МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерних наук

ШАНДРОВИЧ Юрій Ігорович

Математичне та програмне забезпечення класифікації документів в інформаційних системах / Mathematical Tools and Software for Documents Classification in Information Systems

спеціальність: 121 - Інженерія програмного забезпечення
освітньо-професійна програма - Інженерія програмного забезпечення

Кваліфікаційна робота

Виконав студент групи ІПЗм-21
Ю. І. Шандрович

_____

Науковий керівник:
к.т.н., доцент Є. О. Марценюк

_____

Кваліфікаційну роботу
допущено до захисту:

"___" _____ 20___ р.

Завідувач кафедри
_____ **А. В. Пукас**

ТЕРНОПІЛЬ - 2022

ЗМІСТ

ВСТУП

Актуальність теми. Протягом останніх років спостерігається суперечність між швидкістю зростання даних, які доступні людству та можливостями їх інтелектуальної обробки. Більшість знань, якими оперує людство, зберігається у вигляді текстових документів природними мовами, які не супроводжуються додатковими засобами розмітки для інструментів автоматизованої інтелектуальної обробки текстових даних.

Таким чином, експоненційне зростання кількості інформації у багажі знань людства стикається з неможливістю ефективно обробляти їх. Для вирішення цієї суперечності існують системи автоматичної обробки природно-мовних даних та формування баз знань на їх основі. Більшість алгоритмів інтелектуальної обробки даних оперують числовими даними, таким чином базовим завданням будь-якого процесу роботи з текстами природною мовою є представлення текстових одиниць у чисельному вигляді, а у разі маніпулювання об'єктами баз знань – концептами. Базовою та найважливішою частиною таких систем обробки даних, формування баз знань на основі результатів обробки текстів є векторне представлення окремих слів природної мови, семантичних одиниць, а також функція для оцінки ступеня схожості цих об'єктів, які потім і формують концепти.

Від властивостей таких об'єктів та їх якості, тобто від того наскільки близькі або далекі такі вектори у своєму векторному просторі для близьких і далеких за змістом слів один від одного, залежить те, наскільки ефективно працюватимуть алгоритми машинного навчання, що їх використовують. Фактично це базове завдання під час вирішення проблем автоматичної обробки природно-мовних текстів.

Формально завдання може бути поставлене у такому вигляді. Для даного лексикону, тобто набору слів, по множині текстів необхідно скласти відображення слова в лексиконі в деякий вектор. При цьому на цільовому

векторному просторі задана метрика, яка тим менша для двох векторів, чим ближче за змістом два вихідні слова з лексикону. Призначення таких алгоритмів – математичне представлення семантики слів природної мови, яке зручне до роботи інших алгоритмів обробки текстів.

На сьогоднішній день існує велика кількість алгоритмів, що вирішують це завдання. Тим не менш, постійно зростаючий обсяг інформації, представлений у текстовому вигляді, а також зростання вимог до систем інтелектуальної обробки текстів, постійно ставлять нові виклики перед науковим та інженерним співтовариством у пошуку оригінальних рішень цього завдання. Зокрема не так давно, близько десяти років тому, з'явився напрямок, що активно розвивається на стику квантової теорії та інформаційного пошуку, званий Quantum Information Retrieval, яке прагне поєднувати математичні моделі і формалізм квантової теорії з інформаційним пошуком. У тому числі, в рамках інформаційного пошуку, у цьому напрямі вирішується завдання векторизації текстових даних.

Зв'язок роботи з науковими програмами, планами, темами

Напрямок виконаних досліджень безпосередньо пов'язаний з науково-дослідним напрямком кафедри “комп’ютерних наук” Західноукраїнського національного університету.

Мета і задачі дослідження

Метою роботи є розробка методів та засобів класифікації документів в інформаційних системах.

Відповідно до поставленої мети у роботі потрібно вирішити такі основні завдання дослідження:

1. Дослідити відомі підходи та методи до класифікації текстових документів.

2. Дослідити особливості застосування моделей і методів квантової теорії для обробки текстових документів.

3. Розробити метод векторизації слів з використанням апарату квантової теорії ймовірностей для представлення концептів бази знань.

4. Розробити систему класифікації документів в інформаційних системах, яка базується на використанні алгоритму Белла.

5. Реалізувати запропоновані методи та провести відповідні експериментальні дослідження.

Об'єкт дослідження – класифікація документів в інформаційних системах.

Предмет дослідження – методи та програмні засоби для класифікації текстових документів з використанням формалізму квантової теорії.

Методи дослідження

В роботі використовувалися методи квантової теорії ймовірностей, математичної оптимізації, машинного навчання.

Наукова новизна одержаних результатів

Запропоновано метод оцінки схожості фрагментів неформалізованих текстових документів, який відрізняється від відомих використанням апарату квантової теорії ймовірностей, зокрема тесту Белла на основі асиметричної HAL-матриці, що забезпечує можливість класифікації текстових документів в інформаційній системі.

РОЗДІЛ 1
ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ ДО КЛАСИФІКАЦІЇ ТЕКСТОВИХ ДОКУМЕНТІВ

1.1. Особливості класифікації текстових документів.

Класифікація текстів є одним з основних завдань комп'ютерної лінгвістики, оскільки до неї зводиться ряд інших завдань: визначення тематичної приналежності текстів, автора тексту, емоційного забарвлення висловлювань та ін. Для забезпечення інформаційної та суспільної безпеки велике значення має аналіз у телекомунікаційних мережах контенту, що містить протиправну інформацію (у тому числі дані, пов'язані з тероризмом, наркоторгівлею, підготовкою протестних рухів чи масових заворушень).

Широко відомий сучасний підхід до класифікації ґрунтується на методах машинного навчання. У цьому розділі описуються найбільш поширені алгоритми побудови класифікаторів, проведені з ними експерименти та результати цих експериментів.

Прогрес в галузі мікроелектроніки та інформаційних технологій зумовив широке поширення обробки в реальному часі великих потоків даних. Наприклад, багато простих операції повсякденного життя, такі як використання кредитної картки або телефону, вимагають автоматизованого створення, аналізу та обробки різних даних. Оскільки ці операції часто виконуються великою кількістю учасників, необхідні розподілені та масові потоки даних. Так само соціальні мережі містять велику кількість специфічних мережевих і текстових потоків даних. Тому актуальна проблема створення моделей та алгоритмів, що дозволяють ефективно обробляти великі потоки даних, особливо в умовах обмежених часових та інших ресурсів.

Для забезпечення інформаційної та суспільної безпеки важливе значення має аналіз у телекомунікаційних мережах контенту, що містить протиправну інформацію (у тому числі даних, пов'язаних з тероризмом, наркоторговлею, мережевим екстремізмом, підготовкою протестних рухів або масових заворушень) .

1.2. Аналіз відомих методів класифікації документів

Цілями даного огляду є порівняння сучасних методів вирішення задачі класифікації текстів, виявлення тенденцій розвитку даного напряму, а також вибір найкращих алгоритмів для застосування в дослідницьких і комерційних завданнях.

Методи класифікації текстів лежать на стику двох областей - інформаційного пошуку та машинного навчання. Їх схожість полягає у способах представлення самих документів та способах оцінки якості алгоритмів. На сьогоднішній день розроблено велику кількість методів та їх різних варіацій для класифікації текстів. Кожна група методів має свої переваги та недоліки, сфери застосування, особливості та обмеження.

Особливий інтерес представляє випадок, коли дані надходять у вигляді потоку, наприклад, у телекомунікаційних мережах. Певні труднощі виникають через те, що навчання моделі завжди грунтується на сукупності властивостей набору документів. Ці сукупні властивості можуть змінюватися з часом, і при побудові струмового класифікатора необхідно враховувати можливі зміни вихідного розподілу даних [1-5]. Бажано, щоб обраний метод міг підтримувати інкрементне навчання, тобто щоб класифікатор навчався на кожному окремо взятому зразку в режимі реального часу. При інкрементному навчанні навчальні приклади надходять послідовно в процесі роботи алгоритму, тому класифікатор повинен постійно коригувати

результати навчання і донавчатись. При неінкрементному навчанні вся навчальна вибірка надається одночасно повністю. Зрозуміло, що у разі інкрементного навчання поведінка класифікатора в процесі роботи змінюється, що зменшує його передбачуваність і може ускладнити налаштування системи. У той же час інкрементне навчання робить систему набагато більш гнучкою, що адаптується до умов, що змінюються.

Особливості процесу класифікації у потоці пов'язані ще з тим, що не завжди вдається контролювати швидкість надходження даних. Деякі класи документів можуть зустрічатися у потоці лише іноді. Виявити цей рідкісний клас буває непросто, і класифікація текстів у таких випадках стає надзвичайно складною задачею.

Порівняння методів побудови класифікації є досить складним завданням, що різні вхідні дані можуть приводити до різних результатів. Тому необхідно здійснити їх програмну реалізацію та обчислення ефективності на однакових наборах документів для навчання та тестування.

На рисунку 1.1 представлена загальна схема процесу класифікації. Розглянемо кожен із його етапів. Попередня обробка та індексація документів.

Попередня обробка тексту включає в себе токенізацію, видалення функціональних слів (семантично нейтральних слів, таких як прийменники, артиклі та ін.). Далі здійснюється морфологічний аналіз (виробляються розмітка частинами мови та стематизація). Це дозволяє значно скоротити розмірність простору.

В результаті як ознаки документа виступають всі значущі слова, що зустрічаються в документі.

Індексація документів - це побудова деякої числової моделі тексту, яка перекладає текст у зручне для подальшої обробки представлення.

Наприклад, модель "мішка слів" (bag-of-words) дозволяє подати документ у вигляді багатомірного вектора слів та їх ваг у документі [2-6].

Іншими словами, кожен документ – це вектор у багатовимірному просторі, координати якого відповідають номерам слів, а значення координат - значення ваги.

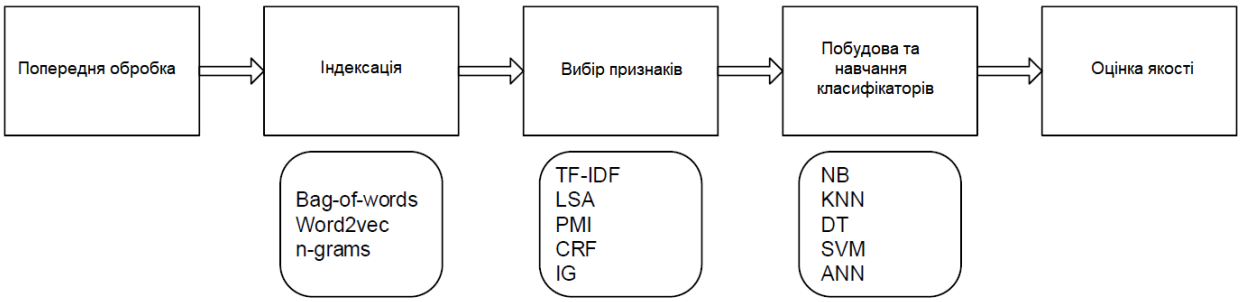

Рис. 1.1. Етапи процесу автоматичної класифікації текстів

Інша поширена модель індексації - Word2vec [7-9]. Вона представляє кожне слово в вигляді вектора, який містить інформацію по контекстних (супутніх) словах.

Ще одна модель індексації заснована на обліку n-грам [2], тобто послідовностей із сусідніх символів. Очевидно, що для навчальних та тестових документів повинен бути застосовуватися той самий метод індексації.

Зменшення розмірності простору ознак. Обчислювальна складність різних методів класифікації безпосередньо залежить від розмірності простору ознак. Тому для ефективної роботи класифікатора часто прибігають до скорочення числа використовуваних ознак (термінів).

За рахунок зменшення розмірності простору термінів можна знизити ефект перенавчання - явище, при якому класифікатор орієнтується на випадкові чи помилкові характеристики навчальних даних, а не на важливі та значущі.

Перенавчений класифікатор добре працює на тих примірниках, на яких він навчався, і значно гірше тестових даних. Щоб уникнути тиснути перенавчання, кількість навчальних прикладів має бути пропорційно числу використовуваних термінів. У деяких випадках скорочення розмірності

простору ознак у 10 разів (і навіть в 100) може призводити лише до незначного погіршення роботи класифікатора.

Існують кілька способів визначення ваги ознак документа. Найбільш поширений – обчислення функції TF-IDF [9-13]. Його основна ідея полягає в тому, щоб більша вага отримували слова з високою частотою в межах конкретного документа і з низькою частотою згадувань в інших документах.

Для зменшення розмірності простору термінів також застосовують латентно-семантичний аналіз (LSA), що використовує сингулярне розкладання матриць [12], поточкову взаємну інформацію (PMI) (різновид асоціативної міри), умовні випадкові поля (CRF) (узагальнення прихованої марківської моделі). Зустрічаются дослідження, в яких застосовуютьються статистичні критерії та відносна ентропія для ймовірнісних розподілів, називається коефіцієнтом посилення інформації, або дивергенцією Кульбака-Лейблера.

Побудова та навчання класифікатора з потужністю методів машинного навчання. Можна, можливо виділити такі методи класифікації:

- імовірнісні (наприклад NB [12]);
- метричні (наприклад, KNN [12-14]);
- логічні (наприклад DT [6, 15]);
- лінійні (наприклад SVM [4-9];
- логістична регресія [16]);
- методи на основі штучних нейронних мереж (наприклад FFBP [17], RNN [18], DAN2 [18], CNN [19]).

Далі узагальнено описуються ці методи, вказуються переваги та недоліки кожного з них.

Метод Байєса (Naive Bayes, NB) відноситься до імовірнісних методів класифікації.

Переваги методу:

- можливість оновлення навчальної вибірки без перенавчання класифікатора;

- стійкість алгоритму до аномальних викидів у вихідних даних;

- відносно проста програмної реалізації алгоритму;

- легка інтерпретованість результатів роботи алгоритму;

- хороше навчання у випадку з лінійно нероздільними вибірками.

Недоліки методу:

- репрезентативність набору даних, використовуваного для алгоритму;

- висока залежність результатів класифікації від обраної метрики;

- велика тривалість роботи через необхідність повного перебору навчальної вибірки;

- неможливість вирішення завдань великого розмірності за кількістю класів та документів.

Метод дерев рішень (Decision Trees, DT) відноситься до логічних методів класифікації.

Деревом рішень називають ациклічний граф, за яким провадиться класифікація об'єктів (у разі текстових документів), описаних набором ознак. Кожен вузол дерева містить умову розгалуження по одному з ознак. У кожного вузла стільки розгалужень, скільки значень має обрану ознаку.

У процесі класифікації здійснюються послідовні переходи від одного вузла до іншого в відповідно до значень ознак об'єкта.

Класифікація вважається завершеною, коли досягнутий один з кінцевих вузлів дерева.

Значення цього листа визначить клас, якому належить аналізований об'єкт. На практиці зазвичай використовують бінарні дерева рішень, у яких прийняття рішення переходу по ребрам здійснюється простою перевіркою наявності ознаки у документі. Якщо значення ознаки менше певного значення, вибирається одна гілка, якщо більше чи одно, інша.

На відміну від інших підходів, представлених раніше, підхід, що використовує дерева рішень, відноситься до символьних (тобто нечислових) алгоритмів.

Коли говорять про вибір найбільш відповідної ознаки, як правило, мають на увазі частотну ознаку, тобто будь-яку ознаку тексту, що допускає можливість знаходження частоти його появи в тексті. Найкращою для поділу є ознака, що дає максимальну на цьому кроці інформацію про категорії. Такою ознакою для тексту може бути, наприклад, ключове слово. З цієї точки зору будь-яку частотну ознаку можна вважати змінною. Тоді вибір між двома найбільш підходящими ознаками зводиться до оцінки ступеня пов'язаності двох змінних. Тому для вибору відповідної ознаки на практиці застосовують різні критерії перевірки гіпотез, тобто критерії кількісної оцінки ступеня пов'язаності двох змінних, поставлених у взаємну відповідність, де 0 відповідає повній незалежності змінних, а 1 - їх максимальної залежності.

Для дослідження зв'язку між двома змінними зручно використовувати представлення спільного розподілу цих змінних у вигляді таблиці спряженості (факторної таблиці, або матриці частот появи ознак). Вона є найбільш універсальним засобом вивчення статистичних зв'язків, тому що в ній можуть бути представлені змінні з будь-яким рівнем виміру. Таблиці сполученості часто використовуються для перевірки гіпотези про наявність зв'язку між двома ознаками за допомогою різних статистичних критеріїв: критерію Фішера (точного тесту Фішера), критерію згоди Пірсона (критерія хі-квадрат), критерію Крамера, критерію Стьюдента (t -критерія Стьюдента) та ін.

Переваги методу:

- відносно проста програмної реалізації алгоритму;

- легка інтерпретованість результатів роботи алгоритму.

Недоліки методу: нестійкість алгоритму по відношенню до викидів у вихідних даних і великий обсяг даних для отримання точних результатів.

Метод опорних векторів (Support Vector Machine, SVM) є лінійним методом класифікації. Нині цей метод вважається одним із найкращих. Розглянемо множину документів, які необхідно розкласифікувати. Порівняємо йому множину точок у просторі розмірності $|D|$.

Вибірку точок називають лінійно розділеною, якщо точки, що належать різним класам, можна розділити за допомогою гіперплощини (у двомірному випадку гіперплощиною є пряма лінія). Очевидний спосіб розв'язання завдання у такому разі – провести пряму так, щоб з одного боку від неї лежали всі точки одного класу, а з іншого – всі точки іншого класу. Тоді для класифікації невідомих точок досить буде подивитися, з якого боку прямої вони виявляться.

У загальному випадку можна провести нескінченну множину гіперплощин (прямих), що задовольняють нашій умові. Ясно, що найкраще вибрати пряму, максимально віддалену від наявних точок. У методі опорних векторів відстанню між прямою і множиною точок вважається відстань між прямою і найближчою до неї точкою з множини. Саме така відстань і максимізується у цьому методі. Гіперплощина, що максимізує відстань до двох паралельних гіперплощин, називається розділяючою (на рисунку 1.2 позначена буквою L). Найближчі до паралельних гіперплощин точки називаються опорними векторами (рисунок 1.2), через них проходять пунктирні лінії. Іншими словами, алгоритм працює в припущенні, що, чим більша різниця або відстань між цими паралельними гіперплощинами, тим менше буде середня помилка класифікатора, так як максимізація зазору між класами сприяє більш впевненій класифікації.

Насправді структура даних найчастіше буває невідома і дуже рідко вдається побудувати роздільну гіперплощину, отже, неможливо гарантувати лінійну роздільність вибірки. Можуть існувати такі документи, які алгоритм віднесе до одного класу, а насправді вони повинні ставитися до протилежного. Такі дані називаються викидами, вони створюють похибку методу, тому краще їх ігнорувати. У цьому полягає суть проблеми лінійної нероздільності.

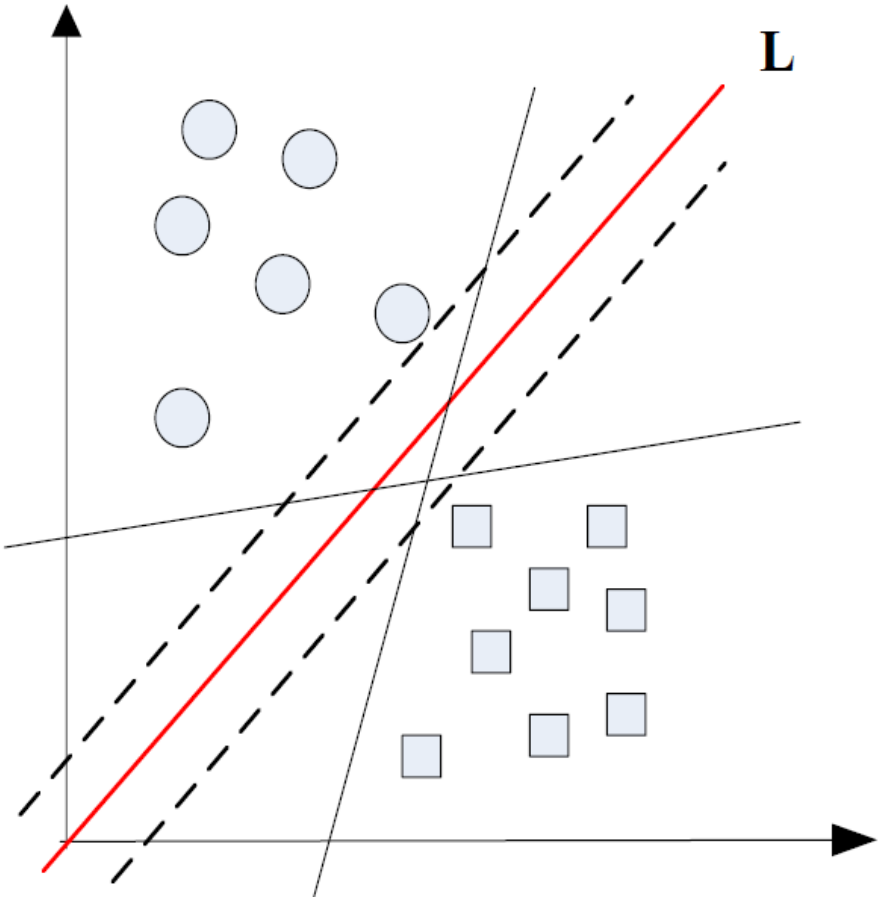

Рис. 1.2. Роздільна гіперплощина у методі опорних векторів

Вибірку називають лінійно нероздільною, з точки, що належать різним класам, не можна розділити за допомогою гіперплощини. Коли такої розділяючої гіперплощини не існує, необхідно перейти від вихідного простору признаків документів до нового, у якому навчальна вибірка виявиться лінійно розділеною. Для цього кожен скалярний добуток необхідно замінити на деяку функцію, що відповідає визначеним вимогам. Наприклад, можна призначати якийсь штраф за кожен неправильно розкласифікований документ. Цю функцію називають ядром. Заміна скалярного добутку функцією-ядром дозволяє перейти до іншого простору ознаків, де дані вже будуть розділені.

У разі лінійної нероздільності проблема пошуку оптимальної роздільної гіперплощини зводиться до завдання, еквівалентного пошуку сідлової точки функції Лагранжа з умовами доповнюючу нежорсткості.

Отримана система рівнянь вирішується методами квадратичного програмування. Це вже суто обчислювальне завдання. Цей варіант алгоритму називають алгоритмом з м'яким зазором (soft-margin SVM), тоді як у чисельному випадку говорять про жорсткий зазор (Hard-Margin SVM).

Переваги методу:

- один із найбільш якісних методів;

- можливість роботи з невеликим набором даних для навчання;

- зведення до завдання опуклої оптимізації, має єдине рішення.

Недоліки методу: складна інтерпретація параметрів алгоритму і нестійкість по відношенню до викидів у вихідних даних.

Логістична регресія (logit model, logistic regression) є лінійним методом класифікації. Цей метод використовується для прогнозування ймоовірності виникнення деякої події щодо значення множини ознак.

Переваги методу:

- є одним із найбільш якісних;

- підтримує інкрементне навчання;

- має відносно просту програмну реалізацію алгоритму.

Недоліки методу: складна інтерпретація параметрів алгоритму і нестійкість по по відношенню до викидів у вихідних даних.

Методи на основі штучних нейронів мереж (рисунок 1.3). Існує велика кількість різновидів нейронних мереж, основні з них – мережі прямого поширення, рекурентні мережі, радіально-базисні функції та самоорганізуючі карти. Налаштування ваг може бути фіксованим або динамічним.

У класичних нейронних мережах (Feed Forward Back Propagation, FFBP) присутні вхідний шар, вихідний шар і проміжні шари: сигнал йде послідовно від вхідного шару нейронів по проміжним шарам до вихідного. Прикладом таких структур є багатошаровий перцептрон.

Для класифікації документа за допомогою нейронної мережі прямого поширення ваги ознак документа подаються на відпоні входи мережі. Активація поширюється по мережі; значення, що виходять на виходах, і є

результат класифікації. Стандартний метод навчання такої мережі – метод зворотного розповсюдження помилки. Суть його в наступному: якщо на одному з виходів для одного з навчальних документів отримано неправильну відповідь, то помилка розповсюдження прямує назад по мережі і ваги ребер змінюються так, щоб зменшити помилку.

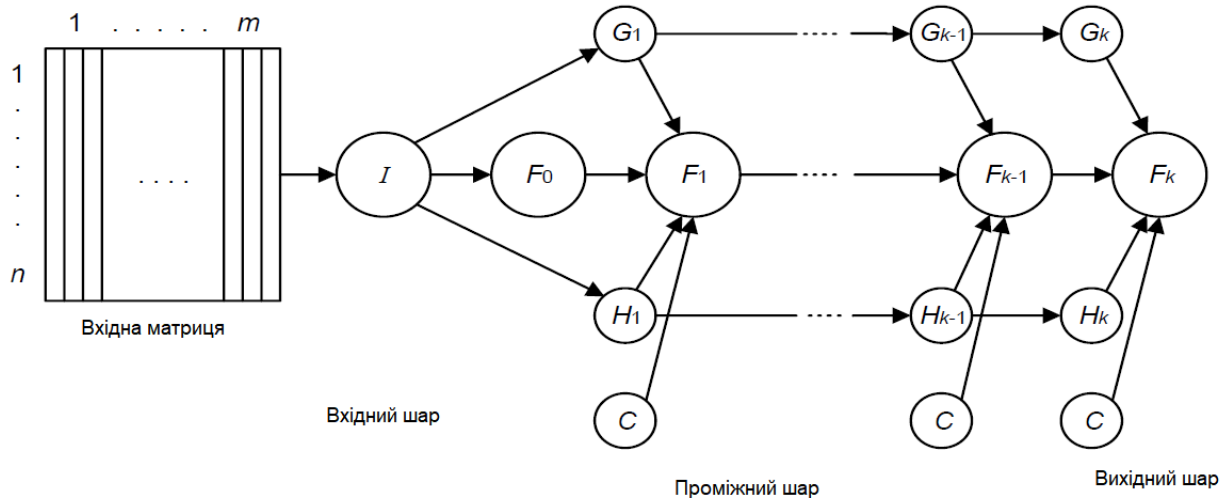

Рис. 1.3. Підхід на основі нейронних мереж

Згорткова нейронна мережа – односпрямована багатошарова мережа із застосуванням операції згортки, при якій кожен фрагмент вхідних даних множиться на матрицю (ядро) згортки поелементно, а результат підсумовується і запиується в аналогічну позицію вихідних даних.

Узагальнена схема CNN наведена на рисунку 1.4.

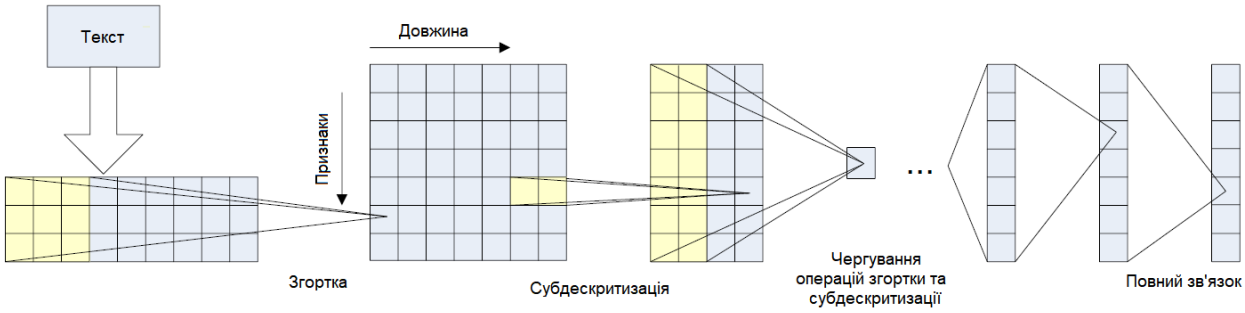

Рис. 1.4. Згорткова нейронна мережа

Рекурентна нейронна мережа виходить з багатошарового перцептрону введенням зворотних зв'язків. Одна з широко поширених різновидностей рекурентних нейронних мереж – мережа Елмана – зображена на рисунку 1.4. У ній зворотні зв'язки йдуть не від виходу мережі, а від виходів внутрішніх нейронів. Це дозволяє врахувати передісторію процесів і накопичити інформацію для вироблення правильної стратегії навчання. Головною особливістю рекурентних нейронних мереж є запам'ятовування послідовностей.

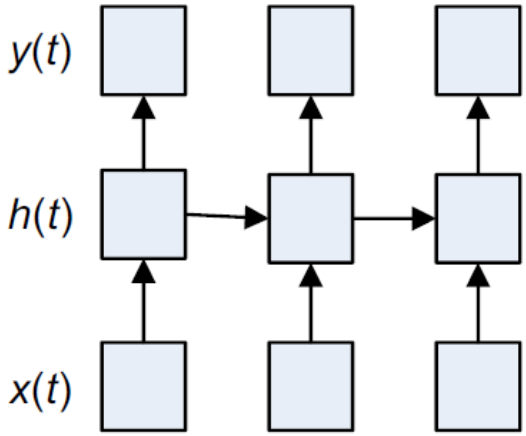

Рис. 1.5. Нейронна мережа Елмана

Переваги методу:

- має дуже високу якість алгоритму при вдалому доборі параметрів;
- є універсальним апроксиматором безперервних функцій;
- підтримує інкрементне навчання.

Недоліки методу:

- ймовірність можливої розбіжності або повільної збіжності, оскільки для налаштування мережі використовуються градієнтні методи;
- необхідність дуже великого обсягу даних них для навчання, щоб досягти високої точності;
- низька швидкість навчання;
- складна інтерпретованість параметрів алгоритму.

Класифікація текстів є одним з основних завдань комп'ютерної лінгвістики, оскільки до неї зводиться ряд інших завдань: визначення тематичної приналежності текстів, автора тексту, емоційного забарвлення висловлювань та ін.

Формально завдання класифікації текстів можна описати в такий спосіб. Є багато документів і множин можливих категорій (класів). Потрібно побудувати класифікатор, що відноситься обраний документ до однієї з декількох заздалегідь визначених категорій на підставі змісту документа. Найбільш поширений сучасний підхід до класифікації ґрунтується на методах машинного навчання. Згідно з цими методами, набір правил або критерій прийняття рішення текстового класифікатора обчислюється автоматично на основі навчальних даних. Навчальними даними є зразки документів кожного класу.

Розв'язання задачі класифікації складається з чотирьох послідовних етапів: попередня обробка та індексація документів, зменшення розмірності простору ознак, побудова та навчання класифікатора за допомогою методів машинного навчання, оцінка якості класифікації.

Для попередньої обробки та індексації документа (тобто при побудові деякої чисельної моделі тексту) зазвичай застосовується одна з трьох моделей: модель «мішка слів», Word2vec та модель, заснована на обліку n-грам. Для реалізації першої та другої моделей необхідні додаткові знання про морфологічну і синтаксичну структуру мови. Застосування символьних n-грам дозволяє не накладати обмеження використання конкретної мови, у деяких випадках є кращим.

Обчислювальна складність різних методів класифікації залежить від розмірності простору ознак. За рахунок зменшення розмірності простору термінів можна знизити ефект перенавчання - явище, при якому класифікатор орієнтується на випадкові або помилкові характеристики навчальних даних, а не на важливі та значущі. Перенавчений класифікатор добре працює на тих примірниках, на яких він навчався, і значно гірше на

тестових даних. Щоб уникнути перенавчання, кількість навчальних прикладів має бути пропорційно числу використовуваних термінів, тому для ефективної роботи класифікатора часто вдаються до скорочення числа використовуваних ознак (термінів). Для зменшення розмірності простору термінів застосовують такі методи, як LSA, TF-IDF, PMI, CRF, IG. Найбільше поширення з них отримав метод TF-IDF.

1.3. Використання квантово-подібних моделей для представлення концептів у базах знань

Математичний апарат, що використовується в квантовій теорії хоч і зародився в рамках фізичних експериментальних досліджень як спроба описати порушення колмогорівської теорії ймовірностей та класичної логіки, проте у даний математичний апарат може бути розглянутий незалежно від фізичного трактування. Квантова теорія, як розділ математики, є не що інше, як узагальнення колмогоровської теорії ймовірностей до поля комплексних чисел з можливістю отримання, як наслідок, "негативних" ймовірностей у тому сенсі, що замість класичної імовірності з'являється поняття амплітуди ймовірності.

Саме амплітуда ймовірності дозволяє розрахувати класичну ймовірність. Тим не менш, такі ймовірності не підкоряються класичній теорії і можливі порушення правил підсумовування та добутку ймовірностей. Такі порушення таки обумовлюються наявністю ефектів контекстуальності аналізованої системи, суперпозиції, а також заплутаності. Основним об'єктом, що описує розподіл станів спостережуваної системи є матриця щільності, яка може бути побудована як для чистих станів, так і для змішаних станів.

Розглянуті вище припущення активно використовується у матеріалах, присвячених використанню квантово-подібних обчислень для завдань текстових документів та інформаційного пошуку в інформаційних системах, а також при моделюванні концептів у базах знань. Використання такого формалізму дозволяє налагодити мости між інформатикою та квантовою фізикою для отримання спільної мови при розв'язанні задач інформатики, в яких завдання або частина задачі зводиться до моделювання квантово-подібних об'єктів.

У цьому розділі представлені роботи, присвячені питанням подання текстових документів, створення векторного семантичного простору для термінів та документів, а також визначення відношень між концептами на базі апарату, який пропонує квантова математика. У своїй частині дані роботи носять теоретичний характер, оскільки, здебільшого, апарат квантової теорії є відносно новий інструмент у сфері автоматичної обробки природно-мовних текстів. Перехід до практичного застосування наведених нижче напрацювань утруднений в силу того, що більшість представлених алгоритмів і методів використовують векторні та матричні представлення дуже великої розмірності.

Тим не менш, дослідження, що проводяться на стику квантової математики та автоматичної обробки текстів, представляють особливий інтерес, оскільки представлена квантово-логічним апаратом сукупність концепцій, таких як заплутаність або суперпозиція, дозволяє по-новому поглянути на питання подання семантики мови. Так, можна провести такі аналогії між поняттями квантової теорії та комп'ютерної лінгвістики. Якщо йдеться про застосування підходу, що відповідає дистрибутивній семантиці, то для одного слова базисним (в поняттях квантової теорії) буде вектор, що представляє контекст, що оточує слово.

Чистий стан, суперпозиція, у такому разі, це невизначеність щодо кількох контекстів слова у навчальній вибірці. Якщо навчальна вибірка складається з набору тематичних текстів, можна говорити про суміші

визначення матриці щільності ймовірностей стану системи – кожної тематики можна визначити свій чистий стан слова. Зрештою заплутаність може означати наявність семантичних відносин між кількома поняттями в тексті.

Методи автоматичної обробки природномовних текстів лежать в основі інформаційного пошуку та базах знань, побудованих автоматизованими засобами, так що дослідження та розробка даних методів за допомогою змішування класичних ідей у цій галузі та ідей квантової математики становлять особливий інтерес. Спочатку інтерес до застосування квантово-подібних моделей зародився в інформаційному пошуку. Одним із завдань інформаційного пошуку є чисельне подання документів та запитів користувачів. Такі представлення зручні, оскільки дозволяють порівнювати між собою документи. Зручним способом представлення є векторні простори – кожен документ, запит чи слово є вектором у певному векторному просторі. Дослідження в частині векторних представлень ведуться ще з робіт Харріса та появи дистрибутивної гіпотези, проте саме розгляд квантовоподібних моделей прийшов із застосування обчислювальної геометрії до інформаційного пошуку [14], а також із необхідності якимось чином враховувати невизначеності, що виникають в текстовому аналізі інформації через те, що математичний апарат квантової теорії виявляється зручним при моделюванні систем, представлених у векторному вигляді з елементами невизначеності щодо точних значень параметрів таких систем.

Можна сміливо сказати, що з перших робіт, у якій висвітлюється питання застосування математики квантової теорії, є робота Яна Рисбергена [15]. Фактично ця робота стала цитуватися більшістю дослідників, які прагнули застосовувати квантово-подібні моделі до інформаційного пошуку та аналізу текстових документів в інформаційних системах. Трохи пізніше вийшла робота Массімо Мелуччі [18], який послідовно, від уявлення окремих слів природної мови, до алгоритмів ранжирування вводить математику, що застосовується у квантовій теорії.

У дослідженнях останніх років виявлено, що характер зв'язків між семантичними одиницями в текстових документах дозволяє застосовувати до опису семантики текстових даних парадигму квантової теорії. Зокрема формалізми, що використовуються для моделювання зв'язків у заплутаних системах, можуть бути перенесені на мову аналізу текстових даних у парадигмі структурної семантики природної мови.

У роботі [16] викладається ідея об'єднання двох підходів до подання семантики текстів: дистрибутивної семантики, при якій семантика окремого слова та слів визначаються статистикою по контексту оточуючих його слів з підходом на основі аналізу синтаксичної структури та функцій окремих слів у реченні.

У [16] використовується формалізм для квантових процесів, розроблений Россом Дунканом, що спирається на теорію категорій і дозволяє уявити квантовий інформаційний процес не як процес перетворення векторів в гільбертовому просторі, а як сукупність об'єктів категорій спеціального виду і функторів.

Кожне слово речення сприймається як окрема квантова підсистема, яка передає деяку інформацію пов'язаних із нею системам. Для представлення таких зв'язків використовується графічна мова, що відображає відношення між об'єктами системи, що розглядається. Сама система - це пропозиція природною мовою для якого, фактично, виконано синтаксичний аналіз і в явному вигляді відображені відношення між словами в реченні.

Структура зв'язків у такій квантово-подібній системі визначається через алгебру Ламбека, яка описує граматику конкретної мови. У цій алгебрі об'єктами є окремі типи слів (іменники, дієслова, прикметники і т.д.), а також зв'язки між ними, які можуть бути в правильно побудованому реченні. Наприклад пропозиція для речення для “Українці ненавидять росіян” може бути представлена наступною схемою, рисунок 1.6:

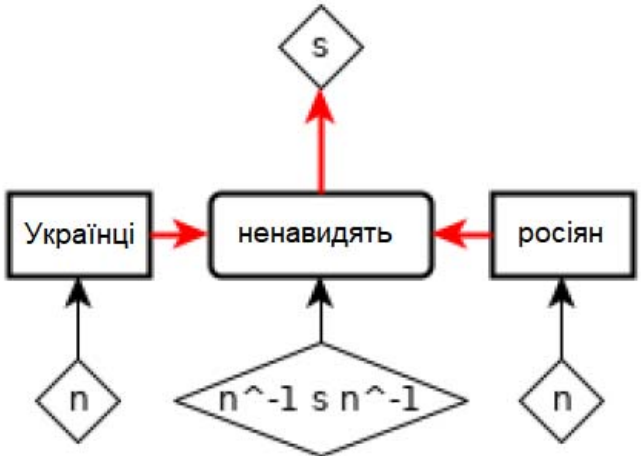

Рис. 1.6: Приклад побудови схеми пропозиції “ Українці ненавидять росіян”

Червоні стрілки позначають напрямок передачі у квантовій системі. Далі, згідно з формалізмом з роботи [16] вся система представляється у вигляді добутку векторів-смислів окремих слів, яке потім редукуються оператором спеціального виду у вектор меншої розмірності.

Відмінною особливістю застосування саме оператора тензорного добутку полягає в тому, що він не має властивості комутативності, а отже за його допомогою можна моделювати некомутативність природної мови. Дійсно, “Українці ненавидять росіян” і “росіяни ненавидять українців” - дві абсолютно різні пропозиції.

Подання векторів авторами у явному вигляді не уточнюється, проте мається на увазі, що контекст деякого слова представляється у вигляді статистики за контекстами, які оточують це слово. Це можуть бути, наприклад, вектори, складені за алгоритмом bag-of-words або з використанням методів тематичного моделювання.

1.4. Постановка задачі дослідження

Метою роботи є розробка методів та засобів класифікації документів в інформаційних системах.

Відповідно до поставленої мети у роботі потрібно вирішити такі основні завдання дослідження:

1. Дослідити відомі підходи та методи до класифікації текстових документів.

2. Дослідити особливості застосування моделей і методів квантової теорії для обробки текстових документів.

3. Розробити метод векторизації слів з використанням апарату квантової теорії ймовірностей для представлення концептів бази знань.

4. Розробити систему класифікації документів в інформаційних системах, яка базується на використанні алгоритму Белла.

5. Реалізувати запропоновані методи та провести відповідні експериментальні дослідження.

Висновки до першого розділу

1. Класифікація текстів є одним з основних завдань комп'ютерної лінгвістики, оскільки до неї зводиться ряд інших завдань: визначення тематичної приналежності текстів, автора тексту, емоційного забарвлення висловлювань та ін. Найбільш поширений сучасний підхід до класифікації ґрунтується на методах машинного навчання.

2. Методи класифікації текстів лежать на стику двох областей - інформаційного пошуку та машинного навчання. Їх схожість полягає у способах представлення самих документів та способах оцінки якості алгоритмів. На сьогоднішній день розроблено велику кількість методів та їх різних варіацій для класифікації текстів. Кожна група методів має свої переваги та недоліки, сфери застосування, особливості та обмеження.

3. В результаті виконаного огляду стану проблеми було встановлено, що дослідження в галузі аналізу текстових документів засобами математичного апарату квантової теорії ведуться за такими основними напрямками: моделювання семантики окремих слів та концептів з використанням апарату квантової теорії ймовірностей, моделювання семантики цілих виразів та пропозицій шляхом явного подання структурних зв'язків між словами.

РОЗДІЛ 2
МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ КЛАСИФІКАЦІЇ ДОКУМЕНТІВ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

2.1. Квантово-механічна аналогія і тест Белла

Для моделювання семантичного простору використовується алгоритм Hyperspace Analogue Language (HAL). У якості найбільш простого підходу з вимірюваної точки зору та з точки зору простоти реалізації для довільної мови до моделювання семантичного простору використовується підхід пакета слів (bag-of-word). В рамках такого підходу зміст кодується лічильниками слів, які входять в контекст об'єкта. Однак цей підхід не враховує порядок слів, що може негативно позначатися на якості моделі. Наприклад, на українській мові якісні моделі в цій частині заборонено забороняти. У той же час модель HAL позбавлена цього недоліку і дозволяє компактно закодувати, хоча і з вмістом інформації, структуру контексту документа або слова.

Нагадуємо, що модель HAL реалізується у наступний спосіб: для текстового корпусу формується векторний простір розмірності лексикону, і кожне слово, для якого будується вектор у цьому просторі, отримує координати, відмінні від нуля лише в тому випадку, якщо деяке слово в тексті правіше цього слова в обмеженому вікні довжини w зустрілось слово, відповідне координаті. У якості значення для цієї координати використовується зворотний розмір вікна відстані до слова.

Тобто, якщо з аналізованого слова слово, відповідна координата вектора якого знаходиться на відстані d слів, а розмір вікна рівний w, то значення координати вектора HAL буде

$$\mathrm{d}_{\mathrm{hal}} = w - d. \quad (2.1)$$

Якщо таке слово зустрілося кілька разів, то в відповідній координаті записується сума відстаней. У загальному випадку значення координат вектора HAL може бути визначено таким чином:

$$\mathrm{W}ord_i = \sum_{j \in \mathrm{p}_i} (w - d_i j + 1). \quad (2.2)$$

де $\mathrm{W}ord_i$ - вектор аналізованого слова, p_i - множина словопозицій, в яких знаходиться i-е слово, $d_i j$ - відстань від слова, відповідного вектору $Word$ до i-го слова в j-й словопозиції, w - розмір вікна HAL. При цьому $d_i j = 0$, якщо $d_i j >$ w або i відповідає слову $Word$.

Нижче наведено приклад побудови такої матриці для простого прикладу з вікном сканування $w = 3$: "Мама прибирала сходи, а потім мама прибирала коридор".

Таблиця 2.1

Приклади векторів згідно моделі HAL

	мама	прибирати	сходи	потім	коридор
мама	0	3 + 3	2	1	2
прибирати	1	0	0	0	3
сходи	2	1	0	3	0
потім	3	2	0	0	1
коридор	0	0	0	0	0

Необхідно відзначити, що фактично матриця HAL моделює відношення порядку між словами в рамках обмеженого вікна сканування w - ненульова координата буде тільки при задоволенні відношення "слово і знаходиться правіше за аналізоване слово в тексті доповнюючи це відношення також зважуванням відстані між словами.

2.2. Формулювання тесту Белла для класифікації текстових документів в інформаційній системі

Для побудови методу порівняння фрагментів текстів прийнято як вектори-контексти слів використовувати нормалізовані рядки HAL-матриці. Такий спосіб формування векторів відповідає векторної моделі, прийнятої в квантовій теорії, де всі вектори, що описують стан системи, що досліджується, повинні бути одиничної довжини.

Документ подається як нормована сума таких векторів. Таким чином, слова та документи в такій моделі видаються у вигляді суперпозиції окремих термінів з лексикону, відображаючи невизначеність щодо значення цих слів чи документів.

У цій роботі досліджується застосування аналізу текстових документів заснованого на тесті Белла, для аналізу текстових документів українською мовою. Нерівності Белла застосовують у фізиці визначення наявності стану заплутаності кількох квантових частинок. Цікаво, що порушення нерівностей Белла було продемонстровано в серії експериментів з тестами для людей, що включають прості питання з відповідями "так-ні". Ця нерівність може порушуватися і для випадку моделювання контекстів природно-мовних текстів.

У цій роботі використовується CHSH форма тесту, подана нижче:

$$S_{bell} = |E(A,B) - E(A,C)| + |E(B,D) + E(C,D)| \quad . \qquad (2.3)$$

де A,B,C,D – деякі тести з наслідками $-1,1$ та $E(X,Y)$ – очікуване значення спільної появи X та Y. Такими тестами може бути, наприклад, перевірка стану системи на конкретне значення (у разі фізичної інтерпретації), у випадку з опитувальниками – відповідь "так" на конкретне

питання, а у випадку з аналізом текстової інформації – відповідність виділеної семантичної категорії чи ні.

Результати розрахунку за формулою (2.3) прийнято поділяти на 3 діапазони. Для класичного теоретико-імовірнісного випадку S_{bell} набуває значення від 0 до 2. $2 < S_{bell} \leq 2\sqrt{2}$ може бути отриманий тоді, коли досліджувані дві квантові частинки перебувають у заплутаному стані. Значення вище $2\sqrt{2}$ називаються несигнальним регіоном.

Позначимо наступну гіпотезу: якщо за певних умов спостерігатиметься порушення нерівності Белла в рамках побудованого векторного простору, це буде говорити про появу додаткових залежностей між подіями в системі, які явно не враховані в моделі, і є сенс спробувати змоделювати дані системи апаратом квантової математики, яка дані залежності явно враховує. Такою моделлю можуть бути, наприклад, матриці густини.

2.3. Алгоритм розрахунку параметра Белла для класифікації текстових документів в інформаційній системі

Зробимо припущення, що порівнювані фрагменти текстів відповідають лише одному слову кожен. Тоді тести у нерівності Белла відповідатимуть перевірці відповідності контексту документа вектору першого чи другого слова. Якщо необхідно розширити цей алгоритм до ланцюжків слів, то можна використовувати той самий принцип як і для формування вектора документа-контексту – підсумувати вектори матриці HAL з ланцюжка слів і суму нормувати. Тепер нам необхідно визначити, як співвідносяться два слова у контексті документа. Вважатимемо що слова близькі за змістом, якщо спостерігається порушення нерівності Белла. Розглянемо проекції вектора документа $|D\rangle$ на площині, утворені векторами слів $|w_1\rangle$ і $|w_2\rangle$. Застосуємо алгоритм Грама-Шмідта для векторів слів $|w_1\rangle$ і $|w_2\rangle$, отримавши таким

чином два ортогональні базиси $\{|u_1\rangle, |u_2\rangle\}$ і $\{|w_1\rangle, |w_2\rangle\}$ відповідно. Далі побудуємо проекцію вектора документа $|D\rangle$ на дані базиси, отримавши, таким чином, подання документа у контексті цих двох слів:

$$|D_{w1}\rangle = a|u_1\rangle + b|u_2\rangle \tag{2.4}$$

$$|D_{w2}\rangle = c|v_1\rangle + d|v_2\rangle \tag{2.5}$$

де коефіцієнти при базисних векторах можуть бути розраховані шляхом проектування вектора документа на базисний вектор з нормуванням довжини проекції. Наприклад, для коефіцієнта a це можна розрахувати так:

$$a = \frac{\langle u_1|D\rangle}{\sqrt{\langle u_1|D\rangle^2 + \langle u_2|D\rangle^2}} \tag{2.6}$$

По-суті, ці коефіцієнти відповідають косинусним відстаням від аналізованого слова до документа. Так як надалі буде зручніше вести розрахунки в рамках одного базису, необхідно використовувати матрицю перетворення до іншого базису. Нижче представлена матриця переходу від базису $\{|v_1\rangle, |v\rangle\}$ до $\{|u_1\rangle, |u_2\rangle\}$:

$$M = \begin{bmatrix} \langle v_1|u_1\rangle & \langle v_2|u_1\rangle \\ \langle v_2|u_1\rangle & v_2|u_2\rangle \end{bmatrix} = \begin{bmatrix} -\frac{p}{\sqrt{1-p^2}} & \frac{\sqrt{1-p^2}}{p} \end{bmatrix} \tag{2.7}$$

Оскільки всі вектори мають одиничну довжину, то $\langle v_1|u_1\rangle = p$ є нічим іншим, як косинус кута між векторами аналізованих фрагментів текстів.

Тепер необхідно визначити тести, які будуть використовуватись у нерівності Белла. Нехай оператор A повертає $+1$, якщо вектор, що розглядається, збігається з вектором u_1 і -1, якщо вектор збігається з ортогональним вектором u_2. Як такий оператор підходить матриця Паулі σ_z:

$$\sigma_z = A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.8}$$

Так як всі обчислення будуть проводитися в одному базисі, то для оператора B, який визначає таку ж логіку як оператор A, але для базису $\{|v_1\rangle, |v\rangle\}$, потрібно виконати перетворення до іншого базису:

$$B = M^{-1} \cdot A \cdot M = \begin{bmatrix} 2p^2 - 1 & 2p\sqrt{1-p^2} \\ 2p\sqrt{1-p^2} & 1 - 2p^2 \end{bmatrix} \tag{2.9}$$

Таким чином, застосовуючи правило Борна для визначення очікуваного значення та з урахуванням нормування векторів, можна, наприклад, визначити ймовірність того, що документ відповідає вектору w_1:

$$\langle A \rangle_D = \langle D|A|D\rangle = a^2 - b^2 = 2a^2 - 1 \tag{2.10}$$

Щоб визначити ступінь відповідності документа одночасно слову w_1 та w_2 можна використовувати комбінацію операторів A та B:

$$\langle A \rangle_D = \langle D|A \cdot B|D\rangle = a^2 - b^2 = 2a^2 - 1 \tag{2.11}$$

Зрештою тест Белла для нашого випадку набуває наступного вигляду:

$$S_{query} = |\langle AB_+\rangle + \langle A_x B_+\rangle| + |\langle AB_-\rangle + \langle A_x B_-\rangle| \tag{2.12}$$

де решта операторів визначено як:

$$A_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, B_+ = -\frac{B+B_x}{\sqrt{2}}, B_- = -\frac{B-B_x}{\sqrt{2}}, B_x = M^{-1} A_x M \tag{2.13}$$

Нагадаємо, що оператор A відповідає операції визначення степеня відповідності контексту документа першому фрагменту тексту і повертає 1, якщо документ повністю йому відповідає, -1 – відповідає ортогональному вектору, оператор A_X відповідає невизначеності щодо схожості семантики слова A та документа – цей оператор набуває максимального значення Якщо косинус кута між документом словом буде відповідати куту в 45 градусів. Відповідно, оператори B і B_X виконують ту ж роль. Введення операторів B_- і B_+ відповідає повороту базису другого фрагмента тексту на 45 градусів, що у випадку з фізичними експериментами робиться найчастіше, оскільки в такому разі максимізується значення тесту Белла.

2.4. Специфіка поведінки тесту Белла під час класифікації текстових документів

Представлене формулювання тесту Белла відрізняється специфічною поведінкою. На рисунку 2.1 представлені різні варіанти відносного розташування базисних векторів w_1 та w_2.

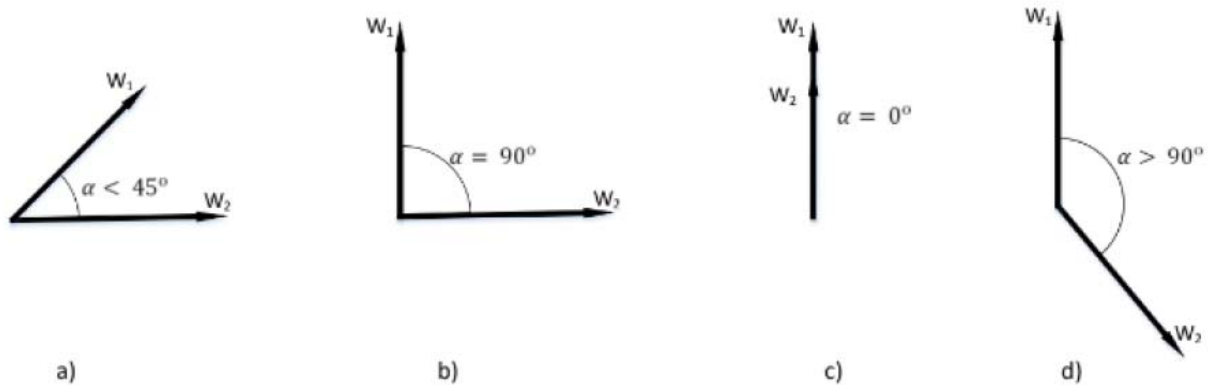

Рис. 2.1. Варіанти відносного розташування базисних векторів w_1 та w_2.

При цьому може бути крайові варіанти розташування базисних векторів слів щодо один одного:

1. кут між векторами менше π/2 (випадок (14.a));

2. кут між векторами дорівнює π/2 (випадок (14.b));

3. кут між векторами дорівнює 0 (випадок (14.c));

4. кут між векторами більше π/2 (випадок (14.d)).

Для випадків (b,c) характерно те, що S_{bell} у виразі (2.12) набуває значення 2. Для випадків кута між базисними векторами більшого ніж π/2 (випадок (14.d)) вираз (55) набуває значення менше 2. Нарешті, випадок (випадок (14.a)) дає значення параметра $S_{bell} > 2$. Останній випадок, згідно з квантовою теорією, відповідає стану заплутаності системи частинок для класифікації документів.

Оцінки, які дає вираз (2.12) можуть бути інтерпретовані так: у разі частої зустрічі термінів один з одним, тобто при колінеарності векторів контекстів цих слів, а також у разі наближення до стану ортогональності двох контекстів, що відповідає контекстам, що не перетинаються, вираз (2.12) дає значення близькі до 2. Тим не менш, максимум цього може бути досягнутий у випадку якщо обидва слова утворюють базиси, які повернені щодо один одного на кут π/4.

У нашому випадку ми можемо говорити про те, що два вектори поділяють загальний контекст між собою – між двома словами є словникові контексти, що частково перетинаються (за побудовою матриці HAL), що може говорити про те, що слова близькі за змістом.

При цьому все це слід розглядати в одному документі, що може порушувати нерівність Белла (тобто спостерігатись випадок (d)), а в іншому – ні, через специфіку побудови векторів класифікованих документів, відповідно w_1 та w_2.

2.5. Векторне представлення концептів бази знань та квантова аналогія

Одним із найважливіших напрямів в інформаційному пошуку є вирішення задачі векторного представлення об'єктів, що беруть участь в інформаційному пошуку, наприклад, слів або документів. Даний напрямок важливий оскільки векторні представлення дозволяють вводити операції порівняння, наприклад, запитів користувача та документів з безпосереднім застосуванням математичного апарату (наприклад використовуючи косинусну відстань), що безумовно збільшує інтерпретованість моделі та можливості її застосування в інших алгоритмах на противагу алгоритмам на основі класичного машинного навчання, працюючих з ознаковими описом об'єктів, які можуть бути коректно інтерпретовані в термінах векторних операцій з них.

Іншою сферою застосування векторних представлень можуть бути бази знань, в яких векторний простір використовується для пошуку асоціативних зв'язків між концептами, представленими векторами цього простору.

У цій роботі використовується гіпотеза про застосування апарату матриць щільності до моделювання векторних (матричних) представлень слів природної мови. Ця гіпотеза спирається на два факти. По-перше, існують експерименти, що показують що вектора, одержувані у вигляді моделювання у межах дистрибутивної гіпотези, тобто за допомогою контекстів слів, мають квантово-подібну статистичну структуру. Це дає деякі підстави припускати, що для моделювання векторних представлень слів може бути використаний апарат квантової теорії. По-друге, можна провести чітку аналогію між процесом відновлення векторних представлень слів та квантовою томографією. Зокрема, базисом такої семантичної томографії буде набір контекстів у яких є аналізоване слово.

Простими векторними представленями таких контекстів можуть бути ті самі Bag-of-Words вектора. Спостережуваною в такій моделі буде ймовірність того, що вибране слово оточуватиме контекст для якого визначається спостереження. Звичайно, у разі прямого використання мішка слів всі контексти будуть унікальні, але при чищенні сміття та додаткової кластеризації таких векторів можна отримати дискретний імовірнісний розподіл векторів контекстів. Дотримуючись обмежень та ймовірнісною мірою можна спробувати отримати матричне представлення слова або матрицю щільності ймовірностей.

Таким чином, процес побудови векторного (матричного) представлення слова може бути представлений як аналог процесу квантової томографії у просторі векторів-контекстів слів.

Як випливає з проведеного огляду методів аналізу текстових документів засобами математичних моделей, які застосовуються в квантовій механіці, використання подібних структур дозволяє отримати додаткові переваги порівняно з класичними методами аналізу текстів. Зокрема, апарат матриць щільності дозволяє представляти розподіл семантичних контекстів слів в матричному вигляді і надає ймовірнісний захід визначення ступеня схожості слова з його контекстом, що спостерігається в документі або з іншим словом. Більш того, математичні об'єкти, що використовуються в квантовій теорії, дозволяють оперувати такими поняттями як суперпозиція та заплутаність, що, як було зазначено раніше, може знайти пряму аналогію в аналізі текстових даних. Таким чином, можна прийти до завдання векторизації текстових даних за допомогою матриць густини ймовірностей як задачі машинного навчання – для даного набору контекстів та розподілу їх ймовірностей необхідно отримати матрицю густини.

Представимо задачу формально. Необхідно розробити алгоритм, що на вхід приймає вибірку з нерозмічених текстових документів й у кожного слова цих текстів повертає вектор (матрицю щільності), тобто побудувати

відображення $a(w) : W \rightarrow R^L$, де W – це множина всіх слів, w – аналізоване слово, L – розмірність векторного простору.

Значення аналізованого слова визначається через його контекст, що відповідає дистрибутивній гіпотезі Харріса. Якщо контекст слова представити вектором, то такому поданні $i-$й координаті вектора буде присвоєно число повторень i-го слова у словнику W у вікні навколо слова w. Так, наприклад для пропозиції "мама прибирала коридор" і словника, що складається з цих трьох слів, контекст слова "прибирати" визначатиметься вектором $v_{\text{прибирати}} = [110101010]$.

Оскільки квантова теорія ймовірностей використовує вектора та матриці для опису досліджуваних систем, вона є зручним засобом для отримання векторного представлення слова. Квантова теорія ймовірностей, у відриві від квантової фізики, є геометричним узагальненням класичної теорії ймовірностей.

Скористаємося аналогією, наведеною в попередньому параграфі. Нехай A_k – це матриця-проектор на підпростір, відповідний деякому контексту k з набору відомих нам контекстів розмірності N. Щоб отримати такий проектор, нам необхідно представити контекст слова як вектор за аналогією з описаним вище методом. Так як матриця A_k, що отримується з такого вектора, повинна володіти властивостями проектора, то умова нормування такого вектора обов'язкова:

$$A_k = \frac{v_k \cdot v_k^T}{|v_k|^2} \tag{2.14}$$

де v_k – вихідний вектор контексту слова.

Імовірність P_k нам також відома, тому що відомий текстовий корпус, на якому проводиться навчання. Для того, щоб отримати розподіл ймовірностей на N-контекстах для досліджуваного слова, достатньо виконати групування векторів контекстів за їх точним збігом і виразити

ймовірність їх появи через їх частоту. Очевидно, що у разі наївної реалізації алгоритму моделювання контексту ми отримаємо рівномірний розподіл ймовірностей, де кожен контекст отримає рівну ймовірність. При цьому не буде враховуватися той факт, що більша частина контексту може складатися зі стоп-слів або слів, що часто зустрічаються, які не дають жодного вкладу в зміст слова. У цьому випадку є сенс виконати попередню обробку, яка полягає в тому, щоб прибрати стопслова, очистити сміття за допомогою tf/idf і виконати кластеризацію отриманих контекстів для утворення груп за якими і будуватиметься ймовірнісний розподіл.

Продовжуючи аналогію, матриця щільності ρ є опис стану «системи» відповідної досліджуваному слову. Саме цю матрицю необхідно відновити, отримавши таким чином матричне представлення узагальненого контексту слова. На відміну від квантової теорії, ми далі розглядатимемо всі матриці як об'єкти над полем дійсних чисел. Таким чином, завдання отримання представлення слова зводиться до завдання пошуку матриці ρ, яка задовольняє рівняння (2.12). Запишемо докладніше представлення матриць ρ і A_k:

$$\rho = \begin{bmatrix} \rho_{11} & \dots & \rho_{1L} \\ \dots & \dots & \dots \\ \rho_{L1} & \dots & \rho_{LL} \end{bmatrix}, A_k = \begin{bmatrix} a_{11} & \dots & a_{1L} \\ \dots & \dots & \dots \\ a_{L1} & \dots & a_{LL} \end{bmatrix} \quad (2.15)$$

Взяття їхнього добутку може бути представлене як такий вираз:

$$Tr(A_k\rho) = \sum_{i=1}^{L}\sum_{j=1}^{L} a_{ij} \cdot p_{ij} = P_k \quad (2.16)$$

При цьому на матрицю щільності, з квантової теорії ймовірностей, накладається ряд обмежень. Таким чином, ми звели завдання семантичної томографії та представлення слова у вигляді матриці щільності через

контексти в яких дане слово спостерігається до завдання розв’язку системи рівнянь (2.16), що породжується контекстами слів.

2.6. Алгоритм отримання матричних представлень концептів бази знань

Алгоритм градієнтного спуску добре вивчений і є відомим способом мінімізації будь-якого функціоналу. Однак, крім необхідності визначення регуляризаторів, також корисно описати сам алгоритм як псевдокод.

В першу чергу, при визначенні регуляризаторів, необхідно відзначити той факт, що на даному етапі ми використовуємо поле дійсних чисел, і, відповідно, для системи обмежень (2.16) останній вираз недійсний, він лише вказує нам на те, що матриця щільності, яку ми хочемо збудувати буде симетричною. З методу градієнтного спуску, який для зміни цільової матриці використовує взяті з різними вагами вихідні проектори на контекст слова, природним чином випливає те, що матриця вже буде симетричною. Третє обмеження $Tr(\rho) = 1$ може бути досягнуто нормуванням матриці за її слідом на кожному k-му кроці градієнтного спуску (це може бути параметром алгоритму), що також дозволить "скидати" градієнтний спуск з локальних мінімумів. Що стосується обмежень, що залишилися, то по них можна безпосередньо записати вирази регуляризаторів:

$$\begin{cases} Tr(p) = 1 \rightarrow (Tr(p) - 1)^2 \\ Tr(p^2) \leq 1 \rightarrow \left(Tr(p^2) - 1\right)^2 \end{cases} \tag{2.17}$$

Ця форма регуляризаторів відповідає припущенню у тому, що значення сліду матриці і сліду її квадрата підпорядковується нормальному розподілу з очікуваним значенням, рівним одиниці, що, суворо кажучи, відповідає реальному обмеженю на форму матриці щільності. Однак, отримання

гладкого функціоналу, який би відповідав вихідним обмеження в термінах принципу максимальної правдоподібності, утруднено і в даній роботі робиться таке припущення про розподіл сліду матриці. Нижче на рисунку 2.2 наведено псевдокод алгоритму побудови матриці густини для слова щодо набору його контекстів.

Data: Ctxs
Probs
eps
Result: P
CtxsProj = EmptyList()
M ← ZeroMatrix()
for $i \leftarrow 1$ **to** $length(Ctxs)$ **do**
    Ctx ← Ctxs[i]
    Ctx ← $\frac{Ctx}{\|Ctx\|}$
    Proj ← $Ctx^T \times Ctx$
    CtxsProj.add(Proj)
    M ← $M + P[i]Proj$
end
$Q_0 \leftarrow -\infty$
Q_1 = Q(M, CtxsProj, P)
i ← 2
$\nabla Q_1 = \nabla Q(M, CtxsProj, P)$
while $\|Q_i - Q_{i-1}\| < eps$ **do**
    $\nabla Q_i \leftarrow \nabla Q(M, CtxProj, P)$
    $\nabla Q_i \leftarrow AdamGrad(\nabla Q_i, \nabla Q_{i-1})$
    M ← $M - \nabla Q_i$
    $\nabla Q_{i-1} \leftarrow \nabla Q_i$
    $i \leftarrow i + 1$
end

Рис. 2.2. Алгоритм отримання матричного представлення слова

У цьому алгоритмі застосовується модифікований алгоритм градієнтного спуску AdamGrad, який динамічно розраховується для кожної змінної, за якою проводиться мінімізація, оптимальне значення градієнтного спуску, виходячи зі значення градієнта на попередніх кроках.

У нашому випадку – це вибір оптимального кроку зміни компоненти матриці щільності виходячи з проекторів на контексти. Такий підхід дозволяє запобігти переміщенню оптимальної точки у просторі пошуку рішень за якимось обраним осям, якщо їм вже знайдено оптимальне значення, тобто це випадок точки, при якому не слід рухатися вздовж відповідного встановленого рядка.

У нашому випадку цей алгоритм також дозволяє позбутися від жорсткої структури матриці щільності, яка формується після серії операцій складання з вагами для проекторів на контексти. Іншими словами, при використанні AdamGrad не всі значення матриць проекторів будуть враховуватися при побудові щільності матриці.

Висновки до другого розділу

1. Розглянуто математичний апарат, необхідний для визначення ступеня заплутаності слів, які перебувають у аналізованому ланцюжку з позицій апарату квантової теорії. Під заплутаністю мається на увазі ступінь семантичної близькості слів чи аналізованих концептів бази знань, які зустрічаються в одному смисловому контексті, зокрема у документі. Також варто відзначити можливі поліпшення щодо застосування нерівності Белла до зазначеного завдання.

2. Розглянуто аналогію між квантовою томографією та процесом побудови векторних представлень слів для текстових документів. Подібна аналогія дозволяє адаптувати математичний апарат, що використовується для

опису процесу квантової томографії, який використовується для опису статистичних властивостей об'єктів, що мають квантово-подібні властивості.

З погляду моделювання семантики такий математичний апарат дозволяє врахувати властивості суперпозиції та заплутаності контекстів слів, які зустрічаються при векторизації аналізованого слова.

РОЗДІЛ 3
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ КЛАСИФІКАЦІЇ ДОКУМЕНТІВ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

3.1. Програмна реалізація методу класифікації документів на базі тесту Белла

В якості основного інструментального засобу для реалізації описаних у алгоритмів вибрано мову програмування Python 3 та пакет додаткових бібліотек Anaconda, який включає велику кількість бібліотек для проведення наукових розрахунків таких як бібліотека Numpy та SciPy.

Основною перевагою Numpy є те, що надаючи простий програмний інтерфейс для матричних обчислень на мові Python, ця бібліотека добре оптимізована під них і, зберігаючи ясність та наочність коду мови програмування, Python розробник не платить за це продуктивністю (у порівнянні з реалізацією тих же обчислень на чистому Python). SciPy у свою чергу надає набір засобів для лінійної алгебри, роботи з розрідженими матрицями, які будуть корисні при розробці алгоритму семантичної томографії.

Крім того, Anaconda поставляється разом з редактором блокнотів, який, за рахунок тісної інтеграції із засобами візуалізації (зокрема matplotlib), дозволяє виконувати обчислювальні експерименти і тут же їх документувати та візуалізувати результати.

Крім того, для виконання експерименту порівняно алгоритму семантичної томографії для векторизації слів текстових документів були використані бібліотека NLTK для обробки документів англійською мовою, а також бібліотека Tensorflow на навчання моделі Word2Vec.

Усі текстові документи піддавалися наступній обробці: з дампа Wikipedia витягувалися лише основні тексти статей, потім кожен текстовий

файл токенізувався і всі слова отриманого потоку токенів піддавалися стемінгу за допомогою стемера Портера, за отриманими словами будувався індекс, а також для кожного документа будувалася матриця HAL, дані якої, своєю чергою використовувалися для розрахунків тесту Белла. Нижче наведено вихідний код для токенізації та стемінгу (рисунок 3.1).

```
stemmer = PorterStemmer()

def splitter(text):
    words = words.split(text.lower())
    words = map(lambda w: w.strip(), words)
    words = filter(lambda w: len(w) != 0, words)
    return words

def stemmer(word):
    return stemmer.stem(word)
```

Рис. 3.1. Фрагмент лістингу коду для токенізації та стемінгу

При побудові індексу виконується обробка текстів згідно з функціями, описаними вище, кожному новому слову в індексі надається чисельний ідентифікатор по порядку і отримані ідентифікатори відображають слова координати матриці HAL. Нижче наведено лістинг побудови індексів з набору документів (рисунок 3.2).

Отриманий індекс використовується для побудови матриць HAL. Нагадаємо, що як основна ідея при побудові даної моделі векторизації текстових даних лежить ідея побудови матриці відношення "слідує за". То для слів розташованих у рядках матриці HAL значення в комірках відповідають тому що:

а) слово в стовпці зустрічалося в тексті праворуч від слова в рядку;

б) загальна сума відстаней, виражених у числі слів від слова в рядку до слова в стовпці дорівнює значенню ячейки.

Таким чином, відношення, яке будує матриця HAL, дозволяє, на відміну від класичного bag-of-word методу, моделювати порядок слів у текстах та відстані між словами.

```
def build_index(text, word_splitter, word_processor, index = {}):
    words = word_splitter(text)
    words=list(words)
    for i in range (len(words)):
        words[i] = word_processor(words[i])
    counter = len(index)

    freq = {}
    for word in words:
        if word in freq:
            freq[word] = freq[word] + 1
        else:
            freq[word] = 1
    a=[]
    for w in freq:
        a.append((freq[w],w))
    a.sort(reverse=True)
    print(a)

    filtered = []
    for word in words:
         if freq[word] > 0:
            filtered = filtered + [word]

    for word in filtered:
        if(not(word in index)):
            index[word] = counter
            counter = counter + 1
    display('Number␣of␣words='+str(len(words)))
```

Рис. 3.2. Фрагмент лістингу коду для побудови індексів з набору документів

Також, при побудові матриці, як і у випадку з bag-of-words, слова розглядаються у вікні кінцевої довжини. Нижче наведено лістинг формування матриці HAL для текстового документа (рисунок 3.3).

Отримана матриця HAL дозволяє виконати векторизацію слів – кожне слово є рядком матриці HAL. Таким чином реалізується ідея гіпотези

Харріса, але з додатковими властивостями, що відображають взаємні розташування слів у контекстах.

```
def build_hal_matrix(
        text_path,
        index,
        window_size,
        word_splitter,
        word_processor,
        encoding):

    hal = np.zeros((len(index), len(index)))
    with codecs.open(text_path, "r", encoding) as f:
            text = f.read().lower()
    words = word_splitter(text)

words = list(map(lambda w: word_processor(w), words))
for i in range(0, len(words)):
    word = words[i]
    word_index_from = index[word]
    for j in range(1, window_size + 1):
        if (i+j) < len(words):
            right_word = words[i + j]
            word_index_to = index[right_word]
            dist = window_size - j + 1
            hal[word_index_from, word_index_to] =
             hal[word_index_from, word_index_to] + dist
return hal
```

Рис. 3.3. Лістинг формування матриці HAL для текстового документа

Векторні представлення із матриці використовуються для обчислень параметра тесту Белла. Слід звернути увагу на те, що так як при оцінці параметра Белла використовується ортогональний базис, що відображає аналізоване слово, для якого вважається параметр, був використаний алгоритм ортогоналізації Грама-Шмідта, який дозволяє по вектору побудувати ортогональний базис, один з векторів яких буде направлений з переданимим вектором. Також в якості порівнюваного текстового фрагмента використовується цілий документ, був використаний алгоритм векторизації

всього текстового документа, що полягає в тому, що всі вектори, складові документ сумуються в один вектор і виконується нормування отриманого значення. Нижче наведено лістинг для обчислення базисів, у яких розглядається текстовий документ (рисунок 3.4).

```
# Gramm-Shmidt method
def gs_cofficient(v1, v2):
    return np.dot(v2, v1) / np.dot(v1, v1)

def multiply(cofficient, v):
    return map((lambda x : x * cofficient), v)

def proj(v1, v2):
    return multiply(gs_cofficient(v1, v2) , v1)

def gs(X):
    Y = []
    for i in range(len(X)):
        temp_vec = X[i]
        for inY in Y :
            proj_vec = proj(inY, X[i])
            temp_vec = np.array(list(map(
                lambda x, y : x - y,
                temp_vec,
                proj_vec
            )))
        Y.append(temp_vec)
    return Y

def define_document_vector(hal):
    result = np.zeros((1, hal.shape[1]))
    for i in range(0, hal.shape[0]):
        result = result + hal[i]
    return result[0]

# compute basis
def define_orths(word_1, word_2):
    o1 = gs(np.array([word_1, word_2]))
    o2 = gs(np.array([word_2, word_1]))
    return (o1, o2)
```

Рис. 3.4. Лістинг для обчислення базисів, у яких розглядається текстовий документ

Отримані базиси використовуються для розрахунку параметра тесту Белла за формулами, наведеними в розділі 2. Нижче представлений лістинг з відповідними функціями (рисунок 3.5).

```
def proj_coef(document_vector, word_vector, orth_word_vector):
    denom = m.sqrt(
             document_vector.dot(word_vector)**2 +
             document_vector.dot(orth_word_vector) ** 2
    )
    return document_vector.dot(word_vector) / denom

def define_proj_coefs(document_vector, word_vector, orth_vector):
    coef1 = proj_coef(document_vector, word_vector, orth_vector)
    coef2 = proj_coef(document_vector, orth_vector, word_vector)
    return (coef1, coef2)

def born_rule(operator, bra, ket):
    return np.matmul(bra, np.matmul(operator, ket))[0][0]

def compute_bell(document_vector, word_1, word_2):
    document_vector = document_vector /
```

Рис. 3.5. Лістинг для обчислення базисів, у яких розглядається текстовий документ

Код, описаний у цьому розділі використовувався щодо експерименту з оцінки працездатності алгоритму порівняння текстових даних між собою з допомогою параметра Белла.

3.2. Програмна реалізація інформаційної системи класифікації документів на основі реалізованих методів

Для тестування запропонованих методів та алгоритмів реалізовану інформаційну систему класифікації документів з метою розсилки цих документів поштовим клієнтом до визначеної групи користувачів. Для

реалізації системи обрано метод – Agile, за допомогою якого можна поступово нарощувати функціональність системи, а для програмної реалізації .Net із використанням мови програмування C# та СУБД MSSQL.

Система використовується декількома групами користувачів, відповідні діаграми варіантів використання для кожної з них представлено на рисунку 3.6

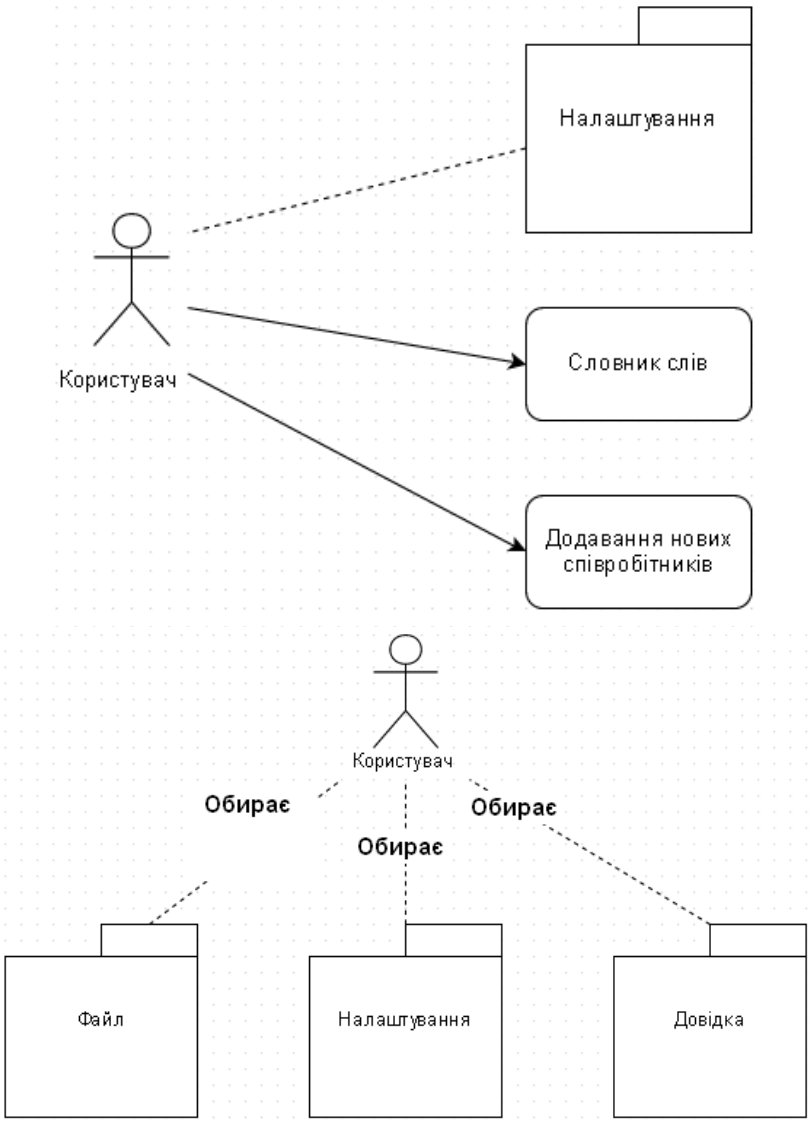

Рис. 3.6. Діаграми варіантів використання

На рисунку 3.7. представлено діаграму класів розробленої системи для класифікації документів.

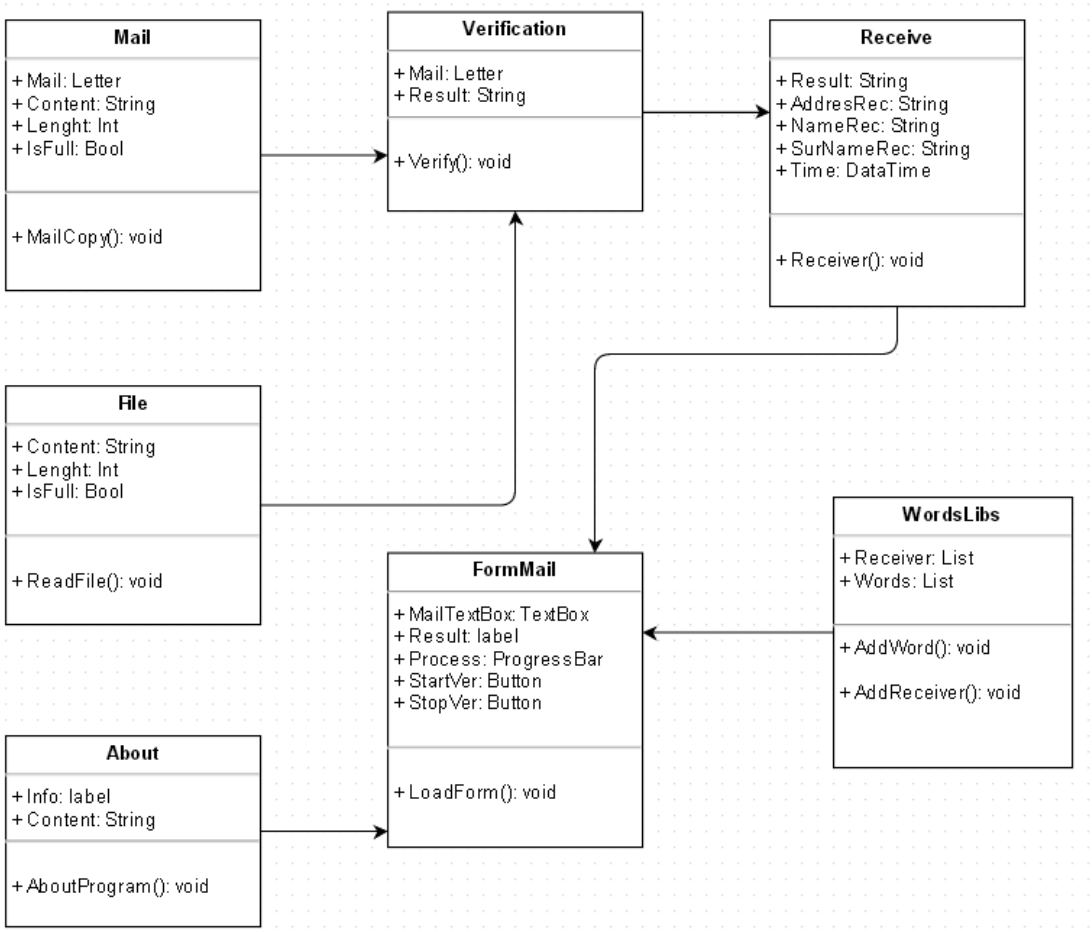

Рис. 3.7. Діаграма класів

На рисунку 3.8 представлено екранну копію форми, яка відображає функціонал для роботи з документами.

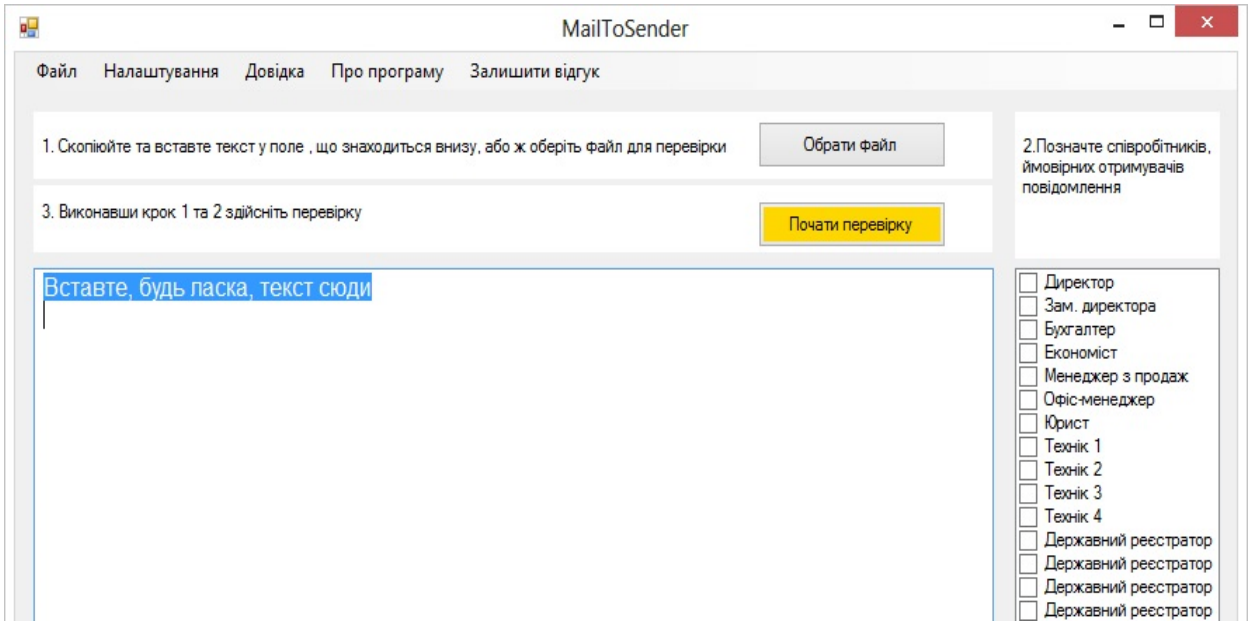

Рис. 3.8. Копія екранної форми для завантаження документів

На рисунку 3.9. представлено копію вікна із візуалізацією перевірки та класифікації документів.

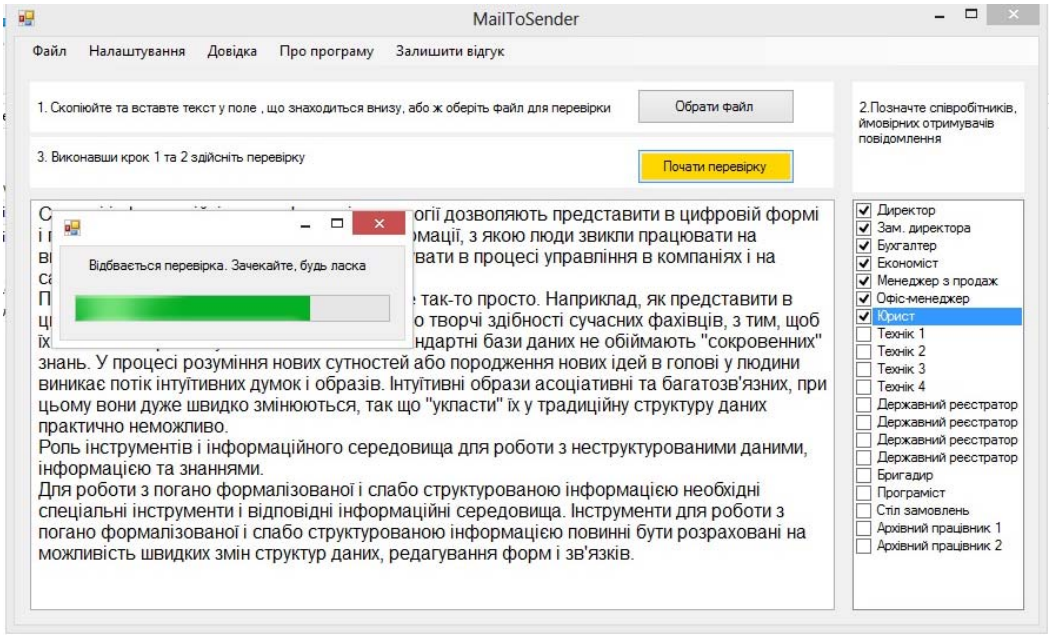

Рис. 3.9. Копія екранної форми для перевірки документів

Відповідно до проведеної класифікації в системі реалізовано функціонал для відправки класифікованого документа до окремої групи користувачів, що відображено на рисунку 3.10.

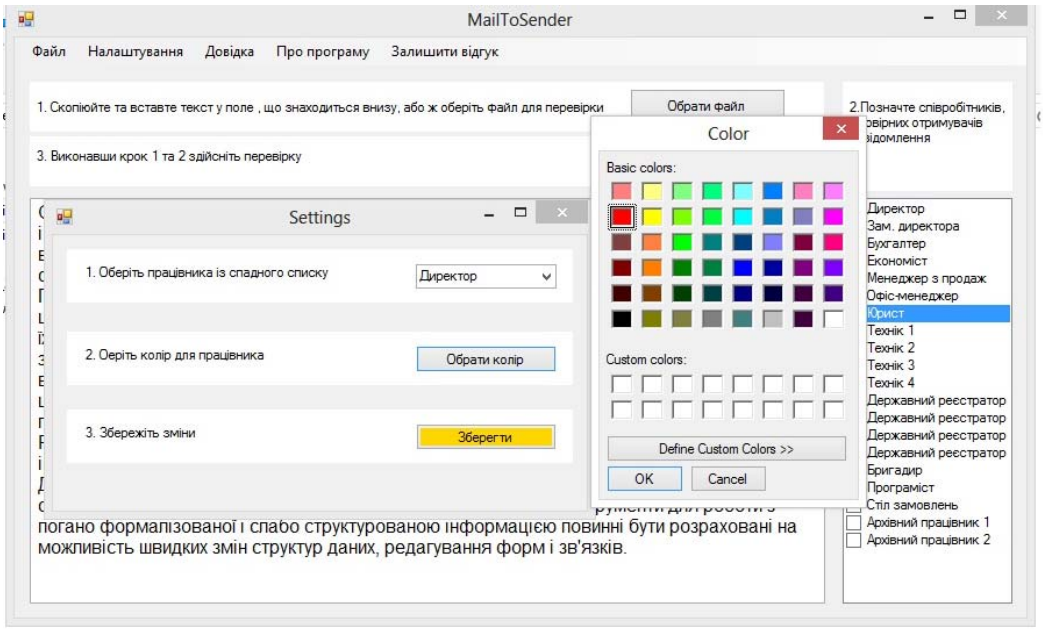

Рис. 3.10. Копія екранної форми для налаштувань системи

На рисунку 3.11 представлено копію екранної форми із встановленням відповідних кольорів документів для кожної групи користувачів інформаційної системи.

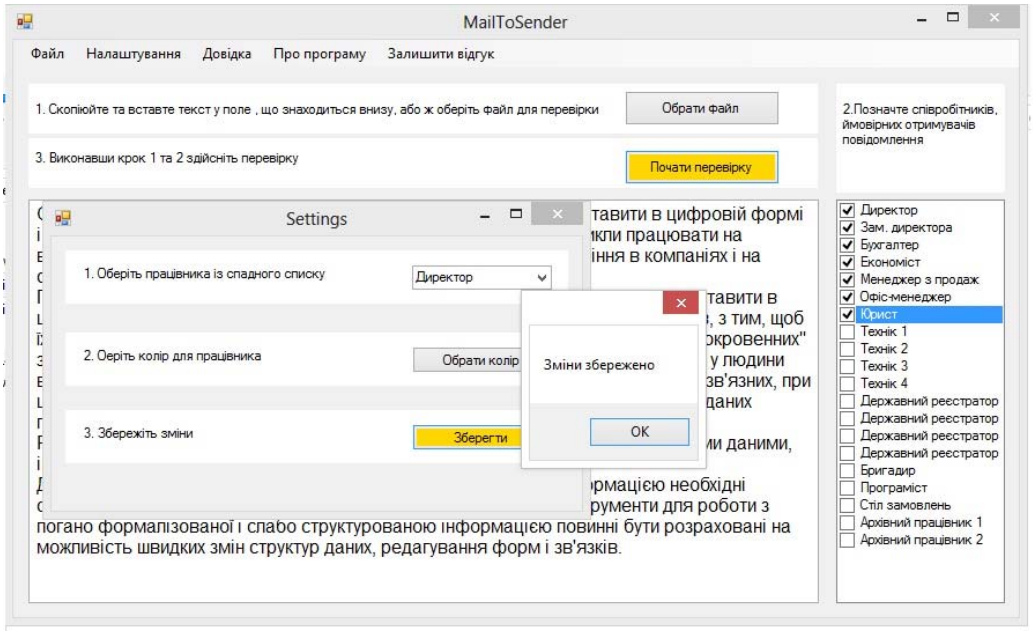

Рис. 3.11. Копія екранної форми для налаштувань системи

На рисунку 3.12. представлено аналіз документу із класифікацією, та виділенням тексту по відповідних класифікаційних ознаках.

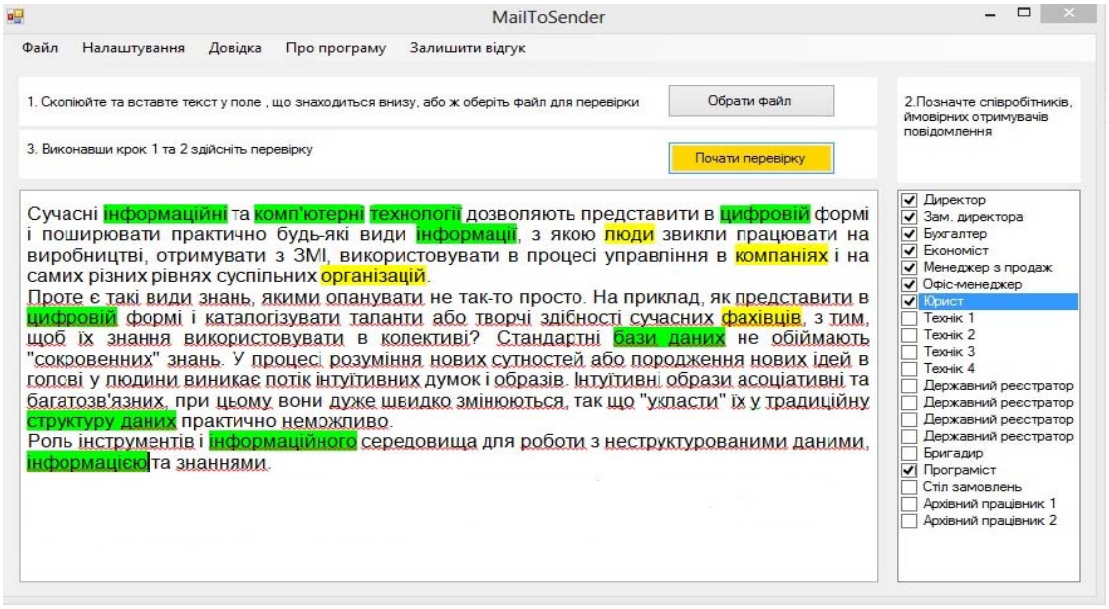

Рис. 3.12. Копія екранної форми з виділенням тексту документів по класифікаційних ознаках

Розроблена система має простий та зрозумілий інтерфейс і практично показує як можна імплементувати запропоновані методи класифікації документів в корпоративній інформаційній системі.

3.3. Експериментальні дослідження алгоритму класифікації на базі тесту Белла

У цьому розділі наведено результати розрахунку тесту Белла для текстів українською мовою для двох тематик: "домашні тварини" та "мова програмування". Тексти оброблялися наступним чином. По-перше, із сторінки витягувався текст, що стосується лише тематики статті і міститься у меню, змісті тощо. Далі з тексту вирізалися всі символи, що не є кириличними літерами і замінювалися на прогалини. Текст переводився в нижній регістр і всі слова в тексті були оброблені стемером Портера, щоб у більшості випадків врахувати одні й ті самі слова, але в різних формах. Після цього складався індекс слів, де кожному унікальному слову у статті зіставлялося число, що відповідає рядку або стовпцю в матриці HAL і будувалася сама матриця, за якою формувався вектор документа та вектора запиту.

Для оцінки працездатності алгоритму на базі тесту Белла було обрано Wikipedia і набір статей, згрупованих за невеликим числом тематик. Було обрано групи текстів присвячених тематиці "мови програмування" та "домашні тварини". Зокрема для першої групи використовувалися статті з назвами "C++, Java, Мова програмування, Програмування". Для другої групи використовувалися статті "Тварини, Домашні тварини, Коти та Собаки". Виконання оцінки близькості текстів виконувалося для назв документів та назв тематик.

Нижче наведено графік залежності результату тесту Белла від розміру вікна при побудові HAL-матриці для тематики "домашні тварини". Слід звернути увагу на те, що в межах усі чотири графіки наближаються до точки насичення, що дорівнює $2\sqrt{2}$. Очевидно це може пояснюватися тим, що зі збільшенням розміру вікна при обчисленні HAL матриці збільшується частка загальновживаних слів, які зустрічаються в текстах схожої тематики. Те саме можна сказати і про значення менше двох. Однак чотири графіки досягають зони "квантової заплутаності" в різні моменти. Так стаття "домашні тварини має таку ж назву що і запит набагато швидше досягає значень вище двох. Слідом йдуть статті “Собаки” та “Коти”, які містять інформацію саме про домашніх тварин. Нарешті, більш загальна стаття під назвою "Тварини" має найбільш пологий графік залежності тесту Белла від розміру вікна. Неважко помітити, що такий тест може бути використаний як ознака для порівняння кількох текстів на близькість до певної тематики чи текстового фрагмента. Слід наголосити, що великі значення по тесту Белла не означають велику заплутаність, проте ознакою присутності заданої тематики у статті може бути розмір вікна при обчисленні HAL-матриці, на якому можуть виявлятися ознаки “квантової заплутаності”.

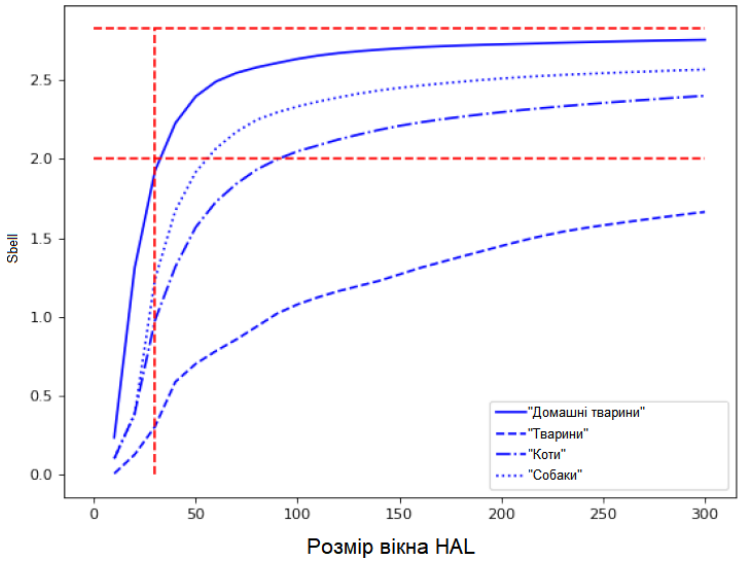

Рис. 3.13. Результати розрахунку параметра Белла для тематики "Домашні тварини"

Схожу ситуацію можна спостерігати у статтях з тематики "Мова програмування". Більш загальна за змістом стаття "Програмування" Також не досягає значень вище 2, в той час як решта з трьох статей досягають, але також при різних значеннях вікон для HAL. Цікаво, що для статті "Java" порогові значення лежать набагато далі, ніж для статей "C++" та "Мова програмування". Це зумовлено тим, що у статті Java терміни "мова" та "програмування" зустрічаються рідше, ніж у двох інших статтях. В цілому, підхід з використанням вимірів тесту Белла застосовується і в цьому випадку.

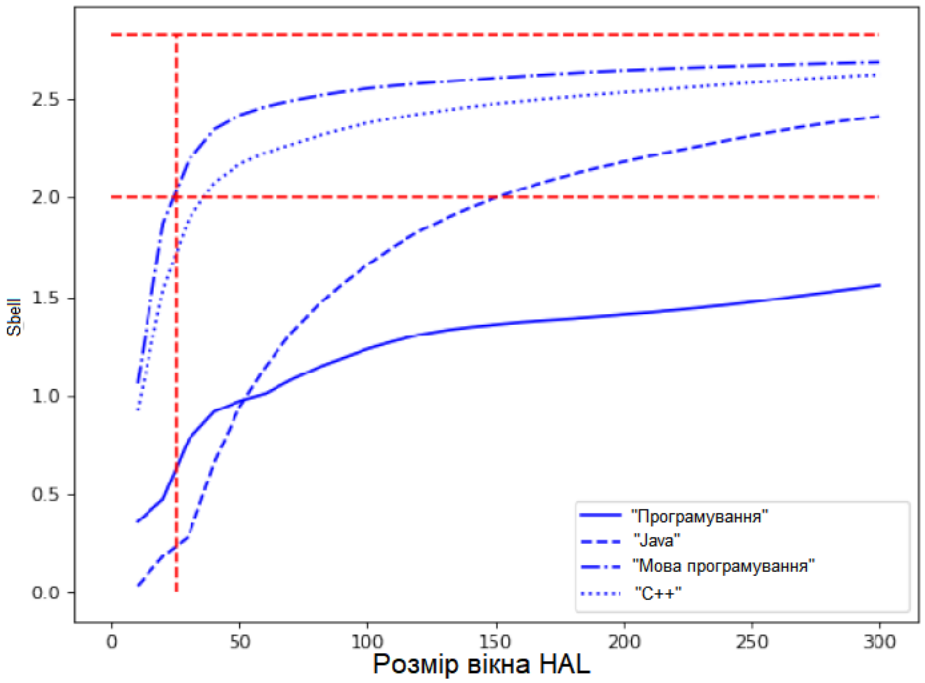

Рис. 3.14. Результати розрахунку параметра Белла для тематики "Мова програмування"

Результати розрахунку параметра Белла для текстових документів показують, що за певних умов, таких як розмір вікна контексту за яким будується матриця HAL, спостерігається порушення нерівності Белла, що говорить про наявність неврахованих залежностей у межах моделі, що використовується. Такий тест може бути використаний для оцінки близькості текстових документів. Для узагальнення розміру текстового фрагмента (в експерименті використовувалося лише два слова) можна використовувати

той самий алгоритм векторизації тексту, який застосовувався для векторизації документа.

У наступному експерименті проводилася оцінка тесту Белла на можливість поділу близьких за контекстом слів. Для цього експерименту було зібрано вибірку з 500 документів, розділених на різні тематики. Як документи були обрані статті з Wikipedia та підручники у форматі txt. У вибірці були представлені такі теми:

1. Історія; 2. Інформатика; 3. Біологія 4. Лінгвістика; 5. Екологія.

До всіх текстів застосовувалася описана вище попередня обробка і після нормалізації слів, для кожної тематики будувалася матриця взаємної інформації (PMI) "слово-слово". Для всіх пар тематик матриці PMI перетиналися за загальним словами і серед отриманого набору пар слів вибиралися пари слів для оцінки близькості. Ось приклад кількох таких пар слів:

1. Електроніка – кібернетика;
2. Гонка – озброєнь;
3. Оксид – азоту;
4. Чарльз – Дарвін;

Таким чином у кожній тематиці були представлені пари слів такі, що вони зустрічаються в кількох тематиках у вибірці, що розглядається. Далі, кожній парі слів зіставлявся набір тематик, який їй відповідає (наприклад для "оксид – азоту" підходять тематики "біологія" та "екологія") і в рамках контексту такої тематики ці слова приймалися схожими за значенням, в інших випадках – ні.

За всіма документами будувалися «локальні» матриці HAL із вікнами різних розмірів (від 5 до 95 з кроком 5). Наприкінці всі матриці для HAL одного розміру поєднувалися в «глобальну» матрицю HAL, що містить сумарну інформацію з усіх документів та тематиок. Локальна матриця HAL використовувалася для побудови вектора документа, а глобальна для побудови слова вектора.

Так само як і в попередньому експерименті для всіх HAL матриць, пар слів і документів виконувалося оцінювання за тестом Белла, але тепер отримані оцінки використовувалися як класифікатор з відсіканням за пороговим значенням тесту – чи слова близькі за значенням у контексті документа чи ні. Нижче наведено графік залежності метрик повноти, точності та f-міри для кращих порогових значень тесту Белла (тобто таких, що дають найбільші значення f-міри).

Очевидно, що зі зростанням розміру вікна загалом падає якість такого класифікування. Це пов'язано з тим, що при зростанні розміру вікна з'являється велика кількість складових шумів за рахунок включення більшого числа слів у контекст на більшій відстані. Це підтверджують і графіки, представлені вище (рисунок 3.15 і 3.16) – зі зростанням вікна всім документів параметр Белла зростатиме.

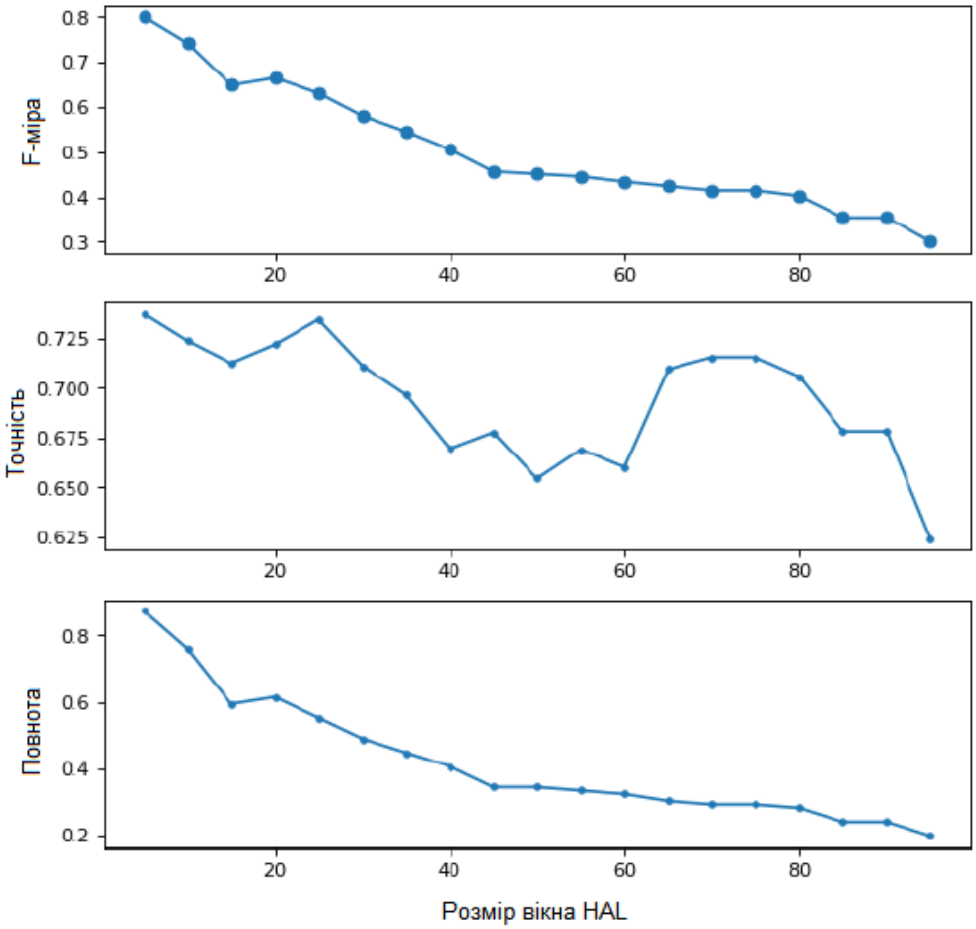

Рис. 3.15. Залежність якості класифікації за значенням тесту Белла від вікна HAL

Також корисно збудувати графік залежності точності від повноти (при змінному значенні порогу тесту Белла). HAL_sz – параметр розміру вікна, що змінюється, при побудові HAL.

Метрика виходить нечутливою у досить великому діапазоні значень тесту Белла, саме за рахунок цього при зміні повноти точність на графіці майже не змінюється.

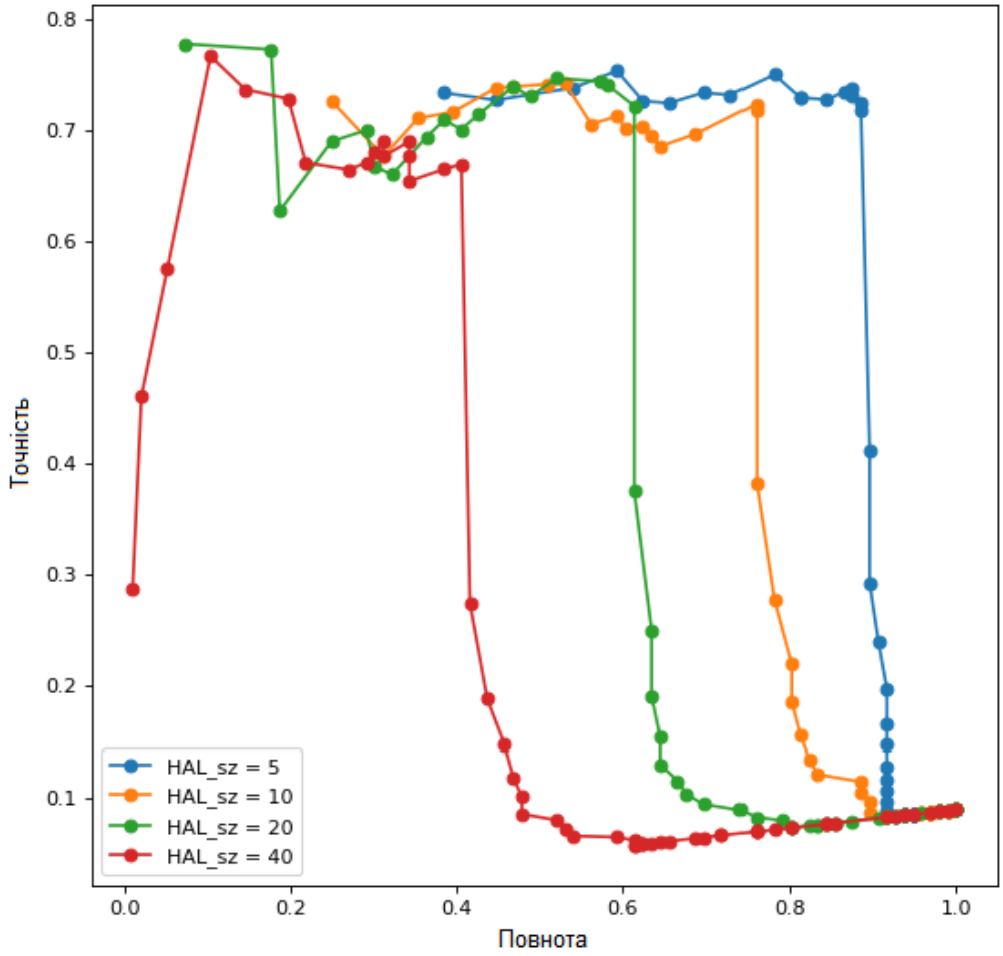

Рис. 3.16. Графік повноти-точності для класифікатора на основі тесту Белла

Загалом можна зробити висновок, що тест Белла задає досить великий проміжок між протилежними мітками.

Висновки до третього розділу

1. Апробовано метод оцінки ступеня близькості двох фрагментів текстів з використанням квантової аналогії та тесту Белла, показано застосування даного алгоритму до ранжування текстових документів в інформаційно-пошукових системах, а також показано можливості даного тесту в частині класифікації текстових фрагментів документів інформаційних систем.

2. Показано аналогію між поняттями квантової математики та аналізом текстових документів, а також між процесом векторизації слів текстових документів та процесом квантової томографії.

ВИСНОВКИ

Основні результати, які отримані в магістерському дослідженні:

1. Класифікація текстів є одним з основних завдань комп'ютерної лінгвістики, оскільки до неї зводиться ряд інших завдань: визначення тематичної приналежності текстів, автора тексту, емоційного забарвлення висловлювань та ін. Найбільш поширений сучасний підхід до класифікації ґрунтується на методах машинного навчання.

2. Методи класифікації текстів лежать на стику двох областей - інформаційного пошуку та машинного навчання. Їх схожість полягає у способах представлення самих документів та способах оцінки якості алгоритмів. На сьогоднішній день розроблено велику кількість методів та їх різних варіацій для класифікації текстів. Кожна група методів має свої переваги та недоліки, сфери застосування, особливості та обмеження.

3. В результаті виконаного огляду стану проблеми було встановлено, що дослідження в галузі аналізу текстових документів засобами математичного апарату квантової теорії ведуться за такими основними напрямками: моделювання семантики окремих слів та концептів з використанням апарату квантової теорії ймовірностей, моделювання семантики цілих виразів та пропозицій шляхом явного подання структурних зв'язків між словами.

4. Розглянуто математичний апарат, необхідний для визначення ступеня заплутаності слів, які перебувають у аналізованому ланцюжку з позицій апарату квантової теорії. Під заплутаністю мається на увазі ступінь семантичної близькості слів чи аналізованих концептів бази знань, які зустрічаються в одному смисловому контексті, зокрема у документі. Також варто відзначити можливі поліпшення щодо застосування нерівності Белла до зазначеного завдання.

5. Розглянуто аналогію між квантовою томографією та процесом побудови векторних представлень слів для текстових документів. Подібна аналогія дозволяє адаптувати математичний апарат, що використовується для опису процесу квантової томографії, який використовується для опису статистичних властивостей об'єктів, що мають квантово-подібні властивості.

З погляду моделювання семантики такий математичний апарат дозволяє врахувати властивості суперпозиції та заплутаності контекстів слів, які зустрічаються при векторизації аналізованого слова.

6. Апробовано метод оцінки ступеня близькості двох фрагментів текстів з використанням квантової аналогії та тесту Белла, показано застосування даного алгоритму до ранжування текстових документів в інформаційно-пошукових системах, а також показано можливості даного тесту в частині класифікації текстових фрагментів документів інформаційних систем. Показано аналогію між поняттями квантової математики та аналізом текстових документів, а також між процесом векторизації слів текстових документів та процесом квантової томографії.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chih-Hung Wu “Behavior-based spam detection using a hybrid method of rule-based Techniques and neural networks”, Expert Systems with Applications 36 4321– 4330 2019

2. Joachims, T. “Text categorization with support vector machines: learning with many relevant features”. In Proceedings of ECML-98, 10th European Conference on Machine Learning (Chemnitz, DE), pp. 137–142 1998.

3. Loubes, J. M. and van de Geer, S “Support vector machines and the Bayes rule in classification”, Data mining knowledge and discovery 6 259-275.2021

4. Chen donghui Liu zhijing, “A new text categorization method based on HMM and SVM”, IEEE2010 Yu-ping Qin Xiu-kun Wang, “Study on Multi-label Text Classification Based on SVM” Sixth International Conference on Fuzzy Systems and Knowledge Discovery 2009

5. Dagan, I., Karov, Y., and Roth, D. “Mistake-Driven Learning in Text Categorization.” In Proceedings of CoRR. 2017

6. MIgual E .Ruiz, Padmini Srinivasn, “Automatic Text Categorization Using Neural networks”, Advaces in Classification Research, Volume VIII.

7. Cheng Hua Li , Soon Choel Park “An efficient document classification model using an improved back propagation neural network and singular value decomposition”, Expert Systems with Applications, 3208–3215, 2019

8. Hwee TOU Ng Wei Boon Goh Kok Leong Low, “Feature Selection, Perception Learning, and a Usability Case Study for Text Categorization”, SIGIR 97 Philadelphia PA, Amy J.C. Trappey a, Fu-Chiang Hsu a, Charles V. Trappey b, Chia-I. Lin “Development of a patent document classification and search platform using a back-propagation network”, Expert Systems with Applications 31 755–765 2016