Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель класифікації текстових документів із адаптивним оновленням ключових слів вихідних класів
(тема)

Виконав:

студент II курсу, групи СПм-21-2

Могилевський Д.І.
(прізвище, ініціали)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва освітньої програми)

Керівник: доц. Барковська О.Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав.
кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Могилевському Дмитру Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Модель класифікації текстових документів із адаптивним оновленням ключових слів вихідних класів

затверджена наказом по університету від " 03 " квітня 2023 р. № 318 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 17 травня 2023 р.

3. Вхідні дані до роботи
Нейромережеві моделі для класифікації тексту
Навчальна вибірка
Тестова вибірка
Обчислювач на базі Google Colab та AMD Ryzen 7 5800H

4. Перелік питань, що потрібно опрацювати у роботі
Аналіз предметної області
Аналіз алгоритмів класифікації текстів
Запропонувати модель класифікації документів
Результати експериментів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайдів презентації – 11 шт.

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Проектування моделі класифікації	3 квітня – 10 квітня	
	кваліфікаційних робіт на основі		
	адаптивного оновлення ключових слів		
2	Визначення діапазону КС у частотному	11 квітня – 15 квітня	
	словнику		
3	Розробка формальної логіки проведення	16 квітня – 20 квітня	
	досліджень		
4	Виконання експериментів для визначення	21 квітня – 5 травня	
	впливу методів препроцесінгу та кількості		
	значущих слів на точність клсифікації		
5	Аналіз отриманих результатів	6 травня – 12 травня	
6	Підготовка пояснювальної записки	13 травня – 16 травня	

Дата видачі завдання 03 квітня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Барковська О.Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 69 с., 16 рис., 11 табл., 2 дод., 14 джерел.

СЛОВНИК, МОДЕЛЬ, TF-IDF, КЛАСИФІКАЦІЯ, ТЕКСТОВИЙ ДОКУМЕНТ.

Метою кваліфікаційної роботи є розробка моделі класифікації текстових документів із адаптивним оновленням ключових слів вихідних класів.

У ході виконання кваліфікаційної роботи проведено аналіз методів класифікації текстових документів. Запропоновано модель класифікації документів зі сховищем, на основі якого відбувається донавчання моделі.

Результати проведених експериментів показали, що кращим методом, який можна буде використовувати у моделі класифікації є стемінг, який має кращі результати для кожної протестованої теми. Авторські ключові слова теж показують хороші результати, але все ж таки менше за результат стемінгу на повному словнику та частковому. Також, виходячи з результатів, у модель класифікації для більш точнішої класифікації треба надати повному словнику більший пріоритет при прийняті рішення по класифікації документа.

ABSTRACT

Master’s thesis: 69 pages, 16 figures, 11 tables, 2 appendices, 14 sources.

DICTIONARY, MODEL, TF-IDF, CLASSIFICATION, TEXT DOCUMENT.

The major goal of this thesis is to develop a model for the classification of text documents with adaptive updating of keywords of the original classes.

In the course of the qualification work, an analysis of text document classification methods was carried out. A document classification model with storage is proposed, on the basis of which the model is retrained.

The results of the conducted experiments showed that the best method that can be used in the classification model is stemming, which has better results for each tested topic. Author keywords also show good results, but still less than the result of stemming on the full dictionary and partial. Also, based on the results in the classification model, for a more accurate classification, it is necessary to give the full dictionary a larger coefficient when deciding on the classification of the document.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АКС – авторскі ключові слова

CNN –convolutional neural network

CBOW – continuous bag of words

DNN –deep neural network

GloVe – global vectors for word representation

KNN – k-nearest neighbor

NLP – natural language processing

PCA – principal component analysis

POS – part of speech

RF – random forest

RNN – recurrent neural network

SVM – support vector machine

TF-IDF – term frequency-inverse document frequency

ВСТУП

За останні роки об'єм текстової інформації, що виробляється в Інтернеті, зріс в кілька разів, але рівень її якості залишається різним. Одним з найбільш важливих завдань при обробці текстової інформації є автоматична класифікація текстів. Це дозволяє ефективно обробляти великі об'єми даних, забезпечуючи швидкий та точний аналіз текстів.

Однак, багато існуючих методів класифікації текстів мають певні обмеження, зокрема, неможливість адаптуватися до змін вихідних класів. Тому метою даного дослідження є розробка моделі класифікації текстових документів із адаптивним оновленням ключових слів вихідних класів.

Основною метою цієї моделі є автоматичне оновлення ключових слів вихідних класів в разі зміни контексту текстів. Це дозволить досягти більш високої точності та ефективності в процесі класифікації текстів.

Галузі, де можна буде застосовувати цю модель:

- комп'ютерне зорове розпізнавання: текстова класифікація використовується для розпізнавання та класифікації текстових елементів у зображеннях або відео, таких як назви товарів на полицях магазинів або автомобільних номерних знаків;
- соціальні медіа та аналітика: текстова класифікація допомагає виявляти настрої, емоції та сутність повідомлень у соціальних мережах, блогах або новинах, що дозволяє визначити настрої громадської думки, відгуки про продукти або бренди, аналізувати тенденції тощо;
- фінансовий аналіз: текстова класифікація використовується для аналізу фінансових звітів, новин та коментарів з метою прогнозування фінансових ринків, виявлення ризикових чинників та класифікації компаній за їхньою фінансовою стабільністю;
- класифікація документів та пошукові системи: текстова класифікація використовується для автоматичної категоризації документів, що допомагає в

організації та швидкому пошуку інформації. Наприклад, в електронній пошті для автоматичного сортування вхідних повідомлень за категоріями або в пошукових системах для ранжування результатів пошуку;

- виявлення спаму: алгоритми текстової класифікації можуть бути використані для автоматичного фільтрування спаму у електронній пошті або соціальних мережах. Вони аналізують текст повідомлень та визначають, чи є вони спамом чи легітимними повідомленнями;

- аналіз настрою: текстова класифікація може бути використана для виявлення тону або настрою тексту, наприклад, відгуків користувачів, коментарів у соціальних мережах або відгуків про продукти. Це може бути корисно для визначення позитивного, негативного або нейтрального відношення до чого-небудь.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність класифікації текстів

У сучасному світі збільшується обсяг інформації, що надходить до нас з різних джерел, зокрема з Інтернету, електронних бібліотек та архівів. Класифікація цієї інформації стає дедалі важливішою задачею, що дозволяє забезпечити зручний доступ до неї, покращити організацію та пошук. Особливо важливою є задача класифікації текстових документів, що дозволяє автоматизувати процес розподілу текстів за темами та категоріями.

Однак існуючі підходи до класифікації текстів мають свої недоліки, зокрема не можуть ефективно працювати з динамічно змінюючимися даними, такими як новини або соціальні мережі. З цією метою можна використовувати методи з адаптивним оновленням ключових слів вихідних класів, що дозволяють підтримувати актуальність класифікації в часі (рисунок 1.1).

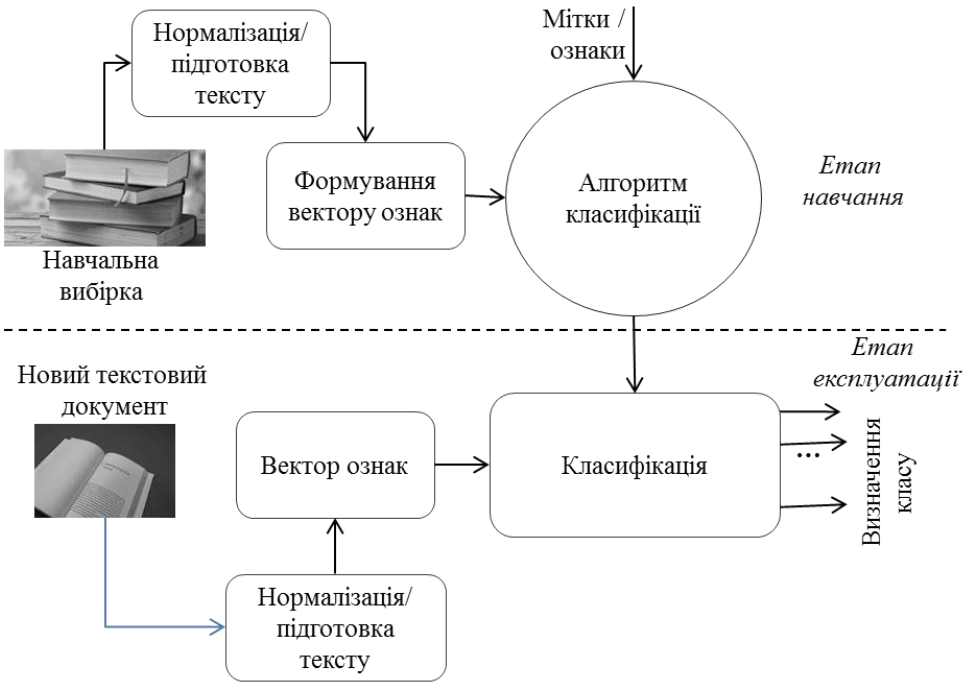

Рисунок 1.1 – Процес класифікації текстових документів

Окрім того, зростає значення розуміння контексту та зв'язків між текстовими документами. Використання моделі з адаптивним оновленням ключових слів вихідних класів може дозволити покращити точність класифікації та забезпечити більш ефективний пошук за пов'язаними темами та категоріями.

Отже, можна стверджувати, що тема моделі класифікації текстових документів із адаптивним оновленням ключових слів вихідних класів є актуальною та важливою для сучасного світу, що стикається зі зростаючим обсягом інформації та потребою у її ефективному організованому збереженні та пошуку [1].

Зі збільшенням обсягу текстової інформації в електронному вигляді продовжує зростати актуальність завдання автоматичної класифікації тексту. Це завдання виникає під час автоматичної обробки новинного потоку та розподілу текстів новин у каталогах (рисунок 1.2). Для зручності користувачів довідники організовані в ієрархічній структурі: довідник складається з кількох підкаталогів тощо.

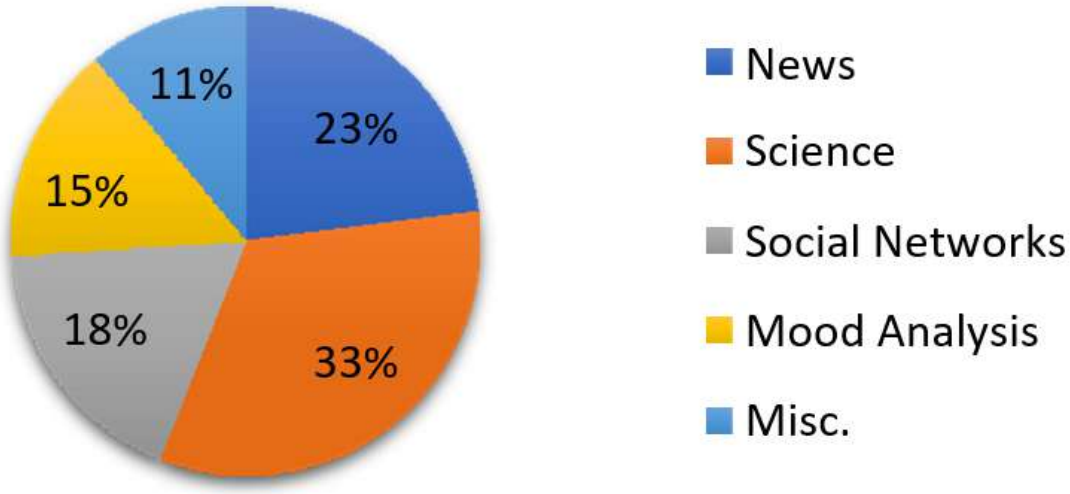

Рисунок 1.2 – Практичне застосування класифікаторів текстових масивів

Завдання класифікації особливо важливе в науковій галузі, де щорічно по кожній дисципліні додаються десятки тисяч монографій, статей, препринтів та

інших видань. Ефективне опрацювання таких масивів та якість пошуку матеріалів, актуальних для певного напряму дослідження, вимагають точного співвіднесення кожного видання з його тематичною категорією [2].

1.2. Галузі застосування текстової класифікації

1.2.1 Медична діагностика

Медична діагностика є однією з найважливіших галузей застосування текстової класифікації. За допомогою цієї технології можна класифікувати тексти медичних записів, що дозволяє швидко та ефективно знаходити необхідну інформацію для діагностики та лікування пацієнта.

Зокрема, за допомогою текстової класифікації можна класифікувати медичні записи за різними параметрами, такими як тип хвороби, стадія захворювання, причина захворювання тощо. Це дозволяє лікарям швидко та точно встановлювати діагнози та обирати необхідний курс лікування.

Крім того, текстова класифікація також застосовується для аналізу медичних звітів та наукових статей, що дозволяє виявляти нові зв'язки та закономірності між хворобами та їх лікуванням. Це може допомогти у розробці нових методів діагностики та лікування різних хвороб, що збільшує ефективність медичної практики та знижує ризик виникнення помилок.

Наприклад, застосування текстової класифікації може допомогти в діагностиці онкологічних захворювань, яка базується на аналізі великої кількості медичних документів. Також, класифікація може використовуватися для ідентифікації ризикових груп пацієнтів, що дозволяє розробити індивідуальний підхід до лікування та профілактики захворювань.

Отже, використання текстової класифікації в медичній діагностиці може покращити якість лікування та збільшити його ефективність, що робить цю галузь особливо важливою для досліджень та розробок у цій області.

1.2.2 Фінансова аналітика

Фінансова аналітика - це область застосування текстової класифікації, яка займається аналізом фінансових даних, зокрема текстових документів, щоб допомогти в процесі прийняття фінансових рішень.

Фінансова аналітика може використовуватися для класифікації фінансових звітів, новин про фінансові ринки, аналізу економічних тенденцій та споживчої поведінки. Наприклад, класифікація новин може допомогти відслідковувати події, які можуть вплинути на фінансові ринки, і приймати рішення на основі цієї інформації.

Також, фінансова аналітика може використовуватися для класифікації документів, що містять інформацію про інвестиції, щоб допомогти інвесторам приймати рішення про потенційні інвестиції або ризики, пов'язані з інвестуванням.

Фінансова аналітика також може використовуватись для виявлення фінансових ризиків та оцінки фінансового стану компаній. Наприклад, текстова класифікація може допомогти відокремити звіти компаній, які містять негативну інформацію, таку як падіння прибутку, збитки або інші фінансові проблеми. Це може допомогти інвесторам та фінансовим аналітикам приймати рішення щодо інвестування коштів або рекомендацій щодо покупки або продажу акцій певних компаній.

Крім того, текстова класифікація може використовуватися для автоматизованого аналізу великої кількості фінансових даних, що допоможе виявити небезпеки та ризики, пов'язані з фінансовою діяльністю компаній, і допоможе уникнути можливих фінансових проблем.

Отже, фінансова аналітика може забезпечити ефективне використання фінансових ресурсів, допомогти зменшити ризики та прийняти обґрунтовані фінансові рішення.

1.2.3 Аналіз соціальних медіа

Аналіз соціальних медіа є ще однією важливою галуззю застосування текстової класифікації. Соціальні медіа включають у себе безліч повідомлень, коментарів, відгуків та інших даних, які можуть бути важливі для бізнесу або організації. Аналіз текстових даних з соціальних медіа може допомогти виявити настрої та погляди користувачів, їх реакції на новини, продукти та послуги.

Наприклад, класифікація коментарів до певного продукту на позитивні та негативні може допомогти виявити сильні та слабкі сторони продукту та покращити його якість. Крім того, аналіз соціальних медіа може допомогти вивчити конкурентів та їхні стратегії, що може бути корисним для формування власної стратегії та планування маркетингових заходів.

Також важливим аспектом аналізу соціальних медіа є виявлення впливових осіб, які можуть мати значний вплив на думки та погляди користувачів. Застосування методів машинного навчання та аналізу текстів може допомогти виявити таких осіб та розуміти їхні погляди та дії.

Отже, аналіз соціальних медіа з використанням текстової класифікації є важливим інструментом для розвитку бізнесу та дослідження думок та настроїв користувачів.

1.2.4 Навчання та освіта

ласифікація текстів знаходить застосування в навчанні та освіті. Наприклад, використання класифікації текстів може допомогти в аналізі текстів студентських робіт та автоматичному оцінюванні якості написання, що дозволяє вчителям і викладачам зосередитись на інших аспектах навчання, які вимагають більшої уваги.

Також, класифікація текстів може бути корисною для автоматичного визначення тематики курсів та підготовки рекомендацій щодо вивчення

конкретних предметів, що допоможе студентам вибрати оптимальний набір курсів для своєї спеціальності.

Крім того, класифікація текстів може бути використана для покращення процесу пошуку інформації в бібліотеках та в Інтернеті. Наприклад, системи класифікації можуть автоматично визначати тему документа та його важливість, що дозволяє швидко і ефективно знаходити необхідну інформацію.

1.3. Мета роботи

Метою даної кваліфікаційной роботи є розробка адаптивної моделі класифікації кваліфікаційних робіт.

Для досягення поставленої мети мають бути вирішені наступні задачі:

- аналіз предметної області;
- визначення діапазону TF словнику, у який потрапляють авторські ключові слова;
- розробка моделі класифікації із адаптивним оновленням ключових слів;
- експериментальні дослідження для визначення впливу лематизації, стемінгу та кількості значущих слів на точність класифікації;
- аналіз отриманих результатів.

2 АНАЛІЗ АЛГОРИТМІВ КЛАСИФІКАЦІЇ ТЕКСТІВ

Класифікація тексту – процедура позначення попередньо визначених міток для тексту – є важливим і важливим завданням у багатьох програмах обробки природної мови , таких як аналіз настроїв , позначення тем, відповіді на запитання та класифікація діалогового акту. В епоху інформаційного вибуху обробляти та класифікувати великі обсяги текстових даних вручну займає багато часу та є проблемою. Крім того, на точність ручної класифікації тексту можуть легко вплинути людські фактори, такі як втома та досвід. Бажано використовувати методи машинного навчання для автоматизації процедури класифікації тексту, щоб отримати більш надійні та менш суб'єктивні результати. Крім того, це також може допомогти підвищити ефективність пошуку інформації та полегшити проблему перевантаження інформацією шляхом визначення місцезнаходження необхідної інформації.

Показана блок-схему процедур (рисунок 2.1), задіяних у класифікації тексту, у світлі традиційного та глибокого аналізу. Текстові дані відрізняються від числових, графічних або сигнальних даних. Це вимагає ретельного опрацювання технік NLP. Першим важливим кроком є попередня обробка текстових даних для моделі.

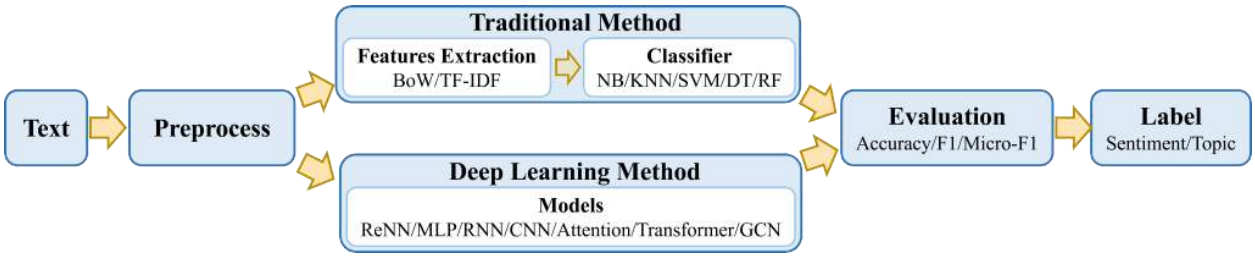

Рисунок 2.1- Блок-схема класифікації тексту з класичними методами в кожному модулі

Традиційні моделі, як правило, потребують штучних методів для отримання хороших зразкових характеристик, а потім класифікації за допомогою класичних алгоритмів машинного навчання. Таким чином, ефективність методу значною мірою обмежена вилученням ознак. Однак, на відміну від традиційних моделей, глибоке навчання інтегрує розробку функцій у процес підгонки моделі шляхом вивчення набору нелінійних перетворень, які служать для відображення функцій безпосередньо на виходах.

З 1960-х до 2010-х років домінували традиційні моделі класифікації тексту. Традиційні методи означають моделі на основі статистики, такі як Naive Bayes, KNN і SVM. У порівнянні з попередніми методами, заснованими на правилах, цей метод має очевидні переваги в точності та стабільності. Однак ці підходи все ще потребують розробки функцій, що займає багато часу та коштує. Крім того, вони зазвичай ігнорують природну послідовну структуру або контекстну інформацію в текстових даних, що ускладнює вивчення семантичної інформації слів. З 2010-х років класифікація тексту поступово змінилася від традиційних моделей до моделей глибокого навчання. У порівнянні з методами, заснованими на традиційних методах, методи глибокого навчання уникають розробки правил і функцій людьми й автоматично забезпечують семантично значущі представлення для аналізу тексту. Таким чином, більшість дослідницьких робіт з класифікації текстів базуються на глибоких нейронних мережах (DNN), які є підходами, керованими даними, з високою обчислювальною складністю. Небагато робіт зосереджені на традиційних моделях для вирішення обмежень обчислень і даних.

Класифікація тексту називається виділенням ознак із необроблених текстових даних і прогнозуванням категорій текстових даних на основі таких ознак. За останні кілька десятиліть було запропоновано багато моделей для класифікації тексту. Для традиційних моделей NB є першою моделлю, яка використовується для завдання класифікації тексту. Після цього пропонуються загальні моделі класифікації, такі як KNN , SVM і RF , які називаються класифікаторами та широко використовуються для класифікації тексту.

Нещодавно eXtreme Gradient Boosting і Light Gradient Boosting Machine , можливо, мають потенціал для забезпечення чудової продуктивності. Для моделей глибокого навчання TextCNN має найбільшу кількість посилань у цих моделях, де модель згорткової нейронної мережі була введена для вирішення проблеми класифікації тексту вперше. Хоча двонаправлене подання кодувальника від Transformers (BERT) не розроблено спеціально для обробки завдань класифікації тексту, воно широко використовується при розробці моделей класифікації тексту, враховуючи його ефективність на численних наборах даних класифікації тексту [3].

2.1 TF-IDF

TF-IDF – це статистичний показник, який оцінює релевантність слова документу в колекції документів.

Це робиться шляхом множення двох показників: скільки разів слово з’являється в документі та зворотної частоти цього слова в наборі документів.

Він має багато застосувань, особливо в автоматизованому аналізі тексту, і дуже корисний для підрахунку слів у алгоритмах машинного навчання для NLP.

TF-IDF був винайдений для пошуку документів та отримання інформації. Він працює, пропорційно збільшуючи кількість разів, коли слово з’являється в документі, але компенсується кількістю документів, які містять це слово. Отже, слова, які часто зустрічаються в кожному документі, як-от це, що та якщо, мають низький рейтинг, навіть якщо вони можуть з’являтися багато разів, оскільки вони не мають великого значення для цього документа зокрема.

Однак, якщо слово «Помилка» з’являється багато разів у документі, а в інших не з’являється багато разів, це, ймовірно, означає, що воно дуже актуальне. Наприклад, якщо ми намагаємося з’ясувати, до яких тем належать деякі відповіді NPS, слово «Помилка», ймовірно, буде пов’язано з темою «Надійність», оскільки більшість відповідей, що містять це слово, стосуватимуться цієї теми.

TF-IDF для слова в документі обчислюється множенням двох різних показників:

- термін частоти слова в документі. існує кілька способів обчислення цієї частоти, найпростішим з яких є необроблена кількість випадків, коли слово з'являється в документі. Крім того, існують способи налаштування частоти за довжиною документа або за частотою необробленого слова, яке найчастіше зустрічається в документі;

- зворотна частота слова в документі в наборі документів. Це означає, наскільки поширеним або рідкісним є слово у всьому наборі документів. Чим ближче до 0, тим більш поширеним є слово. Цей показник можна обчислити, взявши загальну кількість документів, поділивши її на кількість документів, які містять слово, і обчисливши логарифм.

Застосування TF-IDF

- інформаційний пошук. TF-IDF було створено для пошуку документів і може використовуватися для надання результатів, які найбільше відповідають тому, що шукається. Є пошукова система, і хтось шукає «безпека». Результати відображатимуться в порядку відповідності. Це означає, що найрелевантніші спортивні статті матимуть вищий рейтинг, оскільки TF-IDF дає вищу оцінку слову «безпека»;

- вилучення ключових слів. TF-IDF також корисний для вилучення ключових слів із тексту. Слова документа з найвищими балами є найбільш релевантними для цього документа, і тому їх можна вважати ключовими словами для цього документа[4].

2.1.1 Виявлення спаму за допомогою TF-IDF

Основна ідея включення TF-IDF і алгоритму стемінгу Портера в реалізацію алгоритму Наївного Байєса для класифікації спаму полягає в тому, щоб підвищити точність і швидкість обробки алгоритму, оскільки цей метод знаходить дійсні умови за допомогою алгоритму стеммінгу. Стоп-слова

видаляються, а корінь дійсних слів доводиться до кореневої форми за допомогою алгоритму Stemming Портера. Потім частоти термінів і релевантність цих дійсних термінів використовуються для формування векторної таблиці за допомогою методу TF-IDF на самій фазі попередньої обробки, тим самим додатково скорочуючи час попередньої обробки. TF-IDF разом із класифікатором Naive Bayes реалізовано для отримання оптимальних результатів. Як відомо, наївний алгоритм Байєса є простим і ефективним алгоритмом категоризації (рисунок 2.2).

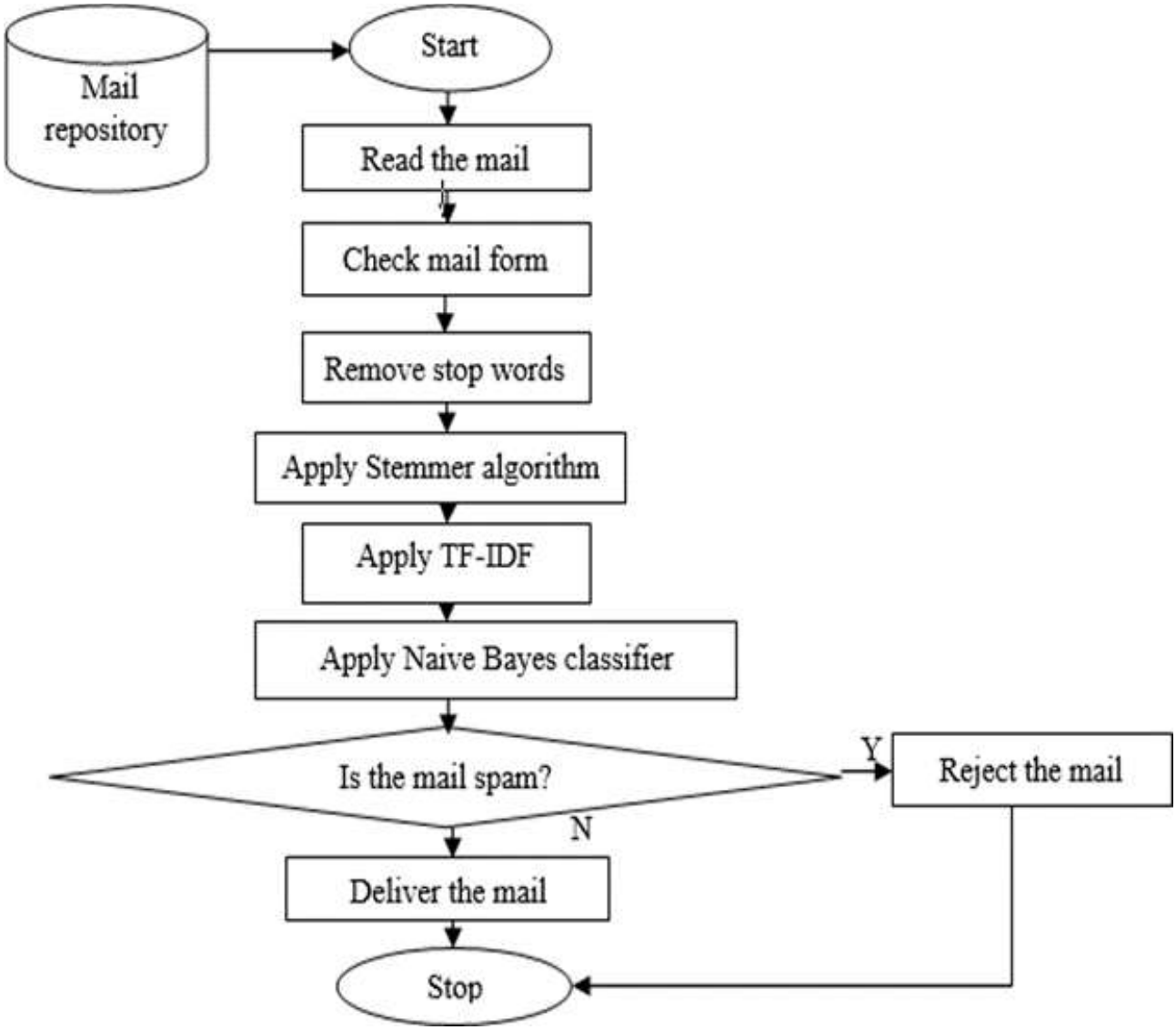

Рисунок 2.2 – Блок-схема процесу фільтрації пошти

Щоб підвищити продуктивність категоризації алгоритму Наївного Байєса в категоризації тексту, заснованого на зважуванні атрибутів TF-IDF, використовується. На малюнку 1 ми показуємо, як працює наша модель, використовуючи чотири методи, а саме видалення стоп-слів, стемінг, TF-IDF і наївний Байєс. Процес починається з отримання листа в папку «Вхідні» користувача, після чого поштова форма перевіряється на наявність повторюваних слів і стоп-слів, які здебільшого нерелевантні для передбачення. Слова, які можуть бути змінені на свою кореневу форму, поділяються на корінь за допомогою алгоритму коріння, після якого до слів із корінням застосовується попередня обробка TF-IDF. Після створення векторної таблиці її перевіряють на наявність спаму за допомогою класифікатора Naive Bayes. Якщо виявляється, що це спам, він відхиляється; в інших випадках пошта доставляється до папки "Вхідні" користувача [5].

2.2 GloVe

Статистика появи слів у корпусі є основним джерелом інформації, доступною для всіх неконтрольованих методів вивчення подання слів, і хоча зараз існує багато таких методів, все ще залишається питання про те, як значення генерується з цієї статистики, і як результуючі вектори слів можуть представляти це значення. У цьому розділі ми проливаємо світло на це питання. Ми використовуємо наші знання, щоб побудувати нову модель представлення слів, яку ми називаємо GloVe, для глобальних векторів, оскільки глобальна статистика корпусу фіксується безпосередньо моделлю. Спочатку ми встановимо деякі позначення. Нехай матриця підрахунків сумісного входження слово-слово буде позначена X , чиї записи $X_{i\ j}$ табулюють кількість разів, коли слово j зустрічається в контексті слова i. Нехай $X_i = k\ X_{ik}$ буде кількістю разів, коли будь-яке слово з’являється в контексті слова i. Нарешті, нехай $Pi\ j = P(\ j\ i) = X_{ij}/X_i$ буде ймовірністю того, що слово j з’явиться в контексті слова i [6].

2.3 Методи на основі KNN

В основі алгоритму KNN лежить класифікація непозначеної вибірки шляхом знаходження категорії з найбільшою кількістю вибірок на k найближчих вибірках. Це простий класифікатор без створення моделі та може зменшити складність через швидкий процес отримання k найближчих сусідів. (рисунок 2.3).

Ми можемо знайти k навчальних текстів, що наближаються до конкретного тексту, який потрібно класифікувати, оцінюючи проміжну відстань. Отже, текст можна розділити на найпоширеніші категорії, які зустрічаються в k текстах навчальних наборів. Удосконалення алгоритму KNN в основному включає схожість ознак, значення K та оптимізацію індексу. Однак через позитивну кореляцію між часово-просторовою складністю моделі та кількістю даних, алгоритм KNN займає надзвичайно багато часу на великомасштабних наборах даних.

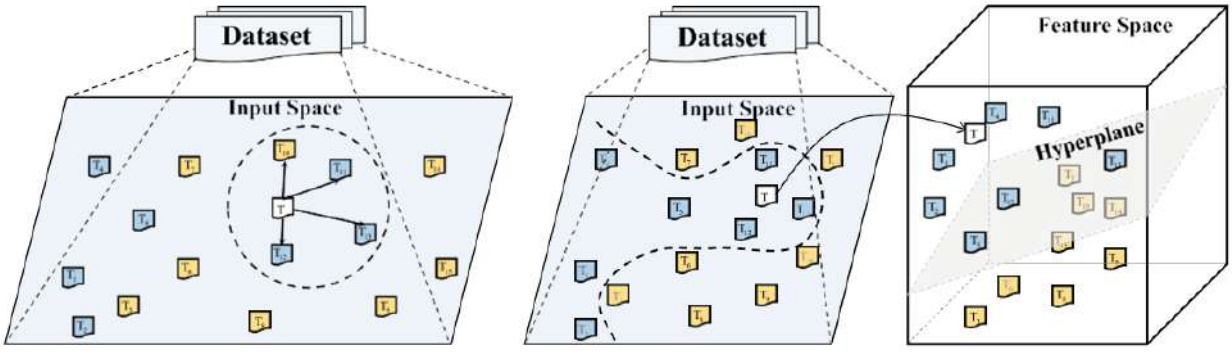

Рисунок 2.3 - Структура KNN де k = 4 (ліворуч) і структура SVM (праворуч). Кожен вузол представляє текст, а вузли з різними контурами представляють різні категорії

Щоб зменшити кількість вибраних ознак, пропонують алгоритм KNN без зважування ознак. Йому вдається знаходити релевантні ознаки, створюючи взаємозалежності слів за допомогою вибору ознак. Коли дані розподілені

надзвичайно нерівномірно, KNN прагне класифікувати вибірки з більшою кількістю даних. Для покращення ефективності класифікації незбалансованих корпусів запропоновано K-зважений K-найближчий сусід (NWKNN). Це надає значну вагу для сусідів у невеликій категорії та невелику вагу для сусідів у широкому класі [3].

2.4 Методи на основі SVM

Кортес і Вапнік пропонують SVM для вирішення бінарної класифікації розпізнавання образів. Joachims вперше використовує метод SVM для класифікації тексту, представляючи кожен текст як вектор. Як показано (рисунок 2.2), підходи на основі SVM перетворюють завдання класифікації тексту на завдання класифікації кількох бінарних даних. У цьому контексті SVM створює оптимальну гіперплощину в одновимірному вхідному просторі або просторі ознак, максимізуючи відстань між гіперплощиною та двома категоріями навчальних наборів, таким чином досягаючи найкращої здатності до узагальнення. Мета полягає в тому, щоб відстань межі категорії вздовж напрямку, перпендикулярного до гіперплощини, була найбільшою. Рівноцінно, це призведе до найнижчого рівня помилок класифікації. Побудова оптимальної гіперплощини може бути перетворена в задачу квадратичного програмування для отримання глобально оптимального рішення. Вибір відповідної функції ядра і вибір функцій є надзвичайно важливими для того, щоб SVM міг мати справу з нелінійними проблемами та стати надійним нелінійним класифікатором. Крім того, метод активного навчання та адаптивного навчання використовується для класифікації тексту, щоб зменшити зусилля з маркування на основі алгоритму навчання під наглядом SVM. Щоб проаналізувати, чому навчаються алгоритми SVM і які завдання підходять, Йоахімс пропонує теоретичну модель навчання, яка поєднує статистичні ознаки з узагальнювальною продуктивністю SVM, аналізуючи особливості та переваги за допомогою кількісного підходу. Трансдуктивна опорна векторна машина

пропонується зменшити помилкову класифікацію окремих наборів тестів із загальною функцією прийняття рішень з урахуванням конкретного набору тестів. Він використовує попередні знання, щоб створити більш відповідну структуру та швидше навчатися.[3]

2.5 Стоп-слово та видалення шуму

Набір токенів, створений під час процедури токенізації, може містити непотрібні або оманливі елементи. Необхідно видалити текстовий шум, наприклад спеціальні символи або зайві символи. Може бути корисним видалити стоп-слова , тобто неінформативні слова, які з’являються у великій кількості, але не мають семантичного значення. Інші процедури нормалізації, такі як нижній регістр, виправлення орфографічних помилок і стандартизація сленгових слів і абревіатур, можуть бути корисними для зменшення кількості різних елементів у просторі ознак.

Ці процеси, однак, можуть спричинити проблеми з тлумаченням (наприклад, англійською мовою «US» можна використовувати як акронім «United States», але малий регістр зробить його невідрізнимим від «us» як займенника) . Крім того, видалення стоп-слів та інших фрагментів тексту може зашкодити новішим підходам. Глибинні моделі навчені розуміти природну мову; елементи, які складають синтаксис речення, можуть бути фундаментальними для моделі для розуміння контексту частини тексту. Таким чином, деструктивні операції не повинні застосовуватися без розбору.[7]

2.6 Токенізація

Основна операція попередньої обробки, яку необхідно застосувати до тексту, це токенізація. Ця процедура визначає рівень деталізації, на якому ми аналізуємо та генеруємо текстові дані, і її можна загалом описати як процес розбиття потоку тексту на менші фрагменти (історично їх називають токенами).

Донедавна більшість моделей NLP використовували слова як атомарну одиницю вибору, але останні підходи полягали в розкладанні тексту на менші одиниці (такі як n-грами символів або навіть більш максимальні форми декомпозиції, такі як базові байти).

Хоча токенізація часто сприймається як належне, сама по собі є складним питанням і широко досліджується. Звичайні методи базуються на правилах і можуть бути такими ж простими, як розділення пробілами, знаками пунктуації та скороченнями. Очевидно, що були розроблені набагато більш витончені підходи, засновані на знаннях, які все ще базуються на лінгвістичних концепціях. Однак нещодавні розробки значно зрушили в бік керованих даними токенізаторів, які дають кращі результати, але в яких токени більше не відповідають традиційному визначенню друкарської одиниці. Іншими словами, сучасне значення «токенізації» стосується завдання сегментації речення на одиниці, які, що важливо, не мають лінгвістичної мотивації чи пояснення. Оскільки деякі з цих методів все ще використовують традиційні токенизатори як початковий крок, звичайні підходи тепер часто називають «попередніми токенизаторами».[7]

2.7 Мішок слів

Найбільш інтуїтивно зрозуміле представлення можна знайти в сумці слів . Цей підхід спрощує частини тексту, розглядаючи їх як невпорядковані набори слів. Очевидно, що це має недолік ігнорування структури речення та семантичних зв’язків між елементами речення (ніби перемішаними всередині «мішка» слів). Тим не менш, незважаючи на сильні припущення, було показано, що він дає хороші результати та широко використовується. Загалом, ефективне виділення ознак зазвичай може призвести до високої продуктивності, навіть якщо відкинути важливу, але складну для інкапсуляції інформацію, саме тому цей підхід широко використовується в NLP, а також в інших сферах машинного навчання (де він згадується як «мішок»).

Оригінальна ідея моделей BoW полягає в тому, щоб представити кожне слово як вектор з одним гарячим кодуванням такого ж розміру, як і словниковий запас. Відразу ж видно, що це може призвести до проблем із розміром, оскільки сам словниковий запас може мати кількість мільйонів. Таким чином, моделі BoW зазвичай реалізуються в поєднанні з методами виділення ознак на основі множинності слів, що дозволяє підтримувати один вектор для кожного документа, а не один для кожного слова. З цією метою частота термінів підраховує кількість разів, коли слово зустрічається в тексті. При застосуванні до корпусу текстів, а не для явного підрахунку, звичайною практикою є використання відносної частоти терміна в тексті по відношенню до інших документів. Можна також помітити, що в дуже великих корпусах загальні слова, зокрема, за своєю суттю менш корисні (оскільки вони зустрічаються в усіх документах, отже, не допомагають їх розрізняти). Отже, TF часто зважується за IDF, що зменшує вплив загальних слів, штрафуючи їхню загальну оцінку (і підвищуючи оцінку рідкісних слів).

Залежно від розміру словника, розмір представлень TF-IDF все ще може бути надмірно великим. Щоб полегшити проблеми, пов’язані зі складністю часу та споживанням пам’яті, можна встановити обмеження на максимальну кількість функцій у векторах (ефективно видаляючи слова з низькими оцінками зі словника). Крім того, також можна використовувати алгоритм зменшення розмірності на повнорозмірних представленнях. Хоча детальний опис виходить за рамки цього огляду, загальна мета цих алгоритмів полягає в тому, щоб знайти відображення між цими представленнями та низьковимірним, стислим простором ознак. Популярні методи включають аналіз головних компонентів PCA, лінійний дискримінантний аналіз і факторізацію невід’ємної матриці .[7]

2.8 Огляд NLP

Обробка природної мови (NLP) — це теоретично вмотивований набір обчислювальних методів для автоматичного аналізу та представлення людської

мови. Дослідження НЛП еволюціонували від ери перфокарт і пакетної обробки (у якій аналіз речення може тривати до 7 хвилин) до ери Google і подібних до неї (у якій мільйони веб-сторінок можна обробити менше ніж секунда).

З моменту народження Інтернету до 2003 року, року народження таких соціальних мереж, як MySpace, Delicious, LinkedIn і Facebook, у Мережі було лише кілька десятків ексабайтів інформації. Сьогодні стільки ж інформації створюється щотижня. Поява соціальної мережі надала людям нові послуги обміну контентом, які дозволяють їм створювати та ділитися власним вмістом, ідеями та думками економічно за часом і з мільйонами інших людей. до Всесвітньої павутини. Цей величезний обсяг інформації, однак, переважно неструктурований (оскільки він спеціально створений для споживання людиною) і, отже, не піддається прямій машинній обробці. Автоматичний аналіз тексту передбачає глибоке розуміння природної мови машинами, реальність, до якої ми ще дуже далекі. До цього часу пошук, агрегація та обробка інформації в режимі он-лайн базувалися переважно на алгоритмах, які спиралися на текстове представлення веб-сторінок. Такі алгоритми дуже добре справляються з пошуком текстів, розбиттям їх на частини, перевіркою орфографії та підрахунком кількості слів. Однак, коли справа доходить до тлумачення речень і вилучення значущої інформації, їхні можливості, як відомо, дуже обмежені [8].

2.8.1 NLP та нейронне навчання

Система NLP використовує речення природною мовою та генерує тип класу (для завдань класифікації), послідовність міток (для завдань маркування послідовності) або інше речення (для забезпечення якості, діалогу, генерації природної мови). Для застосування нейронних підходів НЛП необхідно вирішити наступні дві ключові проблеми:

- закодуйте речення природної мови (послідовність слів) у нейронній мережі;

- створити послідовність міток або інше речення природною мовою.

Вбудовування слів відображає слова у вхідних реченнях у безперервні просторові вектори. Базуючись на вбудовуванні слів, складні мережі, такі як рекурентні нейронні мережі RNN, CNN і мережі самоконтролю, можна використовувати для виділення ознак, враховуючи контекстну інформацію цілого речення для створення контекстно-залежного слова. вбудовування або інтегрування всієї інформації речення для створення вбудовування речення. Контекстно-залежне вбудовування слів можна використовувати для завдань послідовного маркування, таких POS і розпізнавання іменованих об’єктів, а вбудовування речень можна використовувати для завдань на рівні речень, таких як аналіз настроїв і перефразування класифікація. Вбудовування речень також можна використовувати як вхідні дані для іншої RNN або мережі самоконтролю для генерації іншої послідовності, яка формує структуру кодера-декодера для моделювання послідовності до послідовності. Враховуючи вхідне речення, моделювання послідовності до послідовності можна використовувати для створення відповіді на запитання або для виконання перекладу іншою мовою.

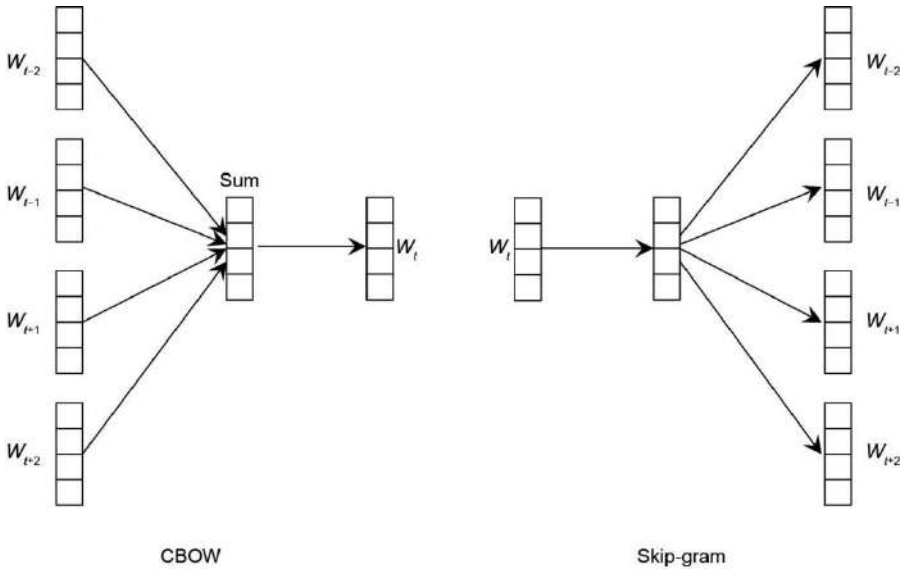

Рисунок 2.4 – Контекстно-незалежні методи вбудовування слів. CBOW: використання контекстних слів у вікні для передбачення центрального слова. Skip-gram: використання центрального слова для передбачення контекстних слів у вікні

Щоб зіставити слово в безперервний семантичний вектор, запропонували моделі CBOW і skip-gram, на основі яких інструмент впровадження word2vec використовується для вивчення векторів вбудовування слів із великим одномовним корпусом. Як показано (рисунок 2.4), модель CBOW передбачає центральне слово, використовуючи навколишні слова у вікні, тоді як модель пропуску грам передбачає слова, що оточують дане слово. Ці дві моделі розроблені на основі принципу «knowing a word by the company it keeps». Крім того, щоб використовувати переваги глобальної статистики спільного входження та значущих лінійних підструктур, запропонували модель GloVe для вивчення вбудовування слів.

Word2vec і GloVe вивчають постійний вектор вбудовування слова; вкладання однакове для слова в різних реченнях. Наприклад, ми можемо дізнатися вектор вкладення для слова «bank», і вкладення залишається незмінним, незалежно від того, чи використовується слово «bank» у реченні «an ant went to the river bank» чи в реченні «that is a good way to build up a bank account». Нібито вкладання слова «bank» у першому реченні має відрізнятися від слова «bank» у другому реченні. Для вирішення цієї проблеми контекстна інформація речення використовується для прогнозування динамічного вбудовування слова [9].

2.8.2 Порівняння NLP бібліотек

Розглянемо бібліотеки NLP, де реалізовано базові методи процесінгу тексту (таблиця 2.1) – токенізація, POS-тегування, розпізнавач іменованих сутностей, розбір залежностей, Text Matcher, Chunking, перевірка орфографії, детектор почуттів, попередньо підготовлені моделі, нейронна мережа, підтримка навчання на GPU.

Підсумовуючи, можна сказати, що SpaCy і SparkNLP потужні інструменти для повного конвеєра NLP і швидкої обробки.

Таблиця 2.1 – Порівняльний аналіз переваг та недоліків NLP бібліотек

Назва бібліотеки	Переваги	Недоліки
Natural Language ToolKit	-найпопулярніша та повна NLP бібліотека; -багато сторонніх розширень; -безліч підходів до кожного завдання NLP; -швидка токенізація речень; -підтримує найбільшу кількість мов порівняно з іншими бібліотеками.	-повільна; -складна для вивчення та використання; -не підтримує моделі нейронних мереж; -немає інтегрованих векторів слів; -у токенізації речень NLTK лише розділяє текст на речення, не аналізуючи семантичну структуру; -обробляє рядки, що не дуже характерно для об'єктно-орієнтованої мови Python.
SpaCy	-найшвидший NLP фреймворк; -активна підтримка та розробка проєкту; -проста у вивченні та використанні, оскільки він має один оптимізований інструмент для кожного завдання; -обробляє об'єкти; більш об'єктно-орієнтована, порівняно з іншими бібліотеками; -використовує нейронні мережі для навчання деяких моделей; -забезпечує вбудовані вектори слів.	-не вистачає гнучкості, порівняно з NLTK; -токенізація речень повільніша, ніж у NLTK; -не підтримує багато мов. Існують моделі тільки для 7 мов і "багатомовні" моделі.
Gensim	-працює з великими базами даних і обробляє потоки даних; -забезпечує векторизацію tf-idf, word2vec, document2vec, прихований семантичний аналіз, прихований розподіл Діріхле; -підтримує глибоке навчання.	-призначений для неконтрольованого (unsupervised) моделювання тексту; -не має достатньо інструментів для забезпечення повного конвеєра NLP, тому його слід використовувати з іншою бібліотекою (SpaCy або NLTK).

Продовження таблиці 2.1

Назва бібліотеки	Переваги	Недоліки
SparkNLP	-повний NLP конвеєр; -оптимізована для нейронних мереж; -має вбудовану перевірку орфографії; -розширена функціональним детектором почуттів; -масштабована.	-надто швидка; -слабка підтримка ком'юніті.

NLTK теж потужна бібліотека, яка може використовуватися для простих процесів NLP або як конвеєр NLP, але, як правило, має кращу криву навчання. Gensim підходить для неконтрольованої кластеризації та векторизації NLP [1].

2.9 Порівняння алгоритмів класифікації

Метрики які використовуются у данному дослідженні[10] P – точність у межах класу – це частка документів, що дійсно належать даному класу, щодо всіх документів, віднесених до цього класу. R – повнота частка знайдених класифікатором документів, що належать класу, щодо всіх документів цього у вибірці. Оскільки задача багатокласова з незбалансованими класами, то будуть використовуватися дві метрики: F1-macro (далі позначається Fm) – обчислює F1-міру для кожного класу, а потім повертає середнє арифметичне за всіма класами, F1-weighted (далі позначається Fw) – обчислює F1 міру для кожного класу, а потім повертає середнє арифметичне за всіма класами з урахуванням частки об'єктів кожного класу в наборі даних. Різниця між цими метриками показуватиме, наскільки добре розпізнаються класи з невеликою кількістю об'єктів.

Вихідний текст перетворюється у «мішок слів» - спрощене представлення тексту, яке показує, які слова зустрілися в тексті, але при цьому не враховує їх порядок. Таке представлення легко запрограмувати, воно зручно для використання в задачах автоматичної обробки тексту. Незважаючи на свою простоту, воно виявляється достатньо корисним і дозволяє успішно вирішити такі

завдання як класифікація тексту, тобто віднесення тексту до визначеної групи/категорії.

Далі, створюючи TF-IDF матриці з трьох варіантів моделей подання тексту та видаляючи стоп-слова на основі списку стоп-слів бібліотеки scikit-learn, перевіряються результати найбільш швидких класифікаторах при використанні п'ятикратної перехресної перевірки зі збереженням розподілу за класами . Наведено результати класифікації (таблиця 2.2).

Додаткові результати показують, що при аналізі методів з відбору ознак на різних класифікаторах, найкращі показники дає REF для кінцевої матриці TF-IDF (таблиця 2.3 – 2.5).

Таблиця 2.2 – Класифікація різних типів текстового опису

Модели представления текста	Logistic Regression		SVC		KNN	
	F_m	F_w	F_m	F_w	F_m	F_w
TF-IDF	0.4998	0.8032	0.7462	0.8978	0.5823	0.7314
TF-IDF з лематизацією	0.5016	0.8050	0.7462	0.8989	0.5859	0.7376
TF-IDF зі стемінгом	0.5025	0.8061	0.7476	0.8974	0.5870	0.7362

Таблиця 2.3 – Результати класифікації з використанням частотного фільтра

Типы текстового описания	SVC	
	F_m	F_w
TF-IDF	0.7454	0.8968
TF-IDF з лематизацією	0.7437	0.8961
TF-IDF зі стемінгом	0.7466	0.8971

Таблиця 2.4 – Результати класифікації з використанням частотного фільтра

Методи відбору ознак	TF-IDF со стемінгом + SVC	
	F_m	F_w
Частотний фільтр	0.7466	0.8971
VarianceThreshold	0.7449	0.8966
χ^2	0.7445	0.8958
RFE	0.7476	0.8979

Таблиця 2.5 – Результати класифікації кінцевої матриці TF-IDF

Методи классифікації	TF-IDF зі стемінгом + RFE	
	F_m	F_w
LogisticRegression	0.5039	0.8056
SVC	0.7476	0.8979
KNN	0.5521	0.7227
DecisionTreeClassifier	0.6209	0.8177
RandomForestClassifier	0.6622	0.8447
AdaBoostClassifier	0.6636	0.8394
MLPClassifier	0.8146	0.9149

Таблиця 2.6 - Результати класифікації векторних уявлень текстів за допомогою GloVe

Типи текстового опису	SVC		MLPClassifier	
	F_m	F_w	F_m	F_w
Текст без змін	0.6910	0.8519	0.6866	0.8442
Текст зі стемінгом	0.6823	0.8420	0.6778	0.8335
Текст з лематизацією	0.6848	0.8488	0.6802	0.8424

Для завдання, що розглядається, використовуємо навчену на текстовому корпусі Вікіпедії модель з 400000 слів з векторами розміру 300. Отримаємо вектори для слів у трьох використовуваних типах текстового опису та середнім їх, щоб отримати векторне подання документів. Перевіримо дані на SVC і MLPClassifier, що виходять, як на найбільш добре працюючих класифікаторах в умовах даної задачі. Наведено результати класифікації (таблиця 2.6) за допомогою п'ятикратної перехресної перевірки із збереженням розподілу за класами на зазначених класифікаторах.

2.10 Аналіз нейро-мережевих моделей для класифікації тексту

2.10.1 BERT

У даній статті [11] представлено BERT та її детальну реалізацію. У даній моделі є два кроки: попереднє навчання та тонке налаштування. Під час попереднього навчання модель навчається на немаркованих даних у різних завданнях попереднього навчання. Для точного налаштування модель BERT спочатку ініціалізується попередньо підготовленими параметрами, а потім усі параметри налаштовуються з використанням позначених даних із наступних завдань. Кожне наступне завдання має окремі точно налаштовані моделі, навіть якщо вони ініціалізуються з тими самими попередньо навченими параметрами. Приклад із відповідями на запитання на малюнку 1 слугуватиме поточним прикладом для цього розділу.

Відмінною рисою BERT є його уніфікована архітектура для різних завдань. Існує мінімальна різниця між попередньо навченою архітектурою та остаточною наступною архітектурою.

Переднавчаня BERT. Окрім вгадування слова в середині речення, під час навчання BERT додатково повинен «сказати», чи слідує друга пропозиція тренувального прикладу за першим. Кожен тренувальний приклад складається з двох пропозицій із пропущеними словами, а завдання, яке ставиться BERT —

вгадати, які слова пропущені (видати їх числовий код) і сказати, чи підходить друга пропозиція до першої. Для цього до початкового ембеддингу слова додаються сегментний та позиційний ембеддінг (рисунок 2.5).

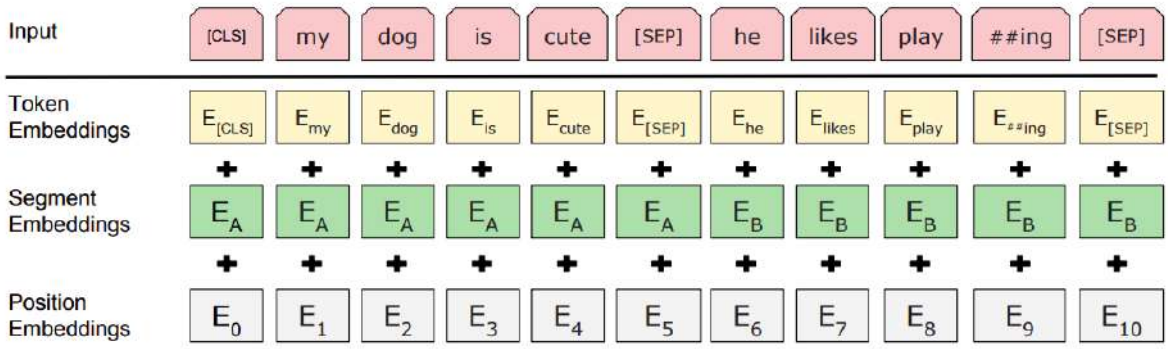

Рисунок 2.5 – Вхідне представлення BERT. Вбудовані вхідні дані є сумою вбудованих маркерів, вбудованих сегментів і вбудованих позицій

Жовтий початковий ембединг слова – число, ID токена з системи WordPiece. У словнику WordPiece, який вибрали творці BERTa, 30 тисяч токенів. Серед них – літери, найпопулярніші слова англійської та окремі склади.

До кожного початкового ембеддингу додається ще число, що означає, що слово стоїть у першому чи другому реченні (segment embedding), і навіть третє позиційний ембеддинг, показує, наскільки далеко два слова стоять друг від друга. Остання потрібна тому, що трансформер (що лежить в основі BERT) приймає весь тренувальний приклад цілком і працює зі словами паралельно. Для моделі немає "минулого контексту". Він буває в рекурентній нейромережі, де наступний крок не можна зробити без попереднього, але трансформер – не рекурентна архітектура. BERT потрібен позиційний вектор – набір із «швидких» та «повільних» синусоїд (рисунок 2.6), які змінюють своє значення з кожним токеном. Якщо у двох слів різні значення швидких графіків, але однакові – повільні, отже, вони поряд. Якщо різні значення мають повільні графіки, значить, слова стоять далеко один від одного.

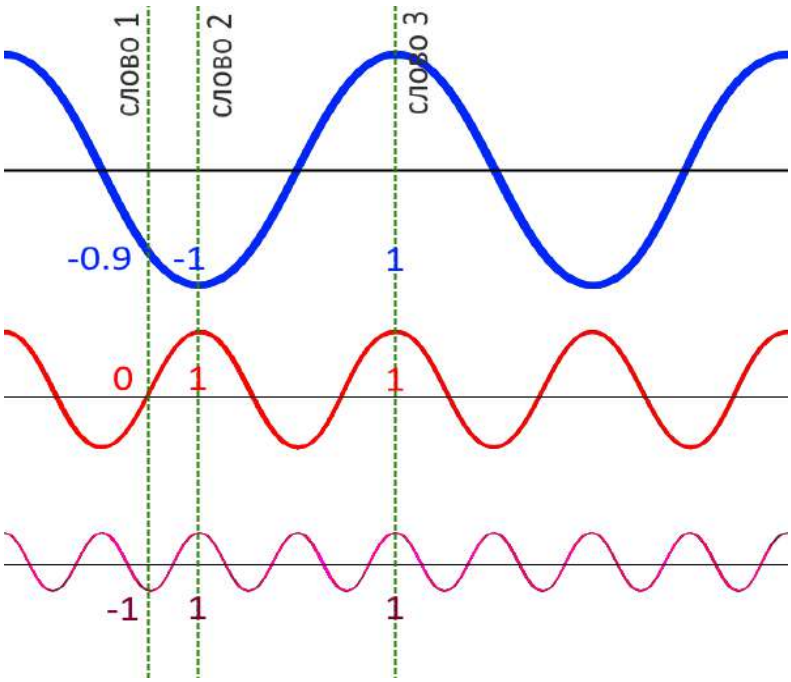

Рисунок 2.6 – Зображення швидких та повільних графіків для визначення слова

Перед реченнями стоїть токен [CLS] — «класифікуючий токен»: передсказуючи його, нейромережа вгадує, чи зв’язаны між собою речення з прикладу (що є визначення їх у класі «зв’язаних» або «не стосується»). Самі речення розділяються токеном [SEP], і всередині них деякі слова замінюються токеном [MASK] — нейросеть повинна передбачити саме ймовірне слово на цьому місці

Маскована мовна модель — компроміс: вона не повинна маскувати надто багато слів (якщо все замаскувати, не залишиться контексту, на якому можна вчитися) або замало (тоді потрібно показувати забагато тренувальних прикладів). Зазвичай вирішують замінювати 15% слів маскою, ці слова вибираються випадково.

Ідея донавчання, або тонкої настройки, або донавчання нейромережі будується на тому, що «базові» знання (тобто матриці ваг), отримані на вирішенні загальних завдань, допомагають краще вирішувати нові, вужчі завдання. Цей принцип - трансферне навчання - сильно просунув NLP: про те, як до нього прийшли і чому він повинен працювати, Системний Блок

розповість в окремій статті.

Для донавчання у BERT не потрібно прати «частину пам'яті» моделі. Натомість «поверх» моделі додають новий шар нейронів. Його матриці ваги заповнені випадковими числами, і потрібно налаштувати їх так, щоб на новому завданні помилка була мінімальною. При цьому всі попередні верстви вже натреновані на навчальному завданні.

Нові тренувальні дані подаються у старому форматі: це можливо завдяки гнучкості трансформерної архітектури.

Настає етап донавчання. Простий приклад нового завдання – зрозуміти, як пов'язані два речення: суперечать один одному, підтверджують чи нейтральні.

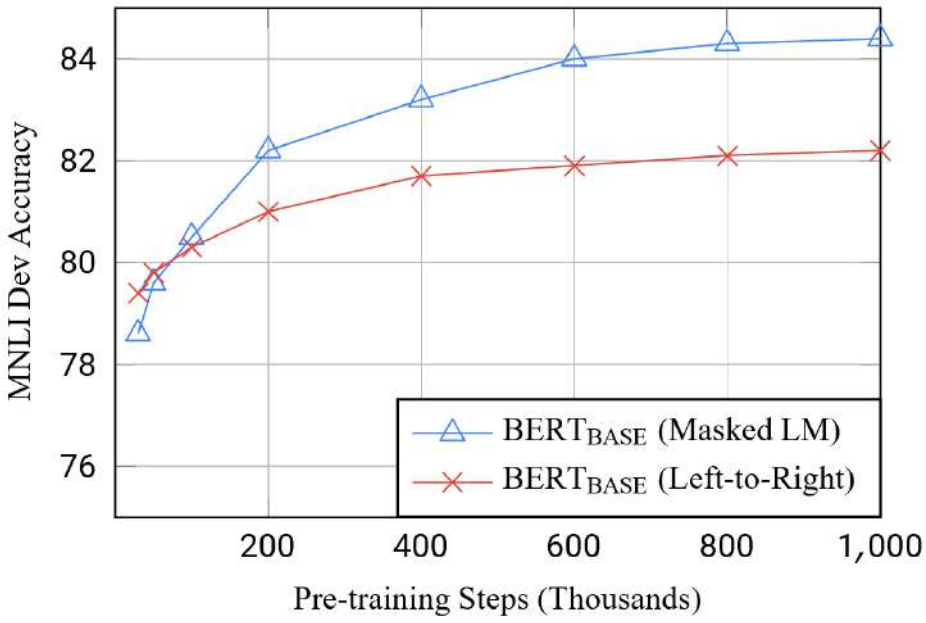

Рисунок 2.7 – Абляція за кількість кроків навчання

Новий шар нейронів має той самий формат, що старі: у ньому є токен C (класифицирующий) – пророкуючи його, модель передбачає, як пов'язані між собою речення (визначає в клас суперечащих, взаємно підтверджуапльних чи нейтральних). Якщо сильно спростити, нейромережа ніби дивиться на своє передбачення про те, чи йдуть ці два речення один за одним (про нього

говорить токен [CLS] з етапу навчання) і на ембеддінг слів. Вся ця інформація підказує, як потрібно класифікувати речення у новому завданні.

З тим що BERT використовує змішану стратегію для маскування цільових токенів під час попереднього навчання з ціллю маскованої мовної моделі (MLM). Нижче наведено (рисунок 2.7) дослідження абляції для оцінки ефекту різних стратегій маскування.

Це показує точність MNLI після тонкого налаштування, починаючи з параметрів моделі, які були попередньо навчені для k кроків. На осі абсцис – значення k.

2.11 Типи BERT

2.11.1 RoBERTa

RoBERTa – це варіант BERT, розроблений для покращення фази навчання. RoBERTa було розроблено шляхом тривалого навчання моделі BERT на більших даних із довшими послідовностями та великими міні-серіями. Дослідники RoBERTa отримали суттєво покращені результати за допомогою деяких модифікацій гіперпараметрів BERT. На відміну від BERT, RoBERTa використовує наступні методи для навчання Robust.

Динамічне маскування. У базовій попередній обробці BERT вхідні послідовності маскуються статично. RoBERTa застосовує техніку динамічного маскування. Вхідні послідовності множаться, щоб збільшити кількість послідовностей, а 15% випадково маскуються. Це дозволяє моделі зчитувати кілька різних шаблонів маскування в одній послідовності і в той же час зменшує потребу в значному збільшенні кількості випадків навчання.

Тренування без передбачення наступного речення. Модель RoBERTa використовує повні речення як вхідні дані. Ці вхідні речення безперервно вибираються з даних із маркером-роздільником, застосованим для позначення меж документа. Дослідник експериментував, видаляючи втрату передбачення

наступного речення, і виявив, що усунення NSP дещо покращує продуктивність подальших завдань.

Навчання на великих міні-серіях. Навчаючи модель у великих міні-серіях, дослідники виявили кращі результати. Збільшення розміру партії та навчальних даних принесло значні покращення. Оригінальний BERT навчається з розміром пакету лише 256 послідовностей для 1 мільйона кроків навчання. RoBERTa навчається 125 тис. кроків із використанням пакету з 2 тис. послідовностей [12].

2.11.2 ALBERT

ALBERT, відомий як «спрощена версія BERT», нещодавно був запропонований для покращення навчання та покращення результатів архітектури BERT шляхом використання методів спільного використання параметрів і факторизації. Модель BERT містить мільйони параметрів, BERT-модель містить близько 110 мільйонів параметрів, що ускладнює навчання, а також занадто багато параметрів впливають на обчислення. Для подолання таких проблем було введено ALBERT, оскільки він має менше параметрів порівняно з BERT. ALBERT використовує дві техніки.

Спільне використання параметрів між рівнями ця техніка, яка використовується для зменшення кількості параметрів у BERT. BERT містить N шарів кодера, наприклад, BERT-base має 12 шарів кодера. Під час навчання параметри вивчаються на всіх рівнях кодера. Коли справа доходить до перехресного шару. Замість вивчення параметрів на всіх рівнях кодувальника. Параметр першого шару кодера використовується спільно з усіма іншими рівнями кодера. Розгортається лише перший шар кодера. Кожен кодер містить підрівні. Багатостороння увага та зворотний зв'язок. Ми вивчаємо параметри кодувальника 1 і ділимося ними з іншими рівнями кодувальника. Щоб досягти цього, ALBERT пропонує різні варіанти

Загальний доступ – усі параметри рівня кодера, включаючи всі підрівні (рівень уваги з кількома головками та рівень прямої передачі), є спільними для

всіх рівнів кодувальника моделі BERT.

Прямий зв’язок – лише параметри рівня прямого зв’язку спільно використовуються з підрівнями прямого зв’язку на всіх інших рівнях кодувальника моделі BERT (рисунок 2.8).

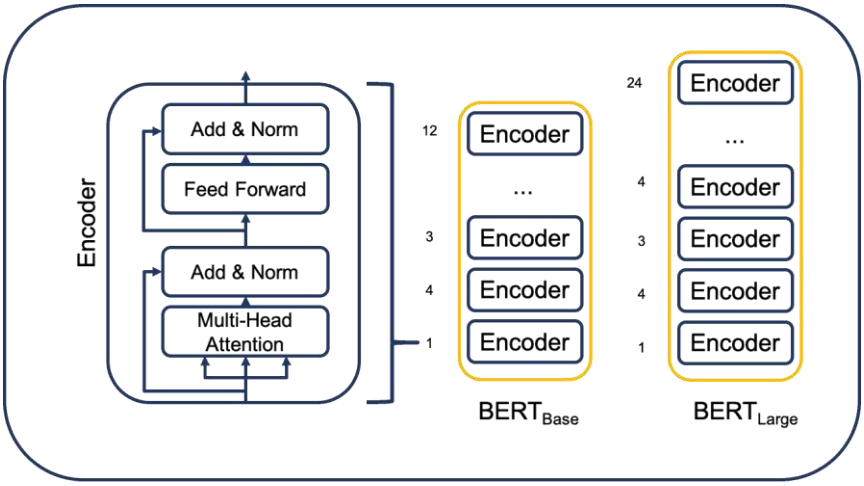

Рисунок 2.8 - Представлення двонаправленого кодувальника у моделі Transformers (BERT)

Спільна увага – лише параметри головного рівня уваги спільні з підрівнями головного уваги на всіх рівнях кодувальника моделі BERT.

За замовчуванням ALBERT використовує загальну техніку, де параметри підрівня прямого зв’язку та уваги з усіма рівнями кодувальника.

Параметризація рівня вбудовування на фактори. Ця також відоме як техніка зменшення. У BERT вкладення прихованого шару та вкладення вхідного шару мають однаковий розмір. У факторизованій параметризації шару дві матриці вбудовування розділені. Це пояснюється тим, що BERT використовує токенізатор word piece для створення токенів. Токени Word piece є неконтекстуальними, вони вивчаються з векторів одноразового кодування. Вбудовування прихованого шару потребує контекстно-залежного навчання. при збільшенні розміру вкладень прихованого шару «H» це також збільшить розмір вкладення частини слова «E». Щоб уникнути збільшення кількості параметрів,

співвідношення розмірів між прихованим шаром і шаром вбудовування розділено. Де ми розбиваємо матрицю вбудовування слів на менші матриці. Використовуючи їх, можна скоротити час навчання та час висновку моделі BERT, скорочується близько 70% загальної кількості параметрів. Як ми бачимо на ілюстрації нижче, модель ALBERT має менший розмір параметра порівняно з відповідною моделлю BERT [12].

2.13 LSTM

Модель LSTM є одним з типів RNN, який використовується для обробки та аналізу послідовних даних, таких як текст, мовлення або часові ряди. Вона була розроблена з метою вирішення проблеми зниклої градієнтної проблеми, що виникає у звичайних RNN.

У моделі LSTM використовуються спеціальні блоки пам'яті, які дозволяють зберігати та використовувати інформацію з попередніх моментів часу. Це досягається за допомогою трьох основних компонентів: "ворота забудови" (forget gate), "ворота входу" (input gate) і "ворота виходу" (output gate). Кожен з цих воріт контролює потік інформації у блок пам'яті та її використання.

Модель LSTM виявляється особливо ефективною при обробці послідовних даних з довготривалими залежностями. Вона широко використовується в таких задачах, як машинний переклад, розпізнавання мови, сентимент-аналіз тексту, генерація тексту та багато інших.

Конфлікт вхідної ваги: для простоти зосередимося на одній додатковій ваговій вазі wji. Припустимо, що загальну похибку можна зменшити, увімкнувши одиницю j у відповідь на певний вхід і утримуючи її активною протягом тривалого часу (поки це не допоможе обчислити бажаний вихід). За умови, що і не дорівнює нулю, оскільки одна і та ж вхідна вага повинна використовуватися як для збереження певних вхідних даних, так і для ігнорування інших, wji часто отримуватиме суперечливі сигнали оновлення

ваги протягом цього часу (j є лінійним): ці сигнали намагатимуться змусити wji брати участь у зберіганні вхідних даних (вмикаючи j) і захищаючи вхідні дані (запобігаючи перемиканню j через невідповідні наступні введення). Цей конфлікт ускладнює навчання та потребує більш залежного від контексту механізму керування «операціями запису» за допомогою вагових коефіцієнтів.

Конфлікт ваги виводу: припустимо, що j увімкнено та наразі зберігає деякі попередні введення. Для простоти зосередимося на одній додатковій вихідній вазі wkj. Той самий wkj має використовуватися як для отримання вмісту j у певний час, так і для запобігання j заважати k в інший час. Поки одиниця j не дорівнює нулю, wkj притягуватиме конфліктні сигнали оновлення ваги, згенеровані під час обробки послідовності: ці сигнали намагатимуться змусити wkj брати участь у доступі до інформації, що зберігається в j та в різний час захист блоку k від впливу j. Наприклад, у багатьох завданнях є певні «короткі помилки затримки часу», які можна зменшити на ранніх етапах навчання. Однак на пізніших етапах навчання j може раптово почати спричиняти помилки, яких можна уникнути в ситуаціях, які вже здавалися контрольованими при спробі взяти участь у зменшуючи більш довгі помилки затримки часу. Знову ж таки, цей конфлікт ускладнює навчання та вимагає більш залежного від контексту механізму для керування «операціями читання» за допомогою ваги виводу [13].

2.14 UNILIM

Подібно до BERT, попередньо навчений UNILM можна налаштувати (з додатковими рівнями, що стосуються конкретних завдань, якщо необхідно), щоб адаптуватися до різноманітних наступних завдань. Але на відміну від BERT, який використовується в основному для завдань NLU, UNILM можна налаштувати за допомогою різних масок самоуважності для агрегування контексту для різних типів мовних моделей, і, таким чином, можна використовувати як для завдань NLU, так і для завдань NLG.

Пропонований UNILM має три основні переваги. По-перше, уніфікована процедура попереднього навчання призводить до єдиного Transformer LM, який використовує спільні параметри та архітектуру для різних типів LM, усуваючи потребу в окремому навчанні та розміщенні кількох LM. По-друге, спільне використання параметрів робить представлення вивченого тексту більш загальними, оскільки вони спільно оптимізовані для різних цілей мовного моделювання, де контекст використовується різними способами, пом'якшуючи надмірне пристосування до будь-якого окремого завдання LM. По-третє, на додаток до його застосування до завдань NLU, використання UNILM як LM послідовності до послідовності робить його природним вибором для NLG, наприклад, для абстрактного підсумовування та генерації запитань.

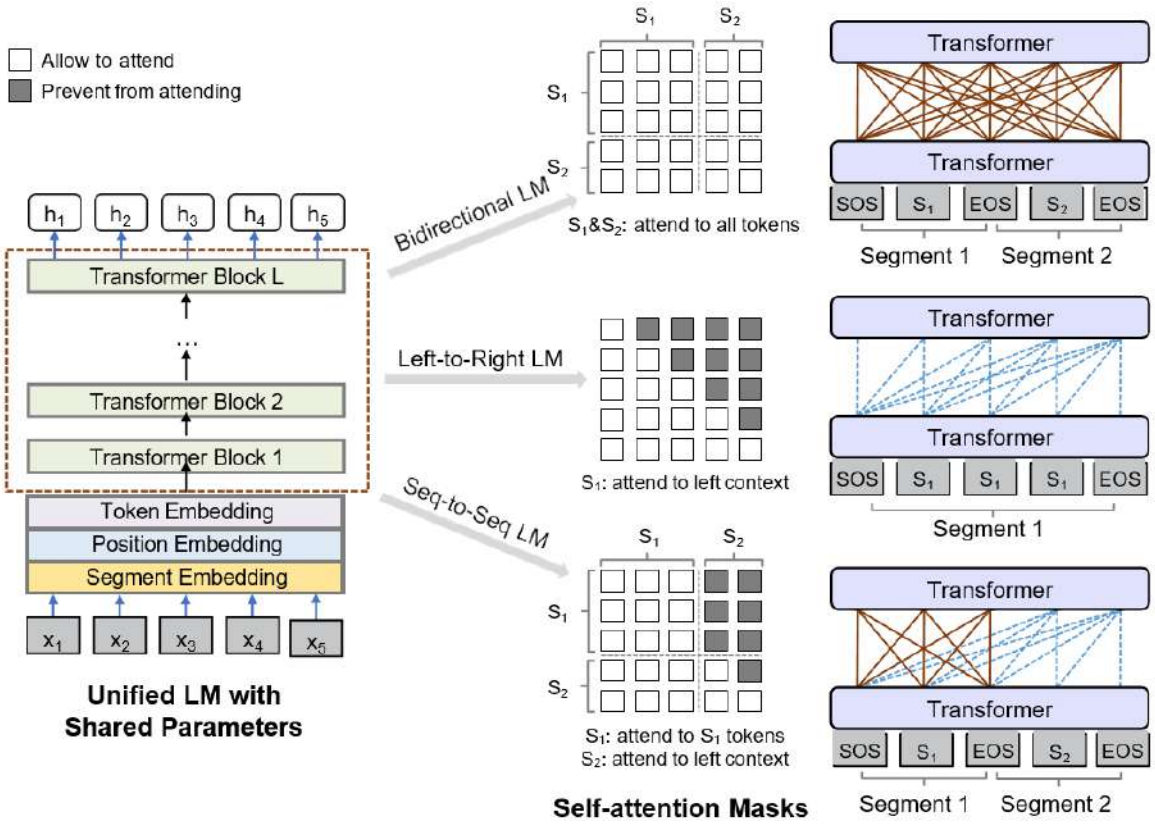

Рисунок 2.9 – Огляд єдиної попередньої підготовки LM

За вхідної послідовності $x = x_1 \ldots x_{|x|}$ UNILM отримує контекстне векторне представлення для кожного маркера. Як показано (Рисунок 2.9), попереднє

навчання оптимізує спільну мережу Transformer щодо кількох неконтрольованих цілей мовного моделювання, а саме односпрямованого LM, двонаправленого LM та послідовного LM. Щоб контролювати доступ до контексту лексеми слова, який потрібно передбачити, ми використовуємо різні маски для самоуважності. Іншими словами, ми використовуємо маскування, щоб контролювати, скільки контексту повинен враховувати маркер під час обчислення його контекстуалізованого представлення. Після попереднього навчання UNILM ми можемо налаштувати його, використовуючи дані про конкретні завдання для подальших завдань [14].

3 ЗАПРОПОНОВАНА МОДЕЛЬ КЛАСИФІКАЦІЇ ДОКУМЕНТІВ

3.1 Запропонована модель класифікації кваліфікаційних робіт на основі адаптивного оновлення ключових слів

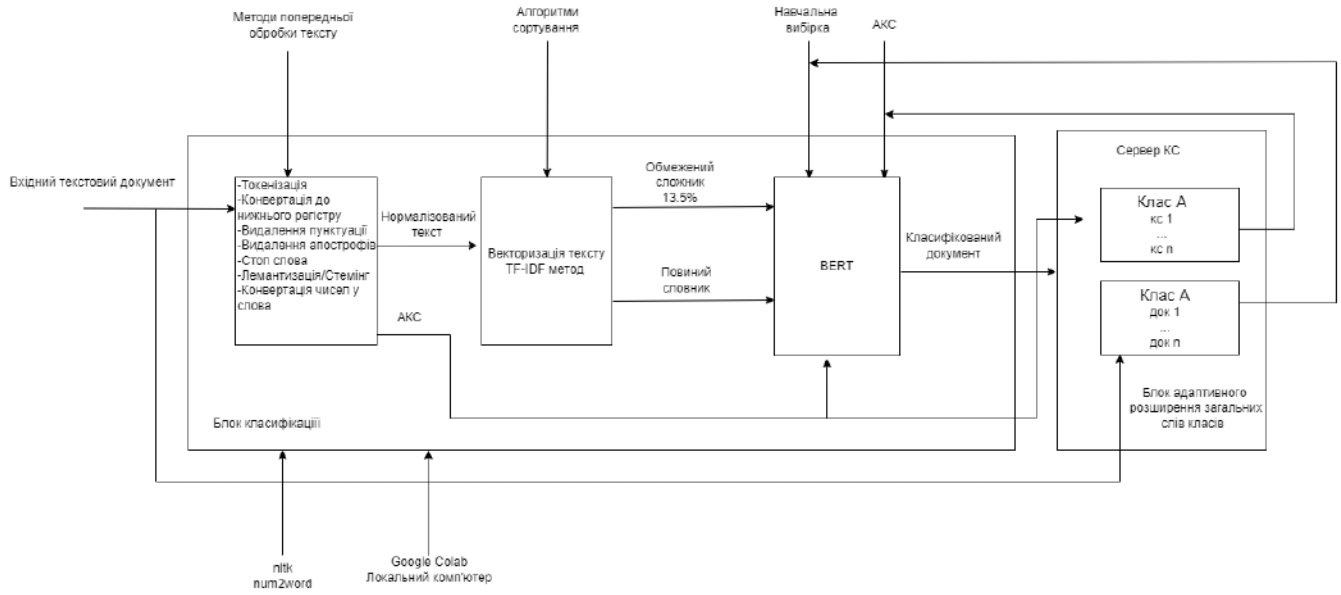

Рисунок 3.1 – Модель класифікації з адаптивним оновленням ключових слів

Запропонована модель класифікації (рисунок 3.1) працює наступним чином: перший етап попередня обробка також взяття АКС для класифікації документа також збереження їх у сервері ключових слів для подальшого динамічного навчання моделі. Обробка тексту яка містить наступні методи обробки такі як:

- токенізація. Розбиває необроблений текст на слова, речення, які називаються токенами. Ці токени допомагають зрозуміти контекст. Токенізація допомагає інтерпретувати значення тексту шляхом аналізу послідовності слів;

- Конвертація до нижнього регистру. Під час обробки тексту кожне речення розбивається на слова, і кожне слово після попередньої обробки розглядається як лексема. Мови програмування вважають текстові дані

наприклад «The» та «the» отже через кодуваня символів буде їх прирывнювати до одного токена. Перетворення на малі літери є обов’язковим етапом попередньої обробки.

- стоп-слова – це слова, які найчастіше зустрічаються, і не надають жодної додаткової цінності вектору документа.Їх видалення збільшить ефективність обчислень і простору. Бібліотека nltk має метод для завантаження стоп-слів.

- пунктуація - це набір непотрібних символів, які є в документах корпусу.

- видалення апострофів. В знаках пунктуації немає апострофа. Тому що коли спочатку видаляються розділові знаки, «don`t» на «dont», і це стоп-слово не буде видалено. Натомість ми видалимо стоп-слова, а потім символи, а потім повторимо видалення стоп-слова, оскільки деякі слова можуть мати апостроф, але не є стоп-словами.

- стемінг. Наприклад, «playing» і «played» – це один і той самий тип слів, які в основному вказують на дію. Stemmer робить саме це, він зводить слово до його основи. Використовується бібліотека під назвою porter-stemmer, яка є стеммером на основі правил. Porter-Stemmer визначає та видаляє суфікс або афікс слова. Слова, надані стеммером, не обов’язково можуть мати декілька значень, але вони будуть ідентифіковані як один токен для моделі.

- перетворення чисел у слова коли зустрічається у реченні наприклад «100 dollars» або «hundred dollars». Але після данної обробки потрібно ще раз зробити видаленя слів та пунктуації так як бібліотека num2word знов додасть слова які нам не потрібні.

Другий етап це знаходження слів які найчастіше повторюються у документі та сортування їх для виділення діапазону в якому будуть знаходитись АКС. Після сортування формується обмежений список це 13.5% від загальної кількості слів в обробленому документі, а також повний список зі всіма обробленими словами.

Третій етап це класифікація документа за допомогою АКС, обмеженого

словника та повного словника. На визначення класу впливає коефіцієнт який буде прив’язаний до словника. Виходячи з експериментів які покажуть який тип словника та з яким типом обробки дасть результат кращий результат віщий коефіцієнт. Документи які не продуть певний поріг будуть відкинуті.

Класифіковані документи йдуть до блока адаптивного розширення значущих слів класів і він йде як датасет для перенавчання моделі для більш точної класифікації документів. Перенавчання наступає у той момент коли досягается первне число вірно класифікований документів.

Початковій датасет на якому навчена модель це 20 newsgroups dataset. Цей набір даних містить близько 18 000 постів новин на 20 тем, розділених на дві підмножини: одна для навчання (або розробки), а інша для тестування (або для оцінки ефективності). Розподіл між поїздом і тестовим набором базується на повідомленнях, опублікованих до та після певної дати.

3.2 Формальна логіка виконання досліджень

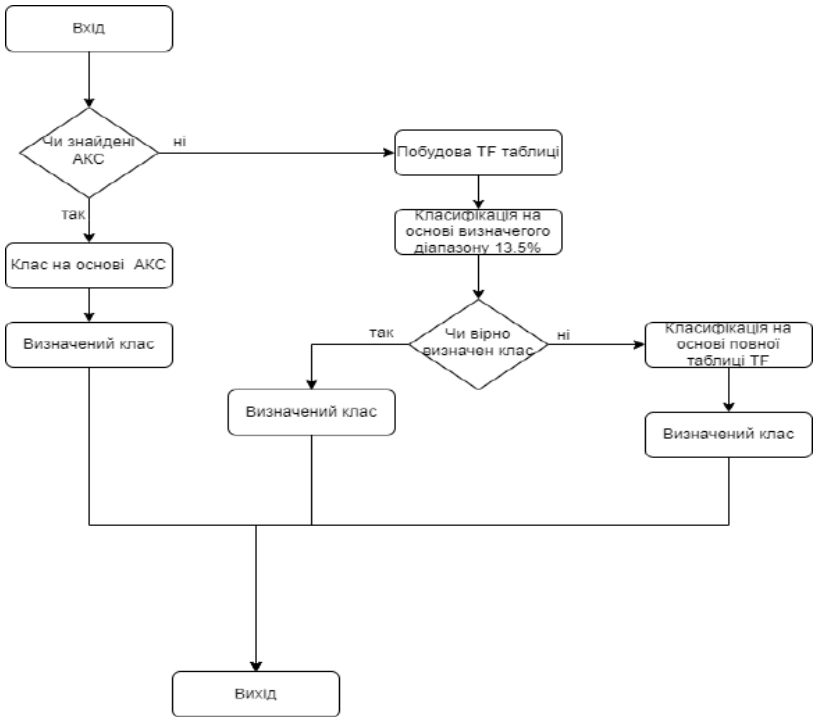

Рисунок 3.2 – Блок-схема формальної логіки досліджень

Дослідження проводилось наступним чином. На вже готову натреновану модель подається текст оброблений та не оброблений. Правильність класифікації перевірялась особисто та записувалось у таблицю і вираховувалось середнє (рисунок 3.2).

Обробка текстів проводилась за допомогою мови програмуваня Python. Далі вже на модель все подавалось в ручну на модель на Google Colab.

3.3 Визначення діапазону КС у частотному словнику

Запропонований підхід визначення діапазону значущих слів частотного словника для досягнення поставленої мети та рішення визначених задач (рисунок 3.3).

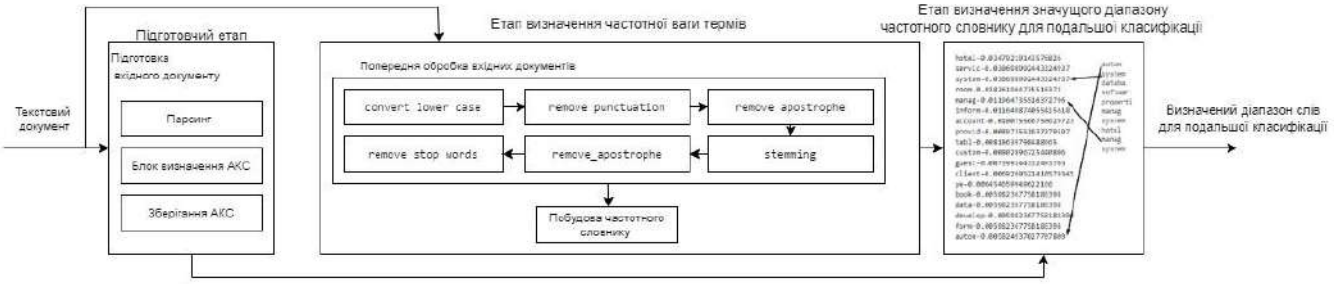

Рисунок 3.3 - Запропонований підхід визначення діапазону значущих слів частотного словника

Запропонована послідовність дій включає три ключові етапи – підготовчий етап, етап визначення частотної ваги термів, етап визначення значущого діапазону частотного словнику для подальшої класифікації.

Розглянемо кожен із етапів окремо. Підготовчий етап включає попередню підготовку вхідного документу, визначення та зберігання АКС. Задачею другого етапу є побудова частотного словнику після виконання препроцесінгу текстового документу, а саме виконання кроків – конвертація до єдиного регістру, видалення знаків пунктуації, стемінг отриманих термів, видалення стоп-слів. Підготовлений набір термів є основою для побудови частотного

словнику (метод TF). Вхідними даними третього етапу є побудований частотний словник та визначені АКС. Задачею третього етапу є визначення значущого діапазону шляхом знаходження усіх АКС у частотному словнику.

Результати роботи запропонованого підходу для трьох різних вхідних документів зображені на (рисунок 3.4а - 3.4в).

```
Author`s key words  ['autom', 'system', 'databa', 'softwar', 'properti', 'manag', 'system', 'hotel', 'manag', 'system']
0 -  hotel -  0.03479219143576826
2 -  system -  0.030698992443324937
4 -  manag -  0.011964735516372796
17 -  autom -  0.005824937027707809
29 -  databa -  0.004408060453400504
50 -  softwar -  0.0033060453400503777
136 -  properti -  0.0015743073047858943
```

а)

```
Author`s key words  ['chatbot', 'voic', 'assist', 'voic', 'command', 'request', 'analysi', 'system']
18 -  voic -  0.005787781350482315
30 -  chatbot -  0.004777216352779054
37 -  request -  0.004042259990813046
45 -  system -  0.0036747818098300414
59 -  command -  0.0029398254478640333
68 -  assist -  0.002572347266881029
187 -  analysi -  0.0011024345429490124
```

б)

```
Author`s key words  ['imag', 'speedup', 'perform', 'time', 'cpu', 'gpu', 'softwar', 'model', 'hardwar', 'model']
0 -  imag -  0.031742738589211617
8 -  gpu -  0.009543568464730291
9 -  time -  0.009128630705394191
15 -  cpu -  0.007468879668049793
18 -  perform -  0.005809128630705394
33 -  hardwar -  0.004149377593360996
93 -  model -  0.0018672199170124482
98 -  softwar -  0.0018672199170124482
234 -  speedup -  0.0010373443983402491
```

в)

Рисунок 3.4 – Результати роботи запропонованого підходу для визначення діапазону значущих слів частотного словнику на основі визначених АКС

Як видно з наведених результатів роботи застосунку, частковий словник впорядкований у порядку зменшення вживаності слів. У проведені експерименту було використано три документи (документ 1, документ 2, документ 3) англійською мовою [1].

4 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ

4.1 Результати експериментів з визначенням діапазону КС

Визначення діапазону та відсотку значущих слів (таблиця 4.1) дозволить виконання подальшої класифікації наукових та дослідницьких робіт при формуванні тематичних каталогів навіть у випадку відсутності переліку авторських ключових слів, які можна використовувати для класифікації [1].

Таблиця 4.1 – Отримані результати

	Кількість слів до препроцесінгу, bt	Кількість слів після препроцесінгу, at	Кількість АКС, n	Діапазон АКС у частковому словнику	Рекомендований відсоток значущих слів частотного словнику для подальшої класифікації	Рекомендований діапазон значущих слів частотного словнику для подальшої класифікації
Документ 1	10154	1296	10	1-137	13,47	1-235
Документ 2	8602	1426	8	19-188		
Документ 3	7341	1302	10	1-235		

Розрахунок рекомендованого відсотку значущих слів для кожного

документа, у якому містяться АКС, виконувався за формулою нижче:

$$C = \frac{bt}{at}, \quad (4.1)$$

Де – bt - кількість слів вхідного документу до препроцесінгу;
at - кількість слів вхідного документу після препроцесінгу.

4.2 Вплив різних методів препроцесінгу на результати класифікаціїї

Визначення впливу на класифікацію вимірювалось за допомогою 40 кваліфікаційних робіт з кафедри електронно обчислювальних машин (таблиця 4.2) та кафедри економічної кібернетики (таблиця 4.3). Виконувалась оцінка впливу методів нормалізації тексту, які були використані на етапі, який передував безпосередній обробці, а саме – стеммінг та лематизація.

Таблиця 4.2 – Оцінка правильності класифікації робіт з ЕОМ

Повний не оброблений текст	Повний текст оброблений за допомогою лематизації	Повний текст оброблений за допомогою стеммінгу
82,21%	86,84%	89,47%

В ході тестування було звернуто увагу на те що коли модель навчена тільки на 20 newsgroups dataset тема більш конкретна то класифікація більш точніша.

Таблиця 4.3 – Оцінка правильності класифікації робіт з ЕК

Повний не оброблений текст	Повний текст оброблений за допомогою лематизації	Повний текст оброблений за допомогою стеммінгу
72,47%	70,59%	75,41%

Такі теми як обробка фото або робота з графікою залом класифікуються краще або теми пов’язані з мережами. Такі теми які пов’язані з розробкою S-bot або розробка сайту для магазину або інформаційна система класифікувались некоректно. Такі самі зауваження можна віднести і до кваліфікаційних робіт з економіки.

І, як видно з результатів, економіка на даному датасеті класифікується гірше ніж роботи з ЕОМ.

4.3 Вплив вибору значущих слів на результат класифікації

В ході даного тестування використовувались ті самі і та сама кількість дипломів як і в попередньому тесті. Результати впливу значущих слів приведені у відпоідних таблицяз для ЕОМ (таблиця 4.4) та для ЕК (таблиця 4.5).

Таблиця 4.4 – Оцінка правильності класифікації у визначеному діапазоні значущих слів для ЕОМ

13.5% слів не оброблених	13.5% слів оброблених за допомогою стеммінгу	13.5% слів оброблених за допомогою лематизації	Авторські ключові слова
57,89%	65,79%	52,63%	81,58%

Таблиця 4.5 – Оцінка правильності класифікації у визначеному діапазоні значущих слів для ЕК

13.5% слів не оброблених	13.5% слів оброблених за допомогою стеммінгу	13.5% слів оброблених за допомогою лематизації	Авторські ключові слова
41,48%	58,82%	41,18%	52,94%

4.4 Аналіз результатів

Виходячи з загальних результатів тестів, кращим методом нормалізації тексту, який можна буде використовувати у моделі класифікації для кожної із протестованих тем, є стемінг. Це можна пояснити тим, що, не дивлячись на те, що лематизація зазвичай вважається більш інформативною, ніж просте стемінг, для технічних текстів, характерною особливістю яких є написання у теперішньому часі та відносно незначна кількість колокацій, ідіом та омонімів, лематизація є надлишковоя як за витрачений часом на обробку тексту, так я впливає на яеість класифікації. Тобто те, що у деяких випадках робить стемінг гіршим за лематизацію (а саме – стемінг на відміну від лематизатора працює із окремии словами, ви витрачаючи час на сканування усього корпусу та визначення частин мови), у даному експерименті не є суттєвим.

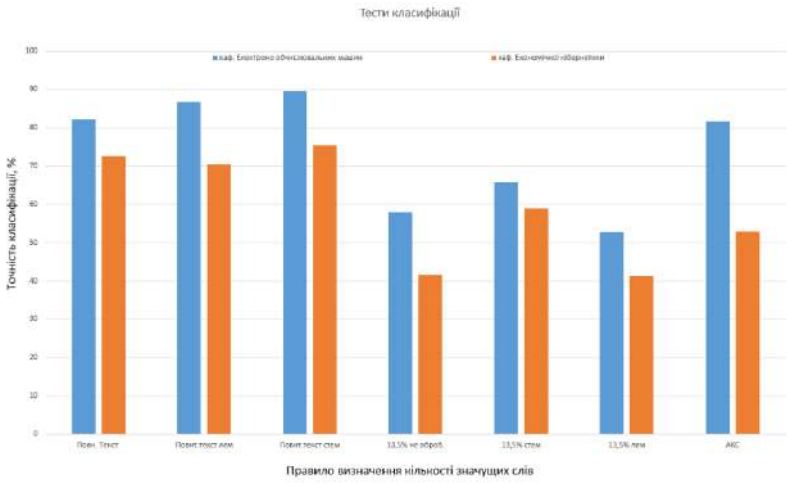

Рисунок 4.1 – Результати класифікації

Експерименти, які проводилися для аналізу впливу кількості значущих слів для класифікації показують, що використання АКС для класифікації є гарним показником, оскільки забезпечують найшвидшу робботу класифікатора (100мс проти 500мс для повного словнику), при цьому уступаючи у точності лише у 0,63% у порівнянні із повним словником.

ВИСНОВКИ

В ході кваліфікаційної роботи було проведено аналіз предметної області. Приведені приклади використання класифікаторів у різних галузях і були порівняні методи класифікації.

Було запропоновано підхід для визначення значущих слів документу для подальшого їх використання у процесі класифікації в якості вектора ознак. В ході виконання роботи було визначено авторські ключові слова, побудовано частковий словник та проаналізована кореляція між АКС та переліком впорядкованих слів частотного словнику на основі методу TF, до якого також входять авторські ключові слова. Визначення діапазону та відсотку значущих слів дозволяє виконувати подальшу класифікацію наукових та дослідницьких робіт при формуванні тематичних каталогів навіть у випадку відсутності переліку авторських ключових слів, які можна використовувати для класифікації.

Також в ході кваліфікаційної роботи були проведені експерементальні дослідженя для виявлення впливу лематизації, стемінгу та кількості значущих слів на точність класифікації. По отриманим результатам класифікація по АКС не уступає класифікації за допомогою повного словника.

з подальшим перенавчанням моделі для більш точної класифікації документів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Olesia BARKOVSKA, Dmytro MOHYLEVSKYI, Yuliia IVANENKO, Dmytro ROSINSKIY “WAYS TO DETERMINE THE RANGE OF KEYWORDS IN A FREQUENCY DICTIONARY FOR TEXT CLASSIFICATION” , International scientific journal “Computer Systems and Information Technologies”, 2023, No 1, pp.14-20.

2. Olesia Barkovska, Vladyslav Kholiev, Anton Havrashenko, Dmytro Mohylevskyi, Andriy Kovalenko, A Conceptual Text Classification Model Based on Two-Factor Selection of Significant Words. Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems. Volume II: Computational Linguistics Workshop, Kharkiv, Ukraine, April 20-21, 2023, 244-255.

3. Li, Qian, et al. "A survey on text classification: From traditional to deep learning." ACM Transactions on Intelligent Systems and Technology (TIST) 13.2 (2022): 1-41.

4. Understanding TF-ID: A Simple Introduction [Електронний ресурс]. Режим доступу: https://monkeylearn.com/blog/what-is-tf-idf/

5. Jaiswal M. et al. “Detecting spam e-mails using stop word TF-IDF and stemming algorithm with Naïve Bayes classifier on the multicore GPU” //International Journal of Electrical & Computer Engineering (2088-8708). – 2021. – Т. 11. – №. 4.

6. Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

7. Gasparetto, Andrea, et al. "A survey on text classification algorithms: From text to predictions." Information 13.2 (2022): 83.

8. Cambria E., White B. “Jumping NLP curves: A review of natural language processing research” //IEEE Computational intelligence magazine. – 2014. – Т. 9. – №. 2. – С. 48-57.

9. Zhou M. et al. “Progress in neural NLP: modeling, learning, and reasoning” //Engineering. – 2020. – Т. 6. – №. 3. – С. 275-290.

10. Лаптев, М. В. "Анализ методов машинного обучения на примере задачи многоклассовой классификации текста." Информатика: проблемы, методы, технологии. 2022.

11. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018). doi:10.48550/arXiv.1810.04805

12. BERT Variants and their Differences [Електронний ресурс]. Режим доступу: https://360digitmg.com/blog/bert-variants-and-their-differences

13. Hochreiter, JS Sepp, and J. Schmidhuber. "Long short-term memory (Neural Computation 9 (8): 1735 {1780, 1997)." Fakultät für Informatik, Technische Universität München, Licence details: https://www. bioinf. jku. at/publications/older/2604. pdf (1997).

14. Dong, Li, et al. "Unified language model pre-training for natural language understanding and generation." Advances in neural information processing systems 32 (2019).