

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій**

«На правах рукопису»
УДК 004.9

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
« » _____ 2024 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою

«Інформаційні управляючі системи та технології»

зі спеціальності 126 «Інформаційні системи та технології»

на тему: «Система розпізнавання та перекладу тексту документів»

Виконав:

студент 2 курсу, групи ІС-33мп
Патока Владислав Володимирович _____

Керівник:

доцент каф. ІСТ, к.ф.-м.н., доц.
Гавриленко Олена Валеріївна _____

Рецензент:

к.н.т., доцент кафедри ІІІ
Олійник Юрій Олександрович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2024 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Патоці Владиславу Володимировичу

1. Тема дисертації «Система розпізнавання та перекладу тексту документів», науковий керівник дисертації Гавриленко Олена Валеріївна, к.ф.-м.н., доц., затверджені наказом по університету від «08» 11 2024 р. № 5016-с
2. Термін подання студентом дисертації «09» 12 2024 р.
3. Об'єкт дослідження: процеси зі сканування фізичної копії документу, розпізнавання на ньому тексту та розділення за іменованими сутностями, а в подальшому його переклад.
4. Вихідні дані: коректно відсканований документ, вірно розпізнаний текст, перекладений текст з якісною оцінкою метрик.
5. Перелік завдань, які потрібно розробити: провести аналіз існуючих рішень, формування вимог до системи, розробка сканеру документів, розпізнавання тексту з зображення, навчання моделі для розпізнавання іменованих сутностей документу, переклад тексту.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: Діаграма прецедентів (А3), Діаграма потоків даних (А3), Діаграма бази даних (А3), Діаграма бізнес-процесів (А3), Діаграма послідовності (А3), Архітектура

інформаційної системи (А3), Діаграма класів інтерфейсу користувача (А3), Діаграма активності (А3).

7. Перелік публікацій:

Патока В.В. Гавриленко О.В.. Штучний інтелект для виявлення та розпізнавання автомобільних номерів на базі архітектури глибокого навчання EfficientNet та адаптації розмірів анкорів. Збірник «Сучасні аспекти та перспективні напрямки розвитку науки», 2024 рік, с. 308-309. DOI 10.36074/liga-inter-19.01.2024.

Патока В.В. Гавриленко О.В.. Провідні OCR моделі для розпізнавання тексту на зображеннях. Журнал «UNIVERSUM», 2024 рік, с. 356-364. DOI 10.36074/universum.9.2024.

8. Дата видачі завдання 02.09.2024 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Визначення вимог та постановка цілей до системи	08.09.2024	
2.	Ознайомлення та вибір технологій та інструментів для виконання завдання	15.09.2024	
3.	Розробка OCR модуля	22.09.2024	
4.	Розробка NER модуля	06.10.2024	
5.	Розробка модуля для сканування документів	13.10.2024	
6.	Інтеграція модулів в єдину систему	20.10.2024	
7.	Тестування та оцінка продуктивності системи	20.10.2024	
8.	Написання документації до розробленої системи	27.10.2024	
9.	Оформлення пояснювальної записки	02.12.2024	

Студент

Владислав ПАТОКА

Науковий керівник

Олена ГАВРИЛЕНКО

РЕФЕРАТ

Система розпізнавання та перекладу тексту документів: 110 с., 28 табл., 15 рис., 8 дод., 22 джерел.

ОБРОБКА ЗОБРАЖЕННЯ, СКАНУВАННЯ ДОКУМЕНТІВ, ПЕРЕКЛАД ТЕКСТУ, ІМЕНОВАНІ СУТНОСТІ, КЛАСИФІКАЦІЯ ДАНИХ, МЕТОДИ РОЗПІЗНАВАННЯ ТЕКСТУ

Наразі вже існує чимало програмних продуктів, які дозволяють перекласти текст на зображенні документу. Однак більшість з них мають певні недоліки, які не дозволяють повністю автоматизувати цей процес. Головною проблемою є те, що вони не надають можливості отримати на виході іменовані сутності цього документу та їх перекладені версії, а також інструментарій для їх обліку, що є вкрай важливим у подібній сфері. Таким чином можна вважати, що розробка подібного продукту є досить актуальною потребою в наш час.

Метою даної роботи є автоматизація процесу цифровізації та перекладу документів шляхом розробки програмного додатку, який дозволяє сканувати документи, визначати іменовані сутності (адреси, телефони, основний текст), перекладати їх та здійснювати облік.

Об'єктом дослідження є процеси сканування фізичних документів, розпізнавання тексту на зображеннях та його переклад після визначення іменованих сутностей.

Предметом дослідження є методи та засоби реалізації системи для обробки текстових даних за допомогою сучасних алгоритмів машинного навчання.

Тези автора на тему «Штучний інтелект для виявлення та розпізнавання автомобільних номерів на базі архітектури глибокого навчання EfficientNet та адаптації розмірів анкорів» були опубліковані в збірнику «Сучасні аспекти та перспективні напрямки розвитку науки», 2024 рік.

Стання автора «Провідні OCR моделі для розпізнавання тексту на зображеннях» опублікована в журналі «UNIVERSUM», 2024 рік.

ABSTRACT

System for document text recognition and translation: 110 p., 28 tab., 15 draw., 8 app., 22 sources.

IMAGE PROCESSING, DOCUMENT SCANNING, TEXT TRANSLATION, NAMED ENTITIES, DATA CLASSIFICATION, TEXT RECOGNITION METHODS

There are already many software products that allow you to translate text in a document image. However, most of them have certain disadvantages that prevent them from fully automating this process. The main problem is that they do not provide the ability to obtain named entities of the document and their translated versions, as well as tools for their accounting, which is extremely important in this area. Thus, it can be assumed that the development of such a product is a rather urgent need in our time.

The aim of this paper is to automate the process of digitalization and translation of documents by developing a software application that allows scanning documents, identifying named entities (addresses, phone numbers, main text), translating them, and keeping track of them.

The object of research is the processes of scanning physical documents, recognizing text in images and translating it after identifying named entities.

The subject of the study is the methods and means of implementing a system for processing text data using modern machine learning algorithms.

The author's thesis on "Artificial intelligence for detection and recognition of license plates based on the EfficientNet deep learning architecture and anchor size adaptation" was published in the collection "Modern aspects and perspective directions of science development", 2024.

The article "Leading OCR models for text recognition in images" was published in the journal "UNIVERSUM", 2024.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	9
ВСТУП.....	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	12
1.1 Використання штучного інтелекту для обробки тексту	12
1.2 Сучасні технології для перекладу тексту	13
1.3 Автоматизація процесу розпізнавання та перекладу тексту документів	15
1.4 Огляд наявних рішень.....	17
1.4.1 Сервіс перекладу Google Translate	17
1.4.2 Система розпізнавання тексту ABBYY FineReader.....	19
1.4.3 Сервіс перекладу Microsoft Translator.....	20
1.5 Постановка задачі.....	22
Висновки до розділу 1	23
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	25
2.1 Мова програмування Python	25
2.2 Бібліотека OpenCV для обробки зображень.....	26
2.3 PyTesseract для розпізнавання тексту	28
2.4 Бібліотека SpaCy для визначення іменованих сутностей	29
2.5 Веб-фреймворк Flask.....	30
2.6 СУБД PostgreSQL	31
2.7 Сервіс машинного перекладу DeepL.....	32
Висновок до розділу 2.....	33
3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	35
3.1 Архітектура програмного застосунку	35
3.1.1 Потоки даних у системі	36
3.1.2 Бізнес-процеси програмного застосунку	37
3.2 Моделювання поведінки системи за допомогою діаграми активності	38
3.3 Діаграма послідовності взаємодії компонентів системи	39
3.4 Програмний модуль для сканування документів.....	40

3.5 Розпізнавання тексту та визначення іменованих сутностей.....	42
3.6 Програмний модуль для перекладу тексту.....	46
3.7 Архітектура бази даних	47
3.8 Користувацький інтерфейс та його особливості.....	50
3.8.1 Структура інтерфейсу користувача.....	51
3.8.2 Огляд сторінок користувацького інтерфейсу.....	53
Висновки до розділу 3	56
4 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	58
4.1 Змістовна постановка задачі	58
4.2 Математична постановка задачі	59
4.2.1 Вхідні дані.....	59
4.2.2 Вихідні дані.....	60
4.2.3 Процес розв'язання	61
4.3 Обґрунтування методу розв'язання.....	63
4.3.1 Оцінка якості роботи NER-моделі.....	63
4.3.2 Оцінка читабельності тексту.....	64
4.3.3 Оцінка якості перекладу	65
4.4 Експериментальне дослідження	66
4.4.1 Оцінка якості роботи NER-моделі.....	66
4.4.2 Оцінка читабельності тексту.....	67
4.4.3 Оцінка якості перекладу	68
Висновок до розділу 4.....	69
5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ	71
5.1 Опис ідеї проєкту	72
5.2 Технологічний аудит ідеї проєкту	75
5.3 Аналіз ринкових можливостей запуску стартап-проєкту.....	77
5.4 Розроблення ринкової стратегії проєкту	88
Висновок до розділу 5.....	97
ВИСНОВКИ.....	98
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	101

ДОДАТОК А	103
ДОДАТОК Б	104
ДОДАТОК В	105
ДОДАТОК Г	106
ДОДАТОК Д	107
ДОДАТОК Е	108
ДОДАТОК Ж	109
ДОДАТОК И	110

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

СУБД – система управління базою даних

ШІ – штучний інтелект

BPMN – Business Process Model and Notation

BERT – Bidirectional Encoder Representations from Transformers

BLEU – Bilingual Evaluation Understudy

DFD – Data Flow Diagram

GPT – Generative Pre-trained Transformer

NER – Named Entity Recognition

NLP – Natural Language Processing

NMT – Neural Machine Translation

OCR – Optical Character Recognition

ВСТУП

Розвиток інформаційних технологій у галузі штучного інтелекту відкриває перед нами безліч нових можливостей, дозволяючи автоматизувати більшість складних та трудомістких процесів. У час, коли обсяги інформації зростають з неймовірною швидкістю, ефективна робота з обліком документів є надзвичайно важливою як для приватних осіб, так і для бізнесу та держави в цілому. Переклад документів є одним з важливих процесів цього розвитку, так як сприяє покращенню комунікації між носіями різних мов.

На сьогоднішній день існує велика потреба в автоматизованих рішеннях для обробки документів, які дозволять легко та швидко перекласти документ на іншу мову. Це особливо важливо в таких сферах, як: міжнародна торгівля, логістика, ділове спілкування, наукові дослідження та подібних – де робота з документами різних мов є невід'ємною частиною діяльності.

Актуальність даної роботи зумовлена зростаючою потребою у програмних рішеннях, які забезпечать автоматизацію процесів розпізнавання та перекладу тексту з паперових документів. У сучасному світі, де ефективність і швидкість обробки інформації є ключовими факторами, такі програмні продукти стають невід'ємною складовою багатьох сфер діяльності.

Вони знаходять широке застосування в міжнародній торгівлі для швидкого перекладу контрактів і митних декларацій, у логістиці для автоматизації обробки накладних і транспортної документації, у наукових дослідженнях для аналізу друкованих джерел, а також в адміністративній роботі для обробки паперового документообігу.

Мета і завдання дослідження. Метою даної роботи стала автоматизація процесу цифровізації та перекладу документів шляхом створення програмного додатку.

Даний додаток має надавати користувачу можливості з: сканування існуючого документу; визначення його іменованих сутностей, таких як: основний текст, адреси, номери телефонів і тому подібні; їх переклад; облік документів.

Для досягнення поставленої мети було висунуто ряд задач, а саме:

- аналіз ринку на наявність подібних готових рішень та інструментів для сканування, перекладу та обліку документів;
- визначення базових вимог до програмного застосунку;
- розробка архітектури інформаційної системи, включаючи необхідні для реалізації поставлених вимог модулі, а також визначення переліку інструментів, необхідних для його реалізації;
- проектування користувацького інтерфейсу додатку;
- власне розробка програмного продукту відповідно до створеної архітектури з урахуванням поставлених перед ним вимог.

Об'єктом дослідження даної наукової роботи є процеси зі сканування фізичної копії документу, розпізнавання на ньому тексту та розділення за іменованими сутностями, а в подальшому їх переклад.

Предметом дослідження виступають методи і засоби реалізації системи для обробки текстових даних з використанням сучасних алгоритмів машинного навчання.

Результати дослідження, представленого в магістерській дисертації, пройшли апробацію на конференціях та в наукових публікаціях. Тези автора на тему «Штучний інтелект для виявлення та розпізнавання автомобільних номерів на базі архітектури глибокого навчання EfficientNet та адаптації розмірів анкорів» були опубліковані в збірнику «Сучасні аспекти та перспективні напрямки розвитку науки», 2024 рік. Стання автора «Провідні OCR моделі для розпізнавання тексту на зображеннях» опублікована в журналі «UNIVERSUM», 2024 рік.

Зважаючи на це, розробка ефективних інструментів для розпізнавання і перекладу тексту стає важливим кроком до оптимізації робочих процесів і зменшення людського фактора у виконанні рутинних задач. Практичне значення одержаних результатів полягає в тому, що розроблений програмний застосунок стане корисним інструментом для значної частки специфічних користувачів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

Штучний інтелект стає все більш і більш важливим інструментом для обробки текстової інформації, особливо в контексті автоматизації процесів розпізнавання, класифікації та аналізу. Використання ШІ для обробки тексту дозволяє значно покращити точність, швидкість і ефективність таких процесів порівняно з традиційними підходами, що ґрунтуються на правилах або простих алгоритмах.

1.1 Використання штучного інтелекту для обробки тексту

Штучний інтелект (ШІ) має великий потенціал у сфері обробки тексту, зокрема в таких завданнях, як розпізнавання, аналіз та переклад текстової інформації. Використання ШІ для цих цілей забезпечує значне підвищення ефективності обробки великих обсягів даних та автоматизацію багатьох рутинних процесів, які раніше вимагали значних зусиль та часу.

Однією з основних проблем, яку вирішує ШІ у сфері обробки тексту, є оптичне розпізнавання символів OCR технології дозволяють автоматично розпізнавати текст на зображеннях або сканованих документах і перетворювати його в редагований текстовий формат. Це особливо важливо для цифровізації великої кількості документів, де текст часто зафіксований в зображеннях або нерозпізнаваних для стандартних програмних засобів формах. Розпізнавання символів передбачає використання різних математичних та статистичних методів, зокрема нейронних мереж, для ідентифікації та класифікації кожного символу на зображенні [1].

Іншим важливим аспектом є обробка природної мови. NLP – це галузь ШІ, яка займається аналізом та інтерпретацією людської мови, щоб перетворити її в структури, з якими може працювати комп'ютер. Одним із основних завдань NLP є розпізнавання іменованих сутностей (NER), тобто автоматичне виявлення та класифікація важливих елементів у тексті, таких як імена, адреси, дати, організації,

тощо. Це дозволяє системам зберігати важливу інформацію та використовувати її для подальших операцій, таких як аналіз або переклад [2].

Окрім того, великою перевагою є можливість машинного перекладу тексту. Сучасні методи машинного перекладу ґрунтуються на нейронних мережах, зокрема на моделях трансформерів, таких як BERT (Bidirectional Encoder Representations from Transformers) та GPT (Generative Pre-trained Transformer), які здатні вчитися на величезних обсягах мовних даних і враховувати контекст кожного слова в реченні для точного перекладу. Це дає змогу значно поліпшити якість перекладу текстів, оскільки моделі можуть враховувати різні граматичні структури та значення слів в контексті.

Крім того, для обробки тексту застосовуються різні методи класифікації та сегментації текстів, що дозволяє автоматично сортувати або розділяти тексти за різними категоріями, зокрема темами, стилем або навіть емоційним забарвленням. Це відкриває нові можливості для автоматизованої обробки великих обсягів даних у таких галузях, як право, фінанси, маркетинг та наука.

Загалом, використання штучного інтелекту для обробки тексту створює значні можливості для автоматизації безлічі процесів у різних сферах, зокрема для автоматичної обробки документів, перекладу текстів, а також для побудови систем, які здатні аналізувати і використовувати текстову інформацію в контексті. Це дозволяє значно підвищити продуктивність праці, знизити кількість помилок і забезпечити більш ефективну обробку та управління даними в різних галузях.

1.2 Сучасні технології для перекладу тексту

Переклад тексту є однією з основних завдань, яке активно досліджується в галузі штучного інтелекту та обробки природної мови. Останніми роками значно покращилися технології машинного перекладу завдяки розвитку нейронних мереж та глибокого навчання, що дозволило забезпечити більш точні та контекстуально правильні переклади. У цій сфері відбулися значні зрушення, і зараз можна

говорити про кілька ключових підходів, що визначають сучасний стан машинного перекладу.

Одним із найпоширеніших методів є нейронний машинний переклад (NMT), який використовує глибокі нейронні мережі для автоматичного перекладу тексту з однієї мови на іншу. Цей метод працює за принципом «переклад як послідовність», де модель навчається на величезних обсягах двомовних даних і здатна генерувати природні переклади, враховуючи контекст кожного окремого слова в реченні. Однією з основних переваг NMT є те, що він значно поліпшує точність перекладу завдяки здатності до навчання на великих корпусах текстів, що дозволяє враховувати складні граматичні структури та специфіку мови [3].

Серед найбільш відомих методів NMT є трансформери, такі як модель BERT, яка здатна обробляти текст у двосторонньому напрямку, що дозволяє досягти високої точності у розумінні контексту. Моделі на основі трансформерів використовуються у багатьох популярних системах машинного перекладу, таких як Google Translate та DeepL. Вони здатні ефективно працювати з великими текстами та забезпечують високий рівень точності перекладу, оскільки враховують не тільки кожне окреме слово, але й загальний контекст цілої фрази або речення.

DeepL, зокрема, є одним із лідерів серед сучасних систем машинного перекладу. Це платформа, яка використовує нейронні мережі для досягнення високої точності перекладу, зокрема завдяки спеціальним архітектурам трансформерів. DeepL відомий своєю здатністю забезпечувати переклади, які звучать природно для носіїв мови, що робить його популярним серед користувачів, які потребують не тільки точності, але й високої якості тексту на виході [4].

Інші методи машинного перекладу, які використовуються для перекладу текстів, включають статистичний машинний переклад (SMT) та правилами керовані системи, хоча в останні роки вони поступово відходять на другий план через низьку гнучкість та точність у порівнянні з нейронними мережами. Важливою перевагою NMT є те, що ці моделі здатні адаптуватися до різних стилів мови, враховуючи різні контексти та синоніми, що значно полегшує переклад текстів у різних сферах, таких як бізнес, наука та техніка.

У контексті автоматизації процесів, таких як обробка документів, машинний переклад також застосовується для інтеграції багатьох мов у єдину систему, де текст може бути швидко перекладений і переданий для подальшої обробки або використання. Це значно прискорює роботу з документами на різних мовах, полегшує міжкультурну комунікацію та підвищує ефективність бізнес-процесів.

Таким чином, сучасні технології машинного перекладу, зокрема на основі нейронних мереж та моделей трансформерів, відкривають нові можливості для обробки тексту, роблячи переклад швидким, точним і ефективним. Вони значно полегшують роботу з великими обсягами текстової інформації та сприяють розвитку комунікації між носіями різних мов.

1.3 Автоматизація процесу розпізнавання та перекладу тексту документів

Автоматизація процесу розпізнавання та перекладу тексту документів є важливою складовою сучасних інформаційних технологій, що дозволяє значно зменшити час і зусилля, необхідні для обробки великої кількості текстової інформації. Вона є необхідною умовою для ефективної роботи з документацією в багатьох сферах діяльності, таких як бізнес, наукові дослідження, державні органи, міжнародна торгівля та інші, де важливо забезпечити швидкий та точний переклад текстів на різних мовах.

Процес автоматизації розпізнавання та перекладу тексту складається з кількох етапів. Першим етапом є оптичне розпізнавання символів, що полягає в перетворенні зображення документу в редагований текст. Технології OCR дозволяють аналізувати зображення та визначати на ньому символи, перетворюючи їх у текстову форму. Однак, цей процес часто стикається з проблемами точності через погану якість зображення, складні шрифти чи різноманітність мовних конструкцій. Для вирішення цих проблем застосовуються сучасні методи штучного інтелекту, зокрема глибоке навчання, що дозволяє досягти високої точності розпізнавання навіть в умовах низької якості зображення.

Другим важливим етапом є визначення іменованих сутностей (NER). Цей етап полягає в аналізі тексту з метою виявлення важливих інформаційних елементів, таких як імена людей, організацій, місць, дати, адреси тощо. Ідентифікація таких сутностей є важливою для подальшого перекладу, оскільки вони часто мають специфічні контексти, які потребують окремої уваги під час перекладу. Наприклад, адреси або імена організацій повинні залишатися незмінними в процесі перекладу, тоді як загальний текст потребує граматичного коректного перекладу з урахуванням контексту.

Наступним кроком є переклад тексту. Для цього використовуються різні технології машинного перекладу, такі як нейронні мережі або трансформери, які навчаються на величезних обсягах двомовних даних. Сучасні системи перекладу, такі як Google Translate або DeepL, здатні забезпечити високий рівень точності перекладу, враховуючи не тільки окремі слова, а й контекст усього тексту. Це дозволяє отримати переклади, які є не тільки граматично правильними, але й природними для носіїв мов. Однак, навіть ці системи мають свої обмеження, зокрема в складних випадках, коли є необхідність зберігати специфіку термінів або інших важливих елементів, що потребують індивідуального підходу.

Одним з важливих аспектів автоматизації процесу розпізнавання та перекладу є інтеграція інструментів для обліку документів, що дозволяє зберігати всі результати обробки в структурованому вигляді. Це забезпечує зручність у подальшому використанні перекладених документів, а також можливість здійснювати пошук, фільтрацію та сортування за різними параметрами, такими як дата, тип документа або інші важливі метадані.

Автоматизація всіх цих процесів дозволить створити потужний інструмент для роботи з документацією, що відкриває нові можливості для підприємств і організацій, дозволяючи швидко та ефективно працювати з великими обсягами текстової інформації, знижуючи витрати на людські ресурси і мінімізуючи кількість помилок. У майбутньому розвиток штучного інтелекту, зокрема в галузі машинного перекладу та обробки тексту, лише покращить ефективність цих

процесів, забезпечуючи ще більш високий рівень точності та швидкості обробки документів.

1.4 Огляд наявних рішень

1.4.1 Сервіс перекладу Google Translate

Google Translate є одним з найпопулярніших і широко використовуваних інструментів для перекладу текстів на різні мови. Це безкоштовний онлайн-перекладач, який підтримує понад 100 мов і здатний здійснювати переклад тексту, зображень і навіть голосу [5]. Для роботи з текстом на зображеннях Google Translate використовує технологію оптичного розпізнавання символів (OCR), що дозволяє розпізнавати текст на фотографіях (рисунок 1.1), сканованих документах або екранах мобільних пристроїв.

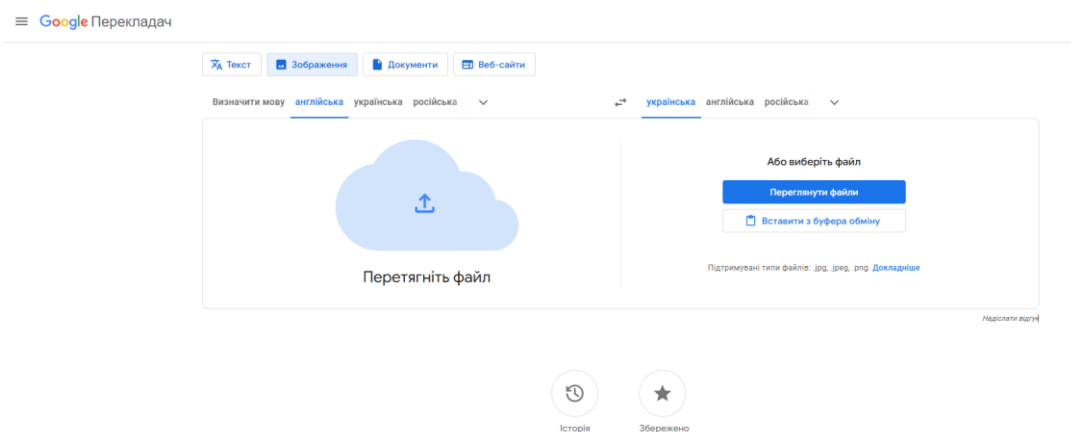


Рисунок 1.1 – Користувацький інтерфейс Google Translate

Однією з ключових переваг Google Translate є інтеграція з іншими продуктами Google, зокрема Google Cloud Vision API, яке надає функціональність OCR. Це дозволяє розпізнавати текст на зображеннях і перетворювати його на редагований текст, а потім одразу перекладати цей текст на потрібну мову. Таким чином, Google Translate автоматизує два важливі етапи обробки тексту – розпізнавання та переклад, що робить цей процес зручним і швидким.

Важливою особливістю Google Translate є можливість перекладати текст не тільки зображень, але й веб-сторінок, документів (PDF, Word) та навіть автоматично перекладати текст в реальному часі, використовуючи камеру смартфона. Цей функціонал широко використовується для перекладу текстів з фотографій, що робить Google Translate незамінним інструментом для швидкої і зручної обробки тексту з будь-яких джерел.

Крім того, Google Translate підтримує автоматичне визначення мови, що дозволяє переводити текст без необхідності вказувати мову джерела. Це робить сервіс ще більш зручним для користувачів, які працюють з багато мовними документами або не можуть визначити мову тексту самотійно.

Google Translate використовує глибокі нейронні мережі для забезпечення високої якості перекладу, що дає йому перевагу перед традиційними методами машинного перекладу. Хоча цей перекладач не є бездоганним і іноді може допускати помилки, він регулярно оновлюється, що дозволяє зменшувати похибки в перекладі і покращувати якість текстів.

Завдяки своїй простоті у використанні, високій швидкості обробки та широким можливостям перекладу, Google Translate є ефективним інструментом для зчитування та перекладу текстів з зображень, що робить його популярним вибором серед користувачів у багатьох сферах діяльності, таких як бізнес, наука, освіта та повсякденне використання.

Попри широкий функціонал та зручність у використанні, Google Translate має певні недоліки, які стають очевидними при вирішенні спеціалізованих завдань. Основна проблема полягає у тому, що сервіс орієнтований на загальний користувацький досвід і не завжди відповідає потребам професійної автоматизації, особливо у сфері роботи з документами.

По-перше, хоча Google Translate ефективно зчитує текст із зображень за допомогою OCR, він не надає можливості автоматично ідентифікувати іменовані сутності, такі як адреси, номери телефонів, імена чи дати. Для користувачів, які працюють з великими обсягами документів і потребують не лише переклад, а й аналітичну обробку тексту, ця обмеженість стає значною перешкодою.

По-друге, Google Translate не має вбудованого механізму для інтеграції результатів обробки у власні системи обліку або бази даних. Це означає, що користувачі не можуть автоматично зберігати іменовані сутності та їх переклад у структурованому вигляді для подальшого використання. У контексті роботи з документами, де важливий облік і структура даних, це вимагає додаткових етапів обробки.

Нарешті, Google Translate має обмежені можливості для кастомізації процесів перекладу. У розроблюваній системі передбачається створення інструментів, які дозволять налаштовувати обробку тексту відповідно до конкретних потреб користувача, забезпечуючи гнучкість у роботі.

Таким чином, хоча Google Translate є потужним інструментом для загального використання, його функціонал не враховує специфічні потреби користувачів, які працюють із документами, потребуючи більшої автоматизації, ідентифікації сутностей та інтеграції з іншими системами. Ці аспекти враховані в розроблюваній роботі, що націлена на вирішення цих недоліків і створення більш спеціалізованого продукту.

1.4.2 Система розпізнавання тексту ABBYY FineReader

ABBYY FineReader є однією з найбільш відомих програм для оптичного розпізнавання тексту і часто використовується для роботи з документами, що потребують переведення в цифровий формат. Продукт має широкий набір функцій для розпізнавання тексту з різних типів зображень, включаючи скановані документи, PDF-файли та фотографії (рисунок 1.2). ABBYY FineReader підтримує багатомовний переклад і надає можливість редагувати результати розпізнавання [6].

Хоча ABBYY FineReader може зчитувати текст і здійснювати переклад через інтегровані сервіси, він не пропонує автоматичного виділення іменованих сутностей або їх обробки у структурованому форматі. Це обмежує застосування

продукту у завданнях, де необхідно розпізнавати, класифікувати та зберігати дані, такі як адреси чи номери телефонів, для подальшого аналізу.

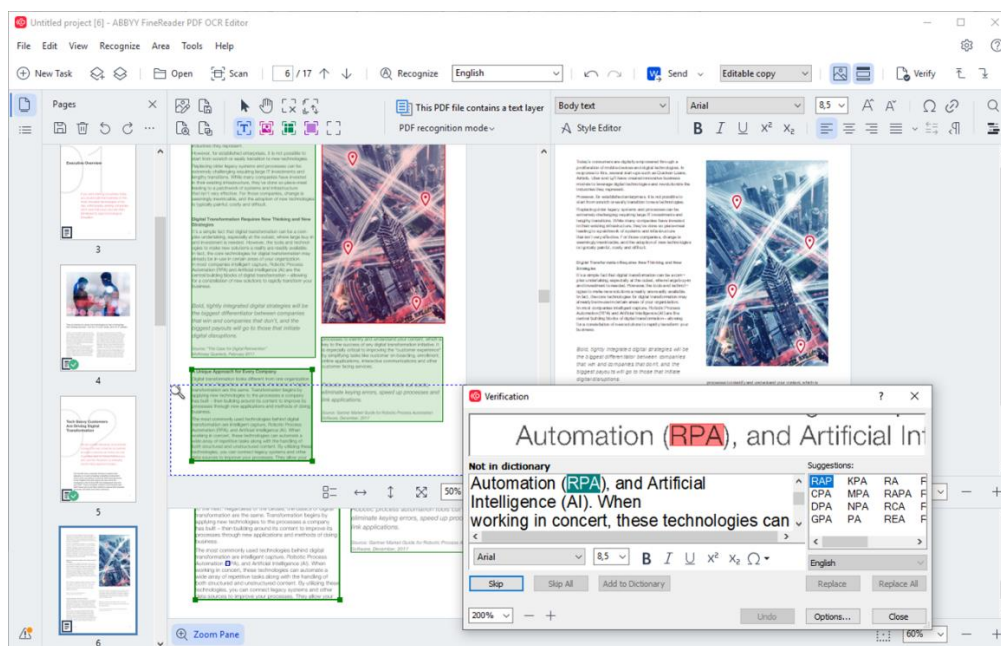


Рисунок 1.2 – Користувацький інтерфейс ABBYY FineReader

Крім того, ABBYY FineReader орієнтований на локальне використання, і хоча це є перевагою для автономної роботи, система менш гнучка для інтеграції з сучасними хмарними платформами або автоматизованими інструментами обліку документів. Розроблюваний у цій роботі продукт враховує необхідність легкої інтеграції з іншими системами, забезпечуючи підвищену ефективність у спеціалізованих процесах.

Ще одним недоліком є фокус ABBYY FineReader на розпізнаванні тексту як основної функції, тоді як переклад є лише допоміжним модулем, що може знижувати його ефективність у багатомовних завданнях порівняно зі спеціалізованими системами перекладу.

1.4.3 Сервіс перекладу Microsoft Translator

Microsoft Translator є частиною екосистеми Microsoft і пропонує потужні можливості для перекладу тексту, включаючи підтримку роботи з OCR через Azure

Cognitive Services. Сервіс дозволяє зчитувати текст із зображень і здійснювати переклад у реальному часі, що робить його ефективним для багатомовних документів (рисунок 1.3) [7].

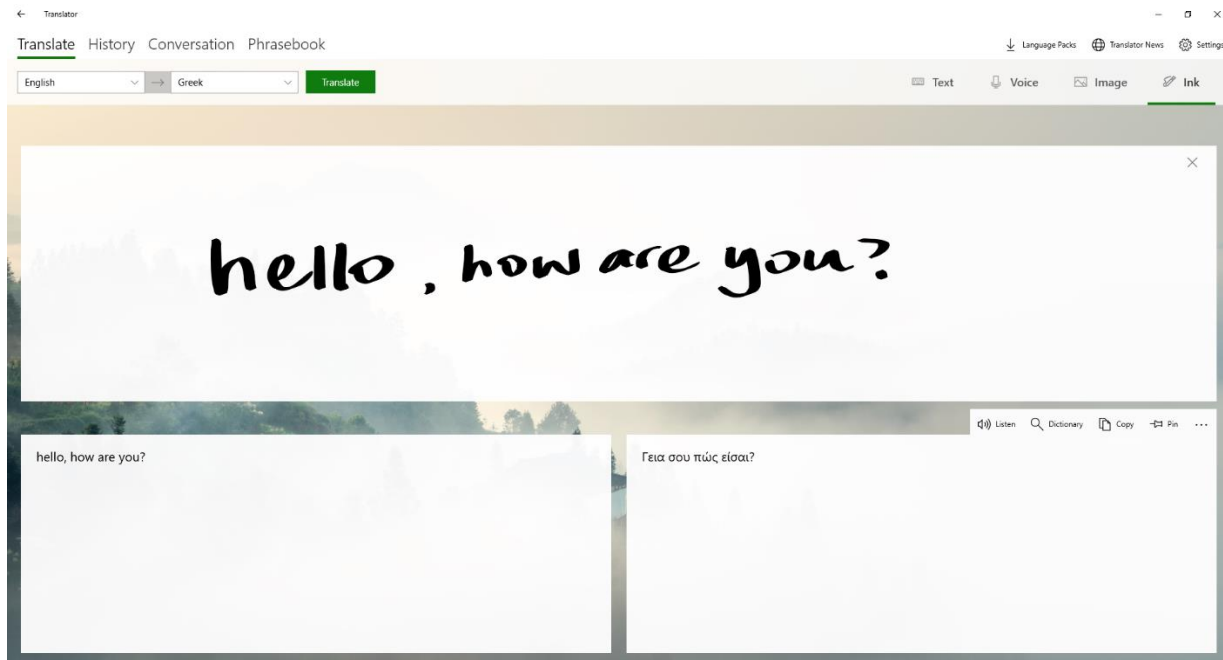


Рисунок 1.3 – Користувацький інтерфейс Microsoft Translator

Microsoft Translator, як і Google Translate, орієнтований на загальне використання, тому не має спеціалізованих функцій для ідентифікації та обробки іменованих сутностей. Це ускладнює роботу з документами, де важлива структурованість даних, наприклад, у фінансовій чи юридичній сфері.

Сервіс також залежить від постійного підключення до інтернету, оскільки більшість обчислень виконується у хмарі. Це обмежує його застосування в умовах, де доступ до мережі нестабільний або відсутній.

Ще одним обмеженням є складність у кастомізації процесів перекладу. Для завдань, що вимагають точного налаштування під конкретні мовні чи контекстуальні потреби, Microsoft Translator менш ефективний. У розроблюваній системі передбачено більшу гнучкість, що дозволить адаптувати алгоритми для більш спеціалізованих потреб користувачів.

1.5 Постановка задачі

Провівши аналіз існуючих рішень, можна прийти до висновку, що їх функціональність не є повною для специфічного роду задач, пов'язаних з перекладом документів. Значним мінусом кожної з них є те, що вони не надають можливості вести облік документів, а також ніяким чином не мають можливості залучити стороннього експерта – перекладача для допомоги з перекладом.

Таким чином головною задачею даної роботи стає розробка інформаційної системи, яка дозволить закрити прогалини у існуючих рішеннях та стане потужним інструментом для вирішення проблем бізнесу і не тільки, надавши можливість не тільки сканування документу, розпізнавання тексту, його переклад, а ще й надасть змогу залучити експерта до перекладу, а також вести облік документів до перекладу в цілому.

До програмного застосунку були висунуті наступні такі вимоги, як:

- можливість сканування фото документу;
- розпізнавання тексту з фото;
- переклад тексту на обрану мову;
- зберігання документу для подальших маніпуляцій;
- можливість залучити експерта-перекладача для редагування перекладу;
- визначення іменованих сутностей документу.

Для реалізації інформаційної системи необхідно виконати наступне:

- розробити алгоритм сканування документу;
- впровадити модель штучного інтелекту для розпізнавання тексту на зображенні;
- навчити та впровадити модель для розпізнавання іменованих сутностей;
- впровадити механізм перекладу тексту;
- розробити базу даних;
- розробити користувацький інтерфейс.

Очікується, що інформаційна система, яка розробляється, буде мати наступні результати:

- можливість легко та швидко перекласти текст з документа;
- функціональність для збереження та обліку документів;
- можливість запиту на переклад у експерта та перегляд його варіацій.

Таким чином, на основі поставленої задачі та вимог до програмного застосунку можна побудувати діаграму прецедентів (додаток А).

Діаграма прецедентів або ж варіантів використання (use-case diagram) відображає функціонал програмної системи, яка розробляється. Вона представляє відношення між акторами, які присутні в системі, та відповідними прецедентами.

Кожен з користувачів системи має свою функціональність, що визначається його роллю. Клієнт може створити персональний кабінет, що дає йому доступ до основних можливостей системи. Після створення кабінету клієнт може завантажувати документи для подальшої обробки. Окрім того, клієнт має можливість отримати різні варіанти перекладу, що створюються іншими користувачами, а саме – перекладачами.

Перекладачі мають можливість редагувати надані варіанти перекладу, щоб забезпечити їх якість та відповідність вимогам.

Адміністратор виконує ключові керуючі функції, зокрема призначення ролей користувачам.

Висновки до розділу 1

У даному розділі було розглянуто теоретичні та практичні аспекти, пов'язані з використанням сучасних технологій штучного інтелекту для автоматизації процесів розпізнавання та перекладу тексту. Проведений аналіз предметної області продемонстрував, що сучасні підходи, такі як оптичне розпізнавання символів і обробка природної мови, є ключовими технологіями для обробки текстової інформації. Також було розглянуто сучасні технології перекладу тексту, серед яких виділяються нейронні машинні перекладачі, що базуються на трансформерних архітектурах.

Огляд існуючих рішень дозволив ідентифікувати їхні сильні та слабкі сторони, зокрема, такі недоліки, як відсутність функціональності для обліку документів та залучення експертів до перекладу. Це підтвердило актуальність розробки спеціалізованої інформаційної системи, яка зможе задовольнити потреби бізнесу та приватних осіб, надаючи інструменти для автоматизації перекладу та обробки документів.

На основі проведеного аналізу було сформульовано завдання дослідження та визначено вимоги до розроблюваної системи. Подальша робота спрямована на створення архітектури програмного забезпечення, розробку алгоритмів і моделей для виконання поставлених задач, а також впровадження зручного інтерфейсу для кінцевих користувачів. Це дозволить забезпечити ефективне рішення для цифровізації, обліку та перекладу документів, яке зможе закрити прогалини, характерні для наявних рішень.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

Інформаційне забезпечення відіграє ключову роль у розробці будь-якого програмного продукту, особливо коли йдеться про системи, що базуються на технологіях штучного інтелекту та обробки тексту. Вибір інструментів та технологій на пряму впливає на ефективність та масштабованість програмного забезпечення в майбутньому.

Розглянемо набір інструментів, які використовуються для реалізації поставленої задачі, а також їхнє конкретне призначення у розробці системи. Такий аналіз дозволить зрозуміти, чому обрані саме ці технології та як вони сприяють досягненню поставленої мети.

2.1 Мова програмування Python

Python є однією з найпопулярніших мов програмування завдяки своїй простоті, зручності у використанні та широкому набору бібліотек, що робить її ідеальним вибором для розробки програмних продуктів у різних сферах, зокрема для роботи з даними, штучним інтелектом, обробкою тексту та зображень. В рамках даного проєкту Python був обраний як основна мова програмування завдяки своїй універсальності та здатності ефективно інтегрувати різні інструменти для реалізації складних задач.

Однією з ключових переваг Python є наявність великої кількості бібліотек, які спрощують вирішення конкретних завдань. Для роботи з оптичним розпізнаванням тексту на зображеннях була використана бібліотека PyTesseract, яка є інтерфейсом для популярної програми Tesseract OCR, що дозволяє зчитувати текст з відсканованих документів. За допомогою цієї бібліотеки було реалізовано ефективне розпізнавання тексту на зображеннях, що є одним з основних етапів процесу обробки документів.

Для обробки природної мови і визначення іменованих сутностей була використана бібліотека SpaCy, яка забезпечує потужні алгоритми для аналізу

тексту, зокрема для виділення сутностей, таких як імена, адреси, організації та інші важливі елементи інформації. Це дозволяє не лише здійснювати переклад тексту, а й аналізувати структуру документа та виявляти ключові елементи, що мають важливе значення для подальшої обробки та обліку.

Одним з основних інструментів для роботи з зображеннями є бібліотека OpenCV. Вона широко використовується для обробки зображень і надає потужні засоби для виконання операцій, таких як корекція контрасту, фільтрація, зміна розміру зображення, а також для підготовки зображень до подальшої обробки OCR. Завдяки OpenCV, вдалося ефективно обробляти та готувати зображення для точнішого розпізнавання тексту, що особливо важливо для зображень з різною якістю або складними фонами.

Також для створення веб-сервісу було використано фреймворк Flask, що дозволяє швидко та зручно створювати серверні додатки з мінімумом налаштувань. Flask дозволяє ефективно організувати комунікацію між фронтендом і бекендом програми, що є необхідним для інтерактивної роботи користувача з додатком.

Універсальність Python дозволяє використовувати його для інтеграції різних інструментів та технологій, що робить можливим автоматизацію процесів сканування, розпізнавання та перекладу документів. Крім того, Python активно підтримує роботу з базами даних, і для цього у даному проєкті використовується PostgreSQL – потужна реляційна база даних, що дозволяє зберігати і управляти великими обсягами інформації, такою як збережені документи та їхні переклади.

Таким чином, Python як основна мова програмування в даному проєкті забезпечує ефективну інтеграцію всіх необхідних технологій, дозволяючи швидко та безпечно реалізувати різноманітні функції та механізми.

2.2 Бібліотека OpenCV для обробки зображень

OpenCV (Open Source Computer Vision Library) – це бібліотека з відкритим вихідним кодом для обробки зображень і комп'ютерного зору, яка була розроблена для забезпечення можливості реалізації складних алгоритмів обробки зображень в

реальному часі. Вона є однією з найбільш популярних та потужних бібліотек у своїй галузі, використовується в багатьох наукових і комерційних проєктах для вирішення завдань, пов'язаних з аналізом зображень та відео.

OpenCV забезпечує широкий спектр інструментів для виконання різних операцій над зображеннями: від базових операцій (фільтрація, зміна розміру, корекція яскравості та контрасту) до складніших задач (виявлення контурів, відстеження руху, калібрування камер, розпізнавання об'єктів). Бібліотека підтримує різноманітні формати зображень і дозволяє працювати як з двовимірними зображеннями, так і з відео потоками [8].

Однією з найбільших переваг OpenCV є її здатність до обробки зображень в реальному часі. Бібліотека забезпечує високу продуктивність, що дозволяє ефективно виконувати складні алгоритми навіть на великих наборах зображень або в умовах обмежених ресурсів. OpenCV має добре розвинену документацію та спільноту, що робить її доступною для широкого кола розробників.

У контексті обробки документів, OpenCV використовується для попередньої обробки зображень перед застосуванням алгоритмів оптичного розпізнавання тексту. До основних етапів обробки, які можна реалізувати за допомогою OpenCV, відносяться:

- забезпечення оптимальної якості зображення для подальшого розпізнавання тексту, для якого застосовуються методи корекції контрасту, яскравості, а також фільтрація шуму, наприклад, використання фільтрів Гауса дозволяє знизити рівень шуму на зображенні, що значно покращує результати OCR;

- перетворення зображення у бінарну форму (чорно-біле зображення) дозволяє спростити задачу для OCR-алгоритму, оскільки контури тексту будуть чітко виділені на фоні, для цього OpenCV має кілька методів для бінаризації, таких як адаптивна бінаризація, що враховує локальні особливості зображення;

- важливим також є виділення контурів та ліній, так як на етапі підготовки зображення важливо виділити основні елементи, такі як поля документа, межі тексту та інші структурні особливості, для цього OpenCV має методи для виділення

контурів (наприклад, метод Canny), які дозволяють чітко окреслити ці елементи на зображенні;

– за допомогою морфологічних операцій, таких як розширення або ерозія, можна видаляти шум або, навпаки, зливати окремі частини зображення для покращення результатів розпізнавання, що дозволяє очистити зображення від дрібних дефектів, таких як дрібні точки або лінії, що можуть спотворювати результати OCR;

– виправлення перспективи та геометричних спотворень, так як документи можуть бути сфотографовані або відскановані під кутом, що призводить до геометричних спотворень, на щастя OpenCV дозволяє застосовувати методи корекції перспективи, такі як перетворення гомографії, щоб виправити подібні деформації та отримати зображення з прямими лініями і правильними пропорціями;

– виявлення текстових блоків і зон інтересу (ROI), так як іноді потрібно працювати лише з певними частинами зображення, наприклад, з текстовими блоками або конкретними елементами документа, для цього OpenCV надає методи визначення та виділення таких зон інтересу, що дозволяє знизити навантаження на систему і зробити обробку зображення більш цілеспрямованою.

Одним з основних переваг використання OpenCV є його здатність обробляти зображення на низькому рівні з максимальним контролем над процесами. Що дозволяє повноцінно налаштовувати обробку зображень відповідно до специфічних вимог проекту, таких як покращення якості сканованих документів або оптимізація процесу розпізнавання тексту.

2.3 PyTesseract для розпізнавання тексту

PyTesseract – це Python-обгортка для Tesseract, одного з найпопулярніших інструментів для оптичного розпізнавання тексту, розробленого Google. Ця технологія дозволяє автоматично витягувати текст з зображень, що особливо корисно для обробки сканованих документів або фотографій.

Алгоритм розпізнавання тексту має декілька основних етапів [9]:

а) на першому етапі PyTesseract аналізує все зображення цілком і визначає, чи містить воно текст, який може бути оброблений, що дозволяє розпізнати наявність текстових блоків і розділити зображення на окремі частини;

б) на другому етапі програма визначає блоки тексту на сторінці, які можуть бути абзацами або колонками, що мають логічно зв'язаний текст, також для кожного блоку PyTesseract зберігає контекст розташування тексту, що дозволяє правильно реконструювати зміст;

в) PyTesseract розбиває кожен блок тексту на окремі параграфи;

г) після поділу на блоки та параграфи PyTesseract визначає окремі рядки в кожному параграфі;

д) на останньому етапі PyTesseract розпізнає слова в кожному рядку, використовуючи алгоритми, що аналізують кожен символ і порівнюють його з навчальними даними Tesseract.

Таким чином, даний інструмент стає корисним для використання у подібних системах та значно спрощує процеси розпізнавання тексту.

2.4 Бібліотека SpaCy для визначення іменованих сутностей

SpaCy – це потужна бібліотека для обробки природної мови (NLP), яка дозволяє проводити глибокий лінгвістичний аналіз текстів та ефективно визначати іменовані сутності (NER). Вона підтримує багато мов і використовує переднавчені моделі для визначення та класифікації сутностей, таких як імена людей, організацій, географічні об'єкти, дати та числа.

Під капотом SpaCy використовує складні алгоритми машинного навчання, які базуються на глибоких нейронних мережах. Моделі, які використовуються для NER, навчалися на великих наборах даних, що дозволяє їм точно розпізнавати іменовані сутності в текстах. Під час аналізу тексту SpaCy спершу виконує токенизацію, розбиваючи текст на окремі токени (слова або фрагменти), після чого

використовуються моделі для визначення, чи є кожен токен частиною іменованої сутності. Це може бути особа, організація, географічна назва або інша категорія.

Процес розпізнавання сутностей у SpaCy полягає в кількох етапах. Спочатку текст токенізується, що дозволяє зменшити слова до їх базових форм. Потім на основі контексту моделі визначають, які з токенів належать до конкретних категорій іменованих сутностей, таких як імена, місця чи дати. Наприклад, фраза "Google Inc." буде розпізнана як організація, а "New York" – як географічна назва. Після цього кожна знайдена сутність позначається спеціальними мітками, які вказують на тип сутності [10].

Для інтеграції SpaCy в проєкті, необхідно спочатку завантажити відповідну модель для обраної мови. Бібліотека надає безліч моделей для різних мов, в тому числі для української. Після завантаження моделі, для аналізу тексту використовуються методи, які токенізують текст та здійснюють класифікацію іменованих сутностей. Результати можуть бути виведені у вигляді списку сутностей з їх типами та позиціями в тексті, що дозволяє подальше використання цієї інформації для зберігання, класифікації або перекладу.

У даному випадку SpaCy використовується для попереднього аналізу тексту і виділення важливих даних, таких як назви компаній, людей чи місць, що можуть потребувати особливого перекладу або обробки. Визначення іменованих сутностей допомагає не тільки покращити якість перекладу, але й структурувати дані, що є важливим для обліку документів.

2.5 Веб-фреймворк Flask

Flask – це фреймворк для створення веб-додатків на Python, який дозволяє швидко і ефективно побудувати серверну частину додатка. Його основною перевагою є простота використання та висока гнучкість, що робить Flask ідеальним для невеликих та середніх проєктів. Завдяки цьому фреймворку можна легко створити сервер, який обробляє HTTP-запити, взаємодіє з базою даних та виконує бізнес-логіку.

Під капотом Flask працює на основі WSGI (Web Server Gateway Interface), що дозволяє інтегрувати його з будь-яким сервером, підтримуючим цей стандарт. Flask обробляє запити через маршрути, які прив'язані до певних функцій (view-функцій), які виконують бізнес-логіку, обробляють введені дані, звертаються до бази даних і повертають відповіді користувачу. Запити можуть бути різних типів, таких як GET, POST, PUT, DELETE, і кожен тип має свій обробник [11].

Flask можна використовувати для організації комунікації між клієнтською частиною та сервером, а також для обробки запитів користувача. Наприклад, користувач може завантажити документ на сервер, ініціювати розпізнавання тексту, отримати результат і навіть запросити допомогу перекладача. Flask забезпечить передачу даних між фронтендом і бекендом, надасть доступ до необхідних API, обробить запити на переклад, а також зберігатиме документи для подальших маніпуляцій.

Процес взаємодії з Flask виглядає наступним чином: користувач надсилає HTTP-запит на сервер (наприклад, через веб-форму або через API-запит), Flask приймає запит і викликає відповідну функцію, що обробляє дані. Ця функція може взаємодіяти з іншими частинами системи, такими як модель для розпізнавання тексту або модель перекладу, після чого сервер повертає відповідь користувачу у вигляді HTML-сторінки або JSON-об'єкта.

Завдяки Flask серверну частину можна гнучко налаштовувати та додавати нові функції.

2.6 СУБД PostgreSQL

PostgreSQL – це об'єктно-реляційна система управління базами даних, яка широко використовується для зберігання і обробки великих обсягів даних. Вона є однією з найбільш надійних СУБД, яка підтримує як SQL, так і функціональні можливості для роботи з NoSQL даними. Це робить PostgreSQL ідеальним вибором для застосунків, що потребують високої надійності, продуктивності та розширюваності.

Однією з головних переваг PostgreSQL є її підтримка транзакцій, що дозволяє забезпечити цілісність даних при одночасному виконанні декількох операцій. Завдяки використанню ACID-принципів (атомарність, консистентність, ізолюваність, довговічність) PostgreSQL гарантує, що навіть у разі помилок чи непередбачуваних збоїв система не втратить або не пошкодить важливі дані [12].

У контексті даного проєкту PostgreSQL є ідеальним варіантом для зберігання різноманітних даних, пов'язаних із документами, такими як текстові дані, метадані про документи (дата сканування, автор, тип документа) і результати аналізу (розпізнаний текст, перекладений текст, іменовані сутності). У базі даних можна створити таблиці для організації цих даних у структурованому вигляді, забезпечуючи легкий доступ до них та можливість виконання складних запитів.

PostgreSQL також підтримує функціональність для роботи з великими обсягами даних, що робить її відмінним вибором для таких завдань, як зберігання сканованих зображень документів (у вигляді бінарних даних або посилань на файли в файловій системі) і текстових даних, що отримуються після обробки з використанням OCR та перекладу. Водночас система підтримує індексацію для швидкого пошуку, що є важливим аспектом для роботи з великими масивами даних.

Інтеграція PostgreSQL з іншими частинами системи через бібліотеки Python, такі як `psycopg2` або `SQLAlchemy`, дозволяє організувати взаємодію з базою даних в зручний та ефективний спосіб. Це забезпечує безпечну та ефективну роботу з даними на всіх етапах.

2.7 Сервіс машинного перекладу DeepL

DeepL – це потужний інструмент для машинного перекладу, який став популярним завдяки своїй високій якості перекладу, особливо в порівнянні з іншими подібними сервісами. У основі DeepL лежать передові моделі глибокого навчання, що дають змогу досягати високого рівня точності та контекстуального

розуміння тексту. Для зручності інтеграції DeepL надає API, яке дозволяє користувачам інтегрувати переклад тексту у свої додатки.

DeepL API підтримує переклад між багатьма мовами і дозволяє здійснювати переклад текстів різної довжини, починаючи від коротких фрагментів і закінчуючи цілими документами. Усі дані, що передаються через API, обробляються на серверах DeepL, що дозволяє зекономити ресурси та не навантажувати сервер користувача додатковими обчисленнями [13]. API забезпечує високу швидкість перекладу, що є критично важливим для роботи з великими обсягами даних.

Інтеграція DeepL у проект дає можливість ефективно автоматизувати процес перекладу текстів, зокрема документів, без необхідності розробляти власну модель для машинного перекладу. Через зручний інтерфейс API, можна легко реалізувати переклад тексту на обрану мову в реальному часі, що значно підвищує зручність роботи.

Інтеграція API DeepL в рамках цього проекту дозволяє користувачам отримати точний переклад текстів із високою швидкістю, зберігаючи при цьому легкість у використанні.

Висновок до розділу 2

Було детально розглянуто інструменти, що використовуються для реалізації проекту, зокрема Python, OpenCV, PyTesseract, SpaCy, Flask, PostgreSQL та DeepL. Кожна з цих технологій має свої особливості і функціональні можливості, які дозволяють ефективно вирішувати різноманітні завдання, пов'язані з обробкою документів, розпізнаванням тексту, перекладом та збереженням даних.

Python виступає основною мовою програмування завдяки своїй універсальності та зручності для роботи з бібліотеками машинного навчання і обробки тексту. OpenCV забезпечує ефективну обробку зображень, необхідну для попередньої підготовки документів до розпізнавання тексту, а PyTesseract реалізує процес OCR для перетворення зображень у текст. Використання SpaCy дає можливість класифікувати іменовані сутності, що є важливою частиною аналізу

документів. Flask надає платформу для побудови серверної частини додатку, що дозволяє організувати взаємодію між користувачем та системою. Водночас, PostgreSQL забезпечує надійне зберігання даних і облік документів, що дозволяє зберігати всю необхідну інформацію для подальшої обробки та аналізу. Нарешті, API DeepL забезпечує високу якість перекладу, що робить систему зручною та корисною для кінцевого користувача.

Комбінація цих інструментів дає змогу створити ефективну, масштабовану та надійну систему для обробки, перекладу та збереження документів, що повністю відповідає вимогам, поставленим перед проектом. Кожен з інструментів був обраний на основі його здатності задовольнити специфічні потреби системи.

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

3.1 Архітектура програмного застосунку

Архітектура інформаційної системи базується на модульному підході, де кожен компонент відповідає за окрему функціональність. Система складається з кількох ключових модулів, що взаємодіють між собою (додаток Б).

Перший модуль, а саме сканування, відповідає за обробку фізичних копій документів. Користувач завантажує зображення документа через інтерфейс додатку, і цей модуль ініціює процес перетворення зображення в цифровий формат для подальшої обробки. Модуль сканування може включати алгоритми, які покращують якість зображення та виправляють перекося, вирівнюючи перспективу, що допомагає отримати чітке зображення для подальшого розпізнавання.

Далі йде модуль з розпізнавання тексту, що використовує технології OCR для перетворення тексту на зображенні в машинно зчитуваний формат. У цьому модулі застосовуються алгоритми для виявлення і класифікації символів, слів та рядків. Також він відповідальний за аналіз розпізнаного тексту з метою виявлення важливих інформаційних елементів, таких як імена, адреси, номери телефонів, дати тощо, використовуючи алгоритми машинного навчання та природної мови, для класифікації сутностей, що дозволяє виділити структуровану інформацію для подальшого перекладу або збереження в базі даних.

Модуль перекладу дозволяє здійснювати переклад розпізнаного тексту на вибрану користувачем мову. Він взаємодіє з API DeepL для перекладу тексту.

Далі йде модуль зберігання документів, який зберігає документи та їх відповідні метадані в базі даних. Модуль дозволяє відслідковувати статуси документів, а також забезпечує можливість пошуку та сортування. Вся інформація про документ, починаючи з оригіналу та закінчуючи перекладеним текстом, зберігається в централізованому сховищі.

Користувацький інтерфейс є важливою частиною архітектури. Це модуль, через який користувач взаємодіє з усіма іншими частинами системи. Інтерфейс

дозволяє користувачу завантажувати документи, переглядати результати, коригувати переклад за допомогою залучених експертів, а також переглядати власні документи.

Таким чином, система побудована на основі чіткої розподіленої архітектури, де кожен модуль виконує свою специфічну функцію, і взаємодіє з іншими для досягнення кінцевої.

3.1.1 Потоки даних у системі

Для опису руху та обробки даних в системі використовується діаграма потоків даних, яка надає можливість візуально представити всі процеси з точки зору обробки інформації. Ця діаграма дозволяє продемонструвати функціональні залежності між різними елементами системи, включаючи вхідні, вихідні та проміжні дані, а також внутрішні сховища інформації.

DFD складається з кількох ключових компонентів. Зовнішні сутності – це об'єкти або користувачі, які взаємодіють із системою, створюючи або отримуючи дані. Процеси представляють собою функціональні модулі, що перетворюють вхідні дані на вихідні, виконуючи обчислення чи інші операції. Сховища даних є місцями для пасивного зберігання інформації, яка може бути використана системою у подальших операціях. Потоки даних виконують роль зв'язків, забезпечуючи передачу інформації між зовнішніми сутностями, процесами та сховищами [14].

За допомогою DFD можна побудувати цілісну модель обробки даних у програмному застосунку, виявляючи взаємозв'язки між його компонентами та оптимізуючи роботу системи. На додатку В представлено діаграму потоків даних, яка деталізує всі етапи руху інформації в межах розробленого програмного забезпечення.

Процес починається з користувача, який завантажує фотографію документа. Завантажене зображення проходить етап кадрування, де виділяються ключові

області для подальшої обробки. Кадроване зображення надходить до модуля розпізнавання тексту, який перетворює візуальну інформацію на текст.

Після отримання тексту система визначає мову документа для налаштування процесу перекладу. На цьому етапі залучається штучний інтелект, який виконує переклад тексту. Оригінальний текст разом із перекладом зберігаються для подальшого перегляду. Користувач отримує можливість переглядати завантажені документи, їхні переклади, а також різні варіанти перекладу, створені перекладачем.

Система також дозволяє переглядати та редагувати запити на переклад, у тому числі вносити зміни до результатів перекладу. Усі дані, такі як список документів, перекладів і запитів, структуруються та доступні для взаємодії.

Ця діаграма відображає структуровану послідовність взаємодії між компонентами та користувачем, починаючи з моменту завантаження документа і закінчуючи отриманням готового перекладу.

3.1.2 Бізнес-процеси програмного застосунку

Для відображення бізнес-процесів у системі широко використовується стандарт BPMN, який дозволяє описати роботу системи за допомогою зручних та інтуїтивно зрозумілих діаграм. Цей стандарт включає набір графічних елементів, що дають змогу точно визначити та візуалізувати складні семантичні конструкції. Моделювання в BPMN здійснюється з використанням базових діаграм, що складаються з кількох основних елементів, які описують загальну логіку окремого процесу. Кожен процес починається з початкової події та завершується кінцевою. Завдяки своїй простоті та універсальності, BPMN дозволяє як технічним спеціалістам, так і користувачам без спеціалізованих знань, швидко розуміти та аналізувати логіку процесу [15]. Таким чином, BPMN є ефективним інструментом для візуалізації та розуміння бізнес-процесів у різноманітних системах. BPMN діаграма для застосунку, що розробляється, зображена на додатку Г.

Перший процес відноситься до авторизації користувача. Користувач спочатку заходить в обліковий запис або реєструє новий, після чого отримує доступ до кабінету користувача, де можна працювати з документами.

Наступний процес – це обробка документів для перекладу. Користувач може завантажити новий документ або відкрити існуючий. Потім система визначає іменовані сутності в тексті, зічитає текст з зображення та здійснює переклад. Якщо користувач потребує додаткової допомоги, він може надіслати запит перекладачеві.

Останній процес передбачає перевірку перекладу експертом. Перекладач ознайомлюється з документом і вирішує, чи приймати його до роботи або відмовитися. Якщо документ прийнято до роботи, перекладач вносить необхідні корективи, після чого зміни зберігаються і відправляються замовнику.

У цій діаграмі чітко відображений шлях від авторизації користувачем до етапу завершення ним процесу перекладу документу.

3.2 Моделювання поведінки системи за допомогою діаграми активності

Діаграма активності відображає динамічні процеси, які відбуваються в системі під час виконання певних функцій або операцій. Вона описує послідовність кроків, умовні переходи та взаємодію між різними компонентами системи в рамках певної діяльності [16]. Цей тип діаграми дозволяє чітко уявити, як виконуються різні етапи роботи системи, що допомагає зрозуміти логіку процесів та оптимізувати їх.

У випадку великої кількості учасників чи складних процесів, діаграма активності допомагає чітко визначити відповідальність кожного учасника в межах виконання конкретних етапів роботи. Завдяки цьому зменшується ризик непорозумінь або неправильного виконання задач, що може призвести до затримок або помилок. Також така діаграма забезпечує прозорість, що допомагає не тільки розробникам, але й кінцевим користувачам або замовникам краще розуміти функціонування системи, сприяючи більш ефективному її використанню.

Діаграма активності для даного застосунку зображена на додатку Д. Ця діаграма описує процес роботи користувача з документами в системі. Процес починається з авторизації користувача. Після цього користувач переходить до перегляду бібліотеки документів, де має можливість завантажити новий документ або переглянути вже існуючий. Якщо документ новий, система дозволяє користувачу завантажити його зображення та вибрати мову оригіналу для подальшого перекладу. Якщо документ вже існує, користувач може безпосередньо перейти до вибору мови перекладу і почати процес перекладу тексту з використанням штучного інтелекту.

Після завершення перекладу тексту, користувач має можливість внести правки в переклад. Правки можуть бути надіслані назад користувачеві для підтвердження чи перегляду. Якщо ж переклад потребує коригування, система генерує запит на виправлення від перекладача. Завершення процесу супроводжується збереженням даних, після чого користувач може переглядати всі внесені правки. У кінці процесу система завершує обслуговування користувача, зберігаючи всі зміни та готовність до подальшої роботи.

3.3 Діаграма послідовності взаємодії компонентів системи

Діаграма демонструє, як різні елементи системи: користувачі, інтерфейси та внутрішні модулі, – взаємодіють між собою шляхом обміну повідомленнями в певному порядку.

Основна задача такої діаграми – це показати хронологічний порядок взаємодії між компонентами для виконання певного сценарію роботи системи. Вона надає можливість наочно уявити, які саме модулі залучені до виконання операцій, у якому порядку викликаються їх функції, та які дані передаються між учасниками цього процесу.

Діаграма послідовності має кілька важливих задач в рамках розробки інформаційної системи. По-перше, вона дозволяє детально проаналізувати логіку процесів і виявити потенційні недоліки у взаємодії компонентів ще на етапі

проектування. По-друге, така діаграма є корисним інструментом для оптимізації роботи системи, оскільки вона дозволяє оцінити ефективність сценаріїв взаємодії.

На додатку Е представлено діаграму послідовності, яка моделює процес взаємодії користувачів із системою. Діаграма показує взаємодію між різними модулями системи для обробки документа. Користувач починає процес, завантажуючи документ через користувацький інтерфейс, після чого інтерфейс передає запит на ініціювання сканування до модуля сканування. Модуль сканування покращує якість зображення і передає його далі для розпізнавання тексту в модуль розпізнавання. Він в свою чергу обробляє зображення, розпізнає текст та витягує з нього іменовані сутності, зберігаючи результат у модулі зберігання для подальшої роботи.

Наступним етапом є передача розпізнаного тексту до модуля перекладу, де він перекладається. Перекладений текст зберігається в базі даних, а модуль зберігання підтверджує успішне збереження. Користувацький інтерфейс отримує документ з бази даних для перегляду, і користувач може ознайомитись із його вмістом.

Якщо користувач вирішує редагувати переклад, він запитує зміну через інтерфейс, що надсилає запит до модуля перекладу для відображення редагованого тексту. Після внесення змін користувач зберігає редагований переклад, який передається до модуля зберігання для оновлення в базі даних. Зберігання підтверджує успішне оновлення, завершуючи весь процес обробки документа.

3.4 Програмний модуль для сканування документів

Сканування документів є критично важливим етапом роботи з текстовою інформацією, яка міститься на зображеннях. Для цього використовується програмний модуль, що базується на бібліотеці OpenCV та алгоритмах обробки зображень. Основні завдання модуля – підготовка зображення, виправлення перспективи та покращення якості для подальшого розпізнавання тексту.

Алгоритм сканування зображення документа (рисунок 3.1), реалізований у програмному модулі, передбачає кілька етапів обробки для отримання чітко вирівняного і коректно відсканованого зображення.

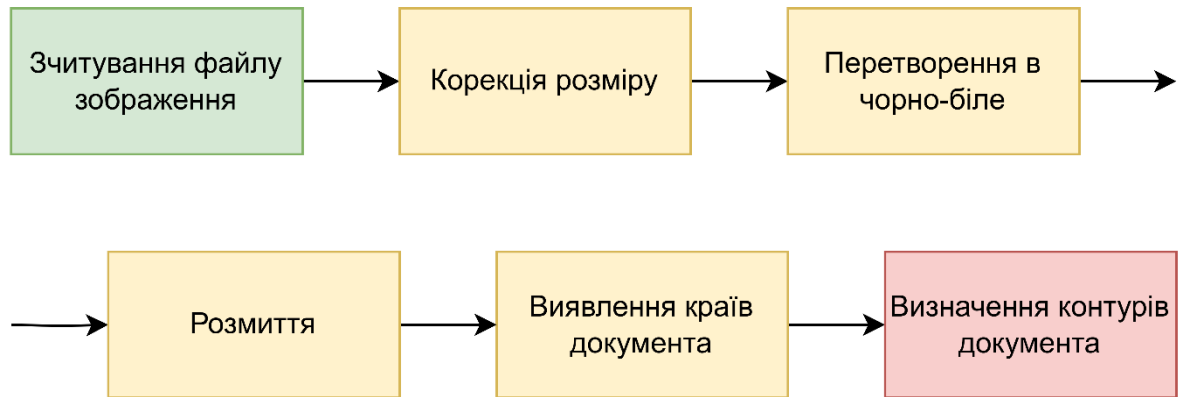


Рисунок 3.1 – Блок-схема алгоритму сканування зображення документа

Спершу зображення зчитується з файлу, після чого алгоритм переходить до корекції розміру зображення. Зображення зменшується до стандартного розміру, що дозволяє зменшити час обробки в подальших етапах. Після цього застосовується корекція яскравості та контрасту зображення. Наступним етапом є перетворення зображення в чорно-біле. Чорно-біле зображення є необхідним для більш точної детекції контурів, а також для спрощення наступних кроків обробки.

Після перетворення в чорно-біле зображення до нього застосовується розмиття, що дозволяє зменшити шум і сприяє більш точному виявленню країв на наступному етапі.

Далі до зображення застосовується алгоритм виявлення країв за допомогою методу Кані. Цей метод знаходить різкі зміни інтенсивності пікселів на зображенні, що дозволяє визначити контури об'єктів на фото.

На наступному етапі алгоритм шукає контури на зображенні, що дозволяє визначити найбільш виразні контури, які є частинами об'єктів на зображенні. Оскільки документ має прямокутну форму, алгоритм сортує контури за площею і вибирає найбільший контур, вважаючи його можливим контуром документа.

Якщо виявлений контур має чотири вершини, що є типовим для прямокутного документа, алгоритм вважає його правильним і переходить до

наступного етапу. Цей етап включає визначення чотирьох точок контурів, які використовуються для виправлення перспективи зображення.

3.5 Розпізнавання тексту та визначення іменованих сутностей

Для задачі розпізнавання використовується бібліотека Pytesseract. Як результат попереднього етапу ми отримуємо текстові дані, які перед переходом до наступних операцій, потрібно підготувати, прибравши некоректні символи і тому подібні елементи.

Розглянемо архітектуру процесу тренування моделі з розпізнавання іменованих сутностей (рисунок 3.2).

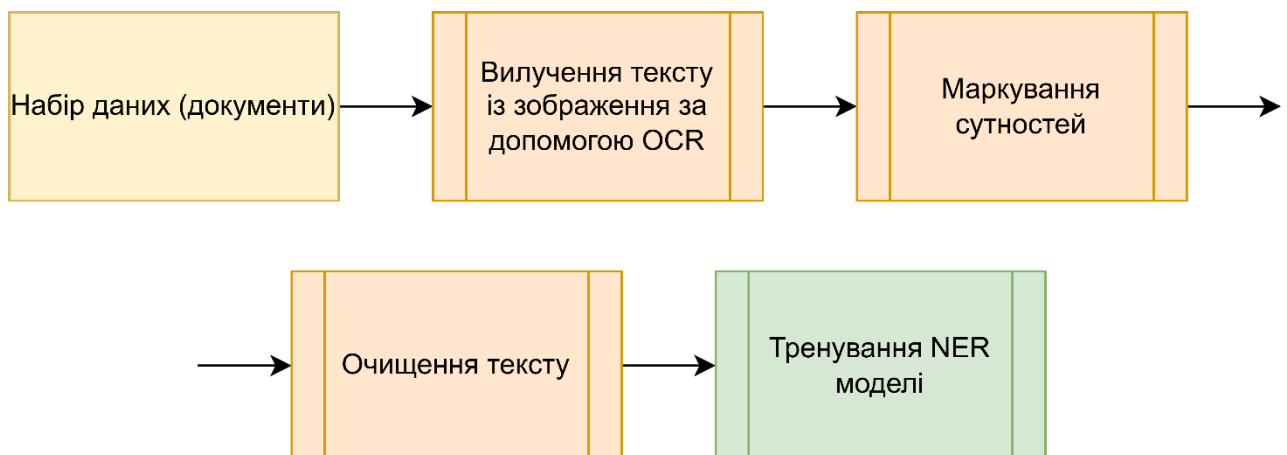


Рисунок 3.2 – Архітектура тренування NER моделі

Для початку необхідно зібрати якомога більше даних, з яких ми витягнемо текст для подальшої роботи. Потім потрібно виконати процес маркування. Цей процес є трудомістким і потребує дуже ретельного підходу. Оскільки це ручний процес, існує великий ризик помилок, тому важливо бути максимально уважним під час маркування. Коли маркування завершено, необхідно провести процес очищення даних. І нарешті, ми передаємо ці дані моделі Spacy та навчаємо модель розпізнавання іменованих сутностей. Таким чином модель з розпізнавання іменованих сутностей буде готова до подальшого використання.

На етапі підготовки та обробки даних основною метою є створення якісного набору для тренування моделі. Для цього спершу збирається набір зображень (для навчання використовувались зображення візитних карток), які слугують основою для подальшого вилучення тексту. Щоб конвертувати текст із зображень у формат, придатний для машинної обробки, використовується Pytesseract. Він дозволяє перетворювати текст із кожного зображення у цифровий вигляд, що забезпечує отримання необхідних текстових даних для подальшої обробки. Процес обробки зображення з використанням Pytesseract можна поділити на декілька етапів: сторінка, блок, параграф, строчка, слово та літера. Загалом результат обробки зображення представлений на рисунку 3.3.

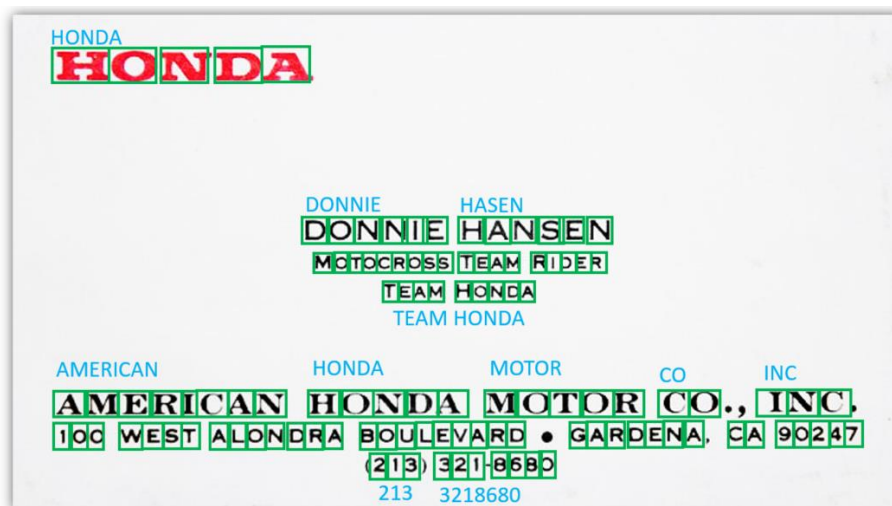


Рисунок 3.3 – Результат обробки зображення з використанням Pytesseract

Після вилучення тексту виконується етап базового очищення, що включає видалення зайвих символів, виправлення помилок та стандартизацію форматів. Очищення є важливим кроком, адже від якості підготовлених даних залежить точність майбутньої моделі. Після очищення текстові дані організуються у форматі CSV, де кожне слово або токен асоціюється з відповідним ім'ям файлу зображення. Це дозволяє зберегти зв'язок між текстом і зображенням, з якого його було отримано, що є важливим для наступних етапів.

Щоб забезпечити якісне навчання моделі, необхідно промаркувати дані, і для цього використовується схема BIO (Begin, Inside, Outside) тегування, що дозволяє

позначати, чи є слово початком сутності, всередині сутності чи поза сутністю (рисунок 3.4).

Text	Label
In	O
Beirut	B-geo
,	O
a	O
string	O
of	O
officials	O
voiced	O
their	O
anger	O
,	O
while	O
at	O
the	O
United	B-org
Nations	I-org

Рисунок 3.4 – Приклад ВІО-тегування

Основними сутностями, які виділяються у тексті, є ім'я особи, посада, організація, телефон, електронна пошта та вебсайт/URL. Процес ВІО-тегування здійснюється вручну, що забезпечує високу точність маркованих даних і створює еталонний набір для тренування та оцінки моделі. Таким чином, після завершення всіх етапів підготовки, включаючи збір, вилучення, очищення та маркування, набір даних стає готовим для навчання моделі.

Для обробки набору даних сперше виконується завантаження позначених даних із CSV або Excel файлів за допомогою команди `open` в Python, що дозволяє отримати доступ до даних для подальшої обробки. Після завантаження даних вони конвертуються у датафрейм, що надає представлення для зручнішої маніпуляції та підготовки даних до навчання моделі.

Після чого проводиться очищення тексту, під час якого видаляються зайві пробіли та спеціальні символи, які не є релевантними для моделі.

Наступний етап – це конвертація даних у формат навчання Spacy, спеціально розроблений для задач розпізнавання сутностей. У форматі навчання для кожного тексту візитної картки наведено оригінальний текст та список сутностей, де кожна сутність представлена початковим та кінцевим індексами, а також відповідним тегом (рисунок 3.5). Цей формат є оптимальним для NER у Spacy, і тому всі текстові дані конвертуються відповідно до нього.

Після завершення цих операцій виконується поділ даних на навчальний та тестовий набори для подальших маніпуляцій з NER моделлю.

```
[(' 040-4852 8881, 90309 52549 fi /laurelsoverseaseducation &€@ laurels overseas educational consultancy pvt. ltd. sea u.k australia canada ireland www.laurelsoverseaseducation.com info@laurelsoverseaseducation.com ',
 {'entities': [(2, 10, 'B-PHONE'),
 (11, 16, 'I-PHONE'),
 (17, 22, 'B-PHONE'),
 (23, 28, 'I-PHONE'),
 (63, 70, 'B-ORG'),
 (71, 79, 'I-ORG'),
 (80, 91, 'I-ORG'),
 (92, 103, 'I-ORG'),
 (104, 108, 'I-ORG'),
 (109, 113, 'I-ORG'),
 (147, 171, 'B-WEB'),
 (172, 197, 'B-EMAIL')]}],

 ('john smith marketing manager web www.psdgraphics.com phone 123-456-7890 mail email@psdgraphics.com ',
 {'entities': [(0, 4, 'B-NAME'),
 (5, 10, 'I-NAME'),
 (11, 20, 'B-DES'),
 (21, 28, 'I-DES'),
 (33, 52, 'B-WEB'),
 (59, 71, 'B-PHONE'),
 (77, 98, 'B-EMAIL')]}]
]
```

Рисунок 3.5 – Формат даних для навчання

З підготовленими даними для навчання можна розпочати тренування моделі за допомогою Spacy.

Для цього отримуємо компонент `ner` із завантаженого пайплайну Spacy. Потім визначаємо кількість ітерацій тренування (`n_iter`). На кожній ітерації перебираємо навчальні дані, створюючи об'єкт `Doc` із тексту, та конвертуємо анотації у об'єкт `Example`. Використовуємо метод `nlp.update`, щоб оновити модель NER за допомогою навчальних прикладів (рисунок 3.6).

```

ner = nlp.get_pipe("ner")
n_iter = 10

for _ in range(n_iter):
    for text, annotations in training_data:
        doc = nlp.make_doc(text)
        example = spacy.training.Example.from_dict(doc, annotations)
        nlp.update([example], drop=0.5)

```

Рисунок 3.6 – Тренування моделі

3.6 Програмний модуль для перекладу тексту

Модуль для перекладу тексту виконує трансформацію текстових даних з однієї мови на іншу, використовуючи API сервісу DeepL. Це потужний інструмент, що базується на нейронних мережах і здатний забезпечувати переклади з високою точністю, враховуючи контекст тексту. Такий підхід дозволяє інтегрувати процес перекладу в систему обробки документів, оптимізуючи її функціональність.

Алгоритм модуля починається з підготовки параметрів запити. До них належать текст, який необхідно перекласти, мова оригіналу та цільова мова. Також передається ключ доступу до API, що ідентифікує клієнта і дозволяє виконати авторизований запит до сервера DeepL. Після цього параметри передаються за допомогою HTTP-запиту методом POST.

Отриманий від сервера результат обробляється модулем, і у випадку успішного виконання запити (код відповіді 200) із JSON-об'єкта витягується перекладений текст. У разі виникнення помилки система повертає відповідний код помилки та її опис, що дозволяє швидко локалізувати проблему. Готовий результат повертається основному модулю, де він може бути використаний для подальшого збереження, відображення або передачі іншій частині системи.

Для реалізації HTTP-запиту та обробки відповіді використовується бібліотека requests, яка дозволяє легко взаємодіяти з API. Перетворення тексту відповіді у формат JSON здійснюється стандартними методами цієї бібліотеки. Гнучкість API DeepL дозволяє працювати з широким спектром мов і забезпечувати високу продуктивність системи навіть при обробці великих обсягів тексту.

Завдяки використанню DeepL API цей модуль демонструє високу ефективність у розв'язанні задач перекладу тексту. Нейронні мережі, на яких базується сервіс, забезпечують точність і врахування контексту, що особливо важливо при роботі з текстами, насиченими специфічною термінологією або складними конструкціями. У результаті система отримує якісний інструмент, що не лише полегшує роботу з текстами, а й значно покращує загальну продуктивність.

3.7 Архітектура бази даних

Архітектура бази даних була спроектована для задоволення основних потреб системи, що розробляється. Вона забезпечує зберігання інформації про користувачів, їхні ролі, завантажені документи, запити на переклад, а також виправлення тексту перекладу. Такий підхід дозволяє системі бути гнучкою та розширюваною, підтримуючи функціонал управління документами, призначення ролей користувачів і обробки запитів перекладу.

База даних містить декілька таблиць, кожна з яких виконує конкретну функцію (додаток Ж).

Перша таблиця, Users, призначена для зберігання даних про користувачів системи. У ній містяться такі поля, як унікальний ідентифікатор користувача UserId, ім'я користувача Username, хеш пароля PasswordHash для безпечного зберігання паролів, а також електронна адреса Email, яка є унікальною для кожного користувача. Ця таблиця є основою для ідентифікації користувачів і контролю доступу до системи (таблиця 3.1).

Таблиця 3.1 – Таблиця БД Users

Поле	Тип даних	Опис
UserId	INT	Ідентифікатор користувача (PRIMARY KEY, AUTO_INCREMENT)
Username	NVARCHAR(100)	Ім'я користувача
PasswordHash	NVARCHAR(255)	Хеш пароля користувача
Email	NVARCHAR(255)	Електронна пошта користувача (UNIQUE)

Друга таблиця, Roles, дозволяє визначати ролі користувачів. Наприклад, це можуть бути ролі звичайного користувача або перекладача. Поле RoleName забезпечує унікальність кожної ролі. Ця таблиця дозволяє легко масштабувати систему, додаючи нові ролі, якщо це буде потрібно (таблиця 3.2).

Таблиця 3.2 – Таблиця БД Roles

Поле	Тип даних	Опис
RoleId	INT	Ідентифікатор ролі (PRIMARY KEY, AUTO_INCREMENT)
RoleName	NVARCHAR(50)	Назва ролі (UNIQUE)

Для реалізації зв'язків між користувачами та їхніми ролями створено таблицю UserRoles (таблиця 3.3). Вона дозволяє встановлювати багато-до-багатьох зв'язок між користувачами та ролями, що забезпечує гнучкість у визначенні прав доступу. Наприклад, один користувач може мати одразу кілька ролей, залежно від його завдань у системі.

Таблиця 3.3 – Таблиця БД UserRoles

Поле	Тип даних	Опис
UserRoleId	INT	Ідентифікатор зв'язку (PRIMARY KEY, AUTO_INCREMENT)
UserId	INT	Ідентифікатор користувача (FOREIGN KEY, посилання на Users)
RoleId	INT	Ідентифікатор ролі (FOREIGN KEY, посилання на Roles)

Завантажені користувачами документи зберігаються в таблиці Documents (таблиця 3.4). У цій таблиці відображаються такі дані, як ідентифікатор документа DocumentId, ідентифікатор користувача, який завантажив документ UserId, назва файлу FileName, шлях до файлу FilePath, дата завантаження UploadedAt, а також мова документа Language. Ця структура дозволяє зручно організувати бібліотеку файлів користувача.

Таблиця 3.4 – Таблиця БД Documents

Поле	Тип даних	Опис
DocumentId	INT	Ідентифікатор документа (PRIMARY KEY, AUTO_INCREMENT)
UserId	INT	Ідентифікатор користувача (FOREIGN KEY, посилання на Users)
FileName	NVARCHAR(255)	Назва файлу
FilePath	NVARCHAR(500)	Шлях до файлу
UploadedAt	DATETIME	Дата завантаження документа (DEFAULT GETDATE())
Language	NVARCHAR(50)	Мова документа

Таблиця TranslationRequests обробляє запити на переклад (таблиця 3.5). У ній зберігається інформація про документ, для якого створено запит, хто створив запит, і, якщо необхідно, кому він призначений. Поле Status вказує на поточний стан запиту, наприклад, "Очікується" або "Завершено". Ця таблиця дозволяє відстежувати та керувати процесом перекладу документів.

Таблиця 3.5 – Таблиця БД TranslationRequests

Поле	Тип даних	Опис
RequestId	INT	Ідентифікатор запиту (PRIMARY KEY, AUTO_INCREMENT)
DocumentId	INT	Ідентифікатор документа (FOREIGN KEY, посилання на Documents)
RequestedBy	INT	Ідентифікатор користувача, який зробив запит (FOREIGN KEY, посилання на Users)
AssignedTo	INT	Ідентифікатор користувача, який виконує переклад (FOREIGN KEY, посилання на Users)
Status	NVARCHAR(50)	Статус запиту (за замовчуванням 'Pending')
CreatedAt	DATETIME	Дата створення запиту (DEFAULT GETDATE())
CompletedAt	DATETIME	Дата завершення перекладу (NULL, якщо не завершено)

Для збереження виправлених текстів створено таблицю TextCorrections (таблиця 3.6). Вона зберігає текст після внесення правок, а також час, коли ці

правки були зроблені. Поле RequestId зв'язує цю таблицю з відповідним запитом на переклад.

Таблиця 3.6 – Таблиця БД TextCorrections

Поле	Тип даних	Опис
CorrectionId	INT	Ідентифікатор виправлення (PRIMARY KEY, AUTO_INCREMENT)
RequestId	INT	Ідентифікатор запиту на переклад (FOREIGN KEY, посилання на TranslationRequests)
CorrectedText	NVARCHAR(MAX)	Текст з виправленням
CorrectedAt	DATETIME	Дата виправлення (DEFAULT GETDATE())

Таким чином, архітектура бази даних створена для забезпечення всіх ключових функцій програми: зберігання інформації про користувачів, документи, запити на переклад та їх обробку. Ця архітектура дозволяє інтегрувати всі процеси в межах одного додатку, забезпечуючи ефективну обробку даних та доступність інформації для користувачів і перекладачів. Завдяки розподіленню даних по логічно структурованих таблицях, система є гнучкою, масштабованою та зручною для роботи з великими обсягами інформації.

3.8 Користувацький інтерфейс та його особливості

Користувацький інтерфейс програми застосунку розроблено з акцентом на простоту, інтуїтивність і функціональність. Основною метою було забезпечення зручного доступу користувачів до всіх можливостей системи, включаючи завантаження документів, подачу запитів на переклад, перегляд результатів та внесення коректив.

Для створення серверної частини програми було використано Flask, популярний мікрофреймворк для Python, який відзначається гнучкістю та простотою використання. Flask забезпечує ефективну обробку запитів від клієнтської частини програми через API. Його легковагова структура дозволяє швидко створювати веб-додатки, надаючи лише необхідний функціонал, який

можна розширювати за допомогою модулів. Flask ідеально підходить для побудови RESTful API, необхідного для комунікації між інтерфейсом користувача і сервером.

Для створення клієнтської частини обрано React.js, одну з провідних бібліотек JavaScript. React.js дозволяє будувати динамічний та інтерактивний інтерфейс із високою продуктивністю завдяки використанню віртуального DOM. Його компонентний підхід сприяє повторному використанню коду, спрощує підтримку та масштабування інтерфейсу. Крім того, React.js легко інтегрується з серверними API, що створює основу для швидкої та плавної роботи додатку.

3.8.1 Структура інтерфейсу користувача

Діаграма класів є важливим інструментом, який дозволяє відобразити структуру системи, а саме взаємозв'язки між її основними компонентами. Вона використовується для моделювання класів, їх атрибутів та методів, а також зв'язків між класами. Основна задача діаграми класів – це надати чітке уявлення про архітектуру системи та допомогти в проектуванні взаємодії об'єктів [18].

Діаграма класів користувацького інтерфейсу є частиною загальної моделі системи, що описує компоненти, які взаємодіють з користувачем. Вона дає змогу визначити структуру сторінок додатку та їхні взаємозв'язки з іншими частинами системи.

Крім того, діаграма класів включає також базові елементи інтерфейсу, такі як кнопки та текстові поля, які є важливими для взаємодії користувача з системою. Ці елементи можуть бути представлені як окремі класи з атрибутами та методами, що визначають їх поведінку.

Діаграма класів користувацького інтерфейсу дозволяє чітко визначити структуру інтерфейсу, взаємодію компонентів та їхні ролі в процесі виконання програми. Вона також допомагає розробникам зрозуміти, як елементи інтерфейсу взаємодіють між собою і з іншими частинами системи.

Діаграма класів користувацького інтерфейсу для застосунку, що розробляється, зображена на додатку И. Вона представляє структуру основних компонентів програми та їх взаємозв'язки. Основний клас системи – App, який містить два атрибути: header та router. Клас App відповідає за рендеринг інтерфейсу програми. Інтерфейс програми складається з кількох сторінок, кожна з яких представлена окремим класом, таким як AuthPage, HomePage, DocumentViewPage, RequestsPage, AdminPanelPage та TranslationEditPage.

AuthPage відповідає за авторизацію користувачів і включає атрибут state для зберігання стану користувача (ім'я користувача та пароль). Цей клас має методи для обробки логіну та реєстрації. HomePage є основною сторінкою для перегляду документів і включає методи для додавання, редагування та видалення документів. Він також має атрибут state для зберігання документів.

DocumentViewPage забезпечує перегляд окремих документів і надає можливість запиту на переклад документа. Цей клас має атрибут state, який містить інформацію про поточний документ. RequestsPage відповідає за відображення запитів на переклад, включаючи можливість перегляду, підтвердження або відхилення запиту. Клас має атрибут state, що зберігає список запитів на переклад.

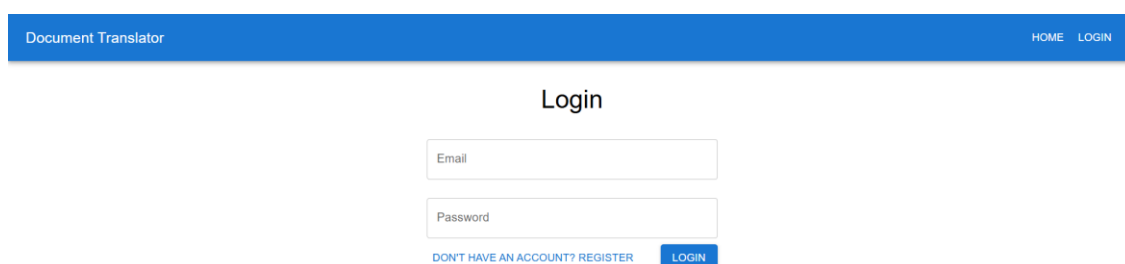
AdminPanelPage слугує інтерфейсом для адміністрування, зокрема для зміни ролей користувачів. Він містить атрибут state для зберігання інформації про користувачів. TranslationEditPage дозволяє редагувати текст перекладу, зберігаючи цей текст у своєму атрибуті state.

Існують також допоміжні класи, такі як Document, User, та TranslationRequest, що моделюють об'єкти документів, користувачів і запитів на переклад відповідно. Клас Document містить ідентифікатор і заголовок документа, User – ідентифікатор, ім'я та роль користувача, а TranslationRequest зберігає інформацію про документ, мову перекладу та статус запиту.

Взаємозв'язки між класами показують, як сторінки та об'єкти пов'язані один з одним, наприклад, клас HomePage має зв'язок з класом Document, а клас RequestsPage пов'язаний із класом TranslationRequest, що відсилає до класу Document для конкретизації перекладу.

3.8.2 Огляд сторінок користувацького інтерфейсу

Розглянемо основні сторінки додатку. Спершу користувача зустрічає сторінка авторизації та реєстрації (рисунок 3.7). Ця сторінка є початковою точкою взаємодії користувача із системою. Вона дозволяє користувачам авторизуватися через форму або зареєструватися, створюючи новий обліковий запис. Валідація даних виконується як на клієнтській стороні (за допомогою React), так і на серверній (за допомогою Flask), що забезпечує захист від некоректного введення.

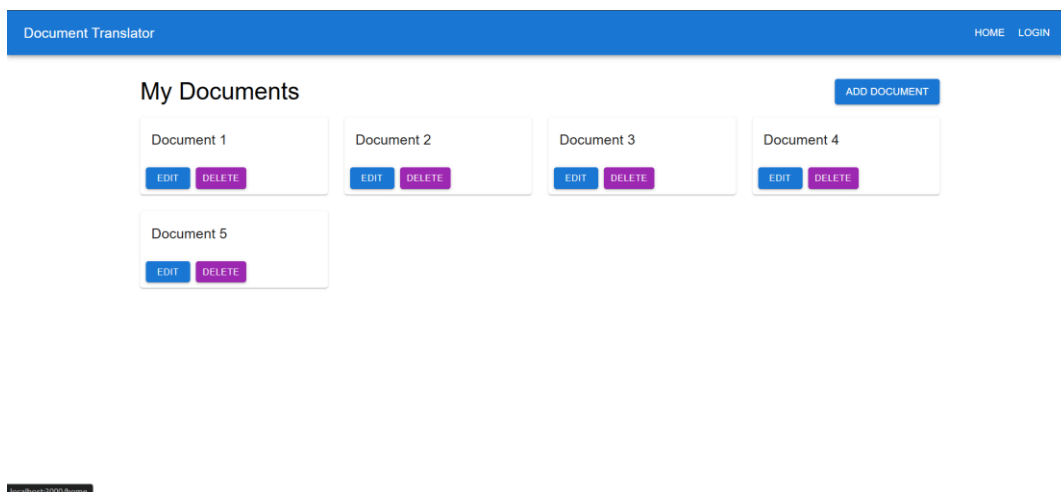


The screenshot shows a web application interface for 'Document Translator'. At the top, there is a blue navigation bar with the text 'Document Translator' on the left and 'HOME LOGIN' on the right. Below the navigation bar, the page is titled 'Login'. There are two input fields: 'Email' and 'Password'. Below the 'Password' field, there is a link that says 'DON'T HAVE AN ACCOUNT? REGISTER' and a blue button labeled 'LOGIN'.

localhost:3000

Рисунок 3.7 – Сторінка авторизації та реєстрації

Центральним елементом програми є головна сторінка (бібліотека документів), де користувач може переглядати список своїх завантажених документів (рисунок 3.8). Запити на сервер надсилаються через API, а відповіді обробляються React-компонентами, які динамічно оновлюють дані на сторінці.



localhost:3000/home

Рисунок 3.8 – Головна сторінка

Сторінка перегляду документа забезпечує візуалізацію документа, завантаженого користувачем, із можливістю перегляду його вмісту (рисунок 3.9). Flask генерує маршрути для завантаження файлів із сервера, тоді як React надає інструменти для масштабування та інтерактивної роботи з вмістом.



Рисунок 3.9 – Сторінка перегляду документа

Ця сторінка також дозволяє користувачу подати запит на переклад, вибравши мову та, за необхідності, перекладача. Flask обробляє запити, додаючи їх до бази даних, а React забезпечує зручну форму взаємодії.

Сторінка перегляду запитів (рисунок 3.10) створена для перекладача та відображає список активних запитів із їх статусом виконання. Перекладач може переглянути деталі запиту та редагувати текст.

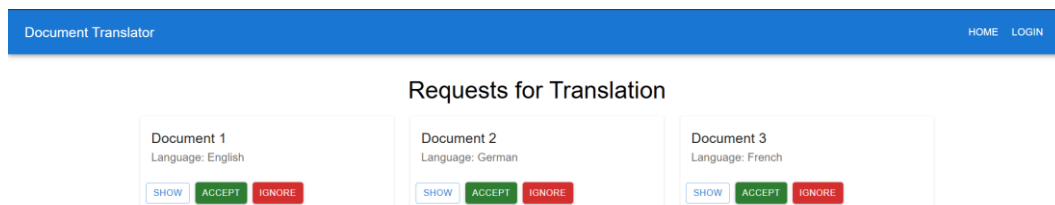


Рисунок 3.10 – Сторінка перегляду запитів на переклад

Сторінка редагування перекладу (рисунок 3.11). Ця сторінка призначена для внесення коригувань до перекладу, створеного системою. React надає текстові поля для редагування, тоді як Flask зберігає зміни у базі даних.

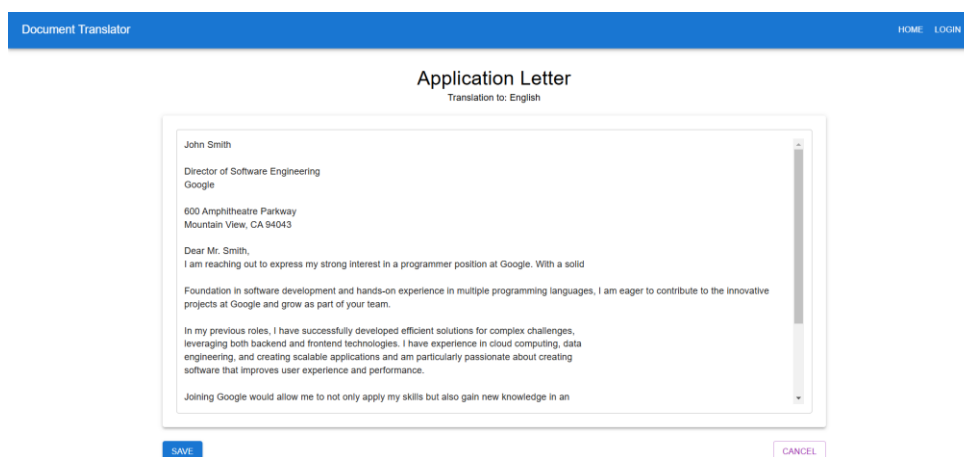
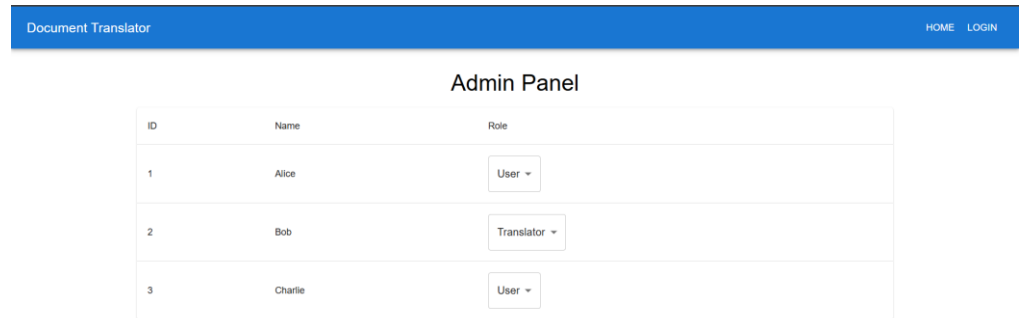


Рисунок 3.11 – Сторінка редагування перекладу

Панель адміністратора дозволяє адміністраторам контролювати систему (рисунок 3.12), керувати ролями користувачів. Всі дії виконуються через захищені маршрути Flask.



ID	Name	Role
1	Alice	User ▾
2	Bob	Translator ▾
3	Charlie	User ▾

Рисунок 3.12 – Панель адміністратора

У даному випадку Flask відповідає за серверну логіку, базу даних та обробку запитів, тоді як React.js реалізує функціональність інтерфейсу. API-комунікація забезпечується за допомогою стандартів REST, що дозволяє легко оновлювати як клієнтську, так і серверну частини.

Ця архітектура забезпечує високу продуктивність і дозволяє зосередитись на вдосконаленні як інтерфейсу, так і серверної частини без надмірного ускладнення розробки.

Висновки до розділу 3

У цьому розділі було детально описано архітектуру та ключові компоненти програмної реалізації системи, створеної для обробки та перекладу тексту із завантажених користувачем документів. Представлена архітектура забезпечує модульність, масштабованість і чіткий розподіл функціональних обов'язків між компонентами.

Розглянуто алгоритми обробки зображень, включаючи корекцію перспективи, виділення текстових областей і підготовку документів до розпізнавання тексту. Описано реалізацію модуля розпізнавання тексту та визначення іменованих сутностей, що використовує сучасні технології комп'ютерного зору та обробки природної мови.

Особливу увагу приділено модулю для перекладу тексту, який інтегрується з API DeepL. Завдяки цьому рішенняю забезпечується коректний до контексту переклад, який може бути додатково скоригований перекладачем.

Архітектура бази даних спроектована з урахуванням потреб у зберіганні даних про користувачів, документи, запити на переклад і внесені корективи. Забезпечено взаємозв'язок між усіма компонентами через чітко визначені таблиці, що гарантує цілісність і доступність даних.

Користувацький інтерфейс, створений за допомогою Flask та React.js, забезпечує інтуїтивно зрозумілий доступ до функціональних можливостей системи. Впроваджено сторінки для авторизації, завантаження документів, перегляду перекладів і внесення змін, що відповідає потребам кінцевих користувачів.

Розроблена система є високоефективним інструментом для роботи з документами.

4 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Змістовна постановка задачі

У контексті розробки даного програмного продукту основна задача полягає в забезпеченні якісного розпізнавання та перекладу текстової інформації. Це завдання включає кілька важливих підзадач, які охоплюють весь процес обробки тексту: від його вилучення з джерела до отримання результатів перекладу.

Першим етапом є розпізнавання іменованих сутностей, що є критично важливим для виділення ключової інформації з тексту, наприклад, імен, організацій, дат або географічних назв. Якість роботи NER-моделі визначається за показниками навченої моделі, які дозволяють оцінити точність і повноту виявлення сутностей.

Другим етапом виступає оцінка читабельності тексту, який є вхідними даними для перекладу. Використання метрик для оцінки якості тексту дає можливість визначити його складність та забезпечити контроль для вихідного матеріалу, що сприяє кращому розумінню його структури та змісту перед перекладом.

Третім етапом є переклад тексту, що здійснюється за допомогою сучасних систем перекладу. Однак якість перекладу потребує об'єктивної оцінки, яку можна отримати за допомогою метрик, що порівнюють результат, який надала система, та референсний текст. Важливо оцінити, наскільки точно перекладений текст відповідає оригіналу з точки зору лексики, граматики та стилю.

Крім того, необхідно забезпечити можливість оцінювання вихідного перекладеного тексту з точки зору читабельності для гарантії збереження якості та складності сприйняття. У випадку виявлення неточностей у перекладі або зниження читабельності тексту, система має дозволити подальше редагування з метою покращення результатів.

Таким чином, задача полягає у створенні комплексної системи, яка забезпечує розпізнавання іменованих сутностей, оцінку читабельності тексту, його переклад та оцінку якості перекладу.

4.2 Математична постановка задачі

Необхідно чітко визначити ключові вхідні дані, параметри оцінювання та бажані результати, що будуть отримані в ході вирішення задачі. Формалізація завдання дозволяє перетворити процес перекладу та аналізу тексту на математично визначену задачу, де можна застосувати відповідні алгоритми та методи оцінювання, забезпечивши можливість не лише отримати прогнозовані результати, але й підвищити якість роботи системи завдяки об'єктивному порівнянню альтернативних підходів.

4.2.1 Вхідні дані

У рамках розв'язання даної задачі маємо набір вхідних даних, які забезпечують формалізацію і підготовку інформації для обчислень. Деталізуємо кожен з них.

Текстовий документ D , який представлений у вигляді множини речень, що дозволяє розглядати текст на рівні елементарних одиниць для подальшої обробки, як-от розпізнавання сутностей або переклад:

$$D = \{S_1, S_2, \dots, S_n\}, \quad (4.1)$$

де S_i – окреме речення;

n – загальна кількість речень у документі.

Мова оригіналу L_{src} . Цей параметр є важливим для визначення лінгвістичних правил і специфіки обробки тексту.

Мова, на яку необхідно виконати переклад L_{tgt} . Вибір мови впливає на функціонування моделі перекладу та оцінку результатів.

Ці дані надходять до наступних моделей:

– NER-модель M_{NER} – інструмент, який використовується для розпізнавання іменованих сутностей у тексті, аналізує кожне речення S_i , визначаючи сутності (наприклад: імена, дати, локації);

– модель для перекладу тексту M_{Trans} – забезпечує переклад тексту з мови L_{src} на L_{tgt} та центральним елементом системи, який виконує основну операцію з трансформації тексту.

У процесі розраховуються наступні метричні моделі оцінки, а саме:

а) метрика читабельності M_{Read} ;

б) метрика оцінки перекладу: M_{BLEU} .

4.2.2 Вихідні дані

Вихідні дані є результатом обробки вхідних даних за допомогою моделей та метрик. Розглянемо, які конкретні результати очікуються на виході системи.

Множина іменованих сутностей E , виділених із тексту:

$$E = \{e_1, e_2, \dots, e_k\}, \quad (4.2)$$

де e_i – іменована сутність;

k – загальна кількість сутностей.

Оцінка складності тексту R_{src} для документа D :

$$R_{src} = M_{Read}(D), \quad (4.3)$$

де D – текстовий документ.

Перекладений текст T :

$$T = M_{Trans}(D, L_{src}, L_{tgt}), \quad (4.4)$$

де D – текстовий документ,

L_{src} – мова оригіналу;

L_{tgt} – мова перекладу.

Оцінка якості перекладу Q :

$$Q = M_{BLEU}(T, T_{ref}), \quad (4.5)$$

де T – перекладений текст;

T_{ref} – еталонний текст.

Оцінка читабельності перекладеного тексту R_{tgt} :

$$R_{tgt} = M_{Read}(T), \quad (4.6)$$

де T – перекладений текст.

Комбінована оцінка точності та повноти у вигляді F_1 -міри.

4.2.3 Процес розв'язання

Процес розв'язання задачі базується на кількох основних кроках.

Перший крок – це оцінка якості роботи NER-моделі. Для вимірювання ефективності розпізнавання іменованих сутностей використовуються такі метрики:

а) точність P_{rec} (Precision) – метрика вимірює частку правильно розпізнаних сутностей серед усіх можливих та обчислюється за формулою [19]:

$$P_{rec} = \frac{TP}{TP+FP}, \quad (4.7)$$

де TP – кількість правильно розпізнаних сутностей;

FP – кількість неправильно розпізнаних сутностей;

б) повнота R_{rec} (Recall) – метрика вимірює частку правильно розпізнаних сутностей серед усіх сутностей, які повинні були бути розпізнані, та розраховується за формулою [19]:

$$R_{rec} = \frac{TP}{TP+FN}, \quad (4.8)$$

де FN – кількість сутностей, які повинні були бути розпізнані, але не були виявлені.

в) F_1 -міра, яка комбінує точність і повноту в одну метрику, що дає більш збалансовану оцінку [20]:

$$F_1 = 2 * \frac{P_{rec} * R_{rec}}{P_{rec} + R_{rec}}, \quad (4.9)$$

Наступним кроком є оцінка читабельності тексту проводиться за допомогою індексу читабельності Флеша-Кінкейда, що дозволяє виміряти складність тексту. Формула для його обчислення [21]:

$$M_{Read}(D) = 0.39 * \left(\frac{W}{Sent}\right) + 11.8 * \left(\frac{Syll}{W}\right) - 15.59, \quad (4.10)$$

де D – текстовий документ;

W – загальна кількість слів у тексті;

$Sent$ – загальна кількість речень у тексті;

$Syll$ – загальна кількість складів у тексті.

Оцінка якості перекладу – це крок, для якого використовується одна з популярних метрик BLEU, що порівнює перекладений текст з референсним [22]:

$$M_{BLEU}(T, T_{ref}) = BP * exp(\sum_{n=1}^N \omega_n * \log P_n), \quad (4.11)$$

де T – перекладений текст;

T_{ref} – еталонний текст;

BP – штраф за довжину;

ω_n – вагові коефіцієнти;

P_n – точність n -грам.

Для остаточної оцінки якості роботи системи необхідно забезпечити виконання наступних умов:

а) максимізація точності NER-моделі, коли $F_1 \rightarrow max$;

б) забезпечення високої читабельності тексту:

$$R_{src}, R_{tgt} > R_{threshold}, \quad (4.12)$$

де $R_{threshold}$ – встановлений поріг, значення якого визначає, чи є текст достатньо зрозумілим і легким для сприйняття;

в) мінімізацію розбіжностей між перекладом і еталоном, так щоб $Q \rightarrow max$.

4.3 Обґрунтування методу розв'язання

4.3.1 Оцінка якості роботи NER-моделі

Розпізнавання іменованих сутностей є ключовим завданням в обробці природної мови. Останні дослідження показують, що сучасні моделі, засновані на глибоких нейронних мережах, зокрема трансформерах, досягли високих результатів у задачах NER. Для оцінки ефективності моделей NER зазвичай використовуються три основні метрики: точність (Precision), повнота (Recall) та F1-міра. Це дозволяє отримати об'єктивну та збалансовану оцінку якості роботи моделі.

Точність вимірює скільки з розпізнаних сутностей є правильними. Цей параметр важливий, коли важливо мінімізувати кількість хибних спрацьовувань моделі. У ситуаціях, де є висока ймовірність неправильного розпізнавання сутностей (наприклад, у випадку складних або схожих іменованих сутностей), точність дає змогу зрозуміти, скільки з розпізнаних сутностей справді є правильними. Це важливо для задач, де хибні спрацьовування можуть мати серйозні наслідки, наприклад, у медичних або юридичних текстах.

Повнота вимірює, скільки з усіх можливих сутностей було вірно виявлено. Повнота критична, коли важливо мінімізувати кількість пропущених сутностей. Повнота важлива для ситуацій, коли необхідно гарантувати, що модель виявляє якомога більше сутностей, навіть якщо це призводить до деякої кількості помилкових спрацьовувань. Наприклад, у випадку аналізу великих обсягів текстів, де пропущені сутності можуть привести до значних помилок, висока повнота є критичною для забезпечення максимальної інклюзивності розпізнавання. Моделі з високою повнотою допомагають переконатися, що важливі сутності не будуть пропущені.

F1-міра є середнім значенням точності та повноти, що дозволяє отримати більш збалансовану оцінку ефективності. F1-міра є найкращою метрикою, коли важливо досягти балансу між точністю та повнотою. Оскільки точність і повнота можуть бути взаємно виключними (наприклад, збільшення точності може знизити

повноту і навпаки), F1-міра дає можливість оцінити загальну ефективність моделі з урахуванням обох аспектів [20]. Це особливо корисно в тих випадках, коли жодна з окремих метрик не може дати повної картини ефективності моделі.

У задачах розпізнавання іменованих сутностей важливо враховувати як кількість помилкових спрацьовувань, так і можливість пропустити сутності. Метрики точності і повноти дають чітке уявлення про ці аспекти окремо, але в реальних умовах зазвичай виникає потреба в компромісі між ними. F1-міра дозволяє отримати збалансовану оцінку, що є критичним для ефективної оцінки NER-моделі, коли обидві метрики важливі для загальної успішності системи.

4.3.2 Оцінка читабельності тексту

Оцінка читабельності тексту є важливою задачею для визначення складності текстів, особливо в контексті машинного навчання та обробки природної мови. Читабельність визначає, наскільки легко або важко буде читати й розуміти текст певному користувачу або системі. Для цього використовуються різні метрики, які допомагають оцінити текстову складність. Однією з таких метрик є індекс читабельності Флеша-Кінкейда (Flesch-Kincaid Readability Index, FKRI), який обрано для цієї задачі.

Індекс читабельності Флеша-Кінкейда є однією з найпоширеніших метрик для оцінки складності англійських текстів. Він визначає, наскільки важко чи легко читати текст, на основі середньої кількості слів у реченні та складів у слові [21].

Індекс Флеша-Кінкейда підходить для оцінки текстів різних жанрів і тем, оскільки він зосереджений на об'єктивних лінгвістичних характеристиках, що відображають складність тексту. Вибір цієї метрики обґрунтовано її здатністю дати чітку кількісну оцінку складності тексту, що дозволяє здійснювати порівняння текстів на основі цих параметрів.

Оскільки метрика дає числове значення, його легко порівнювати для різних текстів або версій текстів, що є зручним для дослідження змін у читабельності при редагуванні чи перекладі.

4.3.3 Оцінка якості перекладу

Для визначення якості перекладеного тексту зазвичай використовуються спеціалізовані метрики, які дозволяють порівнювати перекладений текст із референсним (оригінальним або еталонним). Однією з таких метрик є BLEU (Bilingual Evaluation Understudy), яка є однією з найпоширеніших і надійних у сучасних системах машинного перекладу.

BLEU оцінює точність перекладу на основі співвідношення спільних n-грамів (послідовностей слів або символів) між перекладеним текстом і референсним.

BLEU заснована на порівнянні перекладених n-грамів з еталонними текстами і має кілька ключових характеристик, які роблять її ефективною для оцінки якості машинного перекладу [22]:

- основний принцип BLEU полягає в тому, що якщо певні n-грами (послідовності слів) у перекладі збігаються з n-грамами у референсному тексті, це вказує на високу якість перекладу;

- для уникнення ситуацій, коли коротший переклад отримує вищий бал, BLEU враховує довжину перекладу і накладає штраф на занадто короткі переклади, що дозволяє забезпечити коректність оцінки.

Значення BLEU знаходиться в межах від 0 до 1:

- оцінка 0 ставиться, якщо переклад не має жодних спільних n-грам із референсним текстом (повністю некоректний переклад);

- оцінка 1 означає, що переклад повністю збігається з референсним текстом (ідеальний переклад).

У реалістичних сценаріях значення BLEU зазвичай варіюються від 0 до 0.5, і чим вище значення, тим кращий переклад.

Хоча BLEU є популярною метрикою, існують й інші підходи для оцінки якості перекладу. До таких можна віднести METEOR (Metric for Evaluation of Translation with Explicit ORdering). Метрика, яка намагається врахувати синоніми,

морфологічні варіації та перестановки слів. METEOR часто дає більш точні результати в порівнянні з BLEU, оскільки враховує більше аспектів перекладу;

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) – метрика, яка в основному використовується для оцінки стиснутих (резюмуючих) текстів, але її також можна застосовувати до перекладів, оскільки вона враховує точність та повноту.

Проте BLEU був обраний через його простоту, об'єктивність і доведену ефективність у задачах машинного перекладу.

4.4 Експериментальне дослідження

4.4.1 Оцінка якості роботи NER-моделі

Для оцінки якості роботи моделі розпізнавання іменованих сутностей було реалізовано метод, який дозволяє зіставляти результати передбачень моделі з референсними даними з тестового набору.

Для кожного тексту з тестового набору проводилося прогнозування іменованих сутностей. Прогнозовані сутності зіставлялися з істинними, які були зазначені в розмітці даних. Це дозволило сформуванню списку правильних, хибно доданих та пропущених сутностей, які були використані для обчислення метрик.

Отримані результати для оцінки: Precision – 1.00, Recall – 0.72, та F1-Score – 0.84.

Оцінка точності 1.00 означає, що всі сутності, які модель передбачила, є правильними. Це свідчить про те, що модель добре відрізняє сутності від іншого тексту та не допускає помилок при їх класифікації. Модель не припустила жодної хибної позначки, що є ідеальним результатом.

Однак показник повноти – 0.72 вказує на те, що модель змогла виявити тільки 72% від усіх сутностей, які мали бути розпізнані. Це свідчить про те, що модель не здатна виявити деякі сутності, які є у тексті, що може свідчити про її неповну чутливість до різних варіантів сутностей або труднощі у розпізнаванні деяких

категорій. Вирішенням даної проблеми може стати підготовка більшої кількості тренувальних даних для навчання моделі.

Загальний результат F_1 -міри, що склав 0.84, є збалансованим і показує, що, хоча модель ідеально класифікує розпізнані сутності, їй все ще бракує здатності знаходити всі сутності в тексті.

Загалом, результати свідчать про хорошу роботу моделі з точним розпізнаванням сутностей, проте є потенціал для подальшого вдосконалення її здатності знаходити всі сутності в тексті.

4.4.2 Оцінка читабельності тексту

Оскільки індекс Флеша-Кінкейда розрахована на оцінку англійської мови, то для експерименту можна використати базову реалізацію через бібліотеку `textstat` та її функцію `flesch_kincaid_grade` для розрахунку рівня складності тексту.

Для української мови, зважаючи на те, що бібліотека `textstat` не підтримує українську мову, була використана інша реалізація. Для цього був застосований набір інструментів на базі бібліотеки `srsu`, що підтримує багатомовні моделі, зокрема українську. Для точної оцінки складності тексту українською була створена функція для підрахунку кількості складів у словах. Після цього був розрахований адаптований індекс Флеша-Кінкейда для української мови на основі середньої довжини речень і середньої кількості складів у слові.

Для експерименту було взято текст англійською та його переклад на українську. Це дозволило порівняти складність текстів до і після перекладу.

Результати експерименту для тексту на англійській показали, що його рівень складності згідно з індексом Флеша-Кінкейда становить 11.5. Це означає, що для розуміння цього тексту необхідно мати рівень освіти, що відповідає 11 класу. Такий рівень складності є характерним для багатьох академічних та професійних текстів.

Перекладений на українську текст, за допомогою адаптованої формули, показав результат 16.83. Це значення свідчить про те, що текст став набагато

складнішим після перекладу, що можна пояснити специфікою української мови, зокрема, її граматичною та лексичною структурою. Перекладений текст став важчим для сприйняття, що може вказувати на те, що такі переклади можуть бути менш доступними для широкої аудиторії.

Таким чином, результати експерименту показують суттєву різницю в складності текстів на англійській та українській мовах. Оригінальний англійський текст є зрозумілим для людей з рівнем освіти 11 класу, тоді як перекладений текст потребує вищого рівня освіти. Ці результати підкреслюють важливість врахування мовних особливостей під час перекладу, оскільки вони можуть значно вплинути на читабельність тексту.

Однак важливо зазначити, що цей результат також відображає початкову складність оригінального тексту. Хоча переклад на українську зробив текст складнішим, він все ж зберіг початкову складність англійського тексту. Таким чином, перекладений текст все ще відображає загальну складність і характер оригіналу, але в результаті мовних і структурних особливостей української мови стає складнішим для сприйняття.

Оскільки для експерименту був використаний діловий документ, це додатково підтверджує правильність обраного підходу.

4.4.3 Оцінка якості перекладу

Для забезпечення достовірності результатів ми оберемо декілька референсних перекладів з інших ресурсів, таких як: Google Translator, Gemini та ChatGPT, – що дозволить провести більш об'єктивне порівняння. Це також дозволить оцінити точність і відповідність перекладу до оригіналу, враховуючи різні підходи та варіанти перекладу на інші мови.

Для оцінки перекладу було використано бібліотеку `srasu` для токенизації текстів. Спершу завантажено багатомовну модель `xx_ent_wiki_sm`, яка дозволяє токенизувати. Після цього для оцінки якості перекладу була використана BLEU-

метрика з бібліотеки nltk, яка порівнює перекладений текст з референсними варіантами, надаючи числову оцінку, що відображає ступінь схожості між ними.

Результатом є оцінка в 0.5542, тобто якість перекладу в межах оцінки порівняння з референсними перекладами становить 55.42%. Цей показник вказує на середню схожість між перекладеним текстом та еталонними перекладами. Таке значення може свідчити про те, що переклад має достатньо високий рівень точності, але є можливості для покращень. Зазвичай значення BLEU в межах від 0.4 до 0.6 свідчить про достатньо добрий, але не ідеальний переклад, оскільки існують варіації у виразах та структурі, що не завжди можна точно відобразити в перекладі. При цьому варто розуміти, що в якості еталонних були взяті переклади з конкурентних продуктах, тому такий результат можна вважати цілком оптимістичним.

Висновок до розділу 4

У рамках дослідження було виконано змістовну постановку задачі, на основі якої побудовано математичну модель для розв'язання проблеми автоматизації розпізнавання та перекладу тексту документів. Було чітко визначено вхідні та вихідні дані, а також розроблено методологію, що включає кілька ключових етапів оцінки роботи системи, зокрема: розпізнавання іменованих сутностей, оцінку читабельності тексту та якості перекладу.

Під час експериментів система продемонструвала позитивні результати. Модель розпізнавання іменованих сутностей досягла високої точності, що свідчить про її здатність коректно виділяти ключові елементи тексту. Однак виявлено, що повнота розпізнавання може бути покращена, що відкриває перспективи для оптимізації алгоритмів, використання більших навчальних вибірок або розширення моделі через додаткові шари обробки даних.

Оцінка читабельності показала, що вихідні тексти є зрозумілими й відповідають очікуваному рівню сприйняття, що підтверджує їхню придатність для використання у реальних сценаріях. Щодо якості перекладу, оціненої за допомогою

метрики BLEU, результати виявили середній рівень точності. Це свідчить про базову відповідність перекладу вихідному тексту, однак також вказує на необхідність впровадження додаткових рішень для покращення результатів.

Таким чином, результати дослідження підтверджують високу ефективність розробленої системи та демонструють її потенціал у розв'язанні задач розпізнавання й перекладу документів. Поряд із цим, дослідження визначає напрямки для подальшого вдосконалення, серед яких покращення оцінки повноти розпізнавання сутностей, удосконалення механізмів перекладу, а також оптимізація архітектури моделі. Це дозволить забезпечити більш широкий спектр застосування системи в реальних умовах та буде сприяти її інтеграції у різні сфери діяльності.

5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

У сучасному світі, де обсяги даних стрімко зростають, здатність швидко і ефективно працювати з текстовою інформацією стає критично важливою для багатьох галузей. Особливої уваги потребують завдання цифровізації та обробки документів, які включають розпізнавання тексту, визначення його структури та переклад. Ці процеси є ключовими у міжнародному бізнесі, логістиці, правовій сфері та навіть у повсякденному житті.

Стартапи, що працюють у напрямку автоматизації роботи з текстовими даними, стають важливим елементом технологічного прогресу. Багато з них зосереджені на розробці рішень для розпізнавання тексту, обробки природної мови та машинного перекладу, прагнучи спростити роботу з документами. Відомі платформи, так як Google Cloud Vision чи ABBYY FineReader, демонструють значний прогрес у цій галузі, але більшість існуючих рішень мають обмеження в інтеграції функцій перекладу та структурованого аналізу документів.

Стартап-проект, розроблений у рамках даної роботи, прагне заповнити цю прогалину. Його метою є створення інноваційного програмного рішення, яке автоматизує процеси сканування документів, їх перекладу та обліку. Особливістю розробки стане інтеграція сучасних методів штучного інтелекту.

Проект спрямований на вирішення низки проблем, зокрема обмеження існуючих програмних продуктів у наданні комплексних інструментів для роботи з документами різними мовами. Реалізація передбачає створення зручного у використанні застосунку з широким спектром функцій, який стане важливим інструментом для компаній і користувачів, що працюють з великими обсягами текстових даних.

Ця розробка не лише відображає актуальні потреби ринку, але й відкриває нові можливості для автоматизації рутинних задач, сприяючи підвищенню продуктивності в різних сферах діяльності.

5.1 Опис ідеї проєкту

Основна ідея цього проєкту полягає у створенні інноваційного програмного продукту, який дозволяє автоматизувати всі етапи роботи з документами: від сканування до перекладу та обліку (таблиця 5.1). Така система базується на сучасних алгоритмах штучного інтелекту, які забезпечують високу точність розпізнавання тексту, ідентифікацію ключових елементів документа та якісний переклад.

Таблиця 5.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Програмний продукт для сканування, перекладу та обліку вже оцифрованих документів	Міжнародна торгівля	Змога автоматизувати переклад контрактів, митних декларацій та інших документів, зменшуючи витрати часу на ручну роботу та забезпечуючи високу точність перекладу.
	Логістика	Швидкий переклад транспортних накладних, сертифікатів і супровідних документів дозволяє оптимізувати процеси митного оформлення та координації перевезень.
	Наукові дослідження	Переклад академічних матеріалів, статей та тез дає дослідникам доступ

Зміст ідеї	Напрямки застосування	Вигоди для користувача
		цдо глобальних джерел знань.
	Адміністративна робота	Цифровізація архівів та автоматизований облік документів знижують навантаження на працівників, забезпечуючи легкий доступ до інформації.

Ключовими напрямками застосування програмного продукту є досить велика кількість різних сфер.

По-перше, це міжнародна торгівля. Програмний продукт дозволяє автоматизувати переклад контрактів, митних декларацій та інших документів, зменшуючи витрати часу на ручну роботу та забезпечуючи високу точність перекладу. Це сприяє прискоренню укладання угод та мінімізує ризик помилок через мовні бар'єри.

Також це досить важливо для логістика. Швидкий переклад транспортних накладних, сертифікатів і супровідних документів дозволяє оптимізувати процеси митного оформлення та координації перевезень. Це допомагає уникнути затримок і підвищити ефективність управління ланцюгами поставок.

Вагомий вклад можлививй і в наукові дослідження. Переклад академічних матеріалів, статей та тез дає дослідникам доступ до глобальних джерел знань. Це сприяє підвищенню якості досліджень, дозволяючи швидше знайти релевантну інформацію та обмінюватися результатами зі світовою науковою спільнотою.

Адміністративна робота також є можливо сйерою використання. Цифровізація архівів та автоматизований облік документів знижують навантаження на працівників, забезпечуючи легкий доступ до інформації та

зручний пошук. Це сприяє ефективному управлінню документообігом та збереженню важливих даних.

Для аналізу конкурентоспроможності розробленого програмного продукту було проведено порівняння його основних техніко-економічних характеристик із трьома основними конкурентами на ринку (таблиця 5.2): Google Translate, АBBY FineReader та Microsoft Translator. У порівнянні враховано ключові показники, які визначають ефективність, зручність та функціональність рішень у сфері обробки тексту та перекладу документів. Аналіз дозволив визначити сильні, слабкі та нейтральні сторони розробки, що є основою для подальшого вдосконалення продукту та розробки стратегії його просування на ринок.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Google Translate	АBBY FineReader	Microsoft Translator			
1.	Розпізнавання тексту	Так	Так	Так	Так		+	
2.	Виділення іменованих сутностей	Так	Ні	Ні	Ні			+
3.	Переклад тексту	Так	Так	Ні	Так			+
4.	Інтеграція всіх модулів в одному рішенні	Так	Ні	Ні	Ні			+
5.	Простота використання	Інтуїтивний	Інтуїтивний	Складний	Інтуїтивний			+
6.	Вартість	Низька	Безкоштовно	Висока	Безкоштовно		+	

№ п/ п	Техніко- економічні характерис- тики ідеї	(потенційні) товари/концепції конкурентів				W (слабк а сторо на)	N (нейтрал ьна сторона)	S (сильн а сторо на)
		Мій проект	Google Translate	ABBY Y FineRea der	Microsoft Translator			
7.	Гнучкість налаштува нь	Середня	Середня	Висока	Середня		+	

Ключова перевага продукту, що розробляється – це можливість виконувати всі необхідні операції у єдиному рішенні, що робить його унікальним на ринку. Успішне позиціонування продукту та його конкурентоспроможність залежатимуть від правильної маркетингової стратегії, спрямованої висвітлення унікальних переваг.

5.2 Технологічний аудит ідеї проекту

Технологічний аудит ідеї проекту – це процес комплексного аналізу та оцінки технічних аспектів розробки, який дозволяє визначити наявність необхідних ресурсів, можливих технологічних обмежень, готових рішень, а також оцінити потенціал продукту. Цей аудит включає аналіз існуючих технологій, їхню відповідність поставленим завданням і визначення найбільш ефективних шляхів для реалізації проекту.

Цей процес охоплює декілька ключових аспектів, а саме: аналіз наявних алгоритмів розпізнавання тексту та сучасних технологій перекладу з використанням ШІ, методів обробки даних і безпечного зберігання інформації, а також рівень інтеграції цих рішень у єдину програмну екосистему. У таблиці 5.3 представлено технологічну здійсненність реалізації ідеї проекту.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Розпізнавання тексту	OpenCV, PyTesseract	Наявна	Доступна

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
2.	Визначення іменованих сутностей	SpaCy	Наявна	Доступна
3.	Переклад тексту	DeepL API	Наявна	Доступна
4.	Інтеграція модулів та створення користувацького інтерфейсу	Flask для бекенду, бібліотеки JavaScript для фронтенду	Наявна	Доступна
5.	Система обліку та управління документами	PostgreSQL	Наявна	Доступна
6.	Хостинг та розгортання	Хмарні сервіси (AWS, Azure, Heroku)	Наявна	Доступна

Для реалізації проекту було обрано комплекс сучасних технологій, які забезпечують ефективне виконання кожного етапу роботи застосунку. Для розпізнавання тексту використовується поєднання PyTesseract і OpenCV, що дозволяє підвищити точність шляхом попередньої обробки зображень. Визначення іменованих сутностей реалізується через бібліотеку SpaCy для адаптації під специфічні задачі. Переклад інтегрується через API сервісів, таких як DeepL. Для інтеграції модулів і створення інтуїтивно зрозумілого інтерфейсу використовується Flask на серверній стороні та React або Vue.js для фронтенду. Система обліку документів побудована на базі PostgreSQL, яка забезпечує надійне зберігання даних. Завершальним етапом є хостинг і розгортання на хмарних платформах, таких як AWS чи Heroku, що гарантує стабільність і доступність сервісу.

Аналіз показав, що всі необхідні технології для реалізації проекту доступні авторам, мають наявну підтримку в спільноті та відповідають завданням проекту. Технічна реалізація можлива за допомогою існуючих рішень, які вже перевірені на практиці. Таким чином, реалізація проекту є технологічно здійсненою та може бути виконана у встановлені терміни.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Ринкове середовище є ключовим фактором, який визначає перспективність запуску стартапу. Для оцінки можливостей впровадження проекту було проведено аналіз попиту, динаміки розвитку ринку, обмежень для входу та інших показників, що характеризують стан потенційного ринку (таблиця 5.4). Важливо також визначити ключові потреби цільової аудиторії. Основними споживачами подібних рішень можуть бути міжнародні корпорації, державні установи, бюро перекладів, а також підприємства малого та середнього бізнесу, які займаються міжнародною діяльністю або працюють із багатомовною документацією.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	7 великих компаній
2	Загальний обсяг продаж, грн/ум.одв	12 мільярдів доларів
3	Динаміка ринку (якісна оцінка)	Зростання 20% щорічно
4	Наявність обмежень для входу (вказати характер обмежень)	Високий рівень конкуренції, потреба у якісному програмному забезпеченні
5	Специфічні вимоги до стандартизації та сертифікації	Дотримання ISO стандартів безпеки даних
6	Середня норма рентабельності в галузі (або по ринку), %	Від 20 до 30%

Аналіз показав, що ринок є перспективним для входження. Динаміка ринку демонструє зростання завдяки загальній цифровізації процесів та зростанню міжнародної торгівлі, де попит на автоматизовані рішення тільки збільшується. Основними загрозами є висока конкуренція з боку глобальних гравців, таких як Google, АВВУУ, і Microsoft, а також необхідність дотримання вимог до безпеки та стандартизації. Проте привабливий рівень рентабельності та швидке розширення ніші створюють вагомі ринкові можливості для запуску стартапу, який пропонує унікальну функціональність, що відсутня у конкурентів.

Для успішного запуску стартап-проекту важливо зрозуміти, хто є потенційними клієнтами і які їхні основні вимоги до продукту (таблиця 5.5). Визначення цільових аудиторій дозволяє сформулювати чітку стратегію для задоволення різних потреб ринку.

У таблиці наведено основні категорії споживачів, їхні специфічні потреби та вимоги до продукту, що допоможуть орієнтуватися в ринкових умовах і сприяти більш ефективному розробленню товару.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Переклад офіційних документів	Бізнес у сфері міжнародної торгівлі, логістики	Часті великі замовлення; необхідність високої точності перекладу та швидкої обробки документів	Висока точність перекладу
				Підтримка різних мов
				Гнучка система підписок
2.	Автоматизація обліку документів	Адміністративні організації, архіви, державні установи	Чутливість до стандартів безпеки; великий обсяг документів; потреба у збереженні історичних записів	Захист даних
				Можливість довготривалого збереження інформації
				Простота використання
3.	Доступ до наукової інформації	Наукові установи, освітні заклади, дослідницькі групи	Орієнтація на індивідуальні запити; підтримка нестандартних мов і термінів	Точність термінології
				Підтримка різних форматів документів
4.	Швидкий переклад документів	Малий бізнес, фрілансери, перекладачі	Невеликі разові замовлення; чутливість до ціни; необхідність у швидкому отриманні результатів	Доступність інтерфейсу
				Низька ціна

З аналізу потенційних клієнтів видно, що кожна група має свої унікальні вимоги. Бізнес у сфері міжнародної торгівлі та логістики потребує точності та швидкості перекладу, а державні організації – безпеки й можливості зберігання великих обсягів документів. Наукові установи зацікавлені в точності термінології та підтримці спеціалізованих мов, а малий бізнес та фрілансери більше орієнтуються на доступність і швидкість виконання завдань при мінімальних витратах.

Таким чином, стартап має враховувати ці відмінності і надавати відповідні рішення, які відповідатимуть вимогам цільової аудиторії.

Визначення факторів, що можуть вплинути на успіх чи невдачу проекту, дозволяє ефективно реагувати на загрози і використовувати можливості. У таблиці 5.6 подано основні загрози, які можуть виникнути під час запуску проекту, а також рекомендовані реакції на них.

Таблиця 5.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Сильна конкуренція	Висока конкуренція на ринку перекладу документів	Інвестування в унікальні функції, що відрізняють продукт від конкурентів
2.	Зміни в законодавстві	Можливі зміни в законах щодо обробки персональних даних	Постійне відстеження змін в регулюючих актах та адаптація продукту
3.	Низька поінформованість ринку	Потенційні клієнти можуть не знати про продукт	Запуск маркетингової кампанії для підвищення обізнаності серед цільових груп
4.	Технічні проблеми	Можливі проблеми з точністю перекладу або ідентифікацією даних	Тестування продукту на різних даних та удосконалення алгоритмів

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
5.	Низька готовність до цифрових технологій в деяких сферах	Складність впровадження нових технологій у стартапах	Проведення навчальних програм та консультацій для клієнтів

Таблиця 5.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Зростаюча потреба в автоматизації	Потреба в автоматизованих рішеннях для обробки документів зростає	Швидкий вихід на ринок і пропозиція унікальних автоматизованих рішень
2.	Розвиток штучного інтелекту	Постійне вдосконалення алгоритмів машинного навчання для розпізнавання тексту	Використання новітніх технологій для покращення точності перекладу та розпізнавання
3.	Розширення міжнародної торгівлі	Збільшення обсягів міжнародної торгівлі вимагає перекладу контрактів і митних документів	Розробка спеціалізованих рішень для перекладу юридичних та комерційних документів
4.	Підвищення попиту на багатомовність в бізнесі	Багатомовні платформи для обміну інформацією набирають популярність	Інтеграція підтримки кількох мов в програмне забезпечення та адаптація до специфічних вимог бізнесу

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
5.	Висока зацікавленість в нових технологіях серед малого та середнього бізнесу	Підвищений інтерес до цифрових рішень у малих і середніх підприємствах	Створення гнучких і доступних рішень для малого бізнесу з оптимізацією витрат

Загрози, такі як сильна конкуренція, зміни в законодавстві та технічні проблеми, можуть створювати бар'єри для ринкового впровадження проекту. Однак на ці загрози можна ефективно реагувати через інвестиції в інноваційні функції продукту, регулярну адаптацію до змін у правовій сфері та постійне тестування. Водночас можливості, такі як зростаючий попит на автоматизацію, розвиток штучного інтелекту та збільшення міжнародної торгівлі, створюють сприятливі умови для успішного запуску стартапу. Ось чому компанія має максимально використовувати ці можливості для розширення ринкових позицій, інтегруючи новітні технології та орієнтуючись на потреби малого та середнього бізнесу.

Визначення особливостей конкуренції (таблиця 5.8) допомагає сформулювати стратегію для того, щоб стартап зміг зайняти вигідну позицію на ринку, враховуючи тип конкуренції, рівень конкурентної боротьби, а також специфіку конкурентних переваг. Це дозволяє ефективно орієнтуватися в конкурентному середовищі та розробляти оптимальні стратегії для підвищення конкурентоспроможності.

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Олігополія	Ринок контролюється кількома великими гравцями.	Для підтримки конкурентоспроможності компанії потрібно пропонувати унікальні функції або вищу

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
		якість сервісу, щоб відрізнитися від основних конкурентів.
2. Національний	Конкуренція в межах країни, без виходу на міжнародний рівень.	Сфокусуватися на специфічних потребах локальних клієнтів і адаптувати продукт під локальні вимоги, знижуючи вплив глобальних конкурентів.
3. Внутрішньогалузева	Конкуренція відбувається між компаніями одного сегмента ринку (наприклад, всі працюють у сфері перекладу).	Підвищення якості та інноваційності продукту, залучення нових клієнтів через рекламу та спеціальні пропозиції.
4. Товарно-видова	Конкуренція між продуктами одного виду, але з різними характеристиками (наприклад, переклади для бізнесу, переклади для особистих потреб).	Розробка продуктів, що орієнтовані на специфічні потреби клієнтів, персоналізація сервісу.
5. Нецінова	Конкуренція відбувається через якість продукту, інновації, додаткові функції, обслуговування.	Зосередження на унікальних перевагах продукту, покращення обслуговування клієнтів та пропонування інноваційних рішень для привернення уваги.
6. Марочна	Конкуренція базується на брендах, їхніх репутаціях та впізнаваності на ринку.	Розвиток бренду, зміцнення його репутації через позитивні відгуки клієнтів, PR кампанії, підвищення впізнаваності на ринку.

Аналіз конкурентного середовища показує, що для успішного виходу на ринок стартапу необхідно зосередитися на кількох стратегічних напрямках. Оскільки ринок характеризується олігополією та внутрішньогалузевою конкуренцією, компанії важливо запропонувати унікальні функції та послуги, що відрізнятимуть її від конкурентів.

Крім того, особлива увага повинна бути приділена розвитку бренду та підвищенню якості сервісу, оскільки конкуренція носить неціновий характер і базується на наданні високоякісних послуг. Вибір локального рівня конкуренції дає можливість детально адаптувати продукт під потреби місцевих клієнтів і знизити вплив глобальних конкурентів.

Тепер виконаємо аналіз умов конкуренції на ринку з використанням моделі п'яти сил М. Портера. Цей підхід дозволяє оцінити різні фактори, що впливають на конкурентне середовище, включаючи вплив прямих конкурентів, постачальників, споживачів, заміників, а також потенційних конкурентів. Розуміння цих аспектів є важливим для того, щоб оцінити конкурентну позицію проекту та визначити стратегії для входу і подальшого успішного розвитку на ринку. Відобразимо аналіз в таблиці 5.9.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Google Translate, ABBYU FineReader, Microsoft Translator.	Нові технологічні і стартапи, закордонні розробники	Хмарні сервіси та бібліотеки AI/ML (наприклад, Google Cloud, AWS, OpenAI).	Корпоративні і замовники. Індивідуальні користувачі орієнтовані на безкоштовні або доступні послуги	Використання послуг професійних перекладачів або локальних програм перекладу.
Висновки:	Конкуренти вже мають розвинуті ринкові позиції та	Бар'єри не є непереборними для інноваційних стартапів.	Їх сила полягає у впливі на вартість доступу до обчислюва	Вимоги клієнтів, особливо в корпоративному сегменті, є високими.	Загроза невелика через високі витрати або обмеженість функціоналу

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	значну аудиторію.		льних ресурсів.		порівняно з автоматизованими рішеннями.

Аналіз показує, що ринок має помірну інтенсивність конкуренції. Існують прямі конкуренти з усталеними позиціями, такі як Google Translate, АBBYU FineReader та Microsoft Translator.

Однак ринок залишається відкритим для інноваційних продуктів, які пропонують унікальні функції та високу якість. Бар'єри для входу не є значними, але успішне впровадження залежить від доступу до сучасних технологій, фінансування та чіткої орієнтації на потреби споживачів.

Для того щоб проект був успішним, він повинен мати такі ключові сильні сторони:

- унікальні функції, які не пропонуються конкурентами (наприклад, інтеграція перекладу з автоматизованим розпізнаванням тексту на різних мовах);
- висока швидкість та точність роботи технології;
- орієнтація на специфічні ринкові сегменти, наприклад, логістику чи наукові дослідження;
- проста інтеграція з бізнес-процесами клієнтів;
- конкурентоспроможна ціна та якісний технічний супровід.

Ці висновки ляжуть в основу формування переліку факторів конкурентоспроможності, які розглянуто у таблиці 5.10.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1.	Інтеграція OCR та NER	Поєднання розпізнавання тексту з виділенням іменованих сутностей дозволяє автоматизувати роботу з документами, чого бракує більшості конкурентів.

№ п/п	Фактор конкурентоспроможності	Обґрунтування
2.	Підтримка великої кількості мов	Забезпечує охоплення різних ринків, задовольняючи потреби міжнародної торгівлі та наукових досліджень.
3.	Висока швидкість обробки	Забезпечує конкурентну перевагу для логістичних компаній, де час є критичним фактором.
4.	Легка інтеграція в бізнес-процеси	Простота використання та інтеграції з існуючими системами стимулює вибір продукту споживачами.
5.	Адаптація до сфер діяльності	Орієнтація на специфічні потреби, такі як автоматизація митних документів чи цифровізація архівів.
6.	Доступна ціна	Робить продукт привабливим для малого та середнього бізнесу, зменшуючи бар'єри для впровадження.
7.	Надійна технічна підтримка	Гарантія якісного супроводу та швидкого вирішення проблем підвищує лояльність клієнтів.

За визначеними факторами конкурентоспроможності виконаємо аналіз сильних та слабких сторін стартап-проекту (таблиця 5.11).

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів
1.	Інтеграція OCR та NER	18	+3
2.	Підтримка великої кількості мов	15	+2
3.	Висока швидкість обробки	17	+3
4.	Легка інтеграція в бізнес-процеси	14	+2
5.	Адаптація до сфер діяльності	16	+3
6.	Доступна ціна	12	+1
7.	Надійна технічна підтримка	19	+3

Аналіз показує, що основні сильні сторони проекту – це інтеграція OCR та NER, висока швидкість обробки, а також надійна технічна підтримка. Ці фактори забезпечують проекту значну перевагу на ринку, особливо в нішевих сегментах. У

той же час, фактор доступності ціни отримав лише +1, оскільки конкуренти можуть пропонувати дешевші продукти.

В цьому напрямку можливі вдосконалення через оптимізацію витрат чи розробку гнучких тарифів для різних категорій клієнтів. Загалом, проект демонструє високий потенціал конкурентоспроможності завдяки поєднанню інноваційних рішень і орієнтації на потреби клієнтів.

Для розуміння сильних і слабких сторін проекту, а також оцінки можливостей і загроз, було проведено SWOT-аналіз, результати якого наведено в таблиці 5.12.

Таблиця 5.12 – SWOT-аналіз стартап-проекту

Сильні сторони:	Слабкі сторони:
Інтеграції OCR та NER	Відносно висока ціна в порівнянні з деякими конкурентами.
Багатомовна підтримка	Складність проникнення на ринки з низькою технічною грамотністю клієнтів.
Можливості:	Загрози:
Зростання попиту на автоматизацію обробки документів у різних галузях (фінанси, медицина, юридичний сектор).	Поява нових конкурентів із подібними або вдосконаленими функціями.
Потенціал виходу на міжнародні ринки завдяки багатомовній підтримці.	Підвищення значущості цінової конкуренції через зниження доходів клієнтів.
Підвищення ефективності роботи клієнтів, що сприяє формуванню лояльності.	Ризики, пов'язані з нормативними змінами в області обробки персональних даних.

SWOT-аналіз виявив, що проект має сильний потенціал завдяки інноваційним рішенням і орієнтації на вузькі сегменти ринку. Однак, для забезпечення стабільної конкурентоспроможності, слід приділити увагу ціновій політиці та розширенню доступності на ринках із різними рівнями технічної зрілості. Залучення міжнародних клієнтів і налагодження партнерських зв'язків допоможуть перетворити можливості на реальні переваги, мінімізуючи вплив загроз.

У таблиці 5.13 представлені альтернативи ринкового впровадження стартап-проекту, кожна з яких передбачає певний комплекс заходів для ефективного виходу на ринок. Для кожної альтернативи визначені ймовірність отримання ресурсів та строки реалізації. Це дозволяє оцінити кожен стратегію з точки зору її доцільності та швидкості впровадження на ринку.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Вихід на ринок через партнерства з великими компаніями у сферах фінансів та медицини для інтеграції продукту в їх бізнес-процеси.	Висока, є партнери для співпраці.	12-18 місяців
2.	Запуск з орієнтацією на міжнародні ринки з адаптацією продукту до нових мов та стандартів.	Середня, залежить від можливості масштабування.	18-24 місяці
3.	Підвищення цінової привабливості для малих і середніх підприємств за допомогою знижок і лояльних програм.	Висока, потребує зниження цінової політики на ранніх етапах.	6-9 місяців
4.	Використання специфічних маркетингових стратегій для підвищення впізнаваності бренду через соціальні мережі та блогерів.	Висока, з огляду на ефективність цифрового маркетингу.	6-12 місяців
5.	Окремий фокус на розробку програм для мобільних платформ для більш широкого залучення кінцевих користувачів.	Середня, потребує додаткових інвестицій у технічну команду.	9-12 місяців
6.	Пошук венчурних інвестицій для збільшення фінансування й виходу на нові ринки.	Низька, залежить від умов інвесторів та ринкової ситуації.	12-18 місяців

Після оцінки представлених альтернатив, оптимальним варіантом є третя – підвищення цінової привабливості для малих і середніх підприємств за допомогою знижок і лояльних програм. Цей варіант характеризується високою ймовірністю отримання ресурсів і швидкими строками реалізації (6-9 місяців), що робить його найбільш привабливим для швидкого виведення на ринок, зважаючи на наявність відповідних ресурсів та можливість оперативно залучити цільову аудиторію.

5.4 Розроблення ринкової стратегії проекту

Ринкова стратегія будь-якого проекту починається з розуміння того, хто є його потенційними споживачами. Визначення цільових груп є критично важливим кроком у процесі планування, оскільки це дозволяє фокусувати маркетингові зусилля на конкретних сегментах ринку. Для проекту автоматизації перекладу та обліку документів за допомогою штучного інтелекту були визначені кілька основних груп споживачів, для яких продукт буде найбільш корисним.

Зокрема, проект орієнтований на підприємства, які потребують автоматизації процесів обробки документації, а також на організації, що мають великі обсяги текстових даних для обробки. Такими організаціями є, зокрема, освітні установи, державні органи, а також середній і великий бізнес.

Оскільки цільова аудиторія має різноманітні потреби та очікування, важливо провести сегментацію на кілька основних груп, що дозволить розробити індивідуалізовані маркетингові стратегії.

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Корпоративні клієнти (середній та великий бізнес)	Висока	Високий	Середня	Середня
2.	Освітні та наукові установи	Середня	Середній	Низька	Висока
3.	Державні установи та міжнародні організації	Середня	Низький	Висока	Низька
4.	Малі та середні підприємства	Висока	Високий	Середня	Висока
5.	Фрілансери та індивідуальні користувачі	Середня	Низький	Низька	Висока

Ці групи були визначені на основі аналізу потреб кожної з них у автоматизації обробки та перекладу документів, а також врахування їх готовності впроваджувати новітні технології, здатні оптимізувати внутрішні бізнес-процеси. Найбільш перспективними сегментами є корпоративні клієнти, середні підприємства, а також державні установи, які потребують інтеграції з іншими системами для забезпечення збереження даних та виконання високих стандартів безпеки.

Базова стратегія розвитку описує основний напрямок, в якому компанія або проект буде рухатися для досягнення своїх цілей. Вона включає в себе рішення щодо інновацій, продуктового портфолію, цінової політики та ринкової позиції. Для проекту автоматизації перекладу та обліку документів з використанням штучного інтелекту, основний акцент робиться на технологічних інноваціях, інтеграції з іншими системами та забезпеченні високої якості обробки документів.

Таблиця 5.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні і позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1.	Розробка модулів для інтеграції з корпоративними системами	Диференційований маркетинг	Технологічна інноваційність, інтеграція	Створення інноваційних рішень для автоматизації процесів перекладу
2.	Пропозиція продукту для малих і середніх підприємств	Масовий маркетинг	Доступність, швидкість, точність	Оптимізація витрат та спрощення процесів для МСП
3.	Партнерство з освітніми та державними установами	Фокусування на сегментах	Висока надійність, безпека даних	Рішення для державних організацій і освітніх установ

Зазначена стратегія розвитку дозволить задовольнити запити різних сегментів ринку, забезпечуючи стабільне зростання продукту в довгостроковій перспективі.

У сучасних умовах ринку, де інновації і технології швидко розвиваються, проекту необхідно визначити конкурентну стратегію, що дозволить йому зберегти перевагу над конкурентами. Вибір стратегії конкурентної поведінки безпосередньо впливає на те, як компанія буде взаємодіяти з іншими гравцями на ринку, зокрема, з наявними конкурентами, а також як вона планує інтегрувати нові рішення в поточний ринок.

Одним із основних підходів є стратегія диференціації, яка полягає в тому, щоб проект вигідно відрізнявся від конкурентів, пропонуючи унікальні функціональні можливості, високий рівень інтеграції з іншими системами та підвищену точність обробки даних.

Для визначення конкурентної стратегії важливо розглядати кілька аспектів, таких як можливість копіювання рішень конкурентів, ринкові позиції і технологічні бар'єри для входу на ринок.

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1.	Ні, але є одними з інноваторів	Шукати нових споживачів, зокрема в сегментах, що не охоплені конкурентами	Ні, зосереджуємося на унікальності власних рішень	Диференціація та інновації
2.	Так, з початковою пропозицією на ринку	Забирати частину клієнтів у конкурентів	Так, копіювати базові функції конкурентів для швидкої адаптації	Конкуренція через ціну та функціональність

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
		через цінову стратегію		

Проект не є першим на ринку, однак завдяки своїй унікальній технологічній пропозиції та можливості інтеграції з іншими системами, він може швидко знайти свою нішу. Оскільки конкуренція на ринку автоматизації документообігу досить висока, особливо з боку великих компаній, стратегія повинна орієнтуватися на цінову доступність для малого та середнього бізнесу, а також забезпечення високого рівня кастомізації для великих корпорацій та державних установ.

Позиціонування є важливою складовою маркетингової стратегії, яка визначає, як продукт буде сприйматися споживачами на ринку. Визначення стратегії позиціонування дозволяє чітко окреслити, що саме пропонує продукт, і чому він має бути обраний потенційним споживачем.

Основним орієнтиром для позиціонування є висока точність перекладу, швидкість обробки документації, безпека даних, а також доступність та інтеграція з іншими корпоративними системами. Проект має позиціювати себе як продукт для клієнтів, які потребують високої точності і швидкості у роботі з текстами та документами, особливо в умовах, коли йдеться про юридично значущу документацію або державні акти.

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Висока точність та швидкість виконання	Технологічні інновації	Надійність, висока точність, швидкість	Інновації, точність, надійність
2.	Інтуїтивно зрозумілий інтерфейс	Масовий маркетинг	Простота інтеграції та адаптації	Простота, доступність, швидкість
3.	Безпека даних	Партнерства з великими компаніями	Безпека даних, високий рівень захисту	Безпека, довіра, технологічність

Проект має позиціонувати себе як ефективне і надійне рішення для компаній, що мають велику кількість документів для обробки, а також для тих, хто працює з юридично важливими даними. Стратегії маркетингу будуть орієнтовані на забезпечення довіри до системи та демонстрацію її високої ефективності в різних сферах діяльності.

Розробка ринкової стратегії для даного проекту включає визначення цільових груп, розробку базової стратегії розвитку, а також формування конкурентної стратегії і стратегії позиціонування. Всі ці кроки сприятимуть успішному виведенню продукту на ринок і забезпечать його конкурентоспроможність у довгостроковій перспективі.

5.5 Розроблення маркетингової програми стартап-проекту

У таблиці 5.18 зібрано ключові переваги концепції товару, який буде запропоновано споживачам у межах стартап-проекту. Для кожної потреби споживача визначена вигода, яку надає товар, а також перераховано ключові переваги, що забезпечують його конкурентоспроможність. Ці переваги можуть

бути як існуючими на ринку, так і такими, що потрібно розробити для досягнення високої конкурентної позиції.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Зчитування та переклад тексту з документів	Зчитування тексту навіть з неякісних сканованих документів та переклад на різні мови	Використання найсучасніших технологій OCR (PyTesseract) та NLP (SpaCy), висока точність навіть при низькій якості оригіналів
2.	Обробка даних з документів у великій кількості	Швидка обробка та інтеграція даних в систему без втрат у якості	Розробка масштабованого рішення для обробки великих обсягів даних, підтримка мультiformатності та адаптивність під різні документи
3.	Зниження витрат на ручну обробку документів	Автоматизація процесу зчитування та перекладу, що зменшує людський фактор і час обробки	Імплементация передових технологій машинного навчання для покращення точності перекладу та адаптація до специфічних умов клієнта

Ці переваги допоможуть проекту зайняти конкурентну позицію на ринку, оскільки споживачі отримають ефективне та швидке рішення для обробки документів, що автоматизує багато трудомістких процесів.

Також важливо визначити три рівні товару для кращого позиціонування на ринку (таблиця 5.19).

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Потрібність: швидкий, точний переклад та автоматизоване зберігання документів.
	Вигода: зменшення часу обробки та зниження людських помилок.
II. Товар у реальному виконанні	Програмне забезпечення, що використовує штучний інтелект для обробки та перекладу тексту.

Рівні товару	Сутність та складові
	Інтерфейси для інтеграції в корпоративні системи.
III. Товар із підкріпленням	Підтримка та оновлення програмного забезпечення.
	Додаткові послуги: персоналізація, консультації.

Можемо скласти таблицю 5.20 для визначення цінових меж на основі аналізу ціни на аналогічні або замінні товари, а також доходів цільової групи споживачів. Цей аналіз допоможе встановити обґрунтовані межі ціни для майбутнього товару.

Таблиця 5.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	Товари з обробки тексту, які працюють з OCR та перекладом (наприклад, ABBYY FineReader, Google Cloud Translation)	Платні сервіси для автоматичного перекладу та зчитування (приклади: Google Cloud, Microsoft Azure OCR API)	Середній та високий рівень доходу (корпоративні клієнти, підприємства, державні установи, міжнародні організації)	Верхня межа: 1000–1500 грн/міс (для підписки або доступу до послуг на базі SaaS). Нижня межа: 300–500 грн/міс для малого та середнього бізнесу.
2.	Безкоштовні або дешеві альтернативи (наприклад, Open Source проекти з OCR, Google Translate)	Аналоги з базовим набором функцій для автоматичного перекладу або сканування текст	Середній та низький рівень доходу (малі підприємства, студії, індивідуальні користувачі)	Верхня межа: 500 грн/міс (основний функціонал). Нижня межа: 100–200 грн/міс для простих рішень.

Ціни на товари-замінники вказують на те, що ринок вже має низку доступних продуктів, як для висококласних рішень, так і для більш доступних варіантів,

орієнтуючись на доходи споживачів. Виходячи з цього, встановлення ціни на товар повинно бути в межах цих цінових діапазонів з огляду на конкурентоспроможність, при цьому враховуючи специфічні функціональні переваги вашого продукту.

Для визначення оптимальної системи збуту важливо врахувати специфіку закупівельної поведінки цільових клієнтів, функції збуту, які має виконувати постачальник товару, а також оптимальну глибину каналу збуту. Таблиця 5.21 допоможе організувати дану інформацію.

Таблиця 5.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Великий бізнес та організації потребують комплексних рішень, у тому числі технічної підтримки	Прямий контакт, технічна підтримка, консультації, постачання	Пряма система збуту, 1 рівень	Власна система збуту, власні торгові представники, онлайн-платформи
2.	Малий та середній бізнес потребує гнучких, простих рішень та помірних цін	Стандартне постачання без глибокої консультаційної підтримки	Канал збуту через посередників, 2 рівень	Залучення посередників, таких як дистриб'ютори або агенти
3.	Індивідуальні споживачі шукають простоту в покупці та доступні ціни	Можливість швидкої покупки та доставки	Канал збуту через онлайн-магазини, 1 рівень	Власна система збуту через електронну комерцію

Глибина каналу збуту визначається відповідно до потреб цільових груп та характеру взаємодії з постачальником, що допомагає оптимізувати процес продажу та забезпечити ефективне обслуговування клієнтів.

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 5.22).

Таблиця 5.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Клієнти зацікавлені в продуктах, що надають високі технологічні рішення з економією часу та зручністю використання	Онлайн-ресурси, соціальні мережі, email-розсилки, інтернет-реклама	Інноваційність, економія часу, простота використання	Показати, як продукт покращує ефективність та зручність роботи, підкреслити технологічні переваги	"Покращуйте свою продуктивність з нашим рішенням, яке економить ваш час та дає більше можливостей для розвитку вашого бізнесу."
2.	Споживачі цінують відгуки та рекомендації від користувачів	Відгуки на сайтах, соціальні мережі, онлайн-форуми	Надійність, відгуки користувачів, доступність	Створити довіру до продукту через реальні історії користувачів та їхній досвід	"Наш продукт отримав тисячі позитивних відгуків. Переконайтеся в його ефективності самі!"
3.	Малі бізнеси потребують простих та доступних рішень для початку роботи	Соціальні мережі, рекламні кампанії на вебсайтах та платформах для малого бізнесу	Простота використання, доступність, підтримка для малого бізнесу	Залучити клієнтів, підкреслюючи доступність та зручність продукту для малих підприємств	"Легкий старт з нашим продуктом для вашого малого бізнесу: доступно та просто."

Висновок до розділу 5

Проведений аналіз показав, що існує реальна можливість запуску стартап-проєкту. Попит на запропонований продукт стабільно зростає, оскільки він відповідає актуальним тенденціям на ринку, таким як автоматизація та підвищення ефективності бізнес-процесів. Динаміка ринку свідчить про високий потенціал розвитку, а рентабельність роботи на ринку залишається високою завдяки конкурентним перевагам, що забезпечують унікальність товару. Технологічні інновації й ефективність продукту є основними факторами, які сприяють його конкурентоспроможності.

Перспективи впровадження проєкту виглядають позитивними, оскільки цільова аудиторія має високий інтерес до технологічних рішень, які можуть покращити бізнес-процеси та знизити витрати часу. Хоча бар'єри входження на ринок включають значні початкові інвестиції в маркетинг і розробку продукту, ці перешкоди є подоланими за рахунок готової технології та наявності ресурсів для їх подолання. Стан конкуренції на ринку залишається активним, але конкурентоспроможність проєкту підтверджується завдяки його інноваційним функціям та здатності задовольнити потреби споживачів.

Щодо вибору альтернативи для ринкової реалізації, найбільш доцільною є стратегія з використанням власної системи збуту. Це дозволить зберігати контроль над процесом продажу та взаємодією з кінцевими споживачами, що сприятиме швидкому збору відгуків і вдосконаленню продукту. Власна система збуту також дозволить зменшити залежність від посередників і оптимізувати витрати.

Подальша імплементація проєкту є доцільною, оскільки ринкові умови та попит на інноваційні технології сприяють успішному впровадженню. Успіх реалізації залежатиме від розвитку маркетингових стратегій, вдосконалення функціональності продукту та налагодження партнерських відносин. Ринкове середовище дає всі передумови для успішного запуску проєкту в короткостроковій перспективі.

ВИСНОВКИ

У даній роботі було вирішено низку ключових завдань, спрямованих на розробку та впровадження інформаційної системи для автоматизації процесів обробки, розпізнавання та перекладу тексту документів. Дослідження охопило теоретичні та практичні аспекти, які стали основою для створення інноваційного рішення, що відповідає сучасним тенденціям ринку.

На початку, було проведено аналіз предметної області, який складався із вивчення сучасних технологій оптичного розпізнавання символів і методів обробки природної мови, а також програмних рішень, які дозволяють користуватися цими розробками прохідному користувачу. Це дозволило виявити ключові переваги та обмеження існуючих програмних продуктів. Зокрема, було з'ясовано, що поточні системи часто не враховують специфічні потреби, пов'язані з обліком документів і залученням експертів до перекладу. Таким чином, виникла необхідність розробки спеціалізованої системи, яка зможе забезпечити комплексний підхід до автоматизації процесів перекладу та подальшого обліку документів.

Наступний етап роботи був зосереджений на виборі технічних інструментів, які б дозволили ефективно вирішити поставлені перед проектом вимоги. Мова програмування Python була обрана завдяки її універсальності та великій кількості доступних бібліотек, які дозволяють реалізовувати складні алгоритми машинного навчання та обробки тексту. Для обробки зображень використовувалася бібліотека OpenCV, що забезпечує можливість попередньої підготовки документів, зокрема корекції перспективи та виділення текстових областей. Технологія PyTesseract була інтегрована для реалізації оптичного розпізнавання символів, що перетворює текст із зображень у цифровий формат. Модуль обробки природної мови, реалізований за допомогою бібліотеки SpaCy, дозволив визначати іменовані сутності в тексті, що є критично важливим для його аналізу та структуризації. Для реалізації перекладу було інтегровано API DeepL, що допомагає користувачуподібний по контексту документа текст на іншій мові, який може бути додатково скоригований перекладачем.

Розроблена архітектура програмного забезпечення відповідає принципам модульності, що забезпечує чіткий розподіл функціональних обов'язків між компонентами та можливість масштабування системи. У системі передбачені декілька основних модулів, кожен з яких відповідає за один з етапів обробки документу: сканування, розпізнавання тексту, переклад, а також облік завантажених файлів.

Ключову роль в реалізації архітектури системи відіграє серверна платформа Flask, яка забезпечила взаємодію між користувацьким інтерфейсом додатку та його серверною частиною. Для зберігання та обліку даних використовувалася база даних PostgreSQL, яка надає можливість легко працювати із великими обсягами даних.

Особлива увагу було надано створенню інтуїтивно зрозумілого інтерфейсу для кінцевого користувача. Використання технологій React.js дозволило розробити веб-додаток, який містить веб-сторінки для: авторизації в додатку, завантаження нових документів, їх перекладу. Також, було розроблено особливий інтерфейс для різних за роллю користувачів системи. Таким чином адміністратор може керувати ролями інших користувачів, призначаючи перекладачів. Ті в свою чергу можуть відповідати на запити звичайних юзерів з проханням допомоги у перекладі. Прохідний користувач має можливість надіслати запит на правки перекладу від експерта.

Поставлено математичну задачу та на її основі проведені експериментальні дослідження, які підтвердили ефективність розробленої системи. Алгоритми продемонстрували високий рівень точності при розпізнаванні іменованих сутностей, хоча було виявлено можливості для подальшого вдосконалення повноти моделі. Оцінка читабельності розпізнаних та перекладених текстів засвідчила їх відповідність вимогам, що ставляться до документів відповідного рівня складності. Оцінка якості перекладу, проведена за допомогою метрики BLEU, підтвердила здатність системи надавати точні результати із перспективами для вдосконалення шляхом інтеграції більш складних моделей машинного навчання.

Останній етап роботи стало дослідження ринкових перспектив впровадження проекту. Високий попит на автоматизацію бізнес-процесів подібного роду і конкурентні переваги розробленої системи надають можливості для реалізації даного комерційного продукту. Інноваційність технологій, якість функціональних рішень і позитивний відгук потенційної аудиторії, – підтвердили доцільність запуску даного стартапу-проекту.

Таким чином, результати магістерської роботи підтвердили практичну значущість і конкурентоспроможність запропонованої системи. Розроблений продукт є інноваційним рішенням, яке дозволяє перенести в цифровий світ роботу з паперовими документами, зробивши простіше повсякденні задачі цільових користувачів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. OCR (оптичне розпізнавання символів) – визначення, переваги, проблеми та випадки використання. URL: <https://uk.shaip.com/blog/ocr-definition-benefits-challenges-and-use-cases-infographic/>.
2. What is natural language processing (NLP)? URL: <https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP>.
3. Що таке нейронний машинний переклад? URL: <https://www.linguise.com/uk/блог/керівництво/що-таке-нейронний-машинний-переклад/>.
4. How does DeepL work? URL: <https://www.deepl.com/uk/blog/how-does-deepl-work>.
5. Unveiling the Intricacies of Google Translate: How Does It Work? URL: <https://medium.com/@rgdaksh03122005/unveiling-the-intricacies-of-google-translate-how-does-it-work-8437d5dfbd91>.
6. About ABBYY. URL: <https://www.abbyy.com/company/about-us/>.
7. Microsoft Translator: URL: <https://partner.microsoft.com/en-US/solutions/microsofttranslator#:~:text=Microsoft%20Translator%20is%20a%20cloud,web%20API%20supporting%2050%20languages>.
8. Bradski G. The OpenCV Library. Dr. Dobb's Journal of Software Tools. 2000.
9. Smith R. An Overview of the Tesseract OCR Engine. IEEE Computer Society, 2007. 629–633.
10. Honnibal M., Montani I., Van Landeghem S., Boyd A. spaCy: Industrial-Strength Natural Language Processing in Python. URL: <https://spacy.io/>.
11. Flask Documentation. Pallets Projects. URL: <https://flask.palletsprojects.com/en/2.0.x/>.
12. PostgreSQL Documentation. PostgreSQL Global Development Group. URL: <https://www.postgresql.org/docs/>.
13. DeepL Translator Documentation. URL: <https://www.deepl.com/pro-api>.
14. What is DFD (Data Flow Diagram)? URL: <https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/>.

15. Що таке BPMN-діаграма і навіщо вона потрібна з прикладами. URL: <https://iampm.club/ua/blog/shho-take-bpmn-diagrama-i-navishho-vona-potribna-z-prikladami-2/>.
16. What is Activity Diagram? URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>.
17. What is Sequence Diagram? URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>.
18. What is Class Diagram? URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>.
19. Precision-Recall. URL: https://scikit-learn.org/1.5/auto_examples/model_selection/plot_precision_recall.html.
20. F1 Score in Machine Learning: Intro & Calculation. URL: <https://www.v7labs.com/blog/f1-score-guide>.
21. Flesch–Kincaid readability tests. URL: https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests.
22. Demystifying the BLEU Metric. URL: <https://www.traceloop.com/blog/demystifying-the-bleu-metric#:~:text=A%20BLEU%20score%20is%20an,reference%20for%20one%20candidate%20sentence>.