МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наук
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження архітектурних рішень для поліпшення пошуку слів з іноземної мови

Виконав:
студент 2 курсу групи ІПЗм-21-1
Овчаренко Д. Є.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення

Тип програми Освітньо-наукова

Керівник Ревенчук І. А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. Кафедри _____ проф. Дудар З.В
(підпис)

Харків 2023

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної Інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121- Інженерія програмного забезпечення
(код і повна назва)

Тип програми освітньо-наукова

Освітня програми Програмна Інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Керівник _____
(підпис)
«___»_____20__р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студента Овчаренка Данила Євгеновича
(прізвище, ім'я, по батькові)

1. Тема роботи: дослідження архітектурних рішень для поліпшення пошуку слів з іноземної мови
   Затверджена наказом університету від «_29_»____березня_____ 2023 р. №_302 Ст_
2. Термін подання студентом роботи до екзаменаційної комісії 1 9 т р а в н я 2023 р.
3. Вихідні дані до роботи встановлений календарний план роботи, методичні вказівки до оформлення пояснювальної записки, методи прогнозування засновані на методах рекомендаційних алгоритмів.
4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз предметної галузі і постановка задачі, огляд наявних математичних моделей, модифікація базових алгоритмів, дослідження можливості прискорення базових моделей, створення плану для подальшого дослідження теми.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	03.04.2023	виконано
2	Дослідження алгоритмів для поліпшення пошуку слів	18.04.2023	виконано
3	Здійснення огляду математичних моделей	19.04.2023	виконано
4	Формування вимог	21.04.2023	виконано
5	Розробка та тестування алгоритмів	29.04.2023	виконано
6	Оформлення пояснювальної записки	01.05.2023	виконано
7	Підготовка доповіді та презентації	03.05.2023	виконано
8	Перевірка роботи на плагіат	05.05.2023	виконано
9	Проведення нормоконтролю роботи	07.05.2023	виконано
10	Рецензування роботи	09.05.2023	виконано
11	Занесення роботи в електронний архів	11.05.2023	виконано
12	Попередній захист кваліфікаційної роботи	10.05.2023	виконано
13	Допуск до захисту роботи зав. кафедри	12.05.2023	виконано
14	Захист кваліфікаційної роботи	19.05.2023	виконано

Дата видачі завдання 3 квітня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Ревенчук І. А
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота містить: 69 сторінок, 2 таблиці, 13 рисунків, 14 джерел.

АЛГОРИТМИ, АНАЛІЗ ДАНИХ, БІБЛІОТЕКИ, МАШИННЕ НАВЧАННЯ, СЛОВА, ПЕРЕКЛАД, НЕЙРОННА МЕРЕЖА, ЕФЕКТИВНІСТЬ АЛГОРИТМУ

Об’єктом дослідження є архітектурні рішення для поліпшення пошуку слів з іноземної мови.

Метою роботи є проведення дослідження щодо ефективності архітектурних рішень для поліпшення пошуку слів з іноземної мови, з англійської на українську мови.

У результаті роботи було проведене дослідження архітектурних рішень для поліпшення пошуку слів з іноземної мови та була розроблена WEB-орієнтована програмна система.

ALGORITHMS, DATA ANALYSIS, LIBRARIES, MACHINE LEARNING, WORDS, TRANSLATION, NEURAL NETWORK, ALGORITHM PERFORMANCE

The object of the research is architectural solutions for improving the search for words from a foreign language.

The purpose of the work is to conduct a study on the effectiveness of architectural solutions to improve the search for words from a foreign language, English into Ukrainian language.

As a result of the work, a study of architectural solutions for improving the search for words from a foreign language was conducted and a WEB-oriented software system was developed.

Умови публікації пояснювальної записки

Я, Овчаренко Данило Євгенович,
(прізвище, ім’я, по батькові)

студент(ка) групи ІПЗм-21-1, здобувач вищої освіти на другому (магістерському) рівні,

кафедри Програмної інженерії
(назва кафедри)

заявляю: моя кваліфікаційна робота на тему «Дослідження архітектурних рішень для поліпшення пошуку слів з іноземної мови»
(назва роботи)

що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

В еру цифрових технологій, для людства вже не новина мати із собою всі знання людства у кишені чи портфелі. Ще декілька років тому, було важко уявити наскільки людство вийде вперед у своєму розвитку. Ера індустріалізації вже не здається такою швидкою та значущою як наша ера.

Серед різноманіття інновацій і технологій, предмет моєї магістерської роботи – це дослідження щодо ефективності архітектурних рішень для поліпшення пошуку слів з англійської мови.

Зараз існує дуже багато перекладачів та словників, наприклад: Google Translate, Microsoft Translate, Amazon Translate та не так давно піднявши хвилю емоцій та великого інтересу серед дослідників з багатьох галузей, від лінгвістів до програмістів – перекладач Цукерберга Meta AI Speech Translator, який переводить голос у текст та перекладає його.

Всі ці технології неймовірні та швидкі. Вони дають змогу будь-кому, легко та без зайвих коливань використовувати їх у навчанні, у подорожах або у дослідницьких цілях. Але як же вони працюють? Що під капотом цього дуже простого перекладача? На перший погляд, може здаватися, що це дуже просто, береш слово та по ньому шукаєш тлумачення або переклад. За цим принципом можна використовувати й паперові варіанти словників, які теж, ефективно і точно, хоча довго, видають нам результат пошуку у друкованому вигляді.

Насправді всередині цього чорного ящику є так звані алгоритми. Ці алгоритми і є серцем та окремою ознакою кожного окремого перекладача. Саме характеристики алгоритму визначають точність і швидкість перекладу, необхідні ресурси для функціонування такої системи, а отже і ціну за користування та зрештою популярність системи.

Я проаналізую вже наявні алгоритми пошуку слів з англійської мови як із залученням штучного інтелекту та машинного навчання, так і без них. Також розгляну мій алгоритм, який полегшує пошук слів та робить користувацький досвід унікальним та цікавим.

У цій магістерській роботі, я буду розглядати популярні алгоритми пошуку слів з англійської мови. Наприклад я буду досліджувати алгоритм найпопулярнішого, але далеко не найкращого перекладача Google translate, також не залишу без уваги алгоритм перекладача Amazon Translate, який є прямим конкурентом Google Translate перекладача, але використовує зовсім інший алгоритм для пошуку слів. І нарешті, я опишу та розберу алгоритм зараз популярного та інноваційного програмного продукту від Марка Цукерберга Meta AI Speech Translator.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Огляд стану розв’язання проблеми

Зараз існує дуже багато інструментів для перекладу слів з однієї мови на іншу. Кожен інструмент має свої переваги та недоліки. Зазвичай, прості перекладачі використовують примітивний алгоритм всередині «чорного ящику», тому результати перекладу мають відповідну якість. Більш складні перекладачі можуть перекладати не тільки окремі слова, а ще й цілі словосполучення за допомогою складних алгоритмів, які засновані на машинному навчанні та штучному інтелекті.

Прості перекладачі мають досить обмежений функціонал та невеликі обсяги бази даних зі словами, також, зазвичай, такі програмні продукти не мають широкої підтримки різноманіття мов перекладу. Такі типи перекладачів просто шукають слово за допомогою регулярних виразів та індексів для бази даних для удосконалення пошуку та пришвидшення відповіді БД. Ми набираємо частину або ціле слово у пошуку, програма шукає все зв’язані слова у БД та виводить нам один або декілька перекладів слова з відповідними прикладами вживання, наприклад як Reverso Context, який є потужним електронним перекладачем, але націлений більше на переклад окремих слів ніж словосполучень.

Перекладачі, які використовують у якості ядра машинне навчання є складними та зазвичай багатофункціональними програмними продуктами, які можуть перекладати великі об’єми тексту набраного вручну або навіть сфотографованого, та пропонувати безліч перекладів слова з певною статистикою, такою як популярність/вживаність та емоційне забарвлення слова в залежності від позиції так контексту введеного словосполучення, також такі типи перекладачів можуть пропонувати синоніми до слів та вгадувати наступне слово, яке ви можете написати, за допомогою аналізу даних мільйонів користувачів.

Станом на 2023 рік найпопулярнішими перекладачами є Google Translate, Reverso, Reverso in context, Unbabel, Microsoft Translator, Translator Me,

Yandex.translator, DeepL and Amazon. Translate. Всі ці перекладачі використовують всередині механізм штучного інтелекту з машинним навчанням.

Розглянемо короткий приклад архітектурного рішення для ефективного пошуку слів, яке реалізовано у продукті компанії Amazon “Amazon Translate”.

Amazon Translate є широко використовуваною послугою нейронного машинного перекладу, яка забезпечує швидкий, недорогий, якісний і настроюваний переклад. Розглянемо як команда архітекторів з Amazon побудувала алгоритм перекладу слів та на чому він базується.

Нейронний машинний переклад (NMT) – це тип автоматизації мовного перекладу, який використовує моделі глибокого навчання для отримання точнішого та природнішого перекладу, ніж стандартні статистичні та засновані на правилах алгоритми перекладу (див. рис. 1). Користувачі також можуть використовувати Amazon Translate для локалізації вмісту, наприклад веб-сайтів і програм, для широкого кола користувачів, просто перекладати величезну кількість тексту для аналізу та швидко вмикати міжмовне спілкування між користувачами. Методи глибокого навчання, які застосовуються через нейронну мережу, використовуються механізмами нейронного перекладу, як-от Amazon Translate, для забезпечення більш точного перекладу. Замість кількох слів до або після перекладу слова, нейронна мережа перевіряє повний контекст фрази під час перекладу.

Для виконання перекладу потрібен текстовий файл у кодуванні UTF-8, відомий як вихідний текст. Механізм перекладу аналізує кожне слово у вихідному тексті по одному, щоб створити семантичне представлення. Цим завданням займається кодувальник. Декодер використовує семантичне представлення для перекладу одного слова за раз після його формування. Ми можемо вирішити не надавати певну мову оригіналу за допомогою Amazon Translate, що корисно в ситуаціях, коли користувачі не знають, якою мовою спілкуються. Наприклад, програмі чату служби підтримки може знадобитися обробляти всі мови додатково на англійську.

Також Amazon Translate дозволяє користувачам аналізувати наявні аудіозаписи або транслювати аудіо в режимі реального часу для транскрипції.

Користувачі можуть передавати на службу живий аудіопотік і отримувати у відповідь потік тексту через безпечне з’єднання. Для цього використовується звичайний транскрибаційний алгоритм, який переводить голос у текст, а потім переганяє його через нейронний машинний переклад

Як можна зрозуміти, команда розробників та архітекторів з Amazon змогли створити дійсно складний, а головне ефективний алгоритм на основі NMT.

Не дивлячись на це, Amazon Translate не використовує жоден з алгоритмів корекції точності, який суттєво знижує швидкість перекладу, але надає велику точність у перекладі. Алгоритм Amazon застарів та не використовує такі сучасні технології для пошуку та зіставлення слів, як наприклад Meta AI Speech Translator.

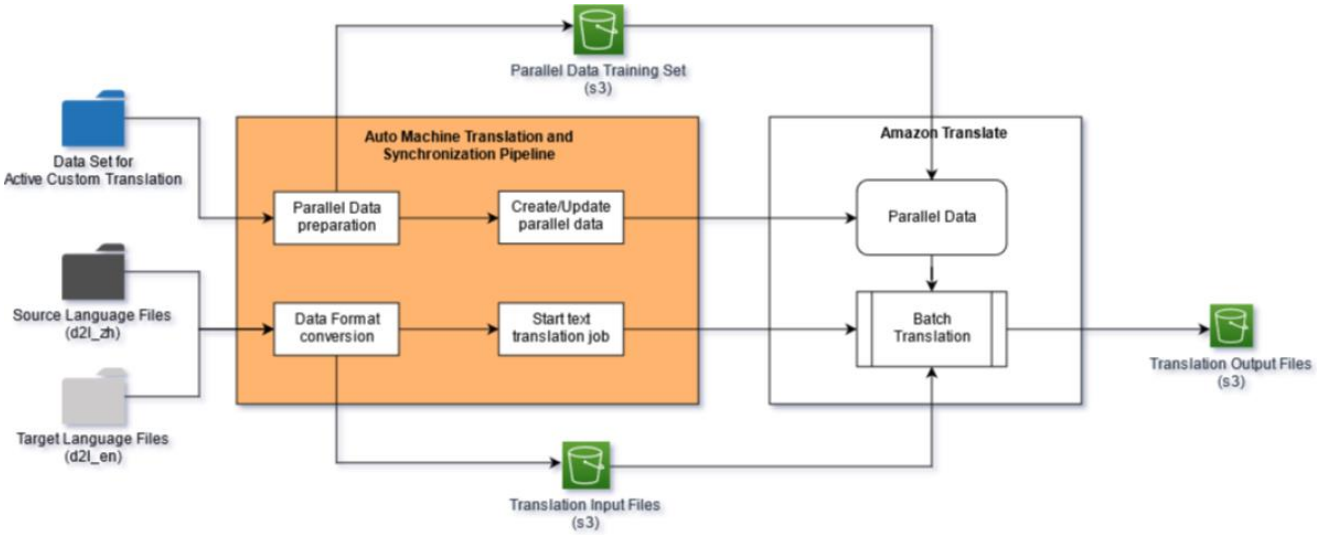


Рисунок 1 – Нейронний машинний переклад (NMT)

Як висновок для цього підрозділу можна виділити головну проблему – це відсутність суттєвих рухів у розвиненні алгоритмів та архітектурних рішень щодо поліпшення пошуку та перекладу слів з іноземної (англійської мови). На прикладі нейронного машинного перекладу (NMT), який виглядає непогано на фоні інших алгоритмів перекладання, але все ж таки вже застарів та не встигає за потребами користувача, ми означили стан розв’язання проблеми архітектурних рішень для пошуку слів з іноземної мови. Результатом цього є відкрита арена для нових, більш сучасних алгоритмів пошуку слів.

1.2 Виявлення проблем та актуалізація рішень

З розвитком економічної глобалізації англійська мова потрібна у все більшій кількості різних випадків у повсякденному житті людей, як і англійський переклад. Традиційний переклад програмного забезпечення англійською мовою здебільшого покладається на машину, що, з одного боку, полегшує життя людей, але, з іншого боку, має й певні недоліки. Наприклад, помилки перекладу характеризуються ітеративною передачею, слабкою логікою та низькою точністю. Традиційний машинний переклад не зміг задовольнити попит людей на швидкість і якість, тому він більше не може задовольняти потреби людей. Тому в цій роботі, як одне з рішень цієї проблеми пропонується розробка алгоритму корекції точності перекладу для програмного забезпечення для перекладу англійської мови як архітектурне рішення для поліпшення пошуку слів з іноземної мови.

Згідно мого рішення, яке буду розібрано у розділах «Аналіз існуючих методів або алгоритмів» та «Проведення досліджень», цей алгоритм демонструє, що найвища точність перекладу англійської мови до корекції становить лише 75,6%, тоді як найнижча точність досягає 98,7% після застосування алгоритму, описаного в цій статті. Різниця в точності між цими двома показниками свідчить про те, що ефективність системи корекції в цій статті робить видатний внесок (див. табл. 1).

Таблиця 1 – Порівняння точності перекладу до та після роботи алгоритму корекції

Номер експерименту	Точність перекладу	
	До корекції	Після корекції
1	68.8	99.8
2	72.7	99.8
3	67.9	99.7
4	72.4	99.8
5	75.6	99.9
Середня точність	71.5	99.1

Також, якщо брати алгоритми, які базуються на штучному інтелекті, то існує так звана алгоритмічна упередженість – одна з найбільш обговорюваних проблем у технології штучного інтелекту, і вона виявляється однією з найскладніших для подолання. Сувора реальність полягає в тому, що упередженість алгоритмів успадковується від людей, які їх створюють і навчають, а також від соціального середовища, в якому ми живемо.

Уявіть собі алгоритм, розроблений для прогнозування роботи людини, використовуючи лише зображення її обличчя. Ми знаємо, що робочі місця непропорційно заповнені представниками різних статей, рас, вікових категорій та інших груп людей, що і буде враховано алгоритмом, який використовується.

Якщо більшість програмістів – білі чоловіки, то алгоритм розробить базу, яка відображатиме цей тип даних.

Що стосується технології перекладу зі штучним інтелектом, то одна з найпоширеніших форм упередженості пов'язана з гендерною ознакою, яка може виникнути, якщо припустити, що медсестри - це жінки, а лікарі - чоловіки. Ця проблема ускладнюється тим, що більшість мов менше покладаються на гендерні займенники, ніж англійська. Google намагався вирішити цю проблему, надаючи переклади з використанням обох родів там, де це доречно, але це додає додаткової складності (алгоритмічне прийняття рішень) до процесу перекладу мови ШІ.

Існують й інші форми упередженості, до яких схильний машинний переклад, і вони також не завжди пов'язані з питаннями моралі. Алгоритми часто схильні до упередженості щодо довжини, що призводить до того, що переклад виявляється коротшим, ніж повинен бути, або навіть викликає упередженість у перекладачів, які перебувають під впливом матеріалу, який вони перевіряють на якість.

Головне питання полягає в тому, як технологія перекладу (яку розробили люди, часто схильні до упередженості) може бути вільною від упередженості?

За останні роки наука про дані досягла значного прогресу, але більша частина цього прогресу є результатом швидших і потужніших обчислювальних технологій, які здатні обробляти більшу кількість даних за коротший проміжок часу. Усі досягнення в галузі штучного інтелекту пов'язані з тим, щоб робити більше і

швидше, щоб алгоритми могли порівнювати більше наборів даних, що дає змогу виявляти більше закономірностей.

Це призвело до підвищення точності перекладу за допомогою ШІ, і є всі підстави для оптимізму, що в майбутньому ця технологія перекладу досягне рівня точності, достатнього для професійного використання в перекладацькій діяльності.

Однак точність перекладу навіть не починає вирішувати мовні проблеми технології перекладу зі штучним інтелектом.

Люди спілкуються не лише ізольованими словами. Коли слова поєднуються з іншими словами, які впливають на значення одне одного, вони будують речення, абзаци та цілі твори усного чи письмового мовлення. Ми створюємо контекст, натякаємо на значення, використовуємо тон і артикуляцію для акцентування, порівнюємо, використовуємо метафори, а також додаємо кольору сатирою, іронією і багатством мовних характеристик, які не можна визначити за допомогою правил.

Спочатку спробуйте пояснити, що таке іронія і чим вона схожа, а чим відрізняється від сарказму та сатири, а тепер спробуйте перетворити це пояснення на набір даних, які алгоритми зможуть застосувати до людської мови.

Забудьте на мить про переклад: ШІ не здатен навіть виявити ці характеристики в мові, не кажучи вже про їхній переклад (це може бути досить складним завданням для досвідчених перекладачів-людей). Якщо алгоритми ШІ коли-небудь стануть достатньо розумними, щоб точно перекладати людську мову в текст (включаючи мови, акценти, діалекти, мовленнєві перешкоди та всі інші характеристики, які людський мозок обчислює природним чином), ми зможемо вирішити більш серйозні проблеми технології перекладу зі штучним інтелектом.

Також серед інших проблем відповідного пошуку слів можна виділити:

– лексико-семантичні проблеми, які можна вирішити, звернувшись до словників, глосаріїв, термінологічних банків та експертів. До таких проблем належать термінологічні альтернативи, неологізми, семантичні лакуни або лексичні мережі. Іншими проблемами є контекстуальні синоніми та антоніми, які впливають на полісемічні одиниці: синоніми та антоніми спрямовані на сприйняття, яке залежить від контексту, щоб

визначити, яке значення є правильним. Ще одна проблема стосується семантичної суміжності, тобто процедури узгодженості, яка працює шляхом виявлення семантичних ознак, спільних для двох або більше термінів;

- граматичні проблеми, які можуть включати питання темпоральності, аспекту - коли дієслово вказує на те, чи дія триває, чи завершилася, - займенників, а також того, чи потрібно робити займенник суб'єкта експліцитним;
- синтаксичні проблеми, які можуть виникати через синтаксичні паралелі, напрямок пасивного стану, фокус - з якої точки зору розповідається історія - або риторичні фігури мови, такі як гіпербола - інверсія природного розташування слів - або анафора - повторення слова або сегмента на початку рядка або речення;
- риторичні проблеми, коли перекладачі стикаються з проблемами, пов'язаними з ідентифікацією та відтворенням фігур думки - порівняння, метафора, метонімія, синекдоха, оксюморон, парадокс та багато інших - а також з дикцією;
- практичні проблеми, які можуть виникати через різницю в офіційному та неофіційному способах звертання на "ти", а також через ідіоматичні фрази, приказки, іронію, гумор і сарказм. Перекладачі можуть зіткнутися з іншими проблемами, зокрема, з перекладом особового займенника "ти" при перекладі маркетингового тексту з англійської на французьку мову. Перекладач повинен вирішити, яке з них - формальне чи неформальне "ви" - є більш доречним, і таке рішення не завжди є очевидним;
- культурні проблеми, які можуть виникати через різні культурні посилання, такі як назви страв, фестивалів і культурних конотацій. Перекладач використовує мовну локалізацію, щоб точно адаптувати перекладений текст до цільової культури. Подумайте про фінансовий переклад, який містить дати. Якщо текст написаний англійською

мовою, найімовірніше, але не стовідсотково, що 05/06/2021 означатиме 5 червня 2021 року. Однак, як відомо, ця ж послідовність іншою мовою означає 6 травня 2021 року.

Як висновок з цього розділу, можна виділити наступне, що зараз існує багато алгоритмів та підходів для пошуку та перекладу слів з англійської мови на українську, але у кожного алгоритму є певні ознаки, які впливають на швидкість, точність, неупереджувальність. Деякі проблеми можна вирішити алгоритмом або декількома алгоритмами у лінійній перспективі, але інші проблеми, навіть у наш час стрімкого розвинення технологій і алгоритмів, не можливо вирішити, принаймні не у цій магістерській кваліфікаційній роботі.

1.3 Постановка задачі

Беремо до уваги інформацію, яка наведена у розділах вище, формуємо задачі чинної роботи – проведення дослідження архітектурних рішень для поліпшення пошуку слів з іноземної мови, у якому порівняти різні алгоритми для покращення пошуку та перекладу слів з однієї мови (англійської) на іншу та визначає їх ефективність.

У категорію «ефективність» алгоритмів входять наступні показники:

- скільки часу займає навчання мережі;
- точність перекладу;
- час підготовки даних типу TEST та TRAIN;
- вага специфічних показників налаштувань пошуку.

Таку велику задачу буде дуже складно виконати без чітких кроків виконання, тому розділимо її на маленькі підзадачі та виконаємо їх покроково:

- ознайомитися та проаналізувати математичне представлення алгоритмів перекладу;
- дослідити алгоритми на можливість подальшої оптимизації;
- побудувати план експерименту дослідження, що передбачає визначення

загальний умов, формалізацію функцію ефективності, визначення правил порівняння моделей та опис можливих помилок та невизначеностй;

- провести дослідження;
- формалізувати результати дослідження.

На основі отриманих результатів буде розроблена система підтримки прийняття рішень, як більш вдосконалений варіант тих, що було розглянуті вище у цьому розділі.

2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ АБО АЛГОРИТМІВ

2.1 Подолання розриву між зіставленням релевантності та семантичним зіставленням для моделювання схожості коротких текстів у Facebook Meta AI Speech Translator

Як було написано у вступі, настав час розбирати новий алгоритм перекладача Facebook Meta AI Speech Translator, який використовує багато засобів та алгоритмів для поліпшення пошуку слів іноземної мови.

Беручи до уваги той факт, що Meta AI Speech Translator – це дуже велика програма, у якій задіяні безліч алгоритмів, я опишу найбільш цікавий із них, який вирішує проблему розриву між зіставленням релевантності та семантичним зіставленням для моделювання схожості коротких текстів.

Основною проблемою інформаційного пошуку (ІП) є зіставлення релевантності, тобто ранжування документів за релевантністю до запиту користувача. З іншого боку, багато завдань НЛП, таких як відповіді на запитання та ідентифікація перефразування, можна вважати варіантами семантичного зіставлення, яке полягає у вимірюванні семантичної відстані між двома фрагментами коротких текстів. Хоча на високому рівні і релевантність, і семантичний збіг вимагають моделювання текстової схожості, багато існуючих методик для одного з них не можуть бути легко адаптовані до іншого. Щоб подолати цю прогалину, ми пропонуємо нову модель HCAN (Hybrid Co-Attention Network), яка складається з (1) гібридного модуля кодування, що включає кодувальники на основі ConvNet і LSTM, (2) модуля зіставлення релевантності, який вимірює збіги м'яких термінів з вагами важливості на різних рівнях деталізації, і (3) модуля семантичного зіставлення з механізмами спільної уваги, які фіксують контекстно-залежну семантичну спорідненість. Оцінки на численних тестах ІК та НЛП демонструють найсучаснішу ефективність порівняно з підходами, які не використовують попереднє навчання на зовнішніх даних. Широкі дослідження абляції свідчать про те, що сигнали реагування та семантичного

узгодження є взаємодоповнюючими в багатьох проблемних ситуаціях, незалежно від вибору базових кодерів.

Тепер розглянемо так звану модель HCAN (Hybrid Co-Attention Network), яка показана на рисунку 2.

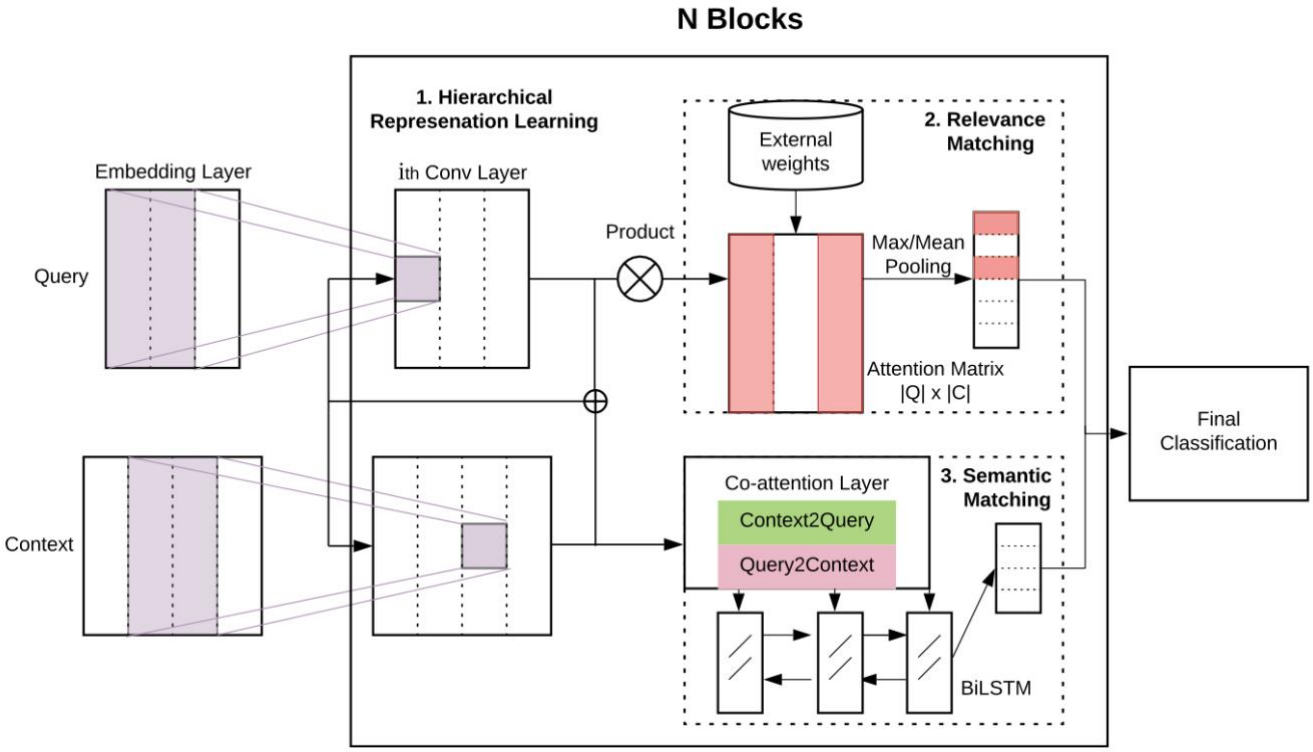


Рисунок 2 – Огляд Hierarchical Co-Attention Network (HCAN)

Як показано на рисунку 2, HCAN складається з трьох основних компонентів:

– модуль гібридного кодера (Hierarchial Representation Learning), який досліджує три типи кодерів: глибокий, широкий і контекстний;
– модуль релевантного зіставлення із зовнішніми вагами для навчання сигналів м'якого зіставлення (Relevant Matching);
– модуль семантичного зіставлення з механізмами спільної уваги для контекстно-орієнтованого навчання репрезентації (Semantic Matching).

Зауважимо, що модулі відповідності та семантичного зіставлення застосовуються на кожному кодерному рівні, і всі сигнали нарешті об'єднуються для класифікації.

Далі коротко розглянемо кожен із складових Hierarchical Co-Attention Network (HCAN).

2.1.1 Hybrid Encoders

Не втрачаючи загальності, ми припускаємо, що вхідними даними для нашої моделі є пари речень (q, c), де (q, c) може посилатися на пару (запит, документ) в умовах пошуку, пару (питання, відповідь) в умовах контролю якості і т.д. Запит q і контекст c позначаються відповідними словами {w1q,w2q,...,wnq} і {w1c,w2c,...,wmc }, де n і m - кількість слів у запиті та контексті. Шар емпіричного розпізнавання слів перетворює обидва представлення в їхні вкладені представлення Q ∈ Rn×L та C ∈ Rm×L, де L - розмірність вкладених слів.

Щоб вивчити ефективні представлення на рівні фраз, ми досліджуємо три різні типи кодерів: глибокий, широкий та контекстний, які детально описані нижче.

Глибинний кодер: Ця конструкція складається з декількох згорткових шарів, складених в ієрархічному порядку для отримання представлень k-грами більш високого рівня. Згортковий шар застосовує згорткові фільтри до тексту, який представлений матрицею вбудовування U (Q або C). Кожен фільтр проходить через вхідні дані вбудовуються інкрементно у вигляді ковзного вікна (з розміром вікна k), щоб зафіксувати позиційне представлення k сусідніх термів. Припускаючи, що згортковий шар має F фільтрів, цей CNN-шар (з підкладками) створює вихідний масив Uo∈R‖U‖×F.

Для спрощення записів ми вилучаємо суперскрипт o з усіх вихідних матриць і додаємо суперскрипт h для позначення виходу h-го згорткового шару. Таким чином, стекування N шарів CNN відповідає отриманню вихідної матриці h-го шару Uh ∈ R‖U ‖×F h через цю формулу:

$$\mathbf{U}^h = \mathbf{CNN}^h(\mathbf{U}^{h-1}), h = 1, \ldots, N,$$

де Uh-1 - вихідна матриця (h - 1) – го згорткового шару;

Зауважте, що U0 = U позначає вхідну матрицю (Q або C), отриману безпосередньо від шару вбудовування слів. Параметри кожного шару ШНФ є спільними для запиту та контексту.

Широкий кодер: На відміну від глибокого кодера, який ієрархічно укладає декілька згорткових шарів, широкий кодер організовує згорткові шари паралельно, причому кожен згортковий шар має різний розмір вікна k для отримання відповідних k-грамних зображень. За наявності N згорткових шарів, розміри вікон шарів CNN будуть [k,k+1,...,k+N -1].

Контекстний кодер: На відміну від глибокого та широкого кодерів, які фіксують k-грамові патерни зі згортками, контекстний кодер використовує двонаправлені LSTM для вилучення довгострокових контекстних ознак. За наявності N шарів BiLSTM вихід на h-му шарі обчислюється як:

$$\mathbf{U}^h = \mathbf{BiLSTM}^h(\mathbf{U}^{h-1}), h = 1, \ldots, N,$$

Три кодувальники представляють різні компроміси. Глибокі та широкі кодери легше використовувати для паралельного формування висновку, і вони набагато швидше навчаються, ніж контекстний кодер. Крім того, використання шарів CNN дозволяє нам явно контролювати розмір вікна для фразового моделювання, що, як було показано, є критично важливим для зіставлення релевантності. З іншого боку, контекстний кодер дає змогу отримати далекосяжні контекстні репрезентації для кожної лексеми. Якщо порівнювати глибинний та широкий кодувальники, то глибинний кодувальник зберігає більше параметрів завдяки повторному використанню репрезентацій з попереднього шару. Ефективність кожного з кодерів - це емпіричне питання, на яке ми спробуємо відповісти експериментально.

2.1.2 Relevance Matching

Цей розділ описує наші зусилля з отримання сигналів відповідності ключових слів для зіставлення релевантності. Ми обчислюємо показник релевантності між запитом і контекстом на кожному рівні кодування, перемножуючи матрицю представлення запиту Uq і матрицю представлення контексту Uc:

$$\mathbf{S} = \mathbf{U}_q \mathbf{U}_c^T, \mathbf{S} \in \mathbb{R}^{n \times m},$$

де Si,j можна вважати оцінкою схожості шляхом зіставлення вектора фраз запиту Uq[i] з вектором фраз контексту Uc[j].

Оскільки запит і контекст мають однакові шари кодера, схожі фрази будуть розміщені ближче у високорозмірному просторі вбудовування, а їхній добуток дасть більшу оцінку. Далі ми отримуємо нормалізовану матрицю схожості Š, застосовуючи функцію softmax до стовпців контексту S, щоб нормалізувати оцінки схожості в діапазоні [0, 1].

Для кожної фрази запиту і наведена вище функція softmax нормалізує оцінки відповідності для всіх фраз у контексті і допомагає розрізняти збіги з вищими оцінками. Точний збіг переважатиме над іншими і дасть показник схожості, близький до 1.0. Потім ми застосовуємо об'єднання максимуму та середнього значення до матриці схожості, щоб отримати вектори дискримінантних ознак:

$$\begin{gathered}
\boldsymbol{Max}(\mathbf{S}) = [\max(\tilde{\mathbf{S}}_{1,:}), ..., \max(\tilde{\mathbf{S}}_{n,:})], \\
\boldsymbol{Mean}(\mathbf{S}) = [mean(\tilde{\mathbf{S}}_{1,:}), ..., mean(\tilde{\mathbf{S}}_{n,:})], \\
\boldsymbol{Max}(\mathbf{S}), \boldsymbol{Mean}(\mathbf{S}) \in \mathbb{R}^n
\end{gathered}$$

Кожну оцінку, отриману в результаті об'єднання, можна розглядати як відповідність певній фразі запиту в контексті, де значення означає значущість сигналу релевантності. Порівняно з максимальним об'єднанням, середнє

об'єднання корисне у випадках, коли фраза запиту відповідає кільком релевантним термінам у контексті.

Варто зазначити, що моделювання важливості термінів може бути важливим для деяких пошукових задач, тому ми вводимо зовнішні ваги як критерії для вимірювання відносної важливості різних термінів і фраз запиту. Ми множимо оцінку після об'єднання на вагу конкретного терміна/фрази запиту. Вони надаються як вхідні дані для остаточного шару класифікації, що позначається ORM:

$$\mathbf{o}_{RM} = \{wgt(q) \odot \boldsymbol{Max}(\mathbf{S}), wgt(q) \odot \boldsymbol{Mean}(\mathbf{S})\},$$
$$\mathbf{O}_{RM} \in 2 \cdot \mathbb{R}^n,$$

де ⊙ - поелементний добуток ваг термінів/фраз запиту з оцінками пулу;
wgt(q)i - вага і-го терміна/фрази в запиті; її значення змінюється в проміжних шарах кодувальника, оскільки глибші/ширші шари кодувальника фіксують довші фрази. Як вагову функцію ми обрали зворотну частоту документів (IDF). Вища вага IDF означає, що документ рідше зустрічається в колекції, а отже, має більший криміногенний потенціал. Метод зважування також дозволяє нам зменшити вплив великих збігів для загальних слів, таких як стоп-слова.

2.1.3 Semantic Matching

На додаток до зіставлення релевантності, ми прагнемо зафіксувати сигнали семантичного зіставлення за допомогою механізмів спільної уваги на проміжних репрезентаціях запиту та контексту. Наш метод семантичного зіставлення поводиться подібно до трансформатора (Vaswani et al., 2017), який також використовує увагу (зокрема, самоувагу) над ієрархічними блоками для фіксації семантики на різних гранулярностях.

Маючи Uq ∈ Rn×F та Uc ∈ Rm×F, згенеровані проміжним кодувальним шаром, ми спочатку обчислюємо білінійну увагу наступним чином:

$$\mathbf{A} = \mathit{REP}(\mathbf{U}_q\mathbf{W}_q) + \mathit{REP}(\mathbf{U}_c\mathbf{W}_c) + \mathbf{U}_q\mathbf{W}_b\mathbf{U}_c^T$$
$$\mathbf{A} = \mathit{softmax}_{\text{col}}(\mathbf{A})$$
$$\mathbf{A} \in \mathbb{R}^{n\times m}$$

де Wq,Wc ∈ RF, Wb ∈ RF×F, а оператор REP перетворює вхідний вектор у матрицю Rn×m шляхом повторення елементів у пропущених вимірах.

Softmaxcol – це оператор софтмакс по стовпчиках. Ми виконуємо спільну увагу у двох напрямках: від запиту до контексту та від контексту до запиту, як показано нижче:

$$\tilde{\mathbf{U}}_q = \mathbf{A}^T\mathbf{U}_q$$
$$\tilde{\mathbf{U}}_c = \mathit{REP}(\max_{\text{col}}(\mathbf{A})\mathbf{U}_c)$$
$$\tilde{\mathbf{U}}_q \in \mathbb{R}^{m\times F}, \tilde{\mathbf{U}}_c \in \mathbb{R}^{m\times F}$$

де maxcol – оператор максимального об'єднання по стовпцях;

Ũq позначає вбудовування контексту з урахуванням запиту шляхом присвоєння сирим представленням запиту вагових коефіцієнтів уваги, тоді як Ũc позначає зважену суму найбільш важливих слів у контексті щодо запиту.

Потім ми використовуємо розширену конкатенацію, щоб дослідити взаємодію між Ũq і Ũc. Нарешті, ми застосовуємо додатковий Bi-LSTM до конкатенаційних контекстних вкладень H, щоб зафіксувати контекстні залежності в послідовності, і використовуємо останній прихований стан (з розмірністю d) як вихідні ознаки модуля семантичного зіставлення OSM:

$$\mathbf{H} = [\mathbf{U}_c; \tilde{\mathbf{U}}_q; \mathbf{U}_c \otimes \tilde{\mathbf{U}}_q; \tilde{\mathbf{U}}_c \otimes \tilde{\mathbf{U}}_q]$$
$$\mathbf{O}_{\text{SM}} = \text{BiLSTM}(\mathbf{H})$$
$$\mathbf{H} \in \mathbb{R}^{m\times 4F}, \mathbf{O}_{\text{SM}} \in \mathbb{R}^{d}$$

2.2 Алгоритм корекції точності перекладу для англомовних перекладацьких програм

Тепер поговоримо про алгоритм, який дуже сильно сприяє поліпшенню пошуку слів, бо він дозволяє знаходити слова, які найкраще підходять для перекладу на ту чи іншу мову.

Алгоритм, який використовується в традиційному машинному перекладі, - це переважно метод конвеєрного перекладу, який полягає в аналізі структури речення, складу речення та частини мови оригінального речення і завершенні перекладу після розуміння повної синтаксичної структури [1]. Цей метод перекладу легко призводить до накопичення помилкових ітерацій та низької точності [2]. З розвитком науки і техніки до технології перекладу висуваються все більш високі вимоги. Ця епоха приносить як можливості, так і виклики для перекладацьких технологій [3].

Враховуючи складність і високу вартість методу автоматичного перекладу, який використовується для побудови системи за звичайних обставин, пропонується метод автоматичного перекладу від дерева залежностей до моделі рядків, а також алгоритм корекції, придатний для перекладу на англійську мову. У порівнянні з традиційним методом, цей метод потребує лише аналізу структури синтезу вихідної мови, що значно зменшує складність побудови системи та ефективно знижує вартість [4]. З метою підвищення точності перекладу, впроваджуючи комбіновану модель класифікації китайських слів, часткових словникових тегів та синтаксичного аналізу, можна зменшити основні помилки вихідної мови в англійському перекладі та підвищити точність функції виведення перекладу [5].

2.2.1 Метод проектування алгоритму корекції похибки

Для проектування алгоритму корекції похибки ми використовуємо дві моделі:

- метод семантичної схожості лексики на основі Hownet;
- комп'ютерний інтелектуальний метод корекції на основі вдосконаленої моделі фразового перекладу.

Спочатку розберемо перший метод на основі Hownet.

Діапазон значень схожості - [0,1], а семантична схожість між різними словами W1 та W2 дорівнює [0,1]:

$$Sim_{semantic}(W_1, W_2) = \max_{i=1,2,\cdots,n;\, j=1,2,\cdots,m} Sim(S_{1i}, S_{2i})$$

де S1i(i=1,2,…,n) представляє n понять, що відповідають словнику W1;
S2i(i=1,2,…,m) - m понять, що відповідають словнику W2. Так звана семантична схожість - це значення з найбільшою схожістю серед множини понять цих двох слів.

Семантична схожість понять словника може бути використана для опису схожості понять словника. Подібність між ієрогліфами p1 та p2 можна обчислити за наступною формулою:

$$Sim(p_1, p_2) = \frac{\alpha}{d + \alpha}$$

де регульований параметр використовується для того, щоб вказати, що значення обох значень є більшими за 0, а відстань між ними виражається через d.

Тепер поговоримо про комп'ютерний інтелектуальний метод корекції на основі вдосконаленої моделі фразового перекладу.

Перетворення з одного текстового формату в інший - це процес корекції англійського перекладу. Тому процес інтелектуальної комп'ютерної корекції англійського перекладу - це, по суті, знову процес перекладу. Порівнюючи

переглянуті результати з оригінальними результатами перекладу, можна отримати більш точний переклад з англійської мови. У цій статті H визначається як неправильний результат перекладу з англійської мови, а D - як правильний результат перекладу з англійської мови. Перетворення від H до D і є процесом перекладу англійської мови. Метод автоматичного перекладу англійської мови, заснований на покращеному режимі перекладу фраз, полягає в наступному:

$$\hat{D} = \arg\max_{c} M\left(D|H\right) = \arg\max_{c} M\left(H|D\right) \cdot M\left(D\right)$$

Найважливішим у машинному перекладі англійської мови є підвищення точності перекладу лексики. Тому інтелектуальна система корекції комп'ютерного перекладу англійської мови може фундаментально вирішити проблему точності перекладу англійської мови. M(D) у формулі нижче представляє точність. Отже, на основі формули оптимізації, яку ми щойно описали, реалізується комп'ютерна інтелектуальна корекція, а конкретний метод полягає в наступному:

$$\hat{D} = \arg\max_{c} M\left(H|D\right)^{\varepsilon} \cdot M\left(D\right)^{\gamma}$$

де ваги ε,γ вагових коефіцієнтів M(H|D) та M(D) представлені відповідно.

Для того, щоб полегшити вираз методу комп'ютерної інтелектуальної корекції потрібно скоротити вищенаведені формули та вивести універсальну формулу для корекції.

Для полегшення опису методу комп'ютерної інтелектуальної корекції на основі покращеної моделі фразового перекладу через H позначено словник, що підлягає корекції, а через D - виправлений словник. Визначено, що в H існує p символів, які позначено через Hp1; в той же час в D існує q символів, які позначено через Dq1. Визначення розбиває Hpq на d випадкових рядків, позначених Ĥ d1, де рядки відповідають фразам у моделі фразового перекладу. Аналогічно, коректурний словник, згенерований коректурним словником, містить d рядків, які

описуються Dd1. Таким чином, розширена форма рівняння виглядає наступним чином:

$$\hat{D} = \arg\max_{D \cdot \tilde{M}_1^d} \sum_{\tilde{M}_1^d} M\left(\tilde{H}_1^d \middle| H_1^p\right) \cdot M\left(\tilde{H}_1^d \middle| \tilde{D}_1^d\right)^{\varepsilon} \cdot M\left(\tilde{D}_1^q\right)^{\gamma}$$

У процесі інтелектуальної корекції англійського перекладу необхідно зосередитися на навичках перекладу та методах словникового запасу в різних сценаріях, а також вичитувати результати перекладу один за одним, і, нарешті, здійснити процес корекції для підвищення точності англійського перекладу. Поєднуючи метод, описаний вищезазначеною формулою, для пошуку словника D, що відповідає словнику H, який потрібно вичитати, реалізувати інтелектуальну корекцію комп'ютерного перекладу англійської мови.

2.2.2 Інші моделі

Дерево залежностей до рядкової моделі.

Форма представлення моделі дерева залежностей до рядка має вигляд <D, S, A>. Тут <D, S> - пара перекладу, D - дерево залежностей вихідної мови, S - рядок слів перекладу в вихідній мові, а A - відношення між D і S. Дерево залежностей вихідної мови D відношення вирівнювання слів. Кожне слово D має характеристику. Англійські літери під кожним словом представляють звукову частину цього слова. Наприклад, іменник - NN, дієслово - VV, прикметник - JJ тощо. Для зв'язку між ієрогліфами та словами в третій частині моделі є англійські S-рядки, що відповідають китайським реченням. З'єднання верхньої та нижньої частин можна використати для ілюстрації конфігурації зв'язку між вузлом китайського слова та англійським словом

Логіко-лінійна модель.

Логіко-лінійна модель – це модель, яку можна оцінити, і для оцінки дивергентного мислення обирається декілька ознак. Для заданого речення: $f^{j}_{1}= f_1\ldots, f_n$, модель перекладу, $e^{j}_{1}= e_1\ldots, e_n$, утворюється з максимальною ентропією:

$$e_1^J = \sum_{m=1}^{M} \lambda_m h_m\left(e_1^J, f_1^J\right)$$

Лог-лінійна модель має високу масштабованість, дозволяє встановлювати характеристики, що відповідають різним цільовим вимогам, і може застосовувати методи машинного перетворення на декількох мовах. Основною формою системи машинного перекладу є можливість прямого та зворотного перекладу та перемикання мовних режимів перекладу. Відповідно до фактичних потреб системи перекладу, налаштуйте систему функцій та визначте вагу відповідного органу та отримайте найвищий бал перекладу за наведеною вище формулою.

Дослідження цього алгоритму та його результати наведені у розділах нижче.

3 ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ

3.1 Дослідження алгоритму подолання розриву між зіставленням релевантності та семантичним зіставленням для моделювання схожості коротких текстів у Facebook Meta AI Speech Translator

Перш за все, нам потрібно визначити еталони та метрики. Я оцінив запропоновану мною модель HCAN на трьох завданнях NLP та двох наборах ІЧ-даних наступним чином:

- підбір відповідей – це завдання полягає в ранжуванні речень-відповідей на основі їхньої схожості з питанням. Ми використовуємо набір даних TrecQA (сира версія) 1 з 56 тис. пар запитань-відповідей. Я показую середню точність (MAP) і середній взаємний ранг (MRR);
- ідентифікація перефразування – це завдання полягає у визначенні того, чи є два речення парафразами одне одного. Я використовую набір даних TwitterURL з 50 тис. пар речень. Я подаю незважене середнє значення оцінок F1 для позитивних і негативних класів (макро-F1);
- семантична текстова подібність (STS) – це завдання полягає у вимірюванні ступеня семантичної еквівалентності між парами текстів. Я використовую набір даних Quora (Iyer et al., 2017) з 400 тис. пар запитань, зібраних на сайті Quora. Я повідомляю про точність передбачення класів;
- пошук твітів – це завдання полягає в ранжуванні твітів за датою за релевантністю щодо короткого запиту. Я використовую набори даних TREC Microblog 2013-2014, підготовлені Pao та ін., де кожен набір даних містить близько 50 запитів і 40 тис. пар запитів і твітів. Я звітую про MAP і точність на рівні 30

3.1.1 Базові показники та імплементація

Щодо завдань на вибір відповіді, визначення перефразування та STS, я порівнювала з такими базовими показниками: InferSent, ESIM, DecAtt та PWIM. Крім того, ми повідомляємо про найсучасніші результати для кожного набору даних з опублікованої літератури. Я також включив поточні сучасні результати BERT для кожного набору даних.

Для завдання пошуку твітів ми здебільшого дотримуємося експериментальних налаштувань, описаних у Rao та ін. (2019). Базові лінії включають класичний метод ймовірності запиту (QL), розширення запиту RM3, навчання ранжуванню (L2R), а також низку нейронних моделей ранжування: DRMM, DUET, K-NRM та PACRR. Для нейронних базових даних я використовував реалізацію в MatchZoo.2 Для L2R я використовував LambdaMART на тих самих наборах ознак, що й Rao та ін. (2019): на основі тексту, URL-адреси та хештегів. Нарешті, я включив результати BERT від Yang та ін. (2019b).

У своїх експериментах я використовую вставки word2vec розміром 300d, які можна навчати, з оптимізатором SGD. Для слів, що не входять до словника, ми ініціалізуємо їхні вставки рівномірним розподілом з [0, 0.1]. Оскільки наша модель також використовує зовнішні ваги, ми обчислюємо значення IDF з навчального корпусу. Кількість шарів згортки N дорівнює 4, а розмір згорткового фільтра k дорівнює 2. Прихована розмірність d дорівнює 150. Ми налаштовуємо швидкість навчання в діапазоні [0.05, 0.02, 0.01], кількість згорткових фільтрів F в діапазоні [128, 256, 512], розмір партії в діапазоні [64, 128, 256] та коефіцієнт відсіву між 0.1 та 0.5.

3.1.2 Результати дослідження

Мої основні результати за наборами даних TrecQA, TwitterURL та Quora показані на рисунку 3, а результати за мікроблогом TREC 2013-2014 - на рисунку 4 (ст. 32). Найкращі показники для кожного набору даних (окрім BERT) виділено жирним шрифтом. Я порівнюю з трьома варіантами нашої моделі HCAN: (1) тільки сигнали релевантності (RM), (2) тільки семантичні сигнали відповідності (SM) і (3) повна модель (HCAN). У цих експериментах я використовую глибинний кодер.

Model	TrecQA		TwitterURL	Quora
	MAP	MRR	macro-F1	Acc
InferSent	0.521	0.559	0.797	0.866
DecAtt	0.660	0.712	0.785	0.845
ESIM_{seq}	0.771	0.795	**0.822**	0.850
ESIM_{tree}	0.698	0.734	-	0.755
$\text{ESIM}_{seq+tree}$	0.749	0.768	-	0.854
PWIM	0.739	0.795	0.809	0.834
State-of-the-Art Models				
Rao et al. (2016)	**0.780**	0.834	-	-
Gong et al. (2018)	-	-	-	**0.891**
BERT	0.838	0.887	0.852	0.892
Our Approach				
RM	0.756	0.812	0.790	0.842
SM	0.663	0.725	0.708	0.817
HCAN	0.774	**0.843**	0.817	0.853

Рисунок 3 – Результати на TrecQA, TwitterURL та Quora

З рисунку 3 видно, що на всіх трьох наборах даних зіставлення за релевантністю (RM) досягає значно вищої ефективності, ніж семантичне зіставлення (SM). На наборі даних TrecQA він значно перевершує інші конкурентні базові лінії (InferSent, DecAtt і ESIM), і все ще порівнянний з базовими лініями на TwitterURL і Quora. Цей висновок свідчить про те, що м'які сигнали відповідності самі по собі є досить ефективними для багатьох завдань моделювання текстової схожості. Однак SM працює набагато гірше на TrecQA і TwitterURL, тоді як розрив

між SM і RM зменшується на Quora. Поєднуючи сигнали SM і RM, ми спостерігаємо послідовне підвищення ефективності HCAN у всіх трьох наборах даних, отримуючи нові найсучасніші (не BERT) результати на TrecQA.

Model	TREC-2013		TREC-2014	
	MAP	P@30	MAP	P@30
QL	0.2532	0.4450	0.3924	0.6182
RM3	0.2766	0.4733	**0.4480**	0.6339
L2R	0.2477	0.4617	0.3943	0.6200
Neural Baselines				
DUET	0.1380	0.2528	0.2680	0.4091
DRMM	0.2102	0.4061	0.3440	0.5424
K-NRM	0.1750	0.3178	0.3472	0.5388
PACRR	0.2627	0.4872	0.3667	0.5642
BERT	0.3357	0.5656	0.5176	0.7006
Our Approach				
RM	0.2818	0.5222	0.4304	0.6297
SM	0.1365	0.2411	0.2414	0.3279
HCAN	**0.2920**	**0.5328**	0.4365	**0.6485**

Рисунок 4 – Результати на мікроблозі ТРЕК 2013-2014

На рисунку 4 ми бачимо, що метод розширення запиту (RM3) перевершує більшість моделей нейронного ранжування, за винятком BERT, що узгоджується з Yang et al. (2019a). Ми припускаємо дві причини:

- твіти набагато коротші, а інформаційний текст є "шумнішим", ніж довші документи в Інтернеті або стрічці новин, для яких і були розроблені попередні нейронні моделі;
- більшість нейронних базових ліній будуються безпосередньо на основі матриці схожості без будь-якого навчання репрезентації, що може бути менш ефективним.

Порівнюючи запропоновані нами підходи на рисунку 4, RM досягає досить хороших результатів, тоді як SM не є ефективним взагалі, що підтверджує нашу

гіпотезу про те, що сигнали узгодження термінів є важливими для задач ІЧ розпізнавання. Цей висновок ще більше підтверджує нашу мотивацію для об'єднання SM і RM. Дійсно, самі по собі методи семантичного зіставлення неефективні, коли запити складаються лише з кількох ключових слів, без великої кількості семантичної інформації, яку можна використати. Однак контекстно-орієнтовані репрезентації, отримані з SM, роблять свій внесок у RM, що призводить до чудових результатів нашої повної моделі HCAN.

Більш детально про результати досліджень Facebook Meta AI Speech Translatorб а саме про алгоритм подолання розриву між зіставленням релевантності та семантичним зіставленням для моделювання схожості коротких текстів буде написано у розділі аналіз результатів дослідження.

3.2 Дослідження алгоритму корекції точності перекладу

Почнемо аналіз алгоритму корекції точності перекладу з чисельного порівняння різних програм для перекладу англійської мови, яке можна побачити у таблиці 2.

Таблиця 2 – Чисельне порівняння різних програм для перекладу англійської мови

Програма для пошуку слів	BLEU value	NIST value
Google Translate	25.43	5.7673
Baidu Translate	25.18	5.7318
Netease translation	24.53	5.7624

Тепер для кращого представлення переведемо таблицю у графік типу гістограма, який зображено на рисунку 5.

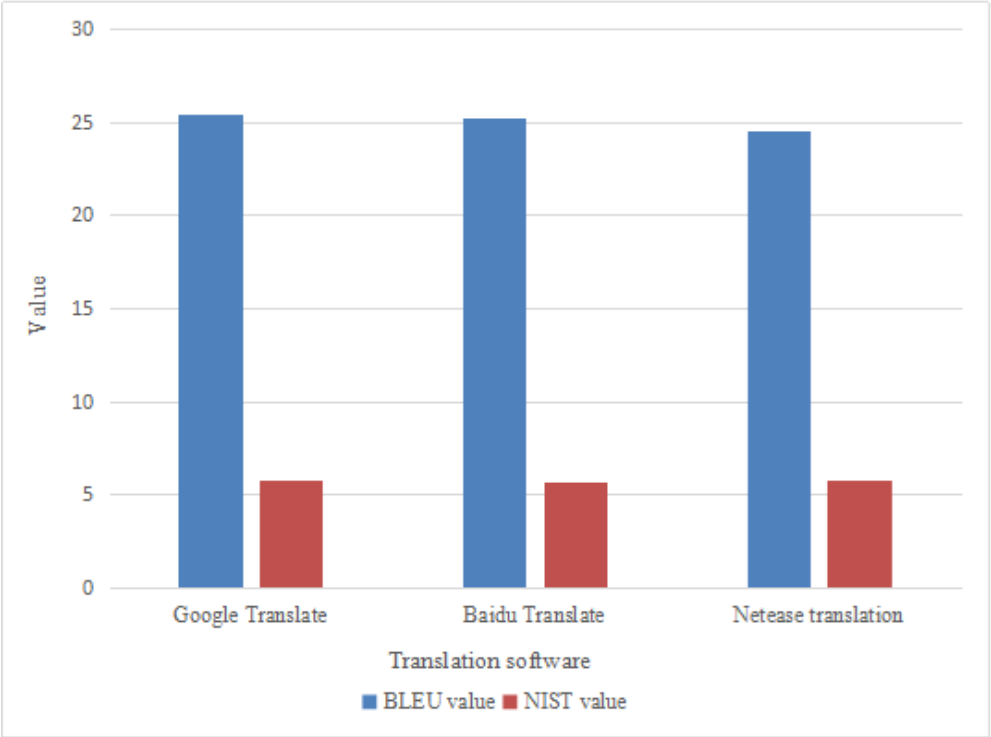


Рисунок 5 – Чисельне порівняння різних програм для перекладу англійської мови

BLEU у наведеній вище таблиці - це порівняльний аналіз одиничних сегментів оцінюваного перекладу та перекладу-еталона. Що більше відповідних сегментів, то вища якість перекладу. NIST - це стандарт вимірювання для оцінювання якості перекладу. Використовується для оцінювання якості перекладу кожної одиниці. Чим вище значення NIST, тим вища якість перекладу. З таблиці 1 і рисунка 1 видно, що значення BLEU і NIST для перекладача Google вищі, ніж для двох інших програм для перекладу. Це свідчить про те, що якість перекладених статей за допомогою Google Перекладача є вищою [9].

Більш детально про результати досліджень алгоритму корекції точності буде написано у розділі аналіз результатів дослідження.

4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

4.1 Аналіз результатів дослідження алгоритму подолання розриву між зіставленням релевантності та семантичним зіставленням для моделювання схожості коротких текстів

Почнемо аналіз результатів дослідження з порівняння результатів трьох різних шифраторів. Результат зображений на рисунку 6.

Encoder	Model	TrecQA		TwitURL	Quora	TREC-2013		TREC-2014	
		MAP	MRR	macro-F1	Acc	MAP	P@30	MAP	P@30
Deep	RM	0.756	0.812	0.790	0.842	0.282	0.522	0.430	0.630
	SM	0.663	0.725	0.708	0.817	0.137	0.241	0.241	0.328
	HCAN	**0.774**	0.843	**0.817**	**0.853**	**0.292**	**0.533**	**0.437**	**0.649**
Wide	RM	0.758	0.806	0.790	0.830	0.278	0.510	0.421	0.617
	SM	0.673	0.727	0.719	0.811	0.138	0.247	0.247	0.336
	HCAN	0.770	**0.847**	0.795	0.843	0.285	0.524	0.435	0.642
Contextual	RM	0.690	0.736	0.811	0.804	0.272	0.503	0.417	0.613
	SM	0.668	0.735	0.730	0.805	0.133	0.256	0.242	0.324
	HCAN	0.739	0.790	0.815	0.826	0.285	0.524	0.434	0.635

Рисунок 6 – Оцінка різних кодерів

Загалом, ефективність глибокого та широкого кодерів є досить близькою, враховуючи, що обидва кодери фіксують однакові типи n-грамних сигналів, що співпадають. Контекстний кодер працює гірше, ніж два інших на TrecQA, але є порівнянним на всіх інших наборах даних. Цей висновок узгоджується з дослідженням Rao та ін. (2017a), яке показує, що сигнали збігу ключових слів є важливими для TrecQA. Також я помітив, що розриви між RM і SM менші для всіх кодерів на Quora. Я підозрюю, що SM більш вимогливий до даних, ніж RM, зважаючи на його більший простір параметрів (насправді, модуль RM не має параметрів, які можна вивчити), а Quora приблизно в 10 разів більший за інші набори даних. Для всіх кодерів об'єднання RM і SM послідовно підвищує ефективність, підтверджуючи, що сигнали релевантності та семантичної відповідності є взаємодоповнюючими, незалежно від вибору базового кодера.

Щоб краще зрозуміти різні механізми роботи кодерів, я варіював кількість шарів кодерів для глибинного та контекстного кодерів на рисунку 7 (оскільки широкий кодер поводиться подібно до глибинного, ми опускаємо його аналіз тут). Моя повна модель HCAN має N = 4. На рисунку 2, частина A, ми бачимо загальне зростання ефективності для RM і HCAN (гребінка) зі збільшенням N, що свідчить про те, що моделювання фраз на великі відстані є критично важливим. Однак збільшення довжини контекстного вікна не допомагає SM на наборах даних TrecQA і TREC- 2013, ймовірно, через домінуючі сигнали збігу біграм (N = 1). Крім того, повна модель HCAN у більшості випадків виявилася кращою, ніж SM і RM окремо, що підтверджує її вищу ефективність, як видно з результатів у наведених вище таблицях. На противагу цьому, збільшення кількості шарів BiLSTM іноді може навіть зашкодити, як показано на рисунку 7, частина B. Це не дивно, оскільки один шар BiLSTM (N = 1) вже може захопити далекосяжну контекстну інформацію, а збільшення кількості шарів може ввести більше параметрів і призвести до надмірної підгонки.

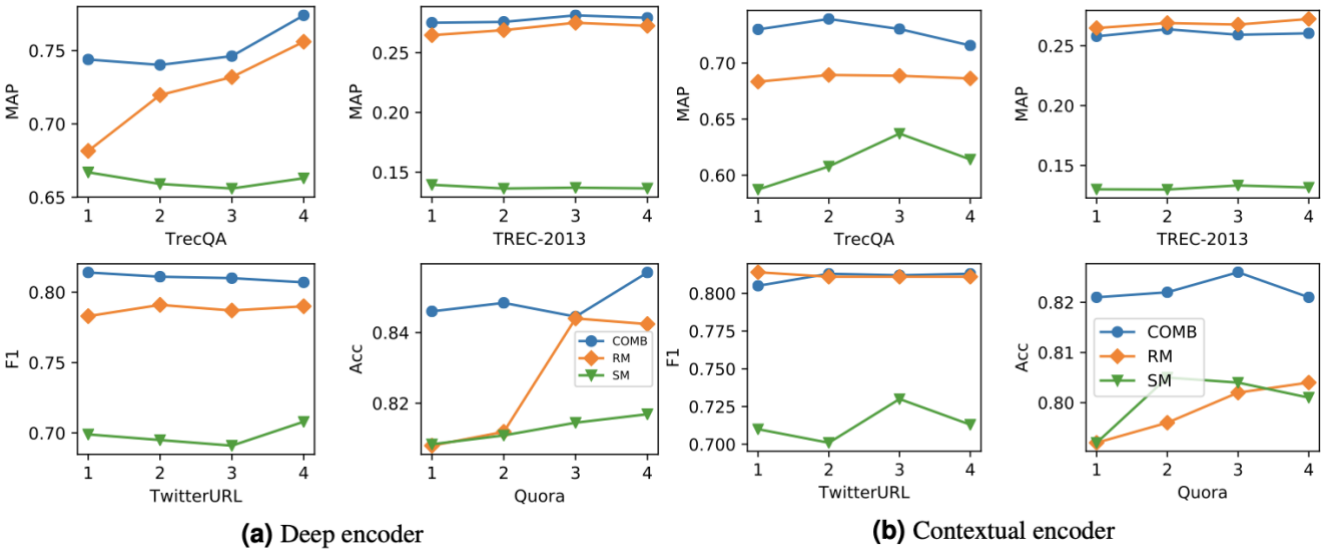
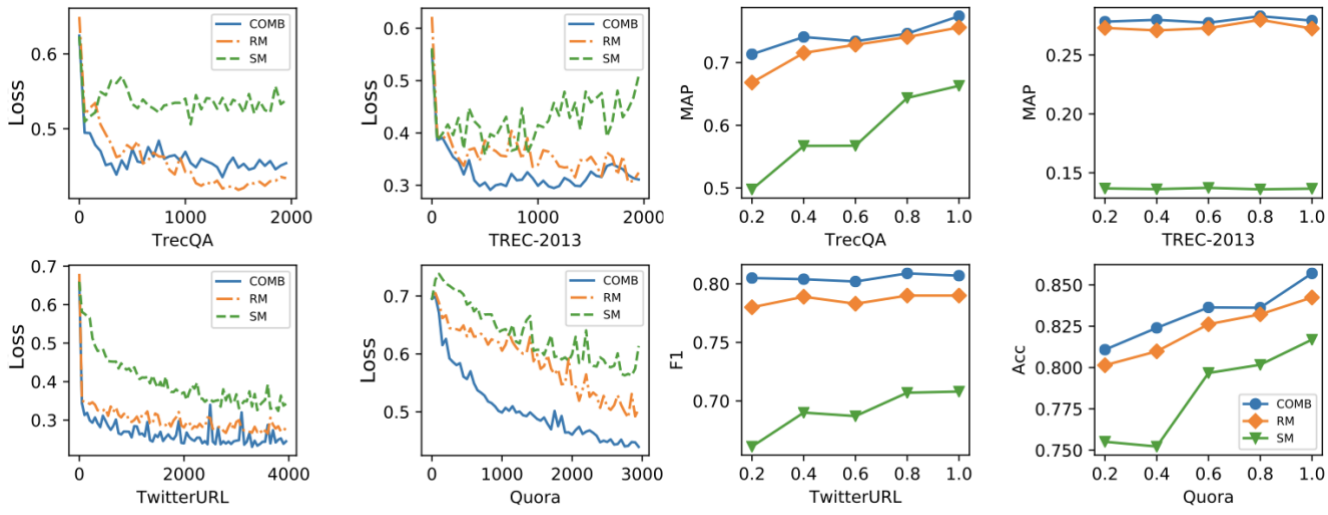


Рисунок 7 – Ефективність моделі з різною кількістю шарів кодера

Я також розробив два експерименти, щоб перевірити, чи повна модель HCAN навчається ефективніше, ніж SM або RM окремо. У цих експериментах ми використовували глибокий кодер. По-перше, на рисунку 8, частина A, показано втрати при перевірці для різної кількості партій. По-друге, я варіював розмір

навчальних даних, випадковим чином вибираючи різні відсотки (від 20% до 100%) від початкового навчального набору, як показано на рисунку 8, частина Б.

На рисунку 8, частина А, ми бачимо, що втрата валідності для повної моделі падає набагато швидше, ніж для RM і SM окремо, особливо для TwitterURL і Quora. На рисунку 8, частина Б, ми бачимо, що, як і очікувалося, всі методи загалом досягають вищих результатів, коли для навчання використовується більше даних. Винятком є SM на наборі даних Twitter TREC-2013, який, як ми бачимо на рисунку 4 (стор. 32), не є ефективним. Інший важливий висновок полягає в тому, що і RM, і HCAN є більш ефективними на даних: для TREC-2013 і TwitterURL обидва методи можуть досягти ефективності, порівнянної з повним навчальним набором, використовуючи лише 20% даних. Я також розробив два експерименти, щоб перевірити, чи повна модель HCAN навчається ефективніше, ніж SM або RM окремо. У цих експериментах ми використовували глибокий кодер. По-перше, на рисунку 8, частина А, показано втрати при перевірці для різної кількості партій. По-друге, я варіював розмір навчальних даних, випадковим чином вибираючи різні відсотки (від 20% до 100%) від початкового навчального набору, як показано на рисунку 8, частина Б.



(a) Validation losses w.r.t. number of batches. **(b)** Effectiveness w.r.t. different percentage of training data.

Рисунок 8 – Експерименти з вивчення ефективності навчання

На рисунку 8, частина А, ми бачимо, що втрата валідності для повної моделі падає набагато швидше, ніж для RM і SM окремо, особливо для TwitterURL і Quora.

На рисунку 8, частина Б, ми бачимо, що, як і очікувалося, всі методи загалом досягають вищих результатів, коли для навчання використовується більше даних. Винятком є SM на наборі даних Twitter TREC-2013, який, як ми бачимо на рисунку 4 (стор. 32), не є ефективним. Інший важливий висновок полягає в тому, що і RM, і HCAN є більш ефективними на даних: для TREC-2013 і TwitterURL обидва методи можуть досягти ефективності, порівнянної з повним навчальним набором, використовуючи лише 20% даних.

Я представляю приклади результатів на рисунку 9, щоб краще зрозуміти поведінку моделі. З міркувань економії місця я показую лише набір даних Quora, але наш аналіз показує схожі результати на інших наборах даних. Стовпчик "мітка" позначає мітку пари речень: 1 означає, що вони семантично еквівалентні, а 0 - що не еквівалентні. Для кожної моделі ми виводимо її передбачену мітку разом з оцінкою достовірності; фрази з великою вагою уваги виділені помаранчевим і червоним кольорами.

Label	SM Score	RM Score	HCAN Score	Sample Pair
1	1, 0.9119	0, 0.9353	1, 0.5496	- How does it feel to kill a human ? - How does it feel to be a murderer ?
1	0, 0.9689	1, 0.8762	1, 0.8481	- What are the time dilation effects on the ISS ? - According to the theory of relativity , time runs slowly under the influence of gravity . Is there any time dilation experienced on the ISS ?
0	0, 0.9927	1, 0.8473	1, 0.7280	- Does RBI send its employees for higher education such as MBA , like sponsoring the education or allowing paid / unpaid leaves ? - Does EY send its employees for higher education such as MBA , like sponsoring the education or allowing paid / unpaid leaves ?

Рисунок 9 – Приклади пар з Quora. Фрази з великою вагою уваги виділені помаранчевим і червоним кольорами.

У першому прикладі SM може правильно визначити, що два речення передають одне й те саме значення з високою достовірністю, тоді як RM зазнає невдачі, оскільки обидва речення не мають впливових словосполучень (з високими вагами IDF). Пара речень у другому прикладі має значний текстовий збіг. Не дивно, що RM прогнозує високу оцінку релевантності, тоді як SM не може вловити їхній

зв'язок. В обох прикладах HCAN здатен інтегрувати SM і RM, щоб зробити правильні прогнози. Оскільки в третьому прикладі спостерігається подібна картина, ми опускаємо детальне пояснення. Загалом, наш кількісний та якісний аналіз показує, що зіставлення за релевантністю краще вловлює сигнали, що перетинаються, тоді як поєднання сигналів семантичного зіставлення покращує навчання репрезентації.

4.2 Аналіз результатів дослідження алгоритму корекції точності перекладу

Після написання та застосування алгоритму корекції точності на реальних даних, я отримав такі результати у порівнянні з даними без застосування алгоритму корекції. На рисунку 10 зображена різниця у процентному співвідношення між даними без корекції та з корекцією.

Experiment serial number	Translation accuracy	
	Before correction/%	After using the correction algorithm/%
1	68.8	99.8
2	72.7	98.8
3	67.9	98.7
4	72.4	99.8
5	75.6	98.9
Mean accuracy	71.5	99.1

Рисунок 10 – Порівняння точності перекладу до і після застосування алгоритму корекції

З рисунків 10 і 11 видно, що найвища точність результатів перекладу з англійської мови до корекції становить лише 75,6%. Після корекції цього тексту найнижча точність досягає 98,7%. Різниця в точності між цими двома показниками свідчить про те, що ефективність системи корекції в цьому документі є значною. З точки зору середньої точності перекладу, середній результат перекладу англійської мови без корекції становить лише 71,5%. Після використання системи для корекції

середня точність підвищується на 27,6%, що ще раз підтверджує ефективність комп'ютеризованого інтелектуального перекладу.

Ефективність комп'ютеризованої системи інтелектуальної корекції англійського перекладу [10, 11].

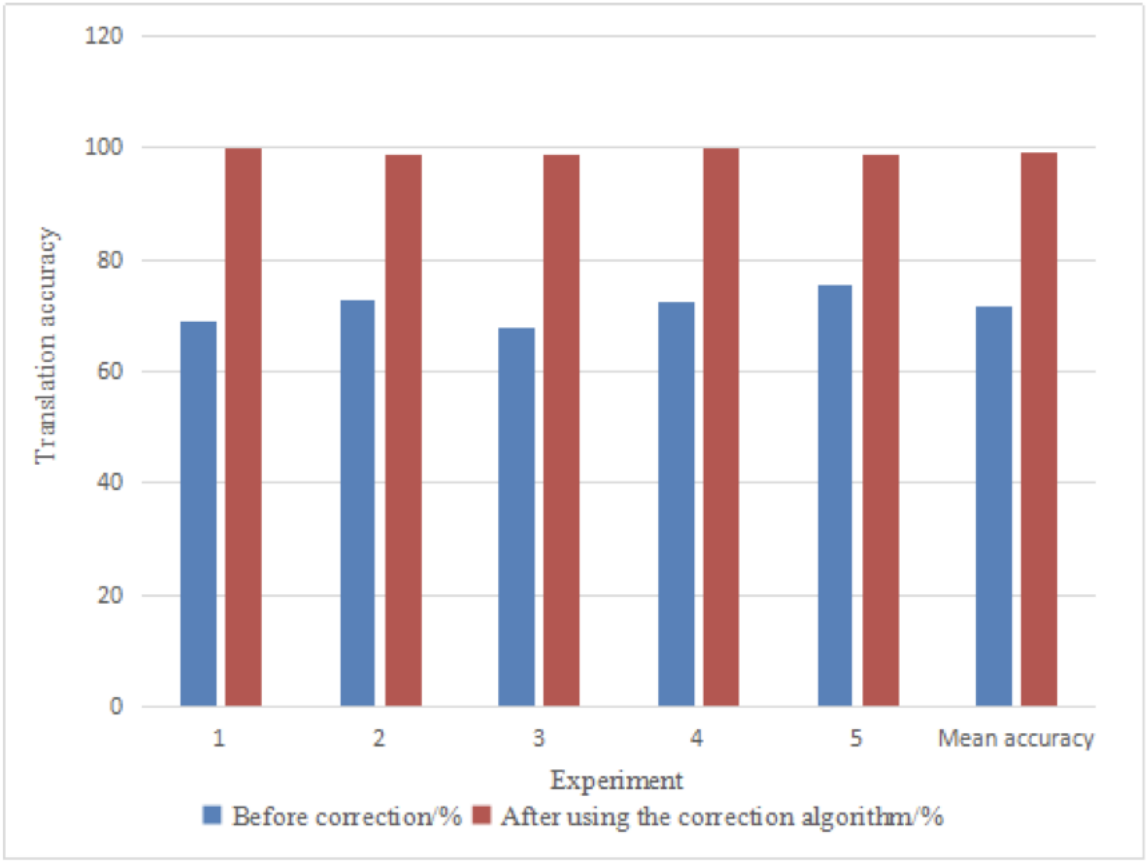


Рисунок 11 – Порівняння точності перекладу до і після застосування алгоритму корекції

У результаті можна сказати, що експерименти довели, що точність перекладу до корекції значно нижча, ніж після корекції, що свідчить про те, що алгоритм корекції в цій статті зіграв дуже хороший ефект.

5 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

Будь-яка розробка програмної системи повинна починатися з її моделювання за допомогою різних засобів. За допомогою UML-моделювання буде більш детально описана розроблювальна система, її функції, робота окремих частин та їх зв’язок між собою.

Діаграма варіантів використання – це початкова форма вимог до системи/програмного забезпечення для системи розробки. Ця діаграма визначає очікувану поведінку програмного забезпечення. Ключова концепція моделювання варіантів використання полягає в тому, щоб допомогти розробити систему на основі точки зору кінцевого користувача.

Для якісного моделювання системи необхідно виділити межі системи, головних акторів, які використовуватимуть систему, та основні функції, які актори використовуватимуть у діаграмі варіантів використання.

Якщо людина вирішила вивчати іноземні мови або вузькі поняття і слова, наприклад, комп’ютерні терміни англійською мовою (що потрібно зробити в рамках магістерської дипломної роботи), то вона може стати користувачем системи, це програмне забезпечення повинно повністю задовольнити потреби клієнта.

Основні потреби користувачів нашої системи, на прикладі лексики профілю навчання, такі:

- пошукова система;
- вміння зберігати слова;
- створити список слів;
- навчальна система для легкого вивчення слів.

На момент розробки цієї програми подібної системи у Free Access не існувало.

Діаграма варіантів використання описує можливі варіанти використання системи на найвищому рівні абстракції. Діаграма розроблювальної системи показана на рисунку 12.

У поєднанні з діаграмою варіантів використання ми можемо отримати огляд основних функцій системи та акторів та їхні можливості в системі.

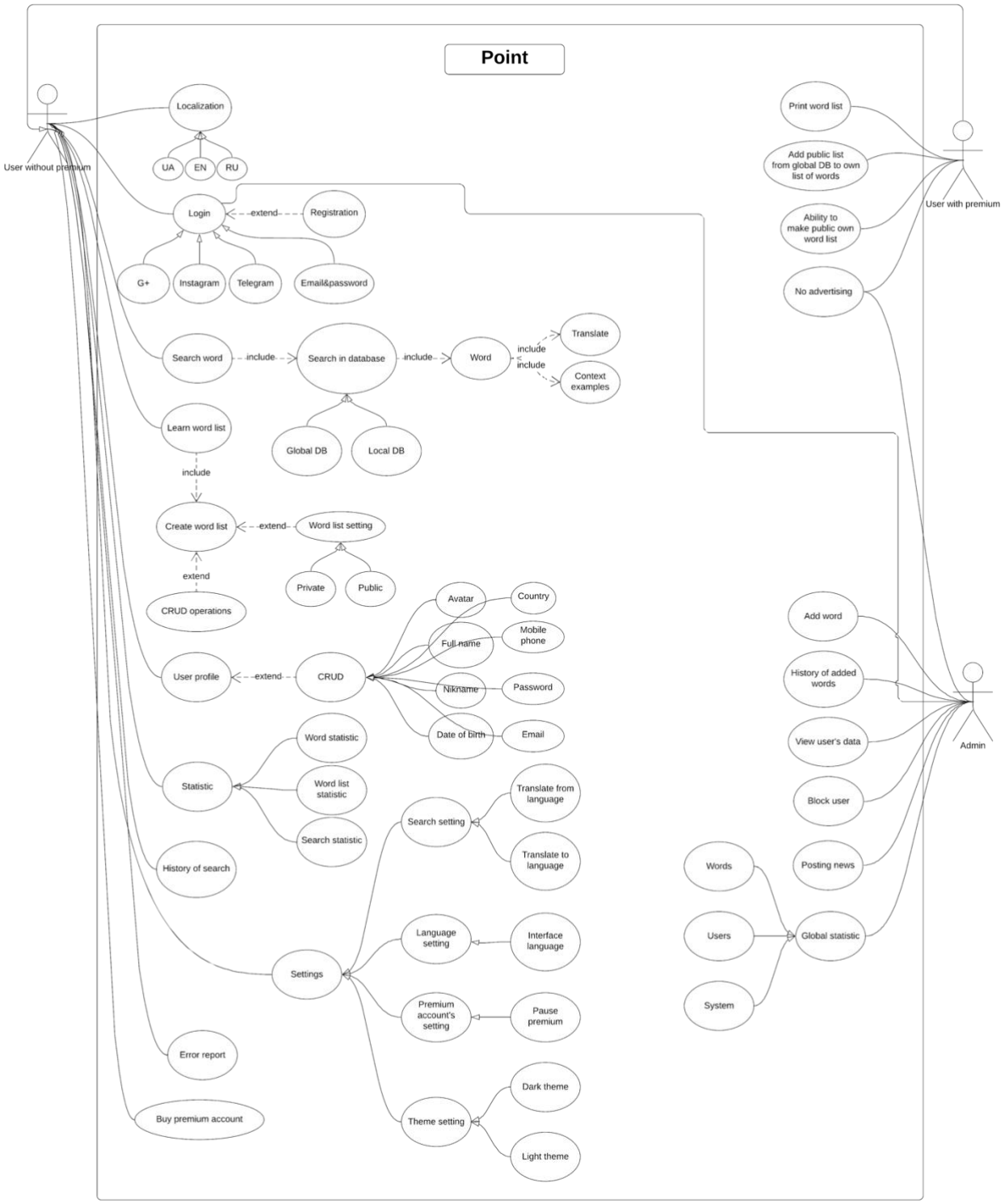


Рисунок 12 – Use Case діаграма для розроблювальної системи

На рисунку 12 зображена діаграма системи, яка має назву Point, та яка показує всі можливості системи.

У системі Point буде 3 актори:

а) звичайний користувач;

б) користувач, який має преміум обліковий запис;

в) адміністратор.

Також є третій тип користувачів – незареєстрований користувач. Цей тип користувача має змогу подивитися першу презентаційну сторінку системи Point, змінити мову інтерфейсу сайту та зареєструватися за допомогою соціальної мережі або електронної скриньки і паролю.

Для повного розуміння побудованої use case діаграми, треба розібрати основні моменти та умови, які зображені на діаграмі, розповісти про можливості системи та описати вірогідні дії акторів та їх відмінності одного від одного.

Основний або найбільш розповсюджений тип користувачів — це звичайний користувач, який зображений на use case діаграмі під назвою «user without premium». Як стає зрозуміло з назви актора — цей користувач не має облікового запису типу преміум, але має змогу його придбати. Припускається, що такий користувач вже зареєстрований у системі або має можливість зареєструватися. Для користувачів такого типу на сайті показується реклама різних видів.

Користувач «user without premium» має такий набір можливих дій:

а) зареєструватися;

б) виконати вхід в систему;

в) змінити мову інтерфейсу системи;

г) шукати слова;

ґ) шукати словосполучення;

д) вибрати тип бази даних для пошуку (глобальна або локальна);

е) створити список слів (тільки приватний);

є) переглянути свій обліковий запис і, при необхідності, змінити дані;

ж) переглянути статистику;

з) переглянути історію пошуку;

и) скористатися налаштуваннями системи: мова інтерфейсу, пошукова система, тема сайту;

і) повідомити про знайдену помилку у системі;

й) придбати обліковий запис типу преміум.

Зареєструватися у веб сервісі Point можна використовуючи звичайну електрону скриньку та пароль або скористатися одним із трьох соціальних мереж:

– Google+;

– Facebook;

– GitHub.

Виконувати вхід в систему можна через цифрову скриньку і пароль або використовуючи соціальні мережі.

Система підтримує три мови локалізації інтерфейсу:

– українська;

– англійська;

– російська.

Мова за замовчуванням вибирається таким чином: якщо користувач знаходиться в Україні, то вибирається українська мова, якщо у Росії, то російська, якщо у будь-якій іншій, то мова буде англійська.

Звичайний користувач має змогу використовувати дві бази даних для пошуку слів:

а) глобальна БД;

б) локальна БД.

Перемикання мов відбувається просто: треба натиснути на кнопку з позначкою земного шару або людини, залежить від вибраного варіанту БД, на панелі пошуку.

Може статися, що необхідного слова у глобальній БД не має, тоді треба додати необхідне слово у локальну базу даних. Щоб потім знайти його, треба на панелі пошуку вибрати відповідну базу. Також у веб сервісу Point є функція, яка має назву: «запропонувати слова». Будь-який користувач, крім адміністратора, може скористатися цією можливістю та покращити систему Point, запропонувавши слово або цілий список слів, яких не має у глобальній базі даних.

Кожен користувач, включаючи адміністратора, може створювати списки слів. Але в системі Point, стосовно можливості створення списків та їх типів є відмінність між користувачами. Списки поділяються на такі типи:

– приватні;

– публічні.

Приватні списки мають змогу створювати звичайні користувачі та з обліковим записом типу преміум.

Публічні списки поділяються на глобальні та локальні.

Локальні публічні списки можуть створюватися лише користувачами з обліковими записами типу преміум.

Глобальні публічні списки можуть створюватися тільки адміністратором. Список такого типу використовується як еталон гарного публічного списку, такі списки слів виступають шаблонами з різних тем, наприклад: дім, місто, ліс і т.д. Також слова з цих списків використовуються у ролі відповідей на запити у пошуковій строфі, тобто вони складають глобальну базу даних веб сервісу Point.

Статистика для звичайного користувача поділяється на такі підрозділи:

– статистика з пошуку слів;

– статистика зі створення та користування списками слів;

– статистика з вивчення для кожного слова;

– статистика з вивчення для кожного списку слів.

Історія пошуку дозволяє легко знайти слова, які ви шукали за весь час використання веб сервісу Point, якщо вона не була видалена. Меню перегляду історії пошуку слів включає сортування, фільтрування та видалення всієї історію або конкретного слова.

Преміум обліковий запис значно поширює можливості користувача веб сервісу Point, а саме надає такі привілеї:

– списки слів можна відредагувати та вивести на друк;

– відкривати свої локальні списки слів (робити їх публічними);

– користуватися публічними списками інших користувачів та системи;

– позбутися реклами на сайті.

Розглянемо користувача, який має обліковий запис типу преміум. На побудованій діаграмі use case (див. Рис. 12, стор. 42) він зображений як актор та має назву: «User with premium». Такий користувач має всі можливості звичайного клієнта системи («user without premium») і преміум можливості. Всі преміум можливості системи Point:

- друк слів;
- публічні списки слів;
- використання публічних списків системи та користувачів;
- відсутність реклами.

Користувач, який має найбільше привілеїв — це адміністратор. Адміністратор зображений на діаграмі use case як актор та має назву «Admin». Адміністратор має такі можливості:

а) додавати слова у глобальну базу даних;

б) дивитися історію додавання слів до БД;

в) перегляд даних користувача та їх редагування;

г) блокування користувачів;

ґ) публікація новин на сайті;

д) глобальна статистика.

Глобальна статистика – це статистика використання слів, діяльності користувачів, кількість слів і користувачів та показники роботи системи.

Для повного розуміння роботи веб сервісу Point побудуємо UML deployment diagram, яка зображена на рисунку 13 (див. стор. 47).

UML deployment diagram — це діаграма, яка показує конфігурацію вузлів і компонентів, які в них знаходяться. Deployment diagram — це вид структурної діаграми, яка використовується для моделювання фізичних аспектів об’єктно орієнтованої системи.

Існує лише один спосіб створити програмне забезпечення або обчислювальні системи, і це архітектура програмного забезпечення. Іншими словами, це абстракція елементів системи на певному етапі їх функціонування [7]. Як правило,

система складається з кількох рівнів абстракції з кількома фазами роботи, кожна з яких також має або може мати свою власну незалежну архітектуру.

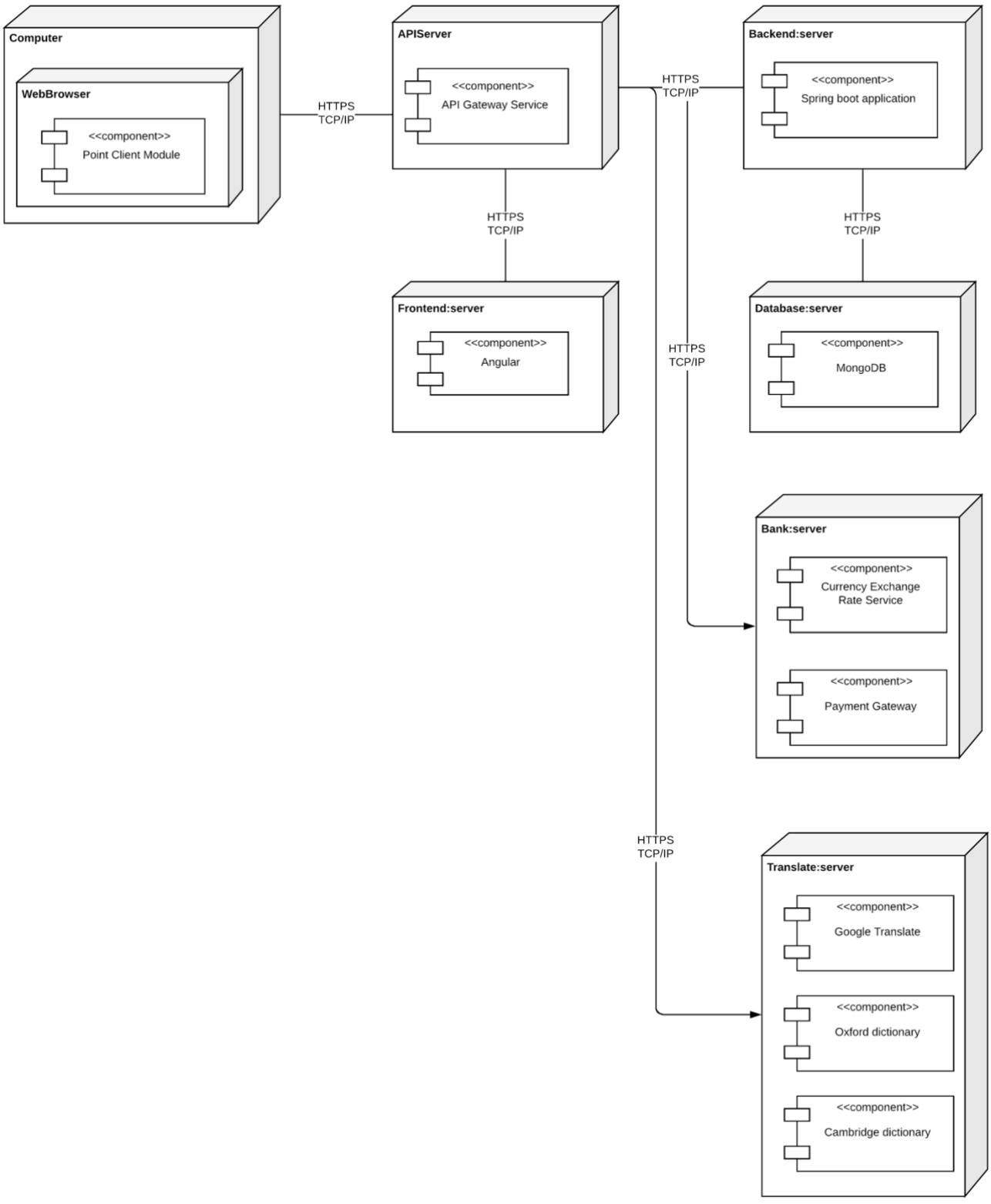


Рисунок 13 – UML deployment diagram

Для коректного функціонування нашої системи, масштабованості та гарної організації роботи тонкого клієнта був обраний архітектурний шаблон трирівневої архітектури, який складається з наступних компонентів:

- браузер або клієнтська програма, наприклад браузер Safari;
- серверна система, що підтримує функції бекенда нашої системи;
- база даних NoSql існує як окремий сервер.

Клієнтська програма, або «тонкий клієнт», як зазвичай кажуть, термінал) надсилає запит до серверного сервера, який, у свою чергу, спілкується з сервером NoSql для бази даних.

Якщо порівнювати нашу архітектуру з архітектурою клієнт-сервер або файл-сервер, то можна виділити наступні переваги обраної (трирівневої) архітектури:

- висока масштабованість;
- істотне налаштування: це ізоляція між рівнями, що дозволяє швидко і легко переналаштувати систему в разі збою або регулярного обслуговування на одному з рівнів;
- значний рівень безпеки;
- надійність;
- значно нижчі вимоги до швидкості каналу/мережі між терміналом і сервером додатків;
- продуктивність і технічні характеристики терміналу можуть бути низькими, що дозволяє користувачам зі слабким ПК використовувати систему без істотних перешкод. Терміналом може бути: комп’ютер, мобільний телефон, будь-яка система, яка підтримує доступ до Інтернету та браузер.

Система сервер, на якому знаходиться бекенд вебсервісу «Вільний словник-перекладач ІТ термінів» використовується для синхронізації даних, їх централізації. На цьому сервері міститься основна бізнес логіка проєкту.

Бекенд зв’язується з сервером бази даних, яка містить дані про користувача, слова, списки слів та мови. Доступ до БД охороняється протоколами безпеки.

Всі сервери є автономними, тобто вони не залежать один від одного.

На сьогодні існує лише два підходи до програмної розробки вебдодатків, а саме: монолітна та мікросервісна архітектура. Для дипломного проєкту вебсервіс «Вільний словник-перекладач IT термінів» була обрана монолітна архітектура.

Система буде побудована згідно архітектури MVC — Model-View- Controller. Де Model відповідає за доступ до інформації та бізнес-логіку програми, View – за представлення, яке отримує клієнт у браузеру і Controller, який відповідає за обробку запитів і повернення відповідних даних клієнтам. Для функціонування системи буде використовуватися REST підхід.

REST — це підхід до архітектури мережевих протоколів, які надають доступ до інформаційних ресурсів. Дані повинні передаватися у вигляді невеликої кількості стандартних форматів (наприклад, HTML, XML, JSON).

Також моя система використовую алгоритми, які були описані та розібрані і розділах вище, для удосконалення перекладання та поліпшення пошуку слів з іноземної мови. У моїй програмі були використані саме алгоритм корекції точності перекладу та алгоритм подолання розриву між зіставленням релевантності та семантичним зіставленням для моделювання схожості коротких текстів, який був використаний із застосуванням Facebook AI Meta API, яке на даний час є платним комерційним програмним забезпеченням.

ВИСНОВКИ

У ході виконання магістерської роботи було визначено проблему ефективності архітектурних рішень для поліпшення пошуку слів з іноземної мови.

Були описані такі алгоритми поліпшення пошуку слів:

- алгоритм корекції точності перекладу;
- алгоритм подолання розриву між зіставленням релевантності та семантичним зіставленням для моделювання схожості коротких текстів.

Був зроблений аналіз результатів дослідження для алгоритму корекції точності перекладу та для алгоритму, який був розроблений командою Facebook Meta AI. У обох аналізах було порівняно результат з алгоримом та без нього.

Я розглянув взаємозв'язок між зіставленням релевантності та семантичним зіставленням і виділив кілька важливих відмінностей між ними. Це важлива проблема, яка лежить в основі багатьох завдань НЛП та IP.

Ретельні експерименти показали, що для багатьох завдань НЛП досить добре працює лише релевантне зіставлення, тоді як семантичне зіставлення не є ефективним для завдань IP. Ми показали, що зіставлення релевантності та семантичне зіставлення є взаємодоповнюючими, і HCAN поєднує в собі найкраще з обох світів для досягнення конкурентної ефективності у великій кількості завдань, в деяких випадках досягаючи найкращих результатів для моделей, які не використовують масштабне попереднє навчання.

Також була розроблена власна система, яка об’єднує вже існуючи інструменти в один та дозволяє не тільки робити переклад із застосуванням описаних алгоритмів поліпшення пошуку слі, а й дивитися тлумачення слів з можливістю створювати цілі списки слів для вивчення.

Аналогів такої системи на ринку немає, тому вона буде успішною. Монетизація буде відбуватися за допомогою контекстної реклами у безкоштовній версії системи і за допомогою впровадження преміум підписки з певним функціоналом, який розширює можливості.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Bai Y., Hu R., Currey A. Auto machine translation and synchronization for "dive into deep learning". *Amazon Science*. URL: https://www.amazon.science/blog/auto-machine-translation-and-synchronization-for-dive-into-deep-learning(date of access: 07.01.2023).
2. DeepL Translate: the world's most accurate translator. *DeepL Translate: The world's most accurate translator*. URL: https://www.deepl.com/en/translator(date of access: 07.01.2023).
3. DeepL translator - google translator competitor - hashdork. *HashDork*. URL: https://hashdork.com/deepl-translator/(date of access: 07.01.2023).
4. Everything you need to know about google translate. *Scientific Editing*. URL: https://www.scientific-editing.info/blog/everything-you-need-to-know-about-google-translate/(date of access: 07.01.2023).
5. Ревенчук І.А., Перцьова К.В., Маренич О.І. Програмна реалізація кластеризації пошукових запитів.-Біоніка інтеллекту.-№.-2(91)2018.-С.86-93
6. Jaludi M. Google translate. *Medium*. URL: https://mariam-jaludi.medium.com/google-translate-b6ad6328e7f2(date of access: 07.01.2023).
7. Machine translation service – amazon translate – amazon web services. *Amazon Web Services, Inc.*URL: https://aws.amazon.com/translate/(date of access: 07.01.2023).
8. What is amazon translate? - amazon translate. URL: https://docs.aws.amazon.com/translate/latest/dg/what-is.html(date of access: 07.01.2023).
9. Sus, B., Tmienova, N., Revenchuk, I., Vialkova, V. Development of virtual laboratory works for technical and computer sciences.- Communications in Computer and Information Science, 2019, 1078 CCIS, P. 383–394.
10. Sus, B., Tmienova, N., Revenchuk, I., Bauzha, O., Stirenko, S. Gamification approach to the creation of virtual laboratory works and educational courses.- CEUR Workshop Proceedings, 2020, 2711, P. 68–78.
11. Sus, B., Revenchuk, I., Tmienova, N., Bauzha, O., Chaikivskyi, T. Software System

for Virtual Laboratory Works.- 2020 IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2020 - Proceedings, 2020, 1, Pp. 396–399, 9322046.

12. Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Don- ald Metzler, Mark D. Smucker, Trevor Strohman, Howard Turtle, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. In Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004), Gaithersburg, Maryland.
13. Christopher J. C. Burges. 2010. From RankNet to LambdaRank to LambdaMART: An overview. Technical Report MSR-TR-2010-82, Microsoft Re- search.
14. Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading Wikipedia to answer open- domain questions. In Proceedings of the 55th An- nual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1870– 1879, Vancouver, Canada.