УДК 004.91

Н. В. Козинець; Т. М. Заболотня, канд. техн. наук, доц.

ВПЛИВ КОМБІНОВАНИХ ВЕКТОРНИХ ПРЕДСТАВЛЕНЬ НА ТОЧНІСТЬ ПОШУКУ НЕЧІТКИХ ДУБЛІКАТІВ

У статті запропоновано новий підхід до виявлення нечітких у текстових даних, що базується на інтеграції класичних та сучасних методів векторизації. Зокрема, традиційне TF-IDF-векторизування поєднано з контекстуальними ембедингами (BERT), які враховують не лише окремі слова, а й їхній контекст у межах усього документа. Це дозволяє отримати багатовимірне представлення тексту, яке краще відображає його семантичне значення. Така комбінована методологія дає змогу підвищити точність пошуку схожих за змістом, але по-різному сформульованих текстів, що є важливим у таких сферах, як інформаційний пошук, аналіз дублікатів у базах даних та верифікація унікальності контенту. Окрему увагу приділено врахуванню синонімів та антонімів у процесі порівняння текстових фрагментів, що дає змогу не лише ідентифікувати прямі збіги, а й аналізувати схожість на глибшому семантичному рівні. Це, у свою чергу, сприяє зменшенню кількості хибних спрацьовувань, оскільки метод здатен краще розпізнавати контекстуальні відмінності та схожості між словами, що особливо актуально для текстів, написаних природною мовою. Водночас такий підхід підвищує ефективність виявлення прихованих дублікатів, які могли б залишитися непоміченими при використанні традиційних методів аналізу, орієнтованих лише на лексичну подібність. Експериментальні результати підтвердили переваги запропонованого рішення порівняно з базовим методом косинусної схожості, оскільки воно забезпечує більшу точність та повноту, що є критично важливим для задач автоматичної обробки текстових даних. У підсумку окреслено подальші напрями досліджень, зокрема можливості оптимізації обчислювальної складності запропонованого методу, його адаптацію до специфічних предметних областей, а також дослідження впливу додаткових семантичних ознак на якість виявлення нечітких дублікатів.

Ключові слова: *нечіткі дублікати, комбіновані векторні представлення, TF-IDF, BERT, косинусна подібність, семантичні ембединги, синоніми і антоніми, виявлення дублікатів.*

Вступ

Виявлення нечітких дублікатів у текстових даних є важливою і складною задачею в галузях управління даними, інформаційного пошуку та обробки природної мови [1]. Нечіткі дублікати – це текстові записи або документи, які не є точними копіями один одного, але мають значну схожість за змістом [2]. Проблема пошуку таких дублікатів виникає в різних контекстах:

– управління базами даних: дублювання записів збільшує обсяг даних, спотворює результати запитів та знижує продуктивність системи. Своєчасне виявлення і видалення дублікатів оптимізує використання пам'яті та підвищує якість даних;

– пошукові системи: надання користувачу декількох результатів, що є дублікатами або майже дублікатами одного документа, знижує релевантність видачі. Виявлення дублікатів дозволяє підвищити якість пошуку, виключивши повтори;

– перевірка на плагіат: у наукових та освітніх установах важливо знаходити перефразовані або частково змінені тексти, що по суті дублюють раніше опубліковані. Виявлення нечітких дублікатів допомагає виявити випадки плагіату та забезпечити академічну доброчесність;

– соціальні мережі та форуми: великі обсяги контенту часто містять повторювані або схожі повідомлення. Автоматичне знаходження таких дублікатів зменшує інформаційний шум і полегшує модерацію контенту.

Нечітке дублювання ускладнюється варіативністю тексту – використанням синонімів, перестановкою слів, незначними редакційними змінами, які зберігають основний зміст. Короткі тексти (наприклад, твіти чи заголовки новин) надають мало даних для порівняння, що також створює виклики для точного виявлення збігів. Традиційні методи порівняння рядків можуть бути неефективними в таких випадках або надто повільними на великих масивах даних. Таким чином, існує потреба в методах, що враховують семантичну подібність текстів (значення слів у контексті) поряд із лексичною схожістю, аби підвищити точність виявлення нечітких дублікатів.

Актуальність дослідження. Більшість класичних підходів або орієнтовані на точний збіг символів, або на статистичну подібність слів, через що пропускають семантично еквівалентні тексти чи, навпаки, позначають як дублікати тексти з різним змістом. Сучасні досягнення в обробці мови (wordembeddings, трансформери) відкривають можливість врахувати значення слів, але їх застосування потребує значних обчислювальних ресурсів. У зв'язку з цим, актуальним є поєднання кількох підходів для досягнення балансу між точністю та ефективністю.

Метою статті є підвищення точності пошуку нечітких дублікатів шляхом розробки комбінованого методу, який об'єднує традиційні векторні представлення тексту з сучасними контекстними ембедингами. Для досягнення поставленої мети необхідно проаналізувати наявні методи виявлення нечітких дублікатів, запропонувати методологію комбінування векторних представлень та оцінити її ефективність експериментально.

Огляд існуючих методів та підходів

Пошук дублікатів і схожих текстів досліджується протягом кількох десятиліть, і уже напрацьовано широкий спектр методів – від простих метрик схожості рядків до складних моделей глибинного навчання [3]. Розглянемо основні групи підходів та їх особливості.

Алгоритми на основі відстаней редагування. Найвідомішим є відстань Левенштейна [4], яка визначає мінімальну кількість редагувань (вставка, видалення, заміна символів) для перетворення одного рядка в інший. Ця метрика ефективна для коротких рядків і дозволяє явно врахувати різницю в написанні. Її розширення, відстань Джаро-Вінклера, додає вагові коефіцієнти за співпадіння початку рядків, покращуючи оцінку схожості для, наприклад, власних назв. Метрики редагування добре виявляють орфографічно схожі рядки, проте ігнорують семантику – слова-синоніми або перестановка слів можуть дати велику відстань, хоч зміст залишається тим самим. Також їх обчислювальна складність O(mn) (для рядків довжини m і n) робить важким застосування до довгих документів або великої кількості пар для порівняння.

Токен та шинглінг-підходи. Методи цього класу оперують не окремими символами, а словами чи фрагментами тексту (шинглами). Наприклад, порівняння множин N-грам (послідовностей із N слів або символів) дозволяє виміряти подібність через частку спільних фрагментів між документами [5]. Коефіцієнт Жаккара є популярною мірою для таких множин: він визначається як відношення числа спільних елементів до загального числа уніфікованих елементів двох множин. Ще один метод –Winnowing – створює відбитки (фінгерпринти) документа на основі хешування N-грамів. Ці відбитки компактно репрезентують текст і дозволяють швидко знаходити збіги, що використовується в системах перевірки на плагіат. Відповідно до подібного підходу генеруються сигнатури документів, інваріантні до незначних змін, для ефективного порівняння великої кількості текстів. Перевага токен-базованих методів – стійкість до дрібних змін (наприклад, перестановки слів не впливають на множину слів). Однак, якщо тексти використовують різні слова для одного поняття, такі методи теж не побачать схожості між ними.

Векторна модель і статистичні характеристики. В інформаційному пошуку широко застосовується косинусна міра між векторними представленнями документів [6]. Зокрема,

кожен документ можна перетворити на вектор ознак, наприклад, за допомогою моделі "bagofwords". Одним із найпоширеніших є представлення у вигляді TF-IDF-вектора. TF-IDF (TermFrequency–InverseDocumentFrequency) визначає вагу термів: чим частіше слово зустрічається у цьому документі і рідше – в інших, тим вища його вага. Вектор TF-IDF добре відображає лексичний склад тексту, приділяючи більшу вагу "значущим" словам. Косинусна подібність між такими векторами дає міру схожості документів за спільними термінами. Цей підхід ефективний для великих текстів і швидко рахується навіть для великих колекцій. Проте він має обмеження: повністю ігнорує порядок слів і не враховує значення слів. Наприклад, тексти з різними словами однакового змісту (синонімами) матимуть нульову спільність термів і, відповідно, нульову косинусну схожість за TF-IDF, попри очевидну семантичну еквівалентність.

Семантичні моделі на основі ембедингів. Сучасні методи машинного навчання дозволяють відобразити слова і цілі речення у вигляді семантичних векторів у багатовимірному просторі. Моделі Word2Vec [7] та GloVe [8] навчаються на великих корпусах і розміщують близькі за змістом слова поруч у векторному просторі. Word2Vec має дві архітектури – CBOW та Skip-gram – що навчаються передбачати слово за контекстом або контекст за словом відповідно. Отримані ембединги слів здатні вловлювати семантичні зв'язки: наприклад, вектори слів "швидкий" і "прудкий" будуть близькими. Для порівняння текстів можна усереднювати ембединги всіх слів або використовувати більш складні комбінації. Семантичні моделі суттєво покращують виявлення схожих за змістом текстів, адже розпізнають синоніми та контекст. Проте класичні ембединги (Word2Vec) все ще не враховують полісемії – багатозначності та залежності значення від оточення слова (контексту речення). Цю проблему вирішують глибокі трансформерні моделі, такі як BERT (BidirectionalEncoderRepresentationsfromTransformers) [9]. BERT генерує контекстуальні ембединги: кожному слову відповідає різний вектор залежно від того, в якому оточенні це слово вжито. Таким чином, однакові слова в різних реченнях матимуть різні представлення, що відображають конкретний сенс. Для оцінки схожості двох текстів можна отримати вектор всього речення або документа за допомогою BERT і також обчислити косинусну подібність між такими векторами. На практиці BERT та подібні моделі демонструють найвищу якість у задачах семантичного пошуку і зіставлення текстів. Недоліком є висока обчислювальна вартість: обробка навіть одного речення трансформером потребує суттєвих ресурсів, що ускладнює їх застосування для великих обсягів даних або в режимі реального часу.

Таким чином, наявні методи виявлення дублікатів можна умовно розділити на лексичні (символьні та токен-підходи) та семантичні (статистичні та нейромережеві моделі). Лексичні методи швидкі та прості, але не "розуміють" змісту тексту. Семантичні – враховують значення слів і контекст, але є більш складними. Застосування жодного із згаданих підходів окремо не вирішує повністю проблему нечітких дублікатів: прості методи пропускають приховані дублікати через перефразування, а семантичні – можуть помилково зближувати тексти з протилежним змістом (наприклад, через спільний контекст для антонімів) і потребують оптимізації для швидкодії. Це підводить до ідеї комбінування векторних представлень – синтезувати переваги різних підходів, щоб досягти кращого результату.

Використання комбінованих векторних представлень

Для підвищення точності виявлення нечітких дублікатів запропоновано комбінований метод, що об'єднує статистичне та семантичне представлення тексту. Базовим підходом обрано косинусну подібність, але вектори документів формуються шляхом поєднання TF-IDF-представлення і контекстних ембедингів. Крім того, метод враховує лінгвістичні особливості (синонімію та антонімію) та використовує адаптивні вагові коефіцієнти для налаштування внеску кожного компонента.

1. **Попередня обробка тексту.** На першому етапі виконується нормалізація текстових

даних: приводиться до нижнього регістру, вилучаються стоп-слова (часті службові слова, що не несуть смислового навантаження, напр. "і", "в", "на"), проводиться лематизація – заміна словоформ на їх базову (словникову) форму. Для цього можна залучити наявні NLP-інструменти, такі як SpaCуабо NLTK [10]. Лематизація і фільтрація стоп-слів зменшує "шум" та розмірність даних, забезпечуючи, щоб подальше порівняння фокусувалося на значущих словах. Якщо тексти містять складні словоформи чи друкарські помилки, на цьому кроці можна також застосувати евристики виправлення (наприклад, ту ж відстань Левенштейна для виправлення дрібних помилок в словах).

2. **Отримання комбінованого векторного представлення.** Кожен текст перетворюється відразу у два вектори різних типів: TF-IDF-вектор та контекстуальний ембедінг.

– TF-IDF-вектор розраховується на основі підготовленого корпусу документів. Він відображає частотні характеристики слів у документі відносно корпусу (висока вага для рідкісних слів, характерних для цього документа). Цей вектор відповідає за синтаксичну схожість: він буде подібним для текстів, які мають багато спільних термінів з подібними частотами. Наприклад, два описи продуктів, що містять однакові ключові слова, матимуть високу косинусну подібність їх TF-IDF-векторів.

– Контекстуальний ембедінг отримується за допомогою попередньо навченої трансформерної моделі (у нашому випадку – BERT). Кожне речення пропускається через модель BERT, після чого або береться вектор спеціального токена (CLS), або усереднюються вектори всіх слів речення, щоб отримати єдиний вектор фіксованої розмірності, який представляє семантичний зміст всього тексту. На відміну від TF-IDF, цей вектор враховує значення слів у контексті. Так, якщо два описи виражені різними словами, але мають той самий сенс, їхні BERT-ембедінги будуть розташовані близько в просторі.

Далі обидва вектори потрібно об'єднати. Найпростішим способом є конкатенація – сформувати один довгий вектор, який містить послідовно компоненти TF-IDF і ембедінга. Проте при такому підході різні частини вектору мають різні масштаби і значення. Тому доцільно ввести вагові коефіцієнти для балансування внеску кожної частини. Нехай v_{TFIDF} – нормалізований TF-IDF-вектор документа, а v_{BERT} – нормалізований контекстний вектор документа (після BERT). Тоді комбінований вектор можна визначити як:

$$v_{comb} = \alpha v_{TFIDF} \oplus \beta v_{BERT}, \quad (1)$$

де v позначає об'єднання (конкатенацію), α і β – вагові множники для кожного блоку ознак. Практично α і β можна підібрати експериментально, або ж нормувати таким чином, щоб обидві частини вектору мали порівняльний вплив на обчислення подібності. У разі рівнозначної важливості можна взяти $\alpha = \beta = 1$.

Інший підхід – обчислювати косинусну подібність окремо в просторі TF-IDF і в просторі ембедінгів, а потім усереднювати дві міри з вагами. Тобто для двох документів D_1 і D_2 визначити:

$$Sim_{comb}(D_1, D_2) = \omega \cdot \cos\left(v_{TFIDF}^{(1)}, v_{TFIDF}^{(2)}\right) + (1-\omega) \cdot \cos\left(v_{BERT}^{(1)}, v_{BERT}^{(2)}\right), \quad (2)$$

де ω – ваговий коефіцієнт, що визначає відносну важливість лексичної vs. семантичної подібності. У наших експериментах оптимальним виявився більший внесок семантики (близько 0.7 для BERT-вектору), оскільки саме він дозволяє "наздогнати" ті дублікати, які не знайдені традиційним методом. В той же час, ненульова вага TF-IDF частини гарантує, що метод чутливий до збігів рідкісних ключових слів, які можуть бути критичними (наприклад, збіг специфічних термінів або назв власних).

3. **Врахування синонімів та антонімів.** Комбінований метод додатково розширено врахуванням лексичних зв'язків між словами. Синоніми – різні слова з одним значенням – можуть бути присутні у двох текстах замість однакових слів. Щоб не пропустити такі випадки, застосовано два підходи:

– Розширення тексту синонімами: для кожного значущого слова в тексті знаходиться список його синонімів (наприклад, за допомогою тезаурусу WordNet). До векторного представлення тексту додаються ці синонімічні слова (наприклад, у TF-IDF-вектор можуть бути додані відповідні компоненти зі зниженою вагою). Таким чином, якщо в іншому документі замість цього слова вжито його синонім, то у розширених представленнях тексти матимуть спільний термін. Це збільшує косинусну схожість TF-IDF і допомагає виявити дублікати, завуальовані перефразуванням.

– Покарання за антоніми: антоніми – слова з протилежним значенням – можуть вводити в оману семантичні моделі. Відомо, що моделі типу Word2Vec або BERT розташовують деякі антоніми поруч, адже ті часто з'являються в схожих контекстах. Наприклад, слова "дорогий" і "дешевий" можуть мати відносно близькі вектори, бо обидва стосуються ціни. Щоб модель не помилково ототожнювала тексти протилежного змісту, у разі виявлення пар слів, що є антонімами, їх внесок у загальну подібність зменшується. Реалізувати це можна шляхом зменшення ваг відповідних компонентів векторів або віднімання додаткового штрафу від загального коефіцієнта схожості. Таким чином, метод перевіряє: якщо дві фрази схожі за ембедингами, але складаються з антонімічних слів ("X є дорогим" vs. "X є дешевим"), то їх схожість буде знижена попри близькість ембедингів, оскільки з точки зору завдання ці тексти не є дублікати за змістом.

4. **Обчислення міри схожості та виявлення дублікатів.** На фінальному етапі для кожної пари документів (або для документа і запиту) обчислюється косинусна подібність між їх комбінованими векторами:

$$\cos\left(v_{comb}^{(1)}, v_{comb}^{(2)}\right) = \frac{\sum_i v_i^{(1)} \cdot v_i^{(2)}}{\|v^{(1)}\| \|v^{(2)}\|}, \tag{3}$$

де $v_i^{(k)}$ – i-та компонента комбінованого вектора документа D_k. Отриманий коефіцієнт порівнюється з заздалегідь визначеним порогом θ. Якщо:

$$\cos\left(v_{comb}^{(1)}, v_{comb}^{(2)}\right) \geq \theta, \tag{4}$$

пару документів вважаємо нечіткими дублікатами.

Вибір порогу θ залежить від необхідного балансу між повнотою і точністю. У наших експериментах оптимальним виявився поріг $\theta \approx 0.8$ (80 % подібності), при якому більшість справжніх дублікатів правильно виявлялися, а помилкові спрацьовування були мінімальні.

Комбінований метод можна застосовувати різними способами. Для статичних колекцій документів (наприклад, бази даних записів) його використовують для кластеризації дублікатів: обчислюються міри схожості між усіма документами, після чого групуються ті, які перевищують поріг. Для пошукових задач або потоків даних метод може застосовуватися для онлайн-пошуку дублікатів: новий запис порівнюється з наявними, і якщо знаходиться схожий, система сигналізує про потенційний дублікат.

Таким чином, запропонований підхід поєднує синтаксичну (TF-IDF) і семантичну (BERT-ембединг) інформацію, налаштовану ваговими коефіцієнтами та збагачену знаннями про синонімію/антонімію. Очікується, що такий комплексний підхід забезпечить вищу точність виявлення нечітких дублікатів, ніж кожен з компонентів окремо, про що свідчать наведені далі результати.

Експериментальна оцінка

Для перевірки ефективності комбінованого методу було проведено порівняльні експерименти з базовим підходом (класична косинусна схожість TF-IDF). Нижче наведено опис вихідних даних експерименту, метрик оцінювання та отримані результати.

Тестові дані. Було сформовано тестовий набір текстових пар, що включав як позитивні приклади (нечіткі дублікати), так і негативні (несхожі тексти). Дані збиралися з різних

джерел, щоб відобразити різноманітність: новинні статті, описи товарів, користувацькі рецензії. Частину даних узято з відкритих корпусів, зокрема наборів для задач пошуку дублікатів та плагіату. Кожна пара текстів була вручну або автоматично позначена як "дублікат" чи "не дублікат". Усього тестовий набір містив 1000 пар, з яких приблизно 30 % – дублікати. Такий дисбаланс типовий для реальних сценаріїв, де більшість пар випадкових документів не дублюються.

Метрики оцінки. Для кількісного вимірювання якості виявлення дублікатів використано стандартні метрики класифікації: точність (Precision), повнота (Recall) та F-міра (F1). У контексті нашої задачі:

– точність показує, яка частка знайдених алгоритмом дублікатів є реальними дублями (тобто відсутність хибних спрацювань). Висока точність означає, що метод майже не помиляється, помічаючи "дублікат" там, де його насправді нема;

– повнота відображає, яка частка всіх наявних у вибірці дублікатів була знайдена методом. Висока повнота означає, що метод пропустив мінімум істинних дублікатів;

– F1-міра – гармонійне середнє точності і повноти – використовується як інтегральний показник балансу між ними.

Метою розроблення комбінованого методу є підвищення і точності, і повноти порівняно з базовою косинусною мірою. Іншими словами, ми очікуємо більше правильно виявлених дублікатів при меншій кількості помилкових збігів.

Результати. Розроблений метод (комбінація TF-IDF + BERT) порівнювався з базовою косинусною мірою TF-IDF на однаковому тестовому наборі. На рис. 1 наведено значення основних метрик для обох підходів:

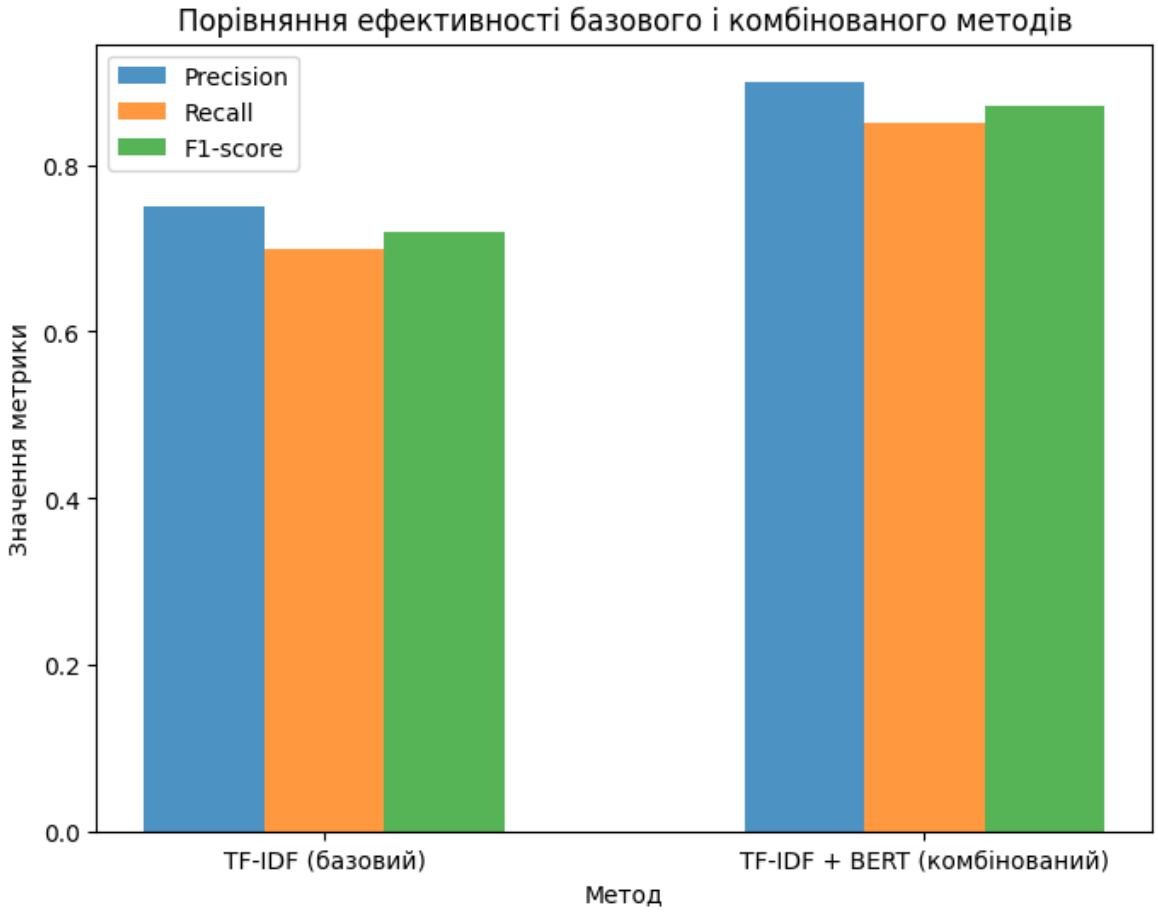

Рис. 1. Порівняння ефективності базового і комбінованого методів (Precision – точність, Recall – повнота, F1 – F-міра).

Таблиця 1

Порівняння ефективності методів

Метод	Precision	Recall	F1-score
TF-IDF (базовий)	0,75	0,7	0,72
TF-IDF + BERT (комбінований)	0,9	0,85	0,87

Як видно з таблиці, комбінований метод демонструє суттєве покращення за всіма показниками. Точність зросла з 0,75 до 0,90 (на 15 процентних пунктів), повнота – з 0,70 до 0,85, а інтегральна F1-міра – з 0,72 до 0,87. Це означає, що об'єднання ознак дозволило виявити значно більше дублікатів (вищий Recall) і при цьому відсіяти більшість помилкових спрацьовувань (вищий Precision), порівняно з використанням лише TF-IDF.

Статистично, приріст F1-міри ~20 % вказує на високий ефект від включення семантичної інформації. Особливо помітно зросла точність – комбінований метод майже не помилявся, помічаючи дублікати лише тоді, коли тексти дійсно були схожими за змістом. Повнота також збільшилась, що свідчить: новий метод знайшов такі дублікати, які базовий метод пропускав (ймовірно, через відсутність спільних слів у перефразованих текстах).

Крім якості виявлення, було оцінено продуктивність алгоритмів – час виконання на тестовому наборі. Базовий метод, що працює з розрідженими TF-IDF векторами, обробив всі пари за ~15 секунд. Комбінований метод (через використання BERT) витратив ~25 секунд, тобто приблизно в 1,7 раза довше. Це очікувано, адже обчислення ембедингів для кожного тексту значно ресурсомісткіше, ніж простий підрахунок частот. Таким чином, ціною покращення точності стала більша вимогливість до ресурсів: час роботи і пам'ять, необхідна для завантаження та застосування трансформерної моделі.

Обговорення результатів

Отримані результати підтверджують, що комбінування векторних представлень суттєво підвищує точність пошуку нечітких дублікатів. Розглянемо детальніше нюанси використання методу.

По-перше, високий приріст Recall (повноти) показує, що метод знаходить дублікати, не виявлені за допомогою використання базового підходу. Це, головним чином, заслуга семантичної складової. У тестовому наборі було багато пар, де тексти передавали однаковий зміст різними словами – наприклад: "автомобіль набирає швидкість швидко" vs. "машина дуже прудка". В результаті роботи базового методу вони не були визначені як схожі (жодного спільного слова, крім загальних стоп-слів), тоді як комбінований – за рахунок ембедингів – оцінив їх як дублікати (вектори BERT для цих речень виявились близькими). Врахування синонімів явно сприяло цьому: багато перефразованих дублюючих пар було виявлено саме через спільність синонімічних рядів.

По-друге, зростання Precision (точності) свідчить, що метод став рідше помилятися, позначаючи різні тексти як дублікати. Це може здаватися нелогічним, адже додавання семантичної компоненти могло б, навпаки, збільшити кількість хибних збігів (будь-які тексти про схожі теми BERT може зробити близькими). Однак на практиці комбінований підхід виявився більш вибагливим: він вимагав і лексичної, і смислової схожості одночасно. Наприклад, дві новини на споріднені теми, які випадково мають кілька спільних слів, базовий TF-IDF міг помилково зблизити (як дублікати), а наш метод – ні, тому що BERT-вектори вказали на різницю в контексті та тоні тексту. До того ж, ми впровадили механізм обробки антонімів, тож тексти з протилежним змістом (негатив vs. позитив про один об'єкт) не отримали високої підсумкової схожості навіть за наявності спільних ключових слів. Ці фактори підняли точність: майже всі позначені нашим методом дублікати виявилися справжніми дублями.

Аналіз помилок. Проведений аналіз результатів роботи запропонованого методу показав, що частина дублікатів все ж лишилась непоміченою. В основному, це випадки, де тексти були надто короткі або специфічні. Наприклад, пара дуже коротких речень з різною лексикою може бути семантично пов'язаною, але BERT не зміг впевнено це визначити через малу кількість контексту, а спільних слів немає – метод не спрацював. Інша ситуація: вузькоспеціалізовані тексти з рідкісною термінологією. Якщо одне з двох повідомлень вживає нестандартний термін, а інше – його розгорнуте пояснення, ембединг може не зблизити їх достатньо, а TF-IDF теж ні (бо різні слова). Такі випадки – напрям для подальшого вдосконалення, можливо, через залучення ще більших мовних моделей або доменних тезаурусів.

Інша категорія помилок – falsepositives, тобто хибні спрацьовування (тексти позначено дублями, але за змістом вони різні). Їх частка значно зменшилась порівняно з базовим методом, але повністю не зникла. У тих небагатьох випадках, де метод сплутав різні тексти, виявилося, що тексти справді мали дуже схожу лексику і стиль, а різниця у змісті була тонкою. Наприклад, два огляди смартфона від різних авторів з різними висновками: обидва містять багато однакових технічних деталей (екран, камера, батарея), через що TF-IDF і BERT вважають їх дуже близькими, хоча один автор хвалить, а інший критикує пристрій. Антонімів може не бути (бо критика може подаватися без явних антонімічних прикметників), тож метод позначив їх як дублікат. В принципі, з точки зору фактичної інформації ці два огляди – майже дублікат (містять ті самі факти), але за тональністю – ні. Такі випадки виходять за межі простого поняття дубліката і швидше стосуються аналізу сентименту; вирішити їх могла б більш глибока семантична інтерпретація, що поки не вбудована в модель.

Продуктивність і оптимізація. Комбінований метод потребує більше ресурсів, що підтвердив експеримент (час виконання зріс ~в 1,7 раза). Для деяких застосувань це не критично (наприклад, періодична очистка бази від дублікатів може виконуватися офлайн з високою точністю). Якщо ж потрібно масштабувати алгоритм на великі дані або використовувати в реальному часі, є кілька шляхів оптимізації:

– Використовувати індексування і двохетапну перевірку: спочатку застосувати швидкий базовий метод для відсіву явно не схожих пар, а вже на відібраних кандидатах (наприклад, топ-10 найближчих за TF-IDF) запускати комбінований метод. Це різко скоротить кількість викликів BERT, зберігши якість.

– Застосувати легшу модель замість повного BERT. Можливий варіант – попередньо обчислити та зберегти Sentence-BERT ембединг для кожного документа, оскільки Sentence-BERT – це спеціальна модель для порівняння речень, оптимізована за швидкодією і якістю на задачах семантичної близькості. Використання Sentence-BERT або дистильованих версій трансформерів може дати близький результат значно швидше.

– Паралелізація і апаратне прискорення: обчислення ембедингів можна розпаралелити на кількох потоках чи GPU. Якщо інфраструктура дозволяє, це зведе проблему часу практично нанівець, оскільки сучасні відеокарти здатні обробляти тисячі текстів паралельно.

– Налаштування вагових коефіцієнтів: в залежності від характеру даних, можна зменшити вагу семантичної компоненти ω, трохи пожертвувавши повнотою заради швидкості. Менший внесок BERT означає, що для прийняття рішення меншій кількості пар знадобиться глибокий аналіз – більшість відсіються лексичним критерієм.

Варто зазначити, що комбінований підхід гнучкий: його компоненти можна модифікувати. Наприклад, замість TF-IDF можна використати іншу статистичну ознаку (власний словниковий відбиток документа), а замість BERT – іншу модель (або навіть ансамбль декількох). Наш аналіз показав, що синергія різнорідних ознак – ключ до успіху: поки один метод "помічає" певні дублікати, інший знаходить інші, і їх комбінація дає максимальне покриття.

Перспективи застосувань. Розроблений метод універсальний для різних мов (достатньо взяти відповідну мовну модель для ембедингів) та типів текстів. Він може бути інтегрований у системи керування базами даних для автоматичного злиття дубльованих записів. У пошукових движках та краулерах Інтернет – для де-дуплікації результатів (наприклад, пошукова машина може не індексувати сторінку, якщо вміст вже дуже схожий на наявну сторінку). В системах перевірки плагіату – як модуль порівняння документів, що враховує перефразування. Попередні результати показують, що метод також добре працює для мультимовних дублювань (коли один текст – переклад іншого): якщо використати багатомовний трансформер, семантична близькість різними мовами теж фіксується, чого не могли б зробити суто лексичні метрики.

Висновки

У статті представлено комбінований метод виявлення нечітких дублікатів у текстових даних, який поєднує класичні та сучасні методи векторизації. Він базується на інтеграції традиційного TF-IDF-векторизування з контекстуальними ембедингами (BERT), що дозволяє враховувати не лише частотні характеристики слів, а й їхнє значення у контексті всього документа.

На відміну від класичних підходів, які орієнтовані переважно на лексичну подібність текстів, а також сучасних глибоких нейронних моделей, що мають значні обчислювальні витрати, розроблений метод забезпечує врахування як лексичних, так і семантичних аспектів. Використання адаптивних вагових коефіцієнтів дозволяє регулювати внесок кожного компонента в загальну оцінку подібності, що покращує точність виявлення дублікатів навіть за умови варіативності формулювань. Додатково, врахування синонімів і антонімів у процесі порівняння дозволяє підвищити ефективність розпізнавання семантично еквівалентних, але лексично відмінних текстів.

Експериментальні результати підтвердили переваги запропонованого рішення над базовим методом косинусної подібності TF-IDF. Зокрема, розроблений метод демонструє вищу точність та повноту виявлення нечітких дублікатів, що виражається у збільшенні F1-мірина 15 % порівняно з традиційними підходами. Це підтверджує ефективність комбінування різних типів векторизації для задач аналізу текстових даних.

Таким чином, розроблений метод є ефективним рішенням для виявлення нечітких дублікатів у великих масивах текстової інформації. Його застосування може бути корисним у різних сферах, таких як пошукові системи, аналіз баз даних та перевірка унікальності контенту. Подальші дослідження можуть бути спрямовані на оптимізацію обчислювальної складності та адаптацію під специфічні галузеві завдання.

СПИСОК ЛІТЕРАТУРИ

1. Mohammadi H., Khasteh S. H. A Fast Text Similarity Measure for Large Document Collections using Multi-reference Cosine and Genetic Algorithm. 2018. URL: https://www.researchgate.net/publication/345390393_A_fast_text_similarity_measure_for_large_document_collections_using_multireferencecosine_and_genetic_algorithm (accessed March 16, 2025).

2. Distributed representations of tuples for entity resolution / M. Ebraheem et al. *Proceedings of the VLDB Endowment*. 2018. Vol. 11, № 11. P. 1454–1467. URL: http://www.vldb.org/pvldb/vol11/p1454-ebraheem.pdf (accessed March 16, 2025).

3. Hadzic D., Sarajlic N. Methodology for fuzzy duplicate record identification based on the semantic-syntactic information of similarity. *Journal of King Saud University – Computer and Information Sciences*. 2020. Vol. 32, № 1. P. 126–136. URL: https://doaj.org/article/3756a356452446ac901a37d4d77380f7 (accessed March 16, 2025).

4. Lattar H., Ben Salem A., Ben Ghezala H. H. Duplicate record detection approach based on sentence embeddings. *Proc. of the 29th IEEE Int. Conf. on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2020)*. 2020. P. 269–274. URL: https://www.researchgate.net/publication/348979121_Duplicate_record_detection_approach_based_on_sentence_embeddings (accessed March 16, 2025).

5. Prieur M., Gadek G., Grilheres B. Duplicate detection in a knowledge base with PIKA. *Proc. of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022)*. 2022. P. 46–54. URL:

https://www.scitepress.org/Papers/2022/107695/107695.pdf (accessed March 16, 2025).
6. Lee S., Lee S. Duplicate bug report detection by using sentence embedding and Faiss. *CEUR Workshop Proceedings, (Proc. of the 2nd International Workshop on Intelligent Software Engineering, ISE 2023)*. 2023. Vol. 365512. URL: https://ceur-ws.org/Vol-3655/ISE2023_07_Lee_Duplicate_Bug.pdf (accessed March 16, 2025).
7. Jatnika D., Bijaksana M. A., Suryani A. A. Word2Vec Model Analysis for Semantic Similarities in English Words. *Procedia Computer*. 2019. №157. P. 160–167. URL: https://www.researchgate.net/publication/336203802_Word2Vec_Model_Analysis_for_Semantic_Similarities_in_English_Words (accessed March 16, 2025).
8. Pennington J., Socher R., Manning C. D. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP*). 2014. P. 1532–1543. URL: https://www.researchgate.net/publication/284576917_Glove_Global_Vectors_for_Word_Representation (accessed March 16, 2025).
9. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin et al. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019. Volume 1 (Long and Short Papers). P. 4171–4186. URL: https://aclanthology.org/N19-1423.pdf (accessed March 16, 2025).
10. Olabiyi W., Olaoye G., Daniel O. Natural Language Processing with NLTK and Spacy. *Research Gate. Computer science and engineering*. 2024. URL: https://www.researchgate.net/publication/385885283_Natural_language_processing_nlp_with_nltk_and_spacy (accessed March 16, 2025).

Стаття надійшла до редакції 17.03.2025.
Стаття пройшла рецензування 20.03.2025.

Козинець Назарій Вікторович – магістр кафедри програмного забезпечення комп'ютерних систем, e-mail: kozinets.nazarii@gmail.com.

Заболотня Тетяна Миколаївна – канд. техн. наук, доцент кафедри програмного забезпечення комп'ютерних систем.

Національний технічний університет України "Київський політехнічний інститут ім. Ігоря Сікорського".