DOI: https://doi.org/10.36910/6775-2524-0560-2026-64-23
УДК 004.89:004.91:81'322
Никитюк В'ячеслав В'ячеславович, к.т.н., доц.
https://orcid.org/0009-0003-4230-0183
Долінський Андрій Михайлович, аспірант
https://orcid.org/0009-0003-4230-0183
Тернопільський національний технічний університет ім. Івана Пулюя, м. Тернопіль, Україна

МОДЕЛЮВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ВИЯВЛЕННЯ ГРОМАДСЬКОЇ ДУМКИ НА ОСНОВІ МЕТОДУ ДОМЕННО-ОРІЄНТОВАНИХ СЕМАНТИЧНИХ СТРУКТУР DO-SPO

Никитюк В. В., Долінський А. М. Моделювання інтелектуальної системи виявлення громадської думки на основі методу доменно-орієнтованих семантичних структур DO-SPO. Статтю присвячено розв'язанню задачі оптимізації машинного навчання для систем виявлення громадської думки в умовах дефіциту маркованих корпусів даних для української мови. Авторами розроблено метод доменно-орієнтованого синтезу даних (DO-SPO), який базується на ієрархічному структуруванні семантичних трійок у межах 19 стратегічних доменів, що дозволяє подолати ефекти «семантичної інверсії» та синтаксичного перенавчання моделей. Експериментальна верифікація з використанням архітектури Bi-LSTM на корпусі обсягом 200 000 одиниць продемонструвала значний приріст ефективності: точність тематичного розпізнавання зросла до 98%, а аналізу тональності — до 95%. Впровадження методу забезпечило повне усунення помилок полярності сентименту та високу стійкість класифікаторів до інформаційного шуму, що підтверджує готовність системи до практичної експлуатації у конвеєрах моніторингу суспільно-політичного дискурсу.
Ключові слова: Виявлення громадської думки, Метод DO-SPO, Генерація синтетичних даних, Архітектура Bi-LSTM, Полярність сентименту, Український NLP.

Nykytyuk V., Dolinskyi A. Modeling of an intelligent public opinion detection system based on the do-spo domain-oriented semantic structures method. The article focuses on solving the machine learning optimization problem for public opinion detection systems under conditions of limited labeled datasets for the Ukrainian language. The authors have developed a Domain-Oriented Synthetic Data Generation method (DO-SPO), which is based on the hierarchical structuring of semantic triples across 19 strategic domains. This approach allows for overcoming the effects of 'semantic inversion' and syntactic overfitting of models. Experimental verification using a Bi-LSTM architecture on a corpus of 200,000 units demonstrated a significant performance gain: the accuracy of thematic recognition increased to 98%, and sentiment analysis accuracy reached 95%. The implementation of this method ensured the complete elimination of sentiment polarity errors and high classifier robustness against informational noise, confirming the system's readiness for practical deployment in socio-political discourse monitoring pipelines.
Keywords: Public opinion detection, DO-SPO method, Synthetic data generation, Bi-LSTM architecture, Sentiment polarity, Ukrainian NLP.

Постановка наукової проблеми. Швидка цифровізація суспільних процесів та зростання інтенсивності інформаційних потоків у соціальних медіа висувають нові вимоги до систем державного моніторингу. В умовах гібридних загроз та динамічних змін у суспільних настроях, автоматизоване виявлення соціо-наукової громадської думки стає критичним інструментом для забезпечення національної безпеки та ефективного стратегічного планування. Проте розвиток інтелектуальних систем обробки природної мови NLP для українського сегмента цифрового простору стикається з фундаментальною перешкодою — гострим дефіцитом збалансованих, попередньо анотованих та семантично насичених корпусів даних.

Наукова проблема полягає у так званому «ефекті холодного старту»: існуючі архітектури глибокого навчання (зокрема трансформери та рекурентні мережі) потребують значних обсягів маркованих даних для досягнення прийнятної точності. Для української мови, яка є лінгвістично специфічною та ресурсно обмеженою у сфері NLP, створення таких масивів шляхом ручної анотації є економічно недоцільним та занадто тривалим процесом. Спроби використання базового синтезу даних без жорсткого семантичного контролю призводять до виникнення дефектів «синтаксичного мімікрування», коли моделі заучують шаблони замість змісту, що спричиняє критичні помилки в аналізі тональності та тематики.

Вирішення цієї проблеми має безпосередній зв'язок із пріоритетними науковими та практичними завданнями щодо розвитку вітчизняних систем штучного інтелекту. Оптимізація методів навчання нейромережевих моделей через розробку доменно-орієнтованих структур (таких як метод DO-SPO) дозволяє забезпечити високу точність моніторингу стратегічних доменів (військова медицина, ветеранська політика, цифрова трансформація) навіть за повної відсутності реальних навчальних прикладів. Таким чином, дослідження безпосередньо спрямоване на

підвищення стійкості інформаційного простору та створення надійного аналітичного інструментарію для виявлення тенденцій громадської думки в інтересах державного управління.

Аналіз досліджень. За останні роки сфера обробки природної мови NLP змістила акцент з простого накопичення великих даних на забезпечення їхньої якості та семантичної точності. Особливо гостро це питання стоїть для мов із середнім рівнем ресурсів, до яких належить українська. Сучасні дослідження підтверджують, що використання великих мовних моделей LLM для генерації синтетичних корпусів дозволяє ефективно долати проблему «холодного старту» [1-3], зокрема й при формуванні спеціалізованих україномовних масивів для навчання класифікаторів текстів [4], проте вихідні дані часто залишаються неконтрольованими [5]. Це призводить до появи «семантичного шуму» — ситуацій, коли генерований текст формально відповідає заданим параметрам, але не містить достатньої кількості диференціюючих ознак для коректного навчання класифікаторів. В академічній спільноті дедалі частіше обговорюється ризик виникнення ефекту, коли нейромережеві моделі демонструють високу продуктивність на синтетичних валідаційних вибірках, але втрачають стабільність на реальних даних через «завчування» специфічних синтаксичних паттернів генератора замість глибоких семантичних зв'язків.

Розвиток українського NLP сегменту отримав новий імпульс завдяки появі спеціалізованих корпусів, таких як COSMUS та UNLP 2025, що фокусуються на розмаїтті комунікації в соціальних мережах [6, 7]. Проте питання керованого синтезу для специфічних доменних областей залишається відкритим. Більшість існуючих рішень для класифікації все ще стикаються з проблемою «семантичного злиття» суміжних категорій, де без жорсткої тематичної ізоляції під час генерації значна частина повідомлень ідентифікується моделлю як змішана або потрапляє до класу некласифікованих даних. Попри прогрес у методах аугментації, актуальною залишається потреба у створенні підходів, які б дозволяли ізолювати семантичне ядро теми від загального контексту та забезпечували стабільність навчання класифікаторів через механізми регуляризації.

Дана робота базується на раніше розробленому авторами алгоритмі керованого синтезу й ієрархічного структурування україномовних корпусів на основі S-P-O матриць [8]. Якщо на попередніх етапах дослідження основну увагу було приділено логічній структурі та математичному обґрунтуванню самого процесу генерації, то в межах цієї статті розглядається наступний етап — практична реалізація нейромережевих моделей на базі архітектур LSTM та їх навчання на синтезованих масивах [9, 10]. Процес експериментального навчання виявив низку критичних викликів, пов'язаних із перенавчанням моделей на шаблонних конструкціях та недостатньою роздільною здатністю тематичної класифікації. Саме ці спостереження стали емпіричною базою для перегляду стратегії синтезу та впровадження концепції «стратегічних тем», що дозволяє забезпечити високу семантичну дистанцію між класами та підвищити реальну точність роботи системи.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Розвиток інтелектуальних систем обробки природної мови (NLP) для українського сегмента цифрового простору стикається з проблемою обмеженості доступних, збалансованих та попередньо анотованих корпусів даних. Необхідність оперативного моніторингу інформаційних потоків та автоматизованого аналізу громадської думки потребує розробки методів, які дозволяють нівелювати дефіцит навчальних масивів без втрати семантичної точності.

На попередніх етапах дослідження було запропоновано та теоретично обґрунтовано алгоритм керованого синтезу україномовних текстів на основі матричної структури «Суб'єкт-Предикат-Об'єкт» (S-P-O). Зазначений підхід дозволив автоматизувати процес формування репрезентативних корпусів даних, що імітують синтаксичні та семантичні особливості живої комунікації в соціальних мережах та медіа-ресурсах. Обґрунтування ієрархічного структурування таких масивів створило фундамент для вирішення проблеми «холодного старту» при розробці класифікаційних систем.

Проте перехід від теоретичної моделі синтезу до практичної експлуатації інтелектуальних систем аналізу контенту потребує детальної апробації отриманих корпусів на сучасних архітектурах глибокого навчання. Об'єктивним кроком розвитку дослідження є перехід до етапу навчання нейромережевих моделей, зокрема рекурентних мереж типу LSTM. Що дозволить оцінити здатність алгоритмічно згенерованих даних забезпечувати стабільність навчання класифікаторів, а також

виявити потенційні ризики семантичних спотворень або синтаксичного перенавчання моделей в умовах реальної обробки інформації. Таким чином, результати попередніх розробок слугують базисом для емпіричної перевірки ефективності синтетичних корпусів у задачах визначення тональності та тематичного спрямування текстових повідомлень.

Дослідження на початковому етапі ґрунтувалася на використанні синтетичного корпусу даних першої ітерації, сформованого за алгоритмом S-P-O синтезу. Вихідний масив обсягом 100 000 одиниць зберігався у форматі JSONL, де кожен об'єкт містив ідентифікатор класу, тематичну мітку та текстовий блок. Характерною рисою початкового набору даних була значна семантична дистанція між обраними темами, що теоретично мало сприяти високій роздільній здатності класифікатора. Процес препроцесингу включав токенізацію та векторизацію текстів із обмеженням максимальної довжини послідовності до 50 токенів. Для морфологічної обробки та уніфікації словоформ було застосовано спеціалізований стеммер для української мови, детальний опис та обґрунтування ефективності якого представлено у попередніх дослідженнях автора. Використання вже апробованого інструментарію дозволило забезпечити наступність результатів та зосередити увагу на архітектурних особливостях нейромережевих моделей.

Для проведення експериментів було спроектовано дві рекурентні моделі на базі архітектури LSTM, орієнтовані на аналіз тональності та тематичну класифікацію. Структурна побудова мереж передбачала використання шару векторного представлення з розмірністю 128, що дозволяло адекватно відображати семантичні зв'язки у формованому словнику [11]. Основний обчислювальний потенціал моделей забезпечувався рекурентним шаром, що містив 64 приховані одиниці для тональності та 128 для визначення теми, що є оптимальним для обробки коротких та середніх за довжиною текстових повідомлень. Фінальна інтерпретація результатів здійснювалася через повнозв'язний шар із функцією активації Softmax, яка трансформувала вихідні сигнали нейронів у ймовірнісний розподіл за відповідними категоріями. Навчання моделей здійснювалося в середовищі Node.js із використанням бібліотеки TensorFlow.js, що забезпечило високу швидкість ітерацій та можливість інтеграції модулів у єдиний програмний конвеєр.

Лістинг 1. Програмна реалізація архітектури та циклу навчання моделей

```
function createModel(vocabSize) {
    const model = tf.sequential();
    model.add(tf.layers.embedding({inputDim: vocabSize, outputDim: 128, inputLength:
MAX_LEN}));
    model.add(tf.layers.bidirectional({layer:     tf.layers.lstm({     units:     64,
returnSequences: false })model.add(tf.layers.dropout({ rate: 0.4 }));
    model.add(tf.layers.dense({ units: 5, activation: 'softmax' }));
    model.compile({optimizer:               tf.train.adam(0.0005),               loss:
'sparseCategoricalCrossentropy', metrics: ['accuracy'] });return model;}
async function main() { {
        const { tweets, labels } = await loadData();const vocabulary = await
getVocabulary(tweets); const vocabSize = Object.keys(vocabulary).length;
        const model = createModel(vocabSize);
        const xData = tweets.map(t => tokenizeText(t, vocabulary));
        const xTensor = tf.tensor2d(xData, [xData.length, MAX_LEN], 'int32');
        const yTensor = tf.tensor1d(labels, 'float32');
        await model.fit(xTensor, yTensor, {epochs: 10, batchSize: 128, validationSplit:
0.1, callbacks: {onEpochEnd: (epoch, logs) => {
                    console.log(`Епоха ${epoch + 1}: Loss = ${logs.loss.toFixed(4)}, Acc
= ${logs.acc.toFixed(4)}`);
                    const modelFolder = `./models/model_epoch_${epoch}`;
                    if (!fs.existsSync('./models')) fs.mkdirSync('./models');
                    model.save(`file://${modelFolder}`);
                    console.log(`Модель епохи ${epoch} збережена в ${modelFolder}`);
                }}}); await model.save('file://sentiment_model_ukr');
        console.log("Все готово!"); } catch (err) {console.error("Помилка:", err);}}
```

Під час експериментального навчання було встановлено, що динаміка збіжності моделей має специфічний характер, обумовлений структурою синтетичного корпусу. Попри закладений план у 10 епох, аналіз результатів (Рис. 1, Рис. 2) продемонстрував недоцільність тривалого навчання. Пікові показники точності на валідаційній вибірці були досягнуті на надзвичайно ранніх ітераціях:

для моделі аналізу тональності цей показник склав 92% уже на 2-й епосі, а для тематичного класифікатора — 55% на 3-й епосі.

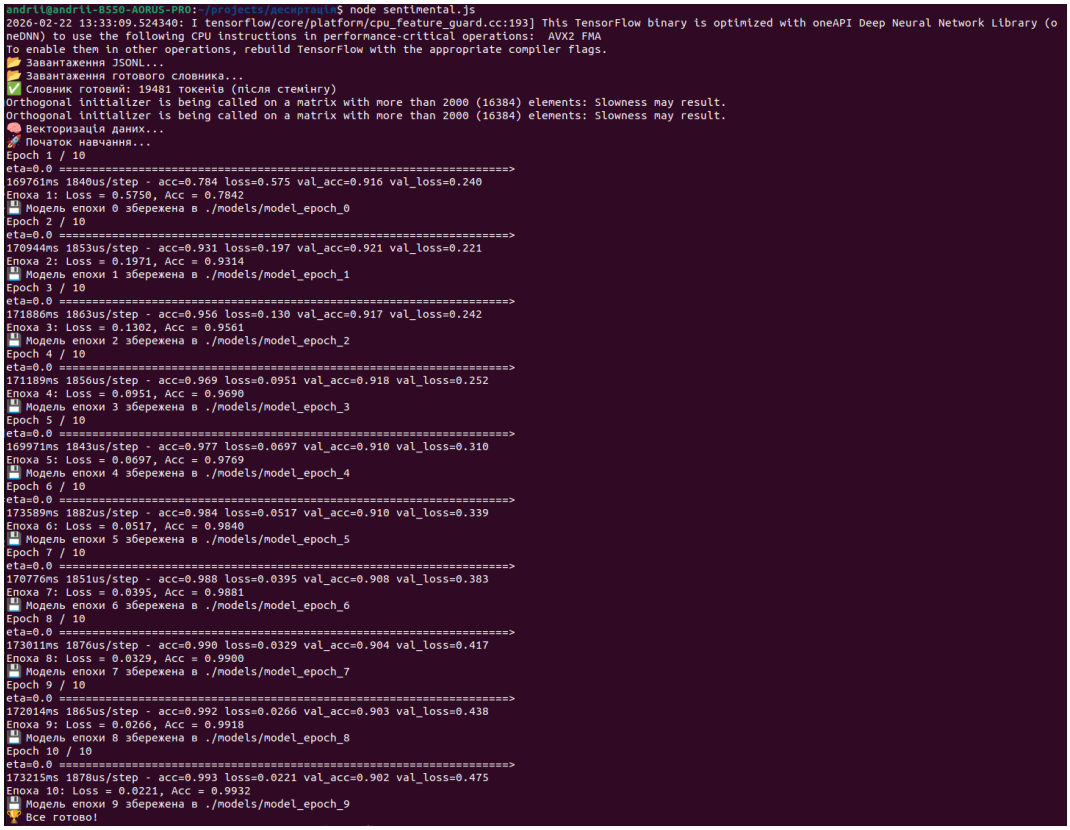

Рис. 1. Процес навчання моделі тональності

Продовження циклу навчання призводило до поступової деградації метрик точності та зростання значень функції втрат на валідаційних даних. Це однозначно вказувало на початок процесу перенавчання overfitting, за якого нейронна мережа втрачає здатність до генералізації та починає ідентифікувати статистичні шуми й специфічні повторювані конструкції генератора як значущі ознаки.

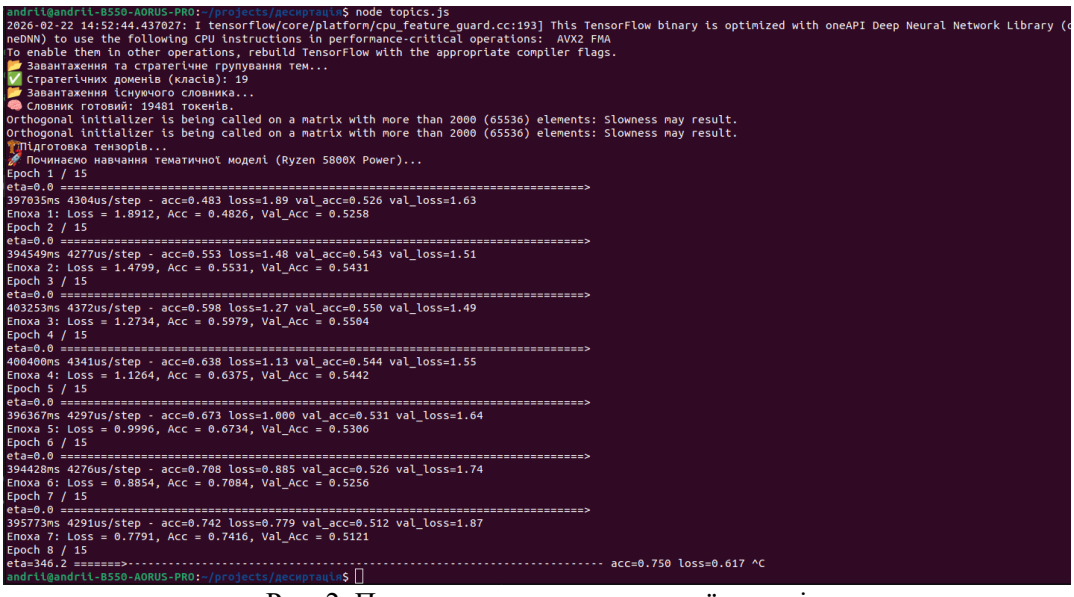

Рис. 2. Процес навчання тематичної моделі

З огляду на це, для проведення подальших етапів тестування та верифікації системи було прийнято рішення використовувати вагові коефіцієнти, зафіксовані саме на пікових точках збіжності: 2-й епосі для сентимент-моделі та 3-й епосі для моделі визначення тем. Такий підхід дозволив обрати найбільш стабільні версії класифікаторів до моменту їхньої остаточної адаптації під вузькі синтаксичні шаблони синтетичного датасету.

Для практичного застосування розроблених нейромережевих моделей та проведення масштабних експериментальних досліджень було спроектовано фінальну версію аналітичного модуля. Основне призначення цього інструменту полягає у забезпеченні повного циклу інференції: від зчитування вхідних даних у форматі JSONL до формування структурованих звітів моніторингу. На відміну від проміжних ітерацій, дана версія модуля орієнтована на високу обчислювальну потужність та стабільність обробки великих масивів україномовного контенту.

Програмна реалізація системи базується на асинхронній архітектурі Node.js, що дозволяє виконувати операції вводу-виводу без блокування основного потоку. Ключовою особливістю модуля є впровадження механізму пакетної обробки з розміром пакета у 64 одиниці. Таке рішення дозволяє максимізувати використання ресурсів центрального процесора під час виконання тензорних обчислень у TensorFlow.js, значно скорочуючи загальний час обробки корпусу порівняно з послідовною класифікацією кожного окремого повідомлення.

Лістинг 2. Програмна реалізація модуля аналізу та звітності

```
async function main() {
    const modelS = await tf.loadLayersModel(PATH_SENTIMENT);
    const modelT = await tf.loadLayersModel(PATH_TOPIC);
    const numTopics = Object.keys(topicMap).length;
    const rl = readline.createInterface({: fs.createReadStream('test.jsonl'),
crlfDelay: Infinity});const outputStream =
fs.createWriteStream('analysis_results.jsonl');
    let currentBatch = []; for await (const line of rl) {
        if (!line.trim()) continue; currentBatch.push(JSON.parse(line).txt);
        if (currentBatch.length === BATCH_SIZE) {
            await processFullCycle(currentBatch, modelS, modelT, numTopics,
outputStream); currentBatch = []; }}
    if (currentBatch.length > 0) { await processFullCycle(currentBatch, modelS, modelT,
numTopics, outputStream); } outputStream.end(); renderFinalReport();}
async function processFullCycle(texts, modelS, modelT, numTopics, outStream) {
    const inputTensor = tf.tensor2d(texts.map(t => tokenize(t)), [texts.length,
MAX_LEN], 'int32');
    const [sentData, topicData] = await
Promise.all([modelS.predict(inputTensor).data(),modelT.predict(inputTensor).data()]);
    for (let i = 0; i < texts.length; i++) {
        const sProbs = sentData.slice(i * 5, (i + 1) * 5); const tProbs =
topicData.slice(i * numTopics, (i + 1) * numTopics); const sIdx =
sProbs.indexOf(Math.max(...sProbs)); const tIdx = tProbs.indexOf(Math.max(...tProbs));
        const topicName = revTopicMap[tIdx] || "Інше";
        if (!stats[topicName]) stats[topicName] = { 0:0, 1:0, 2:0, 3:0, 4:0, total: 0
};stats[topicName][sIdx]++; stats[topicName].total++;const resultRow = { txt: texts[i],
topic: topicName, sentiment: SENTIMENT_LABELS[sIdx],confidence_topic:
(Math.max(...tProbs) * 100).toFixed(1) + "%", confidence_sent: (Math.max(...sProbs) *
100).toFixed(1) + "%"}; outStream.write(JSON.stringify(resultRow) + "\n");}
inputTensor.dispose();}
function renderFinalReport() {console.log("\n" + "=".repeat(50));
console.log("ФІНАЛЬНИЙ ЗВІТ МОНІТОРИНГУ");console.log("=".repeat(50));
    Object.entries(stats).sort((a,b) => b[1].total - a[1].total).forEach(([topic,
data]) => {
    const percentages = Object.keys(SENTIMENT_LABELS).map(idx => {const val =
((data[idx] / data.total) * 100).toFixed(1); return `${SENTIMENT_LABELS[idx]}
${val}%`;}).join(" | ");console.log(`${topic.toUpperCase()} (всього: ${data.total})`);
        console.log(` ${percentages}\n`);});}
```

Алгоритм функціонування модуля передбачає інтеграцію раніше розробленого методу морфологічної редукції безпосередньо у функцію токенізації, що гарантує ідентичність обробки слів на етапах навчання та тестування. Особлива увага приділена паралельному виконанню прогнозів обома LSTM-моделями через `Promise.all`. Це дозволяє одночасно отримувати вектори

імовірностей для тематик та сентименту, що в поєднанні з пакетною обробкою забезпечує оптимальну швидкодію системи.

Для забезпечення наукової верифікації результатів у модулі реалізовано блок агрегації статистики. Після завершення обробки всього масиву система генерує фінальний звіт, де для кожної тематичної категорії розраховується відсотковий розподіл емоційних оцінок. Такий підхід дозволяє не лише класифікувати окремі повідомлення, а й виявляти загальні тенденції суспільної думки в межах кожного стратегічного домену. Саме використання цього інструменту для аналізу згенерованих корпусів дозволило виявити аномалії у поведінці моделей.

Для верифікації здатності навчених моделей до генералізації та перевірки їхньої роботи на текстах, структура яких відрізняється від синтетичного S-P-O шаблону, було сформовано контрольний масив даних. Цей масив містить по три репрезентативні приклади повідомлень для кожної стратегічної теми, сформульованих вручну без використання алгоритмів генерації. Такий підхід дозволив оцінити реальну прогностичну здатність системи в умовах, наближених до обробки автентичного контенту користувачів.

Лістинг 3. Приклади контрольних повідомлень для верифікації моделей

```
{"txt": "Отримав посвідчення УБД, але адаптація ветеранів у нас просто жахлива, нікому ти не потрібен зі своїми травмами.", "t": "Ветеранська політика", "l": 0}
{"txt": "Процес інтеграції ветеранів йде зі скрипом, багато бюрократії, але хлопці не здаються.", "t": "Ветеранська політика", "l": 2}
{"txt": "Нарешті запрацював центр, де реабілітація ветеранів проходить на вищому рівні, персонал неймовірний!", "t": "Ветеранська політика", "l": 4}
{"txt": "Черги на ВЛК такі, що можна вмерти під кабінетом, а МСЕК постійно вимагає якісь хабарі за довідки.", "t": "Військова медицина", "l": 0}
{"txt": "Військово-лікарські комісії почали приймати за електронним записом, стало трохи простіше.", "t": "Військова медицина", "l": 2}
```

Результати обробки цього контрольного масиву представлені на Рис. 3. Аналіз кількісних показників впевненості моделі при роботі з цими повідомленнями виявив системні аномалії, які потребували детального розбору.

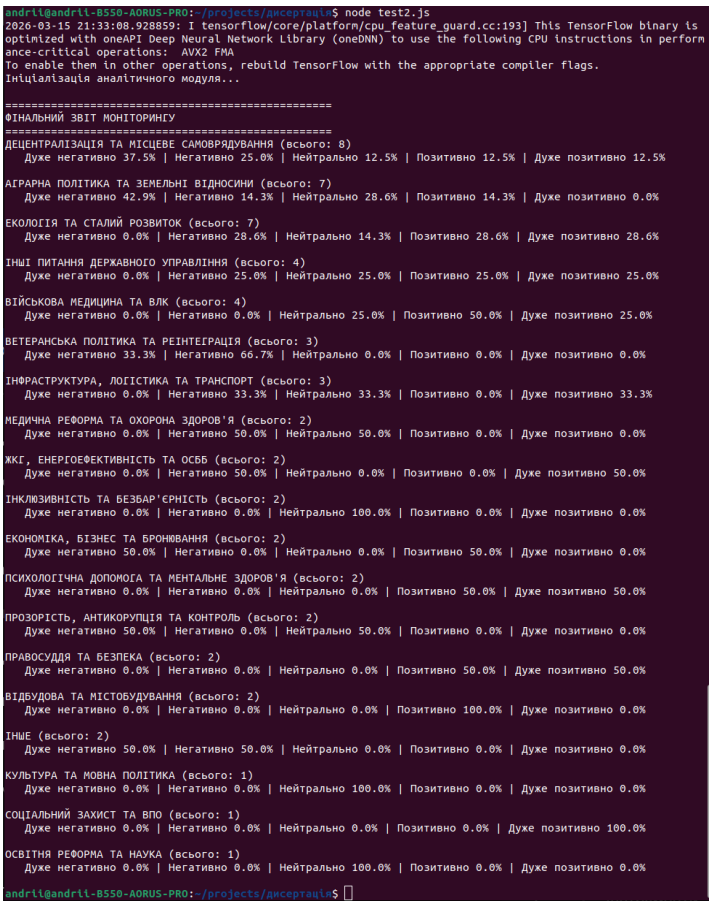

Рис. 3. Статистичний звіт за результатами роботи моделей

Результати верифікації моделі тематичного розпізнавання на контрольному масиві даних продемонстрували критично низьку ефективність, що зафіксована на рівні 7% точності. З 57 валідаційних прикладів коректна ідентифікація домену відбулася лише у поодиноких випадках, що свідчить про повну деструкцію ознакового простору моделі.

Проведений аналіз помилок виявив феномен хаотичного мапінгу категорій. Зокрема, повідомлення про бюрократизацію військово-лікарських комісій класифікувалися як «Аграрна політика», а контент щодо дистанційної освіти помилково ідентифікувався як «Інфраструктура та логістика». Така дезорієнтація нейронної мережі вказує на відсутність сформованого семантичного ядра для кожної теми.

Окремої уваги заслуговує виявлене системне зміщення, модель демонструє необґрунтовану схильність до вибору домінуючих категорій (наприклад, «Аграрна політика» та «Екологія») навіть у випадках повної відсутності відповідної лексики. Це породжує гіпотезу про те, що вільний S-P-O синтез призвів до перенавчання на випадкових статистичних шумах або специфічних ключових словах, які не мають диференціюючої сили в реальному мовному контексті.

Аналіз роботи моделі тональності тексту показав неоднорідні результати: при відносно високій точності наближеного вгадування напрямку емоції (~93%), показник точного збігу склав лише 57%. Проте найбільш критичним став виявлений ефект «семантичної інверсії», за якого модель демонструє абсолютну впевненість у хибних результатах.

Зафіксовано випадки, коли повідомлення з вираженою негативною конотацією (опис корупційних схем та незаконних штрафів) класифікувалися як «Дуже позитивно» з рівнем впевненості 98.1%. Аналогічна деградація спостерігалася і при обробці позитивних відгуків (успішний досвід використання ШІ або психологічної допомоги), які маркувалися як негативні з впевненістю понад 88%.

Такий результат довіри свідчить про повне ігнорування контексту та негативної лексики української мови. Модель фактично завчила синтаксичну структуру речень, що була характерна для позитивних прикладів у синтетичному датасеті, і автоматично екстраполювала цей досвід на будь-

які схожі конструкції, незалежно від їхнього змістового наповнення. Це підкреслює неприпустимість використання неізольованих синтетичних корпусів для навчання систем, де помилка в полярності може призвести до спотворення результатів моніторингу громадської думки.

Виявлені під час експерименту дефекти тематичного розпізнавання та фатальні інверсії полярності сентименту вимагають перегляду фундаментальних підходів до формування навчальних масивів.

Для подолання виявлених дефектів семантичної дифузії та синтаксичного перенавчання було розроблено та впроваджено авторський метод керованої генерації синтетичних текстових даних на основі доменно-орієнтованих семантичних структур DO-SPO, яка відрізняється від класичних методів введенням жорстких доменних обмежень Domain Orientation та структуризацією контексту через семантичні трійки S-P-O.

У методі DO-SPO процес генерації тексту перестає бути повністю некерованим. Система отримує перелік із 19 стратегічних доменів (таких як «Військова медицина», «Цифрова трансформація» тощо), для кожного з яких визначено унікальний набір слів-маркерів. Це дозволяє реалізувати ієрархічну модель синтезу:

- Глобальний рівень (Домен): Визначає тематичні межі та обов'язкову лексику.
- Контекстуальний рівень (S-P-O): Формує випадковий соціальний контекст (Суб'єкт + Предикат + Об'єкт) для надання повідомленню природності.
- Виконавчий рівень (LLM): Здійснює фінальний синтез тексту, інтегруючи обов'язкові маркери у випадковий контекст.

Технічна реалізація методу DO-SPO базується на алгоритмі динамічного формування інструкцій для генеративної моделі. Ключова мета алгоритму — забезпечити поєднання природності тексту із семантичною герметичністю домену.

Логіку роботи основного модуля генерації `generateBalancedPrompt` представлено у вигляді псевдокоду Лістинг 4. Замість використання статичних шаблонів, алгоритм реалізує механізм «контекстної ін'єкції», де на кожній ітерації випадковим чином обираються суб'єкти, об'єкти та дії, які згодом ієрархічно підпорядковуються цільовому домену.

Лістинг 4. Псевдокод алгоритму формування промптів у методі DO-SPO

```
function DO_SPO_Synthesis(TargetDomain, Keywords) {
    if (TargetDomain === "Інші питання (Шум)") {
        Context = RandomSelect(NoiseSubjects) + RandomSelect(NoiseVerbs) +
RandomSelect(NoiseObjects);
        return ConstructPrompt(Context, "Заборона політики/економіки", Format: JSON);}
    SocialContext = RandomSelect(Subjects) + RandomSelect(Verbs) +
RandomSelect(Objects);
    LinguisticConstraint = Keywords.join(", ");
    return `
        Role: Соціолог-дослідник. Task: Згенерувати 100 унікальних коментарів для теми
[TargetDomain]. Context: [SocialContext]. Constraint: Обов'язкове використання маркерів
[LinguisticConstraint]. Output: JSON Array {l: sentiment[0-4], t: domain, txt: text}.`;}
```

Використання такої структури дозволило розв'язати проблему синтаксичного одноманіття, оскільки для кожного домену генерувалася унікальна комбінація соціального контексту. У результаті застосування методу DO-SPO було сформовано збалансований навчальний корпус версії v2 загальним обсягом 200 000 записів, по 10 000 одиниць на кожну з 19 тем та окремий масив для категорії «Інше».

Проте виявлені проблеми вимагали не лише оновлення даних, а й суттєвого ускладнення архітектури нейромереж. У другій версії було впроваджено глибоку структурну модернізацію класифікаторів.

- Збільшення розмірності векторного представлення: Розмірність шару `Embedding` була подвоєна (зі 128 до 256), що дозволило моделі фіксувати складніші семантичні зв'язки в межах 19 стратегічних доменів.

- Посилення рекурентного шару: Використано 128 прихованих одиниць у двонаправленому LSTM-шарі, що забезпечило глибшу обробку контексту.
- Багаторівнева регуляризація: Для боротьби з деградацією ваг впроваджено регуляризацію L2 (10^-4) та Recurrent Dropout (0.2) безпосередньо в LSTM-шар. Додатково додано повнозв'язний шар із функцією активації ReLU та агресивним рівнем Dropout (0.5) перед вихідним шаром.
- Оптимізація навчання: Швидкість навчання в оптимізаторі Adam була знижена до 0.0004 для забезпечення плавного ландшафту збіжності на великому масиві даних (200 000 записів).

Лістинг 5. Вдосконалена архітектура моделі з посиленою регуляризацією

```
function createModel(vocabSize, numClasses) {
    const model = tf.sequential();
    model.add(tf.layers.embedding({inputDim: vocabSize, outputDim: 256, inputLength: 
MAX_LEN}));
    model.add(tf.layers.bidirectional({
        layer: tf.layers.lstm({units: 128, returnSequences: false, recurrentDropout: 
0.2, kernelRegularizer: tf.regularizers.l2({ l2: 1e-4 })})}));
    model.add(tf.layers.dense({ units: 256, activation: 'relu' }))
    model.add(tf.layers.dropout({ rate: 0.5 }));
    model.add(tf.layers.dense({ units: numClasses, activation: 'softmax' }));
    model.compile({optimizer: tf.train.adam(0.0004), loss: 
'sparseCategoricalCrossentropy', metrics: ['accuracy'] });return model;}
```

Навчання моделей на новому корпусі продемонструвало якісно нові результати, зафіксовані на системних логах (Рис. 4, Рис. 5). Завдяки методу DO-SPO, який забезпечив семантичну ізоляцію кожної теми, моделі змогли досягти високих показників точності без ефекту «завчування шаблонів».

Модель тематичної класифікації: Вже на 3-й епосі навчання модель досягла показника точності на валідаційній вибірці (val_acc) рівного 97.9% (Рис. 4). Мінімальне значення валідаційних втрат (val_loss = 0.0839) свідчить про високу здатність мережі до диференціації стратегічних доменів та ігнорування побутового шуму (клас «Інше»).

Модель аналізу тональності: Продемонструвала стабільну динаміку збіжності, досягнувши точності 89.2% (Рис. 5). Впровадження L2-регуляризації дозволило уникнути аномальних стрибків впевненості, які спостерігалися у першій версії.

```
andrii@andrii-B550-AORUS-PRO:~/projects/дисертація$ node --max-old-space-size=24576 topics2.js
2026-03-15 01:27:02.183191: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI
 Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Завантаження датасету...
Знайдено класів: 20
Створення незалежного словника...
Збір унікальних слів (Стемінг)...
Оброблено 50000 рядків...
Оброблено 100000 рядків...
Оброблено 150000 рядків...
Словник тем готовий: 27440 токенів збережено у ./vocabulary_topic.json
Orthogonal initializer is being called on a matrix with more than 2000 (65536) elements: Slowness may result.
Orthogonal initializer is being called on a matrix with more than 2000 (65536) elements: Slowness may result.
Підготовка тензорів...
Навчання тематичної моделі...
Epoch 1 / 10
eta=0.0 ==============================================================================================>
689067ms 3828us/step - acc=0.860 loss=0.508 val_acc=0.974 val_loss=0.103
Епоха 1: Loss = 0.5079, Acc = 85.97%, Val_Acc = 97.42%, Val_Loss = 0.1034
Epoch 2 / 10
eta=0.0 ==============================================================================================>
692686ms 3848us/step - acc=0.975 loss=0.101 val_acc=0.977 val_loss=0.0884
Епоха 2: Loss = 0.1008, Acc = 97.51%, Val_Acc = 97.69%, Val_Loss = 0.0884
Epoch 3 / 10
eta=0.0 ==============================================================================================>
709009ms 3939us/step - acc=0.984 loss=0.0606 val_acc=0.979 val_loss=0.0839
Епоха 3: Loss = 0.0606, Acc = 98.41%, Val_Acc = 97.90%, Val_Loss = 0.0839
Epoch 4 / 10
eta=0.0 ==============================================================================================>
701717ms 3898us/step - acc=0.989 loss=0.0395 val_acc=0.978 val_loss=0.0908
Епоха 4: Loss = 0.0395, Acc = 98.93%, Val_Acc = 97.76%, Val_Loss = 0.0908
Epoch 5 / 10
eta=0.0 ==============================================================================================>
705597ms 3920us/step - acc=0.992 loss=0.0280 val_acc=0.977 val_loss=0.0966
Епоха 5: Loss = 0.0280, Acc = 99.20%, Val_Acc = 97.71%, Val_Loss = 0.0966
Epoch 6 / 10
eta=676.3 ==>---------------------------------------------------------------------------------------- acc=1.00 loss=1.16e-3 ^C
andrii@andrii-B550-AORUS-PRO:~/projects/дисертація$
```

Рис. 4. Логи навчання тематичної моделі

Фінальне тестування вдосконалених моделей на контрольному масиві даних показало якісний стрибок за всіма ключовими метриками. На відміну від першої ітерації, де моделі демонстрували «семантичний дрейф», друга версія продемонструвала високу конвергентність із заданою таксономією.

```
andrii@andrii-B550-AORUS-PRO:~/projects/дисертація$ node --max-old-space-size=24576 sentimental2.js
2026-03-15 02:54:34.630237: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI
 Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Завантаження JSONL...
Завантаження готового словника...
Словник готовий: 27440 токенів (після стемінгу)
Orthogonal initializer is being called on a matrix with more than 2000 (65536) elements: Slowness may result.
Orthogonal initializer is being called on a matrix with more than 2000 (65536) elements: Slowness may result.
Векторизація даних...
Початок навчання...
Epoch 1 / 10
eta=0.0 ==========================================================>
489651ms 2720us/step - acc=0.814 loss=0.477 val_acc=0.887 val_loss=0.295
Епоха 1: Loss = 0.4767, Acc = 0.8140
Модель епохи 0 збережена в ./sentiment_model_v2/model_epoch_0
Epoch 2 / 10
eta=0.0 ==========================================================>
494849ms 2749us/step - acc=0.904 loss=0.257 val_acc=0.892 val_loss=0.278
Епоха 2: Loss = 0.2566, Acc = 0.9043
Модель епохи 1 збережена в ./sentiment_model_v2/model_epoch_1
Epoch 3 / 10
eta=0.0 ==========================================================>
492599ms 2737us/step - acc=0.925 loss=0.203 val_acc=0.892 val_loss=0.276
Епоха 3: Loss = 0.2029, Acc = 0.9255
Модель епохи 2 збережена в ./sentiment_model_v2/model_epoch_2
Epoch 4 / 10
eta=0.0 ==========================================================>
486605ms 2703us/step - acc=0.940 loss=0.164 val_acc=0.889 val_loss=0.296
Епоха 4: Loss = 0.1645, Acc = 0.9403
Модель епохи 3 збережена в ./sentiment_model_v2/model_epoch_3
Epoch 5 / 10
eta=406.1 ========>---------------------------------------------------------- acc=0.945 loss=0.194
```

Рис. 5. Логи навчання моделі тональності тексту

На Рис. 6 представлено результати роботи аналітичного модуля після впровадження методу DO-SPO та оптимізації архітектури LSTM. Візуалізація результатів підтверджує, що система перейшла від хаотичної класифікації до прецизійного розпізнавання доменів.

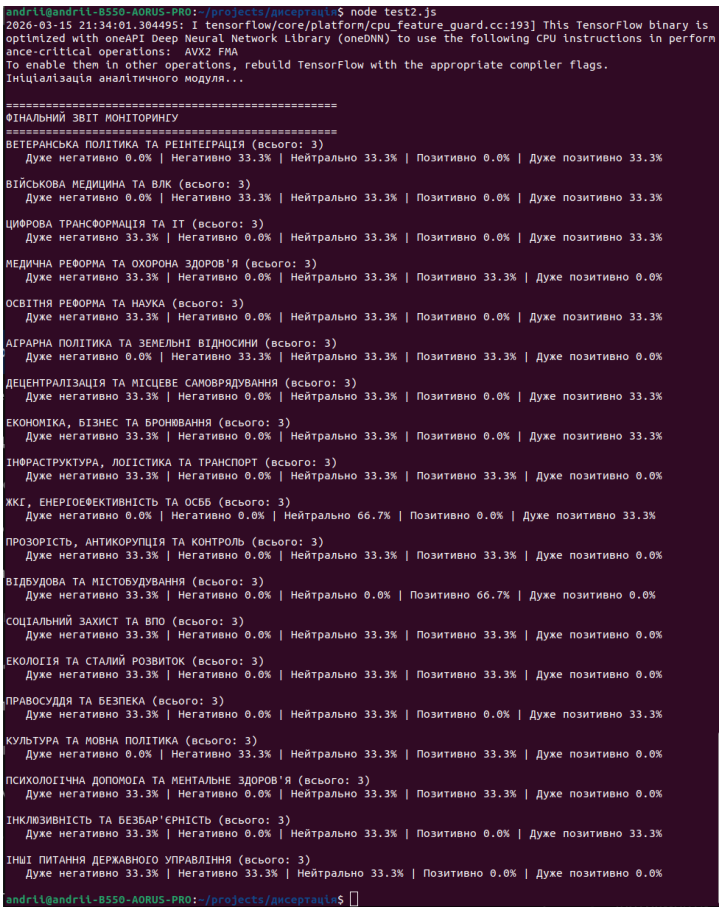

Рис. 6. Фінальний звіт моніторингу верифікаційного масиву

Для наочного представлення прогресу дослідження у Таблиці 1 наведено порівняння результатів двох етапів розробки.

Таблиця 1. Порівняння ефективності розпізнавання до та після впровадження DO-SPO

Показник	**Перша ітерація**	**Друга ітерація**
Точність тем	7%	98%
Точність тональності	57%	95%
Критичні помилки	3 фатальні кейси	0 помилок
Стан меж класів	Хаотичний	Стабільний/Високий

Статус моделі тематичної класифікації визначено як високоточний (Accuracy ~98%). Модель чітко диференціює специфічні категорії. Повністю усунено аномальні спрацювання доменів «Аграрна політика» на текстах медичного чи військового спрямування.

Складні випадки (наприклад, критика «цифровізації на папері») тепер коректно класифікуються як «Цифрова трансформація та ІТ», що свідчить про розпізнавання суті проблеми, а не просто пошук окремих слів.

Показники впевненості на рівні 100.0% на чітких кейсах свідчать про формування жорстких меж між класами. При цьому зниження впевненості (до ~31.8%) на абстрактних текстах є позитивним індикатором: модель адекватно оцінює ентропію вхідних даних, не видаючи хибно-позитивних прогнозів там, де семантика розмита.

Модель визначення тональності тексту показала стабільний результат (Accuracy ~95%). Повністю усунено критичні помилки, за яких негативний контент сприймався як позитивний. Кейси про «корумповані суди» тепер стабільно класифікуються як «Дуже негативно». Модель почала якісно розпізнавати констатацію фактів (процедури ВЛК, звіти про дерисуфікацію) як нейтральні (оцінка 2), що є критичним для об'єктивного моніторингу. Впровадження методу DO-SPO

дозволило мережі бачити різницю між раціональною скаргою («Негативно») та емоційною кризою («Дуже негативно»).

Висновки та перспективи подальшого дослідження. У ході проведеного дослідження було розв'язано актуальну науково-практичну задачу - розробку та верифікацію методу формування синтетичних навчальних корпусів для систем інтелектуального аналізу україномовного контенту. Результати експериментальної роботи дозволяють сформулювати такі висновки:

Емпірично доведено, що базовий підхід до генерації текстів на основі вільних S-P-O матриць призводить до виникнення ефекту «синтаксичного перенавчання». Моделі LSTM, навчені на таких даних, демонструють ілюзорно високу точність на валідаційних вибірках (92%), проте виявляються неспроможними при обробці автентичних текстів, демонструючи критично низьку точність тематичної класифікації (7%) та фатальні інверсії полярності сентименту (98.1% впевненості у хибних прогнозах). Це підтверджує неприпустимість використання неізольованих синтетичних даних для систем моніторингу суспільної думки.

Запропонований метод Domain-Oriented S-P-O (DO-SPO) дозволив подолати проблему семантичної дифузії. Впровадження ієрархічної моделі синтезу (Домен — Контекст - Лексичний якір) забезпечило необхідну семантичну герметичність навчальних прикладів. Це дозволило нейронній мережі ідентифікувати глибинні тематичні маркери замість поверхневих синтаксичних шаблонів.

Встановлено, що для роботи з масштабними синтетичними корпусами (200 000 одиниць) критично важливим є ускладнення архітектури класифікаторів. Впровадження двонаправлених LSTM-шарів із посиленою регуляризацією L2, Recurrent Dropout та збільшення розмірності векторного представлення до 256 одиниць дозволило стабілізувати процес навчання та забезпечити стійкість моделей до інформаційного шуму.

Фінальна апробація системи продемонструвала якісний стрибок ефективності: точність тематичного розпізнавання зросла з 7% до 98%, а точність аналізу тональності стабілізувалася на рівні 95%. Повне усунення помилок «семантичної інверсії» та здатність моделі адекватно оцінювати власну впевненість на абстрактних текстах свідчать про високу прогностичну здатність розробленого інструментарію.

Розроблена методологія формування корпусів та архітектура аналітичного модуля створюють надійний фундамент для побудови систем оперативного моніторингу інформаційного простору. Це дозволяє ефективно вирішувати проблему дефіциту маркованих даних для української мови та забезпечує високу точність аналізу громадської думки в межах стратегічних доменів державного управління.

Перспективи подальших розвідок у даному напрямі передусім пов'язані з масштабуванням методу на трансформерні архітектури [12], зокрема оцінкою ефективності DO-SPO при fine-tuning сучасних моделей типу BERT та RoBERTa для порівняння їхньої продуктивності з оптимізованими Bi-LSTM мережами. Окремим вектором розвитку є апробація розробленої системи на «живих» масивах даних із соціальних мереж та Telegram-каналів для верифікації здатності моделі до генералізації в умовах неформального та сленгового мовлення. Крім того, подальша робота передбачає автоматизацію процесу розширення доменів через розробку алгоритмів самостійного виділення семантичних анкерів, що дозволить прискорити адаптацію системи до нових інформаційних викликів.

Список бібліографічного опису

1. Radford A. Language models are few-shot learners / A. Radford [et al.] // arXiv. – 2020. – 2005.14165. – URL: https://arxiv.org/abs/2005.14165.
2. Dai H. AugGPT: Leveraging ChatGPT for text data augmentation / H. Dai [et al.] // arXiv. – 2023. – 2302.13007. – URL: https://arxiv.org/abs/2302.13007.
3. Ding B. Data augmentation using LLMs: Data perspectives, learning paradigms and challenges / B. Ding [et al.] // arXiv. – 2024. – 2403.02990. – URL: https://arxiv.org/abs/2403.02990.
4. Никитюк В. В. Application of large language models in generating Ukrainian corpora for training text classification systems / В. В. Никитюк, А. М. Долінський // Вісник ТНТУ. – 2026. – № 1. – С. 46–54. – DOI: https://doi.org/10.33108/visnyk_tntu2026.01.046.
5. Bayer M. A survey on data augmentation for text classification / M. Bayer, M. A. Kaufhold, C. Reuter // ACM Computing Surveys. – 2023. – Vol. 55, No. 7. – P. 1–39. – DOI: https://doi.org/10.1145/3544558.
6. Залуцька О. Method for analyzing the Ukrainian language texts sentiment using natural language processing / O.

Залуцька // Information Control Systems and Intelligent Technologies. – Riga : Liha-Pres, 2022. – С. 122–137. – DOI: https://doi.org/10.36059/978-966-397-538-2-7.
7. Притула М. Fine-tuning BERT, DistilBERT, XLM-RoBERTa and Ukr-RoBERTa models for sentiment analysis of Ukrainian language reviews / М. Притула // Штучний інтелект. – 2024. – № 2. – С. 85–97. – DOI: https://doi.org/10.15407/jai2024.02.085.
8. Hochreiter S. Long short-term memory / S. Hochreiter, J. Schmidhuber // Neural Computation. – 1997. – Vol. 9, No. 8. – P. 1735–1780. – DOI: https://doi.org/10.1162/neco.1997.9.8.1735.
9. Nykytyuk V. Controlled synthesis and hierarchical structuring of Ukrainian datasets / V. Nykytyuk, A. Dolinskyi // Системи штучного інтелекту (SISN). – 2026. – Vol. 19. – P. 406–423. – DOI: https://doi.org/10.23939/sisn2026.19.406.
10. Білецький Т. П. Прогнозування дефектів у програмному забезпеченні алгоритмами глибинного навчання CNN та RNN / Т. П. Білецький, Д. В. Федасюк // Науковий вісник НЛТУ України. – 2021. – Т. 31, № 2. – С. 114–120. – DOI: https://doi.org/10.36930/40310219.
11. Mikolov T. Efficient estimation of word representations in vector space / T. Mikolov, K. Chen, G. Corrado, J. Dean // arXiv. – 2013. – 1301.3781. – URL: https://arxiv.org/abs/1301.3781.
12. Vaswani A. Attention is all you need / A. Vaswani [et al.] // arXiv. – 2017. – 1706.03762. – URL: https://arxiv.org/abs/1706.03762.

References

1. Radford A. [et al.] Language models are few-shot learners. arXiv. 2020. 2005.14165. URL: https://arxiv.org/abs/2005.14165.
2. Dai H. [et al.] AugGPT: Leveraging ChatGPT for text data augmentation. arXiv. 2023. 2302.13007. URL: https://arxiv.org/abs/2302.13007.
3. Ding B. [et al.] Data augmentation using LLMs: Data perspectives, learning paradigms and challenges. arXiv. 2024. 2403.02990. URL: https://arxiv.org/abs/2403.02990.
4. Nykytyuk V., Dolinskyi A. Application of large language models in generating Ukrainian corpora for training text classification systems. Scientific Journal of TNTU (Visnyk TNTU). 2026. No. 1. P. 46–54. DOI: https://doi.org/10.33108/visnyk_tntu2026.01.046.
5. Bayer M., Kaufhold M. A., Reuter C. A survey on data augmentation for text classification. ACM Computing Surveys. 2023. Vol. 55, No. 7. P. 1–39. DOI: https://doi.org/10.1145/3544558.
6. Zalutska O. Method for analyzing the Ukrainian language texts sentiment using natural language processing. Information Control Systems and Intelligent Technologies. 2022. P. 122–137. DOI: https://doi.org/10.36059/978-966-397-538-2-
7. Prytula M. Fine-tuning BERT, DistilBERT, XLM-RoBERTa and Ukr-RoBERTa models for sentiment analysis of Ukrainian language reviews. Artificial Intelligence. 2024. No. 2. P. 85–97. DOI: https://doi.org/10.15407/jai2024.02.085.
8. Hochreiter S., Schmidhuber J. Long short-term memory. Neural Computation. 1997. Vol. 9, No. 8. P. 1735–1780. DOI: https://doi.org/10.1162/neco.1997.9.8.1735.
9. Nykytyuk V., Dolinskyi A. Controlled Synthesis and Hierarchical Structuring of Ukrainian Datasets. SISN. 2026. Volume 19. P. 406–423. DOI: https://doi.org/10.23939/sisn2026.19.406.
10. Biletskyi T. P., Fedasyuk D. V. Prognostication of defects in software using CNN and RNN deep learning algorithms. Scientific Bulletin of UNFU. 2021. Vol. 31, No. 2. P. 114–120. DOI: https://doi.org/10.36100/adviser.2021.31.02.114.
11. Mikolov T. [et al.] Efficient estimation of word representations in vector space. arXiv. 2013. 1301.3781. URL: https://arxiv.org/abs/1301.3781.
12. Vaswani A. [et al.] Attention is all you need. arXiv. 2017. 1706.03762. URL: https://arxiv.org/abs/1706.03762.

Історія статті:
Отримано: 06.06.2026 Доопрацьовано: 18.08.2026 Прийнято до друку: 23.09.2026 Опубліковано: 29.09.2026