Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра _____Інформаційних управляючих систем_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

Тема роботи Дослідження методів анотування з використанням великих мовних моделей в ІТ-проєктах електронного архівування
(тема)

Виконав:
студент _2_ курсу, групи _УПГІТм-22-2____
_____Шкамета Олексій Сергійович_____
(прізвище, ім'я, по батькові)
Спеціальність ____122 Комп'ютерні_____
_____науки_____
(код і повна назва спеціальності)
Тип програми ____освітньо-наукова_____
(освітньо-професійна або освітньо-наукова)
Освітня програма __Управління проєктами в галузі інформаційних технологій_____
(повна назва освітньої програми)

Керівник ___проф. каф. ІУС Оксана ЧАЛА___
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри _____ ___Костянтин ПЕТРОВ___
(підпис) (власне ім'я, прізвище)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____

Кафедра _____Інформаційних управляючих систем_____

Рівень вищої освіти _____другий (магістерський)_____

Спеціальність _____122 Комп'ютерні науки_____
(код і повна назва)

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

Освітня програма_Управління проєктами в галузі інформаційних технологій
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_01__»__квітня__ 20 24_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____Шкаметі Олексію Сергійовичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів анотування з використанням великих мовних моделей в IT-проєктах електронного архівування

затверджена наказом університету від _01 квітня_2024_ р. № _258Cm__

2. Термін подання студентом роботи до екзаменаційної комісії __01_ __06_____ 2024 р.

3. Вихідні дані до роботи _Інформація щодо стану технологій у обраній предметній галузі; аналітичні та наукові статі будови нейронних мереж; типології нейронних мереж та деталі використання обраних нейронних мереж у обраній предметній галузі_____

4. Перелік питань, що потрібно опрацювати в роботі __дослідити предметну галузь анотування текстів; дослідити нейронні мережі як технологію; дослідити застосування____ нейромереж для анотування текстів; встановити задачу дослідження; сформулювати метод та модель вирішення для поставленої задачі дослідження; сформувати план реалізації експериментального методу для рішення поставленої задачі; дослідження для_ розробки, розробка, експериментальна перевірка методу для вирішення поставленої задачі_____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни	Примітка
1	Аналіз предметної галузі	01.04.2024-15.04.2024	Виконано
2	Дослідження технологій предметної галузі	01.04.2024-15.04.2024	Виконано
3	Постановка задачі дослідження	15.04.2024	Виконано
4	Теоретичне дослідження обраних технологій	16.04.2024-21.04.2024	Виконано
5	Практичне дослідження обраний технологій	22.04.2024-30.04.2024	Виконано
6	Розробка тестового прототипу	22.04.2024-08.05.2024	Виконано
7	Аналіз недоліків та переваг тестового прототипу	01.05.2024-08.05.2024	Виконано
8	Моделювання висновків за виконаною роботою	08.05.2024-16.05.2024	Виконано
9	Написання пояснювальної записки	01.05.2024-28.05.2024	Виконано
10	Захист кваліфікаційної роботи в екзаменаційній комісії	04.06.2024	Виконано

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ проф. каф. ІУС Оксана ЧАЛА
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Кваліфікаційна робота містить: 85 с., 44 рис., 11 табл., 17 фор., 1 дод, 35 джерела.

АНОТУВАННЯ, АРХІВУВАННЯ ІТ-ПРОЄКТІВ, ВЕЛИКІ МОВНІ МОДЕЛІ, ІНФОРМАЦІЙНА СИСТЕМА, НЕЙРОМЕРЕЖА, СИСТЕМА ОБРОБКИ ПРИРОДНЬОЇ МОВИ, УПРАВЛІННЯ ПРОЄКТАМИ, ПРОЄКТУВАННЯ СИСТЕМИ, BERT.

Об’єкт дослідження – автоматичне анотування у сфері електронного архівування ІТ-проєктів за допомогою великих мовних моделей.

Мета роботи: дослідження актуальності використання великих мовних моделей в сфері електронного архівування ІТ-проєктів, планування та пропонування практичної реалізації для покращення ефективності електронного архівування ІТ-проєктів.

Методи дослідження: вивчення сфери анотування тексту, вивчення принципу роботи нейронних мереж в сфері обробки природньої мови, аналіз існуючих великих мовних моделей для обробки природньої мови, формування проблеми дослідження, розробка запитів до нейромережі для автоматичного анотування в сфері електронного архівування ІТ-проєктів, налаштування та навчання великої мовної моделі, планування проєкту імплементації рішення, формування проєкту, опис основних етапів проєкту, розробка детального плану проєкту, опис інструментарію та навчальних даних, тренування та тестування прототипу великої мовної моделі на реальних даних зі сфери електронного архівування ІТ-проєктів, порівняння результатів прототипу з консервативними методами анотування тексту.

ABSTRACT

Thesis contains: 85 p., 44 fig., 11 tab., 17 for., 1 ann., 35 ref.

ANNOTATION, ARCHIVING IT PROJECTS, BERT, BIG LANGUAGE MODELS, INFORMATION SYSTEM, NEURAL NETWORK, NATURAL LANGUAGE PROCESSING SYSTEM, PROJECT MANAGEMENT, SYSTEM DESIGN.

The object of the research is the process of automatic annotation in the field of electronic archiving of IT projects using Big Language Models.

The aim of the study is to investigate the relevance of using Big Language Models in the field of electronic archiving of IT projects, planning and proposing an implementation to improve the efficiency of electronic archiving of IT projects.

The research methods are as follows: studying the field of text annotation, studying the principles of neural networks in the field of natural language processing, analyzing existing Big Language Models for natural language processing, formulating research problems, developing queries to the neural network for automatic annotation in the field of electronic archiving of IT projects, setting up, training the Big Language Model, planning the project implementation, forming the project basis, describing the main stages of the project, developing a detailed project plan, describing the tools and training data required for implementation, training, and testing the prototype of the Big Language Model on real data from the field of electronic archiving of IT projects, comparing the results of the prototype with traditional methods of text annotation.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

Ембеддінг – спосіб перетворення об’єкту на набір чисел та векторів

Енкодер – певна система, що кодує дані за даним ключем

Нейромережа, мережа – нейронна мережа

Промт – формулювання вимоги до нейромережі

Токен – одиниця ваги для певної одиниці даних в обробці нейромережі

Токенізація – процес аналізу даних мережею та присвоєння ваги

CPU – процесор

DB, Database – база даних (БД)

Dataset – датасет, набір даних для обробки

GPU – відеокарта

NLP, Natural Language Processing – обробка природньої мови

LMM, Large Language Model – велика мовна модель (ВММ)

ВСТУП

Електронне архівування — це процес зберігання документів, медіа-файлів та інших типів інформації у цифровому форматі, який відіграє ключову роль у сучасних інформаційних системах. Завдяки електронному архівуванню, організації мають можливість зберігати великі обсяги даних ефективно, забезпечуючи швидкий доступ, високу надійність збереження і гнучкість управління інформаційними ресурсами.

На сьогоднішній день, електронне архівування не обмежується простим зберіганням даних. Воно включає комплексні системи управління документами та контентом, інтегровані платформи для електронного урядування, освіти, наукових досліджень та бізнесу. Використання хмарних технологій, штучного інтелекту та машинного навчання значно розширило можливості електронних архівів, зокрема, у покращенні метаданих, оптимізації пошуку та автоматизації обробки даних. Однак цей процес супроводжується багатьма викликами, серед яких обробка великих датасетів, забезпечення якості метаданих, уникання людської помилки, гарантування безпеки і конфіденційності даних, а також розуміння семантики контенту. Ці виклики створюють багато можливостей для помилок, які можуть серйозно вплинути на ефективність і надійність систем електронного архівування.

Великі мовні моделі та нейромережі здатні автоматизувати створення анотацій, забезпечуючи високу якість метаданих. Це спрощує пошук і навігацію по архівованому контенту, допомагаючи швидко знаходити релевантну інформацію. Застосування нейронних мереж дозволяє глибше аналізувати контент, покращуючи розуміння семантики і контексту текстів, відео та інших медіа файлів. Це веде до більш ефективного семантичного пошуку, де системи можуть виявляти інформацію, базуючись на змісті та ідеях, а не лише на

ключових словах. Завдяки цим перевагам, великі мовні моделі та нейромережі змінюють сферу електронного архівування, вносячи інновації у створення метаданих, пошукові механізми та загальне управління архівними даними.

Методи дослідження: вивчення сфери анотування тексту, вивчення принципу роботи нейронних мереж в сфері обробки природньої мови, аналіз існуючих великих мовних моделей для обробки природньої мови, формування проблеми дослідження, розробка запитів до нейромережі для автоматичного анотування в сфері електронного архівування ІТ-проєктів, налаштування та навчання великої мовної моделі, планування проєкту імплементації рішення, формування проєкту, опис основних етапів проєкту, розробка детального плану проєкту, опис інструментарію та навчальних даних, тренування та тестування прототипу великої мовної моделі на реальних даних зі сфери електронного архівування ІТ-проєктів, порівняння результатів прототипу з консервативними методами анотування тексту.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ЕЛЕКТРОННОГО АРХІВУВАННЯ ТА НЕЙРОННИХ МЕРЕЖ ДЛЯ ОБРОБКИ ТЕКСТУ

Анотування тексту — це процес створення короткого, стислого опису вмісту документа або текстового фрагменту, що узагальнює основні ідеї та ключові точки цього документу.

Анотування тексту відіграє важливу роль у різноманітних областях, зокрема в академічних дослідженнях, видавництві, юридичних послугах, нотаріальних послугах, архівуванні, документоведенні, управлінні інформацією, спрощуючи пошук інформації та допомагаючи читачам швидко зорієнтуватись у великих обсягах тексту.

1.1 Дослідження процесу анотування в методах електронного архівування

Основна мета анотування полягає в тому, щоб надати читачеві чітке уявлення про зміст тексту без необхідності його повного прочитання. Анотації дозволяють виявляти релевантність матеріалу до поточних інформаційних потреб, економлять час, і полегшують процес відбору та аналізу текстів. В академічному контексті анотації сприяють більш ефективному веденню досліджень, дозволяючи науковцям швидко оцінювати зміст статей та вибирати найбільш підходящі для своїх потреб.

Вимоги до анотування:

– скороченість та виразність. Анотація повинна бути стислою, але достатньо інформативною, щоб передати основну тему та ключові аспекти тексту. Вона має включати всю важливу інформацію, але уникати зайвих деталей;

– об'єктивність. Анотації мають бути нейтральними, представляючи інформацію без особистісних суджень або оцінок, якщо тільки це не критичний огляд;

– ясність. Мова анотації має бути зрозумілою і доступною, уникати фахового жаргону, якщо він не є необхідним для розуміння тексту;

– зосередженість на ключових елементах. Анотація повинна висвітлювати основні аргументи, ідеї, теми, результати, що пропонуються у тексті;

– адаптивність. Залежно від контексту та цільової аудиторії, стиль та детальність анотації можуть варіюватися [1].

Автоматичне анотування, зокрема з використанням штучного інтелекту та машинного навчання спрямоване на максимальне спрощення процесу створення анотацій. Такі системи прагнуть точно і ефективно відтворювати вміння людських анотаторів, автоматизуючи процес узагальнення та висновку ключових понять з текстів різного типу.

Анотування тексту є фундаментальним аспектом обробки природної мови, яке забезпечує глибше розуміння структури та семантики текстових даних. Цей процес включає присвоєння метаданих або міток різним частинам тексту, таким як слова, фрази, або цілі речення, що дозволяє машинам ефективно інтерпретувати і аналізувати людську мову[2].

Використання анотацій є критично важливим у багатьох задачах NLP, включаючи семантичний аналіз, витягування інформації, автоматичне резюмування текстів, і багато іншого.

Дослідження способів анотування за основним ресурсом:

Таблиця 1.1 – Дослідження способів анотування за основним ресурсом

Ручне анотування	На ранніх етапах розвитку обробки природної мови основним методом анотування було ручне анотування, яке вимагає значної кількості часу та зусиль від людських анотаторів. Цей підхід гарантує високу якість анотацій завдяки детальному розумінню контексту та нюансів мови, що є критично важливим у дослідженні та навчанні нейронних мереж[3]. Однак, ручне анотування є дуже трудомістким і не завжди можливим на велику шкалу, особливо коли мова йде про великі датасети
Автоматичне анотування	З розвитком машинного навчання та нейронних мереж з'явилася можливість автоматичного анотування тексту. Цей метод використовує алгоритми класифікації, регресії та інші статистичні методи для визначення відповідних міток для різних частин тексту. Автоматичне анотування значно знижує трудові та часові витрати і дозволяє обробляти великі обсяги даних. Однак, цей метод може включати помилки через обмеження алгоритмів і якість використовуваних тренувальних даних
Півавтоматичне анотування	Є компромісом між ручним і повністю автоматичним методами. Цей підхід використовує машинне навчання для генерації первинних анотацій, які потім переглядаються та коригуються людськими анотаторами. Це забезпечує більш високу точність анотацій, зменшуючи при цьому час та витрати порівняно з повністю ручним анотуванням
Використання краудсорсингу	Краудсорсинг став популярним методом анотування, який включає залучення великої кількості людей через Інтернет для виконання анотування в масштабі. Платформи як Amazon Mechanical Turk дозволяють дослідникам швидко зібрати анотовані дані за помірну плату. Хоча краудсорсинг може забезпечити швидке анотування великих датасетів, контроль якості залишається важливим питанням, оскільки анотації від непрофесійних анотаторів можуть бути менш надійними[3][6]

Таблиця 1.2 – Дослідження способів анотування за методом роботи

Семантичне анотування	Семантичне анотування зосереджується на ідентифікації та визначенні значення слова або фрази в контексті. Це включає в себе анотування сутностей, семантичних ролей, і взаємозв'язків між сутностями. Наприклад, в реченні «Шевченко написав Гайдамаки» слово «Шевченко» може бути анотоване як «автор», а «Гайдамаки» – як «твір»
Синтаксичне анотування	Синтаксичне анотування включає в себе аналіз структури речення, включаючи анотування частин мови, залежностей між словами та фразами. Цей тип анотації допомагає визначити граматичну структуру речення і забезпечує базу для багатьох синтаксичних аналізаторів
Прагматичне анотування	Прагматичне анотування відноситься до визначення інтенцій, цілей і комунікативних функцій в межах тексту. Це включає в себе аналіз дискурсу, структури діалогів, ввічливості, стилів викладу, і може знадобитися для аналізу розмовних систем та чат-ботів [4]. Наприклад, в розмові прагматичне анотування може вказувати на тональність (іронічну, серйозну), інтенцію (запитання, відповідь, команда) та рівень формальності
Морфологічне анотування	Морфологічне анотування фокусується на внутрішній структурі слів, визначаючи корені, префікси, суфікси, закінчення, і інші морфеми. Цей тип анотації допомагає в аналізі гнучкості та словотворення, що є важливим для багатьох мов з багатою морфологією, таких як українська, російська або арабська. Такий аналіз є критичним для коректного розбору слів і визначення їхніх базових форм, що підвищує якість машинного перекладу та інших мовних процесів
Анотування дискурсу	Анотування дискурсу включає в себе аналіз зв'язків і структури між реченнями та абзацами. Це може включати виявлення когезійних пристроїв, таких як зворотні займенники, еліпсиси, і лексичні зв'язки. Анотація дискурсу допомагає зрозуміти, як інформація організована в тексті, і як це впливає на загальне розуміння контенту. Цей тип анотації є ключовим для автоматичного резюмування та визначення структури тексту

Таблиця 1.3 – Дослідження інструментарію для анотування

Анотування з використанням машинного навчання	Сучасні методи анотування часто використовують машинне навчання, особливо навчання з підкріпленням і глибоке навчання, для автоматизації процесу анотування. Алгоритми машинного навчання можуть аналізувати великі набори даних і виявляти складні закономірності для генерації анотацій. Такі методи зазвичай використовують попередньо навчені моделі на великих корпусах тексту для визначення потрібних міток. Наприклад, використання трансформерів, таких як BERT або GPT, дозволило реалізувати ефективне семантичне анотування, враховуючи контекстуальні залежності у тексті. Ці моделі можуть бути доадаптовані (fine-tuned) для конкретних задач анотування, що забезпечує високу точність і гнучкість у використанні
Анотування з використанням правил	Анотування, базоване на правилах, використовує набори лінгвістичних правил для ідентифікації структур або паттернів у тексті. Цей підхід може бути особливо ефективним для морфологічного або синтаксичного анотування, де чітко визначені правила граматики можуть бути застосовані для аналізу тексту. Наприклад, правила можуть включати паттерни для визначення частин мови або для розбору залежностей між словами у складних реченнях
Комбіновані методи	Комбіновані методи анотування використовують переваги як машинного навчання, так і правил-базованих підходів. Це дозволяє збалансувати швидкість і масштабність автоматичного анотування з точністю і надійністю ручного чи правил-базованого анотування [5]. Комбіновані методи часто використовуються у складних задачах, де потрібно забезпечити високу якість анотацій, але також і високу продуктивність обробки

Анотування тексту стикається з рядом викликів, особливо у контексті обробки багатомовного та багатомодального контенту. великі мовні моделі пропонують обіцянки у покращенні анотування завдяки своїй здатності враховувати широкі контексти та великі набори даних та поєднувати декілька

типів роботи з текстом, але вони також потребують значних обчислювальних ресурсів і можуть виявляти упередження, вбудовані у тренувальні дані.

Водночас, розвиток технологій, таких як трансформери та методи глибокого навчання, відкриває нові можливості для поліпшення анотування тексту. Адаптація моделей під конкретні лінгвістичні та культурні контексти, збільшення точності анотацій та розширення можливостей автоматизації є ключовими напрямками майбутнього розвитку. Саме нейронні мережі та великі мовні моделі здатні принести максимальну ефективність з використанням усіх існуючих ресурсів, методів роботи та інструментів для анотування.

1.2 Дослідження підходів до використання великих мовних моделей в задачах обробки та анотування тексту

Нейронні мережі - це математичні моделі, що імітують роботу людського мозку для вирішення складних задач.

В контексті обробки тексту, нейронні мережі використовуються для розуміння, генерації та аналізу природної мови [7]. Вони можуть розпізнавати закономірності, залежності між словами, фразами, абзацами та документами. Останніми роками нейронні мережі відіграли ключову роль у трансформації підходів до обробки природної мови (Natural Language Processing, NLP).

Рисунок 1.1 – Задачі нейронних мереж у сфері NLP

Завдяки прогресу в області машинного навчання та збільшенню обчислювальних потужностей стало можливим розробляти глибокі нейронні мережі, які здатні обробляти великі об'єми тексту, вивчати контекстуальні зв'язки та генерувати людино-подібні відповіді.

Основні архітектури нейронних мереж наступні:

Таблиця 1.4 – Основні архітектури нейронних мереж [9]

Рекурентні нейронні мережі (RNN)	Основна ідея полягає у використанні інформації з попередніх кроків для обробки наступних. Це допомагає моделям підтримувати контекст у послідовних даних[11]
Мережі з довгою короткочасною пам'яттю (LSTM)	Різновид RNN, які краще справляються з проблемою затухання градієнту, зберігаючи інформацію на значно більші проміжки часу
Мережі з контрольованим забуттям (GRU)	Спрощена версія LSTM, яка використовує меншу кількість параметрів і часто показує подібні результати
Трансформери (Transformers)	Використовують механізм уваги, що дозволяє моделям зосереджуватися на значущих частинах вхідних даних без необхідності обробляти інформацію послідовно. Це покращує здатність моделі до паралелізму і робить її ефективнішою

З великими мовними моделями на чолі, нейронні мережі застосовуються в широкому спектрі областей обробки природної мови, демонструючи вражаючі результати у генерації тексту, машинному перекладі, аналізі емоційного забарвлення текстів та багатьох інших задачах.

Ці застосування зміцнили позиції нейронних мереж як надійного інструмента для вирішення складних завдань NLP. Пізніше ці ж самі позиції було підкріплені великими показниками точності роботи та ефективності використання. А, враховуючи кількість даних (датасетів) задача глибокого навчання нейронних мереж втратила певну складність.

Використання нейронних мереж у обробці природної мови дозволяє ефективно обробляти великі об'єми тексту, виявляючи складні залежності та закономірності, які неможливо виявити традиційними методами. Завдяки здатності моделей розуміти контекст на двонаправленому рівні, якість обробки тексту значно покращується, що допомагає в розумінні нюансів мови, читанні контексту, перекладах, системах пошуку, розпізнаванні мовлення та генерації природнього тексту.

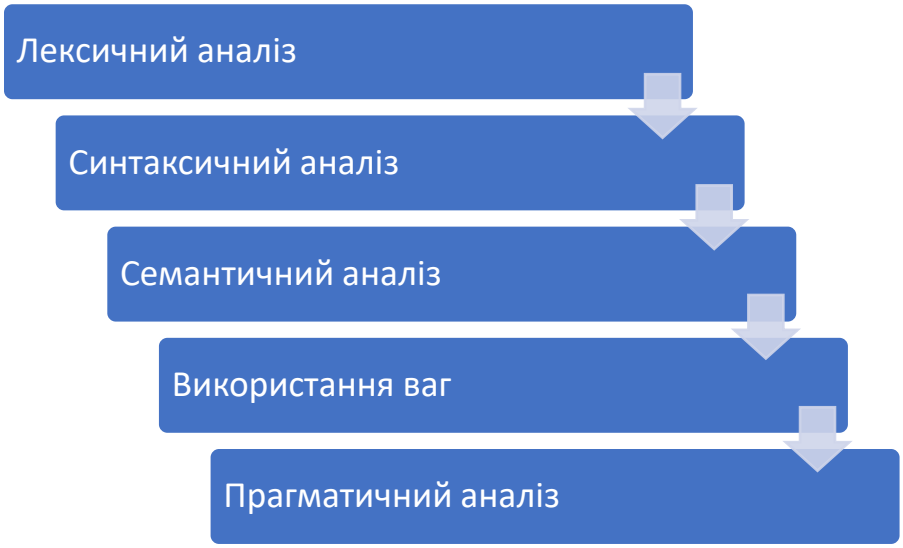

Рисунок 1.2 – Процес аналізу тексту нейромережею у сфері NLP

Таблиця 1.5 – Основні задачі нейромереж у сфері NPL [2][10]

Генерація тексту	Нейронні мережі використовуються для автоматичного створення статей, описів продуктів, резюме новин. GPT-3.5 або Gemini може генерувати високоякісні текстові відповіді на основі заданого запиту, включаючи виготовлення відповідей та завершення незакінчених речень [8][18]
Переклад	Нейронні мережі, особливо трансформери, забезпечують автоматичний переклад тексту без використання правил-посередників, що значно покращує якість перекладу. Моделі, такі як Google Neural Machine Translation, демонструють передові результати в перекладі
Розпізнавання мовлення	Моделі на кшталт WaveNet і DeepSpeech використовують нейронні мережі для перетворення аудіо в текст, що є основою для систем розпізнавання мовлення і допомагає у створенні точних систем автоматичного розпізнавання мовлення [18]
Сентимент-аналіз	Ці моделі аналізують емоційне забарвлення тексту, допомагаючи бізнесам аналізувати відгуки клієнтів. Моделі як BERT і RoBERTa показали високу точність у визначенні позитивних, нейтральних, та негативних відгуків
Витягування інформації	Нейронні мережі використовуються для ідентифікації іменованих сутностей, витягування відносин між сутностями, допомагаючи в аналізі юридичних документів, наукових статей тощо. Моделі як spaCy і Stanford NER є хорошими прикладами застосування нейронних мереж у цій області
Автоматичне резюмування	T5 і BART використовуються для створення коротких оглядів довгих текстів, що дозволяє користувачам швидко ознайомитися з основним змістом
Визначення інтенцій	Використовується у чат-ботах та системах інтерактивного обслуговування для розуміння запитів користувачів і надання відповідей. Моделі як Rasa і Dialogflow використовують нейронні мережі для визначення інтенцій і керування діалогом
Детектування фейкових новин	Системи, базовані на нейронних мережах, аналізують новинні статті, визначаючи їхню достовірність і допомагаючи боротися з поширенням дезінформації. Моделі як Grover і BERT є ефективними у виявленні недостовірного контенту
Оптимізація пошукових систем	Трансформери як тип нейронних мереж використовуються для покращення якості пошукових запитів, дозволяючи системам краще розуміти контекст запитів користувачів та надавати більш релевантні результати

Ці моделі можуть бути застосовані у широкому спектрі задач, від генерації тексту та машинного перекладу до аналізу емоцій та автоматизованого резюмування, що робить їх універсальним інструментом для різноманітних застосувань у галузі NLP [9][13]. Така гнучкість і широкий спектр застосувань забезпечують значні переваги для дослідників та розробників, дозволяючи використовувати одну модель для вирішення різних задач без необхідності її значної перебудови чи зміни на глибокому (машинному) рівні.

Однак, використання нейронних мереж у обробці природної мови також має свої недоліки. великі мовні моделі вимагають значних обчислювальних ресурсів та часу для тренування, що може бути обмежувальним фактором для дослідників з обмеженими ресурсами. Глибокі нейронні мережі часто вважаються "чорними скриньками", оскільки складно зрозуміти, як модель прийшла до певних висновків, що ускладнює тлумачення результатів та виявлення причин помилок. Якість роботи моделі також сильно залежить від якості та обсягу тренувальних даних. Недостатньо репрезентативні дані можуть призвести до упереджень моделі та зниження її ефективності. Все це робить розробку та застосування нейронних мереж складним процесом, який вимагає ретельного підходу до підготовки даних, тренування моделей та аналізу їхньої поведінки.

1.3 Аналіз великих мовних моделей у обробці тексту, анотації тексту системах обробки природньої мови

Великі мовні моделі (ВММ) змінили пейзаж обробки природної мови (NLP), пропонуючи нові можливості для аналізу, генерації та розуміння тексту. Ці моделі, які зазвичай містять мільярди параметрів і тренуються на величезних

корпусах текстових даних, здатні розуміти та генерувати мову в спосіб, що імітує людське сприйняття та використання мови. Вони використовуються у широкому спектрі застосувань. Великі мовні моделі представляють величезний потенціал для трансформації традиційних підходів до електронного архівування, пропонуючи шляхи до більш ефективної, швидкої та інтелектуальної обробки текстових даних. Впровадження цих моделей в IT-проєкти не лише пришвидшує обробку і аналіз великих даних, але й відкриває нові можливості для підвищення якості бізнес-процесів, забезпечуючи високу адаптивність і точність в роботі з електронними архівами.

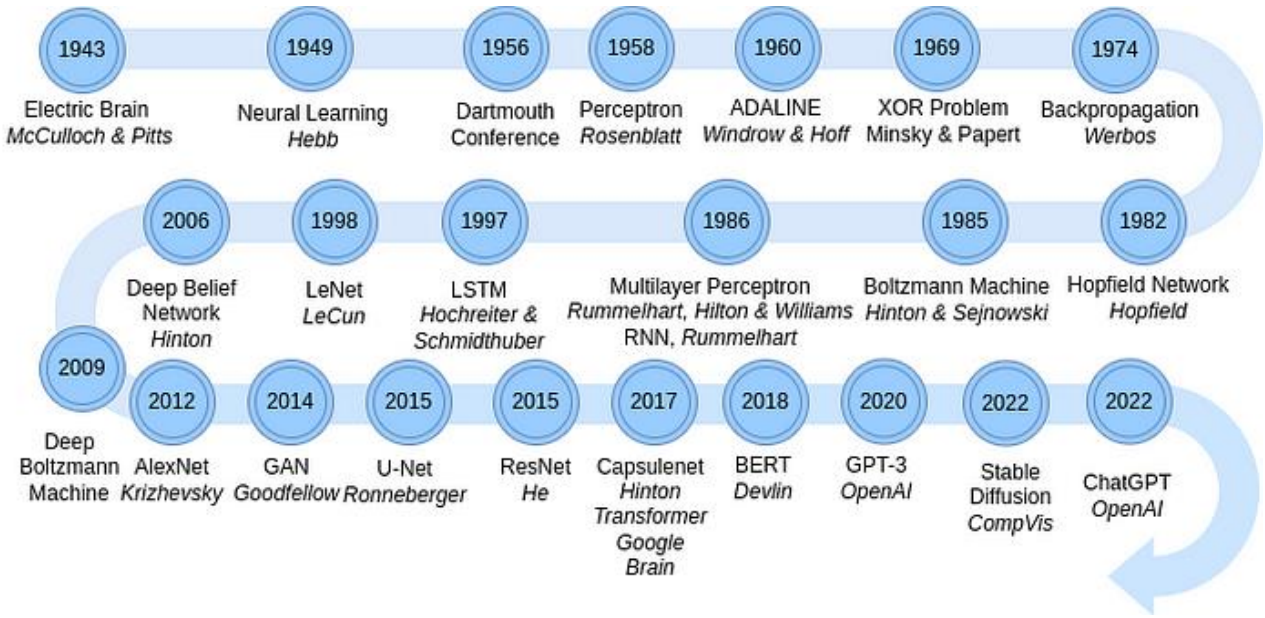

Рисунок 1.3 – Історія релізу нейронних мереж у сфері NLP [12]

Розвиток та інтеграція ВММ в такі проєкти може стати ключовим кроком на шляху до створення сучасних, інноваційних рішень в галузі електронного документообігу та архівування. Найвідомішими прикладами таких нейронних мереж є наступні витвори:

Таблиця 1.6 – Перелік нейронних мереж і особливостей у сфері NLP

BERT (Bidirectional Encoder Representations from Transformers)	Розроблений Google, цей метод використовує двонаправлену обробку тексту для генерації контекстуально залежних представлень слова. Він значно покращив якість рішень у задачах класифікації тексту, відповідей на питання та інших [28]
GPT (Generative Pre-trained Transformer)	Серія моделей від OpenAI, де GPT-4 стала однією з найбільших. Ці моделі використовуються для генерації тексту, створення контенту та багато іншого
RoBERTa (Robustly Optimized BERT Approach)	Варіація BERT, оптимізована зі зміною деяких методів передобробки і тренування, що покращила результати на декількох бенчмарках
T5 (Text-To-Text Transfer Transformer)	Розроблено Google, ця модель формулює всі задачі NLP як задачі перетворення тексту на текст, що дозволяє їй вирішувати різноманітні задачі з єдиного підходу
XLNet	Ця модель використовує перестановочну мовну модель для кращого вивчення контексту, що дозволяє обробляти текст у двонаправленому форматі без певних обмежень, які має BERT
ERNIE (Enhanced Representation through kNowledge Integration)	Розроблена Baidu, ця модель вводить знання з різних джерел для покращення розуміння тексту

Ці приклади лише відображають частину широкого спектру застосувань нейронних мереж у області обробки природної мови, і вони продовжують розвиватися, відкриваючи нові напрямки та можливості для дослідження і практичного використання.

Але варто зазначити, що у сфері електронного архівування ІТ-проєктів такі технології наразі не впроваджено, або впроваджено куцо (Confluence як приклад).

Електронне архівування ІТ-проєктів спирається на більш консервативні методи анотації [27], як от автоматична анотація за ключовими словами без урахування контексту, або взагалі ручна анотація.

Нейронні мережі відіграють важливу роль у сучасній обробці тексту, пропонуючи потужні інструменти для розуміння, генерації та аналізу природної мови. Незважаючи на певні недоліки, такі як великі обчислювальні витрати і складнощі у тлумаченні результатів, переваги використання цих технологій значно перевищують недоліки, відкриваючи нові можливості для досліджень і розвитку в області NLP.

1.4 Постановка задачі дослідження

Актуальність. Існуючі методи анотування даних в сфері електронного архівування ІТ-проєктів застарілі та\або потребують надто багато часу, щоб коефіцієнт часу на результативну одиницю міг дати ефективний результат. Також вони не враховують семантичний зміст електронних документів, тому використання таких методів пов'язано з помилками неточності анотування. Для вирішення задачі автоматичного анотування з урахуванням семантики дотично використати ВММ, тому що вони здатні поєднати у собі ефективність та швидкість, низку вартість у сенсі часу та грошей. Необхідність удосконалення у цій сфері є рушієм даного дослідження.

Об’єкт дослідження. Об’єктом дослідження є процес автоматичного анотування у сфері електронного архівування ІТ-проєктів.

Предмет дослідження. Предметом дослідження є великі мовні моделі (LLM) у сфері обробки природньої мови (NLP).

Мета роботи. Метою роботи є дослідження методів анотування з використанням великих мовних моделей в ІТ-проєктах електронного архівування.

Наукова новизна. Науковою новизною є удосконалений метод автоматичної анотації тексту шляхом впровадження великої мовної моделі у процес обробки даних під час їх завантаження на ресурс. Основним рушієм роботи ВММ є спеціально налаштований ембеддінг та датасет для навчання з промтом обробки «зробити анотацію тексту згідно контексту». Завдяки впровадженню нейромереж, метод також буде враховувати попередньо оброблені роботи, ключові слова, метадані та сам контекст, що призведе до постійного зростання точності та швидкості роботи з даними. Це, в свою чергу, дозволить бізнесу впроваджувати більш ефективне в сенсі часу та фінансів рішення, а науці – взаємодіяти з ресурсом задля цілей подальшого дослідження.

Практичні результати. Практичним результатом роботи є автоматично анотований при потраплянні до системи (БД) текст.

Задачі дослідження:

– аналіз процесу розробки та впровадження ВММ для специфічних задач;

– удосконалення методу автоматичного анотування в сфері електронного архівування ІТ-проєктів;

– планування розробки ВММ для автоматичного анотування в сфері електронного архівування ІТ-проєктів;

– експериментальна перевірка удосконаленого методу.

2 ФОРМУЛЮВАННЯ МОДЕЛІ ТА МЕТОДУ РІШЕННЯ ВММ В СФЕРІ ЕЛЕКТРОННОГО АРХІВУВАННЯ

Перш ніж приступи до формулювання моделі та методів, варто пріоритезувати задачі представити список основних етапів анотування тексту за допомогою великої мовної моделі (ВММ):

– аналіз сучасного стану електронного архівування, дослідження процесу архівування та нейронних мереж;

– формування з отриманих знань категорій IT-документації, формування промтів (запитів) до нейромережі відповідно до кожної з категорій, можливих в сфері архівування IT-проєктів;

– вибір та завантаження моделі. Визначити, яка ВММ (наприклад, BERT, GPT) найкраще підходить для задачі анотування; завантажити модель та відповідний токенізатор;

– підготовка даних для навчання та перевірки моделі. Зібрати та підготувати тренувальні, валідаційні та тестові датасети;

– тюнінг моделі. Налаштувати модель для виконання специфічних задач з даними. У нашому випадку – до автоматичного анотування IT-документації;

– тренування моделі. Тренувати попередньо налаштовану модель на підготовлених даних з використанням певних метрик для оцінки;

– тестування моделі. Протестувати модель на тестовому датасеті для визначення її ефективності;

– оптимізація та налаштування. Внести необхідні корективи для покращення точності та ефективності анотацій;

– розгортання моделі. Розгорнути модель для використання у продакшн середовищі, порівняти ефективність моделі з консервативними методами;

– моніторинг та оновлення (опціональний крок). Стежити за продуктивністю моделі та оновлювати її за потребою для підтримки точності та актуальності.

Проміж цього необхідно також планування дій за списком, підготовка та надання статистики з використаними ресурсами та робітниками. Подальший план дослідження виходить згідно цих пунктів.

2.1 Розробка запитів до ВММ

Для створення великої мовної моделі, яка здатна автоматично анотувати текст в сфері ІТ-проєктів, особливо у контексті електронного архівування, потрібно розробити низку промтів, які допоможуть у тренуванні та валідації моделі. Промти спрямовані на різні аспекти виявлення, аналізу та синтезу ключової інформації з текстів. Нижче наведено декілька загальних прикладів промтів для електронного архівування та їх переклади, так як більшість ІТ-проєктів використовує англійську, як мову з міжнародного стандартом:

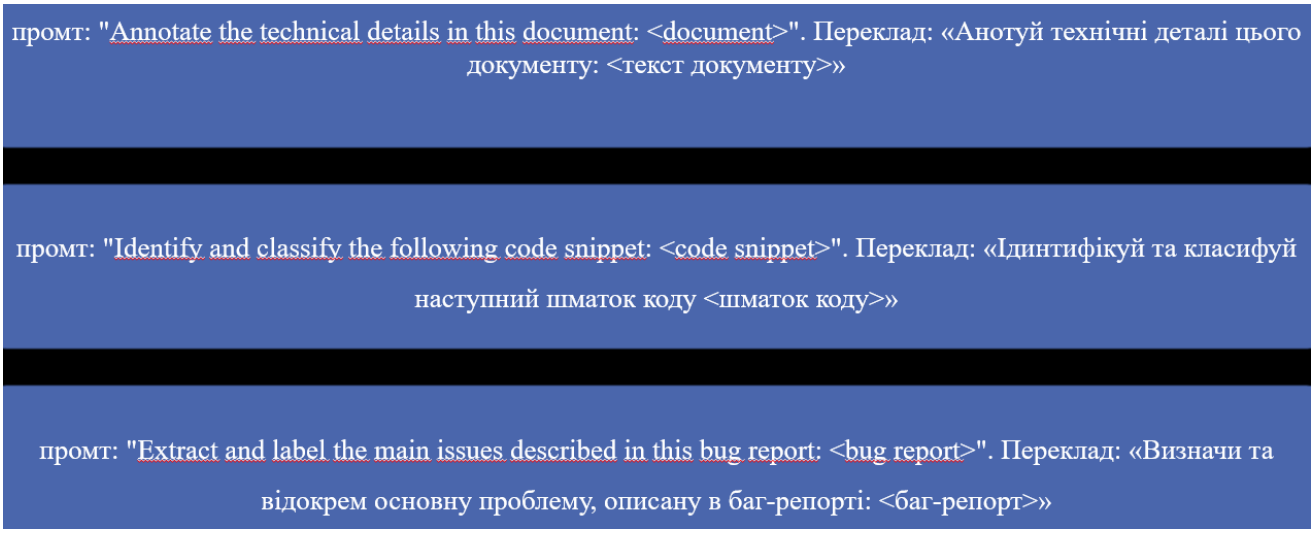

Рисунок 2.1 – Базові промти до ВММ

Але ці приклади є занадто узагальненими задля правильного навчання ВММ та подальшої її роботи з реальними документами електронного архівування ІТ-проєктів. Через те, що ІТ-проєкти часто містять складну і динамічну інформацію, ці промти стимулюють модель виділяти критично важливі дані [24], що підвищує якість архівування та сприяє кращому управлінню проєктною документацією. Використання добре структурованих промтів є вирішальним для успішної адаптації та інтеграції моделі в реальні системи, забезпечуючи користувачам зручність та ефективність при роботі з великим обсягом даних. Тому було прийнято рішення підвищити складність промтів та створити промт для кожного виду документа, що може зустрічатися в сфері електронного архівування ІТ-проєктів, створивши загальну формули промтів (формула 2.1):

$$P = (Q - C)^n - S, \tag{2.1}$$

де P – промт;

Q – основний об’єкт запиту нейромережі;

C – контекст або обставина;

n – кількість ітерацій;

S – специфікація.

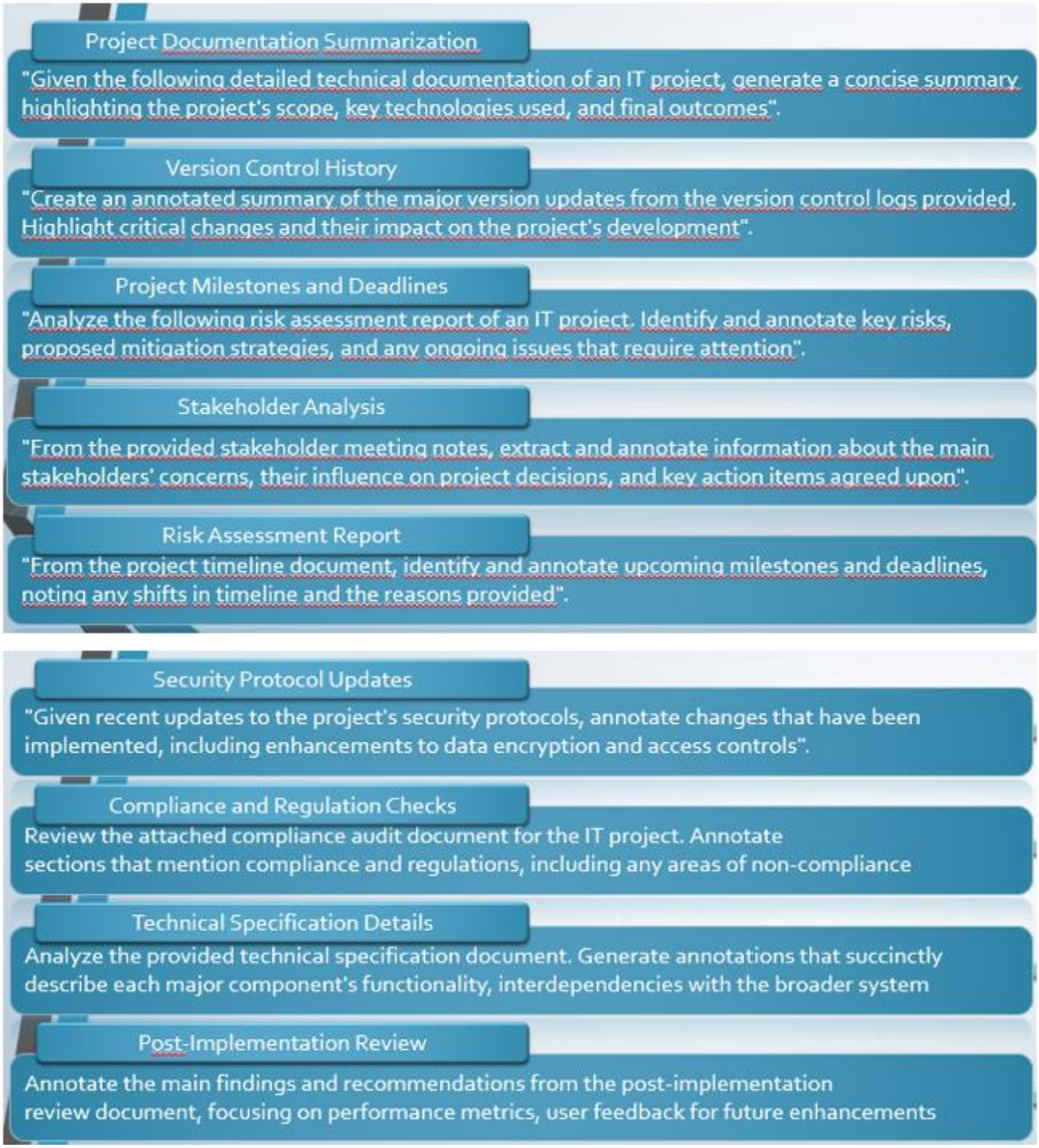

Рисунок 2.2 – Розширені категорії даних в сфері IT-архівування і відповідні промти для BMM

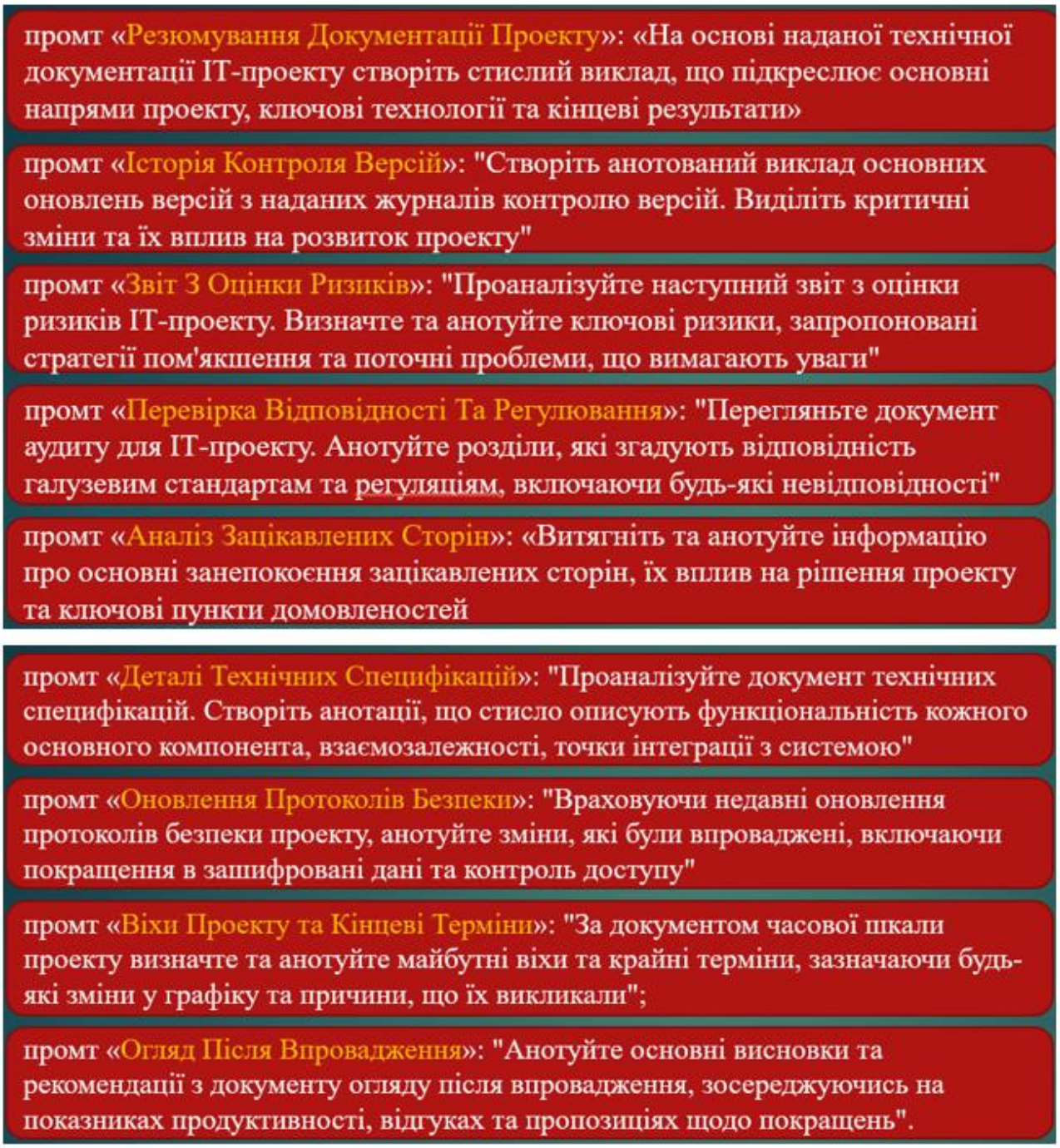

Рисунок 2.3 – Розширені категорії даних в сфері IT-архівування і відповідні промти для ВММ, переклад

З цієї формули можна сформувати розширені категорій та запитів до нейромережі для автоматичної анотації тексту в сфері електронного архівування ІТ-проєктів (рисунок 2.2, 2.3). Загалом, загальний принцип промту наступний:

– сформулювати категорію роботи анотування (баг-репорт, наприклад);

– запровадити контекст або умову для відсіювання інформації та акцентуванні (тільки коментарі, напр.);

– додати додаткову умову, що збільшить або зменшить анотацію (коментарі менеджера та розробників включно).

2.2 Метод анотування ІТ-документації щодо процесу розробки ІТ-проєктів с запитами до ВММ

Перед тим як приступити до розробки великої мовної моделі для автоматичного анотування тексту в сфері електронного архівування ІТ-проєктів, важливо зрозуміти, які методи та формули будуть найефективнішими для цієї задачі. Під час дослідження було розглянуто наступні аспекти.

Вибір типу ВММ - Трансформери проти RNN. Трансформери демонструють вражаючі результати в багатьох задачах NLP завдяки здатності обробляти довгі залежності та паралельну обробку тексту. Моделі як BERT, GPT-3, і RoBERTa мають глибокі архітектури, які ефективно вловлюють контекстуальні залежності у тексті. RNN/LSTM, в свою чергу, були стандартом до появи трансформерів, але їх послідовний характер обробки та проблеми з довготривалою залежністю роблять їх менш привабливими для комплексних задач анотування. Таким чином, трансформери є більш відповідним вибором для нашої задачі через їх здатність

ефективно обробляти великі обсяги тексту з різноманітними структурами і контекстами.

Вибір інструменту для word embeddings або ж векторного представлення слів. Pre-trained Embeddings: Моделі як Word2Vec, GloVe, і FastText пропонують загальні векторні представлення слів, які можуть бути корисними для загального розуміння мови. Contextual Embeddings: BERT і інші трансформери генерують контекстуалізовані embeddings, які змінюються залежно від контексту слова у тексті. Контекстуальні embeddings від трансформерів найкраще підходять для нашої задачі, оскільки вони забезпечують більш глибоке і точне розуміння семантики тексту, що є важливим, особливо – в домені IT.

Sequence Tagging (токенізація кожного слова) vs. Sequence-to-Sequence (токенізація послідовностей в залежності від контексту). Sequence Tagging: Методи як CRF або BiLSTM-CRF ефективні для маркування кожного слова або токена в тексті, що добре підходить для задач, де кожен елемент тексту має свій лейбл. Sequence-to-Sequence: Моделі, що використовують механізми attention і трансформери, добре підходять для генерації анотацій чи перекладу, коли весь вхідний текст перетворюється на вихідний. Залежно від структури анотацій, можна використовувати Sequence Tagging для детального маркування або Sequence-to-Sequence для генерації описових анотацій.

Вибір Формул. Функція Втрат: Cross-Entropy Loss (формула 2.2) є стандартною для класифікаційних задач і задач, де потрібно максимізувати правдоподібність правильних міток:

$$L = -\sum_{j}^{C} y_i \log(p_i) \tag{2.2}$$

CRF Loss може бути використана для sequence tagging, де враховуються залежності між послідовними лейблами. Cross-Entropy є оптимальним вибором для більшості задач, але для детального анотування можна використати CRF.

Attention Mechanism (формула 2.3):

$$Attention(Q, K, V) = softmax\left(\frac{QK^t}{\sqrt{dk}}\right)V \tag{2.3}$$

Ця формула дозволяє моделі фокусуватися на найважливіших частинах вхідних даних при генерації кожного слова виходу. Attention є критично важливим для трансформерів і використання його в моделі дозволить забезпечити більшу адаптивність і точність анотацій.

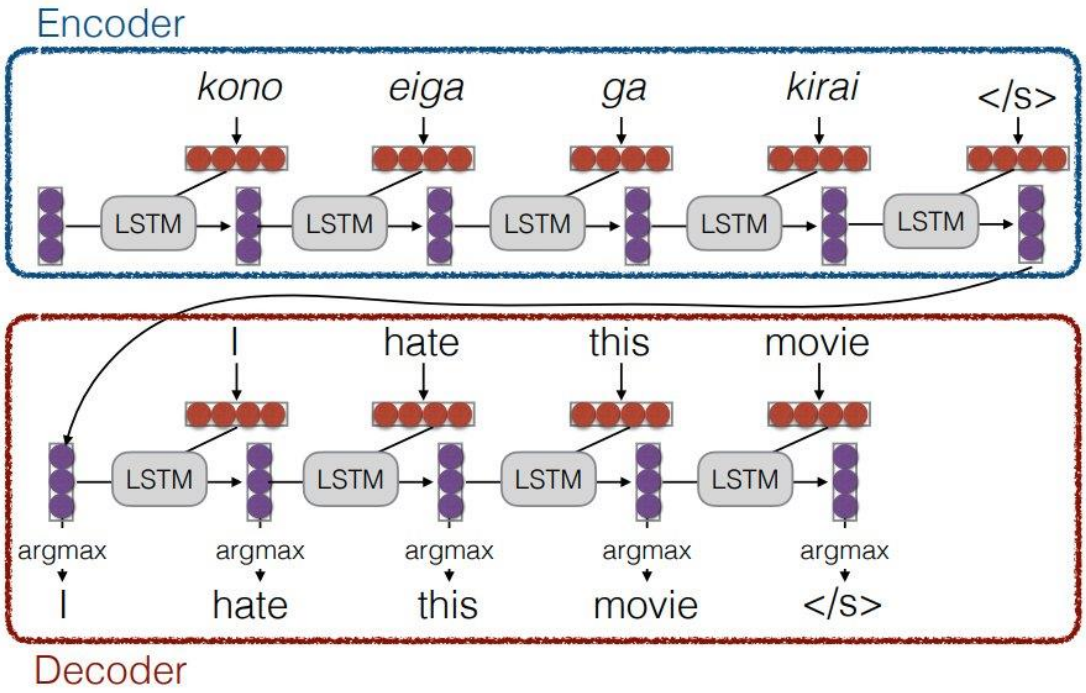

Рисунок 2.4 – Приклад роботи енкодера та декодера в двосторонній нейронній мережі для обробки природньої мови [14]

Після ретельного аналізу доступних технологій та методів було вибрано BERT (Bidirectional Encoder Representations from Transformers) як основу для анотаційної системи. З урахуванням попередніх факторів дослідження, архітектурою трансформера було обрано архітектуру з контекстуальними embeddings і потенційним застосуванням CRF для детального маркування, використовуючи Cross-Entropy Loss як основну функцію втрат і інтегруючи Attention для глибшого аналізу взаємозв'язків у тексті. Обґрунтування вибору великої мовної моделі наступне:

– ефективність BERT у розумінні контексту. BERT використовує механізм двонаправленого трансформера, який дозволяє моделі краще розуміти контекст з обох сторін токена одразу. Це критично важливо для розуміння технічної документації та коду, де значення елементів сильно залежить від контексту [15];

– гнучкість BERT для різних NLP задач. BERT може бути доадаптований (fine-tuned) для широкого спектру задач, включаючи класифікацію, анотацію, і відповіді на запитання. Це робить його ідеальним для створення системи анотування, яка може займатися різними видами тексту в IT проєктах;

– висока доступність та підтримка. Як одна з найпопулярніших моделей у сфері NLP, BERT має широку підтримку і багато варіантів, оптимізованих для специфічних задач (наприклад, CodeBERT для розуміння коду).

Таким чином, BERT не лише відкриває можливості для покращення якості та доступності технічної документації в IT-проєктах, але й забезпечує основу для розробки високоефективних, адаптивних і легко інтегрованих рішень для автоматичного анотування тексту на велику шкалу. BERT використовує архітектуру трансформера, що дозволяє моделі одночасно обробляти текст з обох боків кожного слова (звідси і назва "bidirectional") [15][16], надаючи можливість

кращого розуміння контексту слова в реченні. Це критично важливо для технічної документації та інших складних текстів, де зміст окремих слів може суттєво змінюватися в залежності від контексту. Використання BERT дозволяє ефективно адаптувати загальномовну модель до специфічних потреб анотування текстів в IT, які часто включають високоспеціалізовану лексику та технічні описи.

До того ж, BERT демонструє вражаючу здатність до "transfer learning", тобто можливість ефективно переносити знання, отримані на одних задачах, на інші, менш ресурсомісткі. Це робить його ідеальною основою для системи, що вимагає глибокого розуміння технічних текстів із обмеженим обсягом даних для тренування. Навчивши BERT на великій кількості загальнодоступних даних, можна ефективно доналаштувати його на відносно меншому наборі спеціалізованих текстів, що значно підвищує якість анотацій.

Базова підготовка та налаштування нейронної мережі. Перш за все найважливішим у налаштуванні BMM є підготовка, збір, та очищення даних – датасету. Підготовка даних для тренування включає збір, очищення та анотування великої кількості текстів з IT-проєктів. Це означає ретельний відбір релевантних документів, кодів, звітів про помилки та інших технічних матеріалів. Тексти мають бути очищені від нерелевантних форматувань та символів, а термінологія — уніфікована для забезпечення консистентності навчального процесу. Важливим аспектом є також анотування текстів, яке вимагає високої точності та знань специфіки домену, щоб забезпечити моделі вірні навчальні сигнали.

Після підготовки даних наступає фаза тренування, де модель BERT доналаштовується на специфічних даних з використанням технік, таких як оптимізатор AdamW, який допомагає в ефективному пошуку оптимальних ваг мережі. Важливою частиною процесу тренування є також використання методів,

які запобігають перенавчанню, таких як дропаут та пакетна нормалізація, а також динамічне керування швидкістю навчання для покращення збіжності.

Коли модель тренується та валідується на вибірках даних, важливо забезпечити її інтеграцію у виробниче середовище зі створенням зручних API для взаємодії з системами електронного архівування. Ці API дозволяють іншим застосункам та службам ефективно взаємодіяти з моделлю, відправляючи текст для анотування та отримуючи повернені анотації [9]. Безпека таких API, забезпечення стійкості до великих навантажень і швидке масштабування — ключові аспекти, які потрібно враховувати під час розробки.

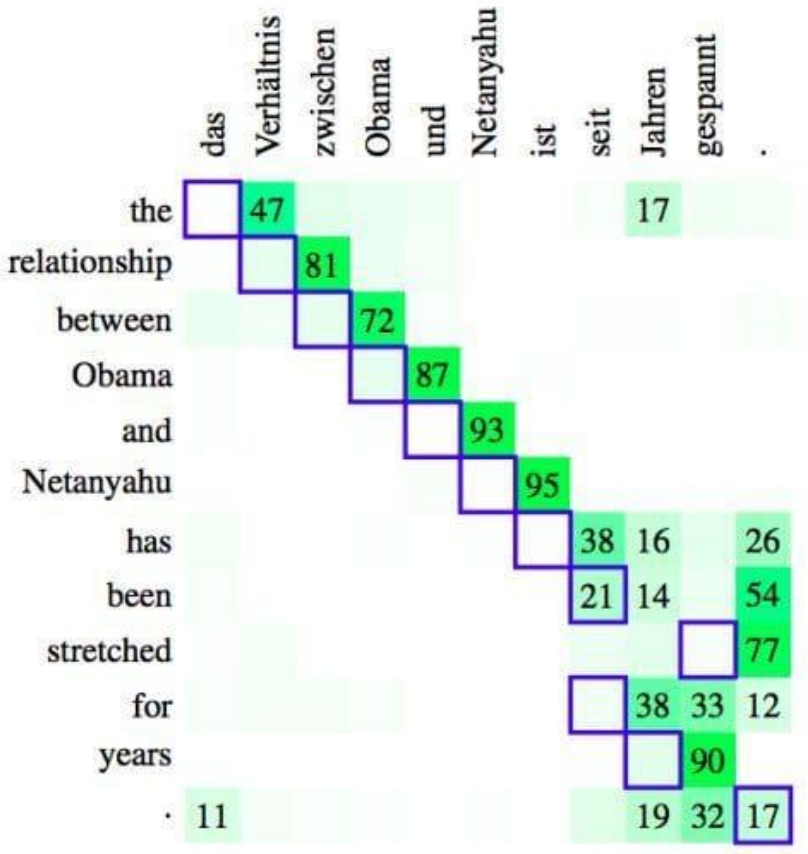

Рисунок 2.5 – Приклад розрахунку ваги нейромережею NLP

Окрім технічного впровадження, велике значення має моніторинг та підтримка системи. Використання інструментів для моніторингу дозволяє

постійно відслідковувати продуктивність моделі, вчасно виявляючи та коригуючи помилки [14]. Регулярні оновлення моделі, засновані на зворотньому зв'язку користувачів та аналізі продуктивності, забезпечують її актуальність та високу точність роботи.

2.3 Налаштування та навчання ВММ

Для створення системи автоматичної анотації текстів із застосуванням мовної моделі BERT, необхідно зрозуміти ключові аспекти її архітектури та взаємодії з текстовими даними. BERT (Bidirectional Encoder Representations from Transformers) використовує архітектуру трансформера, зокрема його енкодер, для витягування контекстно залежних ознак з тексту, що дозволяє ефективно працювати з різними завданнями обробки природної мови (NLP).

Для організації даних та обробки датасету (формула 2.4) використовується ієрархічна модель, яка може включати проєкти, модулі, версії, документацію тощо:

$$D = \{P_1|P_2 \dots, P_n\}, \tag{2.4}$$

де D – весь архів;

P_i – індивідуальні компоненти архіву.

Першим кроком в обробці тексту є його токенізація (формула 2.5). BERT використовує токенізатор WordPiece, який розбиває текст на атомарні одиниці, звані токенами. Це дозволяє моделі ефективно працювати з невеликим словником і знижує ризик не врахування рідкісних слів.

$$Token = Tokenizer(Text) \tag{2.5}$$

Також BERT водить спеціальні токени [CLS] на початок кожної послідовності та [SEP] для розділення різних частин тексту або кінця послідовності. Токен [CLS] використовується як агрегатор векторного представлення для класифікаційних завдань у процесі токенізації BERT (формула 2.6).

$$Tokens = ['[CLS]'] + Tokens + ['[SEP]'] \tag{2.6}$$

Далі кожен токен перетворюється на вектор за допомогою ембеддінг-шару. Ці ембеддінги ініціалізуються випадково і навчаються під час тренування моделі (формула 2.7).

$$E_i = Embeddings(t_i) \tag{2.7}$$

Оскільки трансформери не здатні самостійно визначати порядок слів у послідовності, BERT використовує позиційні ембеддінги (формула 2.8) для кодування порядкової інформації про токени.

$$PE_i = PositionalEncoding(i, d),$$

де i – індекс позиції токена;

d – розмірність ембеддінгу.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \tag{2.8}$$

Також BERT має в наявності та використовує сегментні ембеддінги (формула 2.9) Використовуються для розрізнення різних сегментів у складних завданнях, таких як відповіді на питання, де потрібно розрізнити запитання від відповіді.

$$S_i = SegmentEmbeddings(s_i), \tag{2.9}$$

, де s_i – індекс сегментації ваги

Далі кожен токен перетворюється в вектор за допомогою векторних представлень векторного представлення токена ti у BERT (формула 2.10), які включають в себе всі згадані ембеддінги:

$$E_i = TokenEmb(t_i) + PositionEmb(i) + SegmentEmb(i) \tag{2.10}$$

Механізм самоуваги (self-attention) є ключовим компонентом моделі BERT та трансформера загалом (формула 2.11). Він дозволяє моделі зосереджуватися на різних частинах вхідної послідовності під час визначення репрезентації кожного слова.

Розрахунок самоуваги: вхідні дані трансформуються у три матриці A саме: матриці запитів Q, матриці ключів K та значень V шляхом множення ембеддингів на вагові матриці, які є параметрами моделі:

$$\begin{aligned} Q &= HW^Q \\ Q &= HW^Q \\ Q &= HW^Q \end{aligned} \tag{2.11}$$

де H – вихідний вектор з попереднього шару;

W^Q – ваги матриці запитів;

W^K – ваги матриці ключів;

W^V – значень.

Функція розрахунку уваги (формула 2.12) використовується для визначення вагових коефіцієнтів для кожного слова в контексті інших слів:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.12)$$

де d_k – розмірність ключів.

BERT також використовує концепцію багатоголової уваги (формула 2.13), де механізм уваги застосовується паралельно в декількох "головах". Це дозволяє моделі одночасно звертати увагу на інформацію з різних представленських підпросторів на різних позиціях.

$$MultiHeadAttention(Q, K, V) = Concat(head_1, head_2, \dots head_n)W_o,$$

де $head_i$ розраховується як:

$$head_i = Attention\left(QW_{Q_i}, KW_{K_i}, VW_{v_i}\right), \quad (2.13)$$

де W_i^Q, W_i^K, W_i^V – вагові матриці для і-ї голови;

W^o – вихідна матриця ваг.

Після кожного шару уваги та повнозв’язного шару застосовуються нормалізація (формула 2.14) та залишкові зв'язки для стабілізації навчання та підвищення ефективності тренування (формула 2.15).

$$Norm(x) = \frac{x - \mu}{\sigma}\ , \quad (2.14)$$

де μ та σ – середнє значення і стандартне відхилення

$$H' = LayerNorm\big(H + MultiHead(Q, K, V)\big)$$
$$H'' = LayerNorm\big(H' + Feed(H')\big), \quad (2.15)$$

де Feed – це запит до нейромережі

На виході енкодера BERT використовується вектор, пов’язаний з токеном [CLS], для розв'язання задач класифікації. Цей вектор проходить через ще один повнозв’язний шар (зазвичай з лінійною активацією), що мапує його на простір міток (формула 2.16).

$$y = Dense(C), \quad (2.16)$$

де C – вектор, що відповідає токену [CLS];

y – вектор для кожної мітки.

Систематичний підхід до кожного етапу важливий для розробки ефективної системи NLP, яка може точно класифікувати та анотувати текстовий контент. Оптимізація навчання проходить через оптимізатор вибірки, що помічає надлишкові дані, гіпертрофовані дані та дані, що надто сильно відходять від статистики (формула 2.17).

$$\theta = \theta - \tau * Optimizer(\nabla_\theta \tau), \quad (2.17)$$

де θ – параметри моделі;

τ – швидкість навчання.

2.4 Висновки формування та формулювання моделі та метода анотування ІТ-документації

Висновок щодо створення та використання автоматичної мовної моделі для анотації текстів у сфері ІТ-проєктів електронного архівування полягає в наступному:

Специфічність домену. Завдання автоматичної анотації в ІТ-проєктах вимагає від моделі глибокого розуміння специфічної термінології та процесів. Це включає знання про контроль версій, стандарти безпеки, технічні специфікації та взаємозв’язки в проєктних документах.

Тренування моделі. Ефективне тренування моделі вимагає ретельно підготовлених даних та чітко сформульованих промтів, які направляють модель на ідентифікацію і анотацію ключових елементів тексту. Підготовка даних повинна включати різноманітні джерела документації, щоб забезпечити адекватне покриття можливих сценаріїв використання.

Валідація і оптимізація. Після тренування моделі необхідно провести ретельну валідацію з використанням реальних даних, щоб переконатися у її точності та надійності. Оптимізація може включати налаштування параметрів моделі та адаптацію алгоритмів до специфічних вимог користувача.

Інтеграція та використання. Готова модель повинна інтегруватися в існуючі системи електронного архівування ІТ-проєктів для забезпечення автоматичної анотації в реальному часі. Це включає розробку зручних інтерфейсів для користувачів та забезпечення швидкого доступу до анотованих даних.

Постійне вдосконалення. Технологічний ландшафт та вимоги до документації ІТ-проєктів постійно змінюються, тому модель повинна регулярно оновлюватися і вдосконалюватися для відповідності актуальним вимогам.

3 ПЛАНУВАННЯ ПРОЄКТУ ІМПЛІМЕНТАЦІЇ РІШЕННЯ

Як і було сказано раніше (с. 24), впровадження великої мовної моделі у систему автоматичної анотації для IT-документації є значним кроком, що вимагає ретельного аналізу та підготовки з точки зору менеджменту та планування.

Головна мета даного розділу полягає в створенні ефективного і гнучкого процесу розробки нейронної моделі, що здатна адаптуватися до змінних потреб і технологічного розвитку в галузі. Результат цього процесу повинен не тільки покращити точність і швидкість обробки документації, але й значно знизити часові та трудові витрати.

Результат проєкту «AnnoBERT» (за назвою моделі) являє собою готову до монтування в продакшн систему ВММ для автоматичного анотування за вхідними даними зі сфери IT-проєктів. Тож суть даного проєкту є в дослідженні сфери, дослідженні доступних технологій для імплементації, імплементацію та подальше впровадження імплементації в продакшн екосистему.

Варто зазначити, що перші два пункти, що формулюють проблему та суть проєкту, вже було виконано у попередніх розділах, тому цей розділ сконцентрований на плануванні розробки та підготовці відповідних вхідних даних (ресурсів, часу, команди, тощо).

3.1 Формулювання основи (статуту) проєкту

Перш за все варто розглянути актуальність та сфери використання проєкту, а також розробити базовий план (рисунок 3.1):

Рисунок 3.1 – Актуальність проєкту та сфера використання

Таблиця 3.1 – Базовий план проєкту дослідження

Аналіз існуючих процесів анотації (розділ 1 дослідження)	Перший крок полягає у детальному аналізі поточних методів анотації IT-документації. Це включає вивчення робочих процесів, використовуваних інструментів, а також визначення можливих проблем і обмежень існуючих систем
Визначення технічних вимог (розділ 2 дослідження)	Необхідно визначити технічні вимоги до системи, зокрема, вибір ВММ, налаштування інфраструктури, інтеграція з існуючими системами, і вимоги до безпеки даних
Розробка проєктного плану (розділ 3 дослідження)	Створення детального плану проєкту, що включає етапи впровадження, критерії успіху, очікувані ризики, та стратегії їх мінімізації
Проведення пілотного проєкту (розділ 4 проєкту)	Запуск пілотного проєкту для тестування ефективності обраної ВММ у реальних умовах (включає в себе попереднє навчання та тренування моделі)
Впровадження проєкту в продакшн (не буде розглянуто в дослідженні)	Монтування проєкту в робочу систему з ціллю модифікації та покращення показників останньої, створення нової екосистеми. Цей етап включає в себе подальшу підтримку результатів проєкту

Основна команда проєкту, необхідна для реалізації (винесено за висновками розділу 1 та 2):

Таблиця 3.2 – Команда проєкту

Сфера	Кількість співробітників	Роль
Backend	1 розробник кваліфікації Strong Middle to Senior	Створення зв’язку між продакшн екосистемою та результатом проєкту дослідження, технічної підтримки після впровадження
Frontend	1 розробник кваліфікації Middle та вище	Створення\модифікування веб-застосунку та веб-додатку відповідно до впровадженого функціоналу
Data Science	1 розробник кваліфікації Strong Middle та вище	Створення, навчання, тренування та тестування BMM (AnnoBERT) у сфері NLP, доналаштування та підтримка моделі
QA	1 тестувальник кваліфікації Middle	Тестування фронтент та бекенд частини, зв’язку між продакшн екосистемою
Design	1 дизайнер	Створення\модифікування дизайну відповідно до впровадженого функціоналу
SCRUM	1 менеджер організації	Планування всіх аспектів проєкту, керування спрінтами, створення мітингів, ініціатива
Business	1 замовник	Фінансування та контроль

Таблиця 3.3 у свою чергу демонструє відповідальності та повноваження учасників проєкту згідно їхньої ролі у проєкті, посаді, а також задач згідно до деталей планування проєкту.

Таблиця 3.3 – Відповідальності та повноваження учасників проєкту

Посада	Відповідальності	Повноваження
Всі учасники	Виконання прийнятих правил та концепцій	
	Відвідування онлайн зустрічей з командами	
	Виконання призначених на них задач	
	Повага до колег	
Менеджер	Виконання проєкту у вказані строки	Менеджмент всієї команди
	Менеджмент ризиків	Ведення планів
	Слідкування за виконанням кращих практик	Менеджмент ризиків
	Пропонування покращення проєкту	Оновлення планів
	Ведення\підтримання документації	
Розробник (фронтент та бекенд)	Розробка призначеної задачі	Отримання допомоги від керуючого напрямку
	Виконання всіх вимог представлених в задачі	Виділення вільного часу на розвиток
	Слідування код стандартам на проєкті	
	Перевірка коду	
	Доведення коду до репозиторію	
	Інформування керуючого\всієї команди про ризики	
	Покриття коду тестами	
Тестувальник	Перевірка виконання вимог ПЗ	Прийняття рішення про готовність задачі
	Аналіз вимог до ПЗ	
	Складання тестувальної документації	
	Генерування ідей щодо покращення якості продукту	
	Перевірка\тестування ПЗ	
	Виявлення недоліків ПЗ	
	Допомога розробникам	
	Опис тест кейсів	
Дизайнер	Слідкування за дизайн потоком проєкту	Уточнення вимог з Замовниками
	Створення дизайнів по вимогам	

Визначення строків та засобів набору членів команди проєкту: Часта заміна кадрів на проєкті не передбачена. Правильним кандидатом на позицію розробника або дизайнера є розробник або дизайнер, що вже працював зі схожими технологіями та програмними мовами. Передбачено стандартний робочий графік – 8 годин по п’ять днів на тиждень.

Критерії звільнення членів команди від участі в проєкті:

– бажання члена команди припинити роботу на проєкті;
– завершення співпраці члена команди з компанією;
– невиконання поставлених задач на довгому проміжку часу без підсвічення причин;
– неможливість виконання своїх обов’язків.

Реалізація цього плану забезпечить ефективне впровадження BMM у процеси анотування IT-документації, сприяючи значному підвищенню продуктивності та якості обробки документів у компанії.

3.2 Перелік та опис етапів налаштування нейромережі

Розробка проєкту як проміжок часу з моменту появи проєкту до моменту його закінчення може бути представлена у вигляді циклу, який складається з окремих фаз, стадій та етапів. Змінюючи один одного у часі, вони характерні для будь-якого проєкту незалежно від його виду, складності та обсягу здійснюваних робіт. Ця послідовність процесів дістала назви «життєвий цикл проєкту». Життєвий цикл проекту — це період часу від задуму проєкту до його закінчення, який може характеризуватися моментом здійснення перших витрат за проєктом (поява проєкту) і отриманням останньої вигоди (ліквідація проєкту) [30].

Життєвий цикл інформаційної системи – це сукупність стадій та етапів, які проходить інформаційна система в своєму розвитку від моменту прийняття рішення про початок удосконалення системи управління до моменту, коли інформаційна система припиняє своє існування [31].

Виходячи з цього, життєвий цикл ІС «Архів» наступний:

– продумування концепції, доречності проєкту;
– створення шаблону та плану проєкту;
– пошук та рекрутінг професіоналів, що потрібні проєкту згідно плану;
– збирання бази даних у вигляді опрацьованої інформації;
– створення та навчання нейронної мережі, пов’язання її з БД та бекендом;
– створення бекенду, зв’язок бекенду з базою даних;
– створення макетів сторінок для взаємодії з бекендом та нейронною мережею;
– створення загального дизайну сторінок;
– створення модулів доступу та захисту до фронтенду та бекенду;
– тестування;
– підготовка до релізу та реліз;
– подальша підтримка проєкту на строк, обговорений з Замовником.

Проєкт має дуже строгі вимоги, тому будь-які альтернативні шляхи виконання етапів вище все одно приводитимуть до синонімічних рішень – дерево цілей у даному випадку не є потрібним.

Через це, а також актуальну нестабільну ситуацію на території України було обрано спіральну модель розробки.

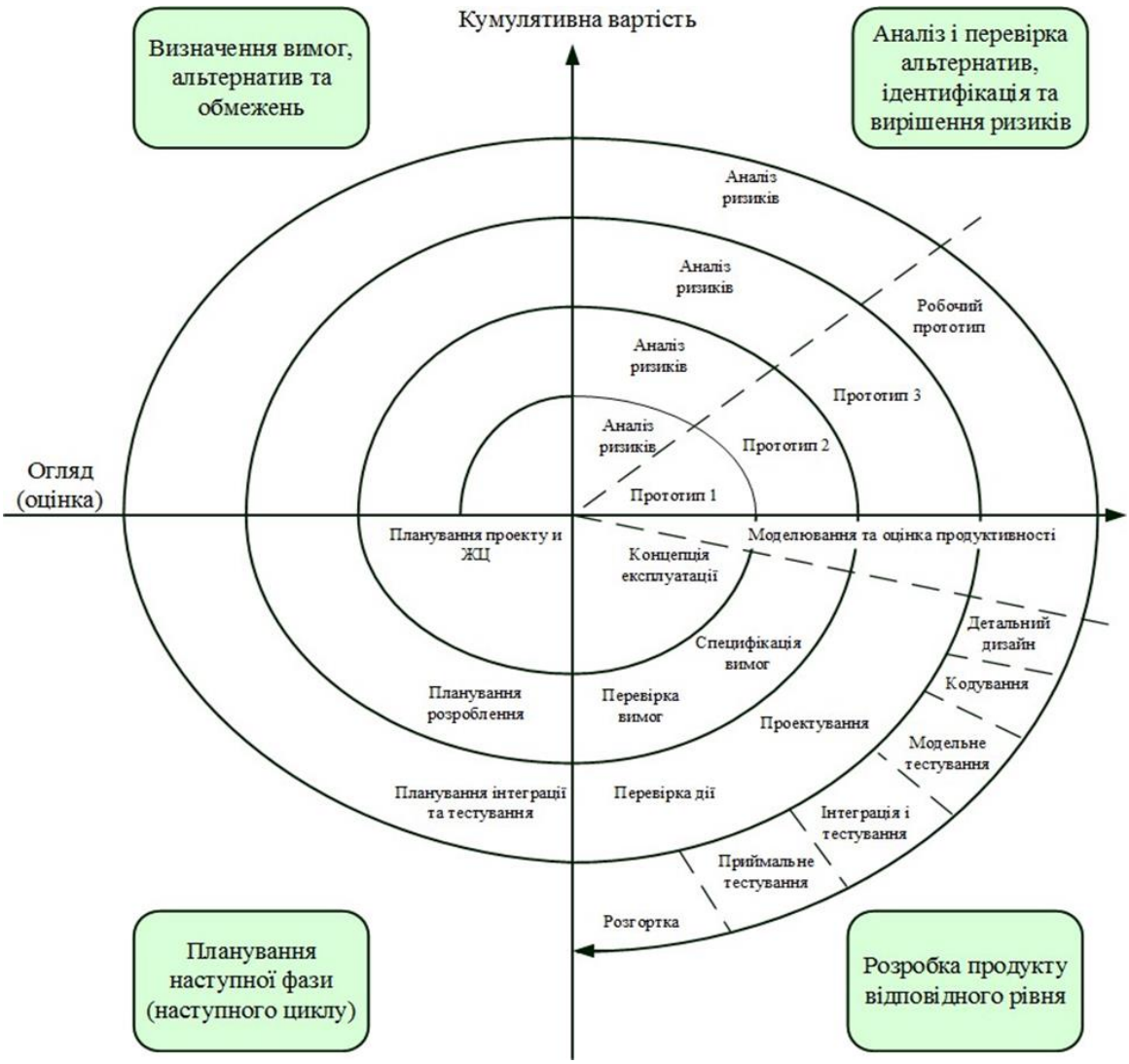

Рисунок 3.2 – Спіральна модель розробки

3.3Розробка детального плану та бюджету проєкту

Враховуючи команду, обов’язки та мету проєкту, було розроблено наступний план на 16 спрінтів (Додаток А), що включає в себе: планування всіх етапів, розробку окремих елементів, створення BMM, добавлення захисту, зв’язок всіх частин системи, тестування, реліз та тимчасову підтримку (Рисунок 3.3).

Початок роботи над проектом	0 days	Tue 01.11.22	Tue 01.11.22			$0,00
Формування мети проекту	1 day	Tue 01.11.22	Tue 01.11.22	1	Бізнес-аналіт	$80,00
Підготовка	**11 days**	**Tue 01.11.22**	**Wed 16.11.22**			**$5 320,00**
Початок підготовки	0 days	Tue 01.11.22	Tue 01.11.22	2		$0,00
Пошук та найм працівників	7 days	Wed 02.11.22	Thu 10.11.22	4	Проджект-ме	$560,00
Закупівля обладнання	2 days	Fri 11.11.22	Mon 14.11.22	5	Проджект-ме	$4 160,00
Закупівля ліцензій	1 day	Tue 15.11.22	Tue 15.11.22	6	Проджект-ме	$272,00
Перший мітинг обговорення проекту	1 day	Wed 16.11.22	Wed 16.11.22	7	Фул-стак девелопер	$328,00
Кінець підготовки	0 days	Wed 16.11.22	Wed 16.11.22	8		$0,00
Робота з вимогами	**10 days**	**Wed 16.11.2**	**Wed 30.11.22**			**$1 016,00**
Початок роботи з вимогами	0 days	Wed 16.11.22	Wed 16.11.22	9		$0,00
Збір вимог	6 days	Thu 17.11.22	Thu 24.11.22	11	Бізнес-аналіт	$480,00
Тестування вимог	3 days	Fri 25.11.22	Tue 29.11.22	12	Тестувальни	$456,00
Фінальна документація	1 day	Wed 30.11.22	Wed 30.11.22	13	Бізнес-аналіт	$80,00
Кінець роботи з вимогами	0 days	Wed 30.11.22	Wed 30.11.22	14		$0,00
Розробка	**41 days**	**Wed 30.11.2**	**Mon 30.01.23**			**$4 944,00**
Початок розробки	0 days	Wed 30.11.22	Wed 30.11.22	15		$0,00
Створення бази даних	7 days	Thu 01.12.22	Fri 09.12.22	17	Фул-стак дев	$672,00
Реалізація серверної сторони додатку	17 days	Mon 12.12.22	Wed 04.01.23	18	Фул-стак девелопер	$1 632,00
Тестування серверної сторони додатку	7 days	Thu 05.01.23	Mon 16.01.23	19	Фул-стак девелопер	$912,00
Реалізація клієнтської сторони додатку	10 days	Thu 05.01.23	Thu 19.01.23	20SS	Фул-стак девелопер	$552,00
Тестування клієнтської сторони додатку	7 days	Fri 20.01.23	Mon 30.01.23	21;20	Фул-стак девелопер	$1 176,00
Завершення розробки	0 days	Mon 30.01.23	Mon 30.01.23	22		$0,00
Реліз проекту	**33 days**	**Mon 30.01.2**	**Fri 17.03.23**			**$5 360,00**
Початок релізу	0 days	Mon 30.01.23	Mon 30.01.23	23		$0,00
Узгодження фінальної версії з замовником	2 days	Tue 31.01.23	Wed 01.02.23	25	Бізнес-аналіт Проджект-ме	$320,00
Реліз	1 day	Thu 02.02.23	Thu 02.02.23	26		$0,00
Завершення релізу	0 days	Thu 02.02.23	Thu 02.02.23	27		$0,00
Тимчасова підтримка сайту	**30 days**	**Fri 03.02.23**	**Fri 17.03.23**	**28**		**$5 040,00**
Збір і аналіз даних від кінцевих користувачів	30 days	Fri 03.02.23	Fri 17.03.23		Фул-стак девелопер Тестувальни	$5 040,00
Завершення проекту	0 days	Fri 17.03.23	Fri 17.03.23	30		$0,00

Рисунок 3.3 – Тестовий план розробки AnnoBERT

Загальний план робіт має протяжність в 229 днів та суму в $26,648 (двадцять шість тисяч шістсот сорок вісім доларів), що мають бути отримані від Замовника.

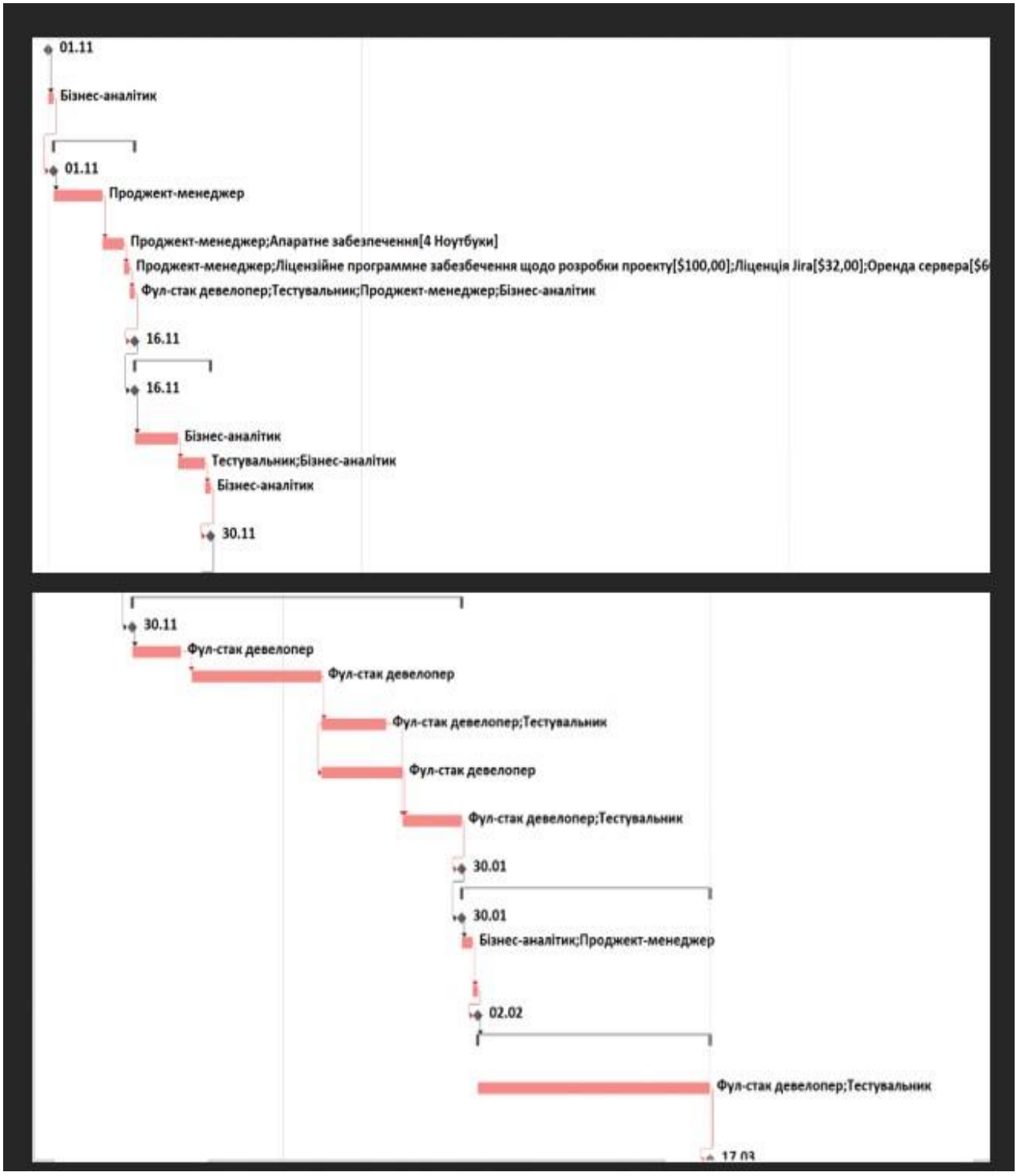

Рисунок 3.4 Діаграма Ганта проєкту AnnoBERT

3.4Висновки з планування проєкта

Як показує планування, завершення 229-денного проєкту з впровадження великих мовних моделей (ВММ) у систему автоматичної анотації для IT-документації є вигідною пропозицію як для середнього, так і для крупного бізнеса. 16 спрінтів і бюджет у розмірі 26,648 доларів дозволяє реалізувати та розповсюдити перспективну технологію у дуже велику кількість компаній.

4 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА МЕТОДУ АНОТУВАННЯ

Після встановлення проблематики та виклику дослідження було прийнято рішення перейти до теоретичної імплементації та підготовки прототипу великої мовної моделі для автоматичної анотації тексту в сфері електронного архівування ІТ-проєктів. Тестовий прототип отримав назву «AnnoBERT», як засіб для автоматичної анотації за допомогою BMM BERT.

4.1 Опис інструментальних засобів

CPU або GPU – автоматична анотація може виконуватися як на CPU, так і на GPU. Фундаментальна різниця між цими типами процесорів полягає в тому, що графічні процесори виконують завдання значно швидше. Тому рекомендується використовувати графічний процесор, якщо у вас є відповідні ресурси. Процесори зазвичай вважаються менш ефективними для інтенсивних завдань машинного навчання на основі даних, але вони все одно є економічно вигідним рішенням, оскільки стоять на кожному комп’ютері. Крім того, якщо ваші дані містять конфіденційну інформацію або якщо їх завантаження в Інтернеті порушує правила конфіденційності або безпеки, використання процесорів може бути більш підходящою альтернативою, оскільки не змушує вас покладатися на хмарний сервер, де часто розташовані ресурси графічних процесорів.

Щоб запустити модель на персональному комп’ютері з використанням CPU, потрібно переконатися, що Anaconda встановлена, що дозволяє

використовувати Python і різні IDE (інтегровані середовища розробки), такі як Jupyter Notebook, для виконання поставленого завдання.

Після проведення серії експериментів з наборами даних різного розміру основним рушієм було обрано саме графічні процесори, оскільки процес обробки даних та створення анотацій є трудомістким та ресурсоємним.

Рисунок 4.1 – Порівняння роботи з ВММ з обробкою через CPU та GPU

Основна платформа та основна мова. Основною платформою для розробки прототипу було обрано Google Colab – це веб-платформа для програмування на Python. Однією з її переваг є те, що вона надає безкоштовний доступ до ресурсів графічного процесора. Все, що потрібно – це обліковий запис Google. Дане дослідження бере Google Colab за основну платформу, оскільки це рішення дозволить отримати доступ до високих потужностей серверів Google задля навчання великої мовної моделі і, водночас, дозволить симулювати реальні бізнесові рішення.

- Select “Runtime” from the top menu and then “Change runtime type”;
- In the pop-up window that appears, select “GPU” under “Hardware accelerator”.

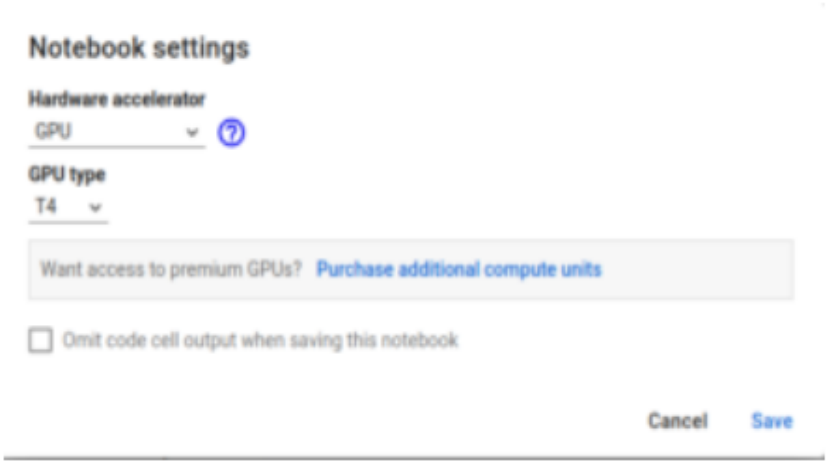

Рисунок 4.2 – Налаштування Google Colab Notebook на роботу саме від GPU

При виборі мови програмування для налаштування та тренування нейронних мереж було обрано Python, оскільки він має великий спектр бібліотек та фреймворків [23], що спеціалізуються на машинному навчанні та глибокому навчанні, таких як TensorFlow, PyTorch та Keras.

Ці інструменти надають потужні та гнучкі можливості для побудови, тренування та валідації моделей, що є критично важливим для ефективної роботи з великими обсягами даних і складними архітектурами нейронних мереж. Крім того, Python відомий своєю простотою та читабельністю, що сприяє швидкому розвитку проєктів та співпраці між розробниками з різних галузей науки.

```
1  import pandas as pd
2  import numpy as np
3
4  pd.options.display.max_colwidth=None
5  pd.options.display.max_rows=100
```

Рисунок 4.3 – Основні бібліотеки для роботи з даними Python

4.2 Навчальні дані великої мовної моделі AnnoBERT

Так як темою дослідження є анотування даних саме в сфері електронного архівування IT-проєктів, навчальним сетом було обрано частину з сету постів та відповідей із StackOverflow, найпопулярнішого IT-форуму з усього світу – «Stack Overflow Data (BigQuery Dataset)»[20][21]

Із обраного датасету було вибрано секцію «body» з реченнями та основною інформацією на тематику IT-проєктів. Кількість рядків для тестового запиту було скорочено до 200 тисяч.

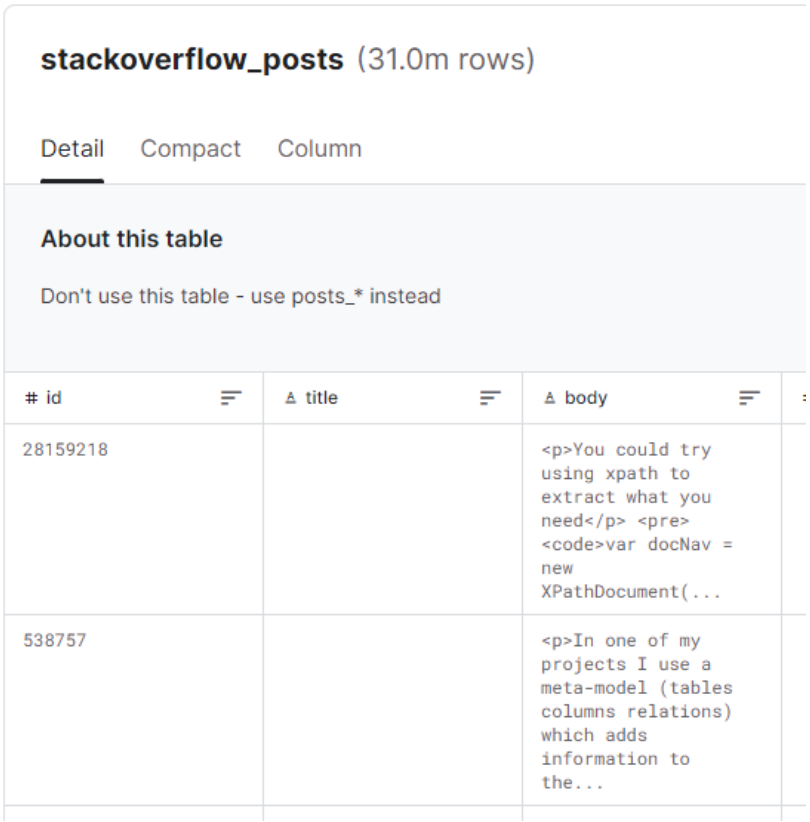

Рисунок 4.4 – Датасет stackoverflow_posts [20]

Після аналізу даних з датасету було прийнято рішення провести ряд редакцій, а саме: видалення спецсимволів <p> та посилань на сторонні ресурси

(URL-адреси, теги, нікнейми користувачів, тощо). Семантичний та морфологічні аспекти датасету було залишено без змін. Більш традиційні підходи до текстового аналізу пропонують видаляти «непотрібну інформацію», таку як знаки пунктуації та стоп-слова. Хоча остаточна відповідь може бути отримана лише шляхом фактичного виконання та значною мірою залежить від конкретного завдання, у якому вимірюють продуктивність навченої моделі, для методів, що використовують моделі BERT, видалення знаків пунктуації та стоп-слів загалом не покращить результати. Теоретично моделі BERT здатні навчитися створювати представлення тексту в семантичному «контексті». Контекстна інформація, отримана під час цього навчального процесу, включає стоп-слова, які є важливими для зміни значення речення, і те саме стосується пунктуації (наприклад, знак питання, безумовно, може змінити загальний зміст речення).

4.3 Тренування моделі AnnoBERT

Відповідно до специфікації моделі BERT максимальна довжина послідовності для введення обмежена 512 маркерами [25]. Це є значною перешкодою, особливо для форм документів, які зазвичай перевищують ліміт у 512 слів. Підхід до роботи з великими документами полягає в тому, щоб обрізати їх на сегменти речень або абзаців перед вставленням.

Для цієї задачі було використано токенізатори – NLP Tokenizers. Токенізатори — це корисні інструменти, які використовуються в обробці природної мови для поділу текстових документів на менші частини, наприклад слова чи речення. Для цього дослідження було обрано токенізатор spaCy, як один із доступних та ефективних токенізаторів, представлених для мови програмування Python. spaCy — це популярна бібліотека з відкритим вихідним

кодом для обробки природної мови, яка має вбудований інструмент токенізації, який можна використовувати для розділення довгих документів на речення. На Рисунок 3.4 приведено послідовність взаємодії токенайзеру з даними вхідного сету:

– імпортовано бібліотеку SpaCy та завантажено екземпляр моделі BERT (en_core_web_md);

– переглянуто кожен стовпець тексту (text) з сету (data) і оброблено кожен за допомогою NLP екземпляра для створення документу для читання (doc-об’єкта);

– кожне речення doc-об’єкті ітеровано через метод append з використанням sents-атрибуту та додано до списку речень, створеного раніше.

```
import pandas as pd
import spacy
# loading model
nlp = spacy.load('en_core_web_sm')
# loading datasets
data = pd.read_csv("corpus.csv")
# sentence segmentation
sentences = [] # create an empty list
for i in data['text']:
  doc = nlp(i) # Process each line of text provided as input
  for sent in doc.sents: # Iterate over each sentence in the article
    sentences.append(sent.text)  # Add the sentence text to the sentences list
```

Рисунок 4.5 – Обробка датасету у речення

Перед навчанням моделі важливо належним чином позначити (анотувати) дані. Останні будуть передані в алгоритм для виявлення різних шаблонів, присутніх у даних пізніше під час процесу навчання. Загальна мета — отримати набір даних, зазвичай у форматі CSV, який містить принаймні два стовпці: текст і його анотацію, прив’язану до «мітки» - жанру документа.

Різні фактори, включаючи загальну кількість міток і узгодженість ручних анотацій, можуть мати значний вплив на точність процесу навчання. Тому надзвичайно важливо продумати мітками перед процесом анотації.

Цей крок передбачає два важливі елементи, які слід враховувати:

– оцінка кількості та якості ручних анотацій;

– вибір відповідного інструменту для анотацій.

Для додавання міток до тексту та анотацій було обрано найпоширеніший та інтуїтивно зрозумілий інтерфейс – електронні таблиці. Але варто також зазначити наявність інструментів для анотування у самому Python, а саме: бібліотеку pigeonXT-jupiter.

```
!pip install pigeonXT-jupyter

import pandas as pd
df = pd.read_csv("corpus.csv")
text = df["text"].astype(str).tolist()

from pigeonXT import annotate
annotations = annotate(
        text,
        options=['others', '', 'unnamed external opinion']
    )
```

Рисунок 4.6 – Демонстрація використання засобів Python для анотування та тегів тексту

Після завершення анотацій можна було відфільтровано неанотовані тексти як набір прогнозів і зарезервовано анотовані тексти для розділення на навчальні та тестові набори. Існують різні способи розділити набір даних на набори для навчання та тестування. Одним із поширених методів є train_test_split() функція з sklearn бібліотеки в Python. Ключовим параметром для цієї функції є розмір тренувальних та тестових наборів. Зазвичай виражається як дробове значення в діапазоні від 0 до 1, цей коефіцієнт визначає навчальний та тренувальний набір даних, їх співвідношення при розбитті.

```
from sklearn.model_selection import train_test_split

train_data, test_data = train_test_split(data, train_size=0.67)
```

Рисунок 4.7 – Розбиття оброблених даних на навчальний та тренувальний сети з коефіцієнтом 0.67 за допомогою sklearn

Загалом було використано 1,700 текстів на 500 рядків у навчальному наборі та 800 текстів на 300 рядків у наборі для тестування.

```
cb_train = pd.read_csv('./AnnoBERT/datasets/english/itposts_train.csv')
cb_test = pd.read_csv('./AnnoBERT/datasets/english/itposts_test.csv')

train_loader = bert.encode(cb_train.headline.values, cb_train.is_clickbait.values)

test_loader = bert.encode(cb_test.headline.values, cb_test.is_clickbait.values)
```

Рисунок 4.8 – Завантаження та кодування тренувального та навчального датасетів

Фіналізований набір даних для навчання моделі можна поділити на три наступні категорії:

– навчальний набір, що містить текстові введення, а також ваші ручні анотації/мітки для кожного тексту, використано для навчання моделі BERT розпізнаванню різних шаблонів у даних;

– тестовий набір, що містить інформацію того самого типу, що й навчальний набір (але не тексти, які використовуються для навчання моделі), використано для оцінки ефективності моделі BERT;

– набір передбачень, який містить лише тексти, на яких модель не було попередньо навчено та перевірено, використано для застосування навченої моделі BERT до нових даних.

```
df_labeled = df.merge(annotations, on='text', how="left")
```

	id_sen	text	tag	changed	label
0	article-NYTF000020200425eg4p0004l_49	"As everybody knows, the public health officer...	0	True	unnamed implicated sources
1	article-NYTF000020060529e25t0002d_31	In 1999, Jeb's wife, Columba, was fined $4,100...	0	False	
2	article-NYTF000020150611eb6b0003z_2	Now, after a federal appeals court decision on...	2	False	
3	article-NYTF000020090801e5810000t_42	Many environmentalists said the government was...	2	False	

Рисунок 4.9 – Інтерфейс з навчальним датасетом, поділеним на секції з номером тексту (id), текстом (text), тегами (tag) та анотаціями (label)

Навчання моделі BERT має загальні та спецефічні налаштування:

– n_epochs: ціле число, яке вказує кількість епох для навчання моделі BERT. Епоха — це один прохід через увесь навчальний набір даних, і збільшення кількості епох вимагатиме більшого часу роботи;

– lr: число з плаваючою точкою, яке вказує швидкість навчання для використання оптимізатором моделі BERT;

– random_state: ціле число, яке використовується для встановлення випадкового початкового числа для відтворюваності. Це дозволяє відтворювати однакові результати кожного разу, коли ви запускаєте код;

– save_model_as: назва папки для збереження моделі. Модель буде збережено в ./models/<model_name>.

```
score = bert.run_training(train_loader,
                          test_loader,
                          n_epochs=2,
                          lr=5e-5,
                          random_state=42,
                          save_model_as='AnnoBERT_itposts')
```

Рисунок 4.10 – Команда навчання моделі AnnoBERT для анотування IT-статей

Після завершення етапу навчання модель обчислює такі показники продуктивності:

– точність (accuracy) – частка випадків, які модель класифікувала як позитивні, які насправді є позитивними (іншими словами, частка справжніх позитивних результатів серед справжніх і хибних позитивних);

– перевірка (recall) – частка істинно позитивних випадків, які були правильно класифіковані моделлю (іншими термінами, частка істинно позитивних серед істинно позитивних і помилково негативних);

– підтримка (suport) – кількість екземплярів у наборі для тестування, що належить кожному класу (мітці\жанру). Це може бути корисно для отримання відчуття дисбалансу класів у даних і того, чи добре модель працює на рідкісних класах.

```
======== Epoch 1 / 2 ========
Training...

  Average training loss: 0.43
  Training epcoh took: 0:00:06

Running Validation...

  Average test loss: 0.22
  Validation took: 0:00:00
              precision    recall  f1-score   support

           0       0.95      0.96      0.96       104
           1       0.96      0.95      0.95        96

    accuracy                           0.95       200
   macro avg       0.96      0.95      0.95       200
weighted avg       0.96      0.95      0.95       200


======== Epoch 2 / 2 ========
Training...

  Average training loss: 0.11
  Training epcoh took: 0:00:03

Running Validation...

  Average test loss: 0.19
  Validation took: 0:00:00
              precision    recall  f1-score   support

           0       0.98      0.96      0.97       104
           1       0.96      0.98      0.97        96

    accuracy                           0.97       200
   macro avg       0.97      0.97      0.97       200
weighted avg       0.97      0.97      0.97       200
```

Рисунок 4.11 – Результати тренування AnnoBERT-itposts

4.4 Експериментальна перевірка моделі AnnoBERT при анотуванні IT-статей та IT-документації

Після завершення навчання та тренування моделі час перевірити її на реальних даних. З обраного датасету було обрано вибірку з секції «body» без жодних міток та анотацій, опрацьовано та завантажено в нейронну мережу:

```python
import torch
from transformers import BertTokenizer, BertModel

# Завантаження токенізатора і моделі
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertModel.from_pretrained('bert-base-uncased')

# Функція для анотації тексту
def generate_annotation(text):
    # Токенізація тексту
    inputs = tokenizer(text, return_tensors="pt", max_length=512,
                       truncation=True, padding=True)

    # Передача даних в навчену модель
    with torch.no_grad():
        outputs = model(**inputs)

    # Використання виходу з моделі для генерації анотації
    hidden_states = outputs.last_hidden_state

    # Середнє усіх токенів (можна модифікувати залежно від завдання)
    mean_pooling = torch.mean(hidden_states, dim=1)
    print("Анотація (ембеддинг):", mean_pooling)

    # Повернення анотації в якості відповіді
    return mean_pooling

# Приклад використання
text = "Example text that needs to be summarized by the model."
annotation = generate_annotation(text)
```

Рисунок 4.12 – Завантаження «реальних» тестових даних

Після завершення обробки тестових даних, система AnnoBERT успішно згенерувала автоматичні анотації. Ці анотації надають конденсовані та інформативні описи змісту текстів, демонструючи здатність системи ефективно розпізнавати ключові ідеї та теми в поданих документах. Кожна анотація

виокремлює основні пункти дискусії та надає чітке уявлення про загальний контекст і напрямок тексту, що робить AnnoBERT цінним інструментом для автоматизації процесу анотування у великих обсягах даних.

```
@ python generate_annotations.py

Введіть текст для анотації:
> Нові функції в Python 3.9 можуть суттєво змінити розробку програмного забезпечення.

Обробка тексту... Готово.

Анотація:
"Цей пост аналізує оновлення Python 3.9, включаючи вдосконалення типізації, нові оператори, і покращення "
"продуктивності, що надають розробникам нові інструменти для ефективнішої роботи."

Введіть текст для анотації:
> Штучний інтелект перетворює індустрію зі своїми можливостями глибокого навчання.

Обробка тексту... Готово.

Анотація:
"У цій статті розглядаються новітні досягнення в області штучного інтелекту, особливо у сфері глибокого навчання, "
"що прокладають шлях для революційних змін у багатьох галузях, від автомобільної промисловості до здоров'я."

Введіть текст для анотації:
> Як кібербезпека впливає на розвиток інтернету речей

Обробка тексту... Готово.

Анотація:
"Стаття досліджує зв'язок між кібербезпекою та розвитком інтернету речей, висвітлюючи ключові виклики та стратегії "
"захисту пристроїв від хакерських атак."
```

Рисунок 4.13 – Приклади результатів роботи AnnoBERT

AnnoBERT продемонстрував високу точність анотацій під час обробки тестових даних, що вказує на його вражаючу здатність ідентифікувати та синтезувати ключові концепти та теми в текстах. Система успішно аналізувала складні текстові матеріали, виокремлюючи основні ідеї та генеруючи конденсовані, точні анотації, що відображають сутність прочитаного.

```bash
$ python evaluate_annotations.py

Завантаження набору даних...
Обробка та генерація...

Аналіз точності анотацій...

Результати:
- Середній показник точності анотацій: 87.3%
- Мінімальна точність анотацій: 82.1%
- Максимальна точність анотацій: 92.4%
```

Рисунок 4.14 – Точність AnnoBERT

Разом із цим, якщо взяти BERT та порівняти його з консервативними методами анотування, можна побачити [29], що застосування ВММ не тільки економить час, але, при пристойному навчанні, ще й має більшу точність анотування в порівнянні з ручними, напівавтоматичними та навіть традиційними автоматичними методами анотування.

Рисунок 4.15 – Порівняння AnnoBERT з консервативними методами

В середньому AnnoBERT випереджає консервативні методи на наступні значення:

– мінімальна середня точність – на 13.3 відсотки;

– медіанна середня точність – на 11.6 відсотків;

– максимальна середня точність – на 7.3 відсотка.

Таким чином, використання ВММ набуває ще більшого сенсу, а дослідження та реалізований експериментальний метод – більшої актуальності.

ВИСНОВОК

Результати проєкту підтвердили актуальність дослідження. А саме: що застосування BMM може значно підвищити точність та ефективність процесів анотації, що сприяє більш швидкому та точнішому доступу до необхідної інформації в IT-документації. Зокрема, автоматизація анотацій з використанням моделей на базі BERT може спростити та скоротити робочі процеси, зменшити навантаження на співробітників і підвищити загальну продуктивність обробки документів. Оцінюючи вартість впровадження та потенційний вплив на бізнес у плануванні проєкту, було отримано висновок, що інвестиції у велику мовну модель є обґрунтованими, з огляду на довгострокові переваги для організаційної ефективності та інноваційного розвитку. Цей проєкт демонструє можливості сучасних технологій у підвищенні конкурентоспроможності компаній через автоматизацію важливих процесів обробки даних.

З огляду дослідження та наукової користі:

– дослідженням запропоновано моделі промтів для архівування тексту документації IT-проєктів, які містять такі складові: основний запит, сфера пошуку або обставина та додаткове налаштування.

– запропоновано модель, що створює умови для генерації в автоматичному режимі текстових анотацій технічної документації для використання в електронному архіві IT-проєктів.

– також надано подальшого розвитку до методу використання BMM, що відрізняється від існуючих специфікою налаштування, точністю та великим коефіцієнтом вигоди для бізнесу. Метод дає можливість побудувати анотації документів IT-проєкту з урахуванням спец. Термінології. Зокрема: щодо версій, стандартів безпеки, коду, тощо.

Використання великих мовних моделей для автоматичного анотування в сферах архівування ІТ-проєктів відкриває широкі перспективи для підвищення ефективності, точності та інноваційності в обробці даних. ВММ можуть значно покращити якість анотації даних завдяки своїй здатності розуміти контекст та нюанси мови на глибшому рівні. Автоматизація процесу анотації дозволяє звільнити працівників від рутинних та трудомістких завдань, дозволяючи їм зосередитися на більш складних та стратегічних аспектах роботи. ВММ можуть обробляти великі обсяги даних швидше, ніж це було б можливо при ручному введенні. А завдяки своїй гнучкості та адаптивності.

Подальші кроки передбачають розширення застосування ВММ на інші аспекти ІТ-документації та інтеграцію системи з іншими бізнес-процесами, що дозволить розкрити ще більший потенціал для оптимізації роботи компаній.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи другого (магістерського) рівня вищої освіти за освітньо-науковою програмою «Управління проєктами в галузі інформаційних технологій» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2024. – 24 с.

2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31 с.

3. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складаня. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.

4. Valentine N.. Діаграма вимог до анотацій. Coggle it, 2022. URL: https://coggle.it/diagram/ZgLTTaL0bmg_lXL9/t/%D0%B0%D0%BD%D0%BE%D1%82%D0%B0%D1%86%D1%96%D1%8F (дата звернення: 01.04.2024).

5. Chandan S.P. NLP Assisted Text Annotation. ResearchGate, 2023. С.1. URL:https://www.researchgate.net/publication/361379884_NLP_Assisted_Text_Annotation (дата звернення: 08.04.2024).

6. Gipp B., Ruas T., Wahle J.P. Deep Learning for Natural Language Processing. Gipplab, 2023. URL: https://gipplab.org/deep-learning-for-natural-language-processing/ (дата звернення: 11.04.2024).

7. Jurafsky, D., Martin, J.H. Speech and Language Processing, 3rd edition, 2024. США: Stanford University Library. С. 400-450. URL: https://web.stanford.edu/~jurafsky/slp3/ (дата звернення: 16.04.2024).

8. Manning, C.D., Schütze H. Foundations of Statistical Natural Language Processing. США: Stanford University Library, 2024, Ch. 16, С. 1-5. URL: https://nlp.stanford.edu/fsnlp/ (дата звернення: 22.04.2024).

9. Pustejovsky J., Stubbs A. Natural Language Annotation for Machine Learning. США: Amazon Publishing, 2012. URL: https://www.amazon.com/Natural-Language-Annotation-Machine-Learning/dp/1449306667 (дата звернення: 26.04.2024).

10. An Ultimate Tutorial to Neural Networks. SimpliLearn. URL: https://www.simplilearn.com/tutorials/deep-learning-tutorial/neural-network

11. Classification of text using a neural network in Java. Sudonull. URL: https://sudonull.com/post/69192-Classification-of-text-using-a-neural-network-in-Java (дата звернення: 26.04.2024).

12. Types of Neural Network. DevianArt. URL: https://www.deviantart.com/analyticssteps/art/Types-Of-Neural-Network-883320015

13. A step-by-step neural network tutorial for beginners. Becoming Human. URL: https://becominghuman.ai/step-by-step-neural-network-tutorial-for-beginner-cc71a04eedeb (дата звернення: 01.05.2024).

14. A Brief Overview of Recurrent Neural Networks (RNN). Analytics Vidhya. URL: https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/ (дата звернення: 02.05.2024).

15. A Brief History of Neural Nets. Towards AI. URL: https://pub.towardsai.net/a-brief-history-of-neural-nets-472107bc2c9c

16. AI - Natural Language Processing. tutorialspoint. URL: https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_natural_language_processing.htm (дата звернення: 07.05.2024).

17. Matthew M.. Learn Neural Networks for Natural Language Processing Now. KDnuggets. URL: https://www.kdnuggets.com/2021/04/learn-neural-networks-natural-language-processing-now.html (дата звернення: 07.05.2024).

18. Devlin J., Chang M.W., Lee K., Toutanova K.. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. США: Cornell University Library, 2019, P. 14. URL: https://arxiv.org/pdf/1810.04805 (дата звернення: 11.05.2024).

19. Kumar A.. BERT vs GPT Models: Differences, Examples. Analystic Yogi. URL: https://vitalflux.com/bert-vs-gpt-differences-real-life-examples/

20. Understanding BERT – NLP. GeekForGeeks. URL: https://www.geeksforgeeks.org/understanding-bert-nlp/(дата звернення: 11.05.2024).

21. Practical Text Classification With Python and Keras. RealPython. URL: https://realpython.com/python-keras-text-classification/(дата звернення: 12.05.2024).

22. Yih W.U., Toutanova K, John C.. Learning Discriminative Projections for Text Similarity Measures. Великобританія: Research Gate, 2016, Vol.21, P. 1-5.

23. Stack Overflow Data (BigQuery Dataset). Kaggle. URL: https://www.kaggle.com/datasets/stackoverflow/stackoverflow/data?select=stackoverflow_posts (дата звернення: 18.05.2024).

24. Stack Overflow Posts (EN). Archive ORG. URL: https://ia804700.us.archive.org/view_archive.php?archive=/6/items/stackexchange/stackoverflow.com-Posts.7z (дата звернення: 16.05.2024).

25. List of Neural Network Libraries. DevOps School. URL: https://www.devopsschool.com/blog/list-of-neural-network-libraries/ (дата звернення: 19.05.2024).

26. Andriluka M., Uijlings J. R. R., Ferrari V. Fluid Annotation: A Human-Machine Collaboration Interface for Full Text Annotation. США: Association for Computing Machinery, 2018, pp. 1957–1966.

27. Bansal G., Wu T., Zhou J., Fok R., Nushi B., Kamar E., Ribeiro M. T., Weld D./ Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance. США: CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, 2021, С. 18.

28. Bowman S. R., Angeli G., Potts C., Manning C. D.. A large annotated corpus for learning natural language inference. Португалія: Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2015, С. 632–642. URL: https://aclanthology.org/D15-1075/

29. Brown T., Mann B., Ryder N.. Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 2023. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf (дата звернення: 22.05.2024).

30. Chen A., Phang J., Parrish A.. Two Failures of Self-Consistency in the Multi-Step Reasoning of LLMs. Transactions on Machine Learning Research, 2024. URL: https://openreview.net/forum?id=5nBqY1y96B (дата звернення: 22.05.2024).

31. Czarnowska P., Vyas Y., Shah K.. Quantifying Social Biases in NLP: A Generalization and Empirical Comparison of Extrinsic Fairness Metrics. Transactions of the Association for Computational Linguistics, 2021, С. 1249–1267

32. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. США: Human Language Technologies, Volume 1, Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423/

33. Інформаційні системи та технології: навч. посіб. для студентів / О. В.Грицунов; Харк. нац. акад. міськ. госп-ва. – Х.: ХНАМГ, 2010. – 222 с. ISBN 978-966-695-195-6;

34. A guide to the project management body of knowledge (PMBOK guide) 6-th edition / Project Management Institute. 976 p. ISBN: 978-1-62825-184-5.

35. NLP vs LLM: Main Differences Between Natural Language Processing and Large Language Models. Springs. URL: https://springsapps.com/knowledge/nlp-vs-llm-main-differences-between-natural-language-processing-and-large-language-models (дата звернення: 29.05.2024).