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 нейромережевої системи для категоризації текстових документів

(тема)

Виконав:

студент II курсу, групи СПм-22-5

Рибалов О.О.
(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: проф. Фесенко Т.Г.
(посада, прізвище, ініціали)

Допускається до захисту

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

2024 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 Рибалову Олександру Олеговичу
(прізвище, ім’я, по батькові)

1. Тема роботи Модель нейромережевої системи для категоризації текстових документів

затверджена наказом по університету від “ 01 ” квітня 2024 р. № 257 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2024 р.

3. Вхідні дані до роботи

1) документація мови програмування Python 3;
2) операційна система macOS Sonoma.

4. Перелік питань, що потрібно опрацювати у роботі

1) аналіз предметної області;
2) аналіз існуючих досліджень;
3) аналіз методів прогнозування часових рядів;
4) програмна реалізація;
5) аналіз результатів;
6) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайд-презентація – 16 слайдів

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	Аналіз проблеми та огляд існуючих рішень	02.04.24-08.04.24	
2	Вибір методики дослідження	09.04.24-16.04.24	
3	Вибір інструментальних засобів	17.04.24-22.04.24	
4	Розробка алгоритмічного забезпечення	23.04.24-06.05.24	
5	Проведення експериментів	07.05.24-23.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	24.05.24-03.06.24	
7	Подання кваліфікаційної роботи керівникові	04.06.24-07.06.24	
	та її попередній захист		
8	Подання кваліфікаційної роботи на	08.06.24-12.06.24	
	рецензування		

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Фесенко Т.Г.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 61 с., 6 рис., 1 табл., 2 дод., 50 джерел.

МОДЕЛЬ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ, КАТЕГОРИЗАЦІЯ ТЕКСТОВИХ ДОКУМЕНТІВ, BERT, DISTILBERT, ШТУЧНИЙ ІНТЕЛЕКТ, ОБРОБКА ПРИРОДНОЇ МОВИ, NLP.

Метою кваліфікаційної роботи є розробка та дослідження моделі нейромережевої системи для категоризації текстових документів. У ході виконання кваліфікаційної роботи.

У ході виконання роботи проведено аналіз існуючих методів розпізнавання текстової інформації, описано архітектуру системи на основі моделі DistilBERT, реалізовано програмну частину та проведено експериментальне дослідження ефективності моделі. Результати дослідження підтверджують високу точність та ефективність розробленої системи.

ABSTRACT

Master’s thesis: 61 pages, 6 figures, 1 tables, 2 appendices, 50 sources.

NEURAL NETWORK MODEL, TEXT DOCUMENT CATEGORIZATION, BERT, DISTILBERT, ARTIFICIAL INTELLIGENCE, NATURAL LANGUAGE PROCESSING, NLP.

The aim of the qualification paper is to develop and study a neural network system model for categorizing text documents.

The research includes an analysis of existing methods for text information recognition, a description of the system architecture based on the DistilBERT model, implementation of the software part, and an experimental study of the model's effectiveness. The research results confirm the high accuracy and efficiency of the developed system.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АРМ – автоматизоване робоче місце

ІАЦ – інформаційно-аналітичний центр

WDS – бездротова розподільча система (англ., Wireless Distribution System)

ВСТУП

У сучасному світі обробка великих масивів текстової інформації вимагає від нас використання технологій штучного інтелекту, які, в свою чергу, повинні мати здатність швидко та точно категоризувати дані. Особливо це стосується сфери юриспруденції, журналістики, криміналістики [10], політології, соціології та інших наук, де точність класифікації документів може мати вирішальні наслідки для результату роботи. Відповідно, великий інтерес викликає розробка нейромережевих систем, спроможних автоматизувати процес віднесення текстів до відповідних категорій.

Збільшення обсягів інформації, що циркулює в цифровому просторі, створює значні виклики для традиційних методів обробки даних. Традиційні алгоритми часто не в змозі ефективно справлятися з такими масивами інформації, що веде до потреби в застосуванні більш передових технік, таких як машинне навчання та глибинне навчання. Нейромережі, зокрема, виявилися надзвичайно ефективними в розпізнаванні, аналізі та категоризації текстових даних, завдяки своїй здатності вчитися на великих наборах даних і вирішувати задачі, які були б надмірними для людського аналітика [31].

Однією з ключових переваг використання нейромереж у обробці текстових даних є можливість вивчення та адаптації до контексту і семантики мови, що дозволяє досягати високої точності у визначенні тематики та інтенцій тексту. Це особливо важливо в таких областях, як автоматизоване модерування контенту, аналіз настроїв, юридичний аналіз [47] та багатьох інших, де важливо не тільки розпізнати конкретні слова, але й зрозуміти загальний контекст і значення тексту.

Розробка та дослідження нейромережевої системи використовує модель DistilBERT для категоризації текстових документів [50]. Модель DistilBERT була обрана з огляду на її високу ефективність та порівняно невеликі вимоги до обчислювальних ресурсів, що робить її ідеальною для задач, де необхідно

збалансувати між точністю виконання та обмеженнями на потужність апаратного забезпечення [15]. Така оптимізація є ключовою для розробки доступних та ефективних систем обробки мови, здатних широко застосовуватися в різноманітних галузях та індустріях.

Представлено опис архітектури системи, викладено методологію дослідження, а також продемонстровано програмну реалізацію обраного методу. Результати експериментів підтверджують ефективність запропонованої системи у задачі класифікації текстів, відкриваючи нові можливості для автоматизації обробки текстових даних і підвищення якості та швидкості аналітичних робіт.

Метою роботи є не тільки продемонструвати практичну цінність нейромережевих технологій у сфері обробки текстових даних але й зробити внесок у подальше розвиток цього напрямку досліджень, підкресливши важливість інтеграції штучного інтелекту в повсякденні задачі обробки інформації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність завдання категорізації текстових документів

Сучасний інформаційний простір перенасичений даними, що стрімко зростають у обсягах. Ця тенденція особливо виразна в юридичній сфері, де велика кількість текстових документів вимагає швидкого і точного аналізу. Точність категоризації таких документів є критичною, адже помилки можуть призвести до правових непорозумінь та фінансових втрат. Відтак, актуальність розвитку та впровадження систем на основі штучного інтелекту, здатних автоматизувати процес класифікації документів, стає беззаперечною.

Важливість автоматизації впорядкування та аналізу юридичної документації не може бути переоцінена. Юридичні фахівці, органи державної влади та бізнес-структури зіштовхуються з викликами, що випливають із потреби ефективного управління масивами даних [3]. Компанії щорічно інвестують значні кошти у створення електронних документів для юридичних справ. Наразі обидві сторони юридичних процесів все частіше звертаються до методів машинного навчання, таких як класифікація тексту, для аналізу великої кількості даних та виявлення відповідних документів для своїх потреб [30]. Штучний інтелект і, зокрема, обробка природної мови (NLP), пропонують рішення, що кардинально змінюють підходи до обробки, пошуку та класифікації текстової інформації [32].

Автоматизація цих процесів за допомогою нейромережевих систем дозволяє суттєво збільшити продуктивність роботи фахівців, зменшити можливість людської помилки та забезпечити швидкий доступ до необхідної інформації. Використання моделей, як-от DistilBERT, які вже продемонстрували свою ефективність у багатьох NLP-задачах, надає цим системам здатність не просто визначати ключові слова, а й розуміти нюанси та контекст тексту, що є вирішальним для правильної категоризації документів

за різними темами [4]. З огляду на це, дослідження, спрямоване на створення та аналіз нейромережевої системи для категоризації текстових документів, є актуальним та своєчасним.

Реалізація такого проекту дозволяє, з одного боку, сприяти правовій точності та відповідальності, з іншого – стимулювати технологічний прогрес у галузі штучного інтелекту. Визнання потенціалу та обмежень існуючих систем є фундаментом для подальших інновацій та вдосконалень, що спрямовані на розробку все більш точних, надійних та ефективних методів класифікації документів.

1.2 Методи розпізнавання текстової інформації

Існують декілька основних методів основаних на штучних нейронних мережах (ШНМ) які використовуються при розпізнаванні та відповідно класифікації текстових документів. Тому важливо розуміти, як ці методи взаємодіють та доповнюють один одного в комплексному аналізі тексту [5].

Частотний аналіз та тематичне моделювання: Ці методи дозволяють виявити основні теми, що зустрічаються в тексті, за допомогою аналізу частоти слова та їх розподілу [22]. Хоча ці методи можуть бути реалізовані без ШНМ, сучасні підходи часто включають елементи машинного навчання для покращення точності та глибини аналізу [6].

Аналіз сентиментів за допомогою ШНМ: Цей метод використовується для визначення емоційного забарвлення тексту, тобто позитивного, негативного, або нейтрального ставлення автора до обговорюваної теми [23]. ШНМ, особливо ті, що включають архітектури глибокого навчання, здатні ефективно впоратися з цим завданням, враховуючи контекстуальні зв’язки між словами [25].

Сетевий аналіз для вивчення зв’язків: Використання ШНМ може значно покращити сетевий аналіз, дозволяючи виявити зв'язки та асоціації між

різними елементами тексту, такими як ключові слова, фрази, персонажі або концепції [27].

Машинне навчання для класифікації текстів: Класифікація текстів – це одна з найпоширеніших задач в NLP, де ШНМ використовуються для автоматичного визначення категорій текстових документів на основі їх змісту [21]. Глибоке навчання значно покращило можливості класифікації завдяки здатності моделювати складні залежності в даних [41].

Видобування інформації для точного ізоляції специфічних даних: Видобування інформації – це процес витягу структурованої інформації з неструктурованого тексту. ШНМ дозволяють автоматизувати цей процес, ефективно виявляючи іменовані сутності, відносини між ними, факти та події з тексту [43].

Кожен із цих методів має свої сильні та слабкі сторони і може бути застосований в залежності від конкретних цілей дослідження та особливостей даних. Зокрема, використання комбінації цих методів може значно покращити розуміння змісту текстових документів, дозволяючи не тільки класифікувати текст за певними критеріями, а й глибше аналізувати його семантику та структуру.

Для ефективної реалізації цих методів важливо правильно підібрати архітектуру нейронної мережі, яка буде адаптована до конкретної задачі. Наприклад, для аналізу сентиментів та класифікації текстів часто використовуються рекурентні нейронні мережі (RNN) [38] та їх розвинені варіанти, як-от LSTM (Long Short-Term Memory) [18] або GRU (Gated Recurrent Units) [14] , які здатні ефективно обробляти послідовні дані. З іншого боку, для задач тематичного моделювання та видобування інформації можуть бути застосовані конволюційні нейронні мережі (CNN), які показують хороші результати у роботі з текстовими даними великого обсягу [48].

Окрім того, значний вплив на розвиток методів NLP зробили трансформерні моделі, такі як BERT (Bidirectional Encoder Representations from Transformers) і GPT (Generative Pre-trained Transformer) [5]. Вони

демонструють високу ефективність у різноманітних задачах обробки тексту, зокрема у розумінні контексту, класифікації текстів, генерації мови та інших. Ці моделі працюють на основі механізму уваги, що дозволяє моделі зосередитися на важливих частинах вхідних даних та краще враховувати контекстуальні зв'язки між словами.

Усі вищезгадані методи та алгоритми є ключовими елементами сучасних систем обробки природної мови та знаходять широке застосування не тільки в академічних дослідженнях, а й у комерційних проектах, зокрема у юридичній аналітиці, маркетингових дослідженнях, системах рекомендацій та багатьох інших [19]. Правильний вибір методу та його адаптація до конкретних задач та даних може суттєво збільшити ефективність роботи з текстовою інформацією. В даній роботі використовується такі методи машинного навчання, як трансформерна архітектура, дистиляція знань та оптимізація з використанням алгоритму Adam, для досягнення відповідних цілей в обробці природної мови (NLP).

1.3 Аналіз існуючих досліджень. Постановка завдання

Evaluating Document Representations for Content-based Legal Literature Recommendations – приведені результати досліджень методів репрезентації документів для рекомендацій юридичної літератури, зосереджуючись на використанні сучасних підходів до представлення текстових документів у юридичній сфері. Автори аналізують різноманітні методи, які використовують вектори слів, трансформери та графи цитувань для створення репрезентацій документів, з метою знайти найбільш ефективні підходи до отримання семантично схожих на запит користувача документів з юридичної бази даних [33].

Кількісна оцінка представлених методів за допомогою двох наборів даних з анотаціями релевантності для 2964 документів, створених на основі відкритих джерел. Експерименти показують, що використання гібридних

методів, які комбінують текстову інформацію з інформацією про цитування, може покращити результати рекомендації. Дослідження спрямоване на зменшення розриву між науковими досягненнями в галузі рекомендаційних систем та їх практичним застосуванням у юридичній сфері.

Дослідження також має відповідну діаграму, де представлений аналіз ефективності різних методів обробки тексту у відношенні до довжини текстових документів, які були проаналізовані. Різні кольори на діаграмі представляють різні методи векторизації тексту, такі як Paragraph Vectors, fastText та інші, включаючи варіанти, оптимізовані для юридичних текстів, а також моделі, які базуються на BERT (Legal-AUEB-BERT-base).

Кожен стовпець на діаграмі відповідає певному діапазону довжини тексту (кількість слів), який розділений на вісім рівних сегментів (бакетів). Це дозволяє виявити, як методи працюють на коротких, середніх та довгих текстах.

Значення MAP на діаграмі показує, як точно кожен метод може знайти та відсортувати релевантні документи, а нижня частина діаграми (MS Precision) демонструє частоту появи релевантних документів у першій частині відсортованих результатів. Високі значення в цих метриках вказують на більш точну та корисну систему рекомендації або класифікації документів.

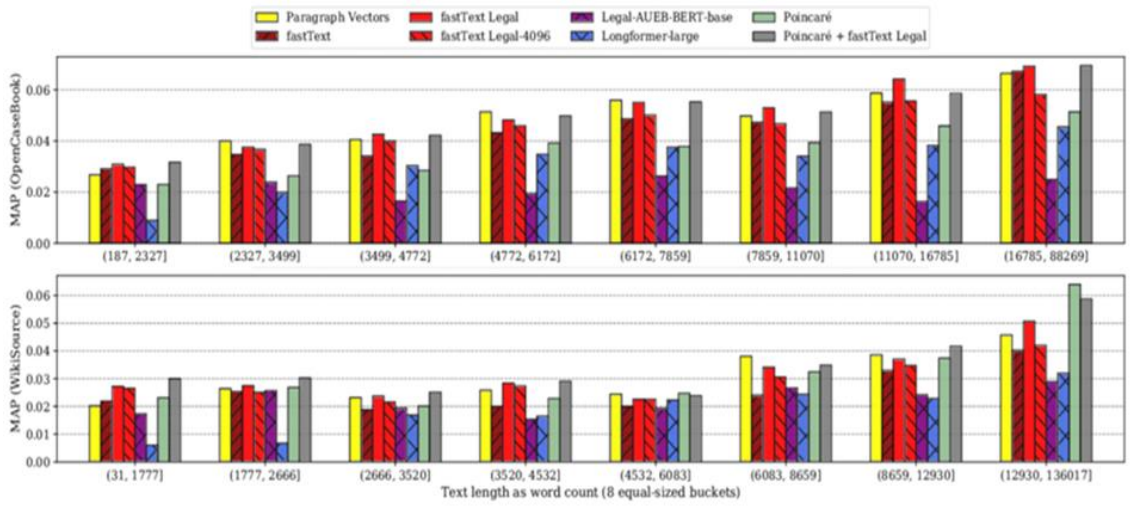

Рисунок 1.1 – Середня точність передбачення (MAP) залежно від кількості слів у документі для набору даних Open Case Book, Wikisource

LEGAL-BERT: The Muppets straight out of Law School – дослідженням в області застосування моделі BERT для завдань обробки природної мови (NLP) у юридичній сфері. Вона розглядає адаптацію моделі BERT до спеціалізованих доменів і виявляє, що загальноприйняті методики попереднього навчання та тонкої настройки, які часто сліпо слідують, не завжди ефективні для юридичної галузі. Автори пропонують систематичне дослідження доступних стратегій при використанні BERT у спеціалізованих доменах, включаючи використання стандартного BERT «з коробки», додаткове попереднє навчання на домен-специфічних корпусах та попереднє навчання BERT з нуля на домен-специфічних корпусах [11]. Їхні висновки вказують на те, що додаткове попереднє навчання або попереднє навчання з нуля на домен-специфічних корпусах дає кращі результати, ніж використання BERT без додаткових налаштувань. Треба зазначити що модель LEGAL-BERT була тренована виключно на юридичних текстах англо-саксонської правової системи, написаних англійською мовою. Відповідно, її ефективність для використання українською мовою є обмеженою.

Natural Language Processing for Information Extraction – охоплює широкий спектр підзадач обробки природної мови (NLP), що стосуються вилучення інформації. Основна мета цієї роботи полягає у визначенні та перетворенні неструктурованої інформації з текстових документів у структуровану форму, яка підходить для комп'ютерної обробки та зберігання [41]. Розглянуто загальні аспекти вилучення інформації, актуальність завдань NLP та їх важливість у сучасному цифровому суспільстві, зокрема, для сфер як юриспруденція, медицина та бізнес-аналітика. Описані основні завдання IE, такі як розпізнавання іменованих сутностей, зв'язування іменованих сутностей, розв'язання кореференції, екстракція відносин та розуміння та доповнення баз знань.

Фундаментальні методи та алгоритми NLP, які є необхідними для розробки та оптимізації нейромережевих систем здатних категоризувати текстові документи. Зокрема, такі системи використовують техніки NLP для

розуміння семантики та контексту тексту, що є ключовими для точної категоризації. Крім того, запропоновано інтеграцію технологій вилучення інформації з системами пошуку інформації, їх застосування та обговорено поточні виклики та напрямки майбутніх досліджень.

Метою роботи є розробка та аналіз нейромережевої системи, здатної ефективно категоризувати текстові документи. В якості класів було обрано розділення на юридичні та неюридичні тексти.

Для досягнення мети необхідно виконати ряд завдань, а саме:

1. Класифікація текстів: Розробити нейромережі, здатної ідентифікувати та категоризувати юридичні документи з високою точністю.

2. Автоматичне узагальнення: Аналіз можливостей системи для створення резюме юридичних текстів з метою підвищення ефективності їхнього використання.

3. Витягання інформації: Дослідження здатності моделі ізолювати ключову інформацію.

2 ШТУЧНА НЕЙРОННА МЕРЕЖА

2.1 Визначення Штучної нейронної мережі

Штучна нейронна мережа – програма або модель машинного навчання, яка приймає рішення способом, схожим на роботу людського мозку, використовуючи процеси, що імітують роботу біологічних нейронів [46]. Кожна нейронна мережа складається з шарів вузлів або штучних нейронів - вхідного шару, одного або кількох прихованих шарів і вихідного шару. Нейронні мережі покладаються на навчальні дані для покращення своєї точності з часом. Після їх точного налаштування на точність вони стають потужними інструментами в інформатиці та штучному інтелекті, що дозволяє класифікувати та кластеризувати дані з високою швидкістю[18]. З завданнями із розпізнавання тексту або зображень моделі ШІ справляються значно швидше і іноді краще ніж звичайна людина. Одним із найвідоміших прикладів нейронної мережі є пошуковий алгоритм Google.

Нейронні мережі іноді називають штучними нейронними мережами (ШНМ) або імітованими нейронними мережами (ІНМ). Вони є підмножиною машинного навчання та є основою моделей глибокого навчання. Глибоке навчання, яке є розширенням концепції нейронних мереж, використовує багатошарові архітектури для вирішення ще більш складних завдань [45]. Ці моделі можуть самостійно вчитися високорівневим абстракціям у даних, використовуючи велику кількість прихованих шарів і величезні обсяги даних для тренування. Це дозволяє їм виконувати складні завдання, такі як автоматичний переклад мов, генерація тексту, автономне керування транспортними засобами та розпізнавання об'єктів у відео в реальному часі.

Однією з ключових особливостей нейронних мереж є їх здатність визначати і важливість кожного атрибуту вхідних даних через процес, відомий як зворотне поширення помилки. Це дозволяє мережі адаптуватися та

оптимізувати свої ваги для кожного нейрона, щоб максимально точно відповідати очікуваним виходам [40].

З появою технологій великих даних і значним збільшенням обчислювальних потужностей, таких як графічні процесори (GPU) і спеціалізовані апаратні засоби, наприклад TPU від Google, можливості нейронних мереж і глибокого навчання значно розширились. Це призвело до значних проривів у багатьох галузях, включаючи медицину, де штучний інтелект застосовується для діагностики захворювань на ранніх стадіях, в автомобільній промисловості для розробки систем автономного водіння та в інших сферах, де потрібна висока точність та швидкість обробки даних[14].

2.2 BERT (Bidirectional Encoder Representations from Transformers)

Покращення в галузі штучного інтелекту та машинного навчання постійно просувають межі можливого, а одним із найбільш значних проривів у цій області стала поява моделі BERT (Bidirectional Encoder Representations from Transformers). Розроблена дослідниками з Google, BERT представляє собою революційний підхід у розумінні мови, який значно покращує здатність комп'ютерних систем розуміти та обробляти природну мову [24].

BERT базується на архітектурі Transformer, яка була представлена в 2017 році та відрізняється своєю здатністю одночасно обробляти всі слова в тексті, на відміну від попередніх моделей, які читали текст послідовно, слово за словом. Ця особливість дозволяє моделі краще вловлювати контекст та значення слів, що робить її надзвичайно ефективною для завдань обробки природньої мови [12].

Одна з ключових особливостей BERT полягає в її здатності до двонаправленого навчання [36]. Традиційні моделі машинного навчання для обробки мови навчаються на текстових даних, аналізуючи їх або зліва направо, або справа наліво. Натомість BERT ефективно використовує контекст з обох

боків для кожного слова в реченні, що дозволяє моделі краще розуміти семантику та відносини між словами [35].

Використання BERT та подібних моделей відкриває нові горизонти в застосуванні штучного інтелекту, особливо у сфері розуміння текстів, автоматичного перекладу, відповідей на запитання та інших завдань, пов'язаних з обробкою природньої мови. Наприклад, завдяки BERT пошукові системи стали здатними краще розуміти нюанси запитів користувачів та надавати більш точні та релевантні результати [49].

Розвиток та впровадження моделей на кшталт BERT є свідченням того, як глибоке навчання та штучний інтелект продовжують трансформувати підходи до обробки та аналізу великих обсягів даних[28]. Це не лише змінює ландшафт технологій, але й створює нові можливості для покращення та оптимізації багатьох процесів у різних галузях.

2.3 DISTILBERT (Bidirectional Encoder Representations from Transformers)

DISTILBERT представляє собою легшу та ефективнішу версію моделі BERT, яка була розроблена для того, щоб зменшити обчислювальні витрати та поліпшити швидкість виконання без істотної втрати в точності. Ця модель використовує техніку, відому як дистиляція знань, де «вчитель» (велика, складна модель) передає свої знання "учневі" (менша, простіша модель)[4].

Дистиляція знань дозволяє DISTILBERT засвоїти важливі характеристики з BERT, зберігаючи при цьому лише частину вихідної кількості параметрів. Наприклад, DISTILBERT зазвичай має приблизно 40% менше параметрів порівняно з BERT, але здатна зберегти 97% точності її «вчителя» на задачах розуміння природної мови [39].

Ключові характеристики DISTILBERT:

1. Ефективність: завдяки значно меншій кількості параметрів, DISTILBERT вимагає менше обчислювальних ресурсів для тренування та

виконання, що робить її більш доступною для застосування на пристроях з обмеженими ресурсами або у ситуаціях, де швидкість є критичною [16].

2. Збереження точності: незважаючи на свою компактність, DISTILBERT здатна виконувати багато задач обробки природної мови з порівняно високою точністю, завдяки ефективній передачі знань від BERT [16].

3. Гнучкість: DISTILBERT може бути застосована до широкого спектру задач NLP, таких як класифікація тексту, відповіді на питання, і навіть генерація тексту, демонструючи універсальність оригінальної моделі BERT, але з меншими вимогами до ресурсів [3].

Дистиляція знань у контексті DISTILBERT полягає в тренуванні моделі «учня» на основі як реальних даних, так і прогнозів моделі «вчителя». «Учень» намагається імітувати поведінку «вчителя», оптимізуючи свої параметри не тільки для правильних відповідей, але й для того, як «вчитель» розподіляє ймовірності між різними класами. Цей процес дозволяє «учневі» вловити тонші нюанси рішень «вчителя», навіть маючи менше параметрів [42].

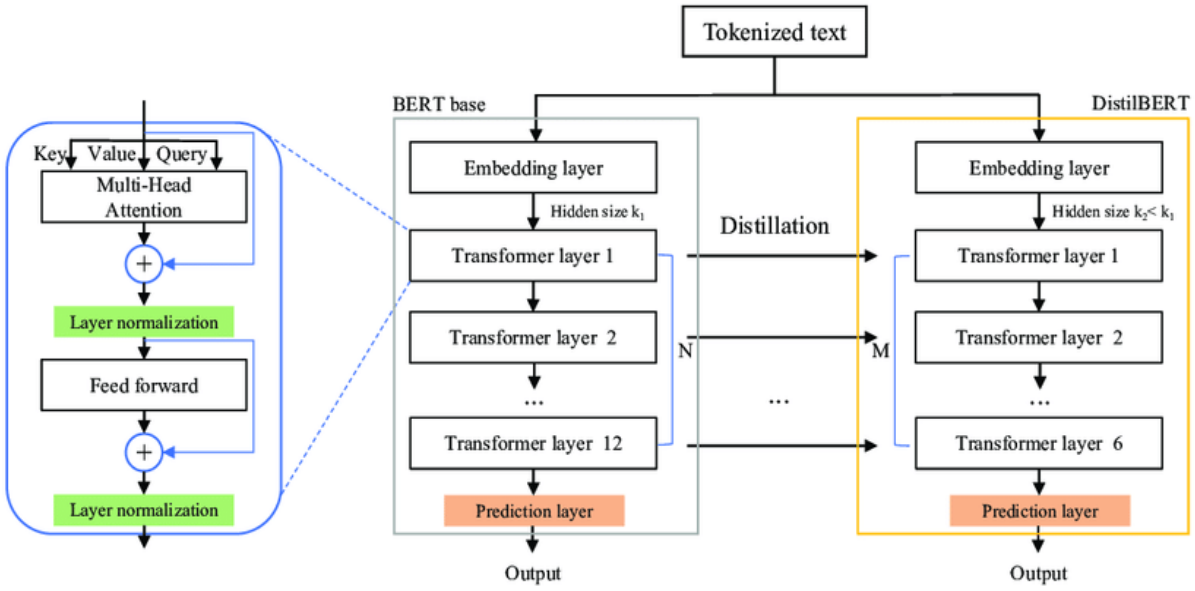

Рисунок 3.1 – Архітектура та компоненти моделі DistilBERT

На рисунку 3.1 продемонстрована архітектура моделі DistilBERT порівняно з BERT. Цей процес включає передачу знань від повної моделі BERT до спрощеної DistilBERT. Це досягається за рахунок тренування DistilBERT на основі виходів BERT, таким чином забезпечуючи, що менша модель засвоює поведінкові шаблони більшої моделі.

Структура окремого блоку трансформера включає операції Multi-Head Attention та Feed Forward, що описуються раніше. Ключовим елементом є механізм уваги, який аналізує взаємозв'язки між усіма словами у реченні, враховуючи їхні «ключі» (Keys), «значення» (Values) та «запити» (Queries), що дозволяє моделі зрозуміти контекст слова не лише залежно від його позиції, але й від його взаємодії з іншими словами [28].

У блоку трансформера виконується також шар нормалізації (Layer Normalization), який стабілізує навчання шляхом нормалізації векторів, що входять до блоку. Це допомагає запобігати занадто великим відхиленням у навчанні та сприяє швидшій та стабільнішій збіжності.

Після кожного блоку уваги, додатковий шар (Feed Forward) обробляє інформацію далі перед тим, як вона потрапить до наступного шару трансформера або виводиться як результат. Шар Feed Forward складається з простих персептронів і відповідає за фінальне трансформування векторів відповідно до задач, які модель має виконати.

У випадку DistilBERT, процес дистиляції включає кероване навчання меншої моделі за допомогою прогнозів більшої моделі BERT, так що DistilBERT «наслідує» роботу BERT. Під час тренування DistilBERT спробує максимально наблизити свої прогнози до виходів BERT, навіть не маючи однакової кількості шарів трансформера. Таким чином, DistilBERT може використовувати меншу кількість ресурсів, зберігаючи при цьому високу точність прогнозів. Ця схема ілюструє, як більш складні та важкі моделі можуть бути «дистильовані» до більш легких версій, що зберігають значну частину їхньої ефективності, але водночас вимагають менше обчислювальних ресурсів і є більш практичними для використання в реальному світі.

BERT base.: токенізований текст. Спочатку текст перетворюється на токени, що є вхідними даними для моделі.

Вектори вкладень (Embedding layer): токени конвертуються в вектори вкладень, які подають числове представлення кожного токена [37].

Transformer layers: BERT базується на N шарах трансформера, де кожен шар складається з блоків уваги (multi-head attention) [8] та блоків прямого поширення (feed-forward networks) [34]. Вони забезпечують здатність моделі розуміти контекст кожного токена з усіма іншими токенами в тексті.

Multi-Head Attention: виконує операції уваги декілька разів паралельно, що дозволяє моделі краще зосереджуватися на різних частинах вхідних даних [46].

Layer Normalization: нормалізація шарів використовується для стабілізації навчання моделі [7].

Feed Forward: це прості шари нейронної мережі, які застосовуються після кожної операції уваги [9].

Prediction Layer: на виході, після останнього шару трансформера, отримані вектори використовуються для виконання специфічних задач, таких як класифікація [29].

DistilBERT. Embedding layer: аналогічно до BERT, текст спочатку токенізується, а потім конвертується в вектори вкладень [4].

Transformer layers: у DistilBERT кількість шарів трансформера зменшена до M (зазвичай M=N/2 для DistilBERT). Кожен шар включає ті ж самі компоненти, що й у BERT, але зменшена кількість шарів дозволяє моделі працювати швидше і вимагати менше обчислювальних ресурсів [15].

Prediction Layer: так само, як і BERT [20], DistilBERT використовує вивчені вектори для передбачення вихідних даних [15].

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Модель нейромережі

Система приймає на вхід текстові документи, які поділяються на дві категорії: юридичні та неюридичні. Тексти перетворюються на послідовності токенів за допомогою Токенізатора(DistilBertTokenizerFast). Для кожного тексту T , він перетворюється на послідовність токенів що можна представти у вигляді виразу:

$$\mathrm{X} = [\mathrm{x}_1 , \mathrm{x}_2 , \ldots , \mathrm{x}_\mathrm{n}], \tag{3.1}$$

де: X — послідовність токенів тексту T;

x_i – індекс токена в словнику моделі;

n – довжина послідовності (максимальна довжина послідовності може дорівнювати 512. Це обумовлено тим, що архітектура BERT, на якій базується DistilBERT, була спроектована з можливістю обробки послідовностей довжиною до 512 токенів. Технічне обмеження пов'язане з розміром позиційних ембедингів, які кодують порядок токенів у послідовності.).

Embedding Layer. Кожен токен x_i перетворюється на високовимірний вектор ембедингу, і до цих ембедингів додаються позиційні ембединги для збереження інформації про порядок токенів. Це можна розрахувати за формулою:

$$E_i = \mathrm{Embedding}(\mathrm{x_i}) + \mathrm{PositionEmbedding}(\mathrm{i}), \tag{3.2}$$

де: E_i – вектор ембедингу для i-го токена;

$Embedding(x_i)$ – ембединг токена x_i, який перетворює індекс токена на високовимірний вектор;

$PositionEmbedding(i)$ – позиційний ембедінг для i-го токена, що додається до ембедингу токена для збереження інформації про порядок токенів у послідовності.

Трансформерні блоки: модель використовує 6 трансформерних блоків, кожен з яких складається з механізму самоуваги та підшару прямого поширення (feed-forward network, FFN), що можна представити у вигляді виразу:

$$\mathrm{H}^{(\mathrm{l})} = \mathrm{FFN}\left(\mathrm{SelfAttention}\left(\mathrm{H}^{(\mathrm{l}-1)}\right)\right), \quad (3.3)$$

де: $\mathrm{H}^{(\mathrm{l})}$ – вихід l-го трансформерного блоку;

FFN – підшар прямого поширення (feed-forward network), який застосовується до результату самоуваги;

$\mathrm{SelfAttention}(\mathrm{H}^{(\mathrm{l}-1)})$ – механізм самоуваги, що дозволяє моделі зосереджуватися на різних частинах вхідної послідовності, використовуючи вихід з попереднього (l−1) блоку як вхід.

Механізм самоуваги дозволяє моделі зосереджуватися на різних частинах вхідної послідовності при обчисленні виходу кожного токена.

Вихідний шар і класифікація. На останньому етапі, вектор останнього токена HN(L) або агрегований вектор проходить через повнозв'язний шар (або шари) для отримання вектора логітів Z, який відображається в ймовірності класів за допомогою softmax, що можна представити у вигляді виразу:

$$\mathrm{P}(\mathrm{class} \mid \mathrm{X}) = \mathrm{softmax}(\mathrm{Z}), \quad (3.4)$$

де: $\mathrm{P}(\mathrm{class} \mid \mathrm{X})$ – ймовірність того, що вхідна послідовність X належить до певного класу;

Z— вектор логітів, отриманий з вихідного шару моделі, який відображає вхідну послідовність до простору класів;9

softmax – функція, яка перетворює вектор логітів Z у розподіл ймовірностей по класах, забезпечуючи, що сума ймовірностей усіх класів дорівнює 1.

Функція втрат і оптимізація. Система використовує Sparse Categorical Crossentropy як функцію втрат для задачі класифікації. Ця функція ідеально підходить для ситуацій, де існує кілька класів, і кожен об'єкт належить рівно до одного класу. Формально, для одного входу, Sparse Categorical Crossentropy можна представити у вигляді виразу:

$$L(y, \hat{y}) = -\Sigma_{i=1}^{C} y_i \log(\hat{y}_i), \quad (3.5)$$

де $L(y, \hat{y})$ – функція втрат (loss function), яка вимірює, наскільки добре модель передбачає правильні класи;

y – істинна мітка класу в форматі цілого числа (не one-hot кодування);

$\hat{y}$ – прогнозований розподіл ймовірностей для всіх класів, отриманий з моделі;

C – кількість класів;

y_i – індикатор, що дорівнює 1 для істинного класу і 0 для інших класів;

$log(\hat{y}_i)$ – натуральний логарифм ймовірності, прогнозованої моделлю для класу i.

Оптимізація. Для оптимізації параметрів моделі використовується оптимізатор Adam. Adam (Adaptive Moment Estimation) який є методом стохастичного градієнтного спуску, і адаптує навчальний крок для кожного параметра на основі оцінок перших і других моментів градієнтів. Він об'єднує переваги двох інших розширень градієнтного спуску: AdaGrad, який працює добре з рідкісними даними, та RMSProp, який працює добре в онлайн- та нелінійних налаштуваннях.

Adam оновлює ваги w використовуючи обчислення:

$$w_{i+1} = w_t - \frac{\eta}{\sqrt{\widehat{w_t}} + \epsilon} \widehat{m_t}, \quad (3.6)$$

де: w_{i+1} – відповідає оновленим значенням ваги моделі на наступному кроці $i + 1$;

w_t – ваги на кроку t;

η – навчальний крок (learning rate);

$\widehat{m_t}$ – оцінка першого моменту (середньозважене значення градієнтів);

$\widehat{v_t}$ – оцінка другого моменту (середньозважене значення квадратів градієнтів);

ϵ – дуже мале число, щоб уникнути ділення на нуль.

Adam автоматично коригує навчальний крок для кожного параметра, що робить його ефективним і знижує потребу в ручній настройці навчального кроку.

У контексті цієї системи, використання Sparse Categorical Crossentropy дозволяє моделі ефективно вчитися відповідати на задачу класифікації, тоді як оптимізатор Adam забезпечує ефективне та адаптивне оновлення ваг моделі на основі її продуктивності [27]. Це допомагає досягти високої точності на тренувальних та тестових наборах даних, як показано у результаті тренувань.

3.2 Опис роботи моделі

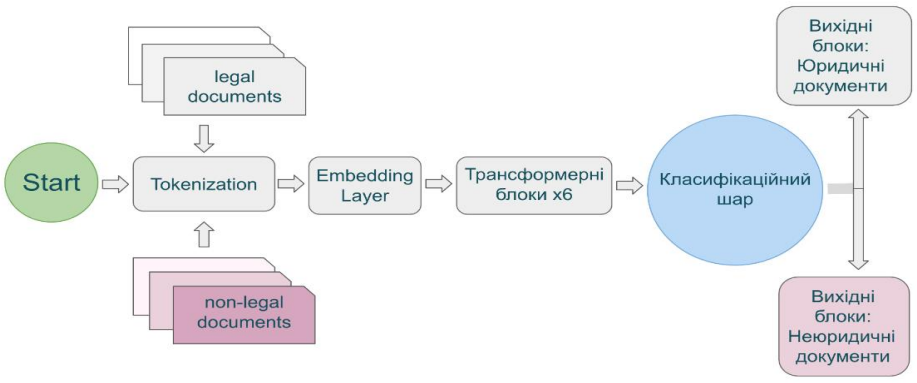

Рисунок 3.2 – Загальна схема нейромережевої системи для категоризації текстових документів

Початок (Start): процес починається з ініціалізації системи та готовності до прийняття вхідних даних.

Вхідні дані: система приймає на вхід два типи документів: юридичні (law_documents) та неюридичні (non-legal documents). Кожен документ має відповідну мітку, яка вказує на його класифікацію.

Токенізація: документи проходять через процес токенізації, де вони перетворюються на послідовності токенів за допомогою DistilBertTokenizerFast. Токени представляють собою мінімальні одиниці тексту, які може обробляти модель.

Embedding Layer: після токенізації, кожен токен перетворюється на вектор у високовимірному просторі, і до нього додаються позиційні ембедінги. Це дозволяє моделі зберігати інформацію не тільки про значення кожного слова, але й про його положення в тексті.

Transformer Blocks: вектори проходять послідовно через шість трансформерних блоків (Transformer blocks x6), де кожен блок виконує операції самоуваги та подальшої обробки для здобуття контекстуалізованих представлень токенів.

Classification Layer: на виході з трансформерних блоків, отримані векторні представлення проходять через класифікаційний шар (Classification Layer), який містить один або кілька повнозв'язних шарів та використовує функцію softmax для визначення ймовірності кожного з класів.

Вихідні блоки: на виході системи, документи класифікуються як юридичні (law_documents Output Blocks) та неюридичні (non-legal documents Output Blocks) згідно з їхніми ймовірностями, які надає класифікаційний шар.

Реалізація включає кілька ключових етапів: конфігурація середовища, завантаження та підготовка даних, токенізація, створення та тренування моделі, та оцінка результатів. Для реалізації поставленої задачі необхідно було створити дві окремі програми: Програму для навчання моделі, Програму для оцінки ефективності моделі.

Лістінг коду для навчання моделі. Конфігурація середовища. На самому початку процесу встановлюються параметри середовища для контролю виконання токенізатора та налаштувань логування TensorFlow, щоб забезпечити оптимальну працездатність і чистоту виводу під час розробки.

Лістинг 3.1 – Фрагмент коду конфігурації середовища

```
import os
import tensorflow as tf

os.environ["TOKENIZERS_PARALLELISM"] = "false"
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '1'
tf.get_logger().setLevel('INFO')
```

Налаштування логування. Конфігурація системи логування налаштована на рівень INFO для забезпечення зручності відстеження подій та забезпечення належної інформаційної підтримки під час розробки та відладки.

Лістинг 3.2 – Фрагмент коду налаштування логування

```
import logging

logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)
```

Завантаження та підготовка даних. Завантаження даних з файлів і їх підготовка включає в себе визначення функції для читання текстів із заданою міткою кінця документу, формування єдиного датасету з мітками.

Лістинг 3.3 – Фрагмент коду завантаження та підготовка даних

```
def load_documents(file_path, delimiter="\n\n\n"):
    with open(file_path, 'r', encoding='utf-8') as file:
        documents = file.read().split(delimiter)
    return [doc.strip() for doc in documents if doc.strip()]

law_documents =
load_documents('documents/extracted_law_texts.txt', '---END-OF-
DOCUMENT---')
```

```
unlaw_documents =
load_documents('documents/extracted_unlaw_texts.txt')
documents = law_documents + unlaw_documents
labels = [1] * len(law_documents) + [0] * len(unlaw_documents)
```

Токенізація та підготовка даних для моделі. Токенізація текстів за допомогою предвстановленого токенізатора DistilBert, їх поділ на тренувальні та тестові набори та підготовка відповідних датасетів для TensorFlow.

Лістинг 3.4 – Фрагмент коду токенізації та підготовки даних для моделі

```
from sklearn.model_selection import train_test_split
from transformers import DistilBertTokenizerFast
from tensorflow import data

tokenizer = DistilBertTokenizerFast.from_pretrained('distilbert-
base-multilingual-cased')
train_docs, test_docs, train_labels, test_labels =
train_test_split(documents, labels, test_size=0.2,
random_state=42)
train_encodings = tokenizer(train_docs, truncation=True,
padding=True, max_length=512)
test_encodings = tokenizer(test_docs, truncation=True,
padding=True, max_length=512)

train_dataset =
data.Dataset.from_tensor_slices((dict(train_encodings),
train_labels))
test_dataset =
data.Dataset.from_tensor_slices((dict(test_encodings),
test_labels))
```

Ініціалізація і тренування моделі. Створення моделі DistilBert для класифікації, її компіляція з використанням оптимізатора Adam та тренування з допомогою TensorBoard для моніторингу процесу.

Лістинг 3.5 – Фрагмент коду ініціалізації і тренування моделі

```
from transformers import TFDistilBertForSequenceClassification

model =
TFDistilBertForSequenceClassification.from_pretrained('distilber
t-base-multilingual-cased', num_labels=2)
optimizer = tf.keras.optimizers.legacy.Adam(learning_rate=5e-5)
```

```
loss =
tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
metrics = ['accuracy']
model.compile(optimizer=optimizer, loss=loss, metrics=metrics)

tensorboard_callback =
tf.keras.callbacks.TensorBoard(log_dir='./logs',
histogram_freq=1)
history = model.fit(train_dataset.shuffle(1000).batch(16),
epochs=3, batch_size=16,
                    validation_data=test_dataset.batch(16),
callbacks=[tensorboard_callback])
```

Оцінка та збереження моделі. Оцінка ефективності моделі на тестових даних та збереження результатів тренування і самої моделі для подальшого використання.

Лістинг 3.6 – Фрагмент коду оцінки та збереження моделі

```
import json

evaluation_results = model.evaluate(test_dataset.batch(16))
logger.info(f"Test Loss: {evaluation_results[0]}, Test Accuracy:
{evaluation_results[1]}")
os.makedirs('result_directory', exist_ok=True)
with open('result_directory/training_history.json', 'w') as f:
    json.dump(history.history, f)

model.save_pretrained('result_directory/model')
```

Після виконання цієї частини коду отримуємо файл training_history.json, який містить інформацію результати навчання моделі. Після виконання всього коду вище, модель зберігається у вигляді двох файлів: tf_model.h5 та config.json. Файл tf_model.h5 містить збережену структуру та ваги моделі, а файл config.json зберігає конфігурацію моделі.

Лістінг коду для оцінки ефективності моделі. Модель була ініціалізована з використанням вагів предтренованої моделі DistilBERT і налаштована на задачу бінарної класифікації.

Імпорт бібліотек і налаштування середовища. На початку коду імпортуються необхідні бібліотеки, включаючи TensorFlow і модулі для обробки тексту та класифікації. Встановлюється змінна середовища для

відключення паралельної токенізації, що запобігає можливим помилкам в многопоточному середовищі. Конфігурація логування встановлюється для збереження результатів тестування в файл.

Рисунок 3.3 – Текст файлу config.json

Лістинг 3.7 – Фрагмент коду імпорту бібліотек і налаштування середовища

```
import tensorflow as tf

from sklearn.metrics import f1_score, precision_score,
recall_score
from transformers import DistilBertTokenizerFast,
```

```
TFDistilBertForSequenceClassification
from tensorflow import data
import logging
import os

os.environ["TOKENIZERS_PARALLELISM"] = "false"
logging.basicConfig(filename='evaluation_test_results.log',
level=logging.INFO)
```

Завантаження та підготовка даних. Функція load_documents читає текстові файли, розділяючи документи за заданими роздільниками. Це дозволяє імпортувати та організувати дані з файлів для юридичних та неюридичних документів окремо, готуючи їх до подальшої обробки.

Лістинг 3.8 – Фрагмент коду завантаження та підготовка даних

```
def load_documents(file_path, delimiter="\n\n\n"):
    with open(file_path, 'r', encoding='utf-8') as file:
        documents = file.read().split(delimiter)
    return [doc.strip() for doc in documents if doc.strip()]

test_law_texts =
load_documents('test_datas/extracted_law_test_texts.txt', '---
END-OF-DOCUMENT---')
test_unlaw_texts =
load_documents('test_datas/extracted_unlaw_test_texts.txt',
'\n\n\n')
test_fiction_texts =
load_documents('test_datas/extracted_fiction_test_texts.txt',
'\n\n\n')
test_wikipedia_texts =
load_documents('test_datas/extracted_wikipedia_test_texts.txt',
'\n\n\n')
```

Токенізація та підготовка тестового датасету. Тексти об’єднуються та токенізуються за допомогою токенізатора DistilBert. Створюється тестовий датасет з кодованих текстів та їхніх міток, який готується до подачі у модель.

Лістинг 3.9 – Фрагмент коду токенізації та підготовки тестового датасету

```
tokenizer = DistilBertTokenizerFast.from_pretrained('distilbert-
base-multilingual-cased')
test_texts = test_law_texts + test_unlaw_texts +
test_fiction_texts + test_wikipedia_texts
test_encodings = tokenizer(test_texts, truncation=True,
```

```
padding=True, max_length=512)
true_labels = [1] * len(test_law_texts) + [0] *
len(test_unlaw_texts) + [0] * len(test_fiction_texts) + [0] *
len(test_wikipedia_texts)
test_dataset =
data.Dataset.from_tensor_slices((dict(test_encodings),
true_labels))
```

Завантаження моделі, її компіляція та оцінка. Модель завантажується з попередньо вказаної директорії та компілюється з використанням визначених параметрів оптимізатора, функції втрат і метрик. Після цього проводиться оцінка моделі на тестовому датасеті.

Лістинг 3.10 – Фрагмент коду завантаження моделі, її компіляція та оцінка

```
model =
TFDistilBertForSequenceClassification.from_pretrained('result_di
rectory/model')
optimizer = tf.keras.optimizers.legacy.Adam(learning_rate=5e-5)
loss =
tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
metrics = ['accuracy']
model.compile(optimizer=optimizer, loss=loss, metrics=metrics)
evaluation_results = model.evaluate(test_dataset.batch(16))
```

Обчислення метрик та логування результатів. Після отримання передбачень моделі вираховуються метрики якості, такі як F1-скор, точність та повнота. Результати реєструються в лог-файлі та виводяться для аналізу.

Лістинг 3.11 – Фрагмент коду завантаження моделі, її компіляція та оцінка

```
predicted_probabilities = model.predict(test_dataset.batch(16))
predicted_labels = tf.argmax(predicted_probabilities.logits,
axis=1)
f1 = f1_score(true_labels, predicted_labels)
precision = precision_score(true_labels, predicted_labels)
recall = recall_score(true_labels, predicted_labels)
logging.info("Eval loss: %f", evaluation_results[0])
logging.info("Eval accuracy: %f", evaluation_results[1])
logging.info("F1 Score: %f", f1)
logging.info("Precision: %f", precision)
logging.info("Recall: %f", recall)
print("Eval loss:", evaluation_results[0])
print("Eval accuracy:", evaluation_results[1])
```

```
print("F1 Score:", f1)
print("Precision:", precision)
print("Recall:", recall)
```

Після виконання цієї частини коду ми отримаємо файл training_history.json який містить інформацію результати навчання моделі. На цьому тренування моделі завершено.

4 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

4.1. Порядок проведення експерименту

Підготовка до експерименту**.** Експеримент розпочався з вибору моделі distilbert-base-multilingual-cased через її багатомовність і здатність ефективно обробляти великі обсяги тексту. Модель була обрана як основа для розробки системи категоризації текстових документів на юридичні та неюридичні. Для тренування та валідації використовувалися дані, отримані з публічно доступних джерел, зокрема, тексти судових рішень для юридичних документів і публікації з соціальних мереж для неюридичних документів. які були взяти з ресурсу lang.org.ua, який є обширним архівом текстів українською мовою, категоризованих за тематикою.

Технічне обладнання. Для проведення експерименту було використано наступне обладнання – комп’ютер: модель: MacBook Pro. Ідентифікатор моделі: MacBookPro17,1. Чип: Apple M1. Загальна кількість ядер: 8 (4 продуктивних і 4 енергоефективних). Оперативна пам'ять: 16 ГБ. Версія системного прошивання: 10151.121.1. Версія завантажувача ОС: 10151.121.1.

Налаштування експерименту. Перед тренуванням моделі було здійснено відстеження файлів за допомогою Git LFS, щоб ефективно керувати великими обсягами даних. Модель була ініціалізована з наступними параметрами, які вказано у файлі *config.json*:

- активаційна функція: GELU;
- кількість шарів трансформера: 6;
- кількість голів уваги: 12;
- максимальна довжина послідовності: 512 токенів;
- розмір вокабуляру: 119547.

Ці параметри були обрані для оптимізації продуктивності та точності моделі.

Процес тренування. Ініціалізація та тренування моделі: Модель була ініціалізована з використанням попередньо навченої моделі DistilBERT та налаштована для задачі бінарної класифікації. Тренування проводилося на визначених наборах даних протягом трьох епох, де кожна епоха дозволяла моделі краще адаптуватися до особливостей даних [2].

Моніторинг результатів. Використовуючи показники втрат (loss) та точності (accuracy) на тренувальному та валідаційному наборах даних, було здійснено моніторинг процесу навчання. Значення втрат знижувалися з кожною епохою, тоді як точність зростала, що свідчить про ефективне навчання моделі.

Аналіз результатів. Після завершення тренування моделі було проведено детальний аналіз результатів.

Після провденення тренування був отриманий файл training_history.json, що містить інформацію про зміну значень втрат (loss) та точності (accuracy) моделі під час навчання та валідації на даних.


```
{
    "loss": [0.0095907844626663, 0.001435651909599651, 1.386978735574243e-05],
    "accuracy": [0.9978122115135193, 0.9997499585151672, 1.0],
    "val_loss": [0.004804324824362993, 1.6209438399528153e-05, 5.182974746276159e-
06],"val_accuracy": [0.9984999895825, 1.0, 1.0]
}
```


Рисунок 4.1 – Текст файлу training_history.json

Код має таку структуру. Структура даних:

loss: список, який показує значення функції втрат на тренувальному наборі даних після кожної епохи.

accuracy: список, який вказує точність моделі на тренувальному наборі після кожної епохи.

val_loss: список значень функції втрат на валідаційному наборі даних після кожної епохи.

val_accuracy: список, який показує точність моделі на валідаційному наборі після кожної епохи.

Розшифровка. Перша епоха:

loss: 0.0096 – модель мала відносно низьку початкову втрату на тренувальних даних.

accuracy: 99.78% – висока точність на тренувальному наборі.

val_loss: 0.0048 – втрати на валідаційному наборі були нижчими, ніж на тренувальному.

val_accuracy: 99.85% – точність на валідаційному наборі також висока.

Друга епоха:

loss: 0.0014 – значне зниження втрат на тренувальному наборі, що свідчить про ефективність навчання.

accuracy: 99.97% – подальше поліпшення точності.

val_loss: 1.6209e-05 – втрати на валідаційному наборі стали значно нижчими, майже нульовими.

val_accuracy: 100% – ідеальна точність на валідаційному наборі.

Третя епоха:

loss: 1.387e-05 – втрати на тренувальному наборі знизились до мінімуму.

accuracy: 100% – модель досягла ідеальної точності на тренувальних даних.

val_loss: 5.183e-06 – втрати на валідаційному наборі ще зменшились.

val_accuracy: 100% – збереження ідеальної точності на валідаційному наборі.

4.2 Тестування моделі

Тестування моделі відбувалось наступним чином:

1. Завантаження та Підготовка Тестових Даних. Були завантажені та підготовлені тестові дані з чотирьох різних категорій: юридичні тексти, неюридичні тексти, тексти художньої літератури та статті з Вікіпедії. Це забезпечило різноманітність тестового набору для оцінки загальної здатності моделі [2].

2. Токенизація Тестових Текстів. Використовуючи токенизатор DistilBertTokenizerFast, було здійснено токенизацію текстів, адаптуючи їх для подальшого використання моделлю. Було враховано максимальну довжину послідовності у 512 токенів для забезпечення консистентності вхідних даних.

3. Завантаження Предтренованої Моделі. Модель була ініціалізована з використанням вагів предтренованої моделі DistilBERT і налаштована на задачу бінарної класифікації.

4. Компіляція Моделі. Для оптимізації моделі було використано оптимізатор Adam, налаштований на навчання з швидкістю 5e-5. Також було встановлено втрату SparseCategoricalCrossentropy та метрику точності.

5. Оцінка Моделі на Тестовому Наборі Даних. Модель була оцінена на підготовленому тестовому наборі даних. Використовувалася партія по 16 прикладів для ефективності обчислень.

6. Вирахування параметрів ефективності моделі. На основі предсказань моделі були вираховані параметри ефективності моделі - втрати, точність, F1-оцінка, точність (precision) та повнота (recall). Результати показали високу ефективність моделі: мінімальні втрати, висока точність та високі значення F1-оцінки, точності та повноти.

7. Логування та Вивід Результатів. Результати експерименту були залоговані та виведені для подальшого аналізу в файл evaluation_test_results.log (рисунок 4.2)

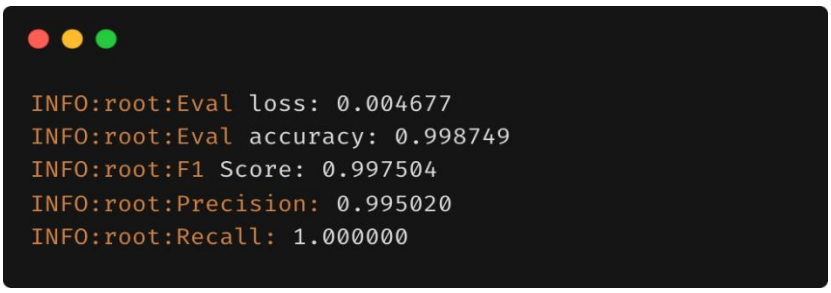

Рисунок 4.2 – Evaluation_test_results.log

4.3 Результати і аналіз експериментальних досліджень

Результати експериментальних досліджень цієї нейромережевої системи для категоризації текстових документів, заснованої на моделі distilbert-base-multilingual-cased,про високу ефективність і точність моделі у виконанні поставленого завдання. Деталізовано результати представлені у таблиці 4.1.

Таблиця 4.1 – Параметри ефективності моделі

Metric	Value
Eval loss	0.004677
Eval accuracy	0.998749
F1 Score	0.997504
Precision	0.995020
Recall	1.000000

Eval loss: це середня втрата (або помилка) моделі на тестових даних. Чим нижче цей показник, тим краще модель впоралася з завданням.

Eval accuracy: це точність моделі, відсоток правильно класифікованих прикладів серед усіх тестових прикладів.

f1 Score: F1-оцінка – це гармонічне середнє між точністю (precision) та відтворенням (recall). Висока F1-оцінка (близько 0.99) вказує на добрий баланс між точністю та відтворенням.

Precision: точність – це відсоток правильних позитивних передбачень відносно усіх позитивних передбачень, які зробила модель.

Recall: відтворення – це відсоток правильних позитивних передбачень відносно усіх позитивних прикладів у тестовому наборі.

Основні результати та досягнення дослідження.

1. Вибір та обґрунтування моделі. Модель DistilBERT була обрана через її здатність забезпечити високу точність обробки тексту при відносно низьких вимогах до обчислювальних ресурсів. Це робить її особливо привабливою для завдань, де потрібна ефективність і швидкість, зокрема для автоматизованої категоризації великих обсягів текстових даних.

2. Підготовка даних та експериментальна база. Для тренування та валідації моделі використовувалися дані з публічно доступних джерел: тексти судових рішень для юридичних документів та публікації з соціальних мереж для неюридичних документів. Це забезпечило різноманітність та релевантність даних, необхідних для високоякісного навчання моделі.

3. Налаштування та параметри моделі. Модель була налаштована з використанням оптимальних параметрів, включаючи активаційну функцію GELU, 6 шарів трансформера, 12 голів уваги, максимальну довжину послідовності у 512 токенів та розмір вокабуляру у 119547 токенів. Ці параметри були обрані для забезпечення оптимальної продуктивності та точності моделі.

4. Процес тренування та результати. Тренування моделі проводилося протягом трьох епох, що дозволило моделі поступово адаптуватися до особливостей навчальних даних. Моніторинг показників втрат та точності на тренувальному і валідаційному наборах даних продемонстрував стабільне зниження втрат і зростання точності, що свідчить про ефективне навчання моделі [50].

5. Оцінка продуктивності та ефективності. Після завершення тренування модель була детально оцінена на тестових даних. Результати показали високу точність категоризації текстових документів, що підтверджує здатність моделі ефективно вирішувати завдання бінарної класифікації текстів на юридичні та неюридичні.

6. Технічне обладнання. Експерименти проводилися на комп’ютері MacBook Pro з чипом Apple M1, який забезпечив необхідну обчислювальну

потужність для тренування та тестування моделі. Це демонструє можливість використання доступного апаратного забезпечення для виконання складних обчислювальних завдань.

7. Практичне значення та застосування. Розроблена система може бути використана для автоматизації процесів категоризації текстових документів у різних сферах, включаючи юриспруденцію, журналістику, бізнес-аналітику та інші галузі, де точність і швидкість обробки текстів мають критичне значення. Система здатна значно підвищити ефективність роботи, зменшити ризик людських помилок та забезпечити швидкий доступ до необхідної інформації.

8. Подальші дослідження та перспективи. Результати дослідження відкривають можливості для подальших розробок у сфері обробки природної мови. Майбутні дослідження можуть включати адаптацію моделі для інших мов, розширення функціональності системи для роботи з різними типами текстових даних та інтеграцію з іншими технологіями штучного інтелекту.

Загалом, кваліфікаційна робота підтверджує важливість і перспективність використання нейромережевих технологій для автоматизації аналізу текстових документів. Запропонована система на основі DistilBERT демонструє високу ефективність і може стати основою для подальших досліджень і розробок у галузі обробки природної мови.

Отримані результати підтверджують вибір архітектури distilbert-base-multilingual-cased як основи для системи категоризації текстових документів. Завдяки своїй багатомовності та здатності ефективно обробляти великі тексти, ця модель виявилася оптимальним рішенням для задачі категорізації текстових документів. Ці результати вказують на великий потенціал використання нейромережевих систем, зокрема, моделі DistilBERT, для автоматизації процесів категоризації юридичних та неюридичних текстових документів. Отримані дані можуть слугувати основою для подальших досліджень та розробки систем автоматизованого аналізу текстів у різних доменах.

ВИСНОВКИ

Досліджено сучасні методи та підходи до категоризації текстових документів із використанням штучних нейронних мереж, зокрема моделей BERT та DistilBERT. Визначено переваги та недоліки цих методів у контексті їх застосування в сфері категоризації текстових документів.

Проаналізовано існуючі дослідження в галузі категоризації текстових документів, включаючи роботи з використанням моделей BERT та Legal-BERT, та зроблено висновки щодо їх ефективності у вирішенні завдань автоматизованого розпізнавання та класифікації текстової інформації.

Запропоновано модель нейромережевої системи для категоризації текстових документів на основі моделі DistilBERT, яка поєднує високу точність та ефективність з низькими вимогами до обчислювальних ресурсів. На основі проведеного дослідження можна зробити висновок про виключну ефективність моделі DistilBERT у задачах категоризації текстових документів.

Розроблено програмну реалізацію системи, яка включає етапи токенізації, підготовки даних, навчання моделі та її оцінки. Програмна реалізація дозволяє автоматизувати процес категоризації документів та забезпечує високу точність результатів.

Апробовано розроблену систему на тестових наборах даних, що включають юридичні та неюридичні тексти. Експериментальні дослідження підтвердили високу ефективність та точність моделі, зокрема досягнуто значення F1-оцінки 0.997504, що свідчить про високий рівень класифікації текстових документів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Рибалов О. О., Фесенко Т. Г. Дослідження засобів інтелектуального аналізу текстових документів [Текст] / О. О. Рибалов, Т. Г. Фесенко // Збірник наукових праць XVI Міжнародної науково-практичної конференції «Академічна й університетська наука: результати та перспективи», 12–13 грудня 2023 року. - Полтава: Полтавська політехніка, 2024. - С. 330–331

2. Рибалов О.О. Оцінка ефективності нейромережевої системи для категоризації текстових документів [Текст] / О. О. Рибалов // 28-й Міжнародний молодіжний форум «Радіоелектроніка та молодь XXI століття», 16-18 квітня 2024 року: зб. матеріалів форуму. Т.5., конференція «Проблеми комп'ютерної інженерії та захисту інформації». - Харків: ХНУРЕ, 2004. - С. 56-57. [Електронний ресурс] – Режим доступу : www URL: https://openarchive.nure.ua/server/api/core/bitstreams/f5c2e4b8-7275-4fd2-b2ef-b5feeca3c89c/content

3. Adoma A. F. Comparative Analyses of Bert, Roberta, Distilbert, and Xlnet for Text-Based Emotion Recognition [Text] /A. F. Adoma, N-M. Henry, W. Chen // IEEE Xplore. 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). 18-20 December 2020. [Електронний ресурс] – Режим доступу : www URL: https://ieeexplore.ieee.org/abstract/document/9317379 (date of access: 09.03.2024)

4. Aguilar G. Knowledge Distillation from Internal Representations [Text] / G. Aguilar, Y. Ling, Y. Zhang, B. Yao B, X. Fan, C. Guo // Cornell University. - 8 Oct 2019. [Електронний ресурс] – Режим доступу : www URL: https://ar5iv.labs.arxiv.org/html/1910.03723 (date of access: 09.03.2024)

5. Alaparthi S. Bidirectional Encoder Representations from Transformers (BERT): A sentiment analysis odyssey [Text] / S. Alaparthi, M. Mishra // Cornell University. - 2 Jul 2020. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2007.01127(date of access: 09.03.2024)

6. An Efficient Sparse Inference Software Accelerator for Transformer-based Language Models on CPUs [Text] / H. Shen, H. Meng, B. Dong, Z. Wang, O. Zafrir, Y. Ding, Y. Luo, H. Chang, Q. Gao, Z. Wang, G.Boudoukh, M.Wasserblat // Cornell University. - 28 Jun 2023. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2306.16601 (date of access: 09.03.2024).

7. Aslani T. V. LayerNorm: A key component in parameter-efficient fine-tuning [Text] / T. V. Aslani, H. Liang // Cornell University. - 29 Mar 2024. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2403.20284 (date of access: 09.03.2024)

8. Baan J. Understanding Multi-Head Attention in Abstractive Summarization [Text] / J. Baan, M. ter Hoeve, M. van der Wees, A. Schuth, M. de Rijke // Cornell University. - 10 Nov 2019. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1911.03898 (date of access: 09.03.2024)

9. Belcak P. Fast Feedforward Networks [Text] / P. Belcak, R. Wattenhofer // Cornell University. - 28 Aug 2023. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2308.14711 (date of access: 09.03.2024)

10. Cardoza C. Text analysis framework for understanding cyber-crimes [Text] / C. Cardoza, R. Wagh // International Journal of advanced and applied sciences. 2017/10/1. - P. 58-63. [Електронний ресурс] – Режим доступу : www URL:
https://scholar.google.co.in/citations?view_op=view_citation&hl=en&user=-XfWKOwAAAAJ&citation_for_view=-XfWKOwAAAAJ:W7OEmFMy1HYC. (date of access: 09.03.2024)

11. Chalkidis I. LEGAL-BERT: The Muppets straight out of Law School [Text] / I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras & I. Androutsopoulos // In Findings of the Association for Computational Linguistics: EMNLP 2020 - P. 2898–2904, Online. Association for Computational Linguistics. [Електронний ресурс] – Режим доступу : www URL: https://aclanthology.org/2020.findings-emnlp.261/(date of access: 09.03.2024)

12. Devlin J. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Text] / J. Devlin, M.-W. Chang, K. Lee, K. Toutanova // Cornell University. - 24 May 2019. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1810.04805(date of access: 09.03.2024)

13. Deepa A. Effective deep learning approaches for summarization of legal texts [Text] / A. Deepa, W. Rupali // Journal of King Saud University-Computer and Information Sciences. - 2022/5/1. - P. 2141-2150. [Електронний ресурс] – Режим доступу : www URL: https://scholar.google.co.in/citations?view_op=view_citation&hl=en&user=-XfWKOwAAAAJ&citation_for_view=-XfWKOwAAAAJ:8k81kl-MbHgC (date of access: 09.03.2024)

14. Dey R. Gate-variants of Gated Recurrent Unit (GRU) neural networks [Text] / R. Dey, F. M. Salem // IEEE Xplore. 60th International Midwest Symposium on Circuits and Systems (MWSCAS). - 06-09 August 2017. [Електронний ресурс] – Режим доступу : www URL: https://ieeexplore. ieee.org/abstract/document/8053243 (date of access: 09.03.2024)

15. DistilBERT : вебсайт. [Електронний ресурс] – Режим доступу : www URL: https://huggingface.co/docs/transformers/model_doc/distilbert (дата звернення: 09.03.2024)

16. Dogra V. Analyzing DistilBERT for Sentiment Classification of Banking Financial News [Text] / V. Dogra, A. Singh, S. Verma, Kavita, N. Z. Jhanjhi, M. N. Talib In: Peng SL., Hsieh SY., Gopalakrishnan S., Duraisamy B. (eds) Intelligent Computing and Innovation on Data Science. Lecture Notes in Networks and Systems, vol 248. Springer, Singapore, 2021. P. 501-510 [Електронний ресурс] – Режим доступу : www URL: https://link.springer.com/chapter/10.1007/978-981-16-3153-5_53(date of access: 09.03.2024)

17. Geron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow [Text] / A. Geron - O'Reilly Media. 2019. P. 129-134. [Електронний ресурс] – Режим доступу : www URL: http://14.139.161.31/OddSem-0822-1122/Hands-On_Machine_Learning_with_Scikit-Learn-Keras-and-TensorFlow-

2nd-Edition-Aurelien-Geron.pdf (date of access: 09.03.2024) ??

18. Graves A. Long Short-Term Memory [Text] / A. Graves // Supervised Sequence Labelling with Recurrent Neural Networks. - Vol. 385. - Springer-Verlag GmbH Berlin Heidelberg, 2012. [Електронний ресурс] – Режим доступу : www URL: https://link.springer.com/chapter/10.1007/978-3-642-24797-2_4 (date of access: 09.03.2024)

19. Grimmer J. Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts [Text] / J. Grimmer, B. M. Stewart // Cambridge University Press. - 04 January 2017. [Електронний ресурс] – Режим доступу : www URL: https://www.cambridge.org/core/journals/political-analysis/article/text-as-data-the-promise-and-pitfalls-of-automatic-content-analysis-methods-for-political-texts/F7AAC8B2909441603FEB25C156448F20 (date of access: 09.03.2024)

20. Guskin S. Dynamic-TinyBERT: Boost TinyBERT's Inference Efficiency by Dynamic Sequence Length [Text] / S. Guskin, M. Wasserblat, K. Ding, G. Kim // Cornell University. - 18 Nov 2021. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2111.09645 (date of access: 09.03.2024)

21. Ikonomakis M. Text Classification Using Machine Learning Techniques [Text] / M. Ikonomakis, S. Kotsiantis, V. Tampakas // WSEAS TRANSACTIONS on COMPUTERS. - Issue 8. - Volume 4. - August 2005. - P. 966-974/ [Електронний ресурс] – Режим доступу : www URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b0485fba23aabda526358f31cb5a382b66a08270 (date of access: 09.03.2024)

22. Jelodar H. Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey [Text] / H. Jelodar, Y. Wang, C. Yuan , X. Feng, X. Jiang, Y. Li, L. Zhao // Cornell University. - 6 Dec 2018. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1711.04305 (date of access: 09.03.2024)

23. Kanakaraj M. NLP based sentiment analysis on Twitter data using ensemble classifiers [Text] / M. Kanakaraj, R. M. R. Guddet // IEEE Xplore. 3rd

International Conference on Signal Processing, Communication and Networking (ICSCN). - 26-28 March 2015. - Chennai, India. [Електронний ресурс] – Режим доступу : www URL: https://ieeexplore.ieee.org/abstract/document/7219856 (date of access: 09.03.2024)

24. Kaur S. Causal Categorization of Mental Health Posts using Transformers [Text] / S. Kaur, R. Bhardwaj, A. Jain, M. Garg, C. // Cornell University. - 6 Jan 2023. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2301.02589 (дата звернення: 09.03.2024)

25. Khan M. T. Sentiment analysis and the complex natural language [Text] / M. T. Khan, M. Durrani, A. Ali, I. Inayat, S. Khalid & K. H. Khan // Complex Adaptive Systems Modeling. - Volume 4, article number 2, Published: 03 February 2016. [Електронний ресурс] – Режим доступу : www URL: https://link.springer.com/article/10.1186/s40294-016-0016-9. (date of access: 09.03.2024)

26. Kingma D. P. Adam: A Method for Stochastic Optimization [Text] / D. P. Kingma, J. Ba // Cornell University. - 22 Dec 2014. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1412.6980 (date of access: 09.03.2024)

27. Klingler N. Graph Neural Networks (GNNs) – 2024 Comprehensive Guide [Text] / N. Kligler // *viso.ai.* [Електронний ресурс] – Режим доступу : www URL: https://viso.ai/deep-learning/graph-neural-networks/ (date of access: 09.03.2024)

28. Knowledge Distillation from Bert in Pre-Training and Fine-Tuning for Polyphone Disambiguation [Text] / H. Sun, X. Tan, J. Gan, S. Zhao, D. Han, H. Liu, T. Qin, T. Liu // IEEE Xplore. Automatic Speech Recognition and Understanding Workshop (ASRU). - 14-18 December 2019. [Електронний ресурс] – Режим доступу : www URL: https://ieeexplore.ieee.org/abstract/document/9003918 (date of access: 09.03.2024)

29. Lu J., Henchion M., Bacher I., Namee B. M. A Sentence-level Hierarchical BERT Model for Document Classification with Limited Labelled Data

[Text] / J. Lu, M. Henchion, I. Bache, B. M. Namee // Cornell University. - 12 Jun 2021. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2106.06738 (date of access: 09.03.2024)

30. Mahoney C. J. Framework for Explainable Text Classification in Legal Document Review [Text] / C. J. Mahoney, J. Zhang, N. Huber-Fliflet, P. Gronvall, H. A. Zhao // IEEE International Conference on Big Data (Big Data). [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1912.09501 (date of access: 09.03.2024)

31. Minaee S. Deep Learning Based Text Classification: A Comprehensive Review [Text] / S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, J. Gao // Cornell University. - 6 Apr 2020. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2004.03705 (date of access: 09.03.2024)

32. Noguti M.Y. Legal Document Classification: An Application to Law Area Prediction of Petitions to Public Prosecution Service [Text] / M. Y. Noguti, E. Vellasques, L. S. Oliveira // Cornell University. - 13 Oct 2020. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2010.12533 (date of access: 09.03.2024)

33. Ostendorff M. Evaluating Document Representations for Content-based Legal Literature Recommendations [Text] / M. Ostendorff, E. Ash, T. Ruas, B. Gipp, J. Moreno-Schneide, G. Rehm // Conference: ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law. June 2021. [Електронний ресурс] – Режим доступу : www URL: https://www.researchgate.net/publication/353521029_Evaluating_document_representations_for_content-based_legal_literature_recommendations#:~:text=We%20note%20that%20prior%20work,AI%20research%20has%20focused (date of access: 09.03.2024)

34. Press O. Improving Transformer Models by Reordering their Sublayers [Text] / O. Press, N. A. Smith & O. Levy // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. July 2020. - P. 2996–3005.[Електронний ресурс] – Режим доступу : www URL:

https://aclanthology.org/2020.acl-main.270/(date of access: 09.03.2024)

35. Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey [Text] / B. Min, H. Ross, E. Sulem, A. P. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heinz, D. Roth // Cornell University. - 1 Nov 2021. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2111.01243(date of access: 09.03.2024)

36. RoBERTa: A Robustly Optimized BERT Pretraining Approach [Text] / Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov // Cornell University. - 26 Jul 2019. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1907.11692 (date of access: 09.03.2024)

37. Rogers A. Primer in BERTology: What we know about how BERT works [Text] / F. Rogers, O. Kovaleva, A. Rumshisky // Cornell University. - 27 Feb 2020 [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2002.12327 (date of access: 09.03.2024)

38. Salehinejad H. Recent Advances in Recurrent Neural Networks [Text] / H. Salehinejad, S. Sankar, J. Barfett, E. Colak, S. Valaee // Cornell University. - 29 Dec 2017. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1801.01078 (date of access: 09.03.2024)

39. Sanh V. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter [Text] / V. Sanh, L. Debut, J. Chaumond, T. Wolf // Cornell University. - 1 Mar 2020. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1910.01108 (date of access: 09.03.2024)

40. Shrikumar A. Learning Important Features Through Propagating Activation Differences [Text] / A. Shrikumar, P. Greenside, A. Kundaje // Cornell University. - 10 Apr 2017. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1704.02685 (date of access: 09.03.2024)

41. Singh S. Natural Language Processing for Information Extraction (Department of Computing, Faculty of Science and Engineering, Macquarie University, Australia) [Text] / S. Singh //*arXiv:1807.02383v1* [cs.CL]. - 6 Jul 2018.

- P. 1-24. [Електронний ресурс] – Режим доступу : www URL: https://www.researchgate.net/publication/326264437_Natural_Language_Processing_for_Information_Extraction (date of access: 09.03.2024)

42. Staliūnaitė I. Compositional and Lexical Semantics in RoBERTa, BERT and DistilBERT: A Case Study on CoQA [Text] / I. Staliūnaitė, I. Iacobacci // Cornell University. - 17 Sep 2020. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/2009.08257 (date of access: 09.03.2024)

43. Structured information extraction from complex scientific text with fine-tuned large language models [Text] / A. Dunn, J. Dagdelen, N. Walker, S. Lee, A. S. Rosen, G. Ceder, K. Persson, A. Jain // Cornell University. - 10 Dec 2022.. [Електронний ресурс] – Режим доступу : www URL: https://ar5iv.labs.arxiv.org/html/2212.05238 (date of access: 09.03.2024)

44. Taye M. M. Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions [Text] / M. M. Taye // Data Science and Artificial Intelligence, Philadelphia University, Amman 19392, Jordan. MDPI 25 April 2023. [Електронний ресурс] – Режим доступу : www URL: https://www.mdpi.com/2073-431X/12/5/91 (date of access: 09.03.2024)

45. Trends in Deep Learning Methodologies. Algorithms, Applications, and Systems. A volume in Hybrid Computational Intelligence for Pattern Analysis [Text] / Editors: V. Piuri, S. Raj, A. Genovese, R. Srivastava. - Academic Press, 2021 // ScienceDirect. [Електронний ресурс] – Режим доступу : www URL: https://doi.org/10.1016/C2019-0-04635-3 (date of access: 09.03.2024)

46. Voita E., Talbot D., Moiseev F., Sennrich R., Titov I. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned [Text] / E. Voita, D. Talbot, F. Moiseev, R. Sennrich, I. Titov // Cornell University. - 23 May 2019. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1905.09418 (date of access: 09.03.2024)

47. Wagh R. S. Knowledge Discovery from Legal Documents Dataset using Text Mining Techniques [Text] / R. S. Wagh // International Journal of Computer Applications (0975 – 8887). - Volume 66. - № 23, 2013. - P. 32–34. [Електронний

ресурс] – Режим доступу : www URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=f13c319356249b5f05062424de3d60d0366238b1 (date of access: 09.03.2024)

48. Wu H. Aspect-based Opinion Summarization with Convolutional Neural Networks [Text] / H. Wu, Y. Gu, S. Sun, X. Gu // Cornell University. - 30 Nov 2015]. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1511.09128 (date of access: 09.03.2024)

49. Hao Y. Visualizing and Understanding the Effectiveness of BERT [Text] / Y. Hao, L. Dong, F. Wei, K. Xu // Cornell University. - 15 Aug 2019. [Електронний ресурс] – Режим доступу : www URL: https://arxiv.org/abs/1908.05620 (date of access: 09.03.2024)

50. Фесенко Т.Г., Рибалов О.О., Хрусталев Є.К., Снігур А.Р. Методи і засоби інтелектуального аналізу текстових документів: бібліометричне дослідження. Збірник наукових праць. – Полтава: ПНТУ, 2024. Т. 3 (77). С. 148-151.