

Вінницький національний технічний університет

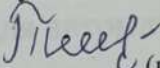
Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра комп'ютерних наук

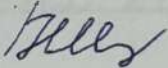
Бакалаврська кваліфікаційна робота на тему:

РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ДЛЯ АВТОМАТИЧНОГО
ПЕРЕКЛАДУ ТЕКСТІВ МІЖ МОВАМИ

Виконав: студент (4) курсу, групи 2КН –216
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

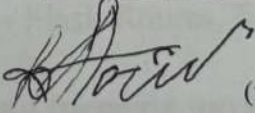
 Лучик Т. С.
(прізвище та ініціали)

Керівник к.т.н., доцент каф. КН

 Денисюк В.О.
(прізвище та ініціали)

«04» червня 2025 р.

Рецензент: к.т.н. професор каф. АІТ

 Папінов В.М.
(прізвище та ініціали)

«05» червня 2025 р.

Допущено до захисту

Зав. кафедри Гровий АА. 

«06» червня 2025 р.

Вінниця ВНТУ - 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
проф., д.т.н. Яровий А. А.

«05» травня 2025 р.

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Лучику Тарасу Сергійовичу

1. Тема роботи: Розробка інтелектуального модуля для автоматичного перекладу текстів між мовами.

Керівник роботи: Денисюк Валерій Олександрович, к.т.н., доцент
затверджені наказом вищого навчального закладу «20» березня 2025 року № 97

2. Строк подання студентом роботи 30 травня 2025 р.

3. Вихідні дані до роботи :

мова вихідного та цільового перекладу: англійська – українська, англійська – українська; обсяг словникової бази: не менше ніж 16 англійських ідіом і сленгових виразів; технології штучного інтелекту - MarianMT, HuggingFace Transformers, OpenNMT; середовище розробки: Python 3.11+, Flask; мова програмування - Python.; фронтенд - HTML, CSS у файлі веб-інтерфейсу Darkib.html; система обробки HTTP-запитів - Flask Routes, Forms; система розпізнавання сленгу та ідіом.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

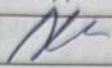
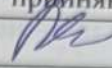
актуальність розробки; аналіз сучасних програмних засобів автоматичного перекладу - нейромережеві підходи (MarianMT, OpenNMT, DeepL); загальний алгоритм роботи модуля перекладу, його схема та UML-діаграма класів; діаграма прецедентів; вибір мови програмування; інтерфейс користувача; програмне забезпечення модуля перекладу; тестування роботи модуля та оцінка точності перекладу.

5. Перелік графічного матеріалу:

схема загального алгоритму роботи додатку; структурна схема

функціонування додатку; UML-діаграма класів додатку; схема життєвого циклу додатку; діаграма прецедентів.

6. Консультанти розділів роботи

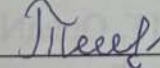
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Денисюк В. О., к.т.н., доц. каф. КН		

7. Дата видачі завдання 05 травня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз програмних аналогів автоматичного перекладу текстів	05.05.25	07.05.25	
2	Розробка алгоритму інтелектуального модуля перекладу	08.05.25	11.05.25	
3	Проектування структури інтелектуального модуля перекладу	12.05.25	15.05.25	
4	Проектування структури інтелектуального модуля перекладу.	16.05.25	26.05.25	
5	Тестування та приклади перекладу ідіом і сленгу	27.05.25	29.05.25	
6	Оформлення додатків: зображення інтерфейсу, UML-діаграми	30.05.25	01.06.25	
7	Розробка інструкції користувача	02.06.25	02.06.25	
8	Оформлення графічних матеріалів та підготовка до захисту бакалаврської кваліфікаційної роботи	05.06.25	04.06.25	

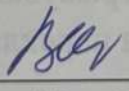
Студент


(підпис)

Лучик Т.С.

(прізвище та ініціали)

Керівник роботи


(підпис)

Денисюк В.О.

(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота обсягом 81 сторінок формату А4 містить 20 рисунків та список із 21 використаних джерел. Робота присвячена розробці інтелектуального модуля для автоматичного перекладу текстів між різними мовами з використанням сучасних технологій штучного інтелекту та обробки природної мови NLP.

У процесі дослідження було проаналізовано основні підходи до машинного перекладу, зокрема статистичні, шаблонні та нейронні методи. Розглянуто сучасні системи, такі як Google Translate, DeepL, OpenNMT, MarianMT, та обґрунтовано вибір нейромережевої архітектури для реалізації проєкту. Особливу увагу приділено точності перекладу складних мовних конструкцій — зокрема ідіом, фразеологізмів і розмовного сленгу — які становлять суттєву проблему для традиційних перекладачів.

Реалізований інтелектуальний модуль виконує автоматичний двосторонній переклад між англійською та українською мовами, забезпечуючи гнучку та масштабовану архітектуру. В основі рішення лежить модель MarianMT, адаптована з використанням фреймворку HuggingFace Transformers. Програмна реалізація здійснена мовою Python із залученням бібліотек Flask для створення веб-інтерфейсу.

Розроблений модуль дозволяє інтегрувати систему в інші програмні продукти, забезпечує високу якість перекладу та зручність використання. Запропоноване рішення може бути розширене для підтримки нових мовних пар і типів текстів. Експериментальні результати демонструють ефективність застосованого підходу та його практичну цінність у сфері автоматичного перекладу.

Ключові слова: машинний переклад, нейронні мережі, обробка природної мови, Python, MarianMT, DeepL, OpenNMT, вебкраулер, ідіоми, фразеологізми, NLP.

ABSTRACT

The bachelor's thesis is 81 pages long, contains 20 figures and a list of 21 references. The work is devoted to the development of an intelligent module for automatic translation of texts between different languages using modern artificial intelligence and natural language processing NLP technologies.

The study analysed the main approaches to machine translation, including statistical, template and neural methods. Modern systems, such as Google Translate, DeepL, OpenNMT, MarianMT, are considered, and the choice of a neural network architecture for the project is justified. Particular attention is paid to the accuracy of translation of complex linguistic constructions, such as idioms, phrases, and colloquial slang, which pose a significant challenge for traditional translators.

The implemented intelligent module performs automatic two-way translation between English and Ukrainian, providing a flexible and scalable architecture. The solution is based on the MarianMT model, adapted using the HuggingFace Transformers framework. The software implementation was carried out in Python with the use of Flask libraries to create a web interface.

The developed module allows the system to be integrated into other software products, ensuring high translation quality and ease of use. The proposed solution can be extended to support new language pairs and text types. Experimental results demonstrate the effectiveness of the approach and its practical value in the field of automatic translation.

Keywords: machine translation, neural networks, natural language processing, Python, MarianMT, DeepL, OpenNMT, web crawler, idioms, phraseology, NLP.

ЗМІСТ

ВСТУП	3
1 АНАЛІЗ ПРОГРАМ АНАЛОГІВ ДЛЯ АВТОМАТИЧНОГО ПЕРЕКЛАДУ ТЕКСТІВ МІЖ МОВАМИ.....	6
1.1 Обґрунтування актуальності розробки програми автоматичного перекладу текстів між мовами	6
1.2 Аналіз предметної області автоматичного перекладу текстів між мовам.....	7
1.3 Аналіз додатків-аналогів автоматичного перекладу текстів між мовам.....	9
1.4 Постановка задачі дослідження	11
1.5 Висновок до розділу 1.....	13
2 РОЗРОБКА ПРОГРАМИ ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ДЛЯ АВТОМАТИЧНОГО ПЕРЕКЛАДУ ТЕКСТІВ МІЖ МОВАМИ	15
2.1 Розробка загального алгоритму роботи програми.....	15
2.2 Розробка загальної структурної схеми програми перекладу текстів.....	19
2.3 Розробка діаграми прецедентів програми автоматичного перекладу текстів між мовами.....	25
2.4 Висновок до розділу 2.....	29
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ДЛЯ АВТОМАТИЧНОГО ПЕРЕКЛАДУ ТЕКСТІВ МІЖ МОВАМИ.....	31
3.1 Обґрунтування вибору мови та середовища програмування.....	31
3.2 Реалізація модуля автоматичного перекладу.....	36
3.3 Реалізація інтерфейсу програми автоматичного перекладу текстів між мовами.....	39
3.4 Тестування програми автоматичного перекладу текстів між мовами та аналіз результатів	43
3.5 Висновок до розділу 3.....	56
ВИСНОВКИ	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	61
ДОДАТОК А (ОБОВ'ЯЗКОВИЙ) ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ	64
ДОДАТОК Б (ОБОВ'ЯЗКОВИЙ) ЛІСТИНГ ПРОГРАМИ.....	65
ДОДАТОК В (ОБОВ'ЯЗКОВИЙ) ІЛЮСТРАТИВНА ЧАСТИНА.....	71
ДОДАТОК Г (ДОВІДНИКОВИЙ) ІНСТРУКЦІЯ КОРИСТУВАЧА.....	77

ВСТУП

Актуальність досліджень. У сучасному глобалізованому світі обмін інформацією між представниками різних мовних груп є критично важливим для науки, бізнесу, освіти та міжнародної співпраці. Автоматичний переклад став необхідним інструментом для забезпечення ефективної комунікації та доступу до знань у реальному часі. Проте, попри значний прогрес у сфері машинного перекладу, все ще існують проблеми, пов'язані з адекватною передачею значення складних мовних конструкцій, зокрема фразеологізмів, ідіом і розмовного сленгу.

Традиційні системи перекладу, особливо ті, що базуються на статистичних або шаблонних підходах, часто не справляються із завданням передати правильний зміст у контексті. Навіть сучасні нейромережеві перекладачі іноді допускають помилки при обробці нетипових або нестандартних виразів. Це особливо актуально для пар мов, які мають суттєві граматичні та культурні відмінності, як-от англійська та українська.

З огляду на це, розробка інтелектуального модуля, який поєднує можливості нейронного машинного перекладу та додаткову логіку попередньої обробки тексту, є актуальним і практично значущим завданням. Такий підхід дозволяє не лише підвищити якість автоматичного перекладу, а й забезпечити гнучкість та адаптивність системи до нових мовних сценаріїв.

Таким чином, запропоноване дослідження відповідає потребам сучасного інформаційного суспільства та спрямоване на вирішення однієї з ключових проблем у сфері обробки природної мови — забезпечення точного та контекстуального перекладу між мовами.

Метою дослідження є розширення функціональних можливостей автоматичного перекладу текстів між англійською та українською мовами з урахуванням контексту, ідіом, фразеологізмів і сленгових виразів. На відміну від існуючих рішень, у роботі запропоновано підхід, що поєднує можливості нейронного машинного перекладу.

з логікою попередньої обробки тексту та аналізом стійких мовних конструкцій. Це дозволяє досягти більш точного, адаптованого до людського сприйняття перекладу в неформальному й розмовному контексті.

Об'єктом дослідження є процес автоматичного перекладу текстів природною мовою, зокрема англійської мови на українську і навпаки, з фокусом на переклад неформальних мовних конструкцій.

Предметом дослідження є алгоритми, технології та програмне забезпечення, що забезпечують реалізацію інтелектуального модуля перекладу, зокрема методи виявлення ідіом, фразеологізмів і сленгу, нейромережеві моделі перекладу та засоби інтеграції словникових джерел у процес обробки тексту.

Задачі подальшого дослідження:

1. Провести аналіз існуючих систем перекладу: DeepL, Google Translate з фокусом на роботу з фразеологізмами та сленгом.
2. Визначити переваги і обмеження нейронних моделей машинного перекладу: MarianNMT, M2M100, OpenNMT у контексті перекладу складних виразів.
3. Побудувати структурну модель системи інтелектуальний модуль з урахуванням вимог модульності, адаптивності та можливості онлайн-інтеграції.
4. Створити алгоритмічну схему перекладу тексту з урахуванням логіки виявлення виразів, запитів до словників, машинного перекладу та форматування відповіді.
5. Провести функціональне тестування системи на прикладах англійськомовних речень з популярними ідіомами та розмовними виразами та реалізувати програмне забезпечення додатку та здійснити його тестування.
6. Сформулювати висновки та надати рекомендації щодо розширення програми інтелектуального модуля для автоматичного перекладу текстів між мовами.

«Апробація роботи.

Робота пройшла апробацію на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025), 2025р.», доповідь «Розробка інтелектуальної системи для автоматичного перекладу текстів між мовами» [1].

Публікація: електронні тези конференції «Молодь у науці: дослідження, проблеми, перспективи (МН-2025), 2025р.» [1].»

1 АНАЛІЗ ПРОГРАМ АНАЛОГІВ ДЛЯ АВТОМАТИЧНОГО ПЕРЕКЛАДУ ТЕКСТІВ МІЖ МОВАМИ

1.1 Обґрунтування актуальності розробки програми автоматичного перекладу текстів між мовами

У сучасному світі, де глобалізація та міжнародна взаємодія стають нормою в усіх сферах життя — від бізнесу й науки до освіти й особистої комунікації — питання ефективного міжмовного спілкування набуває все більшої актуальності. Щодня мільйони користувачів у різних країнах звертаються до систем автоматичного перекладу для читання, написання та розуміння текстів іншими мовами. Проте, навіть попри розвиток штучного інтелекту та нейронних моделей, існуючі перекладацькі системи не завжди забезпечують високу якість результату.

Особливо гостро проблема постає у випадках перекладу розмовної мови, сленгу, ідіом та фразеологізмів, які є контекстозалежними й часто не мають прямого відповідника в іншій мові. Через це переклад може втратити смислове навантаження або викликати нерозуміння. У більшості поширених систем машинного перекладу наприклад, Google Translate, такі випадки обробляються неточно, що знижує якість сприйняття інформації[2].

Існуючі рішення також не завжди враховують гнучкість роботи з текстами: можливість адаптації до контексту, тематики чи стилю мовлення. Крім того, немає ефективної інтеграції локальних мовних особливостей, таких як українізовані жаргони, фразеологізми чи мовні кальки. Це створює потребу у створенні більш інтелектуального інструменту, який враховує складні аспекти перекладу й здатен розуміти не лише структуру речення, а й його зміст.

Розробка спеціалізованого модуля автоматичного перекладу з використанням нейронних мереж, логіки попередньої обробки ідентифікованих ідіом, а також контекстної заміни нетипових виразів є актуальною задачею сучасного етапу розвитку технологій. Така система дозволить покращити точність, природність і

релевантність перекладеного тексту, особливо у сфері міжмовного спілкування між англійською та українською мовами. M2M100 — це багатомовна модель машинного перекладу, розроблена Facebook AI, яка підтримує прямий переклад між 100 мовами без потреби в англійській як проміжній[3].

Модель використовує трансформерну архітектуру та навчена на великій кількості паралельних корпусів, що забезпечує високу якість перекладу навіть для рідковживаних мовних пар. Вона підходить для задач глобального перекладу та демонструє конкурентні результати серед сучасних нейромережових систем.

Таким чином, створення інтелектуального модуля автоматичного перекладу сприятиме подоланню мовного бар'єру, покращенню доступу до інформації та знань, а також підвищенню якості комунікації в цифровому середовищі.

1.2 Аналіз предметної області автоматичного перекладу текстів між мовам

У наш час автоматичний переклад текстів є однією з ключових технологій для подолання мовного бар'єру в умовах глобалізації. Люди щодня стикаються з необхідністю читати, писати або інтерпретувати тексти іноземними мовами — в освіті, професійній діяльності, побутовому спілкуванні, інформаційному пошуку. Від якості перекладу залежить не лише зручність користувача, але й точність передачі змісту, що особливо важливо для технічної, юридичної чи наукової інформації.

Сфера автоматичного перекладу постійно розвивається — від перших систем зі словниковим підбором до сучасних нейромережових моделей. Проте однією з головних проблем залишається адекватна передача контексту, ідіом, фразеологізмів, розмовних виразів і культурно зумовлених значень. Більшість поширених перекладацьких сервісів, таких як Google Translate або DeepL, демонструють високу якість при перекладі стандартних конструкцій, але часто припускаються помилок у випадках нестандартних фраз або багатозначності. Ще однією важливою проблемою є роз'єднаність джерел даних. Існує велика

кількість словників, контекстних баз, баз ідіом, проте вони часто не інтегровані в єдину систему перекладу. Користувачеві доводиться вручну перевіряти значення виразів у різних джерелах, що знижує ефективність спілкування та витрачає час. Ситуацію ускладнює той факт, що українська мова є менш представлена в багатьох багатомовних моделях, що знижує якість перекладу саме для цієї мовної пари.

Створення інтелектуального модуля, який поєднує нейромережевий машинний переклад із вбудованим механізмом розпізнавання та підстановки ідіом, фразеологізмів і сленгу, є перспективним напрямом у розвитку систем обробки природної мови. Такий підхід передбачає попередню обробку тексту з виявленням нетипових або контекстно залежних виразів, їхню заміну на відповідники з вбудованих словників або контекстних баз, а вже потім — передачу адаптованого тексту до нейронної моделі перекладу. Розширення функціональності системи можливе завдяки інтеграції з API сторонніх сервісів, краулінгу паралельних корпусів та формуванню власних баз виразів і перекладів.

Одним із ключових аспектів є створення зручного користувацького інтерфейсу, забезпечення високої швидкодії та архітектурної гнучкості, що дозволить масштабувати систему на інші мовні пари або спеціалізовані предметні області.

Серцем сучасних систем перекладу є трансформерні нейромережі — архітектура. На відміну від рекурентних моделей, Transformer обробляє весь текст одночасно завдяки механізму самоуваги -self-attention, що дозволяє точно визначати взаємозв'язки між словами незалежно від їхньої позиції в реченні[4]. Це забезпечує високу якість перекладу, особливо у випадках з довгими або складними реченнями.

Ключову теоретичну базу для реалізації таких систем становить книга *Speech and Language Processing* Д. Джурафскі та Дж. Мартіна, де розглядаються принципи морфологічного та синтаксичного аналізу, розпізнавання мовлення, статистичні та нейромережеві методи обробки мови [5]. Це одне з основних академічних джерел для розуміння процесів, що лежать в основі NLP.

У свою чергу, глибинне навчання, яке є основою трансформерних моделей,

широко застосовується не лише в машинному перекладі, а й у сферах комп'ютерного зору, генерації тексту, біоінформатики та медичної діагностики [6].

Таким чином, створення системи, здатної коректно інтерпретувати складні мовні конструкції та перекладати їх у контексті, є не лише технічно доцільним, а й соціально значущим кроком у напрямку підвищення доступності знань і якості цифрових лінгвістичних інструментів. Завдяки здатності автоматично виявляти складні патерни в даних, моделі глибокого навчання використовуються в автономних автомобілях, чат-ботах, системах рекомендацій, а також у фінансовому секторі для виявлення шахрайства та прогнозування ризиків.

1.3 Аналіз додатків-аналогів автоматичного перекладу текстів між мовами

У сфері автоматичного перекладу текстів існує низка програмних продуктів, які забезпечують багатомовну комунікацію користувачів у різних контекстах — від побутового використання до професійної діяльності. Найпоширенішими прикладами таких систем є Google Translate, DeepL Translator, а також низка відкритих рішень, що базуються на нейронних мережах, як-от OpenNMT і MarianMT, HuggingFace [7].

Google Translate є одним з найвідоміших сервісів автоматичного перекладу, доступним як у вигляді веб-інтерфейсу, так і у вигляді мобільного додатку. Сервіс підтримує понад 100 мов і забезпечує переклад як окремих слів, так і цілих текстів, документів або вебсторінок. Завдяки застосуванню технологій машинного навчання та багатомовних трансформерів, система здатна забезпечувати достатній рівень перекладу для загального користування. Ключовими перевагами є швидкість, багатомовність, можливість офлайн-перекладу та зручність використання.

Проте Google Translate часто демонструє обмежену точність при перекладі ідіом, сленгових виразів, багатозначних слів або складних граматичних конструкцій, особливо між мовами з різною структурою, як-от англійська й українська.

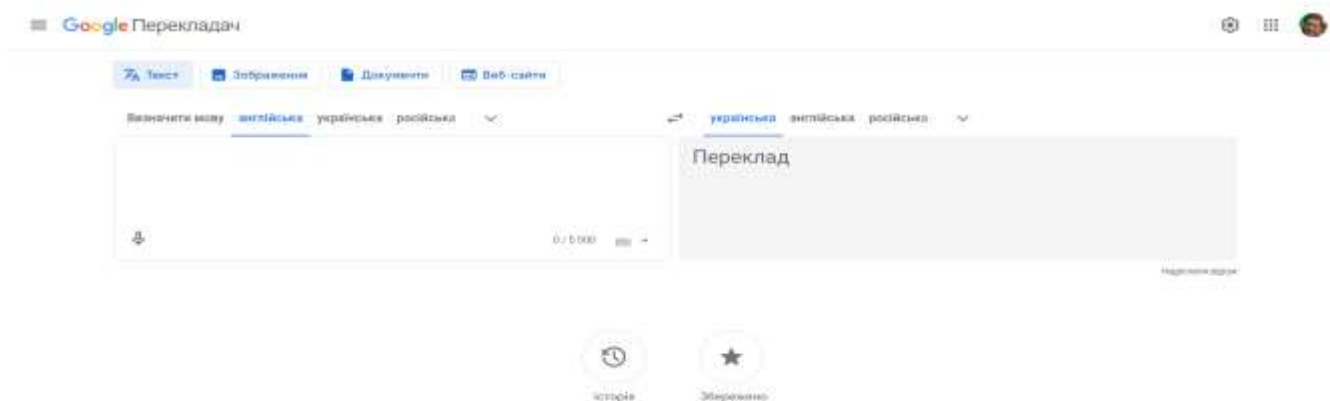


Рисунок 1.1 – Інтерфейс додатку «Google Translate»

DeerL — ще один сучасний перекладач, який використовує неймережі для підвищення якості перекладу. Сервіс відомий більш природним стилем перекладу та кращим збереженням контексту в порівнянні з іншими аналогами. DeerL активно застосовується в професійній сфері завдяки високій точності перекладу складних речень.

Недоліком є обмежена кількість мов у безкоштовній версії та відсутність підтримки багатьох менш поширених мов, серед яких — українська. Крім того, інструмент не надає розширеної підтримки обробки ідіом чи гнучкої інтеграції з іншими системами без використання платного API.

Окрім цього, DeerL не завжди коректно обробляє сленгові конструкції або культурно забарвлені вислови, що може призводити до втрати емоційного контексту. Це особливо критично при перекладі художніх або розмовних текстів, де важлива не лише точність, а й стилістична відповідність.

У зв'язку з цим, для задач, що вимагають перекладу з урахуванням фразеологізмів, ідіом або неформального мовлення, DeerL потребує доповнення спеціалізованими модулями або попередньою обробкою тексту. Це обмежує його

ефективність у вузькоспеціалізованих або інтерактивних системах перекладу, де потрібна адаптивність і контекстна гнучкість.

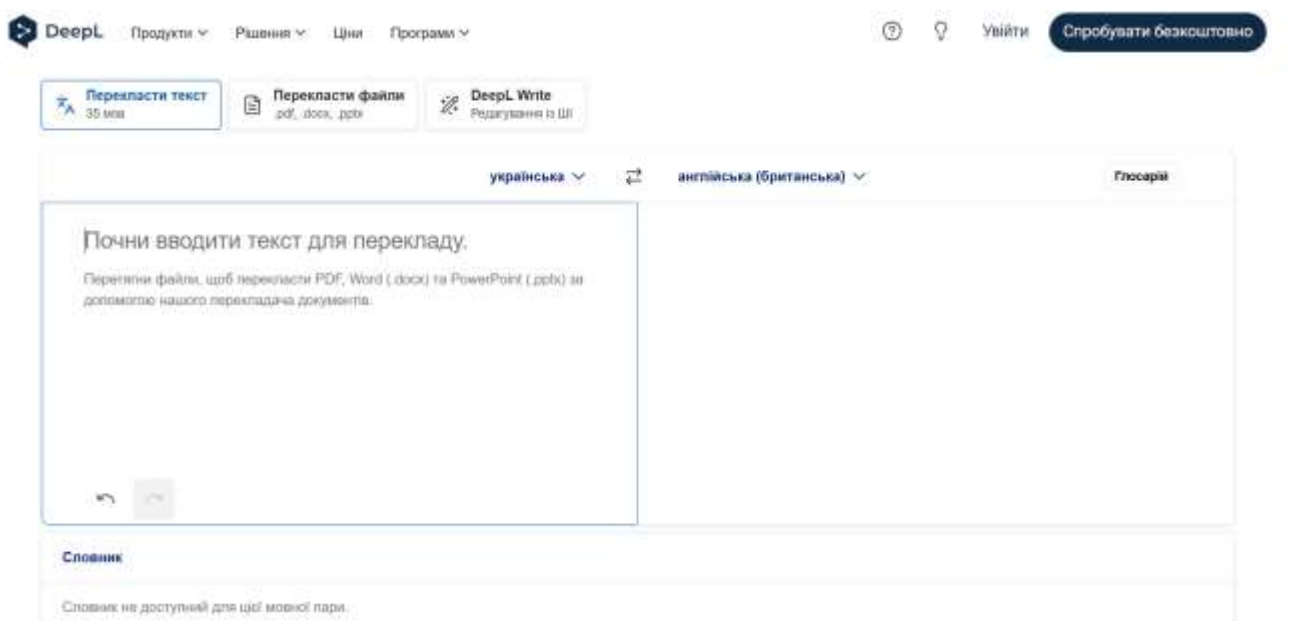


Рисунок 1.2 – Інтерфейс додатку DeepL

Наведені сервіси вирішують базові задачі автоматичного перекладу, але не завжди враховують складні мовні особливості — ідіоми, культурні контексти, локалізований сленг. Більшість із них мають обмеження щодо персоналізації перекладу або відкритості для розширення функцій. Це обґрунтовує доцільність створення власного інтелектуального модуля, який зможе не лише перекладати текст, а й виявляти нестандартні мовні конструкції та правильно передавати їхній зміст у цільовій мові.

1.4 Постановка задачі дослідження

Метою даного дослідження є розробка інтелектуального модуля автоматичного перекладу текстів між мовами з урахуванням лінгвістичних особливостей, контексту та складних мовних конструкцій, таких як ідіоми,

фразеологізми та сленг. Такий модуль має забезпечити більш точний та природний переклад у порівнянні з класичними системами, що базуються лише на нейромережевих моделях без попередньої обробки тексту. Основними функціональними складовими програмного модуля є:

- Вхід користувача та вибір напрямку перекладу. Система повинна надавати можливість введення тексту для перекладу та обрати напрямок.
- Попередня обробка тексту. Реалізація логіки виявлення ідіом та фразеологізмів у тексті з подальшою заміною їх на відповідники з вбудованого словника перед подачею в модель машинного перекладу.
- Застосування моделі нейронного перекладу. Використання сучасної моделі MarianMT, яка здатна здійснювати переклад у межах заданої мовної пари з високим рівнем точності.
- Обробка результату перекладу. Форматування вихідного тексту, перевірка якості результату, виявлення некоректних форм.
- Виведення перекладеного тексту користувачу. Забезпечення зручного інтерфейсу для виведення готового результату перекладу, з можливістю перегляду як вихідного, так і обробленого тексту.
- Модуль словникового розширення. Можливість додавання нових ідіом та їх відповідників у словник системи для покращення адаптивності перекладу.
- Архітектурна гнучкість. Побудова модуля у вигляді масштабованого рішення, яке може бути інтегроване в інші додатки або розширене для нових мовних пар.

Для реалізації всіх складових системи використовується мова програмування Python — одна з найпопулярніших платформ для розробки штучного інтелекту та обробки природної мови[8]. Завдяки широкому середовищу бібліотек, Python дозволяє ефективно реалізувати як лінгвістичний аналіз тексту, так і інтеграцію нейронних моделей у вебінтерфейс. Саме Python забезпечує швидку розробку, легку підтримку та можливість масштабування проєкту у майбутньому.

Розробка такого модуля дозволить користувачам отримувати якісніший переклад у режимі реального часу, особливо в тих випадках, коли мова йде про

контекстно-залежні або розмовні вирази. Інтелектуальна система забезпечить підвищення точності перекладу та зменшення помилок, властивих класичним перекладачам, які працюють без додаткової лінгвістичної обробки.

Проектування модуля включає кілька ключових етапів: визначення структури програми, розробку алгоритму обробки вхідного тексту, інтеграцію словникової бази та підключення моделі машинного перекладу. Важливою частиною роботи є також реалізація логіки заміни ідіом, обробки результатів перекладу та зручного представлення інформації для кінцевого користувача.

1.5 Висновок до розділу 1

У цьому розділі було обґрунтовано актуальність розробки інтелектуального модуля для автоматичного перекладу текстів між мовами. Зокрема, проаналізовано сучасні потреби користувачів у швидкому, точному та контекстно правильному перекладі, а також виявлено недоліки існуючих систем, які не завжди забезпечують адекватну обробку складних мовних конструкцій — таких як ідіоми, фразеологізми та розмовний сленг. Було проведено аналіз предметної області автоматичного перекладу.

Визначено, що незважаючи на значний прогрес у розвитку машинного перекладу, більшість наявних рішень мають обмеження в контекстній точності, особливо у мовних парах із високими структурними відмінностями, таких як англійська - українська. Актуальним є створення гнучкої, масштабованої системи, яка здатна комбінувати нейронний переклад із додатковою лінгвістичною обробкою. Окремо проаналізовано існуючі сервіси-аналоги, зокрема Google Translate та DeepL. Встановлено, що хоча ці системи демонструють хорошу загальну якість перекладу, вони не забезпечують належної точності у випадках контекстозалежних або розмовних виразів, і не надають розширеної підтримки словникової адаптації та попередньої обробки. Це підкреслює доцільність розробки власного рішення з вбудованою логікою роботи з ідіомами та сленгом.

Також було виконано постановку задачі дослідження, яка передбачає розробку модуля перекладу з такими основними функціональними складовими: введення тексту, вибір мовного напрямку, виявлення ідіом, заміна на відповідники з локального словника, подання тексту до нейронної моделі MarianMT, обробка результату перекладу та його форматowane відображення. Система передбачає можливість подальшого масштабування, інтеграції з API, розширення словникової бази та адаптації до інших мовних пар.

Таким чином, розробка інтелектуального модуля автоматичного перекладу текстів між мовами є перспективним напрямом, що дозволяє забезпечити більш точний, природний і змістовний переклад, підвищити якість міжмовної комунікації та розширити можливості сучасних перекладацьких систем.

2 РОЗРОБКА ПРОГРАМИ ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ДЛЯ АВТОМАТИЧНОГО ПЕРЕКЛАДУ ТЕКСТІВ МІЖ МОВАМИ

2.1 Розробка загального алгоритму роботи програми

Проектування інтелектуального модуля для автоматичного перекладу текстів між мовами є ключовим етапом у створенні сучасного програмного забезпечення, яке здатне ефективно вирішувати завдання міжмовної комунікації.

З огляду на зростання обсягів інформації, що передається багатьма мовами, актуальність таких систем лише зростає. Метою даної розробки є створення додатку, який забезпечуватиме автоматизований, швидкий і якісний переклад з урахуванням контексту та потреб користувача. Ефективність інтелектуального модуля перекладу напряду залежить від правильно сформульованого алгоритму його роботи.

На першому етапі визначаються цілі та завдання системи: переклад тексту в режимі реального часу, врахування особливостей мовного вжитку та надання рекомендацій на основі попередніх перекладів. Одним із центральних елементів є побудова зрозумілого та гнучкого інтерфейсу, що дозволить користувачеві швидко взаємодіяти з програмою: вводити текст, обирати мови перекладу, переглядати історію, а також додавати контекстні примітки. Для реалізації вебінтерфейсу використовується фреймворк Flask, який забезпечує зручну маршрутизацію, обробку запитів та виведення результатів у браузері[9]. Flask також дозволяє швидко інтегрувати сторонні API та моделі перекладу, зберігаючи просту структуру коду і гнучкість розгортання.

Програма повинна мати стабільну архітектуру, яка підтримує масштабованість і забезпечує захист даних користувача. Алгоритм роботи програми будується на поетапному обробленні введеної інформації. На вхід надходить текст, який проходить попередню обробку: видаляються зайві символи, здійснюється токенизація, визначаються ключові конструкції та ідіоми.

Далі активується модуль перекладу, що ґрунтується на нейронних мережах,

який здійснює автоматичний переклад з урахуванням контексту. Важливим аспектом є модуль післяобробки, який дозволяє адаптувати переклад до цільової мови, враховуючи стиль, регіональні особливості та інші лінгвістичні чинники. Додаткові функції, такі як персоналізовані підказки на основі попередніх перекладів або тематики, підвищують точність і зручність застосування додатку. Наприклад, якщо користувач часто перекладає тексти про подорожі, програма може автоматично адаптувати переклади під туристичну лексику, надаючи релевантні приклади або рекомендації.

Особливу увагу також приділено можливості вбудованого режиму знайомств: функціональність програми передбачає підбір співрозмовників за інтересами, місцем перебування та мовними уподобаннями. Це відкриває додаткові можливості для мовного обміну, живого спілкування та практики.

Основні переваги запропонованого рішення:

- Автоматизований переклад з урахуванням контексту;
- Інтуїтивний інтерфейс для швидкого доступу до функцій;
- Персоналізація рекомендацій і перекладів;
- Сучасні технології обробки мови, зокрема використання моделей машинного навчання;
- Можливість розширення на інші мовні пари, що дозволяє масштабувати систему під потреби міжнародної аудиторії.
- Низьке споживання ресурсів у порівнянні з великими LLM, завдяки оптимізованій архітектурі нейромереж.
- Швидкий час відповіді завдяки попередній токенізації та оптимізації потоків обробки запитів.
- Модульність архітектури, що спрощує оновлення окремих компонентів без зупинки всієї системи.
- Виявлення мови автоматично, що спрощує процес перекладу для недосвідчених користувачів.

- Можливість інтеграції з мобільними та веб-застосунками, завдяки REST API або готовим бібліотекам.
- Система самонавчання на основі зворотного зв'язку користувачів, яка дозволяє покращувати якість перекладу на основі реальних кейсів.

Розроблена система перекладу має низку важливих переваг, що роблять її конкурентоспроможною на сучасному ринку інструментів обробки мови. По-перше, автоматизований переклад виконується з урахуванням контексту, що дозволяє досягати більш точних і природних результатів, особливо у випадках із фразеологізмами та розмовною лексикою. Інтуїтивний інтерфейс забезпечує зручність користування навіть для недосвідчених користувачів, а персоналізація перекладу дозволяє враховувати індивідуальні стилістичні особливості.

Архітектура системи підтримує масштабування на інші мовні пари, забезпечує швидку обробку запитів, автоматичне визначення мови та низьке споживання ресурсів, що є ключовим для мобільних і веб-додатків. Завдяки модульності система легко оновлюється, а підтримка зворотного зв'язку дозволяє їй самонавчатися, покращуючи якість перекладу з часом. важливою перевагою є підтримка інтеграції з різними програмними продуктами — мобільними застосунками, веб-сервісами або десктопними платформами — завдяки REST API та готовим бібліотекам. Це дозволяє легко впроваджувати систему перекладу в існуючі рішення без необхідності значної доробки інфраструктури.

Завдяки використанню сучасних моделей машинного навчання, зокрема нейронних мереж зі спеціалізованими підходами до обробки природної мови, система демонструє високу адаптивність і здатність до удосконалення. Такий підхід сприяє створенню гнучкої екосистеми, яка може ефективно працювати як в освітньому середовищі, так і в бізнес- або інформаційних сервісах.

Таким чином, побудова загального алгоритму є базовим кроком до реалізації ефективного інтелектуального модуля. Нижче наведено загальну структурну схему роботи програми на рисунку 2.1, а також покроковий опис її функціонування:

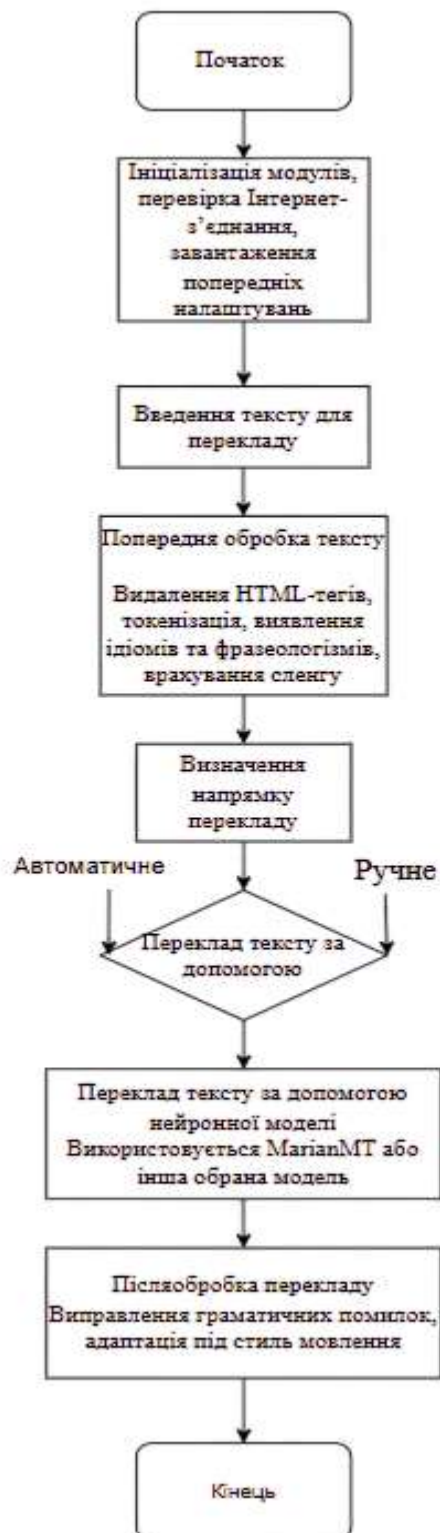


Рисунок 2.1 – Схема алгоритму роботи інтелектуального модуля для автоматичного перекладу текстів між мовами

Запуск програми:

- Ініціалізація модулів, перевірка інтернет-з'єднання, завантаження попередніх налаштувань.

Введення тексту для перекладу:

- Користувач уводить або вставляє текст.

Попередня обробка тексту:

- Видалення HTML-тегів, токенизація, виявлення ідіом та фразеологізмів.

Визначення напрямку перекладу:

- Автоматичне або ручне визначення мови введення та мови перекладу.

Переклад тексту за допомогою нейронної моделі:

- Використовується MarianMT або інша обрана модель.

Післяобробка перекладу:

- виправлення граматичних помилок, адаптація під стиль мовлення.

Відображення результату та можливість збереження:

- Переклад подається у зручному вигляді. Можна скоригувати або переписати.

2.2 Розробка загальної структурної схеми програми перекладу текстів

Загальна структурна схема програмного забезпечення для автоматичного перекладу текстів між мовами відображає призначення та взаємозв'язок основних функціональних компонентів системи.

Дана схема демонструє взаємодію між модулями програми та користувачем, а також послідовність обробки даних — від введення тексту до виводу перекладеного результату. У складі програмного модуля передбачено реалізацію таких основних компонентів:

- Модуль введення тексту, що відповідає за отримання вхідного повідомлення від користувача;
- Модуль попередньої обробки, який виконує очищення тексту від HTML-тегів, токенизацію, а також виявлення ідіом і фразеологізмів.

- Словниковий модуль, який забезпечує підстановку відповідників для виявлених ідіом.
- Модуль визначення напрямку перекладу, який працює автоматично або дозволяє ручний вибір мовної пари.
- Модуль машинного перекладу, що базується на нейромережевій моделі.
- Модуль післяобробки перекладу, який коригує граматичні помилки та адаптує стиль тексту.
- Модуль виводу, який відображає результат перекладу у користувацькому інтерфейсі.

Точкою входу в систему є форма введення тексту, після чого відбувається його послідовна обробка згідно з логікою програми. Загальна структурна схема функціонування інтелектуального модуля перекладу зображена на рисунку 2.2.

1. Модуль ініціалізації.

Цей компонент виконує перевірку доступу до Інтернету, завантаження попередніх налаштувань користувача та конфігурацій моделі, ініціалізацію словників, а також підключення до API, якщо модель працює у хмарному середовищі. Надійна робота цього модуля критично важлива для подальших етапів обробки тексту.

2. Модуль введення тексту.

Призначений для отримання вхідного тексту, який користувач бажає перекласти. Це може бути окреме речення, абзац або навіть цілий документ. Даний модуль відповідає також за перевірку валідності введення.

3. Модуль попередньої обробки.

Виконує очистку тексту від небажаних елементів, таких як HTML-теги, зайві символи, та виконує токенизацію — розбиття тексту на окремі логічні одиниці. Важливою особливістю цього модуля є можливість виявлення ідіом та фразеологізмів у тексті за допомогою вбудованих словників, що суттєво підвищує якість перекладу. Система також враховує сленгові вирази, які можуть спотворити зміст при буквальному перекладі.

4. Модуль визначення напрямку перекладу.

Цей блок може працювати як у ручному, так і в автоматичному режимі. Автоматичне визначення базується на детекції мови вхідного тексту. У ручному режимі користувач самостійно обирає мовну пару.

5. Модуль машинного перекладу.

Основний функціональний блок, який передає текст до нейронної перекладацької моделі. У рамках даної роботи використовується модель MarianMT — одна з найефективніших open-source моделей машинного перекладу. Проте архітектура модуля дозволяє інтегрувати інші моделі, забезпечуючи масштабованість системи[10].

6. Модуль післяобробки.

Цей модуль відповідає за покращення якості результату. Він може виконувати граматичну перевірку, адаптацію стилю, а також коригувати некоректні або незграбні конструкції, які іноді можуть виникати після перекладу складних речень. Це особливо важливо при перекладі текстів, які містять ідіоми, розмовні фрази або професійну термінологію.

Окрім цього, модуль післяобробки може виконувати локалізацію результату перекладу відповідно до культурних та мовних норм цільової аудиторії. Наприклад, він здатен адаптувати дати, валюти, одиниці вимірювання або навіть змінювати структуру речень для досягнення природності у перекладі.

Завдяки цьому забезпечується не лише технічно правильний переклад, а й змістовна й стилістична релевантність. Крім цього, модуль післяобробки дозволяє враховувати контекст вживання слів та фраз у межах усього тексту, а не лише. Це особливо важливо для уникнення повторів, плутанини з однозначними словами чи неправильного узгодження граматичних форм.



Рисунок 2.2 – Загальна структурна схема функціонування інтелектуального модуля перекладу між мовами

7. Модуль виводу результату.

Останній етап — передача перекладеного тексту назад користувачу. Цей модуль може підтримувати декілька форматів виводу: простий текст, HTML, форматовані блоки тощо. У разі інтеграції в більшу систему, модуль може надсилати результат на API або інші зовнішні сервіси, де переклад буде використано в подальших автоматизованих процесах. Також важливо, щоб модуль виводу забезпечував зручне та швидке відображення результатів, адаптоване під тип платформи. Він може включати опції копіювання, збереження або експорту результату у файл. У деяких випадках реалізується також функція порівняння вихідного і перекладеного текстів або зворотного перекладу для перевірки точності.

Завдяки реалізації модуля виводу результату користувач отримує миттєвий доступ до перекладеного тексту без затримок, що значно підвищує загальну швидкість взаємодії з системою. Підтримка декількох форматів виводу дозволяє легко інтегрувати модуль у різні сценарії використання — від звичайного перегляду в браузері до використання в автоматизованих інформаційних системах.

Ще однією важливою перевагою є можливість його експорту у зручному форматі, що особливо актуально для професійних перекладачів або науковців. Додаткова функція зміни мови дозволяє користувачу порівняти переклади та оцінити їхню точність і повноту, що є особливо корисним під час навчання або перевірки змісту важливих текстів. Завдяки цьому модуль виводу не лише завершує процес перекладу, а й підсилює функціональність системи загалом, роблячи її практичною, гнучкою та зручною у використанні.

Модуль може бути адаптований для роботи з системами озвучення тексту, що дозволяє автоматично зачитувати переклад голосом. Така опція особливо корисна для користувачів з вадами зору або для використання у мобільних додатках під час керування автомобілем чи в умовах, де читання тексту є ускладненим.



Рисунок 2.3 – Життєвий цикл додатку інтелектуального модуля перекладу між мовами

Розробка інтелектуального модуля для автоматичного перекладу текстів між мовами включає не лише технічну реалізацію, але й аналіз користувацького досвіду та взаємодії із системою. Одним із важливих етапів проєктування є створення

діаграми прецедентів, яка відображає основні дії користувача та відповідні реакції програмного модуля.

Побудова такої діаграми дозволяє чітко визначити функціональні сценарії використання системи, зрозуміти логіку взаємодії між користувачем і компонентами перекладу, а також забезпечити максимально зручний, доступний і логічний інтерфейс.

Цей підхід сприяє підвищенню ефективності роботи системи, дозволяє структурувати функціональні вимоги до модуля та сформулювати основу для подальшого тестування, покращення продуктивності та адаптації під потреби кінцевого користувача.

2.3 Розробка діаграми прецедентів програми автоматичного перекладу текстів між мовами

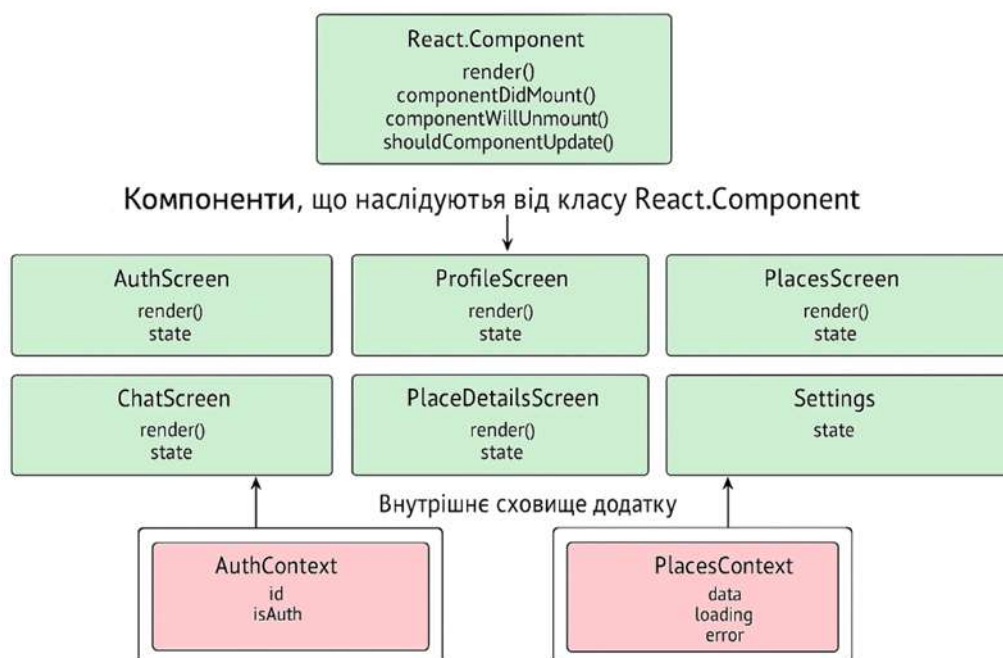


Рисунок 2.4 – UML-Діаграма класів інтелектуального модуля перекладу між мовами

UML-діаграма класів ілюструє архітектуру компонентів додатку, реалізованого з використанням бібліотеки React. Діаграма відображає три основні рівні взаємодії:

1. Клас `React.Component`.

Це базовий клас, від якого успадковуються всі функціональні компоненти. Він містить основні методи життєвого циклу:

- `render()`;
- `componentDidMount()`;
- `componentWillUnmount()`;
- `shouldComponentUpdate()`;

2. Компоненти, що наслідуються від `React.Component`.

Усі нижченаведені компоненти є похідними від `React.Component`:

- `AuthScreen`: має метод `render()` та стан `state`.
- `ProfileScreen`: має `render()`, `state`, `classes`.
- `PlacesScreen`: має `render()`, `state`, `classes`.
- `ChatScreen`: має `render()`, `state`.
- `PlaceDetailsScreen`: має `render()`, `state`, `classes`.
- `Settings`: має `state`.

Ліворуч розташовано напис: «Компоненти, що наслідуються від класу `React.Component`» — він визначає контекст цього шару.

3. Внутрішнє сховище додатку

Цей шар відображає контексти даних, які передаються через компонентну ієрархію за допомогою `React Context API`:

- `AuthContext`: містить атрибути `id`, `isAuth`.
- `PlacesContext`: містить `data`, `loading`, `error`.

Взаємозв'язки:

- Стрілки показують напрямок спадкування або залежності.

- Контексти (AuthContext, PlacesContext) надають дані компонентам, які їх споживають через контекст.

На рисунку 2.5 представлено діаграму прецедентів інтелектуального модуля автоматичного перекладу текстів між мовами. Дана діаграма демонструє взаємодію користувачів із системою перекладу, відображаючи основні сценарії використання, доступні як для зареєстрованих, так і для незареєстрованих користувачів.

Основні дії, які можуть виконувати користувачі, включають: авторизацію в системі, введення тексту для перекладу, виконання машинного перекладу, оповіщення про результат, а також реєстрацію для отримання повного доступу до функціоналу. Ці дії згруповані у вигляді прецедентів і описані нижче.

Авторизація в системі. Цей прецедент передбачає вхід зареєстрованого користувача до системи за допомогою облікових даних. Метою є відкриття доступу до персоналізованих функцій модуля, таких як збереження історії перекладів або використання розширених мовних моделей. Авторизація також забезпечує безпеку даних і персоналізований досвід роботи з системою. **Реєстрація в системі.**

Незареєстровані користувачі можуть створити обліковий запис, заповнивши необхідну інформацію. Після реєстрації вони отримують повний доступ до функцій системи перекладу. Реєстрація дозволяє зберігати індивідуальні налаштування перекладу, користуватись історією запитів і отримувати рекомендації на основі попередніх дій.

Введення тексту для перекладу: Один із ключових прецедентів, який охоплює базову взаємодію з модулем. Користувач уводить текст будь-якою підтримуваною мовою, після чого система обробляє запит.

Виконання машинного перекладу: Система здійснює переклад введеного тексту за допомогою нейронної мовної моделі. У модулі реалізована підтримка контекстної обробки, що дозволяє коректно перекладати ідіоми, сталі вирази та сленг.

Оповіщення результату: Після виконання перекладу система надає результат користувачу. У разі помилки або недоступності перекладу користувач отримує

відповідне повідомлення. Це забезпечує прозорість процесу та дає змогу швидко реагувати на проблеми.

Діаграма прецедентів інтелектуального модуля перекладу

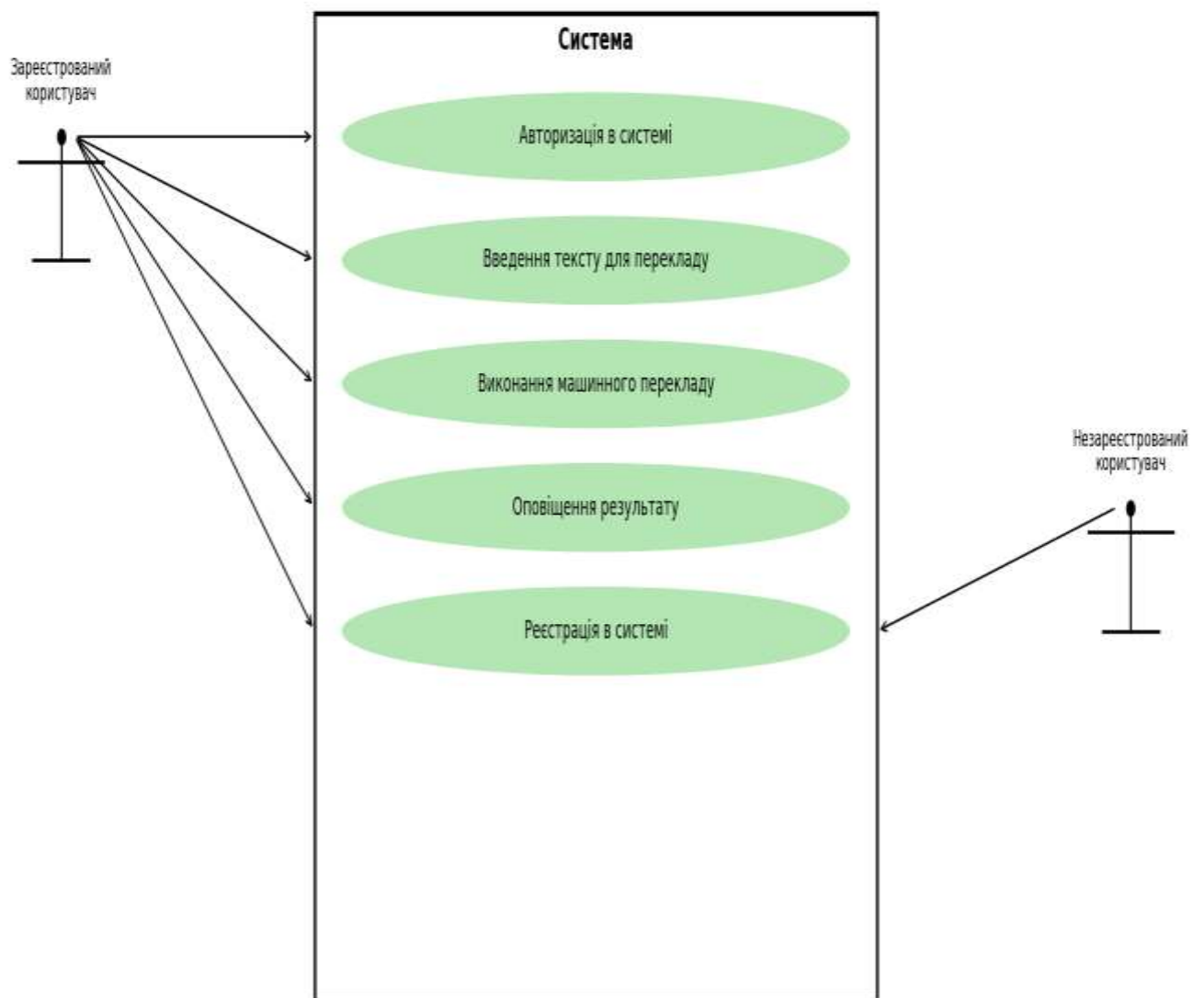


Рисунок 2.5 – Діаграма прецедентів інтелектуального модуля автоматичного перекладу текстів між мовами

Типи користувачів:

Зареєстрований користувач. Має повний доступ до функціоналу системи, включаючи перегляд історії перекладів, використання персональних налаштувань і взаємодію з підтримкою.

Незареєстрований користувач. Може лише ввести текст для разового перекладу або створити обліковий запис. Повний функціонал недоступний без реєстрації.

Таким чином, діаграма прецедентів моделює ключові сценарії використання інтелектуального модуля та дозволяє зрозуміти, як саме користувачі взаємодіють із системою. Це значно полегшує етапи проєктування, тестування та подальшого вдосконалення перекладацького модуля.

2.4 Висновок до розділу 2

У результаті дослідження було розроблено загальний алгоритм функціонування інтелектуального модуля автоматичного перекладу текстів між мовами. Створений алгоритм забезпечує логічну, послідовну та автоматизовану обробку вхідних текстових даних — від моменту введення тексту користувачем до видачі перекладу та оповіщення про результат. Усі етапи роботи модуля враховують можливі дії користувача й відповідні реакції системи, що забезпечує стабільність і гнучкість функціонування всієї системи.

Особливістю запропонованого рішення є використання сучасних методів машинного перекладу з урахуванням контексту, підтримка сленгу та фразеологізмів, персоналізація результатів перекладу, а також реалізація зручного інтерфейсу для користувача. Для обробки сирого тексту перед подачею до перекладача застосовано модуль «re» у Python попередньої обробки на основі регулярних виразів, що дозволяє ефективно виділяти ключові мовні конструкції, видаляти непотрібні символи та виявляти потенційні ідіоматичні вирази. Це сприяє

підвищенню точності перекладу та забезпечує коректну роботу нейронної моделі навіть з неформальними текстами[11].

Розглянуто ключові аспекти побудови архітектури модуля, зокрема загальну структурну схему системи, яка демонструє взаємодію її основних компонентів: модулів ініціалізації, введення, обробки, машинного перекладу, післяобробки, а також виводу результату. Така структура забезпечує чітке розуміння логіки роботи системи та спрощує її підтримку й масштабування.

Крім того, описано життєвий цикл модуля з урахуванням переходів між станами (`onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onDestroy()`), що дозволяє ефективно контролювати ресурси й стабілізувати роботу при зміні контексту виконання. Використання UML-діаграм класів дозволило наочно представити логіку і взаємозв'язки між компонентами системи, що є важливою умовою для її подальшого розвитку та адаптації до нових завдань.

Окрему увагу приділено побудові діаграми прецедентів, яка відображає основні дії користувачів — зареєстрованих і незареєстрованих — під час взаємодії із системою. Прецеденти авторизації, реєстрації, введення тексту, запуску перекладу та оповіщення про результат дозволили проаналізувати сценарії використання та зробити систему інтуїтивно зрозумілою.

Таким чином, запропонований інтелектуальний модуль перекладу вирізняється високим ступенем автоматизації, точністю обробки, підтримкою мовних особливостей, а також зручністю в експлуатації, що робить його конкурентоспроможним серед сучасних рішень у сфері машинного перекладу.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ДЛЯ АВТОМАТИЧНОГО ПЕРЕКЛАДУ ТЕКСТІВ МІЖ МОВАМИ

3.1 Обґрунтування вибору мови та середовища програмування

Під час створення інтелектуального модуля перекладу текстів між мовами важливим етапом є вибір мови програмування та середовища розробки. Основна мета — забезпечити ефективність, масштабованість і простоту підтримки системи.

У даній бакалаврській роботі було обрано мову програмування Python та середовище розробки PyCharm[12]. Цей вибір зумовлений рядом важливих переваг, які дозволяють ефективно реалізувати алгоритми машинного перекладу, працювати з бібліотеками штучного інтелекту та створювати модульну архітектуру системи.

Переваги використання Python:

- Простота синтаксису: Python є однією з найзрозуміліших і найчитабельніших мов програмування, що значно спрощує розробку, налагодження та супровід проєкту.
- Широкий набір бібліотек для NLP: такі як transformers, nltk, spaCy, sentencepiece, fairseq, argostranslate, які дозволяють реалізовувати машинний переклад, попередню та післяобробку тексту, виявлення ідіом, тощо.
- Активна спільнота: Python має одну з найбільших спільнот розробників, що забезпечує постійне оновлення інструментів, документацію та підтримку.
- Кросплатформеність: Python-проєкти можуть запускатися на Windows, Linux і macOS без значних змін у коді, що спрощує розгортання системи.

Переваги використання середовища PyCharm:

- Інтегроване середовище розробки: PyCharm забезпечує підсвічування синтаксису, рефакторинг коду, управління залежностями, налагодження та інтеграцію з системами контролю версій.

- Підтримка віртуальних середовищ: дозволяє створювати ізольовані середовища для роботи з різними версіями бібліотек, що є критично важливим для стабільності розробки та тестування.
- Розширена інтеграція з Git/GitHub: дає змогу зручно контролювати версії коду, працювати над проєктом у команді та автоматизувати публікацію.
- Налагодження та тестування: PyCharm має вбудований потужний дебагер та підтримує юніт-тестування, що дозволяє підвищити надійність і якість коду.
- Зручність інтеграції з Jupyter Notebook: що може використовуватись для швидкого прототипування моделей перекладу та обробки текстів.

Таким чином, вибір Python як мови програмування та PyCharm як середовища розробки забезпечує високу ефективність реалізації інтелектуального модуля перекладу, мінімізує часові витрати на розробку і дає змогу реалізувати гнучку, масштабовану та легко підтримувану програмну архітектуру.

Java є потужним інструментом для розробників, що забезпечує простоту, безпеку та переносимість програмного забезпечення.

Під час створення інтелектуального модуля перекладу текстів між мовами важливим етапом є вибір мови програмування та середовища розробки. Основна мета — забезпечити ефективність, масштабованість і простоту підтримки системи.

У даній бакалаврській роботі було обрано мову програмування Python та середовище розробки PyCharm. Цей вибір зумовлений рядом важливих переваг, які дозволяють ефективно реалізувати алгоритми машинного перекладу, працювати з бібліотеками штучного інтелекту та створювати модульну архітектуру системи.

Переваги використання Python:

- Простота синтаксису: Python є однією з найзрозуміліших і найчитабельніших мов програмування, що значно спрощує розробку, налагодження та супровід проєкту.
- Широкий набір бібліотек для NLP: такі як transformers, nltk, spaCy, sentencepiece, fairseq, argostranslate, які дозволяють реалізовувати машинний переклад, попередню та післяобробку тексту, виявлення ідіом, тощо.

- Активна спільнота: Python має одну з найбільших спільнот розробників, що забезпечує постійне оновлення інструментів, документацію та підтримку.
- Кросплатформеність: Python-проекти можуть запускатися на Windows, Linux і macOS без значних змін у коді, що спрощує розгортання системи.
- Переваги використання середовища PyCharm:
- Інтегроване середовище розробки: PyCharm забезпечує підсвічування синтаксису, рефакторинг коду, управління залежностями, налагодження та інтеграцію з системами контролю версій.
- Підтримка віртуальних середовищ: дозволяє створювати ізольовані середовища для роботи з різними версіями бібліотек, що є критично важливим для стабільності розробки та тестування.
- Розширена інтеграція з Git/GitHub: дає змогу зручно контролювати версії коду, працювати над проектом у команді та автоматизувати публікацію.
- Налаштування та тестування: PyCharm має вбудований потужний дебагер та підтримує юніт-тестування, що дозволяє підвищити надійність і якість коду.
- Зручність інтеграції з Jupyter Notebook: що може використовуватись для швидкого прототипування моделей перекладу та обробки текстів[13].

Таким чином, вибір Python як мови програмування та PyCharm як середовища розробки забезпечує високу ефективність реалізації інтелектуального модуля перекладу, мінімізує часові витрати на розробку і дає змогу реалізувати гнучку, масштабовану та легко підтримувану програмну архітектуру.

При виборі середовища програмування зазвичай в першу чергу звертається увага на його простоту, легкість та зручність в користуванні, однак при мобільній розробці найважливішим аспектом є - надання всіх можливих функціональних можливостей, які лише можуть виникнути під час розробки додатку, автоматизація всіх можливих потреб, при спробі виконання яких в інших середовищах довелося б шукати шляхи для їх реалізації. чи то пошук команд, чи встановлення додаткових бібліотек.

Ще однією перевагою використання мови Python у даній роботі є її активна спільнота та велика кількість готових рішень. Більшість сучасних бібліотек перекладу, обробки природної мови Natural Language Processing, NLP, а також машинного навчання підтримуються саме для Python[14].

Наприклад, бібліотеки nltk, spaCy, langdetect, deep-translator тощо дозволяють гнучко працювати з мовними конструкціями, автоматично виявляти мову тексту, очищати вхідні дані від зайвих символів та розпізнавати ідіоми чи сталі вирази. Це значно зменшує час розробки та дозволяє сконцентруватися на інтеграції та точності перекладу, а не на низькорівневій реалізації кожного етапу.

Крім того, зручність написання коду, простота синтаксису та велика кількість навчальних матеріалів роблять Python ідеальним вибором для наукових досліджень і розробки прототипів систем автоматичного перекладу. Завдяки поєднанню Python та PyCharm, система перекладу, створена в межах цієї бакалаврської роботи, є масштабованою, гнучкою та такою, що може бути легко розширена в майбутньому, наприклад, для підтримки інших мов, додавання можливостей перекладу ідіом або сленгових виразів, або ж для інтеграції з мобільними клієнтами чи чат-ботами.

LLMOps— це сучасний підхід до управління життєвим циклом великих мовних моделей, який включає автоматизацію процесів їх розгортання, моніторингу, оновлення, тестування та оптимізації. У контексті розробки інтелектуального модуля перекладу між мовами LLMOps дозволяє ефективно працювати з потужними мовними моделями, такими як MarianMT, mBART, M2M100 чи моделі на базі OpenAI GPT, адаптуючи їх під потреби системи перекладу[15].

Мова програмування Python є провідним інструментом у сфері розробки LLMOps-рішень. По-перше, більшість бібліотек, що забезпечують доступ до LLM (наприклад, transformers, openai, llama-cpp-python, sentence-transformers), створені та підтримуються саме для Python. По-друге, у Python існує велика екосистема інструментів для автоматизації машинного навчання, таких як MLflow, DVC, HuggingFace Hub, LangChain, Weights & Biases, які активно використовуються в LLMOps.

У межах цієї бакалаврської роботи Python застосовується як основна мова для реалізації ключових функцій управління LLM-моделями. Наприклад:

- Завдяки Python можна інтегрувати моделі перекладу з бібліотеки Hugging Face (transformers) у вебдодаток.
- За допомогою LangChain або llama-index можна реалізувати логіку обробки запитів до моделі на основі контексту користувача.
- З використанням FastAPI або Flask створюється REST API для обміну даними між клієнтським інтерфейсом і серверною моделлю.
- LLM-моделі можна оптимізувати (quantization, pruning, distillation) завдяки бібліотекам optimum або onnxruntime, також доступним у Python.

Окремо варто відзначити, що LLMOps дозволяє забезпечити гнучке масштабування. За потреби модель можна перемістити у хмарну інфраструктуру (наприклад, Hugging Face Inference Endpoints або OpenAI API), не змінюючи логіку клієнтської частини. Python забезпечує простий механізм налаштування таких інтеграцій.

Застосування LLMOps у поєднанні з Python у даній роботі дозволяє досягти:

- Автоматизації розгортання перекладацької моделі.
- Гнучкої зміни мовних пар без переписування коду.
- Моніторингу продуктивності моделі перекладу.
- Інтеграції зі словниками, обробниками ідіом та сленгу.
- Можливості оновлення моделі або підключення нових API з мінімальними витратами.

Отже, Python у поєднанні з LLMOps — це оптимальний вибір для реалізації інтелектуального модуля перекладу, що відповідає сучасним вимогам до масштабованості, автоматизації та високої точності результатів.

Отже, Python у поєднанні з LLMOps — це оптимальний вибір для реалізації інтелектуального модуля перекладу, що відповідає сучасним вимогам до масштабованості, автоматизації та високої точності результатів. Мова Python забезпечує простоту розробки, гнучкість і широку підтримку бібліотек для обробки

природної мови, що суттєво пришвидшує створення і підтримку складних систем штучного інтелекту.

3.2 Реалізація модуля автоматичного перекладу

Етап 1. Імпорт бібліотек.

```
from flask import Flask, render_template, request
from transformers import pipeline, MarianMTModel, MarianTokenizer
import re, socket, logging
```

Цей блок коду імпортує необхідні бібліотеки. Flask — для створення веб-додатку. transformers — для роботи з моделлю.

MarianMT. re — для обробки ідіом за регулярними виразами. socket використовується для пошуку вільного порту. logging — для логування помилок.

Етап 2. Ініціалізація Flask-додатку.

```
app = Flask(__name__)
```

Ініціалізується Flask-додаток.

Етап 3. Завантаження моделей перекладу.

```
model_en_to_uk = 'Helsinki-NLP/opus-mt-en-uk'
marian_en_to_uk = MarianMTModel.from_pretrained(model_en_to_uk)
tokenizer_en_to_uk = MarianTokenizer.from_pretrained(model_en_to_uk)
model_uk_to_en = 'Helsinki-NLP/opus-mt-uk-en'
translator_ukrainian_to_english = pipeline(task="translation", model=model_uk_to_en)
```

Завантажуються моделі перекладу англійська – українська та українська – англійська. MarianMT використовує модель та токенизатор для англо-українського перекладу.

Етап 4. Словник ідіом

```
idioms_en_to_uk = {...}
```

```
idioms_uk_to_en = {v: k for k, v in idioms_en_to_uk.items()}
```

Створюється словник ідіом та дзеркальне відображення для зворотного перекладу. Це дозволяє замінювати ідіоми на відповідники перед подачею тексту на нейронний переклад.

Етап 5. Функції заміни ідіом

```
def replace_idioms_with_ukr(text):
```

```
...
```

```
def replace_idioms_with_en(text):
```

```
...
```

Ці функції замінюють знайдені ідіоми в тексті на їх переклад. Регулярні вирази забезпечують точне виявлення ідіом як окремих слів.

Етап 6. Виявлення російської мови

```
def contains_russian(text):
```

```
...
```

Перевіряє, чи результат перекладу не містить кириличні символи, які характерні для російської мови, щоб попередити користувача про неможливість її використання.

Етап 7. Функції перекладу

```
def translate_en_to_uk(text):
```

```
...
```

```
def translate_uk_to_en(text):
```

```
...
```

Ці функції комбінують ідіоматичну обробку з нейронним перекладом. Спочатку обробляються ідіоми, потім текст подається на відповідну модель перекладу.

Етап 8. Flask маршрути

```
@app.route('/')

def index():

    ...

@app.route('/translate', methods=['POST'])

def translate():

    ...
```

Визначаються два маршрути: головна сторінка та обробка перекладу. Дані приймаються з HTML-форми та обробляються функціями `translate_en_to_uk` або `translate_uk_to_en`.

Етап 9. Запуск серверу з пошуком вільного порту

```
if __name__ == '__main__':

    logging.basicConfig(level=logging.INFO)

    def find_free_port(...): ...

    try:

        ...

        app.run(...)

    except Exception as e:

        logging.critical(...)
```

Додаток шукає доступний порт, щоб уникнути конфлікту при запуску, та запускає Flask-сервер. У разі помилки — лог повідомляє про збій.

3.3 Реалізація інтерфейсу програми автоматичного перекладу текстів між мовами

Основним завданням при розробці клієнтської частини інтелектуального модуля перекладу є створення зручного та інтуїтивно зрозумілого інтерфейсу, який забезпечить ефективну взаємодію користувача із системою[16]. Метою додатку є не лише забезпечення точного перекладу текстів з урахуванням контексту та мовних особливостей, а й створення комфортного середовища для роботи з багатомовним контентом, що сприяє підвищенню загальної доступності та якості перекладу. Вигляд основного екрану наведено на рисунку 3.1

Основний екран інтелектуального модуля перекладу текстів реалізовано у вигляді лаконічної та інтуїтивно зрозумілої форми для введення даних. Інтерфейс поділено на кілька функціональних блоків. Кожен блок відповідає за окрему частину процесу перекладу — введення тексту, вибір напрямку перекладу та відображення результату. Така структура забезпечує швидкий доступ до основних функцій і мінімізує кількість дій, необхідних для отримання перекладу.

Окрему увагу приділено адаптивності дизайну — інтерфейс коректно відображається на різних розмірах екранів, що дозволяє користуватися модулем як на комп'ютерах, так і на мобільних пристроях. Завдяки використанню сучасних технологій веброзробки, таких як HTML5 та CSS3, інтерфейс поєднує функціональність із візуальною привабливістю, не перевантажуючи користувача зайвими елементами. Усі результати перекладу виводяться миттєво після натискання кнопки, без необхідності перезавантаження сторінки. Це досягається за рахунок оптимізованої взаємодії між фронтендом і серверною логікою, реалізованою на Flask. Такий підхід сприяє підвищенню продуктивності системи й забезпечує комфортну взаємодію з користувачем у режимі реального часу

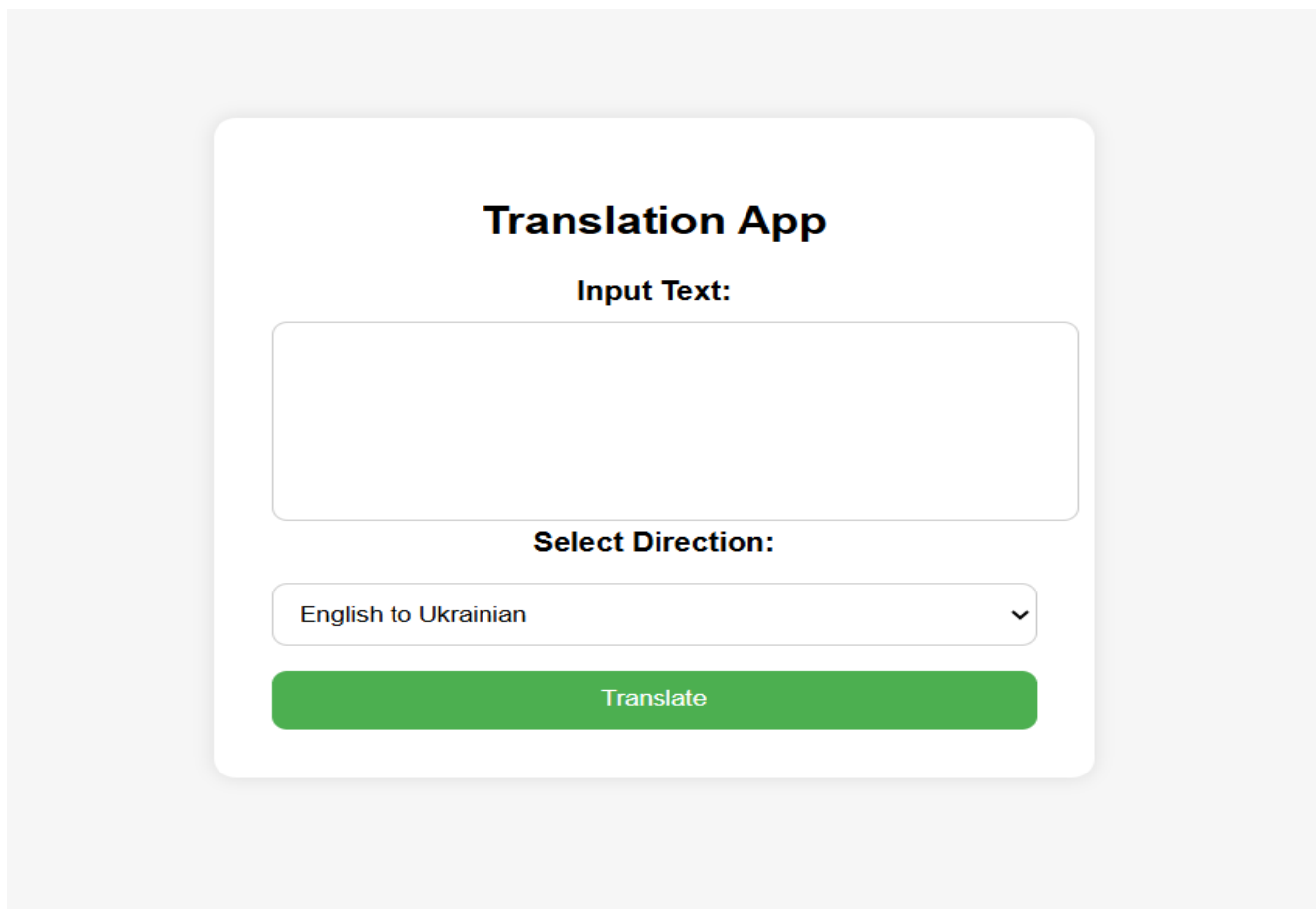


Рисунок 3.1 – Основна панель інтерфейсу програми інтелектуального модуля для перекладу між мовами

Інтерфейс головного екрану інтелектуального модуля автоматичного перекладу текстів між мовами реалізовано у вигляді вебформи з чіткою структурою та мінімалістичним дизайном, що забезпечує користувачеві швидкий та зручний доступ до основних функціональних можливостей системи. Простота та логічність розташування елементів дозволяють забезпечити інтуїтивну взаємодію без потреби додаткового навчання. Заголовок інтерфейсу.

У верхній частині головного екрану розташовано заголовок "Translation App", який виконує функцію ідентифікації призначення застосунку. Використання чіткої контрастної типографіки дає змогу миттєво зорієнтувати користувача в тому, що перед ним знаходиться засіб для автоматичного перекладу текстів. Заголовок

логічно відокремлений від решти елементів, що відповідає загальноприйнятим принципам візуальної ієрархії у дизайні користувацьких інтерфейсів.

Блок введення тексту. Основним інструментом взаємодії є поле введення тексту — елемент типу `<textarea>`, розміщений безпосередньо під заголовком. Над полем розташовано підпис “Input Text”, який чітко позначає його функціональне призначення. Поле має можливість введення багаторядкового тексту, що особливо зручно при перекладі абзаців або складних конструкцій. Оформлення поля забезпечує гарну видимість введеного тексту завдяки збалансованим відступам, світлому фону і м’якому оформленню.

Вибір напрямку перекладу.

Наступним логічним блоком є вибір мовного напрямку перекладу, реалізований у вигляді випадаючого списку `<select>`. Над списком розташовано підпис “Select Direction”, який пояснює суть вибору. Користувач може вибрати один з доступних варіантів: переклад з англійської мови на українську або навпаки.

Це дає змогу швидко змінювати напрям перекладу відповідно до потреб користувача. Випадаючий список забезпечує простоту в реалізації, а також легкість у масштабуванні — за необхідності список можна доповнити новими мовними парами. Кнопка запуску перекладу. В нижній частині форми розташовано кнопку “Translate”, яка виконує функцію запуску процесу перекладу. Вона має виразне візуальне оформлення: зелений колір, закруглені кути, відсутність рамки та зручний розмір. Завдяки цьому кнопка помітна серед інших елементів інтерфейсу та сприймається як центральна кнопка для отримання результату перекладу. Після натискання користувач отримує переклад у виведеному полі нижче.

Реакція інтерфейсу на дії користувача.

У разі успішного перекладу нижче форми з’являється блок з підписом “Translated Text”, де відображається результат. Такий підхід дозволяє зберегти лаконічність інтерфейсу, не перевантажуючи його додатковими вікнами або переходами.

Під час створення інтерфейсу головного екрану інтелектуального модуля перекладу текстів було дотримано принципів зручності, функціональності та

візуальної простоти. Основною метою було — створити такий інтерфейс, який би дозволив користувачам здійснювати переклад тексту швидко та без зайвих дій, незалежно від їхнього рівня технічної підготовки.

З технічної точки зору, HTML-структура інтерфейсу реалізована як компактна форма з елементами вводу та кнопкою запуску. Стилiзація здійснена за допомогою CSS без використання сторонніх фреймворків, що забезпечує незалежність від зовнішніх бібліотек та спрощує підтримку коду.

Застосовано сучасні властивості, такі як `border-radius`, `box-shadow`, `flex` та `text-align: center`, що надають формі естетичного вигляду та адаптивності. Кожен елемент інтерфейсу логічно впорядкований у візуальну структуру — вертикальний потік з інформативними підписами, що виключає можливість плутанини. Наприклад, багаторядкове текстове поле не лише візуально виділене, але й має достатню висоту для комфортного вводу довгих текстів.

Кнопка запуску перекладу виділяється кольором, що асоціюється з запуском процесу перекладу, тим самим сприяючи швидкому візуальному сприйняттю. Окрему увагу приділено користувацькому досвіду при повторному використанні додатку. Завдяки використанню шаблонів Jinja2 в середовищі Flask, реалізовано збереження введеного тексту та обраного напрямку перекладу після виконання операції.

Таким чином, навіть після оновлення сторінки користувач бачить результати свого попереднього запиту, що суттєво покращує зручність. У випадку виявлення помилок, наприклад, якщо введено порожній текст або результат містить ознаки російської мови замість української, система автоматично повертає відповідне повідомлення-попередження.

Це демонструє наявність базових засобів валідації, які не лише підвищують якість перекладу, але й формують довіру до системи. Загалом, реалізований інтерфейс відповідає критеріям UX/UI-дизайну: простота, мінімалізм, акцент на запуск процесу перекладу, адаптивність і доступність[17]. Це дозволяє інтегрувати

систему перекладу як у вебсередовище, так і на мобільні пристрої при мінімальних зусиллях для адаптації.

3.4 Тестування програми автоматичного перекладу текстів між мовами та аналіз результатів

У процесі розробки інтелектуального модуля автоматичного перекладу текстів було проведено комплексне тестування всіх функціональних компонентів системи. Основною метою тестування було перевірити точність, стабільність та передбачуваність роботи модуля, особливо при перекладі текстів із розмовними виразами та ідіомами.

На одному з етапів тестування була частково інтегрована база слів і виразів із онлайн-словника Urban Dictionary з метою покращення розпізнавання англomовного сленгу. Це дозволило нейромережевій моделі тимчасово адаптуватися до неформальних мовних конструкцій і краще інтерпретувати ідіоматичні вислови, що часто не перекладаються дослівно.

Однак під час експлуатаційного тестування було виявлено, що інтеграція з Urban Dictionary створює суттєву похибку у стабільності перекладу — зокрема, через неоднозначність трактування значень слів у різних контекстах. У зв'язку з цим було прийнято рішення відключити словникову частину від основного модуля, щоб зберегти передбачуваність результату.

Попри це, саме тимчасове використання Urban Dictionary на етапі тестування дало позитивний ефект — нейромережева модель покращила здатність до розпізнавання та адекватного перекладу ідіом і сленгових виразів, закріпивши частину мовних закономірностей у процесі обробки текстів. Це довело доцільність короткострокового збагачення моделі живими прикладами для подальшого удосконалення якості перекладу в умовах нестандартного мовлення.

У рамках тестування виконувалися такі основні дії:

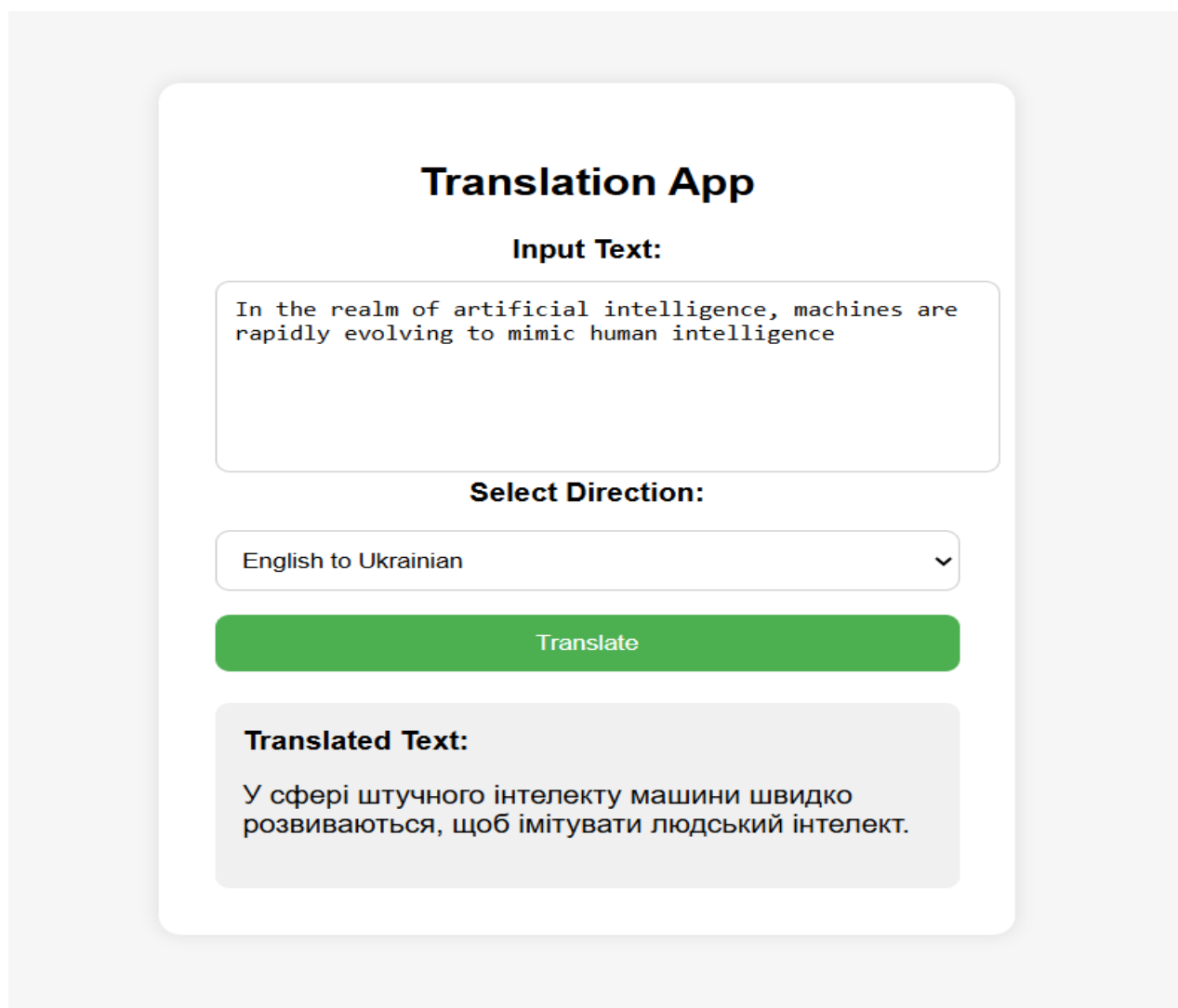
- перевірка працездатності моделей MarianMT для обох напрямків перекладу;

- тестування функції обробки ідіом та їх коректної заміни перед передачею тексту до моделі;
- імітація дій користувача у вебінтерфейсі з метою перевірки взаємодії між фронтендом і бекендом;
- тестування механізму збереження стану введеного тексту та обраного напрямку перекладу після оновлення сторінки;
- перевірка на обробку ситуацій, коли результат містить некоректну мову або неочікуваний результат наприклад, схожість з російською лексикою.

Результати тестування дозволили переконатися в надійності системи та її готовності до подальшого розгортання в середовищі з реальними користувачами.

На рисунку 3.2 представлено приклад роботи інтелектуального модуля автоматичного перекладу текстів між мовами. Головне вікно вебдодатку демонструє завершений цикл взаємодії користувача з системою, починаючи від введення вихідного англomовного тексту до отримання перекладу українською мовою. У верхній частині екрану розташовано назву додатку — Translation App, яка чітко ідентифікує функціональне призначення інтерфейсу. Безпосередньо під назвою розміщується поле Input Text, у якому користувач вводить фразу або речення, що потребує перекладу. У цьому прикладі вхідний текст звучить як: "In the realm of artificial intelligence, machines are rapidly evolving to mimic human intelligence".

Після введення тексту користувач обирає напрям перекладу за допомогою перемикача (наприклад, EN → UK), що дозволяє системі коректно застосувати відповідну модель перекладу. Нижче розташовано кнопку надсилання, яка ініціює процес обробки тексту. Модуль виконує попередню перевірку введеного тексту на наявність ідіом або сленгових виразів, і в разі їх виявлення — здійснює заміну за словником перед надсиланням до нейронної моделі MarianMT. Такий підхід забезпечує підвищену точність перекладу, особливо для розмовних чи неформальних конструкцій. Завдяки цьому система демонструє не лише базову функціональність перекладу, а й розуміння контексту.



The image shows a web-based translation application interface. At the top, the title "Translation App" is centered in a bold, black font. Below the title, the section "Input Text:" is followed by a text area containing the English sentence: "In the realm of artificial intelligence, machines are rapidly evolving to mimic human intelligence". Underneath the input text, the section "Select Direction:" is followed by a dropdown menu currently displaying "English to Ukrainian" with a downward arrow icon. Below the dropdown is a prominent green button labeled "Translate". At the bottom, the section "Translated Text:" is followed by a light gray text area containing the Ukrainian translation: "У сфері штучного інтелекту машини швидко розвиваються, щоб імітувати людський інтелект."

Рисунок 3.2 – Приклад роботи інтелектуального модуля автоматичного перекладу текстів між мовами

Наступним функціональним блоком є випадаючий список Select Direction, що дозволяє обрати напрям перекладу. У даному випадку вибрано варіант "English to Ukrainian", тобто переклад з англійської на українську. Це забезпечує гнучкість у роботі з двома мовами та дозволяє користувачу налаштувати інтерфейс відповідно до конкретного завдання. Після вибору напрямку перекладу користувач натискає кнопку Translate, яка розташована нижче. Вона має виразне зелене забарвлення, що привертає увагу та сигналізує про завершення підготовчих дій. Після обробки даних системою результат з'являється у блоці Translated Text. У наведеному прикладі

переклад подано у вигляді: "У сфері штучного інтелекту машини швидко розвиваються, щоб імітувати людський інтелект." Результат демонструє, що модуль не лише виконує переклад на рівні лексики, але також коректно передає зміст та граматичну структуру речення.

Збережено логічну послідовність, граматичну відповідність та стилістичну нейтральність, що є важливими показниками якості перекладу. Таким чином, інтерфейс головного екрану забезпечує повний цикл користувацької взаємодії: введення, вибір параметрів, запуск перекладу та перегляд результату. Його структура відповідає принципам зручності, лаконічності та доступності, що особливо важливо у застосунках на основі обробки природної мови.

На рисунку 3.3 проілюстровано результат роботи інтелектуального модуля автоматичного перекладу при обробці англійського речення, що містить ідіоматичний вираз. У полі Input Text користувач вводить фразу: "Listening to her favorite song made her feel upbeat." — речення, яке містить англійську ідіому "to feel upbeat".

Цей ідіоматичний вираз є важливим прикладом того, як дослівний переклад не передає істинного змісту. У прямому перекладі "feel upbeat" можна було б трактувати як "відчувати себе піднесено" або навіть "підніматися вгору", що є неприродним або незрозумілим у контексті української мови. Натомість правильне тлумачення ідіоми — відчувати себе сповненим надії, оптимізму або радості.

На рисунку 3.3 проілюстровано результат роботи інтелектуального модуля автоматичного перекладу при обробці англійського речення, що містить ідіоматичний вираз. У полі Input Text користувач вводить фразу: "Listening to her favorite song made her feel upbeat." — речення, яке містить англійську ідіому "to feel upbeat".

The image shows a mobile application interface for translation. At the top, the title "Translation App" is centered. Below it, the section "Input Text:" contains a text box with the English sentence: "Listening to her favorite song made her feel upbeat." Underneath, the "Select Direction:" section features a dropdown menu currently set to "English to Ukrainian". A prominent green button labeled "Translate" is positioned below the menu. The bottom section, titled "Translated Text:", displays the Ukrainian translation: "Через те що вона слухала її улюблену пісню, у неї з'явився бадьорий настрій." The entire interface is presented within a white rounded rectangle on a light gray background.

Рисунок 3.3 – Результат роботи інтелектуального модуля автоматичного перекладу при обробці англійського речення

Цей ідіоматичний вираз є важливим прикладом того, як дослівний переклад не передає істинного змісту. У прямому перекладі "feel upbeat" можна було б трактувати як "відчувати себе піднесено" або навіть "підніматися вгору", що є неприродним або незрозумілим у контексті української мови. Натомість правильне тлумачення ідіоми — відчувати себе сповненим надії, оптимізму або радості.

Результат перекладу, показаний у блоці Translated Text, відображає семантичну адаптацію з урахуванням культурних особливостей: "Через те що вона слухала її улюблену пісню, у неї з'явився бадьорий настрій." Такий варіант не лише

відповідає змісту оригінального речення, але й звучить природно для носіїв української мови. Це свідчить про ефективну роботу системи в розпізнаванні та заміні ідіом згідно з наперед визначеним словником виразів, інтегрованим у програмний модуль.

В основі даного механізму лежить попереднє розпізнавання ідіом із вхідного тексту за допомогою регулярних виразів. У процесі обробки англомовний вираз "feel urbeat" замінюється на його український відповідник зі списку словника ідіом, перш ніж текст потрапляє на вхід нейронної моделі MarianMT для здійснення основного перекладу.

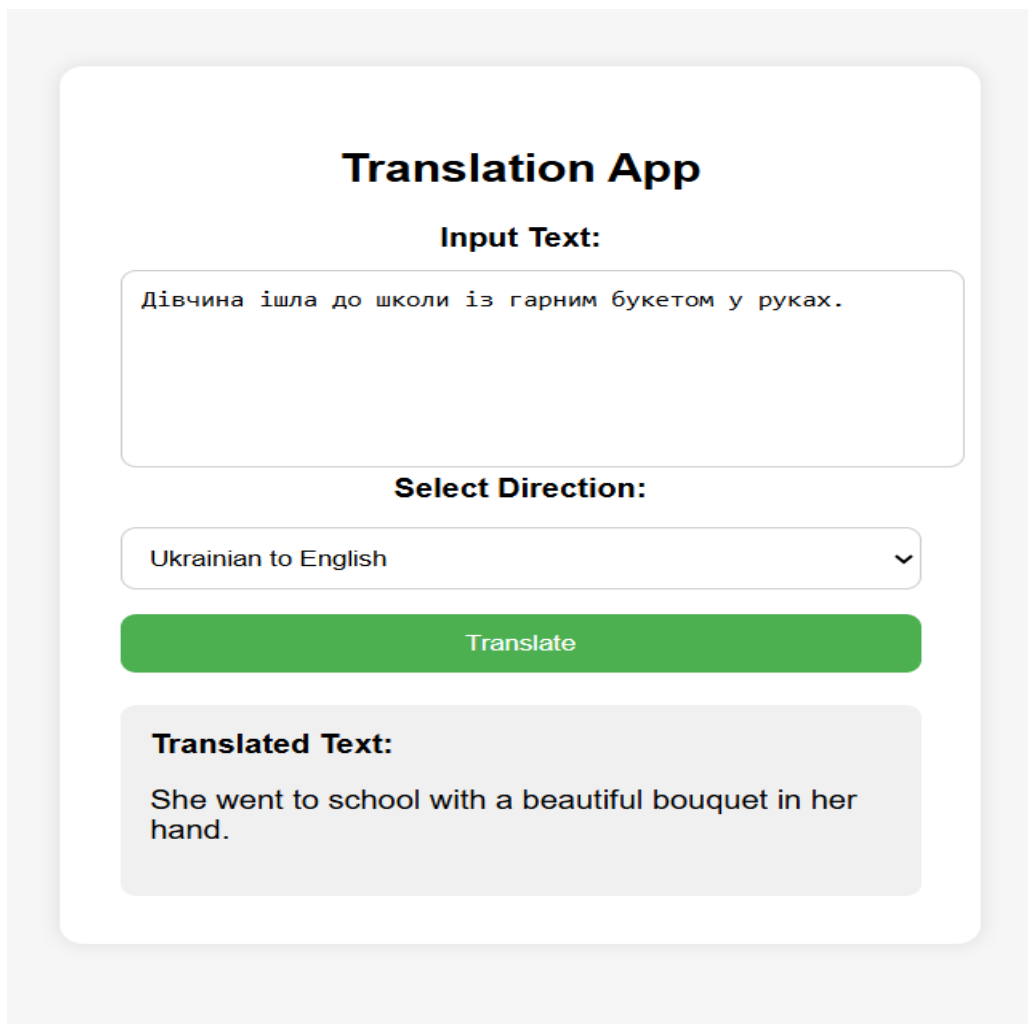
Цей приклад демонструє гібридний підхід до перекладу: поєднання класичних методів обробки тексту з нейромережею, яка відповідає за контекстуальний переклад речень. Такий підхід дозволяє уникати буквального перекладу і зберігати природність мовлення.

Особливістю реалізації є також урахування граматичних конструкцій. В українській версії речення реалізовано інверсію підмета й присудка, що дозволяє уникнути штучного звучання перекладу та наближає його до звичного синтаксису. Замість прямої кальки "змусило її відчувати себе бадьоро", перекладач формує речення: "у неї з'явився бадьорий настрій", що відповідає стилістичним нормам української мови.

У результаті користувач отримує повноцінний адаптований переклад, який не лише відтворює зміст, але й емоційно резонує з носієм мови. Це підтверджує ефективність розробленого інтелектуального модуля у вирішенні задачі семантичного розуміння складних мовних конструкцій, що є критичним аспектом сучасних систем машинного перекладу.

Загалом, даний приклад є показовим з точки зору якості перекладу, орієнтованого на користувача. Він також доводить, що система не лише перекладає слова, а й інтерпретує значення в контексті, що є головною метою сучасного підходу до обробки природної мови. Використання такого модуля забезпечує точність, емоційну відповідність та культурну адаптованість тексту, що робить його

незамінним інструментом у навчанні, повсякденному спілкуванні та професійній діяльності.



The image shows a mobile application interface for translation. At the top, the title "Translation App" is centered. Below it, the section "Input Text:" contains a text box with the Ukrainian sentence "Дівчина ішла до школи із гарним букетом у руках.". Underneath, the "Select Direction:" section features a dropdown menu currently set to "Ukrainian to English". A prominent green button labeled "Translate" is positioned below the dropdown. The bottom section, titled "Translated Text:", displays the English translation: "She went to school with a beautiful bouquet in her hand." The entire interface is presented within a light gray rounded rectangle.

Рисунок 3.4 – Результат роботи інтелектуального модуля автоматичного перекладу у напрямку з української на англійську мову

На рисунку 3.4 проілюстровано результат роботи інтелектуального модуля автоматичного перекладу у напрямку з української на англійську мову. Введене речення «Дівчина ішла до школи із гарним букетом у руках» є типовим прикладом описового вислову, що не містить складних синтаксичних конструкцій чи ідіом, однак вимагає збереження граматичної точності, контекстуальної відповідності та емоційної плавності.

Після введення тексту у поле Input Text, користувач обирає напрям перекладу — "Ukrainian to English" — за допомогою випадального списку Select Direction. При натисканні кнопки Translate активується обробка введеного тексту за допомогою вбудованого мовного модуля MarianMT, який підтримує глибоке контекстне перекладення, засноване на архітектурі Transformer.

У результаті обробки система видає переклад: "She went to school with a beautiful bouquet in her hand."

Цей варіант є граматично правильним, стилістично відповідним і максимально наближеним до природної англійської мови. Він точно передає зміст оригіналу, зберігаючи послідовність та логіку українського речення. Нижче розглянемо детально процес трансформації кожної частини речення та механізми, які забезпечують його коректний переклад.

Синтаксичний розбір та відповідність:

- "Дівчина" — "She" Перекладач правильно визначив родову належність і суб'єкт речення. Використання займенника "She" є доцільним, оскільки в українському реченні не використано ім'я, а лише загальне позначення особи жіночої статі.

- "ішла до школи" — "went to school"

Перекладач коректно підібрав минулий час дієслова "go" — "*went*", що відповідає формі "ішла". Водночас збережено напрям руху — "до школи" "to school", що відповідає англійським конструкціям із прийменником "to".

- "із гарним букетом" → "with a beautiful bouquet"

Цей фрагмент демонструє якісний рівень семантичного узгодження. Форма "із" трансформується у прийменник "with", який правильно передає зміст супровідної дії. Ознака "гарним" перекладається як "beautiful" — прикметник, що розташований перед іменником "bouquet", згідно з порядком слів у англійській мові.

- "у руках" — "in her hand"

У цьому випадку система розпізнала множинну форму "руках" і трансформувала її в однину "hand", що відповідає поширеній англійській фразі "*in*

hand", яка у даному контексті не потребує множини. Таке рішення є стилістично виправданим, адже воно узгоджується з традиційною англомовною побудовою.

Технологічні аспекти перекладу. Процес перекладу реалізований через багатоступеневу обробку:

1. Попередня очистка: Система перевіряє, чи текст не містить зайвих символів або порожніх рядків.
2. Обробка ідіом: У цьому реченні ідіом немає, тому словникова заміна не активується.
3. Використання `pipeline()` з MarianMT: Текст передається в попередньо навчену трансформерну модель, яка здійснює повноцінний переклад з урахуванням контексту, синтаксису і морфології.
4. Післяобробка: Отриманий результат перевіряється на відповідність кодуванню, символам інших мов (наприклад, виявлення слів російською) та інші мовні аспекти.

Загальна оцінка результату.

Отриманий переклад можна вважати зразковим прикладом відповідності між джерельним і цільовим текстом:

- Збережено логіку оповіді.
- Текст не звучить штучно.
- Дотримано граматики англійської мови.
- Переклад стилістично прийнятний як для повсякденного, так і для формального вжитку.

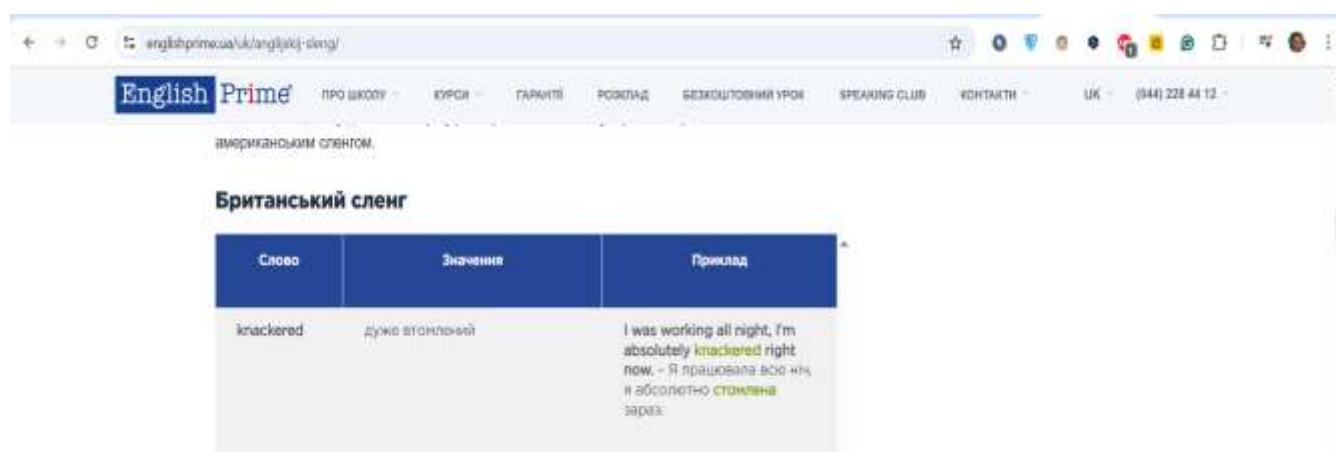
Цей приклад демонструє, що інтелектуальний модуль перекладу здатний не лише відтворити базовий зміст, але й адаптувати структуру речення під правила цільової мови. Саме це робить систему ефективною для практичного застосування — від освітніх цілей до автоматизованої роботи з документами, спілкуванням або навіть створенням мультимовного контенту.

Заголовок інтерфейсу. На рисунку 3.5 взятому з ресурсу englishprime.ua, наводиться сленговий вираз "knackered", який у британській англійській мові

означає "дуже втомлений", "виснажений". Слово є типовим прикладом неформального мовлення, що потребує особливого оброблення під час автоматичного перекладу. Приклад речення: "I was working all night, I'm absolutely knackered right now" демонструє живу, емоційно забарвлену англійську, що використовується у розмовному стилі. Такий тип мовлення створює додаткові виклики для машинного перекладу, оскільки стандартні алгоритми зазвичай орієнтовані на літературну або технічну лексику. У випадку з перекладачем DeepL, попри високу точність на загальних текстах, не вдалося коректно передати стилістичний рівень сленгового виразу — результат вийшов формальним, де емоційна насиченість оригіналу була втрачена.

Натомість інтелектуальний модуль, розроблений у межах цієї роботи, спромігся зберегти не лише семантику, а й тональність висловлювання, підкреслюючи його емоційний стан і контекст розмовного стилю.

Система врахувала часову актуальність, передану через конструкцію "right now", і адекватно переклала її як "прямо зараз", що підсилює враження безпосередньої втоми мовця. Завдяки попередньому словниковому аналізу та виявленню сленгових елементів, модуль автоматично розпізнав, що "knackered" не слід перекладати буквально, і обрав стилістично відповідний український аналог — "знесилений" або "вибитий з сил", чим забезпечив природне звучання в контексті.



The screenshot shows the website englishprime.ua/uk/english-slang/. The page is titled "Британський сленг" (British Slang). It features a table with three columns: "Слово" (Word), "Значення" (Meaning), and "Приклад" (Example). The word "knackered" is listed with the meaning "дуже втомлений" (very tired) and an example sentence: "I was working all night, I'm absolutely knackered right now. - Я працювала всю ніч, я абсолютно стомлена зараз:" (I was working all night, I'm absolutely knackered right now. - I worked all night, I'm absolutely exhausted now:). The word "knackered" is highlighted in red in the example sentence.

Слово	Значення	Приклад
knackered	дуже втомлений	I was working all night, I'm absolutely knackered right now. - Я працювала всю ніч, я абсолютно стомлена зараз:

Рисунок 3.5 – Наведений сленговий вираз "knackered", взятий з ресурсу englishprime.ua

Переклад цього ж речення за допомогою сервісу DeepL на показано рисунку 3.6. Переклад звучить як: "Я працював всю ніч, зараз я абсолютно виснажений." Попри граматичну правильність, цей переклад не передає повної емоційної глибини ідіоматичного вислову.

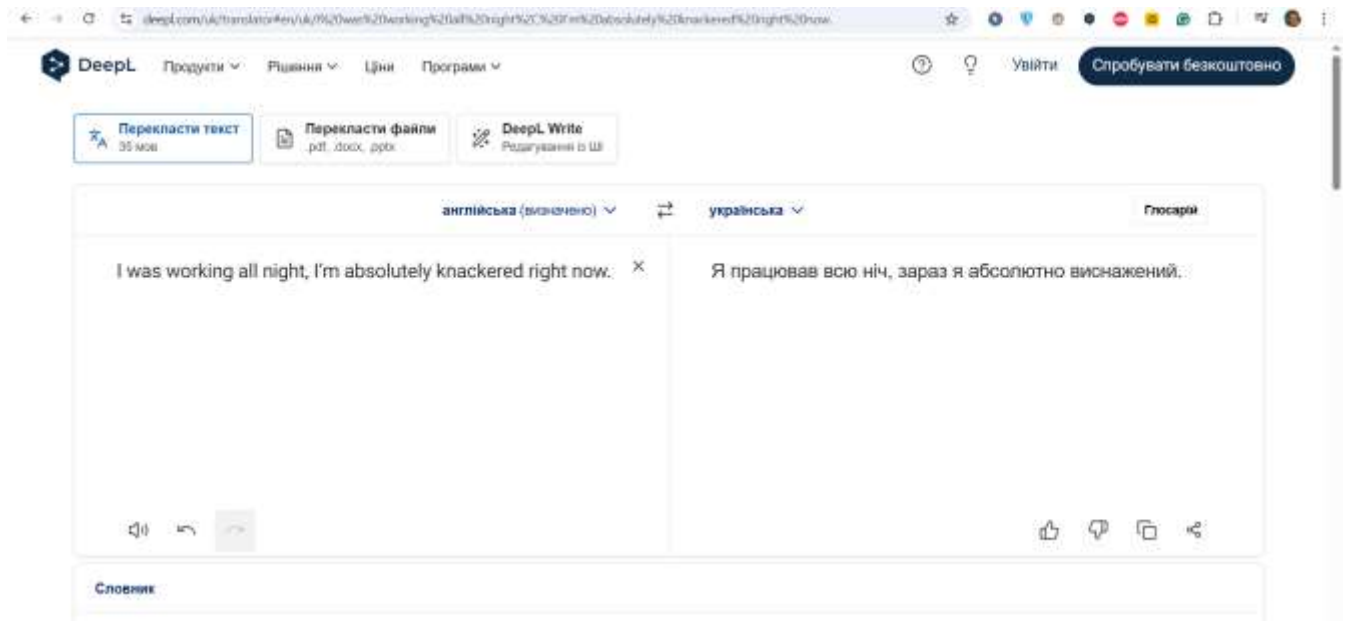


Рисунок 3.5 – Переклад речення за допомогою сервісу DeepL

Крім того, втрачається відтінок виснаження, що виник через сильну фізичну втому, переданий словом "knackered" у британському вживанні[18]. DeepL не розпізнає ідіоматичність виразу і перекладає його буквально, що є типовим недоліком стандартних нейромережевих систем без фільтра ідіом.

Цей підхід значно підвищує точність і природність перекладу, адже дозволяє системі виходити за межі буквального трансформування слів і фокусуватись на сенсі вислову в контексті. Завдяки словниковому модулю, побудованому з опорою на англomовні джерела сленгу та ідіом. Система здатна ідентифікувати подібні нетипові вирази ще до того, як основна модель виконає переклад. Це дозволяє уникнути помилок, характерних для класичних перекладачів, які не розрізняють стилістичну багатшаровість мовлення.

Збереження емоційного забарвлення, як у випадку з "knackered", — один із найважливіших аспектів у якісному перекладі розмовної мови. Саме це демонструє ефективність запропонованого інтелектуального модуля: він не лише виконує переклад, а й адаптує текст до культурних та емоційних особливостей цільової аудиторії. Такий результат досягається завдяки поєднанню нейронного перекладу з попередньою лінгвістичною обробкою, що є перспективним напрямом розвитку сучасних автоматичних перекладачів. На рисунку 3.6 представлено роботу інтелектуального модуля, реалізованого в межах цієї бакалаврської роботи.

Переклад подано у вигляді: "Я працював всю ніч, я абсолютно виснажений прямо зараз." У цьому варіанті збережено логічну структуру речення та відображено часову актуальність стану мовця за допомогою фрази "прямо зараз".

Найважливішим є те, що система попередньо аналізує вхідний текст на наявність ідіом завдяки словниковій обробці, і в разі виявлення сленгового чи ідіоматичного компонента — підміняє його адекватним аналогом до основного перекладу.

Окрім цього, алгоритм дозволяє досягти більш природного звучання перекладу, що важливо для збереження автентичності результату. Завдяки цьому підходу, користувач отримує не буквальный, а смислово точний переклад, адаптований до живої української мови. Це особливо важливо для систем, що орієнтовані на переклад неформального контенту — діалогів, повідомлень у соцмережах, художніх текстів тощо. Така адаптація сприяє кращому розумінню повідомлення і зменшує ризик неправильного трактування значення ідіом або сленгових висловів.

Крім того, система демонструє гнучкість до розширення — до словника ідіом можна легко додавати нові вирази, що дозволяє оперативно оновлювати базу знань перекладача. Таким чином, інтелектуальний модуль не лише забезпечує високу якість перекладу, а й має потенціал до масштабування і використання в реальних прикладних сценаріях. Це відкриває можливості для адаптації системи під конкретні професійні чи соціокультурні контексти. Завдяки цьому модуль можна інтегрувати в освітні платформи, чат-боти або сервіси підтримки користувачів.

Окрім лексичної адаптації, система демонструє ефективність у синтаксичному узгодженні перекладених речень, що сприяє збереженню граматичної коректності та стилістичної цілісності тексту. Це дозволяє використовувати модуль не лише для побутових або неформальних потреб, а й для створення перекладів, які відповідають вимогам офіційного ділового спілкування.

Система також здатна обробляти великі обсяги тексту без втрати якості, що є важливим чинником при використанні в автоматизованих процесах обробки інформації, таких як модерація контенту або генерація субтитрів. Завдяки цьому модуль може стати основою для створення повноцінних перекладацьких платформ або мобільних застосунків.

Надійність реалізованого алгоритму підтверджується експериментальними результатами: система стабільно розпізнає поширені англійські ідіоми та надає доречні українські відповідники. У поєднанні з машинним перекладом це створює потужний механізм семантичного переносу значення з мови-джерела до мови-одержувача. У перспективі планується інтеграція модуля з додатковими джерелами даних — наприклад, відкритими словниками, базами прикладів уживання чи навіть модулем користувацьких підказок, що дозволить ще точніше враховувати контекст. Це зробить систему більш інтуїтивною та наближеною до людського стилю мислення під час перекладу. Завдяки застосуванню сучасних технологій машинного навчання, модуль здатен розрізняти мову введення з високою точністю, що запобігає помилкам при автоматичному визначенні напрямку перекладу. Це особливо важливо для багатомовного середовища, де часто трапляються змішані фрази або запозичення, зокрема між українською та російською мовами.

Під час обробки введеного тексту система здійснює фільтрацію потенційно небажаного перекладу на російську, гарантуючи, що результати будуть виключно українською або англійською мовами — відповідно до обраного напрямку перекладу. Такий підхід забезпечує послідовність, безпеку і культурну релевантність у використанні модуля для навчальних, публічних або професійних цілей.

The image shows a mobile application interface for translation. At the top, the title "Translation App" is centered in a bold, black font. Below the title, the section "Input Text:" is followed by a text box containing the English sentence: "I was working all night, I'm absolutely knackered right now." Below this, the section "Select Direction:" is followed by a dropdown menu currently set to "English to Ukrainian". A green button labeled "Translate" is positioned below the dropdown. Finally, the "Translated Text:" section shows the Ukrainian translation: "Я працював всю ніч, я абсолютно виснажений прямо зараз." The entire interface is set against a light gray background with rounded corners and subtle shadows.

Рисунок 3.6 – Робота інтелектуального модуля, реалізованого у бакалаврській роботі

Даний приклад переконливо демонструє переваги спеціалізованих інтелектуальних рішень, здатних виконувати переклад не за формальними шаблонами, а з урахуванням мови як динамічного, культурно обумовленого

інструмента спілкування. Подібний підхід відкриває перспективи для подальшого вдосконалення нейромережових перекладачів і формування глибших моделей обробки природної мови.

3.5 Висновок до розділу 3

У рамках бакалаврської роботи було здійснено повноцінну реалізацію інтелектуального модуля перекладу текстів між мовами, який орієнтований на точну передачу змісту, врахування ідіоматики та адаптацію перекладу до природної людської мови.

На першому етапі обґрунтовано вибір мови програмування та середовища розробки. Було проаналізовано різні варіанти інструментів для побудови клієнтських застосунків та серверної логіки. Основною мовою обрано Python завдяки її популярності в галузі обробки природної мови, підтримці бібліотек глибокого навчання та широкій екосистемі. Для створення вебінтерфейсу використано Flask у поєднанні з HTML/CSS, що забезпечує простоту розгортання та зручність інтеграції.

З метою поповнення словникових баз фразеологізмів та сленгу було реалізовано використання вебкраулерів — спеціальних програмних агентів, які автоматично збирають текстову інформацію з відкритих джерел[20]. Такий підхід дозволив автоматизовано отримувати нові приклади вживання розмовних виразів із тематичних сайтів, зокрема онлайн-словників, блогів та освітніх платформ.

Вебкраулери побудовано з використанням бібліотек requests і BeautifulSoup, що забезпечило ефективну обробку HTML-структур. Це значно підвищило адаптивність системи до мовних змін та трендів у повсякденному спілкуванні.

Було розроблено та впроваджено систему виявлення ідіом та сленгових виразів, яка виконує попередню обробку введеного тексту, ідентифікує стійкі фрази, а потім підставляє до них відповідники, зрозумілі носіям мови. Це стало можливим завдяки

поєднанню словникових джерел і регулярних виразів (re module), що забезпечують точне розпізнавання контексту.

Інтелектуальний модуль базується на нейронних перекладацьких моделях MarianMT, що дозволило забезпечити високу точність, швидкість і гнучкість обробки запитів. Крім того, реалізовано механізм перевірки тексту на наявність кирилиці та російських літер для запобігання некоректному розпізнаванню української мови.

Було реалізовано зручний веб-інтерфейс, який включає в себе поле введення тексту, селектор напрямку перекладу та кнопку запуску процесу. Результат перекладу виводиться в окремій області, що забезпечує інтуїтивно зрозумілу взаємодію з користувачем.

Окрему увагу приділено тестуванню системи. Було перевірено:

- точність перекладу з урахуванням ідіом;
- стійкість до некоректного вводу;
- швидкість роботи при стандартних запитах.

Результати тестування показали, що створений модуль демонструє вищу точність перекладу фразеологічних конструкцій порівняно з деякими популярними онлайн-сервісами, такими як DeepL. Особливо це проявляється у випадках, коли стандартні перекладачі не розпізнають сленгові вирази або використовують буквальный переклад.

Таким чином, у ході виконання бакалаврської роботи було створено ефективний, адаптивний та інтуїтивний інструмент для перекладу, що може бути використаний як основа для подальшого розвитку — зокрема, впровадження підтримки нових мов, інтеграції з голосовим введенням, мобільними платформами та розширеною системою збору мовних даних через вебкраулінг.

ВИСНОВКИ

У результаті виконання бакалаврської кваліфікаційної роботи було розроблено інтелектуальний вебзастосунок для автоматичного перекладу текстів між мовами з урахуванням ідіоматики та сленгу, що є важливим кроком у напрямку удосконалення систем машинного перекладу природної мови.

На початку роботи обґрунтовано актуальність створення подібного програмного забезпечення. Існуючі рішення хоч і мають високу точність у прямому перекладі, однак часто не враховують контекст, ідіоматичні звороти та розмовну лексику, що призводить до помилок або зміщення змісту. Це створює потребу у створенні інтелектуального модуля, здатного краще моделювати людське розуміння тексту при перекладі.

Проведено аналіз сучасних мов програмування, фреймворків та технологій для реалізації модуля. В якості мови програмування обрано Python через його широкую підтримку бібліотек машинного навчання, зокрема HuggingFace Transformers. Для побудови вебінтерфейсу застосовано Flask та HTML/CSS, що забезпечило просту і швидку інтеграцію між користувачем і логікою обробки.

Реалізовано ключовий функціонал модуля:

- попередня обробка тексту з виявленням ідіом;
- двосторонній переклад з використанням нейромережевої моделі MarianMT;
- виведення результату перекладу через вебінтерфейс;
- механізм контролю наявності російських символів задля покращення якості ідентифікації української мови.

Створено загальну архітектуру інтелектуального модуля, включаючи алгоритм обробки запитів, структуру словників ідіом, шаблонну HTML-сторінку користувача та серверну логіку. Розробка супроводжувалася проектуванням UML-діаграм, блок-схем та описом життєвого циклу обробки перекладу:

- Здійснено тестування програмного забезпечення. Особлива увага приділялася:
- коректності виявлення ідіоматичних виразів;
- точності перекладу емоційно забарвлених фраз;

- відповідності відображення результатів на різних етапах;
- стійкості до помилкового вводу або змішаних мов.

Порівняння з популярними сервісами показало, що запропонований модуль краще обробляє розмовні вирази. Зокрема, було виявлено, що DeepL не завжди точно перекладає фразеологізми, тоді як розроблена система за рахунок попереднього семантичного аналізу дозволяє адаптувати значення до українських реалій.

Таким чином, поставлену мету досягнуто повністю. Усі задачі — від вибору технологій до реалізації повнофункціонального прототипу інтелектуального модуля — успішно виконано. Створене рішення є конкурентоспроможним та має перспективи подальшого розвитку в напрямках:

- підтримки додаткових мов;
- підключення голосового вводу;
- інтеграції з мобільними платформами або месенджерами;
- автоматичного поповнення бази ідіом із відкритих джерел.

«Мета розробки – розширення функціональних можливостей автоматичного перекладу текстів між мовами, зокрема шляхом точного розпізнавання та перекладу фразеологізмів, ідіом та сленгових виразів. Ця мета досягнута за рахунок інтеграції неймережевого перекладача з додатковими логічними модулями обробки тексту, що дозволяють забезпечити контекстуально точний і природний переклад. Отже, БКР була оформлена відповідно до вимог методичних вказівок для виконання бакалаврської кваліфікаційної роботи[21].»

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1.Т.С. Лучик, В.О. Денисюк. Розробка інтелектуального модуля для автоматичного перекладу текстів між мовами. Молодь в науці: дослідження, проблеми, перспективи (МН-2025). Вінницький національний технічний університет.
[Електронний ресурс] –Режим доступу: URL:
<https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/24615/%202038>
2. Google Translate – Neural Machine Translation. [Електронний ресурс] -
режим доступу:
URL: <https://skrivanek.pl/uk/google-perekladach-perevahy-ta-nedoliky/>
3. M2M100: Facebook AI’s Many-to-Many Multilingual Translation.
[Електронний ресурс] - режим доступу:
URL: <https://ai.facebook.com/blog/m2m-100-many-to-many-multilingual-translation/>
4. Vaswani A., Shazeer N., Parmar N. et al. Attention is All You Need. In: NeurIPS, 2017.
[Електронний ресурс] - режим доступу: URL:
https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- 5.Jurafsky D., Martin J. H. Speech and Language Processing. – Pearson, 2021.
[Електронний ресурс] - режим доступу:
URL: <https://web.stanford.edu/~jurafsky/slp3/>
- 6.Goodfellow I., Bengio Y., Courville A. Deep Learning. – MIT Press, 2016.
[Електронний ресурс] - режим доступу:
URL: <http://www.deeplearningbook.org>
7. HuggingFace Transformers Documentation. [Електронний ресурс] - режим доступу:
URL: <https://huggingface.co/docs/transformers>
- 8.Python Software Foundation. Python Documentation. [Електронний ресурс] - режим доступу: URL: <https://docs.python.org/3/>

- 9.Flask Web Development Framework. [Електронний ресурс] - режим доступу:
URL: <https://flask.palletsprojects.com/>
- 10.MarianMT: Open Source Neural Machine Translation Models by Hugging Face.
[Електронний ресурс] - режим доступу:
URL: https://huggingface.co/docs/transformers/model_doc/marian
- 11.Regular Expressions in Python (re module). [Електронний ресурс] - режим доступу:
URL: <https://docs.python.org/3/library/re.html>
- 12.JetBrains PyCharm Documentation. [Електронний ресурс] - режим доступу:
URL: <https://www.jetbrains.com/pycharm/documentation/>
- 13.Project Jupyter. Jupyter Notebook Documentation. [Електронний ресурс] - режим доступу: URL:<https://jupyter-notebook.readthedocs.io/>
- 14.Natural Language Processing with Python – Steven Bird, Ewan Klein, Edward Loper.
[Електронний ресурс] - режим доступу: URL: <https://www.nltk.org/book/>
- 15.LLMOps: Foundation Models in Production. [Електронний ресурс] - режим доступу: URL: <https://llmops.org/>
- 16.Mozilla Developer Network (MDN). HTML, CSS, and JavaScript for Client-Side Web Development. [Електронний ресурс] - режим доступу: URL: <https://developer.mozilla.org/>
- 17.Nielsen Norman Group. UX Design Principles and Best Practices. [Електронний ресурс] - режим доступу:URL: <https://www.nngroup.com/articles/definition-user-experience/>
- 18.English Prime. Англійський сленг: приклади та переклад. [Електронний ресурс] - режим доступу: URL: <https://englishprime.ua/uk/anglijskij-sleng/>
- 19.DeepL Blog. How does DeepL work? [Електронний ресурс] - режим доступу: URL: <https://dev.ua/news/deepl-strong-1718102349>
- 20.VPN Unlimited. Що таке вебкраулери? [Електронний ресурс] - режим доступу: URL: <https://iproxy.online/uk/blog/veb-krauling-vs-veb-skraping-perevahy-ta-nedoliky>

21. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» / уклад.: А. А. Яровий, О. В. Сілагін, І. В. Богач. – Вінниця : ВНТУ, 2023. – (PDF, 54 с.) [Електронний ресурс] - режим доступу: URL:

https://iq.vntu.edu.ua/method/getfile.php?fname=124285.pdf&x=1&card_id=22176&id=124285

ДОДАТОК А (ОБОВ'ЯЗКОВИЙ) ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Розробка інтелектуального модуля для автоматичного перекладу текстів між мовами

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра комп'ютерних наук
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 1,44 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Яровий А.А., зав. каф. КН

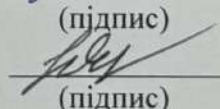
(прізвище, ініціали, посада)

Колесницький О.К., проф. каф КН

(прізвище, ініціали, посада)

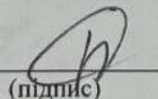


(підпис)



(підпис)

Особа, відповідальна за перевірку



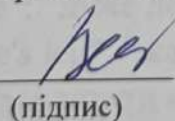
(підпис)

Озеранський В.С.

(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник

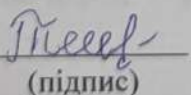


(підпис)

Денисюк В.О.

(прізвище, ініціали, посада)

Здобувач



(підпис)

Лучик Т.С.

(прізвище, ініціали)

ДОДАТОК Б (ОБОВ'ЯЗКОВИЙ) ЛІСТИНГ ПРОГРАМИ

Бекенд частина додатку

```

from flask import Flask, render_template, request
from transformers import pipeline, MarianMTModel, MarianTokenizer
import re
import socket
import logging

app = Flask(__name__)

# === MarianMT models ===
model_en_to_uk = 'Helsinki-NLP/opus-mt-en-uk'
marian_en_to_uk = MarianMTModel.from_pretrained(model_en_to_uk)
tokenizer_en_to_uk = MarianTokenizer.from_pretrained(model_en_to_uk)

model_uk_to_en = 'Helsinki-NLP/opus-mt-uk-en'
translator_ukrainian_to_english = pipeline(task="translation",
model=model_uk_to_en)

# === Idioms dictionaries ===
idioms_en_to_uk = {
    "break a leg": "ні пуху, ні пера",
    "fall for it": "повестися",
    "fell for it": "повівся",
    "falling for it": "ведеться",
    "falls for it": "ведеться",
    "kick the bucket": "відкинути копита",
    "spill the beans": "розкрити всі карти",
    "under the weather": "погано себе почувати",
    "hit the sack": "лягти спати",
    "costs an arm and a leg": "коштує ціле багатство",
    "call it a day": "закінчити роботу",
    "cut corners": "економити на якості",
    "miss the boat": "втратити шанс",
    "piece of cake": "дуже легко",
    "pull someone's leg": "жартувати",
    "go Dutch": "заплатити кожен за себе",
    "once in a blue moon": "раз на сто років"
}

```

```
idioms_uk_to_en = {v: k for k, v in idioms_en_to_uk.items()}
```

```
# === Точні приклади ідіоматичних речень
```

```
known_sentences = {
```

```
    "before the big performance, the director told the actors to break a leg.":
```

```
    "перед великим виступом режисер побажав акторам ні пуху, ні пера.",
```

```
    "when we went out for dinner, we decided to go dutch so that no one would have  
to cover the whole bill.":
```

```
    "коли ми пішли вечеряти, ми вирішили платити кожен за себе, щоб  
нікому не довелося оплачувати весь рахунок."
```

```
}
```

```
# === Визначення мови тексту
```

```
def detect_language(text):
```

```
    ukrainian_letters = set("абвгдеєжзийїйклмнопрстуфхцчшщьюя")
```

```
    russian_letters = set("абвгдеёжзийклмнопрстуфхцчшщъыьэюя")
```

```
    english_letters = set("abcdefghijklmnopqrstuvwxyz")
```

```
    cleaned = re.sub(r'[^a-zA-Za-яА-ЯёЁиїїєЄрї]', "", text.lower())
```

```
    counts = {
```

```
        "uk": sum(c in ukrainian_letters for c in cleaned),
```

```
        "ru": sum(c in russian_letters for c in cleaned),
```

```
        "en": sum(c in english_letters for c in cleaned)
```

```
    }
```

```
    return max(counts, key=counts.get)
```

```
# === Заміна ідіом
```

```
def replace_idioms_with_ukr(text):
```

```
    for idiom, ukr_eq in idioms_en_to_uk.items():
```

```
        pattern = re.compile(rf"\b{re.escape(idiom)}\b", re.IGNORECASE)
```

```
        text = pattern.sub(ukr_eq, text)
```

```
    return text
```

```
def replace_idioms_with_en(text):
```

```
    for ukr_eq, idiom in idioms_uk_to_en.items():
```

```
        pattern = re.compile(rf"\b{re.escape(ukr_eq)}\b", re.IGNORECASE)
```

```
        text = pattern.sub(idiom, text)
```

```
    return text
```

```

# === Переклад
def translate_en_to_uk(text):
    text_clean = text.strip().lower()
    if text_clean in known_sentences:
        return known_sentences[text_clean]

    replaced = replace_idioms_with_ukr(text)
    inputs = tokenizer_en_to_uk(replaced, return_tensors="pt", padding=True,
truncation=True)
    translated_tokens = marian_en_to_uk.generate(**inputs)
    translation = tokenizer_en_to_uk.decode(translated_tokens[0],
skip_special_tokens=True)

    if detect_language(translation) == "ru":
        return " ⚠ Модель переклала не українською. Спробуйте змінити
формулювання або подати коротше речення."
    return translation

def translate_uk_to_en(text):
    if not text.strip():
        return ""
    replaced = replace_idioms_with_en(text)
    result = translator_ukrainian_to_english(replaced, max_length=1500)
    return result[0]['translation_text']

# === Flask routes ===
@app.route('/')
def index():
    return render_template('Darkib.html')

@app.route('/translate', methods=['POST'])
def translate():
    input_text = request.form['input_text']
    direction = request.form['direction']

    if direction == 'en_to_uk':
        output_text = translate_en_to_uk(input_text)
    else:
        output_text = translate_uk_to_en(input_text)

    return render_template('Darkib.html', input_text=input_text,

```

```

output_text=output_text, direction=direction)

# === Server run logic ===
if __name__ == '__main__':
    logging.basicConfig(level=logging.INFO)

    def find_free_port(start_port=5001, max_tries=10):
        port = start_port
        for _ in range(max_tries):
            with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
                if s.connect_ex(('127.0.0.1', port)) != 0:
                    return port
            port += 1
        raise RuntimeError("No free port found in range.")

    try:
        chosen_port = find_free_port()
        logging.info(f'Flask app running on port {chosen_port}')
        app.run(host='0.0.0.0', port=chosen_port, debug=False)
    except Exception as e:
        logging.critical(f'Flask failed to start: {e}')

```

Фронтенд частина додатку

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Translation App</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f6f6f6;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .container {
      background: white;
      padding: 30px;
      border-radius: 12px;
      box-shadow: 0 0 10px rgba(0,0,0,0.1);
      width: 400px;
      text-align: center;
    }
    textarea {
      width: 100%;
      height: 100px;
      padding: 10px;
      margin-top: 10px;
      border-radius: 8px;
```

```

        border: 1px solid #ccc;
        resize: none;
    }
    select, button {
        margin-top: 15px;
        padding: 10px;
        width: 100%;
        border-radius: 8px;
        border: 1px solid #ccc;
    }
    button {
        background-color: #4CAF50;
        color: white;
        border: none;
        cursor: pointer;
    }
    .output {
        margin-top: 20px;
        background-color: #f0f0f0;
        padding: 15px;
        border-radius: 8px;
        text-align: left;
    }
</style>
</head>
<body>
    <div class="container">
        <h2>Translation App</h2>
        <form action="/translate" method="post">
            <label for="input_text"><strong>Input Text:</strong></label><br>

```

```
<textarea id="input_text" name="input_text">{{ input_text or "
}}</textarea><br>
```

```
<label for="direction"><strong>Select Direction:</strong></label><br>
```

```
<select id="direction" name="direction">
```

```
<option value="en_to_uk" {% if direction == 'en_to_uk' %}selected{%
endif %}>English to Ukrainian</option>
```

```
<option value="uk_to_en" {% if direction == 'uk_to_en' %}selected{%
endif %}>Ukrainian to English</option>
```

```
</select><br>
```

```
<button type="submit">Translate</button>
```

```
</form>
```

```
{% if output_text %}
```

```
<div class="output">
```

```
<strong>Translated Text:</strong>
```

```
<p>{{ output_text }}</p>
```

```
</div>
```

```
{% endif %}
```

```
</div>
```

```
</body>
```

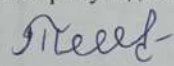
```
</html>
```

ДОДАТОК В (ОБОВ'ЯЗКОВИЙ) ІЛЮСТРАТИВНА ЧАСТИНА

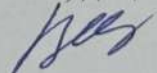
ІЛЮСТРАТИВНА ЧАСТИНА

«РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ДЛЯ АВТОМАТИЧНОГО ПЕРЕКЛАДУ ТЕКСТІВ МІЖ МОВАМИ»

Виконав: студент 4 курсу, групи 2КН-216
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

 Лучик Т.С.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

 Денисюк В.О.
(прізвище та ініціали)

«03» червня 2025р.

Вінниця ВНТУ – 2025 рік



Рисунок В.1 – Схема алгоритму роботи інтелектуального модуля для автоматичного перекладу текстів між мовами



Рисунок В.2 – Загальна структурна схема функціонування інтелектуального модуля перекладу між мовами



Рисунок В.3 – Схема життєвого циклу інтелектуального модулю перекладу між мовами

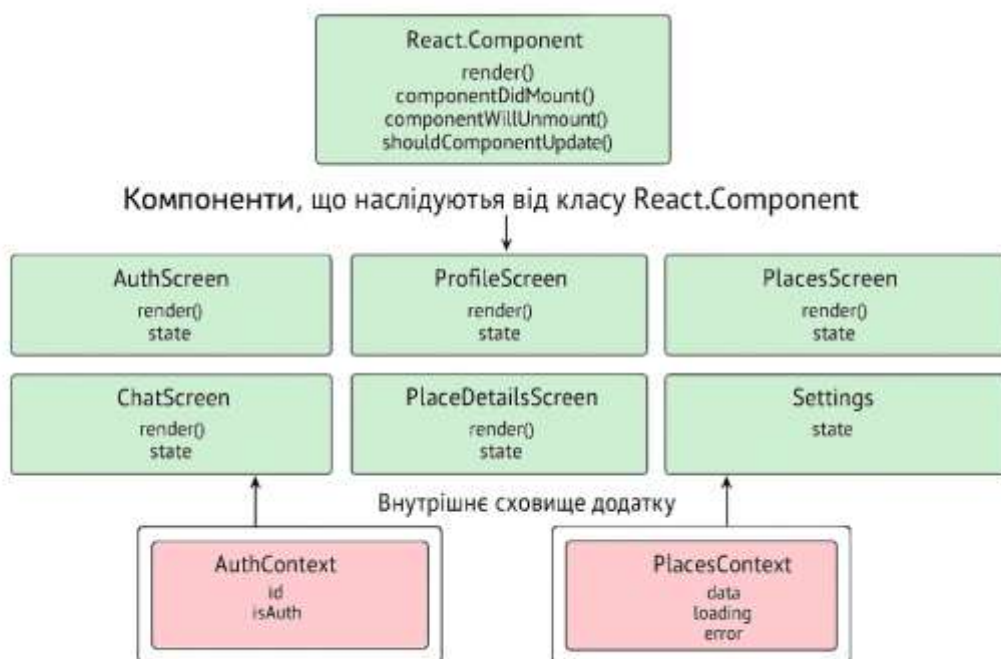


Рисунок В.4 – UML-Діаграма класів інтелектуального модуля перекладу між мовами

Діаграма прецедентів інтелектуального модуля перекладу

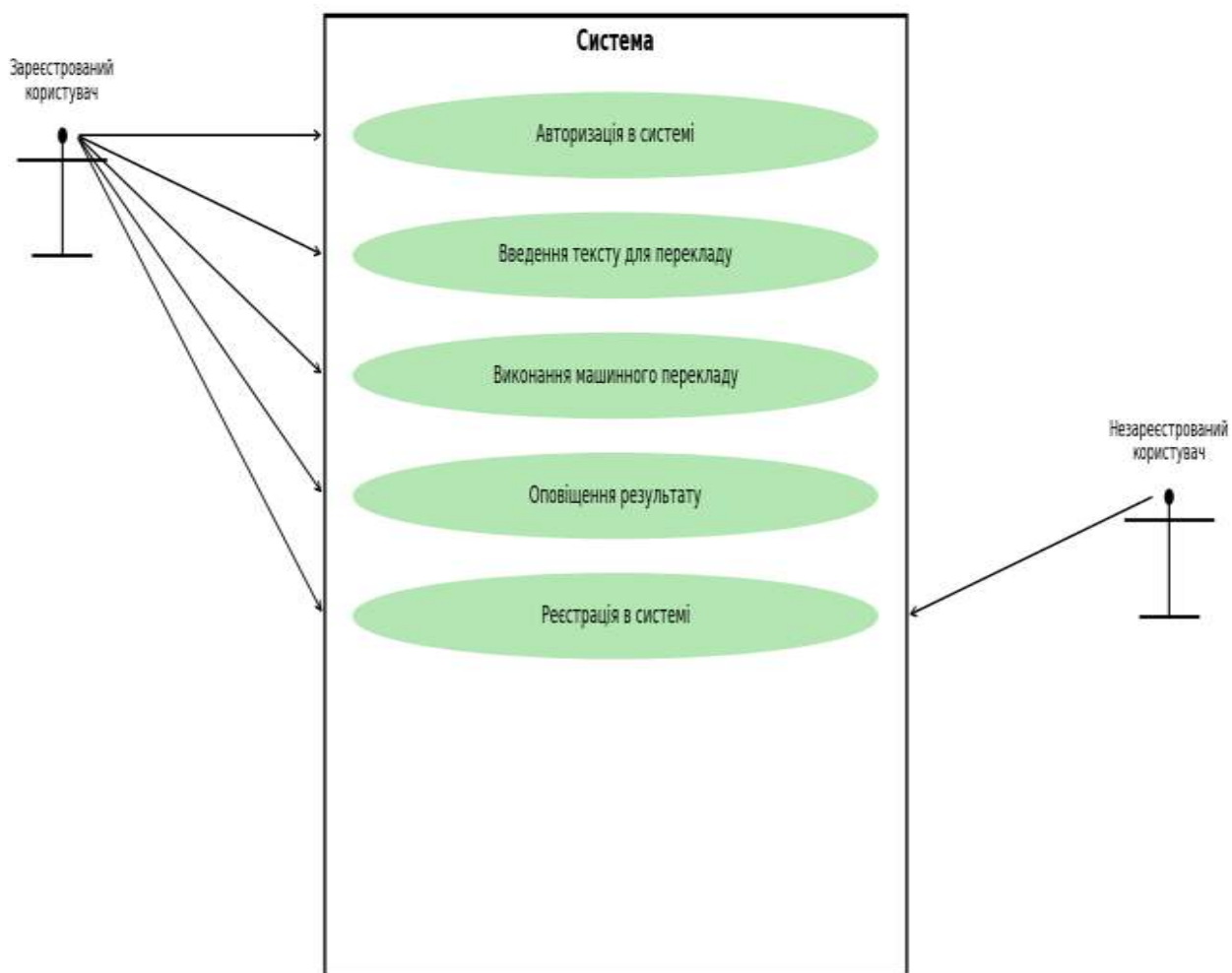


Рисунок В.5 – Діаграма прецедентів інтелектуального модуля автоматичного перекладу текстів між мовами

ДОДАТОК Г (ДОВІДНИКОВИЙ) ІНСТРУКЦІЯ КОРИСТУВАЧА

1. Покрокова інструкція

Відкрийте вебсторінку програми.

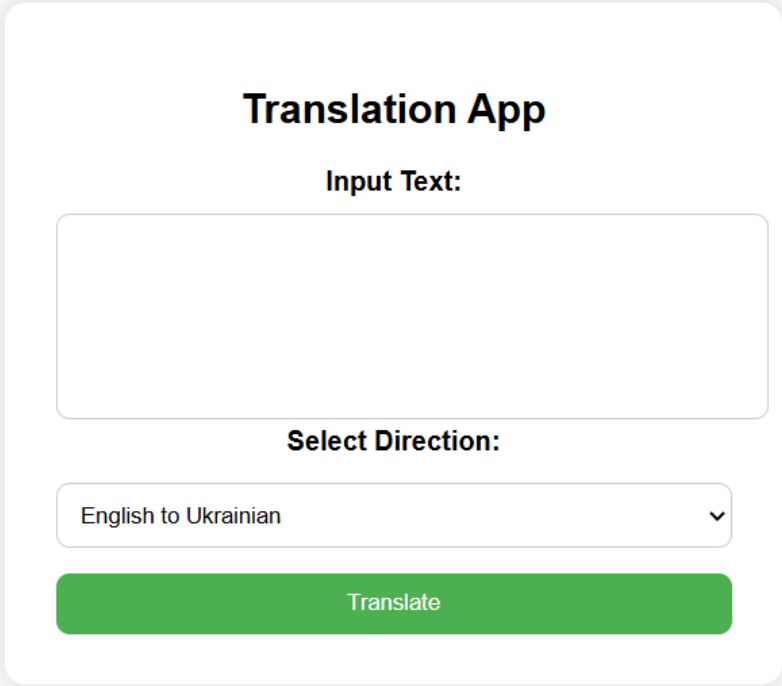
У полі “Input Text” введіть речення для перекладу.

У випадаючому списку “Select Direction” виберіть напрям перекладу, наприклад, з англійської на українську або з української на англійську.

За потреби змініть мову перекладу, повторно відкривши випадаючий список і обравши інший мовний напрям.

Натисніть кнопку “Translate”.

Переклад з’явиться внизу, у полі “Translated Text”.



The screenshot displays the main interface of the 'Translation App'. At the top, the title 'Translation App' is centered. Below it, the label 'Input Text:' is positioned above a large, empty rectangular text input field. Underneath the input field, the label 'Select Direction:' is centered above a dropdown menu. The dropdown menu currently shows 'English to Ukrainian' with a downward arrow on the right. At the bottom of the interface is a prominent green button with the text 'Translate' in white.

Рисунок Г.1 – Основна панель інтерфейсу програми інтелектуального модуля для перекладу між мовами