інструментарію Nx. Виконано порівняльний аналіз методів інтеграції мікрофронтендів: інтеграції на етапі збірки, ізоляції через iframe та інтеграції на етапі виконання. Обґрунтовано вибір патерну Shell-Remote як основи для проектування розподіленої системи управління задачами. Розроблено методику організації монорепозиторію з чітким поділом на шари (Applications, Feature Libraries, Shared UI, Data Access), що забезпечує повторне використання коду та уникнення дублювання. Проведено експериментальне дослідження ефективності запропонованого підходу: встановлено, що використання бандлера Rspack скорочує час збірки більш ніж у 10 разів порівняно із Webpack, а застосування стратегії ледачого завантаження (Lazy Loading) зменшує обсяг початкового завантаження сторінки на 81%. Отримані результати підтверджують доцільність застосування досліджуваного архітектурного підходу для побудови корпоративних (Enterprise-рівня) веб-систем.

Ключові слова*: мікрофронтенд архітектура, Module Federation, монорепозиторій, Nx, Rspack, Shell-Remote, масштабованість, продуктивність збірки.*