

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук


КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА


на тему Спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі


Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань


Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності

Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 4 курсу, група КН-19-1  В.С. Одуха
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: д.т.н., доцент кафедри КН  Е.А. Манзюк
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КН  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор  О.В. Бармак
Підпис Ініціали, прізвище

01 06 2023 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

Освітня програма освітньо-професійна програма підготовки бакалавра

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук



(підпис)

д.т.н., професор О.В. Бармак

« 06 » 03 2023 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі»

2. Завдання видано студенту Одусі Владіславу Станіславовичу
(прізвище, ім'я, по батькові)

3. Керівник роботи доцент кафедри КН Манзюк Едуард Андрійович
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від « 01 » 03 2023 р. № 5

5. Дата видачі завдання студенту: « 03 » 03 2023 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

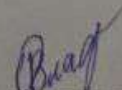
Провести аналіз предметної області автоматизованого перекладу текстів, проаналізувати особливості визначення якості перекладу. Розглянути методи визначення якості автоматизованого перекладу з використанням нейронної мережі. Дослідити існуючі рішення щодо подібних задач. Розробити й програмно реалізувати прикладний застосунок для визначення якості автоматизованого перекладу з використанням нейронної мережі. Виконати дослідження ефективності розробленого способу визначення якості автоматизованого перекладу з використанням нейронної мережі.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником	грудень 2022	виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2023	виконано
3	Робота над розділом 1 – Характеристика предметної області та постановка задачі	січень 2023	виконано
4	Робота над розділом 2 – Спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі	березень 2023	виконано
5	Робота над розділом 3 – Програмна реалізація способу визначення якості автоматизованого перекладу з використанням нейронної мережі	квітень 2023	виконано
6	Оформлення пояснювальної записки згідно вимог	травень 2023	виконано
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2023	виконано
8	Захист кваліфікаційної роботи бакалавра на засіданні Екзаменаційної комісії	червень 2023	виконано

Виконавець:

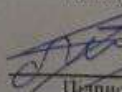
студент 4 курсу, група КН-19-1
Курс, група виконавця


Підпис

В.С. Одуха
Ініціали, прізвище

Керівник:

д.т.н., доцент кафедри КН
Науковий ступінь, посада


Підпис

Е.А. Манзюк
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: Спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-19-1 Одуха Владіслав Станіславович

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН Манзюк Едуард Вікторович

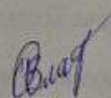
Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
77	28	10	27	2

Кваліфікаційна робота бакалавра присвячена розробці способу визначення якості автоматизованого перекладу з використанням нейронної мережі та відповідної інформаційної системи у вигляді застосунку, що дозволяє за текстовим вмістом автоматизованого перекладу визначати відносну числову оцінку його якості.

Ключові слова: автоматизований переклад, нейронна мережа, якість автоматизованого перекладу.

Виконавець: студент 4 курсу, група КН-19-1
Курс, група виконавця


Підпис

В.С. Одуха
Ініціали, прізвище

Зміст

Перелік скорочень	3
Вступ.....	4
Розділ 1 Характеристика предметної області і постановка задачі.....	6
1.1 Аналіз предметної області автоматизованого перекладу та необхідності визначення його якості	6
1.2 Огляд нейронних мереж для задач обробки природної мови	13
1.3 Аналіз існуючих рішень для подібних задач	15
1.4 Мета, задачі та вимоги до реалізації програмної системи.....	19
Розділ 2 Технологія визначення якості автоматизованого перекладу з використанням нейронної мережі	20
2.1 Опис і кроки технології визначення якості автоматизованого перекладу з використанням нейронної мережі	20
2.2 Спосіб визначення якості автоматизованого перекладу.....	24
2.3 Підготовка зразків робочих вхідних даних для інформаційної системи	28
2.4 Якість визначення перекладу за допомогою нейронної мережі BERT	31
2.5 Проектування структури застосунку для визначення якості автоматизованого перекладу.....	34
2.6 Даталогічна модель бази даних застосунку для визначення якості автоматизованого перекладу.....	39
2.7 Висновки до розділу 2	42
Розділ 3 Програмна реалізація застосунку для визначення якості автоматизованого перекладу з використанням нейронної мережі	44
3.1 Структура модулів застосунку для визначення якості автоматизованого перекладу	44
3.2 Особливості реалізації застосунку для визначення якості автоматизованого перекладу	46
3.3 Опис функціональних можливостей застосунку для визначення якості автоматизованого перекладу.....	53
3.4 Висновки до розділу 3	60
Висновки	62
Перелік посилань.....	64
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations Transformers
GPT	Generative Pre-trained Transformer
AI	Artificial intelligence
QA	Quality Assurance
ПЗ	Програмне забезпечення
НМ	Нейронна мережі
БД	База даних
СКБД	Система керування базами даних
ID	Identity Document
FK	Foreign Key
SST	Stanford Sentiment Treebank
QQP	Quora Question Pairs
MNLI	Multi-Genre Natural Language Inference

Вступ

Кваліфікаційна робота бакалавра присвячена розробці способу визначення якості автоматизованого перекладу з використанням нейронної мережі та відповідної інформаційної системи у вигляді застосунку, що дозволяє за текстовим вмістом автоматизованого перекладу визначати відносну числову оцінку його якості.

Актуальність – питання якості перекладу залишається актуальним, оскільки автоматичний переклад часто містить помилки, неправильно передає відтінки смислу та інші неточності. Визначення якості перекладу є важливим завданням для того, щоб забезпечити його використання в реальних ситуаціях.

Об'єкт дослідження – процес визначення якості автоматизованого перекладу текстів.

Предмет дослідження – моделі, методи, алгоритми та засоби для визначення якості автоматизованого перекладу текстів.

Мета кваліфікаційної роботи бакалавра – розробка й програмна реалізація способу визначення якості автоматизованого перекладу з використанням нейронної мережі.

Завдання кваліфікаційної роботи бакалавра – Провести аналіз предметної області визначення якості автоматизованого перекладу текстів. Виконати аналіз методів визначення якості автоматизованого перекладу з використанням нейронної мережі. Визначити особливості використання нейронних мереж при визначенні якості автоматизованого перекладу тексту. Виконати аналіз існуючих рішень щодо подібних задач. Розробити спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі. Спроекувати структуру прикладного застосунку для визначення якості автоматизованого перекладу з використанням нейронної мережі. Спроекувати структуру бази даних застосунку для визначення якості автоматизованого перекладу з використанням нейронної мережі. Виконати вибір засобів розробки застосунку для визначення якості автоматизованого перекладу з використанням

нейронної мережі. Виконати програмну реалізацію застосунку для визначення якості автоматизованого перекладу. Провести тестування розробленого застосунку.

Розділ 1 Характеристика предметної області і постановка задачі

1.1 Аналіз предметної області автоматизованого перекладу та необхідності визначення його якості

Для реалізації способу визначення якості автоматизованого перекладу тексту необхідно провести детальний аналіз предметної області, зокрема, розглянути особливості автоматизованого перекладу та сучасні методи визначення його якості.

За останні десятиріччя сфера інформаційних технологій набула стрімкого розвитку по всьому світу. Щосекунди відбувається обмін величезною кількістю інформації між безліччю користувачів всесвітньої мережевої павутини. Будь-яка загальнодоступна інформація може бути переглянута з будь-якої точки світу, завдяки використанню інформаційних технологій. Однією з найпоширеніших проблем при передачі інформації між користувачами може бути нерозуміння мови носія інформації та її отримувача. Адже, за даними однієї з найбільш авторитетних баз даних «Ethnologue», стосовно кількості мов, у світі налічується більше 7 тисяч мов [1]. Таким чином, споживачі інформації можуть просто не зрозуміти її, через причину незнання мови.

Велике різноманіття мов, що ускладнювало передачу інформації, призвело до виникнення систем автоматизованого та машинного перекладу, тобто систем, які використовують інформаційні технології для створення перекладу текстів. Це допомогло вирішити проблему відмінності мов між користувачами всесвітнього інформаційного простору та значно полегшило передачу інформації у будь-яку точку світу.

На перший погляд, може здатись, що автоматизований та машинний переклад означає один і той самий процес. Проте, ці поняття суттєво відрізняються одне від одного.

Під поняттям машинного перекладу розуміється підвид автоматизованого перекладу, під час якого використовуються різні типи програмного забезпечення на основі алгоритмів машинного навчання та методів штучного інтелекту [2].

Автоматизований переклад це більш широке поняття, яке включає в себе будь-який процес перекладу тексту з однієї мови на іншу, використовуючи інформаційні технології [3]. Крім машинного перекладу, автоматизований переклад включає в себе також переклад на основі правил та глибинного навчання. Таким чином, під поняттям автоматизованого перекладу можна розміти будь-який переклад створений з використанням спеціалізованого програмного забезпечення.

Отже, можна виділити такі основні типи автоматизованого перекладу, які відрізняються методами, що застосовуються для перекладу текстів [4]:

- Машинний переклад. Використовує машинне навчання та статистичні моделі.
- Переклад на основі правил. Використовує граматичні правила, словникові данні та лінгвістичні особливості мов для яких здійснюється переклад.
- Глибинне навчання. Використовує методи та системи штучного інтелекту для розуміння тексту машиною та здійснення його перекладу.
- Переклад з використанням пам'яті. Використовує раніше перекладенні тексти, та здійснює переклад на їхній основі.
- Гібридний переклад. Використовує поєднання різних методів перекладу, від машинного навчання до перекладу на основі правил.

Таким чином, основною особливістю автоматизованого перекладу, та фактором, який пов'язує різні типи автоматизованого перекладу є використання інформаційних технологій. Це значна перевага над ручним перекладом, адже економить час та розумові ресурси перекладача, необхідні для створення якісного перекладу. Зважаючи на це, можна виділити основні переваги автоматизованого перекладу [5]:

- Швидкість. Автоматизований переклад тексту створюється набагато швидше за ручний. У першому випадку текст певний текст може бути перекладений за секунди, тоді як перекладач витратить на переклад такого ж тексту хвилини, а той години часу.

- Економія коштів. Зазвичай автоматизований переклад коштує значно дешевше ніж послуги професійного перекладача.
- Масштабованість. Автоматизований переклад може перекладати великі за розміром тексти швидше ніж перекладач із безліччю обмежень в ресурсах.
- Надійність. Автоматизований переклад може бути більш надійним та якісним у випадку, коли перекладач погано обізнаний із термінологією тексту, який потрібно перекласти.

Автоматизований переклад тексту це досить складний процес який складається з декількох послідовних етапів (Схема 1.1) [6]. Для початку роботи необхідно обрати текст, який потрібно перекласти та визначити мову цього тексту та мову на яку необхідно перекласти. Тобто, оригінальний текст буде виступати в ролі вхідних даних для процесу автоматизованого перекладу.

На першому кроці необхідно проаналізувати вхідний текст. Це означає, що програма аналізує граматичні конструкції, які є в тексті. Також, до уваги беруться лексичні та семантичні особливості тексту. Не варто й забувати про контекст та сенс вхідного тексту.

На другому Після аналізу тексту відбувається пошук відповідності термінів початкової мови тексту до термінів у мові на яку необхідно цей текст перекласти. Цей крок можна вважати початковим кроком формування повноцінного перекладу тексту.

На третьому кроці відбувається формування структури речень. Цей крок неможливий без двох попередніх, адже аналіз тексту допомагає визначити структуру речень початкового тексту, а пошук відповідностей відіграє важливу роль в формуванні структури перекладеного тексту.



Рисунок 1.1 – Функціональна схема автоматизованого перекладу тексту [6]

Коли структура речень сформована програма створює переклад тексту. Цей переклад можна вважати чорновим адже, останнім кроком при автоматизованому перекладі текстів є перевірка перекладу. На останньому кроці переклад проходить детальну перевірку на виявлення помилок. У випадку виявлення помилки текст корегується. Даний крок повторюється доти, доки в тексті не буде виявлено жодної помилки.

Таким чином, по завершенню автоматизованого перекладу, користувач отримує готовий переклад тексту на бажану мову. Переклад тексту повинен відбуватись в строгій послідовності кроків, адже зміна порядку дій може призвести до погіршення перекладу.

Автоматизований переклад це дуже зручний та швидкий спосіб для перекладу текстів. Проте, при такому виді перекладу може виникнути ряд проблем, які в подальшому можуть вплинути на якість створеного перекладу. Найпоширенішими проблемами під час автоматизованого перекладу є [7]:

– Полісемія (багатозначність). Тобто, це здатність одного слова мати різне значення в залежності від контексту в якому це слово використане [8]. Таким чином, це може призвести до неправильного перекладу слів та зміни оригінального сенсу тексту.

– Граматика. Під час автоматизованого перекладу, програма може використати неправильну форму слова, що призведе до виникнення граматичних помилок у перекладі.

– Контекст. Під час автоматизованого перекладу система не завжди здатна правильно зрозуміти контекст та сенс певного речення з тексту. Це може призвести до часткової або повної втрати сенсу оригінального тексту.

– Мовний колорит. Кожна мова має свої культурні особливості, які можуть впливати на сенс тексту. Особливо це стосується різного роду ідіом, афоризмів, прислів'їв та народних висловів, які суттєво відрізняються в різних народах.

– Стилiстичні особливості. Кожен написаний текст має свої стилістичні особливості в залежності від автора. Система автоматизованого перекладу не здатна з точністю відтворити стилістику автора оригінального тексту.

– Обмежена мовна підтримка. Деякі непоширені мови можуть мати меншу мовну підтримку, що може призвести до погіршення якості перекладу.

Незважаючи на те що ці проблеми можуть значно зменшити якість перекладу, їх можна легко вирішити поєднавши автоматизований та ручний переклади.

Автоматизований переклад текстової інформації набув широкого використання у багатьох сферах життя людини (Рисунок 1.2). Адже, це допомагає розуміти людей з будь-якої країни не знаючи їхньої мови.

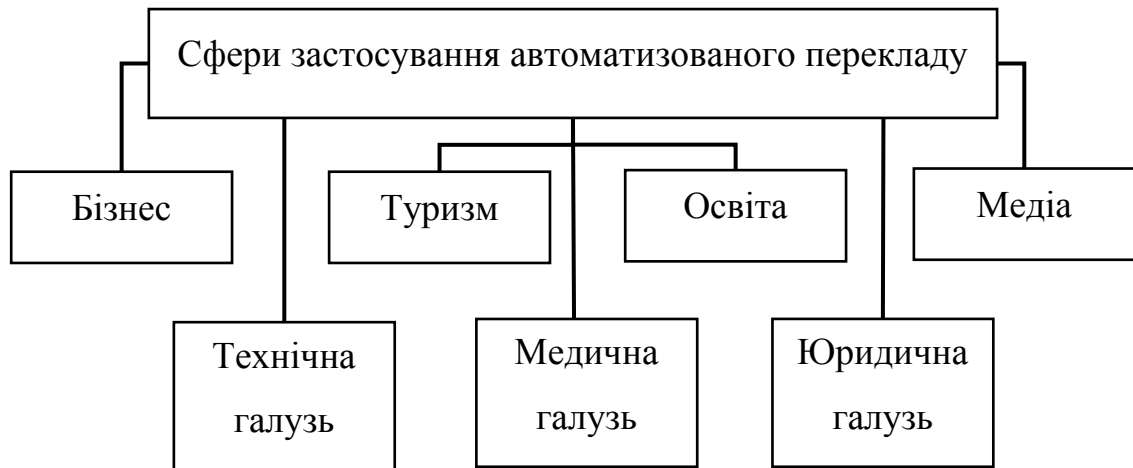


Рисунок 1.2 – Сфери застосування автоматизованого перекладу

У сфері бізнесу автоматизований переклад відіграє дуже важливу роль під час спілкування компаній зі своїми міжнародними партнерами та закордонними клієнтами. Також, це допомагає проводити активну рекламну компанію не тільки на ринку своєї країни, а й поза її межами. Це в свою чергу збільшує прибутки компанії та підвищує значущість автоматизованого перекладу у цій сфері.

Сервіси автоматизованого перекладу можуть бути дуже корисними для туристів, які подорожують країнами не знаючи їхніх мов. Це полегшує взаємодію туристів із місцевим населенням при пошуку житла, покупці товарів чи просто ознайомленні з місцевими звичаями та традиціями.

Однією з найпоширеніших сфер застосування автоматизованого перекладу є сфера освіти. Це допомагає обмінюватись досвідом та знаннями студентам та викладачам з різних країн. Також, це допомагає при вивченні дисциплін, адже, багато корисної інформації написана зарубіжними авторами.

В юридичній сфері та інших суміжних сферах пов'язаних із законом автоматизований переклад може знадобитись при роботі з документами написаними на інших мовах. Крім цього, засоби автоматизованого перекладу використовуються при веденні міжнародної політики між державами.

В медичній сфері засоби автоматизованого перекладу можуть використовуватись для спілкування між пацієнтами та лікарями, у випадку коли пацієнти лікуються закордоном. Також, аналогічно при навчанні лікарів, безліч літератури може бути написана на інших мовах, що призводить до необхідності використання автоматизованого перекладу.

Щодо технічної галузі, то в цьому випадку автоматизований переклад може бути корисним при роботі з інструкціями та документаціями написаними іншими мовами. Також, багато сучасних інженерних програм здебільшого мають англійський інтерфейс, тому для інженерів, які не знають англійську, може знадобитись допомога перекладача.

Використання автоматизованого перекладу в медіа просторі зумовлене тим, що багато репортажів та статей можуть бути створені на незнайомій людам мові. Тому, щоб дізнатись потрібну інформацію з декількох джерел варто скористатись послугами сервісів автоматизованого перекладу.

В повсякденному житті також не обійтись без автоматизованого перекладу, особливо людям які активно користуються інтернет послугами. Існування систем автоматизованого перекладу звільняє людей від необхідності детально вивчати безліч мов, щоб розуміти інформацію від іншомовного джерела. Проте, наскільки б хорошою та продуманою не була б система автоматизованого перекладу, вона все одно може робити помилки, які впливатимуть на якість перекладу. Таким чином, щоб зменшити кількість неякісного перекладу, необхідно перевіряти переклад на якість.

В широкому понятті, якість – це набір властивостей об'єкта, які визначають рівень придатності його використання за призначенням [9]. Стосовно сфери перекладів, то якістю перекладу є сукупність норм та правил, яким повинен відповідати переклад.

Якість автоматизованого перекладу може залежати від багатьох факторів, серед яких можна виділити основні [10]:

- дотримання змісту оригінального тексту;
- дотримання послідовної оповіді тексту;

- врахування культурних особливостей;
- відповідність термінів;
- правильна адаптація власних назв та імен;
- граматично правильна побудова речень.

Кожна критерія по своєму впливає на якість перекладу, і лише врахування їх усіх дасть дійсно якісний результат.

1.2 Огляд нейронних мереж для задач обробки природної мови

Більшість інформації, яка зберігається в інформаційному просторі, існує у формі цифрових текстів написаних людиною. Для того щоб ефективно працювати та аналізувати таку інформацію необхідно навчити комп'ютер розуміти мову людини. Для комп'ютера така мова називається природною мовою, тобто мовою створеною людьми без використання жодного програмного забезпечення [11].

Процесом обробки природної мови (Natural Language Processing – NLP) займається окрема галузь комп'ютерних наук – комп'ютерна лінгвістика [12]. Основним завданням цієї галузі є розробка методів та алгоритмів, які допомагають аналізувати природну мову.

NLP вирішує безліч завдань, пов'язаних з обробкою природної мови. Основними серед них є [13]:

- Класифікація. Розподіляє тексти на певні класи відповідно до контексту або інших параметрів.
- Машинний переклад. Створення перекладу текстів із однієї мови на іншу.
- Управління голосом. Перетворює звукові дані на текст або навпаки.
- Релевантний пошук. Здійснює пошук потрібної інформації у великих масивах даних.
- Аналіз тональності. Розпізнає настрої та емоції, які є в тексті.

- Перевірка грамотності тексту. Перевіряє текст на наявність граматичних, орфографічних, лексичних та інших помилок.

- Генерація тексту. Генерує текст відповідно до теми та інших потреб.

Методи обробки природної мови широко використовуються в різних галузях та сферах життя людини, таких як медицина, освіта, бізнес та фінансова справа, туризм, і звичайно ж, повсякденне життя.

У процесі аналізу природної мови системи NLP використовують безліч алгоритмів, від статистичних моделей до машинного навчання. Використання нейронних мереж широко використовується для вирішення задач NLP. Основною перевагою використання нейронних мереж в даній галузі є їх здатність до постійного навчання. Це дозволяє зробити інструмент обробки та аналізу природної мови більш точним.

Найпоширенішими сучасними нейронними мережами, які вирішують завдання розуміння природної мови, вважаються:

- BERT;
- GPT;
- BARD.

Нейронна мережа BERT (Bidirectional Encoder Representations from Transformers) використовує у своїх алгоритмах двонаправлене навчання, що дозволяє моделі краще розуміти контекст ніж при однонаправленому навчанні. Нейронна мережа вивчає контекстні зв'язки між словами в тексті. Модель аналізує весь текст одночасно, невідмінну від інших нейронних мереж, які роблять це з права на ліво або навпаки. Це дозволяє нейронній мережі BERT робити висновки про контекст слова на основі інших слів, які оточують його з усіх боків [14].

GPT (Generative Pre-trained Transformer) – це нейронні мережі, розроблені американською компанією OpenAI. Дана нейронна мережа здатна аналізувати запити користувачів та відповідати на них згенерованим текстом, який дуже подібний до людського. GPT-3 вважають однією з найкращих систем для обробки природної мови та генерування змістовних і зрозумілих текстів,

відповідно до заданої теми. Компанія OpenAI проводить постійні дослідження в галузі штучного інтелекту, та в березні 2023 року випустила нове покоління нейронних мереж GPT-4, які можуть аналізувати не тільки текстові запити, а й зображення [15]. Нейронні мережі GPT, однозначно можна вважати проривом в галузі обробки та розуміння природної мови. Проте, є дуже вагомий недолік при використанні нейронних мереж GPT пов'язаний із їхнім навчанням. Нейронні мережі навчені на інформації, яка була розміщена в мережі «Інтернет» до 2021 року. Усе що було створено пізніше, нейронним мережам цього покоління – недоступне та невідоме. Тому, деяка інформація, яку надає мережа може бути застарілою та неактуальною.

Нейронна мережа BARD відноситься до генеративних нейронних мереж, тобто до тих, що можуть створювати новий контент. Основним завданням BARD є генерація текстового контенту відповідно до запиту користувача. Нейронні мережі BARD та GPT є дуже подібними. Проте, завдяки інтеграції BARD із пошуковою системою Google Search, дана нейронна мережа отримує усю актуальну інформацію в режимі реального часу [16].

Для розробки способу визначення якості автоматизованого перекладу було обрано використовувати нейронну мережу BERT, адже її можна легко інтегрувати у власний проєкт завантаживши необхідний код. Ще однією перевагою використання нейронної мережі BERT є те, що вона не потребує навчання, адже нейронна мережа вже навчена. Достатньо лише налаштувати нейронну мережу під своє конкретне завдання.

1.3 Аналіз існуючих рішень для подібних задач

Системи автоматизованого перекладу значно полегшили комунікацію між людьми з різних частин світу. Використання такого програмного забезпечення звільняє людей від необхідності вивчення безлічі мов на високому рівні, який необхідний для вільного спілкування. Проте, машинний переклад не завжди дає точні та правильні результати. При використанні машинного перекладу можуть

виникнути проблеми пов'язані з граматикою та іншими мовними аспектами. Тому важливо перевіряти згенерований програмою переклад на наявність помилок.

Для визначення якості перекладу можна використовувати сервіси, які аналізують текст на наявність помилок та надають підказки, як їх виправити. Одним із таких сервісів є популярний вебресурс «Grammarly», заснований українцями у 2009 році [17]. Основною метою сервісу є перевірка англomовних текстів на помилки та надання варіантів їх виправлення (Рисунок 1.3). Оскільки, українці найчастіше перекладають тексти саме на англійську мову, то даний сервіс є дуже популярним та корисним. «Grammarly» відмінно підходить для завдання перевірки якості автоматизованого перекладу.

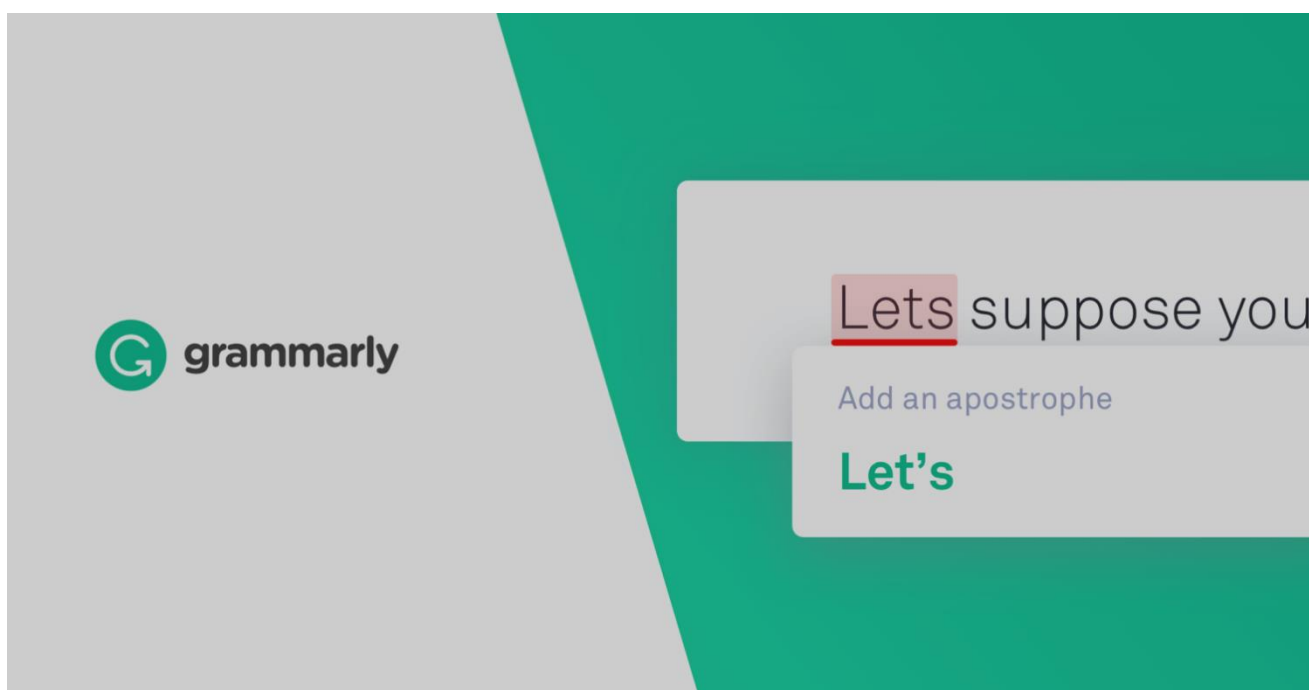


Рисунок 1.3 – Пошук та виправлення помилок сервісом «Grammarly» [17]

«Grammarly» дуже добре справляється з пошуком помилок в тексті. Також, функція, яка надає варіанти виправлення помилок є дуже зручною, особливо для людей, які погано знають англійську мову. Єдиним недоліком розглянутого сервісу можна назвати те, що є можливість перевіряти тексти лише англійською мовою.

Поширеною проблемою під час використання систем автоматизованого перекладу є невідповідність термінології оригінального тексту та згенерованого перекладу. Ця проблема може призвести до виникнення безлічі проблем при використанні неякісного перекладу. Тому, варто детально перевіряти термінологію у вхідному тексті та його перекладі. Для вирішення цієї проблеми існує сервіс «Xbench» [18]. Для використання сервісу достатньо завантажити оригінальний текст та його переклад в віконний додаток. Далі, завдяки великій базі термінології, сервіс перевірить відповідність термінів в обох документах та покаже результат користувачеві (Рисунок 1.4).

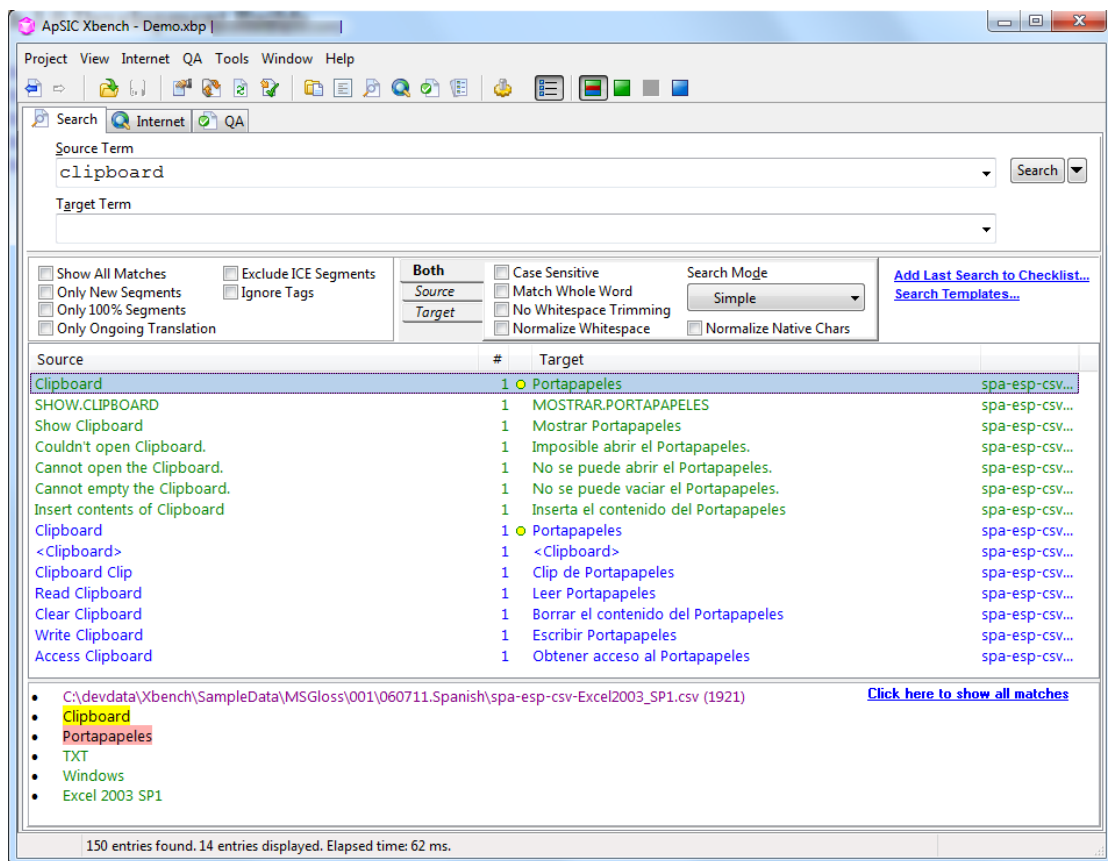


Рисунок 1.4 – Використання сервісу «Xbench» [18]

Недоліком сервісу «Xbench» є його трохи складний інтерфейс та необхідність завантаження додатку на ПК. Щодо пошуку помилок в термінології, сервіс справляється відмінно.

Ще одним поширеним сервісом для визначення якості перекладу є сервіс «QA Distiller» [19]. Даний сервіс широко використовується в спільноті

перекладачів для пошуку помилок в двомовних файлах. Перевагою даного сервісу є перевірка саме двомовних файлів, тобто оригіналу та перекладу. Це означає що програма враховує не лише граматичні аспекти перекладу, а й контекст та зміст оригінального тексту.

«QA Distiller» призначений для порівняння двох двомовних файлів та пошуку помилок у них. По завершенню порівняння програма надає звіт про помилки в окремому файлі, що дуже зручно для перегляду та пошуку помилок (Рисунок 1.5).

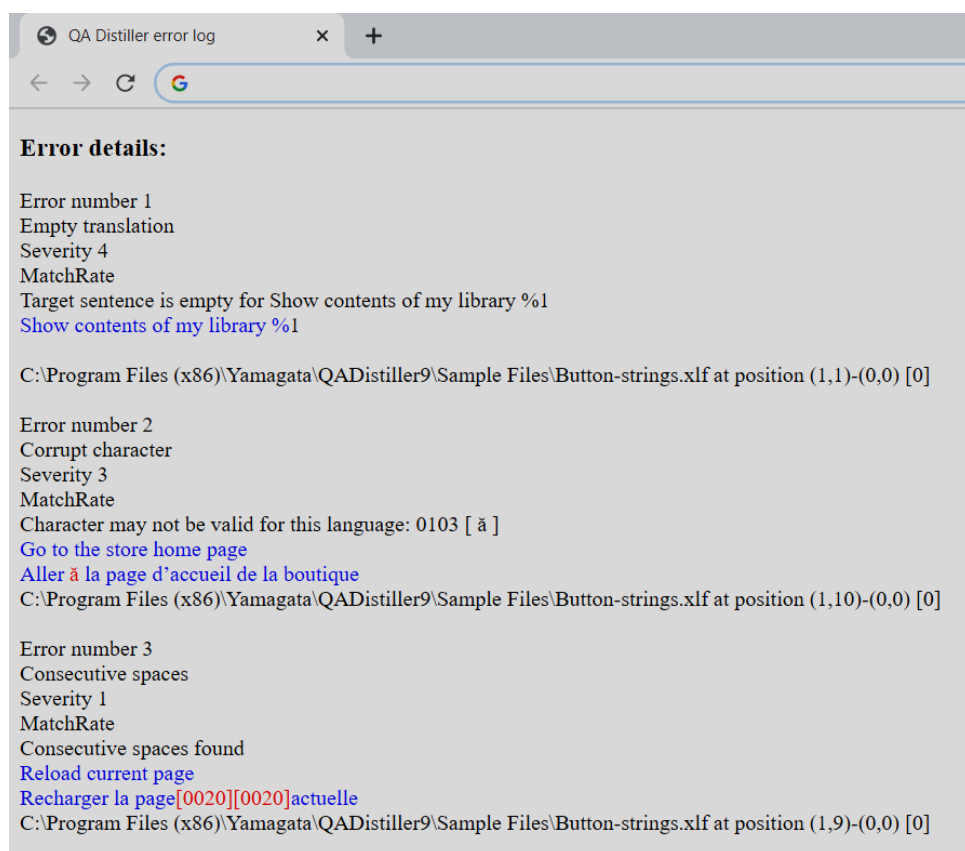


Рисунок 1.4 – Звіт про помилки наданий сервісом «QA Distiller» [19]

Існує ще багато сервісів для перевірки текстів на помилки та визначення якості автоматизованих перекладів, що доводить про актуальність та необхідність дослідження даної предметної області. Отже, розробка додатку для визначення якості автоматизованого перекладу з використанням нейронної мережі є актуальним завданням.

1.4 Мета, задачі та вимоги до реалізації програмної системи

Метою кваліфікаційної роботи бакалавра є розробка й програмна реалізація способу визначення якості автоматизованого перекладу з використанням нейронної мережі.

Для досягнення мети необхідно вирішити такі задачі:

- розробити спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі.
- спроектувати структуру прикладного застосунку для визначення якості автоматизованого перекладу з використанням нейронної мережі.
- спроектувати структуру бази даних застосунку для визначення якості автоматизованого перекладу з використанням нейронної мережі.
- виконати вибір засобів розробки застосунку для визначення якості автоматизованого перекладу з використанням нейронної мережі.
- виконати програмну реалізацію застосунку для визначення якості автоматизованого перекладу з використанням нейронної мережі.
- провести тестування розробленого застосунку для визначення якості автоматизованого перекладу з використанням нейронної мережі.

Результатом виконання поставлених задач повинен бути застосунок з можливістю визначення якості автоматизованого перекладу тексту з використанням нейронної мережі.

Розділ 2 Спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі

2.1 Опис і кроки способу визначення якості автоматизованого перекладу з використанням нейронної мережі

Для визначення якісного автоматизованого перекладу було прийнято рішення використовувати нейронну мережу BERT. Обрана нейронна мережа показує дуже хороші результати у вирішенні багатьох задач, щодо обробки природної мови. Також, основною перевагою BERT є те, що мережа вже навчена на великій кількості даних, що звільняє розробників від необхідності навчати нейронну мережу.

Навчання нейронної мережі BERT можна розділити на два етапи: попереднє навчання та донавчання. При попередньому навчанні нейронна мережа отримує загальні знання про мовні конструкції та вчиться розуміти сенс текстів. Для цього етапу потрібно використовувати величезну кількість даних, ресурсів та часу. Саме тому, розробники мають змогу завантажити та використовувати в своїх проєктах вже навчену нейронну мережу. Це значно економить час та ресурси на розробку ПЗ.

Донавчання передбачає підлаштування вже навченої нейронної мережі під певну NLP задачу. Найпопулярнішим методом, який для цього використовується є занулення фінального шару ваг з метою навчання нейронної мережі на новому наборі даних, менших за обсягом. Таким чином, нейронна мережа навчається виконувати конкретну задачу.

Отже, для визначення якості автоматизованого перекладу необхідно довчити нейронну мережу BERT на підготовлених наборах даних. Тому, спосіб визначення якості автоматизованого перекладу можна розділити на два етапи: донавчання нейронної мережі BERT та використання навченої системи для визначення якісного перекладу (Рисунок 2.1).

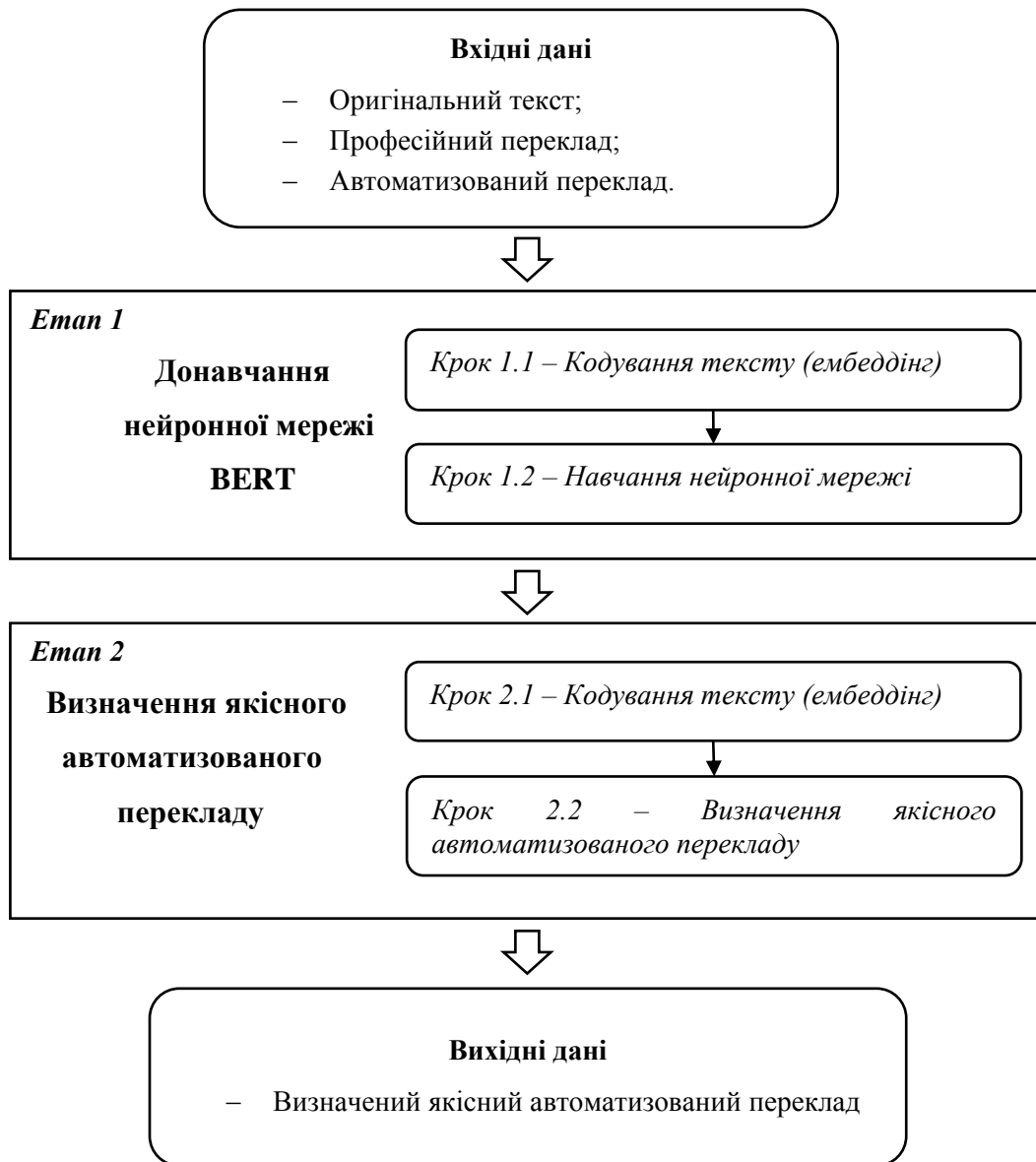


Рисунок 2.1 – Схема способу визначення якості автоматизованого перекладу з використанням нейронної мережі

Вхідними даними для першого етапу будуть набори текстів, які складатимуться з оригінального тексту та його професійного перекладу. Просто підготувати набори вхідних даних недостатньо, потрібно їх ще правильно обробити для подальшої передачі нейронній мережі. Тому, етап донавчання нейронної мережі BERT складається з таких послідовних кроків (Рисунок 2.2):

- ембедінг;
- навчання нейронної мережі.



Рисунок 2.2 – Етап донавчання нейронної мережі BERT

Оскільки мовна модель працює із текстом, а комп'ютер – із числами, то необхідно вхідні дані для нейронної мережі перетворити в числову інформацію. Для вирішення цього завдання текст ділиться на окремі частини та кодується. Текст можна ділити на окремі слова, склади чи навіть букви. Цю закодовану частину тексту називають ембедінгом (від англ. *embedding* – вкладати), тому що текстове значення вкладається у дійсне число [20]. Отже, ембедінг – це числове позначення частини тексту. Ембедінг потрібно провести як для оригінального тексту так і для перекладу, адже обидва тексти виступають у ролі вхідних даних для нейронної мережі.

Після того як вхідний текст буде оброблений та представлений у зрозумілій формі для нейронної мережі, його можна передавати мережі для аналізу та навчання. Таким чином, по завершенню першого етапу буде отримано навчену під конкретне завдання нейронну мережу BERT. Під конкретним завданням розуміється визначення якісного автоматизованого перекладу.

Вхідними даними для другого етапу будуть не лише оригінальні тексти та набори їхніх автоматизованих перекладів, а й раніше навчена нейронна мережа (Рисунок 2.3). Під наборами перекладів тексту розуміється декілька автоматизованих перекладів одного і того ж оригінального тексту, створених з використанням різних сервісів.

Другий етап є основним для користувача застосунку, адже цей етап матиме візуальне представлення, тобто інтерфейс користувача. Другий етап складається з таких послідовних кроків:

- ембеддінг;
- визначення якісного перекладу.

Як і на першому етапі, на другому також необхідно провести ембеддінг кожного автоматизованого перекладу з набору та відповідного оригінального тексту.

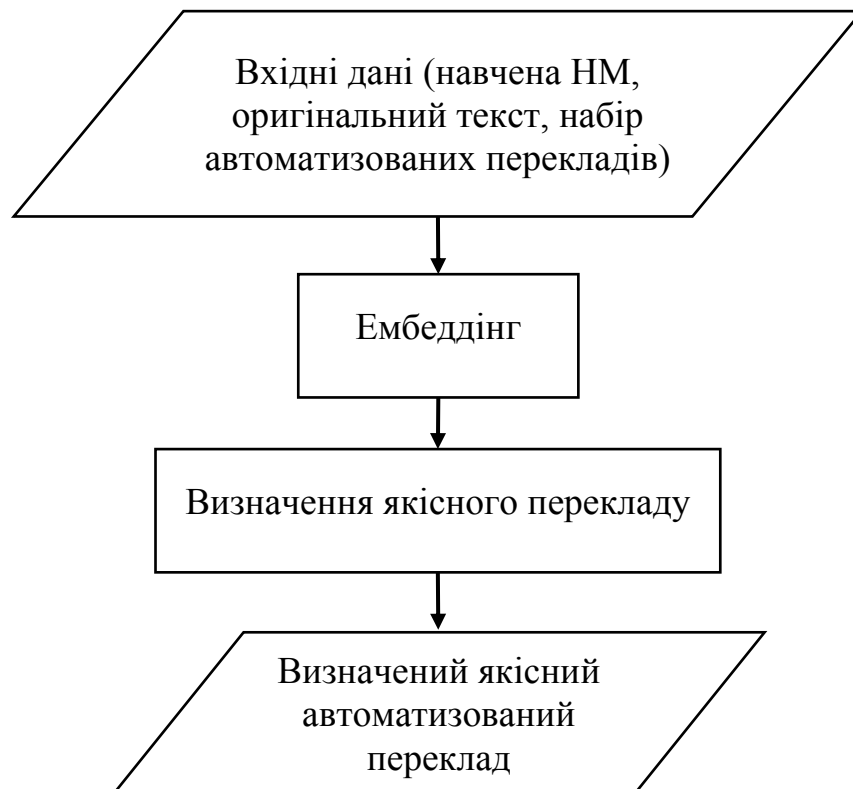


Рисунок 2.3 – Етап визначення якісного автоматизованого перекладу

На наступному кроці, навчена нейронна мережа отримує на вхід числові вектори оригінального тексту та набір числових векторів автоматизованих перекладів цього тексту. Нейронна мережа порівнює та аналізує передані їй дані, та визначає найкращий автоматизований переклад із набору, базуючись на раніше сформованих зв'язках.

Таким чином, результатом роботи другого етапу та усієї технології визначення якості автоматизованого перекладу, є визначений найякісніший переклад з набору декількох автоматизованих перекладів оригінального тексту.

2.2 Спосіб визначення якості автоматизованого перекладу

Основою методу визначення якості автоматизованого перекладу є використання нейронної мережі BERT. Оскільки дана мережа приймає на вхід не символічні значення тексту а їхню закодовану в числа форму, необхідно обрати алгоритм для векторизації тексту.

Однією з найпоширеніших технологій для векторизації тексту є алгоритм Word2Vec. Даний алгоритм аналізує текстову інформацію та за певними правилами присвоює кожному слову, складу чи літері певний числовий код, який ще називають семантичним вектором [21].

Просто надати словам певний числовий код недостатньо, адже сенс створення семантичних векторів полягає у контексті тексту, який аналізується. Відповідно до цього, сенс слова полягає не в тому з яких букв та звуків воно складається, а як часто воно зустрічається серед інших слів тексту.

Для визначення значень, які надаються словам будується матриця, в якій записуються усі слова із тексту. Рядками та стовпцями матриці є перелік слів, а перетин рядків і стовпців – це значення, яке вказує яку кількість разів одне слово зустрілось біля іншого (Таблиця 2.1).

Якщо виписати перший рядок таблиці (15, 0, 0) то можна отримати смисловий вектор слова «абстрактний». У випадку визначення якісного

автоматизованого перекладу тексту формується дві матриці. У першій матриці кодуються слова оригінального тексту, а в другій – його перекладу.

Таблиця 2.1 – Приклад векторизації термінології тексту

	Абстрактний	Клас	Штучний	...	Інтелект
Абстрактний	-	15	0		0
Клас	15	-	1		2
Штучний	0	1	-		35
...					
Інтелект	0	2	35		-

Наступним кроком йде порівняння створених векторів. На цьому кроці порівнюються вектори оригінального тексту та перекладу. Для цього часто використовується така метрика схожості, як косинусна схожість. Під косинусною схожістю розуміється міра схожості між двома векторами, що визначається значенням косинуса кута між ними [22]. Як результат, чим більша косинусна схожість двох векторів, тим більш схожими вважаються ці вектори. Аналогічно можна вважати, що якщо косинусна схожість між векторизованим текстом та його перекладом досить велика, то переклад вважається дуже точним. І навпаки, у разі низької косинусної схожості, якість перекладу буде поганою.

Перевагою використання нейронної мережі BERT є її двонаправленість. Тобто, на відмінну від інших подібних мереж, BERT аналізує весь текст одночасно. Спрощена архітектура нейронної мережі BERT зображена на рисунку 2.4.

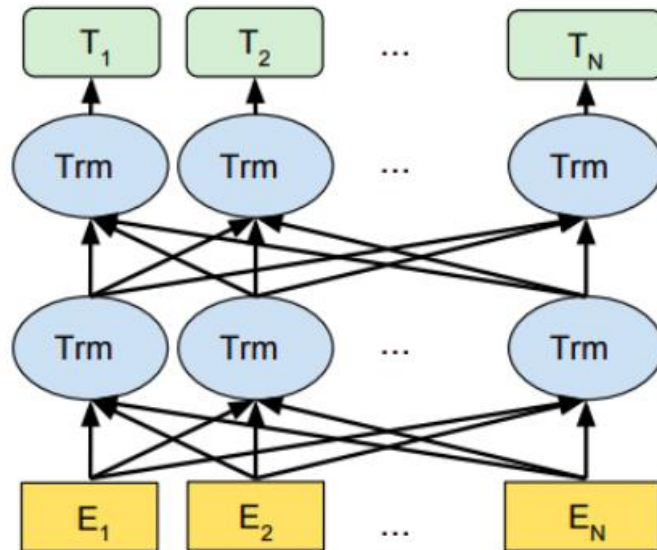


Рисунок 2.4 – Спрощена архітектура нейронної мережі BERT [23]

Тобто, нейронна мережа може враховувати контекст слова дивлячись на сусідні слова з обох боків одночасно, а не лише з якогось одного боку.

Ще однією перевагою використання нейронної мережі BERT у проєктах є можливість налаштування мережі під певні завдання. Дана ідея ґрунтується на тому, що нейронна мережа отримує базові знання, тобто матрицю ваг, до того, як мережа буде використана для вирішення конкретного завдання. Як результат, це допомагає краще вирішувати вузькі завдання щодо обробки природної мови.

Для того щоб навчити модель вирішувати конкретну задачу, поверх існуючої матриці ваг, додають ще один шар нейронів. На початку ваги цього шару заповнюються випадковими числами, які потрібно налаштувати під власні потреби. Нові тренувальні дані подаються в векторизованому форматі, тобто слова закодовані в числа.

Завдання визначення якісного перекладу можна звести до задачі класифікації, де різні автоматизовані переклади будуть класифікуватись як якісні або не якісні. На етапі навчання новий шар нейронів міститиме класифікуючий токен S , який визначатиме як пов'язані два речення між собою (Рисунок 2.5). Тобто, модель визначатиме чи є автоматизований переклад тексту якісним.

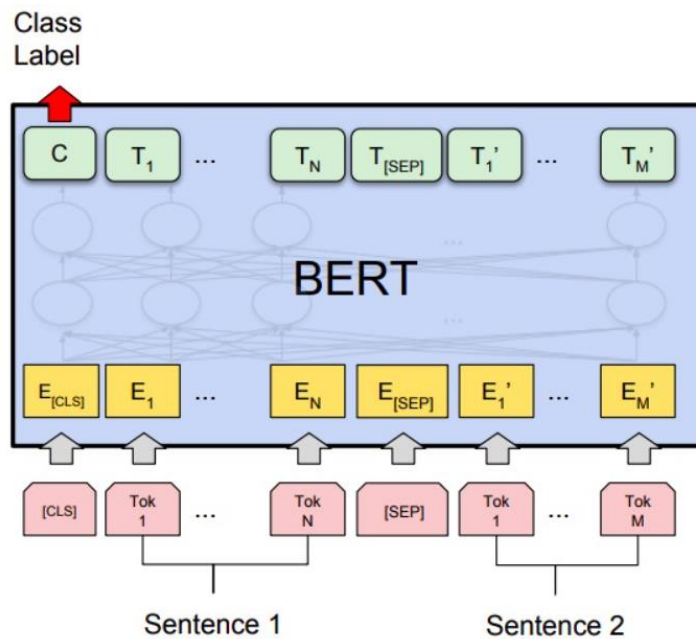


Рисунок 2.5 – Процес донавчання нейронної мережі під конкретну задачу [23]

Коли нейронна мережа буде навчена її можна використовувати для визначення якісного перекладу. На вхід, як і раніше, подаються векторизовані тексти, проте тепер їх значно більше (Рисунок 2.6). Завдання нейронної мережі полягає в порівнянні векторів за косинусною схожістю та визначення найкращого за якістю автоматизованого перекладу тексту.

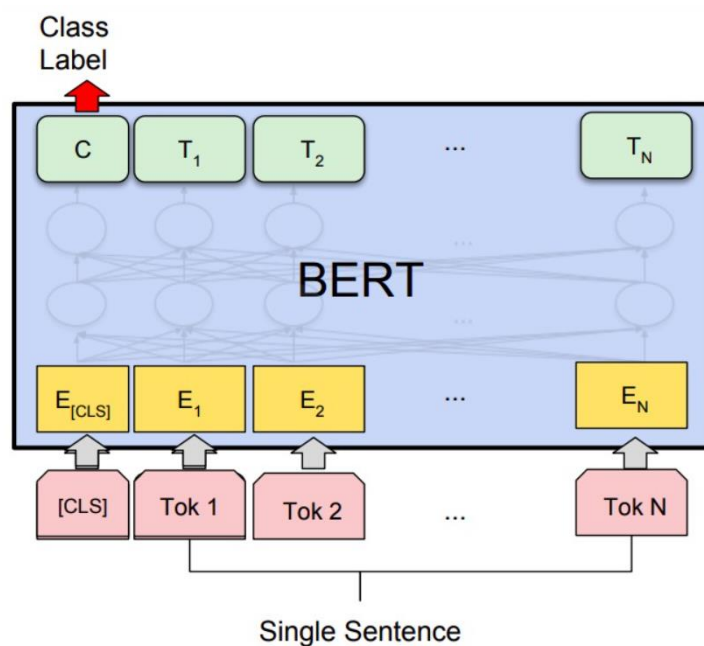


Рисунок 2.6 – Процес визначення якісного автоматизованого перекладу використовуючи навчену НМ [23]

Передбачення класифікації токена C полягає в визначенні пари двох речень (оригінального тексту та автоматизованого перекладу), де значення косинусної схожості максимальне. Передбачення відбувається з використанням токена CLS, який модель навчилася передбачати на навчанні.

Таким чином відбувається налаштування нейронної мережі BERT для вирішення конкретної задачі, яка полягає в визначенні якісного автоматизованого перекладу тексту.

2.3 Підготовка зразків робочих вхідних даних для інформаційної системи

Кожна нейронна мережа потребує великої кількості даних для навчання. Ці дані повинні бути завчасно сформовані, та відповідати критеріям, необхідним для подальшої роботи системи, яка використовуватиме вже навчену нейронну мережу.

Для способу визначення якості автоматизованого перекладу, наборами вхідних даних будуть оригінальні тексти та їхні автоматизовані переклади. Вхідні дані для системи будуть розділені на дві категорії: дані для навчання та дані для тестування роботи нейронної мережі.

Якість автоматизованого перекладу можна визначати за декількома параметрами, серед яких основними є правильна послідовність викладу інформації, відповідність термінології та вірна граматична конструкція речень. Врахування усіх показників – це досить складна та масштабна задача, тому було прийнято конкретизувати поставлене завдання та визначати якість перекладу за одним параметром який аналізує лише відповідність термінології у тексті. Адже саме з термінологією виникає найбільше проблем при створенні автоматизованого перекладу тексту.

Найбільшу кількість термінології можна знайти у різноманітних термінологічних словниках та довідниках. Зазвичай такі довідники створюються

під конкретну сферу діяльності людини. Терміни у словниках сортуються за алфавітом та подаються у вигляді: термін – пояснення.

Найбільшу кількість термінів та їхніх пояснень можна знайти на вебресурсі «Вікіпедія». Для формування бази вхідних даних було обрано взяти термінологію зі сфери комп'ютерних наук [24]. Перевагою використання даних із обраного ресурсу є те, що Вікіпедія дозволяє перемикаати мову сторінок та одразу ж отримувати переклад потрібного терміну.

Набір вхідних даних для навчання формуватиметься із пояснення терміну на українській мові та його перекладу англійською (Рисунок 2.7). Готовий набір навчальних даних може складатись з довільної кількості елементів, проте, чим більшою буде їхня кількість тим краще буде навчена нейронна мережа. А від якості навчання нейронної мережі залежатиме подальше визначення якості автоматизованого перекладу.

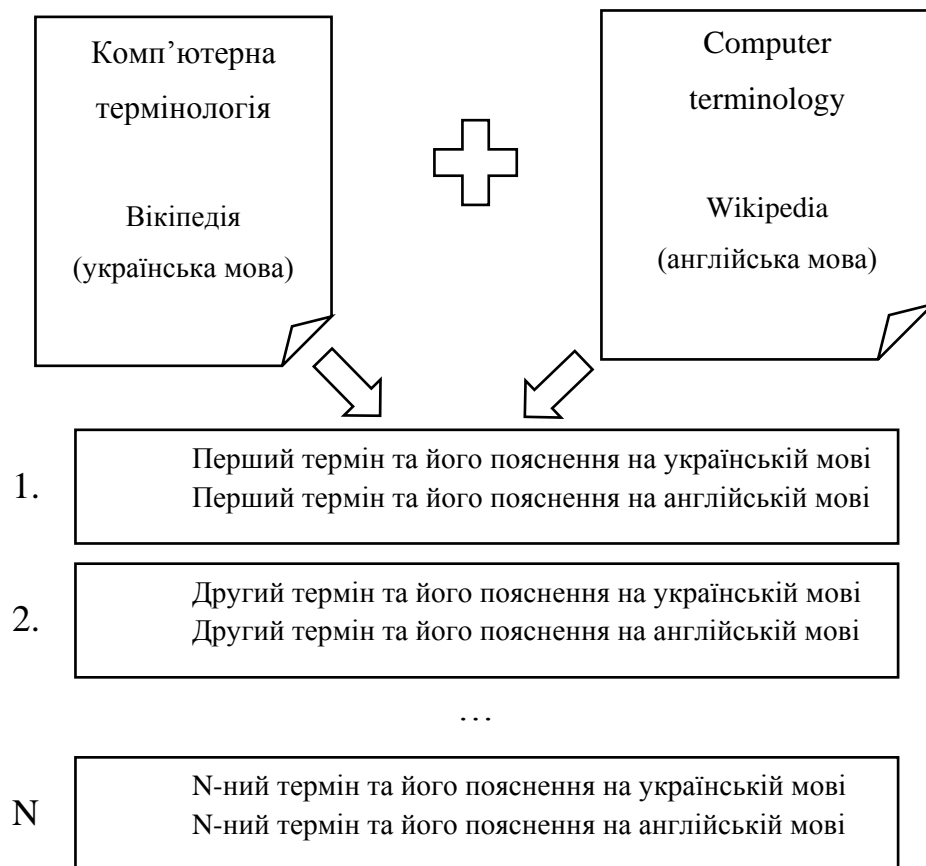


Рисунок 2.7 – Формування набору вхідних даних для навчання нейронної мережі

Після того як нейронна мережа була навчена на підготовлених даних, можна переходити до наступного етапу роботи системи. На наступному етапі застосунок повинен визначати найякісніший автоматизований переклад із набору декількох перекладів, створених різними системами автоматизованого перекладу текстів. Отже, один елемент з набору даних для тестування системи складається з одного речення оригінального тексту (пояснення терміну українською мовою) та декількох варіантів перекладу цього речення створених різними системами автоматизованого перекладу (Рисунок 2.8).

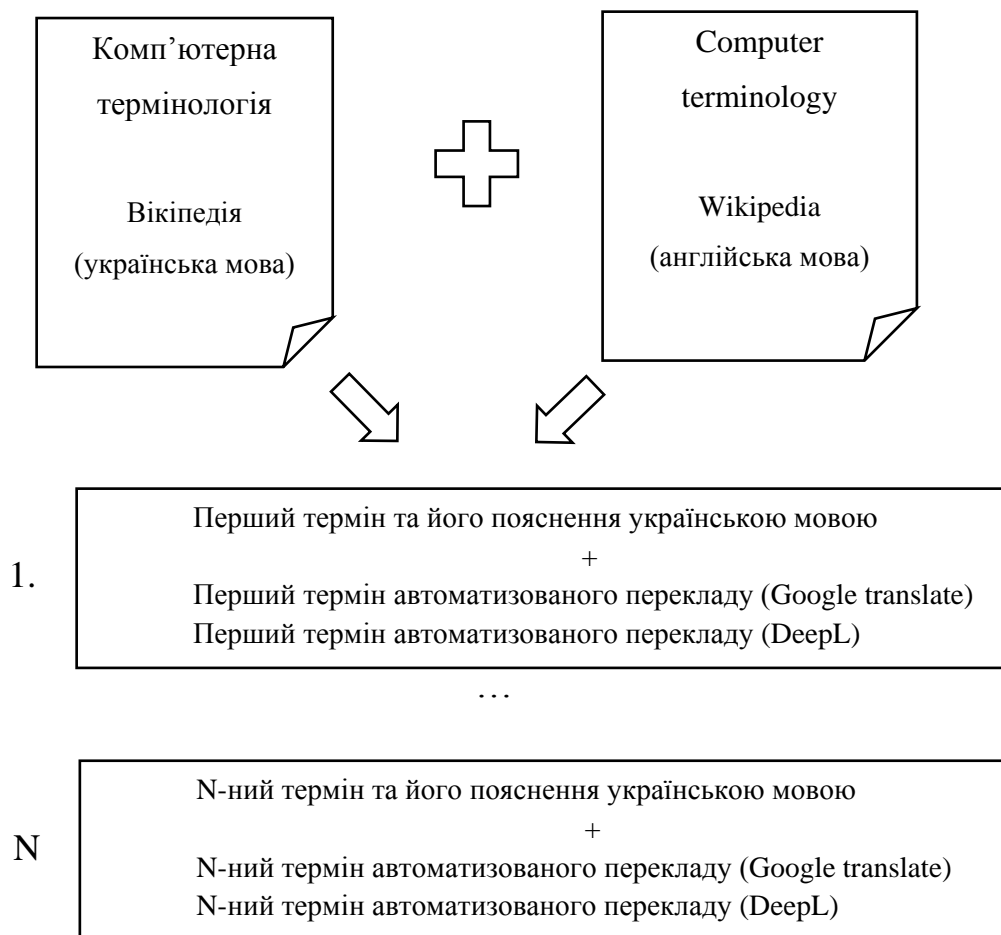


Рисунок 2.8 – Формування набору вхідних даних для тестування системи

Формування наборів вхідних даних для інформаційної системи дуже важливий процес, адже для будь-якого ПЗ необхідно використовувати велику кількість даних для навчання, обробки та аналізу. Особливу увагу варто звернути

при формуванні вхідних даних для навчання нейронної мережі, адже їх повинно бути багато. Також, дані для навчання повинні мати високу якість.

2.4 Якість визначення перекладу за допомогою нейронної мережі BERT

Якість визначення перекладу з використанням нейронної мережі BERT може бути дуже високою, завдяки його здатності розуміти контекст та використовувати глибокі семантичні залежності. BERT навчається на великому корпусі тексту, що дозволяє моделі усвідомлювати різноманітні контексти та нюанси мови. Його архітектура заснована на трансформерних механізмах, які дозволяють зберігати та використовувати інформацію з усього речення при визначенні перекладу.

За конкретними дослідженнями, проведеними в галузі машинного перекладу, BERT продемонстрував високу якість визначення перекладу на різних мовах. Наприклад, дослідження, проведені командою дослідників з Google AI, показали, що BERT здатний досягати кращих результатів у порівнянні з іншими схожими моделями на багатьох мовах, включаючи англійську, французьку та німецьку. Відмічалася здатність BERT до вирішення семантичних амбігвітностей та кращого узагальнення значень слів та фраз у контексті, BERT показує найвищі результати, що зображено (Рисунок 2.9).

#	Model	SST-2	QQP	MNLI-m	MNLI-mm
		Acc	F ₁ /Acc	Acc	Acc
1	BERT _{LARGE} (Devlin et al., 2018)	94.9	72.1/89.3	86.7	85.9
2	BERT _{BASE} (Devlin et al., 2018)	93.5	71.2/89.2	84.6	83.4
3	OpenAI GPT (Radford et al., 2018)	91.3	70.3/88.5	82.1	81.4
4	BERT ELMo baseline (Devlin et al., 2018)	90.4	64.8/84.7	76.4	76.1
5	GLUE ELMo baseline (Wang et al., 2018)	90.4	63.1/84.3	74.1	74.5
6	Distilled BiLSTM _{SOFT}	90.7	68.2/88.1	73.0	72.6
7	BiLSTM (our implementation)	86.7	63.7/86.2	68.7	68.3
8	BiLSTM (reported by GLUE)	85.9	61.4/81.7	70.3	70.8
9	BiLSTM (reported by other papers)	87.6 [†]	– /82.6 [‡]	66.9 [*]	66.9 [*]

Рисунок 2.9 – Результати дослідження моделей для перекладу

Дослідження проводилось за показниками SST-2,QQP,MNLI-M.

SST-2 (Stanford Sentiment Treebank) – є набором даних для аналізу настроїв тексту, розробленим на базі даних з твітів та рецензій на фільми. Завдання полягає в класифікації тексту на позитивний або негативний настрій.

QQP (Quora Question Pairs): – є набором даних, який містить питання з платформи Quora та відповіді на них. Завдання полягає в визначенні, чи є два питання еквівалентними за змістом.

MNLI-M (Multi-Genre Natural Language Inference, matched) - набір даних для завдання природньої мови, що визначає відношення між двома реченнями: "протиставлення", "припущення" або "нейтральність". Версія "matched" використовує збалансовані жанри тексту.

MNLI-MM - є аналогом MNLI-M, але використовується для порівняння моделей на базі несумісних даних.

Тобто дані для тренування та оцінки можуть містити жанри, які не були представлені в тренувальному наборі.

Враховуючи такі результати, BERT може вважатися однією з найефективніших моделей для перекладу текстів на українську мову та інші мови.

З точки зору функціоналу, програмний код реалізує функцію `execute_analysis`, яка виконує аналіз подібності між джерелом тексту (`source_text`) та цільовим текстом (`target_text`), використовуючи модель BERT. Це все необхідно для того, щоб розділити текст на токени, оцінюючи їх залежно від контексту на основі модельних оцінки.

Функція `compute_similarity` обчислює схожість між джерелом тексту та цільовим текстом. Вона викликає `get_bert_embedding` для отримання векторних представлень обох текстів. Потім використовується функція `cosine` для обчислення косинусної близькості між цими векторами.

Косинусна близькість між двома векторами A і B обчислюється за допомогою наступної формули:

$$\text{Similarity} = 1 - \frac{\sum_{i=1}^n A_i \cdot B_i}{\|A\| \cdot \|B\|} \quad (2.1)$$

де A_i і B_i – значення векторів A і B для i -го елемента, $\|A\|$ і $\|B\|$ – скалярні норми векторів A і B .

Ця формула обчислює косинус кута між векторами A і B , де скалярний добуток векторів відображає схожість між ними, а норми векторів нормалізують значення, яке належить проміжку від нуля до одиниці. Більша значення косинусної близькості вказує на більшу схожість між векторами.

Результат, який представляє схожість текстів, повертається. На основі значення кожного токена ми отримуємо масив векторних представлень, що згодом перетворюється у загальну оцінку перекладу за допомогою функції `compute_similarity` до `source_text` та `target_text`. Вони обчислюють схожість між ними на основі векторних представлень BERT за допомогою косинусної близькості.

Результат зберігається в змінній `similarity`. Варто зазначити, що ця оцінка існує в межах від 0.00 до 1.00, що відповідно дорівнює 0% та 100%. Тобто якщо оцінка 0.00 - то якість перекладу текстів найгірша, адже схожість не зустрічається. Якщо 1.00 то це означає повну схожість текстів, тобто ідеальний переклад.

На практиці ми маємо безліч слів, що можуть мати подібне значення, або схожий дослівний переклад, проте нейромережа токінізує текст та оцінює його відносно його позиції в тексті, що є досить тонким моментом.

Наприклад, розглянемо два слова - "сонце" та "світло". Обидва слова можуть мати подібне значення, оскільки "сонце" є джерелом світла. Проте, за допомогою векторного представлення BERT та косинусної близькості, можна виявити, яке з цих слів ближче до слова "день".

Припустимо, ми маємо наступні векторні представлення для цих слів: "сонце": [0.2, 0.7, 0.5, 0.1], "світло": [0.3, 0.6, 0.4, 0.2], "день": [0.8, 0.2, 0.4, 0.6]. Застосуємо косинусну близькість до векторів "сонце" та "день": `cosine_similarity("сонце", "день") = 0.89`. А тепер порівняємо косинусну близькість між "світло" та "день": `cosine_similarity("світло", "день") = 0.81`.

Отже, на основі косинусної близькості векторів, можна сказати, що слово "сонце" має більшу схожість до слова "день" (з більшим значенням косинусної близькості), ніж слово "світло". Таким чином і визначається якість визначення перекладу за допомогою нейронною мережі BERT.

2.5 Проєктування структури застосунку для визначення якості автоматизованого перекладу

Структура інформаційної системи способу визначення якості автоматизованого перекладу з використанням нейронної мережі складається з чотирьох підсистем (Рисунок 2.10):

- підсистема для роботи з базою даних;
- підсистема для навчання нейронної мережі BERT;
- підсистема для визначення якісного автоматизованого перекладу;
- підсистема для роботи з інтерфейсом користувача.

Відокремлення роботи із даними у окремо підсистему значно полегшує роботу із даними. Адже, коли усі дані правильно зберігаються в одному місці, з ними набагато простіше працювати. Виконуючи запити до бази даних можна додати дані в БД, отримати дані, а також відредагувати за потреби.

Підсистема для навчання нейронної мережі BERT включає в себе не лише процес донавчання нейронної мережі під конкретну задачу, а й підготовку вхідних даних для навчання. Дана підсистема пов'язана із БД, адже усі дані для навчання зберігаються в базі даних. Підсистема отримує дані про текстовий вміст оригінальних текстів та їхніх професійних перекладів та формує датасети вхідних даних. Перед тим як відправити ці датасети на входи нейронної мережі проводиться попереднє кодування тексту (ембеддінг). Після цього нейронна мережа працює із закодованим текстом та визначає значення ваг, необхідних для визначення якісного автоматизованого перекладу.



Рисунок 2.10 – Схема структури інформаційної системи способу визначення якості автоматизованого перекладу з використанням нейронної мережі

Підсистема для визначення якісного автоматизованого перекладу, це власне, робота із введеними користувачем текстами. Підсистема проводить кодування текстів та передає їх раніше навченій нейронній мережі. Нейронна мережа працює із текстами та визначає який автоматизований переклад із набору декількох перекладів найякісніший. Отриманні дані записуються у БД та відображаються користувачеві, як результат роботи усієї інформаційної системи.

Підсистема для роботи з інтерфейсом користувача передбачає проектування структури застосунку з яким взаємодіятиме кінцевий користувач програмного продукту. Структура застосунку для визначення якості автоматизованого перекладу передбачає можливість введення користувачем оригінального тексту та додавання декількох варіантів його перекладу,

створеного різними системами автоматизованого перекладу текстів. Тому, необхідно спроектувати зручний та зрозумілий інтерфейс користувача.

Інтерфейс застосунку складатиметься із двох віконних вкладок: вкладка для додавання текстів та вкладка з результатами аналізу введених текстів. Користувач матиме змогу вільно перемикатись між вкладками, проте вкладка з результатами аналізу відобразить інформацію лише після правильного додавання текстів на попередній вкладці.

Основна частина інтерфейсу з яким має взаємодіяти користувач знаходиться на першій вкладці «Додавання текстів» (Рисунок 2.11). Тут розміщено дві форми, перша з яких призначена для введення оригінального тексту. В другу форму користувач повинен ввести декілька окремих варіантів автоматизованого перекладу оригінального тексту. Важливо, щоб переклади відрізнялись та були створені різними системами автоматизованого перекладу текстів. Для зручності та можливості перемикання між варіантами перекладу, на формі передбачений елемент перемикання. Це дасть змогу користувачеві швидко переміщатись між доданими варіантами перекладу та редагувати їх, при необхідності.



Рисунок 2.11 – Схема інтерфейсу додавання текстів для аналізу

Для зручного сприйняття результатів роботи системи створено окрему вкладку, яка відобразатиметься після проведення аналізу текстів. На цій вкладці користувач зможе побачити найякісніший автоматизований переклад з доданих ним раніше (Рисунок 2.12).

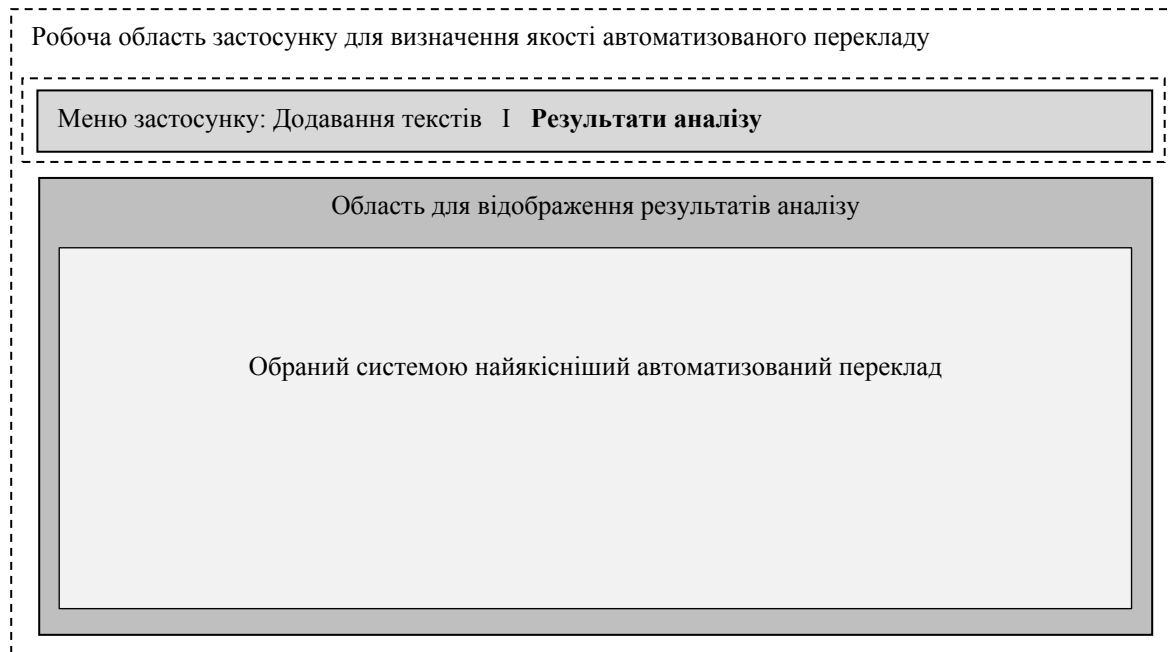


Рисунок 2.12 – Схема інтерфейсу відображення результатів аналізу

Крім цього, при виникненні помилок користувач повинен бути сповіщений про помилку у зрозумілій для нього формі. Застосунок повинен адекватно реагувати на неправильні дії користувача та давати зрозумілі пояснення при виникненні проблем. Адже, якщо користувач просто отримає код помилки без будь-яких пояснень, він не зможе зрозуміти що він зробив не правильно.

Отже, взаємодія користувача із застосунком для визначення якості автоматизованого перекладу починається із входу у систему (Рисунок 2.13). Після цього користувач має змогу додати тексти (оригінал та переклад) у відповідні поля. Після введення текстів користувач може їх переглядати та редагувати за потреби.

Коли користувач буде задоволений текстами, які він додав та буде готовий отримати результат, користувач зможе перейти на наступний крок

взаємодії із системою – отримання результатів аналізу. Проте, перед тим, як отримати результати створена інформаційна система повинна перевірити чи введені користувачем тексти відповідають вимогам. У разі виникнення помилок, користувач буде проінформований про причину збою у системі. Коли користувач виправить допущенні помилки, ітерація перевірки повториться. Цей процес продовжуватиметься доти, доки користувач не введе тексти, відповідно до вимог. Після цього, користувач зможе побачити результати роботи системи у вигляді обраного найякіснішого автоматизованого перекладу тексту.

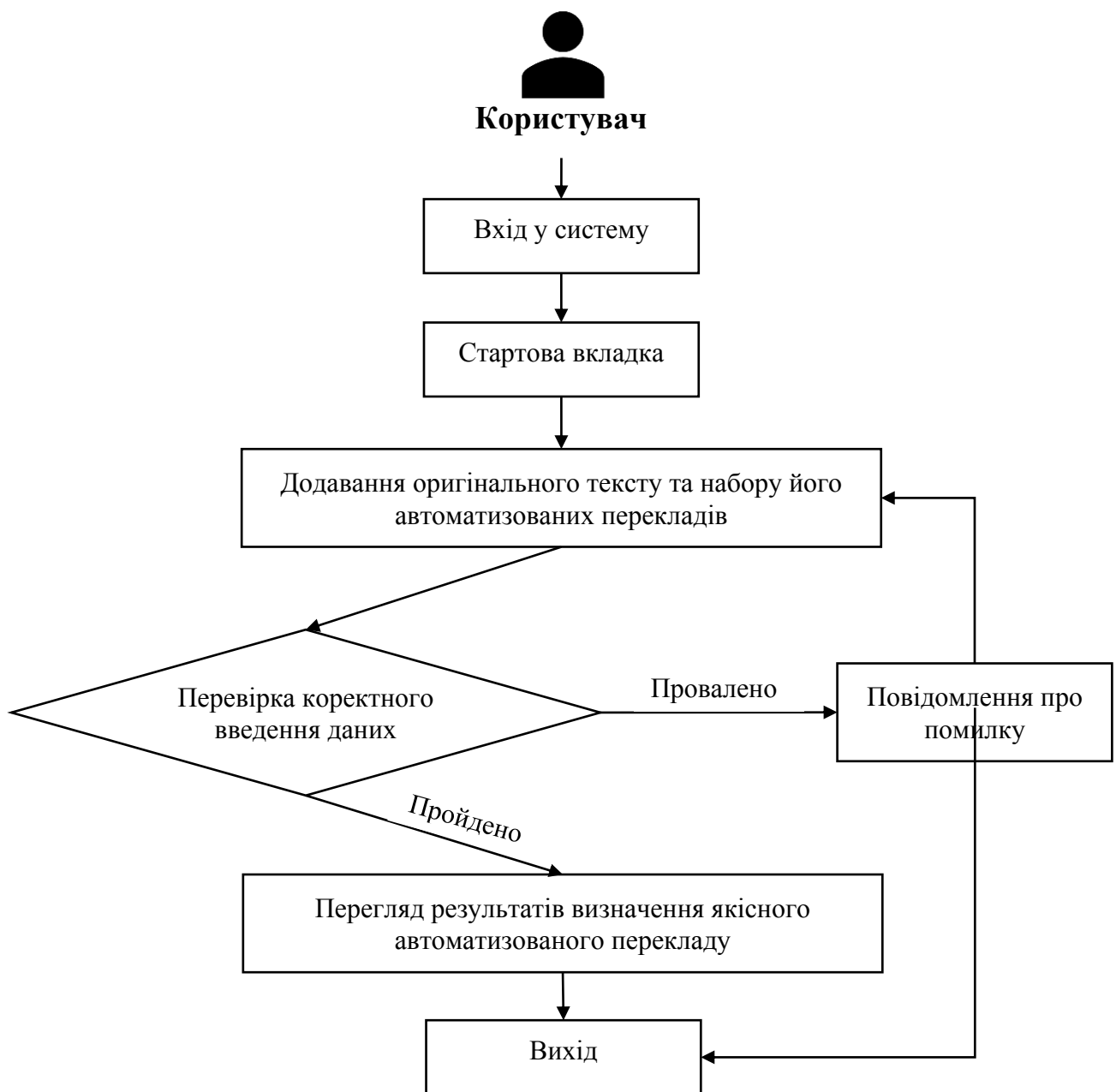


Рисунок 2.13 – Діаграма активності користувача застосунку для визначення якості автоматизованого перекладу

Розділення роботи системи на окремі взаємопов'язані підсистеми значно спрощує розробку програмного застосунку. Адже, різний функціонал знаходитиметься в різних файлах програми, що допоможе уникнути плутанини та нагромадження зайвого коду. Аналогічно, розділення інтерфейсу користувача на окремі вкладки допоможе уникнути плутанини при використанні застосунку. Це значно спрощує взаємодію користувача із системою, адже окремий функціонал знаходитиметься на окремих вкладках.

2.6 Даталогічна модель бази даних застосунку для визначення якості автоматизованого перекладу

Для збереження даних для навчання нейронної мережі та її подальшого тестування необхідно спроектувати структуру бази даних. БД застосунку для визначення якості автоматизованого перекладу складається з п'яти таблиць: Professional translations, Original texts, Languages, Automated translations, Automated translation systems (Рисунок 2.14).

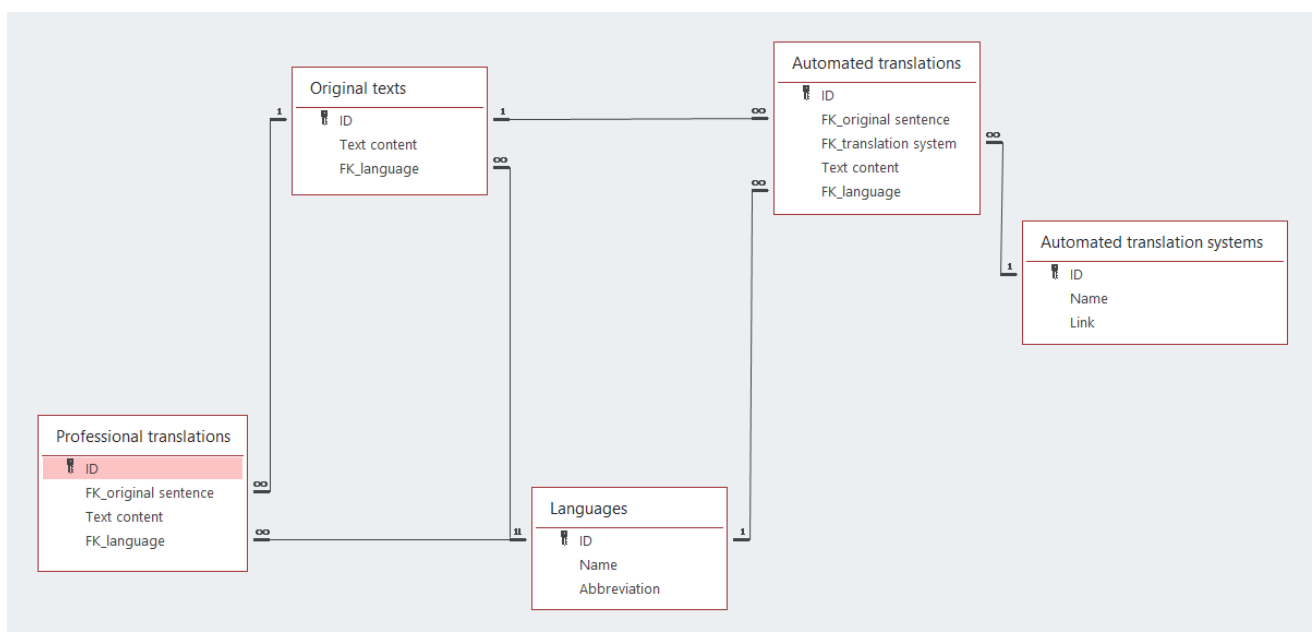


Рисунок 2.14 – Даталогічна модель бази даних застосунку для визначення якості автоматизованого перекладу

Таблиця «Original text» зберігає оригінальні тексти, на основі якої буде проводитись навчання нейронної мережі. У даному випадку в ролі текстів використовуються терміни та їхні пояснення із термінологічного словника. Кожен термін буде розглядатись як окреме речення. Також, з врахуванням подальшого розвитку функціоналу інформаційної системи, таблиця зберігатиме мову, якою написаний текст. Це необхідно для того, щоб в майбутньому мати змогу навчати нейронну мережу на декількох мовах. Отже, таблиця «Original texts» має такі атрибути: ID, Text content, FK_language (Таблиця 2.2).

Таблиця 2.2 – Атрибути таблиці «Original texts»

№ п/п	Назва	Тип даних	Опис
1.	ID	int	Первинний ключ.
2.	Text content	text	Текстовий вміст речення (термін та його пояснення).
3.	FK_language	int	Вторинний ключ. Зв'язок між текстом та мовою на якій він написаний. Необхідний для визначення мови тексту.

Для навчання нейронної мережі необхідно підготувати набори вхідних даних, які включатимуть терміни написанні двома мовами. Англomовний переклад термінів взятий із ресурсу «Вікіпедія», в контексті даної задачі буде вважатись професійним. Тому, для збереження перекладу термінів було створено таблицю «Professional translations» з такими атрибутами: ID, FK_original sentence, Text content, FK_language (Таблиця 2.3)

Таблиця 2.3 – Атрибути таблиці «Professional translations»

№ п/п	Назва	Тип даних	Опис
-------	-------	-----------	------

1.	ID	int	Первинний ключ.
2.	FK_original sentence	int	Вторинний ключ. Зв'язок між перекладом та оригінальним реченням.
3.	Text content	text	Текстовий вміст перекладу
4.	FK_language	int	Вторинний ключ. Зв'язок між текстом та мовою на яку він перекладений. Необхідний для визначення мови тексту.

Для зберігання інформації про мову, якою написані тексти створено таблицю «Languages» з такими атрибутами: ID, Name, Abbreviation (Таблиця 2.4).

Таблиця 2.4 – Атрибути таблиці «Languages»

№ п/п	Назва	Тип даних	Опис
1.	ID	int	Первинний ключ.
2.	Name	text	Назва мови
3.	Abbreviation	text	Скорочення мови (Ua/ Eng)

Оскільки, після навчання нейронної мережі її роботу потрібно протестувати на визначення якості автоматизованого перекладу, то необхідно підготувати дані для тестування. Для збереження автоматизованих перекладів тексту спроектовано окрему таблицю в базі даних під назвою «Automated translations» з таким набором атрибутів: ID, FK_original sentence, FK_translation system, Text content, FK_language (Таблиця 2.5)

Таблиця 2.5 – Атрибути таблиці «Automated translations»

№ п/п	Назва	Тип даних	Опис
1.	ID	int	Первинний ключ.
2.	FK_original sentence	int	Вторинний ключ. Зв'язок між автоматизованим перекладом та

			оригінальним реченням.
3.	FK_translation system	int	Вторинний ключ. Зв'язок між автоматизованим перекладом та системою перекладу.
4.	Text content	text	Текстовий вміст перекладу
5.	FK_language	int	Вторинний ключ. Зв'язок між текстом та мовою на яку він перекладений.

Оскільки, для визначення якісного перекладу необхідно створити декілька варіантів автоматизованого перекладу одного і того ж речення використовуючи різні системи перекладу, виникає необхідність у створенні таблиці, яка зберігатиме дані про системи перекладу. Тому, в БД створено таблицю «Automated translation systems» з такими атрибутами (Таблиця 2.6):

Таблиця 2.6 – Атрибути таблиці «Automated translation systems»

№ п/п	Назва	Тип даних	Опис
1.	ID	int	Первинний ключ.
2.	Name	text	Назва системи автоматизованого перекладу
3.	Link	text	Посилання на систему автоматизованого перекладу

Спроектвана структура бази даних застосунку для визначення якості автоматизованого перекладу дозволяє зберігати усі необхідні дані, зокрема оригінальні тексти та їхні переклади, які необхідні для навчання нейронної мережі та демонстрації роботи програми.

2.7 Висновки до розділу 2

Для визначення якості автоматизованого перекладу використовується нейронна мережа BERT, яка вже навчена на великій кількості даних та

пристосована для виконання базових задач пов'язаних із обробкою природної мови. Проте, при використанні нейронної мережі для розв'язку конкретних завдань необхідно довчити мережу на відповідних зразках вхідних даних.

Оскільки, було прийнято рішення визначати якість автоматизованого перекладу аналізуючи відповідність термінології в тексті, то вхідними наборами для навчання нейронної мережі будуть терміни із термінологічного словника та їхні професійні переклади. Таким чином, аналізуючи ці дані нейронна мережа навчиться розпізнавати якісний переклад із набору автоматизованих перекладів створеним різними системами автоматизованого перекладу текстів.

Для прикладного застосування способу визначення якості автоматизованого перекладу було вирішено реалізувати віконний застосунок. Робота із застосунком складатиметься із двох послідовних етапів. На першому етапі користувач повинен ввести оригінальний текст та декілька його автоматизованих перекладів. Після перевірки правильності введених даних та визначення найкращого автоматизованого перекладу користувач перейде до другого етапу роботи з застосунком. Другий етап – це, власне, перегляд результатів роботи інформаційної системи.

Для реалізації описаного застосунку було прийнято рішення використовувати такі засоби розробки як платформа .NET та мова програмування Python. Для збереження усі необхідних даних для навчання нейронної мережі та збереження результатів роботи програми використовуватиметься СКБД SQL Server. Поєднання обраних засобів розробки дає змогу виконати поставлені завдання щодо прикладної реалізації способу визначення якості автоматизованого перекладу з використанням нейронної мережі.

Розділ 3 Програмна реалізація застосунку для визначення якості автоматизованого перекладу з використанням нейронної мережі

3.1 Структура модулів застосунку для визначення якості автоматизованого перекладу

Для розробки застосунку для визначення якості автоматизованого перекладу з використанням нейронної мережі було прийнято рішення використати мову програмування Python. Python є однією з найбільш популярних мов програмування у світі, особливо в галузі машинного навчання та обробки даних [25].

Однією з переваг використання Python є велика кількість бібліотек, призначених для машинного навчання та обробки природної мови. Наприклад, бібліотеки Tensorflow та PyTorch дозволяють створювати та навчати нейронні мережі, що є необхідним для розробки застосунку для визначення якості автоматизованого перекладу. Крім того, Python є дуже простою та зрозумілою мовою програмування, що дозволяє швидко розробляти та тестувати програмне забезпечення. Python також підтримує багато парадигм програмування, включаючи об'єктно-орієнтоване, процедурне та функціональне програмування [26].

Для забезпечення функціональності програми прийнято розділяти програмний код на окремі взаємопов'язані модулі. Це дозволяє зробити програмний код більш організованим та керованим, адже, кожен модуль виконуватиме конкретний функціонал. Такий підхід значно спрощує розробку, тестування та підтримку життєвого циклу програми.

Відповідно до розробленого у 2 розділі способу визначення якості автоматизованого перекладу програмну реалізацію було розділено на 5 окремих компонентів (Рисунок 3.1).

Основним компонентом у структурі можна вважати модуль Main. Тут зосереджений весь код, який відноситься до реалізації графічного інтерфейсу

користувача. Також, даний модуль можна вважати точкою входу у програму, адже саме тут розпочинається робота користувача із застосунком.

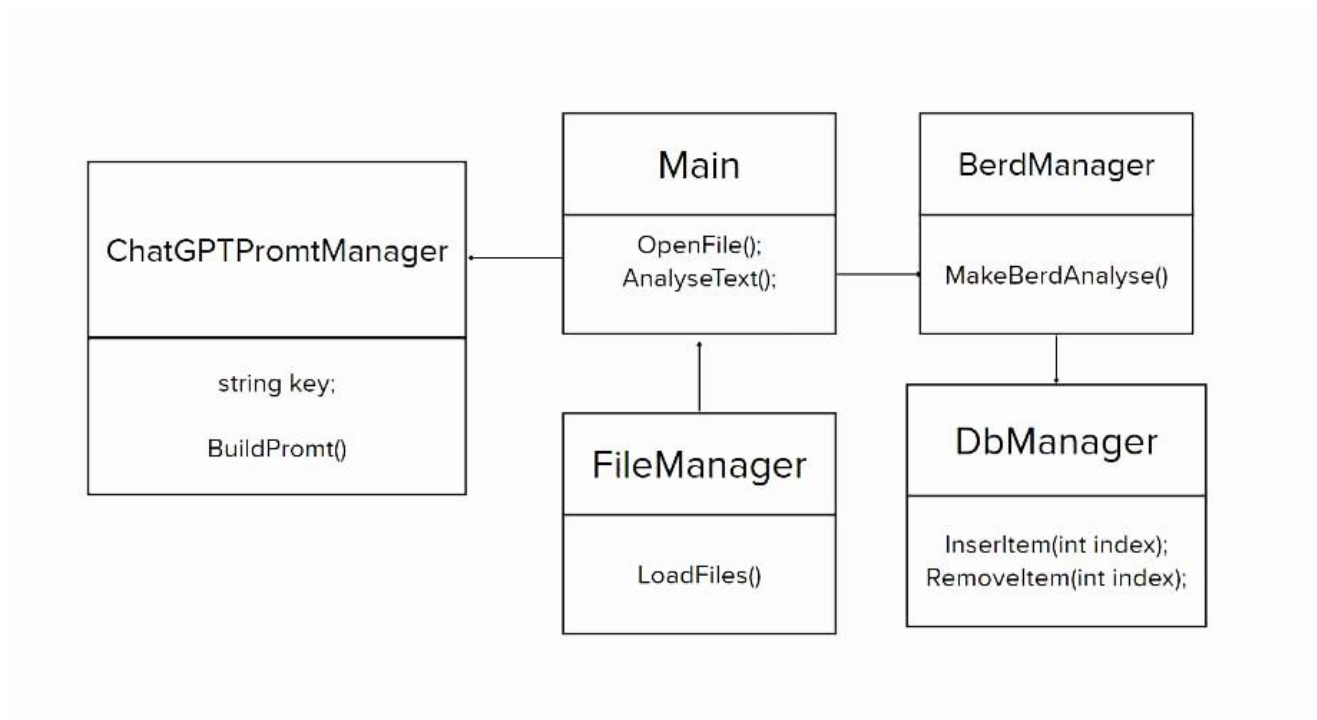


Рисунок 3.1 – Діаграма класів застосунку для визначення якості автоматизованого перекладу

Програмна логіка для завантаження файлів із текстами, також, винесена в окремий модуль – FileManager, який взаємодіє із головним модулем через інтерфейс користувача. Таким чином, весь програмний код, який відповідає за роботу з файлами, зберігається в одному місці та не засмічує зайвим кодом інші класи.

Оскільки, робота програми полягає в визначенні якості автоматизованого перекладу, який поріняють із професійним перекладом то потрібно визначити джерело для генерації професійного перекладу. Для вирішення цього завдання було прийнято рішення використати чат-бот ChatGPT, який добре виконує функцію перекладача [27]. Для того щоб отримати цей переклад автоматично, в програмі реалізовано окремий клас, який взаємодіє з ChatGPT. Описаний модуль називається ChatGPTPromtManager.

Наступний модуль BerdManager – це програмна реалізація способу визначення якості автоматизованого перекладу з використанням нейронної мережі. Саме у цьому модулі відбувається підключення та налаштування нейронної мережі BERT, яка визначає якість автоматизованого перекладу використовуючи косинусну близькість. Даний модуль взаємодіє з основним модулем, через передачу вхідних текстів від користувача та відображення результатів роботи системи.

Останній модуль DbManager – це модуль для роботи із базою даних. У цьому модулі відбувається підключення до БД. Також, тут реалізовані методи для додавання даних у БД та їхній перегляд.

Таким чином, розроблена структура модулів застосунку дозволяє відокремити різний функціонал на окремі взаємопов'язані модулі. Це дозволяє зробити програму більш гнучкою та полегшує розуміння її програмної реалізації.

3.2 Особливості реалізації застосунку для визначення якості автоматизованого перекладу

При розробці програмних продуктів, перш за все, варто продумати реалізацію інтерфейсу користувача. Інтерфейс користувача може бути двох видів: командний та графічний. У випадку командного інтерфейсу користувач взаємодіє із програмою через введення команд у спеціальній консолі. Графічний інтерфейс передбачає створення вікон програми із елементами керування, такими як кнопки, поля для введення тексту, вкладки та інші елементи. Графічний інтерфейс є більш зручним та інтуїтивно зрозумілим, адже користувачу не потрібно розбиратись, яку команду ввести для отримання бажаного результату. Достатньо натиснути на потрібну кнопку і програма відобразить результат у зручному для користувача форматі. Ще однією перевагою графічного інтерфейсу є зручна обробка виникнення помилок. Тобто, при виникненні помилки, можна показати користувачу спеціальне вікно із причиною виникнення помилки та варіантами її вирішення.

Мова програмування Python більше орієнтована на розробку програм із командним інтерфейсом. Проте, існує спеціальна бібліотека, яка дає змогу через програмний код створити графічний інтерфейс користувача. Ця бібліотека називається tkinter. Для початку, необхідно підключити бібліотеку у проєкт. Після цього, використовуючи методи підключеної бібліотеки, можна створити повноцінний графічний інтерфейс користувача.

Графічний інтерфейс користувача складається із чотирьох окремих вкладок, кожна з яких наповнена різним функціоналом (Рисунок 3.2). Генерація описаного інтерфейсу користувача, який включає вікно із чотирма різними вкладками, реалізована з використанням методів бібліотеки tkinter. Дані методи дозволяють встановити розміри вікна застосунку та додавати необмежену кількість вкладок. Після того як вкладки додано, можна наповнювати вікна застосунку потрібними елементами шляхом додавання елементів через програмний код. Розташування елементів у вікні застосунку налаштовується через відступи по осі X та Y. Також, можна налаштувати інші стилі елементів, такі як шрифти, розмір та колір.

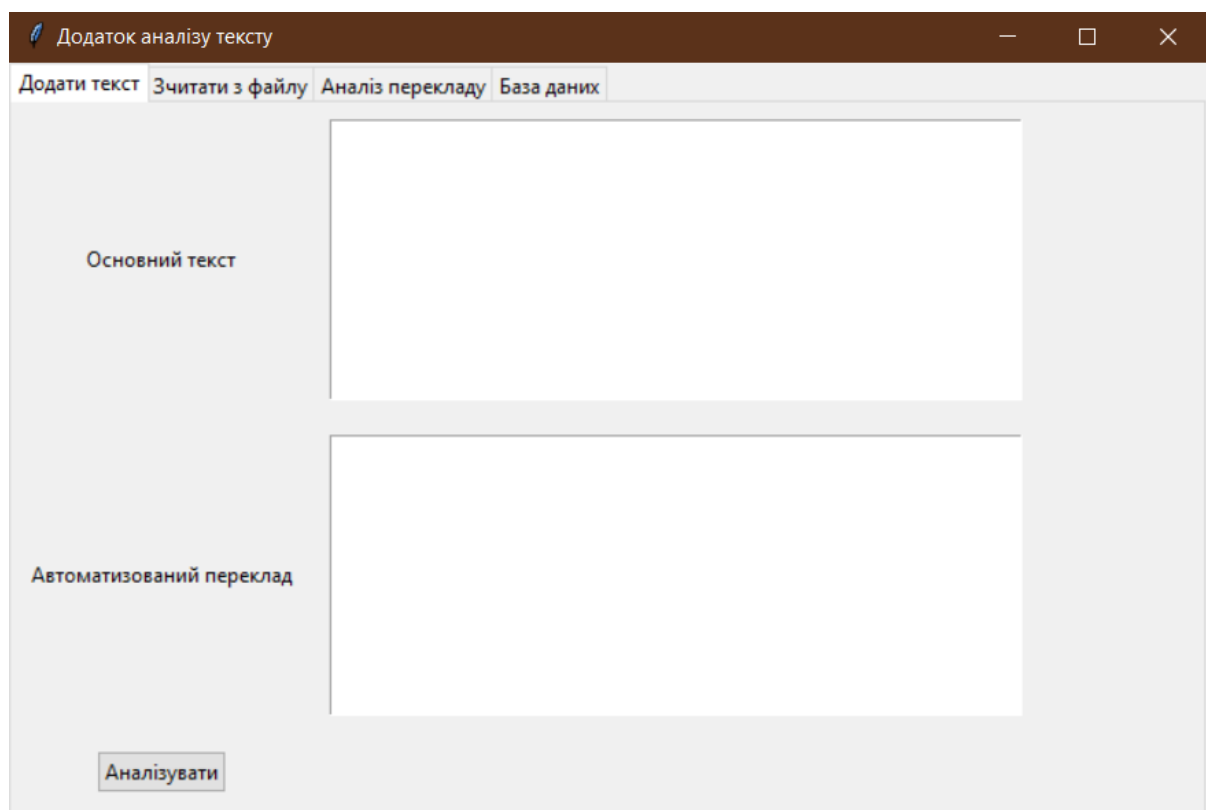


Рисунок 3.2 – Графічний інтерфейс користувача

На першій вкладці розміщено два поля для введення тексту. Перше поле призначене для введення тексту оригіналу, друге – для його автоматизованого перекладу (Рисунок 3.3).

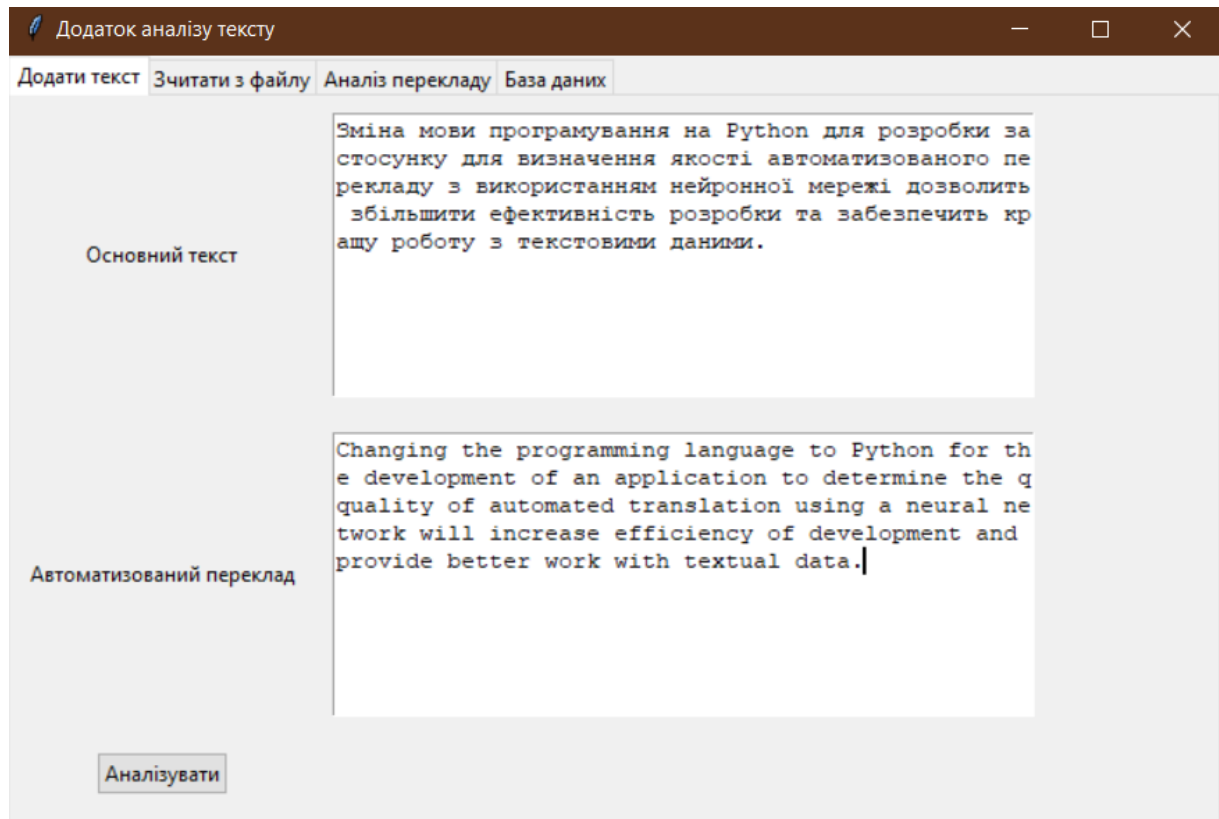


Рисунок 3.3 – Процес додавання тексту для аналізу

Користувачу не обов'язково вводити текст вручну у відповідні поля, адже в програмі передбачена можливість відкриття тексту із файлів (Рисунок 3.4). Це дозволяє спростити процес роботи з великими обсягами текстової інформації. Також, це дозволяє використовувати завчасно підготовлені тексти замість того щоб повторно вводити їх в програму.

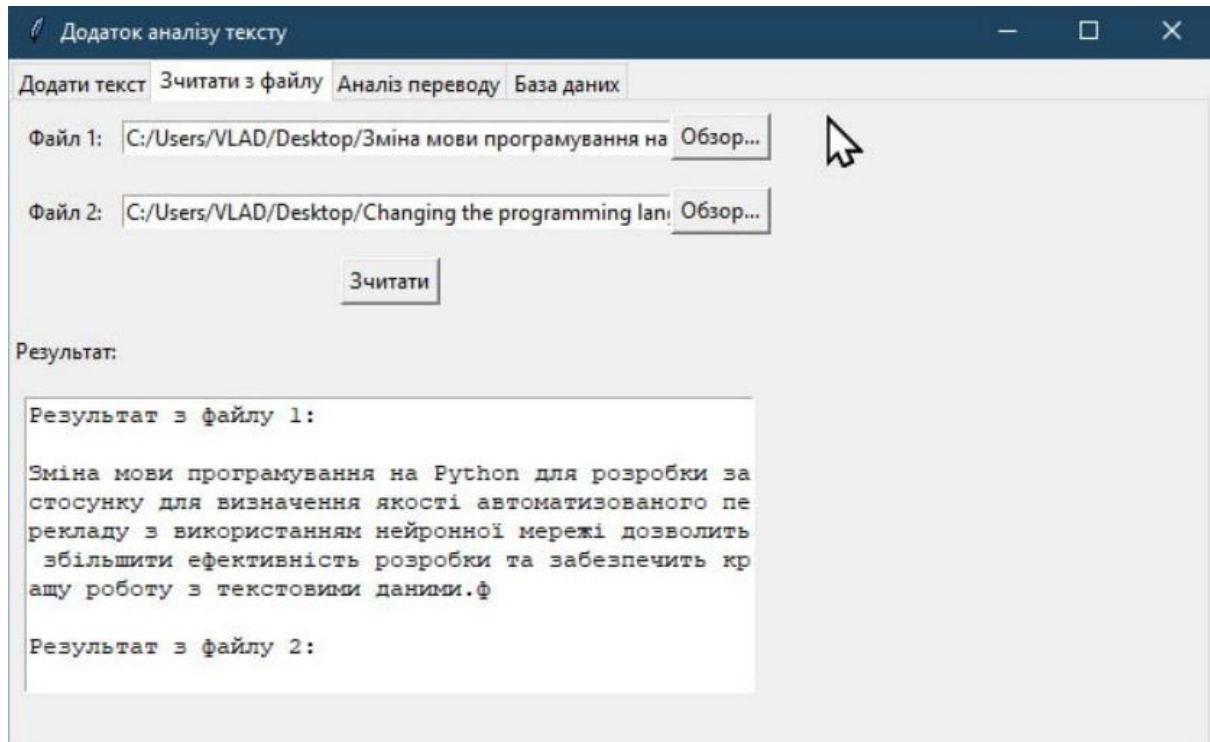


Рисунок 3.4 – Додавання текстів для аналізу із файлів

Алгоритм зчитування текстів із файлів складається з декількох послідовних кроків, які включають перевірки завантажених файлів зазначеним вимогам (Рисунок 3.5). Спочатку, користувачу потрібно перейти на вкладку «Зчитати з файлу». На цій вкладці буде дві кнопки для окремого завантаження кожного файлу. Коли користувач завантажить файли спрацьовує перша перевірка, яка перевіряє формат файлів, та чи обидва файли завантажені. Якщо формат файлів не відповідає текстовому або користувач завантажив лише один файл система відкриє вікно з текстом помилки.

Якщо користувач завантажив обидва файли, які відповідають зазначеному формату спрацьовує друга перевірка. Мета цієї перевірки полягає в визначенні мови, якою написані тексти у файлах. Дана перевірка стосується не лише текстів із файлу, а й текстів доданих вручну. Це пояснюється тим, що програма працює лише з україномовними та англійськими текстами. Перевірка мови тексту необхідна для обробки помилок, з якими може зіткнутись користувач під час використання програми. Наприклад, користувач може додати тексти на невідомій для програми мові, або додати два тексти написані однією мовою. В такому випадку, програма не зможе проаналізувати тексти та

визначити якість перекладу. Тому проведення цієї перевірки є важливим етапом перед проведенням аналізу текстів.

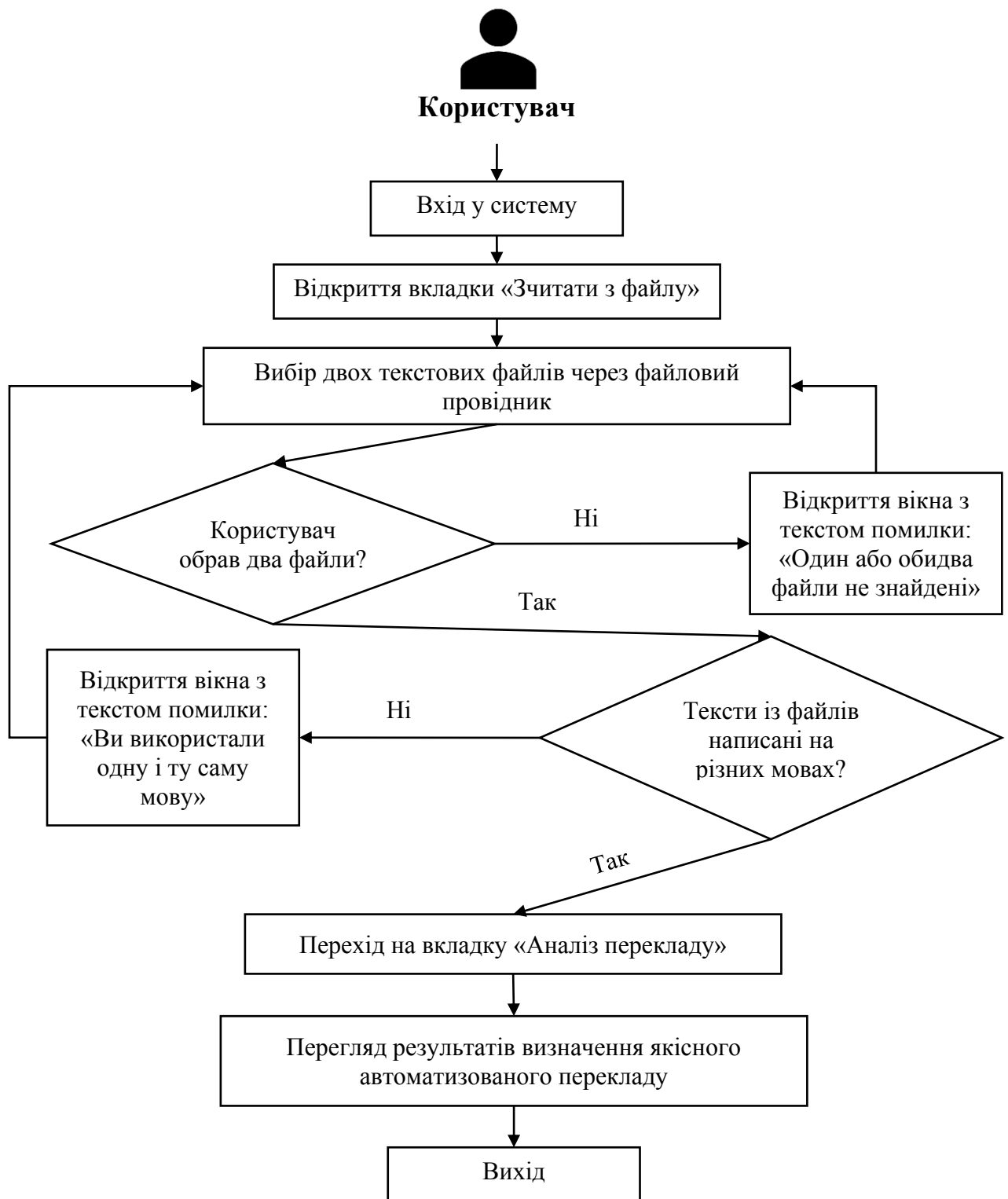


Рисунок 3.4 – Схема алгоритму завантаження та перевірки файлів

Після того, як користувач завантажить у програму тексти (оригінал та переклад) будь-яким зручним для нього способом, і ці тексти пройдуть усі попередні перевірки, стане доступною функція аналізу тексту. Ця функція визначатиме якість автоматизованого перекладу тексту.

Для вирішення даного завдання використовується попередньо навчена нейронна мережа BERT, архітектура якої описана в п.п 2.2. Для того щоб використовувати нейронну мережу необхідно підключити її в проєкт, як сторонню бібліотеку. Далі необхідно завантажити BERT-енкодер та токенайзер, який використовуватиметься для кодування текстів (ембедінгу). Кодування тексту відбувається з використанням методів підключеної бібліотеки. Таким чином, вхідні тексти перетворюються на набір токенів, які далі використовуватимуться як вхідні дані для нейронної мережі.

Для визначення якості автоматизованого перекладу необхідно порівняти автоматизований переклад із професійним перекладом. За професійний переклад приймається переклад згенерований через ChatGPT. Для автоматизованої роботи із ChatGPT через програмний код існує бібліотека openai. Дана бібліотека дозволяє створити з'єднання із чат-ботом, що відкриває можливість передавання запитів та отримання відповідей від ChatGPT всередині програми. Таким чином, користувачу не потрібно самостійно відкривати ChatGPT щоб створити необхідний переклад. Програма зробить це замість користувача. Такий підхід значно спрощує використання застосунку та економить час користувача.

Згенерований через ChatGPT професійний переклад, також, необхідно перетворити на набір токенів. Аналогічно як із текстами завантаженими користувачем, згенерований переклад кодується через використання методів бібліотеки для роботи з нейронною мережею BERT.

Далі, усі векторизовані тексти передаються на входи для нового шару нейронної мережі BERT. Нейронна мережа аналізує тексти та порівнює їх використовуючи косинусну схожість.

Результатом виконання даного алгоритму буде число в діапазоні від 0 до 1 (Рисунок 3.5). Чим ближчим буде це число до 1 тим якіснішим буде переклад, і навпаки, чим меншим буде число – тим гіршою буде якість перекладу.

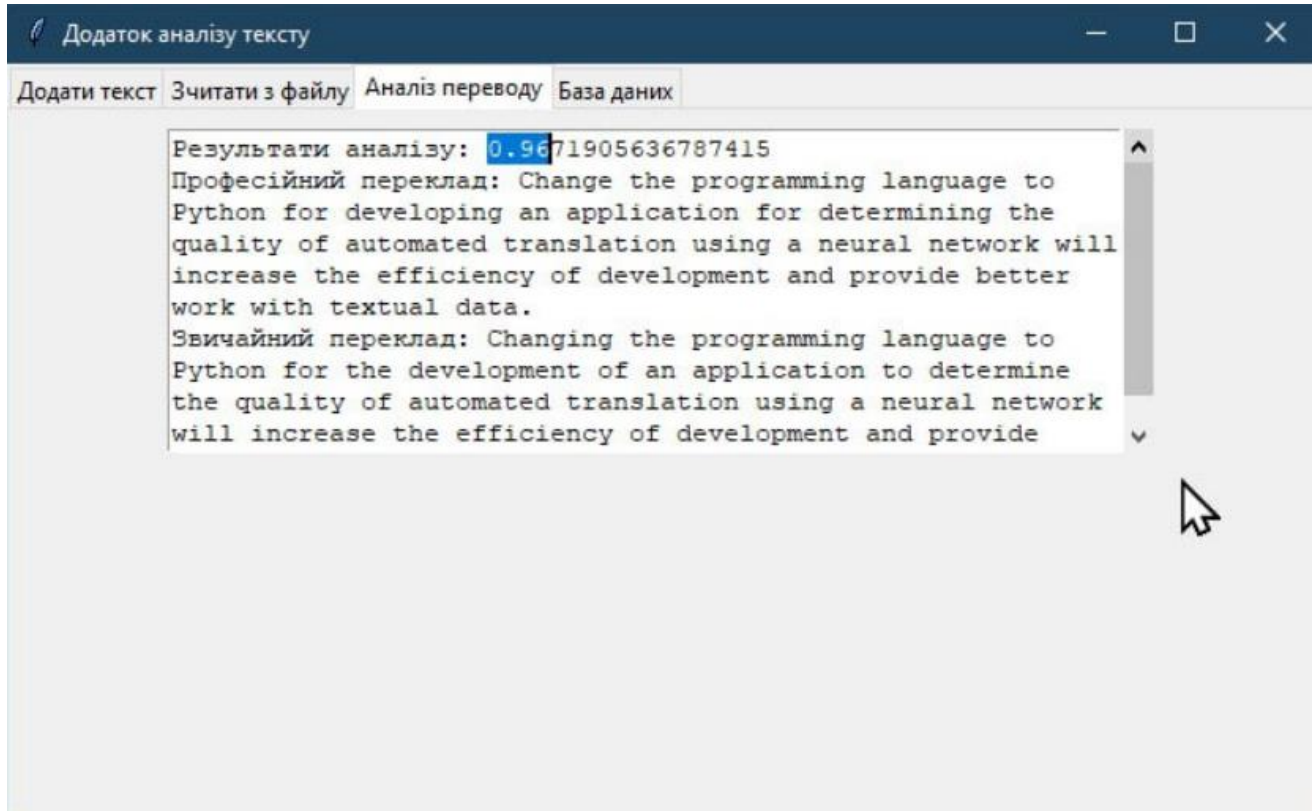


Рисунок 3.5 – Результати роботи способу визначення якості автоматизованого перекладу тексту з використанням нейронної мережі

Таким чином, використовуючи мову програмування Python вдалося реалізувати прикладне застосування розробленого раніше способу визначення якості автоматизованого перекладу з використанням нейронної мережі. Підключення додаткової бібліотеки дало змогу створити зручний та зрозумілий інтерфейс користувача. А використання раніше навченої нейронної мережі BERT значно пришвидшило реалізацію інформаційної системи, адже не потрібно було навчати нейронну мережу самостійно.

3.3 Опис функціональних можливостей застосунку для визначення якості автоматизованого перекладу

Після запуску застосунку для визначення якості перекладу перед користувачем відкривається вікно програми. Окремі етапи роботи програми розділені на окремі вкладки (Рисунок 3.6). Такий принцип дозволяє користувачу краще орієнтуватись у застосунку.

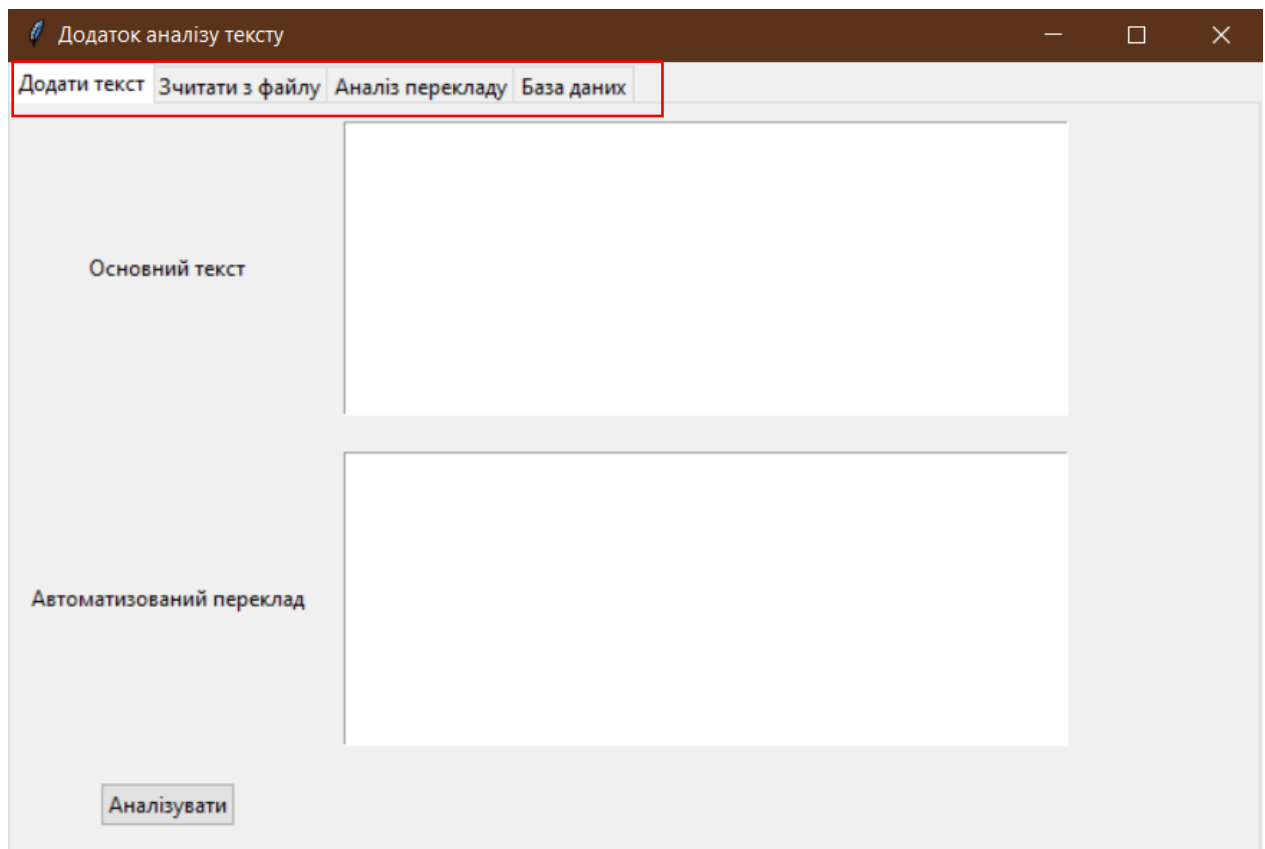


Рисунок 3.6 – Вигляд застосунку після запуску

На першій вкладці розташовані два поля, які призначені для введення тексту оригіналу та його автоматизованого перекладу. Для того, що перейти до аналізу тексту користувачу необхідно заповнити обидва поля (Рисунок 3.7). Після цього необхідно натиснути кнопку «Аналізувати» та перейти на вкладку «Аналіз перекладу».

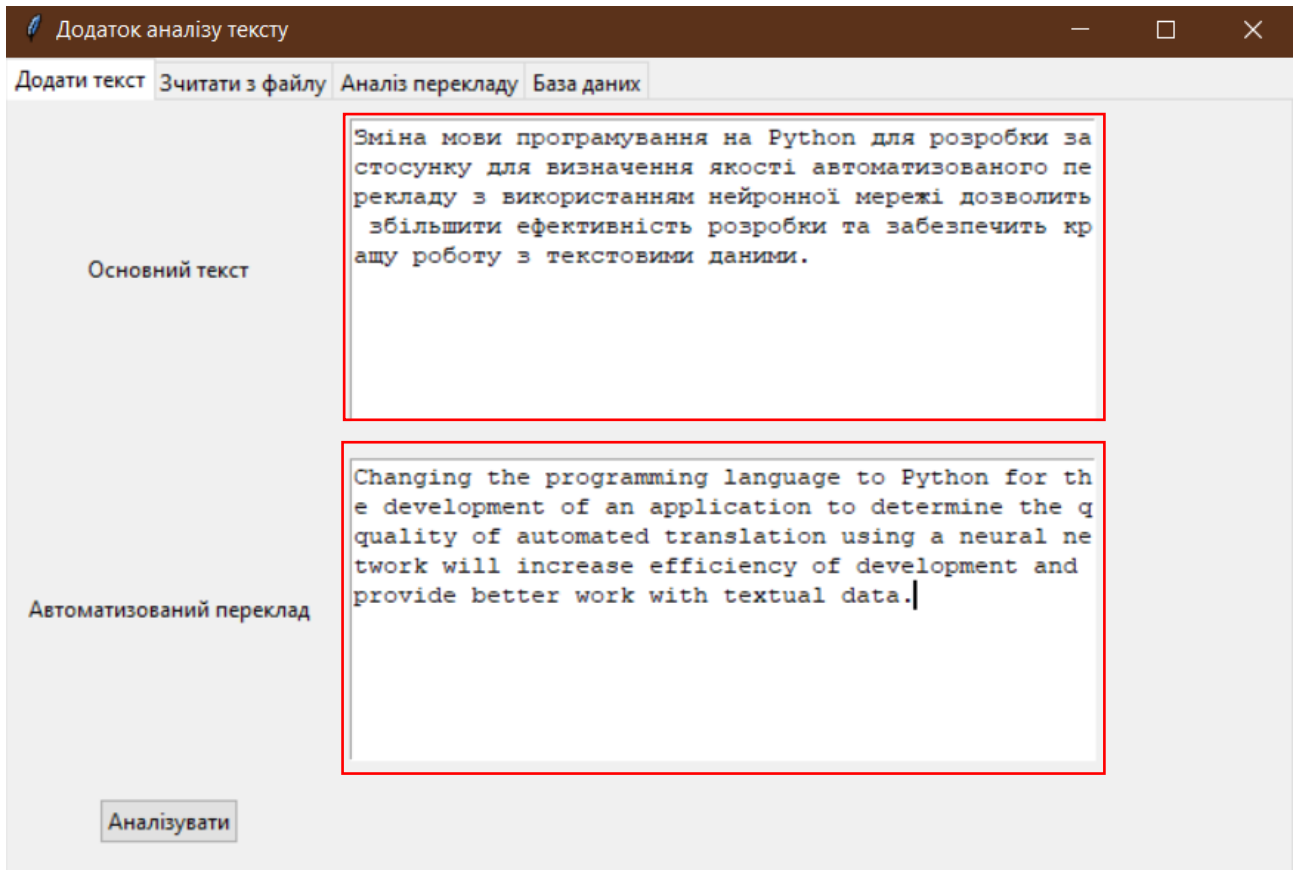


Рисунок 3.7 – Додавання текстів для аналізу

Перед тим як проаналізувати тексти, система визначає мови на яких вони написані. Якщо система виявляє, що мови введених текстів однакові, виводиться повідомлення про помилку (Рисунок 3.8). Повідомлення інформує користувача про те, що необхідно використовувати різні мови для проведення аналізу тексту. Ця перевірка необхідна для того щоб уникнути ситуації, коли користувач випадково вводить тексти однаковою мовою.

Для перевірки алгоритму визначення мови та виведення повідомлення про помилку, у випадку використання однієї мови, створено тест-кейс (Таблиця 3.1).

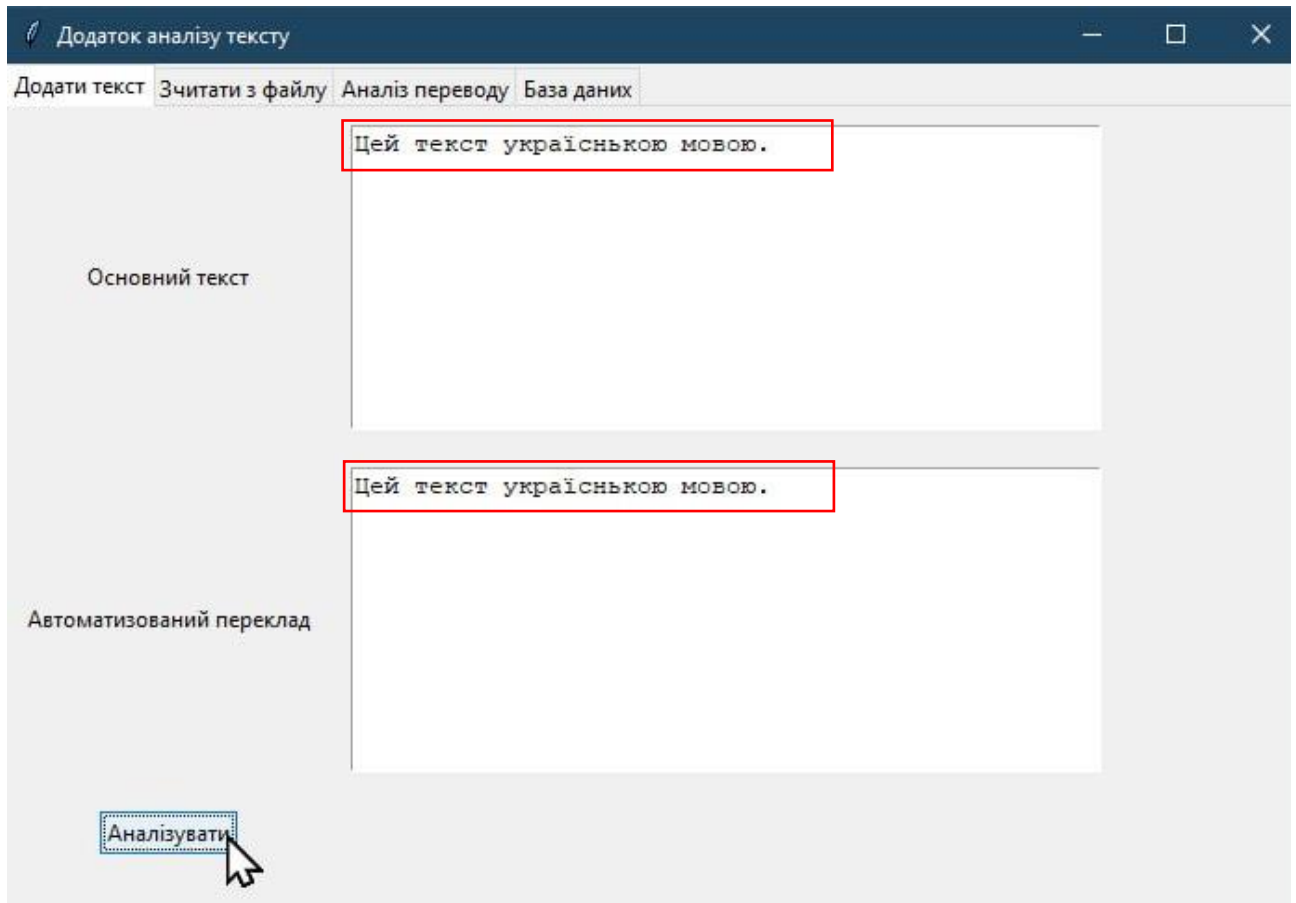


Рисунок 3.8 – Приклад введення текстів на одній мові

Отже, для проведення перевірки необхідно ввести в поля тексти написані на одній мові, наприклад, на українській. Після цього потрібно натиснути на кнопку «Аналізувати» та перейти на вкладку «Аналіз перекладу», де буде відображена помилка з відповідним текстом. Також, система правильно визначає мову на якій написані тексти (Рисунок 3.9). Таким чином, це не дає змоги системі аналізувати тексти, які не містять перекладу.

Таблиця 3.1 – Тест-кейс АТ0001

Тест-кейс ID: АТ0001	Пріоритет: 2	Створено: 18.05.2023, В. Одуха
Назва: Перевірка алгоритму визначення мови введених текстів		
Вхідні дані: Основний текст = «Цей текст українською мовою», Автоматизований переклад = «Цей текст українською мовою»		
Кроки	Очікуваний результат	
<i>Передумова:</i> Тексти мають бути написані однією мовою 1. Відкрити застосунок 2. Відкрити вкладку «Додати текст»	На вкладці «Аналіз перекладу» в полі відображається текст помилки: «Помилка, ви використали одну й ту саму мову: uk»	

<ol style="list-style-type: none"> 3. Ввести тексти в поля 4. Натиснути кнопку «Аналізувати» 5. Відкрити вкладку «Аналіз перекладу» 6. Порівняти фактичний результат з очікуваним 	
Результат виконання тест-кейсу: пройдено успішно	

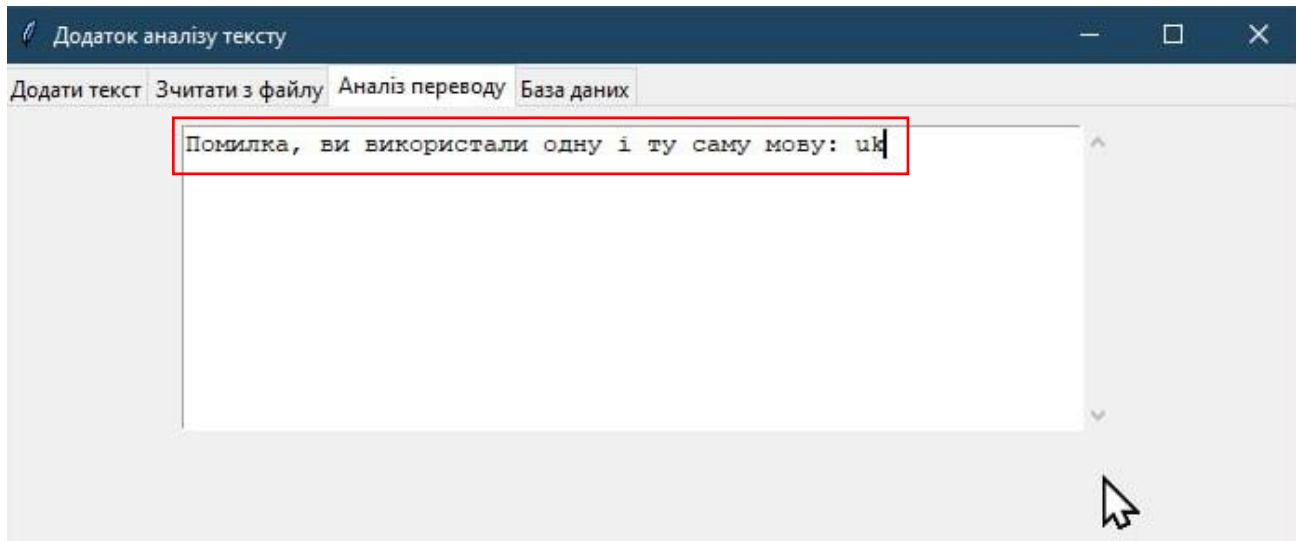


Рисунок 3.9 – Відображення помилки про використання однієї мови

При бажанні користувач може додати тексти використовуючи файловий провідник. Для цього необхідно створити два окремі текстові файли та зберегти на ПК. Далі, необхідно відкрити вкладку «Зчитати з файлу» та додати два файли у відповідні поля (Рисунок 3.10). Після того як файли додано, програма зможе їх прочитати та провести аналіз текстової інформації, збереженої у завантажених файлах.

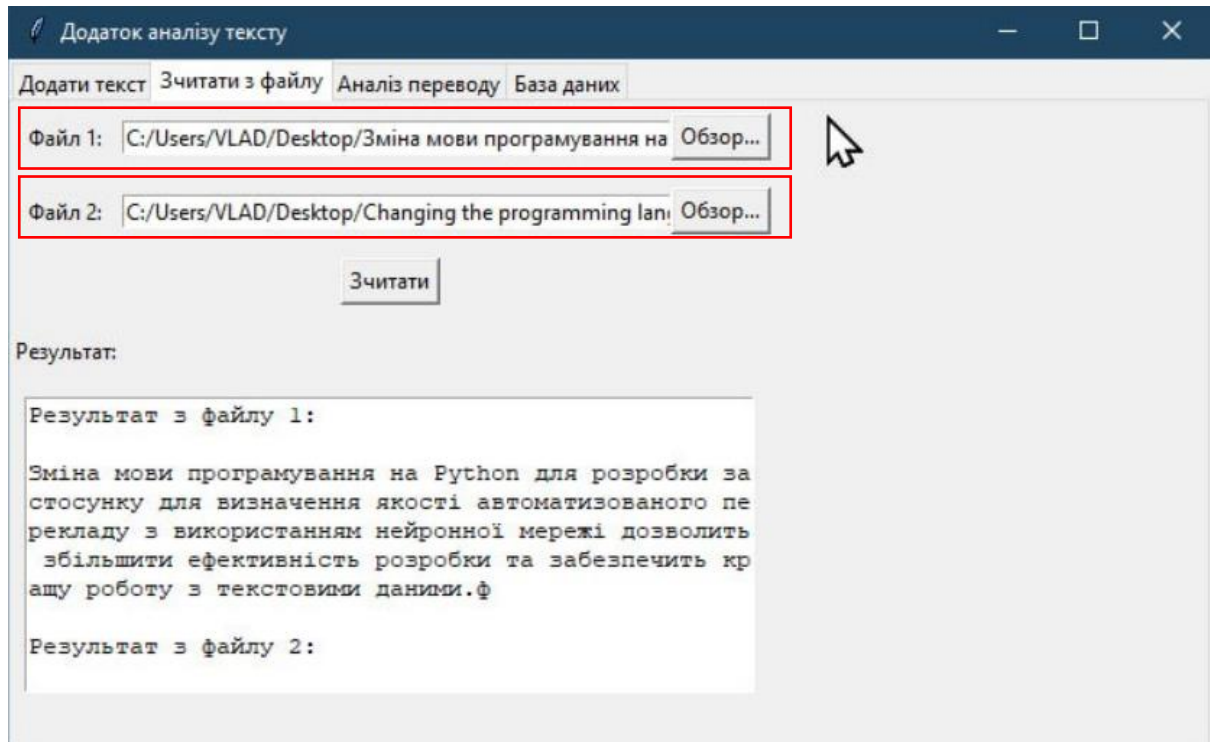


Рисунок 3.10 – Додавання текстових файлів через файловий провідник

При виникненні проблем із завантаженням файлів користувачу відобразиться нове вікно із помилкою (Рисунок 3.11). Для перевірки процесу додавання текстових файлів та відображення вікна з помилкою створено тест-кейс (Таблиця 3.2).

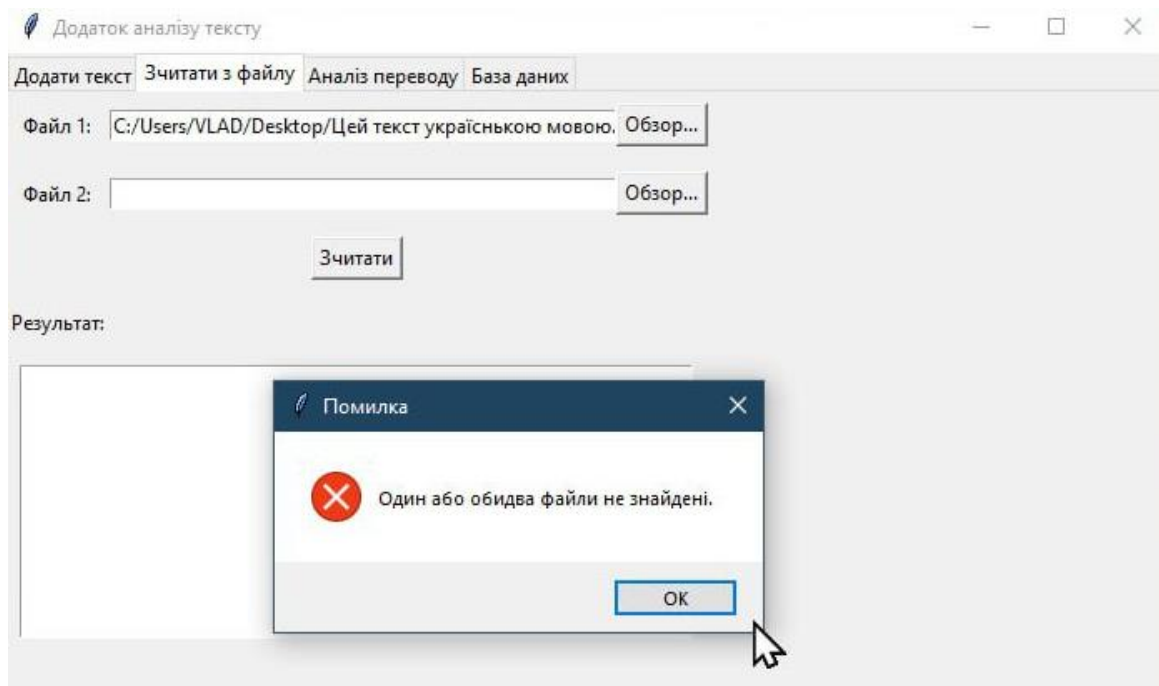


Рисунок 3.11 – Помилка при завантаженні файлів

Таблиця 3.2 – Тест-кейс АТ0002

Тест-кейс ID: АТ0002	Пріоритет: 2	Створено: 18.05.2023, В. Одуха
Назва: Перевірка процесу завантаження файлів через файловий провідник		
Вхідні дані: Текстові файли		
Кроки	Очікуваний результат	
<p><i>Передумова:</i> Користувач повинен завантажити один файл</p> <ol style="list-style-type: none"> 1. Відкрити застосунок 2. Відкрити вкладку «Зчитати з файлу» 3. Натиснути кнопку «Обзор» для вибору першого файлу 4. Вибрати текстовий файл 5. Натиснути кнопку «Зчитати» 6. Порівняти фактичний результат з очікуваним 	Відкрилось нове вікно із текстом помилки: «Один або обидва файли не знайдені»	
Результат виконання тест-кейсу: пройдено успішно		

Таким чином, виключається ситуація коли користувач намагається проаналізувати текст тільки з одного файлу, або з одним файлом є проблеми (файл відсутній або файл не правильного формату).

Коли користувач додасть до програми два тексти та додані тексти пройдуть усі попередні перевірки, користувач зможе переглянути результати аналізу. Для цього потрібно відкрити вкладку «Аналіз перекладу», де у відповідному полі буде відображено результат якості автоматизованого перекладу тексту (Рисунок 3.12).

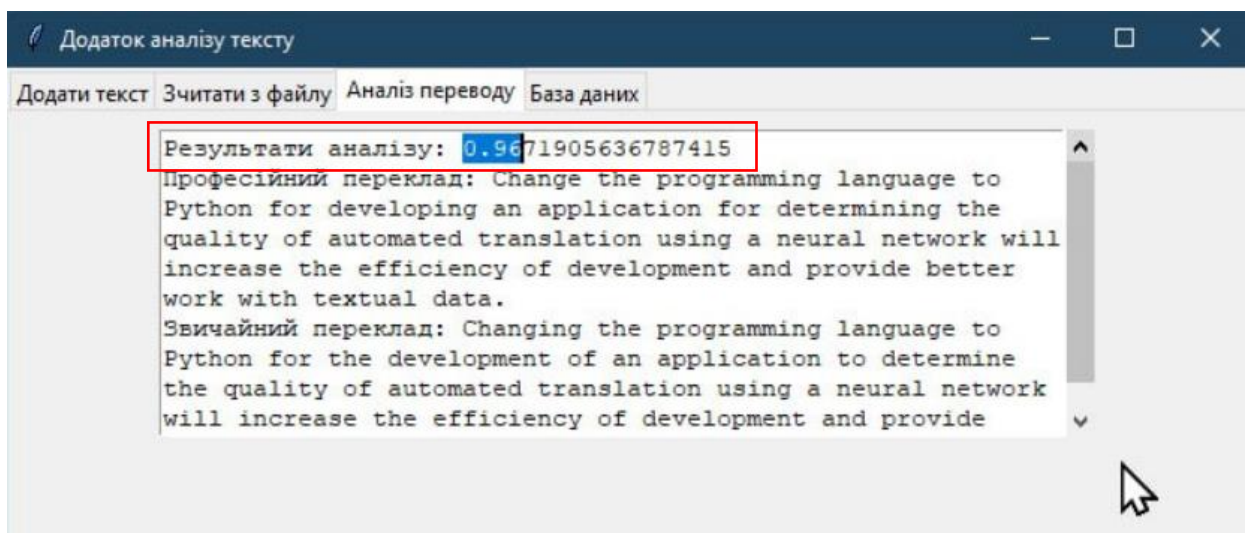


Рисунок 3.12 – Результати роботи застосунку

Результати аналізу текстів зберігаються до бази даних, тому користувач може у будь-який час переглянути усі попередні спроби визначення якості автоматизованого перекладу. Для цього потрібно перейти на вкладку «База даних» та ввести номер проведеного раніше аналізу текстів. Після цього, в полі нижче відобразяться тексти, які були проаналізовані та результати аналізу (Рисунок 3.13). Також, у БД зберігається інформація про те, якою мовою написані тексти.

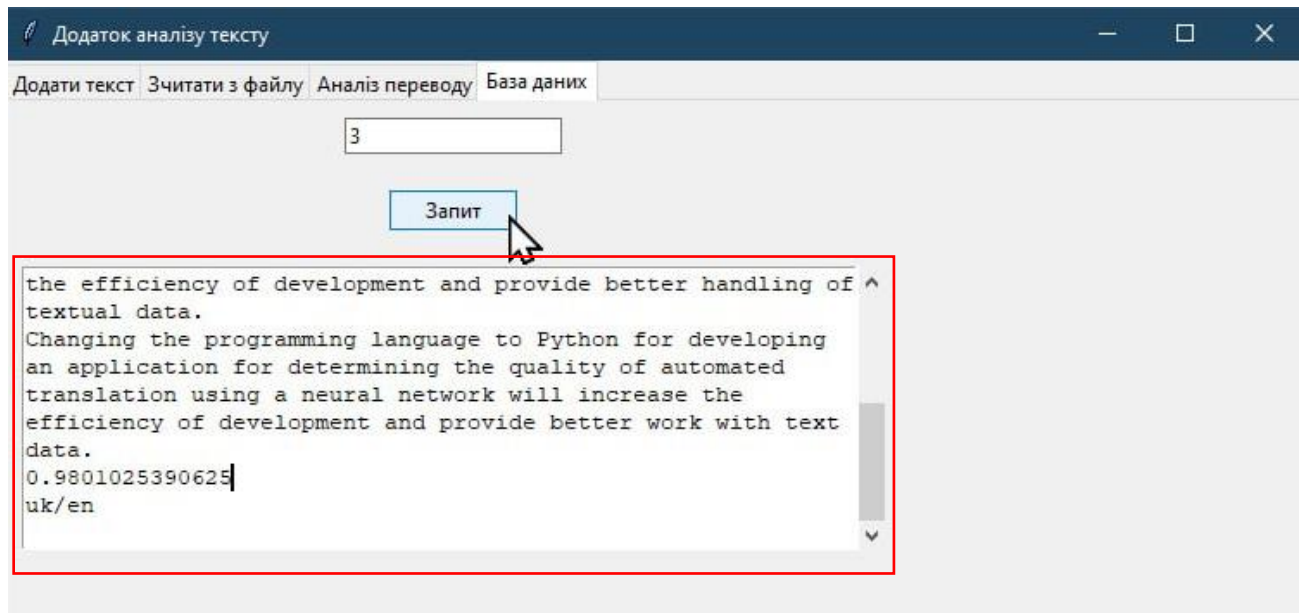


Рисунок 3.13 – Перегляд даних збережених в БД

Для перевірки алгоритму відображення даних із БД створено відповідний тест-кейс (Таблиця 3.3).

Таблиця 3.3 – Тест-кейс АТ0003

Тест-кейс ID: АТ0002	Пріоритет: 2	Створено: 18.05.2023, В. Одуха
Назва: Перевірка алгоритму відображення даних із БД		
Вхідні дані: ID запису = «3»		
Кроки	Очікуваний результат	
<i>Передумова:</i> Користувач повинен провести аналіз текстів інакше БД	В полі відобразились тексти, які були	

буде пустою 1. Відкрити застосунок 2. Відкрити вкладку «База даних» 3. Ввести ID запису в відповідне поле 4. Натиснути кнопку «Запит» 5. Порівняти фактичний результат з очікуваним	проаналізовані. Крім цього, відображаються результати визначення якості перекладу та мови, на яких написані тексти.
Результат виконання тест-кейсу: пройдено успішно	

Отже, користувач може проводити аналіз текстів необмежену кількість разів та зберігати результати у БД. Це дозволить користувачеві переглядати попередні результати аналізу текстів для визначення якості автоматизованого перекладу. Проведене тестування розробленого застосунку показує, що програма працює правильно та у разі виникнення помилок повідомляє про них користувачеві.

3.4 Висновки до розділу 3

Реалізований застосунок для визначення якості автоматизованого перекладу з використанням нейронної мережі пропонує користувачу зручний та зрозумілий графічний інтерфейс. Графічний інтерфейс користувача створено з використанням спеціальної бібліотеки tkinter, адже мова програмування Python більше орієнтована на створення командних інтерфейсів.

Застосунок передбачає можливість ручного введення текстів для аналізу або додавання текстів через файловий провідник. В обох випадках тексти проходять попередні перевірки, які включають визначення мови та перевірку відповідності файлового формату. У разі виникнення будь-яких помилок, програма сповіщає про це користувача.

Програмна реалізація способу визначення якості автоматизованого перекладу полягає у використанні нейронної мережі. Для цього використовується вже навчена нейронна мережа BERT, яку можна легко підключити в проєкт як сторонню бібліотеку та використовувати для власних потреб.

Отже, результатом прикладного застосування розробленого способу визначення якості автоматизованого перекладу з використанням нейронної мережі є програмний застосунок, який дозволяє порівнювати автоматизовані переклади та професійні переклади та визначати їхню подібність.

Висновки

Кваліфікаційна робота бакалавра присвячена розробці способу визначення якості автоматизованого перекладу з використанням нейронної мережі BERT та відповідної інформаційної системи у вигляді застосунку, що дозволяє за текстовим вмістом автоматизованого перекладу визначати відносну числову оцінку його якості.

Першим етапом виконання кваліфікаційної роботи бакалавра було проведення детального аналізу предметної області, пов'язаної із обробкою природної мови. Також, було розглянуто етапи створення автоматизованого перекладу та проблеми, які можуть виникнути при створенні перекладу з використанням інформаційних систем. В процесі проведеного дослідження було встановлено, що автоматизований переклад не завжди якісний та може мати ряд недоліків, тому визначення якості автоматизованого перекладу це важливе завдання.

Згідно з проведеним дослідженням було розроблено спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі. В якості нейронної мережі було прийнято рішення використати вже навчену нейронну мережу BERT. Використання нейронної мережі BERT забезпечує високу точність і надійність аналізу текстів, оскільки модель була натренована на великій кількості текстових даних. Це дозволяє отримати якісні результати в оцінці якості автоматизованого перекладу текстів і забезпечити користувачам надійність у роботі з програмою.

Відповідно до розробленого способу визначення якості автоматизованого перекладу було реалізовано застосунок на мові програмування Python. Застосунок дозволяє додавати тексти двома методами: ручне введення тексту в відповідні поля або завантаження текстових файлів. Результатом роботи застосунку є аналіз текстів та визначення якості автоматизованого перекладу, шляхом порівняння автоматизованого та професійного перекладів тексту.

Отриманий результат доводить, що мета та задачі кваліфікаційної роботи досягнуті. Проведене тестування розробленого застосунку показує, що інформаційна система працює вірно та надає користувачу результат визначення якості автоматизованого перекладу. Це означає, що розроблений спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі може бути застосований для отримання відносної числової оцінки якості перекладу.

Перелік посилань

1. Ethnologue. Languages of the World. URL: <https://www.ethnologue.com/>
2. Gengo. Machine translation. URL: <https://gengo.com/machine-translation/>
3. Вікіпедія. Автоматизований переклад. URL:
https://uk.wikipedia.org/wiki/Автоматизований_переклад
4. GTE Localize. What types of machine translation should you see? URL:
<https://gtelocalize.com/what-types-of-machine-translation-should-you-use/>
5. Освіта.ua. машинний переклад тексту: Переваги та недоліки URL:
<http://osvita.dream.net.ua/vnz/reports/journalism/25750/>
6. Phrase. Machine Translation: What it is, how it works, and when to use it.
URL: <https://phrase.com/blog/posts/machine-translation/>
7. Лещенко Г. В., Гречуха Л. О. Автоматизований переклад: проблеми та перспективи. Всеукраїнська науково-практична конференція «Романо-германські мови у контексті неперервної освіти», 2019, Черкаси. С 127-132
8. Вікіпедія. Багатозначність URL:
<https://uk.wikipedia.org/wiki/Багатозначність>
9. Вікіпедія. Якість. URL: <https://uk.wikipedia.org/wiki/Якість>
10. Мова Club. Стандарти перекладів – які критерії якісного перекладу?
URL: <https://mova-club.kiev.ua/byuro-perekladiv/novyny/132-standarty-perekladiv-iaki-kryterii-iakisnoho-perekladu>
11. Вікіпедія. Природна мова URL:
https://uk.wikipedia.org/wiki/Природна_мова
12. Енциклопедія сучасної України. Комп'ютерна лінгвістика. URL:
<https://esu.com.ua/article-4396>
13. Metinvest Digital. Що таке обробка природної мови (NLP)? URL:
<https://metinvest.digital/ua/page/1052>
14. Search Medium. BERT Explained: State of the art language model for NLP. URL: <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>

15. OpenAI. GPT-4 is OpenAI's most advanced system, producing safer and more useful responses. URL: <https://openai.com/product/gpt-4>
16. Digitaltrends. How to use Google Bard, the latest AI chatbot service. URL: <https://www.digitaltrends.com/computing/how-to-use-google-bard/>
17. Grammarly. URL: <https://app.grammarly.com/>
18. Xbench. URL: <https://www.xbench.net/>
19. QA Distiller. URL: <https://www.qa-distiller.com/en>
20. Вікіпедія. Вкладання слів URL:
https://uk.wikipedia.org/wiki/Вкладання_слів
21. Gensim. Word2vec embeddings. URL:
<https://radimrehurek.com/gensim/models/word2vec.html>
22. Вікіпедія. Векторна модель URL:
https://uk.wikipedia.org/wiki/Векторна_модель
23. Devlin J., Chang M., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Computation and Language. Cornell University, 2019. С 1-16
24. Вікіпедія. Комп'ютерна термінологія. URL:
https://uk.wikipedia.org/wiki/Комп'ютерна_термінологія
25. EPAM. Python: матеріали для самопідготовки. URL:
<https://training.epam.ua/News/Items/240?lang=ua>
26. Microsoft. Документація по Python. URL:
<https://learn.microsoft.com/ru-ru/windows/python/>
27. OpenAI. ChatGPT. URL: <https://openai.com/blog/chatgpt>

ДОДАТКИ

Додаток А

Програмні коди

Лістинг файлу main.py

```

import chatgpt_prompt as api
import bert_quality_assesment as bert
import database as db
from langdetect import detect
import tkinter as tk
from tkinter import ttk, scrolledtext, filedialog

class App(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Додаток аналізу тексту")
        self.geometry("700x500") # Задаємо розмір вікна

        # Створення вкладок
        self.notebook = ttk.Notebook(self)
        self.add_text_tab = ttk.Frame(self.notebook)
        self.read_from_file = ttk.Frame(self.notebook)
        self.results_tab = ttk.Frame(self.notebook)
        self.analysis_data_tab = ttk.Frame(self.notebook)

        # Додавання вкладок до ноутбука
        self.notebook.add(self.add_text_tab, text="Додати текст")
        self.notebook.add(self.read_from_file, text="Зчитати з файлу")
        self.notebook.add(self.results_tab, text="Аналіз перекладу")
        self.notebook.add(self.analysis_data_tab, text="База даних")
        self.notebook.pack(expand=True, fill=tk.BOTH)

        # Кнопка додавання тексту
        self.add_text_button = ttk.Button(self.add_text_tab, text="Аналізувати", command=self.add_text)
        self.add_text_button.grid(row=2, column=0, padx=10, pady=10)

        # Поля "Обрати" для зчитування даних з файлів

        self.file1_label = tk.Label(self.read_from_file, text="Файл 1:")
        self.file1_label.grid(row=0, column=0, pady=10)

        self.file1_path = tk.StringVar()
        self.file1_entry = tk.Entry(self.read_from_file, textvariable=self.file1_path, width=50)
        self.file1_entry.grid(row=0, column=1, pady=5)

        self.browse1_button = tk.Button(self.read_from_file, text="Обрати...", command=self.browse_file1)
        self.browse1_button.grid(row=0, column=2, pady=5)

        self.file2_label = tk.Label(self.read_from_file, text="Файл 2:")
        self.file2_label.grid(row=1, column=0, pady=10)

        self.file2_path = tk.StringVar()
        self.file2_entry = tk.Entry(self.read_from_file, textvariable=self.file2_path, width=50)
        self.file2_entry.grid(row=1, column=1, pady=5)

        self.browse2_button = tk.Button(self.read_from_file, text="Обрати...", command=self.browse_file2)
        self.browse2_button.grid(row=1, column=2, pady=5)

        self.read_button = tk.Button(self.read_from_file, text="Зчитати", command=self.read_files)
        self.read_button.grid(row=2, column=0, columnspan=3, pady=5)

        self.data_label = tk.Label(self.read_from_file, text="Результат:")
        self.data_label.grid(row=3, column=0, pady=10)

        self.data_text = tk.Text(self.read_from_file, height=10, width=50)
        self.data_text.grid(row=4, column=0, columnspan=3, pady=5)

        # Поля для введення тексту
        self.text_label1 = ttk.Label(self.add_text_tab, text="Основний текст")
        self.text_label1.grid(row=0, column=0, padx=10, pady=10)
        self.text_entry1 = tk.Text(self.add_text_tab, height=10, width=50)
        self.text_entry1.grid(row=0, column=1, padx=10, pady=10)

        self.text_label2 = ttk.Label(self.add_text_tab, text="Автоматизований переклад")
        self.text_label2.grid(row=1, column=0, padx=10, pady=10)
        self.text_entry2 = tk.Text(self.add_text_tab, height=10, width=50)
        self.text_entry2.grid(row=1, column=1, padx=10, pady=10)

        # Мітка для відображення результатів
        self.results_label = scrolledtext.ScrolledText(self.results_tab, wrap="word", width=60, height=10)
        self.results_label.pack(padx=10, pady=10)

        # Введення PRIMARY KEY щоб отримати результат з бд
        self.text_key_db = tk.Entry(self.analysis_data_tab)

```

```

self.text_key_db.grid(row=0, column=0, padx=10, pady=10)

self.button_db = ttk.Button(self.analysis_data_tab, text="Занит", command=self.get_data_db)
self.button_db.grid(row=1, column=0, padx=10, pady=10)
self.text_db_label = scrolledtext.ScrolledText(self.analysis_data_tab, wrap="word", width=60, height=10)
self.text_db_label.grid(row=2, column=0, padx=10, pady=10)

# Перегляд файлу
def browse_file1(self):
    file_path = filedialog.askopenfilename(filetypes=[("Text Files", "*.txt"), ("All Files", "*.*")])
    self.file1_path.set(file_path)

def browse_file2(self):
    file_path = filedialog.askopenfilename(filetypes=[("Text Files", "*.txt"), ("All Files", "*.*")])
    self.file2_path.set(file_path)

# зчитати дані
def read_files(self):
    file1_path = self.file1_path.get()
    file2_path = self.file2_path.get()

    try:
        with open(file1_path, 'r', encoding='utf-8') as file1, open(file2_path, 'r') as file2:
            data1 = file1.read()
            data2 = file2.read()

            self.data_text.delete('1.0', tk.END)
            self.data_text.insert(tk.END, "Результат з файлу 1:\n\n")
            self.data_text.insert(tk.END, data1)
            self.data_text.insert(tk.END, "\n\nРезультат з файлу 2:\n\n")
            self.data_text.insert(tk.END, data2)
            self.add_text(data1, data2)
    except FileNotFoundError:
        tk.messagebox.showerror("Помилка", "Один або обидва файли не знайдені.")

# отримати дані з бд
def get_data_db(self):
    key = self.text_key_db.get()
    list = []
    list = db.get_data(key)
    print(list)
    # Очищаем содержимое scrolledtext перед вставкой новых данных
    self.text_db_label.delete("1.0", "end")

    for item in list:
        self.text_db_label.insert("end", str(item) + "\n")

# визначення мови перекладу
def detect_language(self, text):
    language = detect(text)
    if language in ['uk', 'en']:
        return language
    else:
        return None

def add_text(self, native, automate_translate):
    native_text = ""
    automate_translate_text = ""
    if len(native.strip()) != 0 and len(automate_translate.strip()) != 0:
        native_text = native
        automate_translate_text = automate_translate
    else:
        # Получение введенного текста
        native_text = self.text_entry1.get("1.0", tk.END).strip()
        automate_translate_text = self.text_entry2.get("1.0", tk.END).strip()

    # Проверка текста
    language_form_one = self.detect_language(native_text)
    language_form_two = self.detect_language(automate_translate_text)

    # Очищаем содержимое scrolledtext перед вставкой новых данных
    self.text_db_label.delete("1.0", tk.END)

    if language_form_one is None or language_form_two is None:
        temp = "Ви використали невідому мову " + str(language_form_one) + " " + str(
            language_form_two)
        self.results_label.insert(tk.END, temp)
        return

    if language_form_one == language_form_two:
        temp = "Помилка, ви використали одну і ту саму мову: " + str(language_form_one)
        self.results_label.insert(tk.END, temp)
        return

    # Анализ текста
    professional_translate = ""

    if language_form_one == "uk":

```

```

professional_translate = api.generate_text(
    "Make a translation of the text from Ukrainian to English, you only need a translation, "
    "without unnecessary text and explanations - " + native_text)

if language_form_one == "en":
    professional_translate = api.generate_text(
        "Make a translation of the text from English to Ukrainian, you only need a translation, "
        "without unnecessary text and explanations - " + native_text)

similarity_score = bert.execute_analysis(professional_translate, automate_translate_text)
# Обновление метки с результатами
temp = "Результати аналізу: " + str(
    similarity_score) + "\nПрофесійний переклад: " + professional_translate + "\nЗвичайний переклад: " +
automate_translate_text
self.results_label.insert(tk.END, temp)

# Добавление данных в базу данных
abbreviation = language_form_one + "/" + language_form_two
db.database_call(native_text, automate_translate_text, professional_translate, similarity_score, abbreviation)

if __name__ == "__main__":
    app = App()
    app.mainloop()

```

Лістинг файлу database.py

```

import sqlite3

def database_call(native, automate, professional, score, abbreviation):
    # Підключення до бази даних (якщо база даних не існує, вона буде створена)
    conn = sqlite3.connect('mydatabase.db')

    # Створення курсора
    cursor = conn.cursor()

    # Виконання SQL-запиту для створення таблиці
    cursor.execute(
        'CREATE TABLE IF NOT EXISTS translator (id INTEGER PRIMARY KEY, native_text TEXT, automate_translation , '
        'professional_translation TEXT, quality_score REAL, abbreviation TEXT)')

    # Виконання SQL-запиту для вставки даних
    cursor.execute(
        'INSERT INTO translator (native_text, automate_translation, professional_translation, quality_score, abbreviation) '
        'VALUES (?, ?, ?, ?, ?)', (native, automate, professional, score, abbreviation))

    # Застосування змін до бази даних
    conn.commit()

    # Закриття з'єднання
    conn.close()

def get_data(key):
    # Підключення до бази даних
    conn = sqlite3.connect('mydatabase.db')

    # Створення курсора
    cursor = conn.cursor()

    # Виконання SQL-запиту для читання даних за PRIMARY KEY
    cursor.execute('SELECT * FROM translator WHERE id = ?', (key,))

    # Отримання одного запису з результату запиту
    row = cursor.fetchone()

    # Перевірка, чи знайдено запис
    if row:
        # Закриття з'єднання
        conn.close()
        return row
    else:
        # Закриття з'єднання
        conn.close()
        return "Error."

```

Лістинг файлу chatgpt_prompt.py

```

import openai

# KEY OpenAI
openai.api_key = "sk-vfncK42IewyM34tA5rguT3B1bkFJZPHcn10o4p1SFNhFyu2y"

def generate_text(prompt):
    response = openai.Completion.create(
        engine="text-davinci-002",
        prompt=prompt,
        max_tokens=1024,
        n=1,
        stop=None,
        temperature=0.5,
    )

    message = response.choices[0].text.strip()
    return message

```

Лістинг файлу bert_quality_assesment.py

```

import torch
from transformers import BertTokenizer, BertModel
from scipy.spatial.distance import cosine

def execute_analysis(source_text, target_text):
    # Завантаження BERT-енкодера та токенайзера
    tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-cased')
    model = BertModel.from_pretrained('bert-base-multilingual-cased')

    # Функція для отримання векторного представлення тексту
    def get_bert_embedding(text):
        input_ids = torch.tensor([tokenizer.encode(text, add_special_tokens=True)])
        with torch.no_grad():
            last_hidden_states = model(input_ids)[0]
        return torch.mean(last_hidden_states, dim=1).squeeze().numpy()

    # Функція для обчислення схожості між текстами на основі BERT-енкодера та косинусної близькості
    def compute_similarity(source_text, target_text):
        source_embedding = get_bert_embedding(source_text)
        target_embedding = get_bert_embedding(target_text)
        similarity = 1 - cosine(source_embedding, target_embedding)
        return similarity

    # Приклад порівняння векторних представлень текстів за допомогою косинусної схожості
    similarity = compute_similarity(source_text, target_text)
    print("Similarity score:", similarity)
    return similarity

```

Додаток Б

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

СПОСІБ ВИЗНАЧЕННЯ ЯКОСТІ АВТОМАТИЗОВАНОГО ПЕРЕКЛАДУ З
ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ

Виконав: студент 4 курсу групи КН-19-1 Одуха Владіслав Станіславович
Керівник: доцент кафедри КН Манзюк Едуард Андрійович

АКТУАЛЬНІСТЬ



Питання якості перекладу залишається актуальним, оскільки автоматичний переклад часто містить помилки, неправильно передає відтінки сенсу та інші неточності. Визначення якості перекладу є важливим завданням для того, щоб забезпечити його використання в реальних ситуаціях.



ЗАВДАННЯ

Кваліфікаційна робота бакалавра присвячена розробці способу визначення якості автоматизованого перекладу з використанням нейронної мережі та відповідної інформаційної системи у вигляді застосунку, що дозволяє за текстовим вмістом автоматизованого перекладу визначати відносну числову оцінку його якості.

Завдання №1

Проведення детального аналізу предметної області, пов'язаної із обробкою природної мови, створенням автоматизованого перекладу та проблемами, які з цим пов'язані.

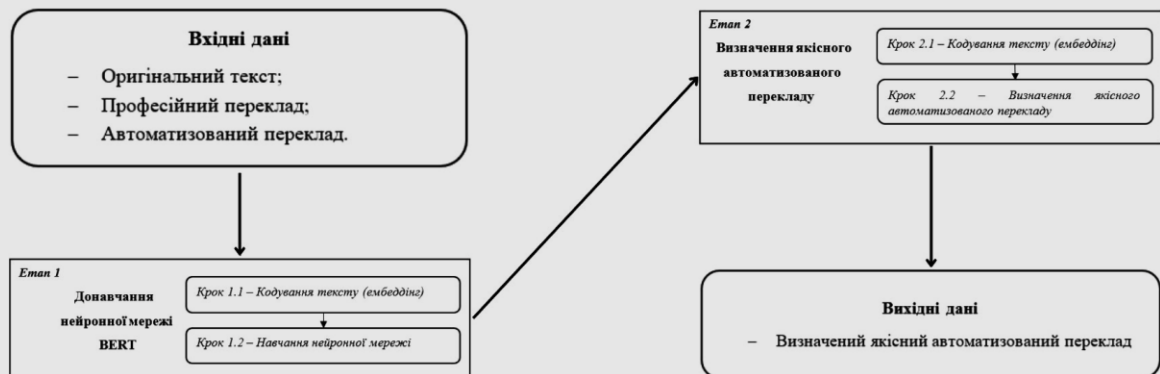
Завдання №2

Розробка способу визначення якості автоматизованого перекладу з використанням нейронної мережі

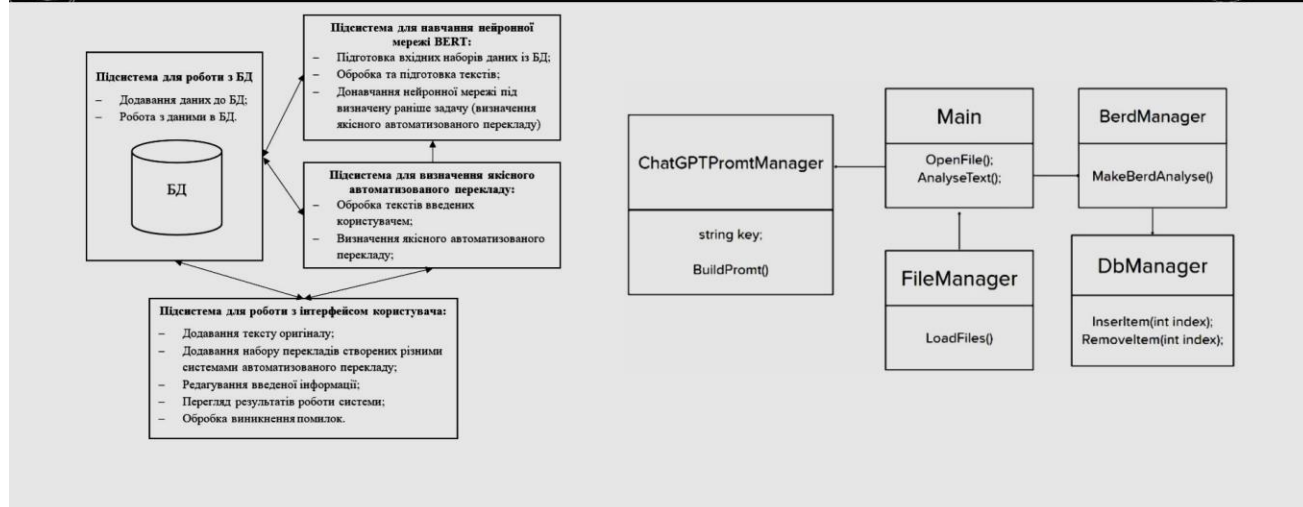
Завдання №3

Програмна реалізація застосунку для визначення якості автоматизованого перекладу з використанням нейронної мережі, відповідно до розробленого способу.

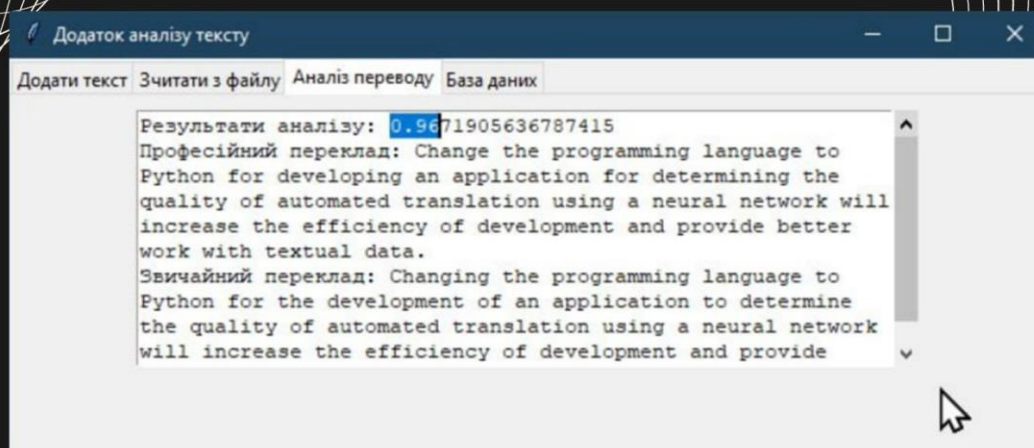
СХЕМА СПОСОБУ ВИЗНАЧЕННЯ ЯКОСТІ АВТОМАТИЗОВАНОГО ПЕРЕКЛАДУ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ



СТРУКТУРА КОМПОНЕНТІВ СПОСОБУ ВИЗНАЧЕННЯ ЯКОСТІ АВТОМАТИЗОВАНОГО ПЕРЕКЛАДУ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ



РЕЗУЛЬТАТИ РОБОТИ ЗАСТОСУНКУ ДЛЯ ВИЗНАЧЕННЯ ЯКОСТІ АВТОМАТИЗОВАНОГО ПЕРЕКЛАДУ



ВИСНОВКИ

- 01** ПРОВЕДЕНО ДЕТАЛЬНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ
- 02** ВИЗНАЧЕНО АКТУАЛЬНІСТЬ ДАНОЇ ТЕМИ
- 03** РОЗРОБЛЕНО СПОСІБ ВИЗНАЧЕННЯ ЯКОСТІ АВТОМАТИЗОВАНОГО ПЕРЕКЛАДУ
- 04** СПРОЄКТОВАНО СТРУКТУРУ КОМПОНЕНТІВ РОЗРОБЛЕНОГО СПОСОБУ
- 05** РЕАЛІЗОВАНО ЗАСТОСУНОК ДЛЯ ВИЗНАЧЕННЯ ЯКОСТІ ПЕРЕКЛАДУ
- 06** ПРОВЕДЕНО ТЕСТУВАННЯ РЕАЛІЗОВАНОГО ЗАСТОСУНКУ

**ДЯКУЮ ЗА
УВАГУ**

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 3.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 9%**

ID: 114182 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Додано в БД: 2023-05-28 Автора: В.С. Одуха Керівники: Е.А. Манзюк Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	63392	983	2837 (4%)	37 (4%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:
Кафедра КН

ID перевірки:
1015291598

Дата перевірки:
28.05.2023 17:28:38 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
28.05.2023 17:30:18 EEST

ID користувача:
100005671

Назва документа: КН-19-1 Одуха

Кількість сторінок: 63 Кількість слів: 10250 Кількість символів: 79386 Розмір файлу: 1.69 MB ID файлу: 1014963757

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

3.87% Схожість

Найбільша схожість: 1.53% з джерелом з Бібліотеки (ID файлу: 1009540885)

3.49% Джерела з Інтернету 272 Сторінка 65

2.4% Джерела з Бібліотеки 99 Сторінка 67

1.38% Цитат

Цитати 7 Сторінка 68

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування 14 сторінок

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі

Автор: студент групи КН-19-1 Одуха Владислав Станіславович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: д.т.н., доцент Манзюк Е.А.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<i>відповідає</i>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі Одухи В.С., не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції; серед запозичень знаходяться загальновідомі терміни, скорочення.

Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:

- за системою Anti-Plagiarism: 3%;

- за системою Unichек: 3.87%.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості є допустимим.

Керівник роботи



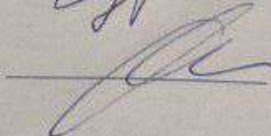
Едуард МАНЗЮК

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ



Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студента *гр. КН-19-1* *Одуха Владислава Станіславовича*

за темою Спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі

1. Актуальність теми

Розвиток технологій автоматизованого перекладу супроводжується розробкою методів, які дозволяють вдосконалити якість перекладу та виводять його на рівень, при якому важко відрізнити від перекладу здійсненого фахівцями в цій галузі. Проте на сьогодні такі системи ще мають аспекти вдосконалення та наявний досить широкий спектр автоматизованих систем. Результати їхньої роботи різняться залежно від прикладної тематики перекладу, стилю викладення інформації та інших аспектів. Тому важливим та актуальним завданням є розробка систем оцінювання якості перекладу автоматизованими системами, яка б дала змогу оцінити якісні параметри реалізованого перекладу. Зазначене завдання є актуальною задачею комп'ютерних наук.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

За стандартом, а саме описом предметної області, об'єктами вивчення та діяльності є математичні, інформаційні, імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи і технології отримання, зберігання, обробки, передачі та використання інформації. Мета роботи полягає в розробці й програмній реалізації способу визначення якості автоматизованого перекладу з використанням нейронної мережі. При вирішенні поставленої задачі використано математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач, які виникли при реалізації мети роботи. Результати кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

При роботі над кваліфікаційною роботою бакалавра *Одуха Владислав Станіславович* продемонстрував належні знання та вміння, своєчасно реалізовував поставлені завдання на відповідних етапах дослідження. Під час написання пояснювальної записки, розробки прикладного програмного забезпечення продемонстрував

наявні компетентності та результати навчання. Опанував професійні вміння за напрямком «Комп'ютерні науки».

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Одержані в роботі результати отримані студентом особисто та реалізовано усі поставлені задачі. Отримані результати підтверджують ефективність розроблених методів.

5. Ступінь оволодіння методами дослідження

При реалізації кваліфікаційної роботи показав належний рівень компетентностей та володіння необхідними методами, методиками та технологіями предметної області комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Тема роботи повною мірою обґрунтована та розкрита належним чином. Проведено аналіз відомих досліджень та актуальності теми кваліфікаційної роботи. Поставлені завдання реалізовано та розроблено програмне забезпечення для апробації запропонованого метода.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Записка до кваліфікаційної роботи за структурою, логічністю викладення матеріалу, аргументованістю, структурою послідовного викладення необхідною мірою відповідає стандартам.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Розроблений у роботі спосіб може бути використаний при оцінці якості автоматизованого перекладу у практичних задачах застосування з орієнтацією на специфіку предметної області, де застосовується переклад.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи належний рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник



д.т.н., доцент каф. КН Едуард МАНЗЮК



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-19-1 Одуха Владислав Станіславович
за темою: Спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі

1. Актуальність обраної теми

Питання якості перекладу залишається актуальним, оскільки автоматизований переклад часто містить помилки, неправильно передає відтінки смислу та інші неточності. Визначення якості перекладу є важливим завданням для того, щоб забезпечити його використання в реальних ситуаціях.

2. Повнота розкриття мети та завдань роботи

Протягом виконання кваліфікаційної роботи бакалавра мету та завдання роботи було розкрито у повній мірі, оскільки було проведено детальний аналіз предметної області, описано кроки способу та реалізовано його прикладне застосування у вигляді програмного застосунку.

3. Зміст кожного розділу роботи

Записка кваліфікаційної роботи бакалавра містить три розділи. У першому розділі проведено аналіз предметної області та встановлено актуальність обраної теми. У другому розділі описано кроки способу визначення якості автоматизованого перекладу з використанням нейронної мережі. У третьому розділі описано процес реалізації прикладного застосування розробленого методу у вигляді застосунку для аналізу перекладу.

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблений спосіб визначення якості автоматизованого перекладу з використанням нейронної мережі може бути застосований для отримання відносної числової оцінки якості автоматизованого перекладу.

5. Якість оформлення кваліфікаційної роботи бакалавра

Записка оформлена відповідно до усіх вимог та правил. Викладення матеріалу логічне та послідовне.

6. Недоліки кваліфікаційної роботи бакалавра

Рекомендовано вдосконалити програмну реалізацію додавши можливість аналізу декількох автоматизованих перекладів одного тексту та вибору якіснішого перекладу серед запропонованих варіантів.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент

доц. каф. ТМІТ (ХНУ) Олег Публицький