Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження та застосування методів глибинного навчання у задачах соціалізації людей з афазією
(тема)

Виконав:
студент 2 курсу, групи СШІм-21-2
Бєляєв В. С.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

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

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник проф. каф. ШІ Рябова Н. В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

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

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Бєляєву Владиславу Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та застосування методів глибинного навчання у задачах соціалізації людей з афазією

затверджена наказом університету від 31 березня 20 23 р. № 306Ст

2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 20 23 р.

3. Вихідні дані до роботи Проведення аналізу предметної галузі, виявлення наявних проблем соціалізації людей, хворих афазією та аналіз існуючих систем, які допомагають у спілкувані людям з афазією. Дослідження наукових статей про різні методи розв'язання задачі машиного перекладу. Визначення метрики для порівняння якості перекладу зробленого нейронними мережами. Програмне забезпечення: IDE-засіб VisualStudio Code, IDE-засіб Google Colab; бібліотеки для глибокого навчання: tensorflow, trax.

4. Перелік питань, що потрібно опрацювати в роботі Аналіз предметної галузі. Аналіз реалізованих систем. Визначення формату вхідних та вихідних даних для навчання нейронних мереж. Аналіз наявних методів для розв'зання задачі машинного перекладу. Порівняльне дослідження визначених методів, нейронних мереж. Розробка моделей визначених нейронних мереж. Порівняльний аналіз якості перекладу розроблених моделей на реченнях різної довжини.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Приклад взаємодії шару уваги із кодувальником та декодером, Рисунок 2 – Представлення реалізації уваги «Scaled Dot-Product Attention», Рисунок 3 – Представлення блоку gated recurrent units, Рисунок 4 – Представлення блоку long short-term memory, Рисунок 5 – Представлення архітектури створеної рекурентної нейронної мережі з шаром уваги, Рисунок 6 – Загальна архітектура нейронної мережі «Transformer», Рисунок 7 – Графічне зображення шару «Multi-head attention».

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Визначення проблеми	04.04-05.04.2023	Виконано
2	Проведення аналізу предметної області	06.04-09.04.2023	Виконано
3	Аналіз реалізованих систем	10.04.2023	Виконано
4	Дослідження наукової літератури даних	11.04-17.04.2023	Виконано
5	Визначення формату вхідних та вихідних даних	18.04-19.04.2023	Виконано
6	Аналіз наявних наборів даних та створення набору даних	19.04-24.04.2023	Виконано
7	Аналіз наявних методів розв'язання задачі	25.04-27.04.2023	Виконано
8	Розробка нейронних мереж для машинного перекладу	28.04-05.05.2023	Виконано
9	Аналіз отриманих результатів якості перекладу мережами	06.05.2023	Виконано
10	Оформлення пояснювальної записки	07-12.05.2023	Виконано
11	Захист перед ЕК	18.05.2023	

Дата видачі завдання 3 квітня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ проф. каф. ШІ Рябова Н. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 60 с., 17 рис., 2 табл., 3 дод., 21 джерел.

АФАЗІЯ, ГЛИБИННЕ НАВЧАННЯ, ОБРОБКА ПРИРОДНЬОЇ МОВИ, ATTENTION, BEAM SEARCH, BLEU, GRU, LSTM, RNN, SEQ2SEQ, TRANSFORMER

Об’єктом дослідження є процес спілкування людей, хворих афазією з іншими.

Предметом дослідження є глибокі нейронні мережі та інформаційні технології, що дозволяють розпізнавати природну мову для перетворення її у зрозумілий вид для людей, хворих афазією.

Мета дослідження: визначити методи глибинного навчання, які розв’язують задачу машинного перекладу, перевірити можливість використання наявних методів на наборі даних з реченнями зі смайлів, визначити модель з найкращою якістю перекладу, шляхом порівняння декількох моделей за якістю перекладом на реченнях з різною довжиною.

Проведено аналіз літературних джерел для виявлення наявних методів глибинного навчання для задачі машинного перекладу. Було реалізовано три різні моделі нейронних мереж, які роблять переклад речень. У результаті досліджень та порівняння якості перекладу було визначено архітектуру нейронної мережі, яка може перекладати речення довжиною до 30 слів зі збереженням змісту, але з граматичними помилками.

Визначені нейронні мережі можна буде застосовувати у мобільному додатку або розгорнути в хмарному середовищі для використання в інформаційній системі, яка буде допомагати у спілкуванні людям з афазією.

ABSTRACT

Explanatory note: 60 p., 17 fig., 2 tabl., 3 ann., 21 sources.

APHASIA, ATTENTION, BEAM SEARCH, BLEU, DEEP LEARNING, GRU, LSTM, NLP, RNN, SEQ2SEQ, TRANSFORMER

The object of research is the process of communication of people with aphasia with others.

The subject of research is deep neural networks and information technologies that allow recognizing natural language to transform it into an understandable form for people with aphasia.

The purpose of the study: is to determine the methods of deep learning that solve the problem of machine translation, to check the possibility of using these methods on a data set with sentences of emojis, to determine the model with the best translation quality by comparing several models for the quality of translation on sentences of different lengths.

An analysis of literary sources was carried out to identify existing methods of deep learning for the task of machine translation. Three different models of neural networks that translate sentences were implemented. As a result of research and comparison of the translation quality, a neural network architecture was determined that can translate sentences up to 30 words long with the content preserved, but with grammatical errors.

The identified neural networks can be applied in a mobile application or deployed in a cloud environment for use in other information systems that will help people with aphasia communicate.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Афазія – розлад, що виникає внаслідок пошкодження ділянок мозку, які відповідають за промову;

МРТ – магнітно-резонансна томографія;

США – Сполучені Штати Америки

ACC – augmentative and alternative communication – підсилювальна та альтернативна комунікація;

BLEU – BiLingual Evaluation Understudy – метрика для автоматичної оцінки реузльтату машинного перекладу;

GRU – gated recurrent units – керований рекурентний блок;

LSTM – long short term memory – довга коротко-строкова пам’ять;

NLP – natural language processing – обробка природньої мови;

ReLu – Rectified Linear Unit – блок лінійної функції випрямлення;

RNN – recurrent neural network – рекурентна нейронна мережа;

Seq2Seq – sequence-to-sequence – модель глибинного навчання, яка перетворює вхідну послідовність символів у іншу послідовність символів.

ВСТУП

У сучасному світі майже кожна людина взаємодіє із додатком з нейронною мережею. Вони використовуються для розв’язку повсякденних проблем, проблем безпеки тощо. Також нейронні мережі використовуються й у медицині, допомагаючи лікарям при визначені діагнозу, лікуванні або лікувальної терапії, допомоги людям з різними хворобами у повсякденному житті.

Афазія одна із хвороб, яку можливо визначити за допомогою таких додатків. Цей розлад виникає внаслідок пошкодження ділянок мозку, які відповідають за промову та погіршує можливість людини розмовляти та розуміти мову в різних її видах. На жаль існує багато видів афазії з різними особливостями, які впливають на спілкування людини з іншими, її повсякденне життя. Зазвичай афазія виникає у людей похилого віку, бо найчастіше вона виникає після інсульту або черепно-мозкової травми. Але кількість людей, які мають цей розлад, збільшується кожен рік більше ніж на 200 тисяч. Тим паче, що в Україні очікується зменшення середнього віку людини, яка перенесла інсульт, із-за війни.

Легкі форми афазії лікуються або можуть зникнути самостійно через деякий час, але люди з тяжкими видами можуть не розуміти мову та мати складнощі навіть вимовляти слоги слів. За допомогою сучасних технологій існує можливість допомогти їм у вирішені повсякденних проблем з комунікацією та дати можливість повернутися у соціальне життя.

Метою є дослідити можливості нейронних мереж для вирішення отриманих задач та визначити, які найкраще підходять для подальшого використання в інформаційній системі, яка буде вже взаємодіяти з користувачем.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

На жаль сьогодні понад вісімдесяти відсотків людей ніколи не чули про таку хворобу як афазія, хоча вже більше чотирьох мільйонів людей мають цей розлад. Афазія – це розлад, що виникає внаслідок пошкодження ділянок мозку, які відповідають за промову, тобто цей розлад сприяє на можливість людини розмовляти та розуміти мову як в усному вигляді, так і в письмовому [1]. Цей розлад виникає у разі пошкодження однієї або кількох ділянок мозку людини, які відповідають за мовні функції. Лікарі виділяють наступні основні причини виникнення афазії:

– інсульти, тобто хронічне порушення мозкового кровообігу;

– черепно-мозкові травми;

– пухлини у голові;

– при захворюванні на енцефаломієліт;

– при порушенні будови тканин мозку, тобто при хворобі Альцгеймера, хворобі Піка, хворобі Крейтцфельдта-Якоба та інших;

– при ускладненні нейрохірургічних операцій;

– при отруєнні важкими отрутами.

Можна сказати, що найчастіше афазія виникає після інсульту, адже за статистикою кожен четвертий інсульт у США приводить до цього розладу.

Афазія, як і інші хвороби, має декілька видів з різними особливостями. Виділяють дві категорії афазії, які ще поділяються на декілька типів. Найпоширенішим типом текучої афазії є афазія Верніке, яка виникає при пошкодженні скроневої частки мозку, що зображено на рисунку 1.1. При такій афазії людиможуть говорити довгими реченнями, додаючи непотрібні або вигадані слова, отже ці речення не мають очевидного значення. Також люди з таким видом афазії мають труднощі з

розумінням мови й часто не усвідомлюють своїх розмовних проблем. Наприклад є інша форма текучої афазії, аномічна афазія, яка є легшою. Люди з таким видом розладу розуміють інших та можуть вільно спілкуватися, але при цьому часто використовують доповнювальні слова та іноді шукають синоніми, інші слова для опису слова, які забули або не можуть виговорити.

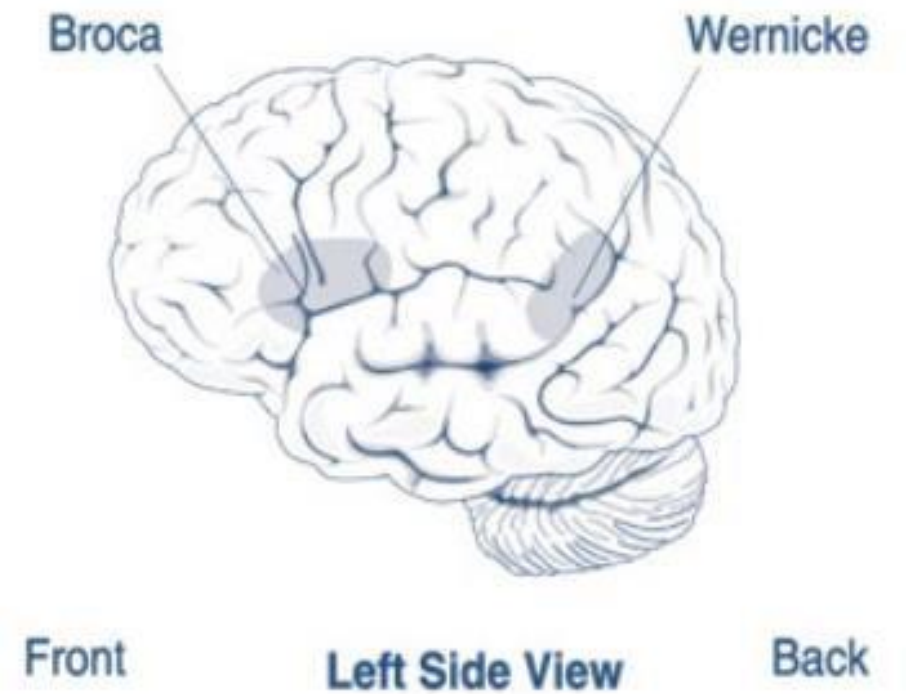


Рисунок 1.1 – Ділянки мозку, уражені афазією Брока та Верніке

Афазія Брока є найпоширенішим типом нетекучої афазії, ці люди розуміють мову і часто говорять короткими фразами з великим зусиллям, часто пропускають маленькі слова, прийменники, при цьому вони знають, що хочуть сказати. На відмінно від афазії Верніке, люди з афазією Брока усвідомлюють свої труднощі, а також ще мають проблеми з руховим апаратом, бо пошкоджена ділянка мозку також частково відповідає за роботу кінцівок людини.

Найважчим типом афазії вважається глобальна, адже для її виникнення необхідне пошкодження великих частин мовних областей

мозку. При такій афазії люди взагалі можуть бути не в змозі сказати навіть кілька простих слів та мають великі проблеми із розумінням інших [2]. Існують й інші типи афазії, кожна з яких має свої наслідки, все залежить від частин мовних областей мозку, які були пошкоджені.

Отже, люди, з розладом афазія, стикаються повсякденно з проблеми, які пов’язані з комунікацією з іншими людьми у магазинах, кав’ярнях, лікарнях тощо. За допомогою використання сучасних технологій, у тому числі й нейронних мереж, можна допомогти таким людям у реабілітації та розв'язання повсякденних проблем.

1.2 Актуальність розглянутої проблеми

Сьогодні існує багато різних сервісів, додатків окрім МРТ для виявлення афазії на ранніх стадіях розладу, адже найчастіше він виникає у людей після інфаркту або черепно-мозкової травми. Більша кількість таких сервісів використовують нейронні мережі по розпізнаванню природної мови длявизначення чи є у людини афазія та який її тип [3]. Також є багато реабілітаційних сервісів, які допомагають людям з такою хворобою відновити свої мовні можливості [4]. Адже бувають випадки коли при деяких типах афазії можливо відновити повністю або зменшити вплив цього розладу на мовні здібності. Наприклад, інформаційні системи, які генерують повсякденні зображення та перевіряють чи вірно людина назвала об’єкт, який був виділений або з участю іншої людини описує зображення.

На жаль кількість людей, які мають афазію збільшуються з кожним роком, існує невелика кількість сервісів, які допомагають у спілкуванні, розв'язання проблем соціалізації таких людей. На сьогодні методи глибинного навчання дозволяють створити інформаційну систему з нейронними мережами, які будуть використовуватися для обробки природної мови, для трансформування мовних речень у послідовність

цифрових зображень, наприклад смайлів, при цьому зберігаючи зміст речення. А також й надавати можливість людям з афазією висловлювати свої думки у вигляді послідовності смайлів, а потім перетворювати їх у текстові речення, що допоможе у вирішенні повсякденних проблем пов’язаних з комунікацією людей з таким розладом. Адже, наприклад, за статистикою лише у США кожного року кількість людей з різними видами афазії збільшується більше ніж на 80 тисяч, а ще очікується зменшення середнього віку українця з інфарктом у найближчі роки, що може призвести до збільшення кількості людей, які будуть мати розлад афазії.

Отже, нейронні мережі, які будуть трансформувати текстові речення у речення з емодзі, зможуть спростити процес спілкування для більшості людей, хворих афазією, тобто розв’язати деякі їх проблеми, наприклад, взаємодію чи спілкування в лікарнях, державних органах, у магазині при виборі чи покупці необхідних речей тощо.

1.3 Аналіз реалізованих систем

На цей час існують декілька мобільних додатків, які допомагають у спілкуванні та лікуванню людям, які мають проблеми з мовними можливостями, а саме:

– додатки компанії Lingraphica;

– «Proloquo2Go AAC».

Компанія Lingraphica пропонує різні мобільні додатки, які допомагають у спілкуванні з іншими та є безплатними. Кожен з цих додатків пропонує набір фраз із зображенням, які об’єднані за тематикою спілкування (рисунок 1.2). Тобто один додаток допомагає має повсякденні прості словосполучення, які допомагають при спілкуванні з лікарем, інший же додаток має свій власний набір словосполучень, які належать до повсякденної активності людини.

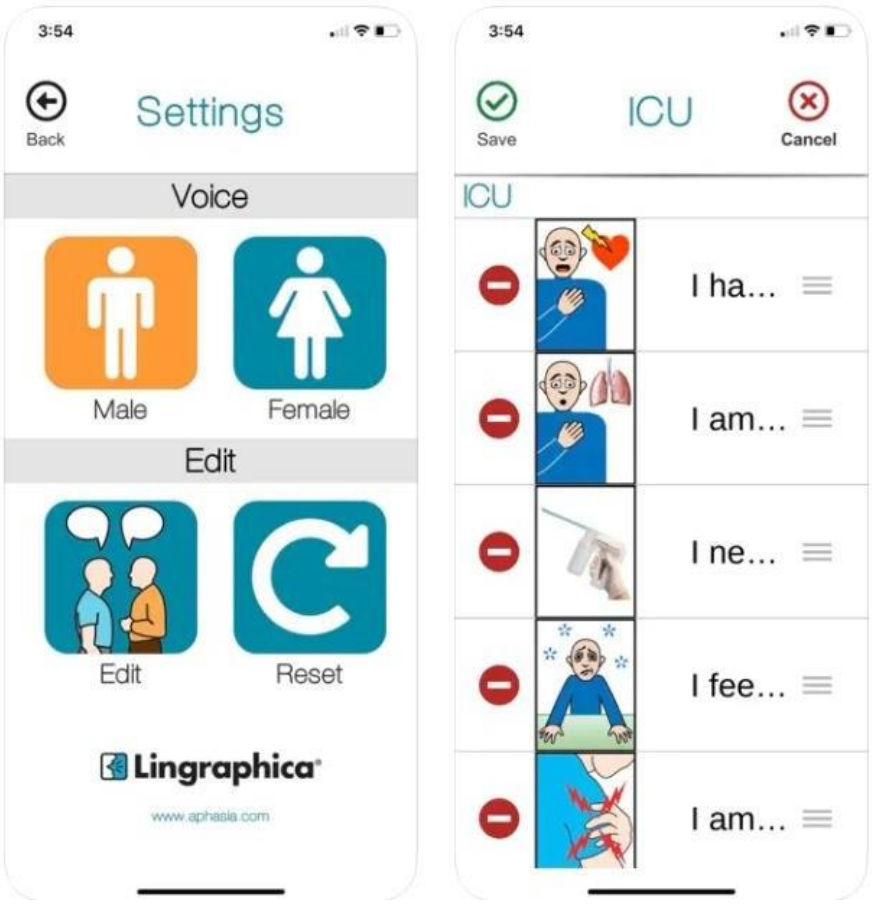


Рисунок 1.2 – Приклади екранів мобільного додатку «SmallTalk Intensive Care» компанії Lingraphica

Інший мобільний додаток «Proloquo2Go AAC» (рисунок 1.3) має власний набір зображень декількома мовами для спілкування з іншими людьми. Він також може озвучувати набрану послідовність зображень різними голосами як за статтю, так і за віком. Відмінно від інших додатків «Proloquo2Go AAC» дозволяє створювати групи зображень для швидкої навігації між ними. Цей додаток дає більше можливостей у спілкуванні з іншими людьми, адже дозволяє створювати послідовність зображень у ньому використовуючи текстовий редактор. Але він лише дозволяє використовувати слова, які мають зображення. Отже, додаток «Proloquo2Go AAC» має більше функцій, які допомагають у спілкуванні з іншими людьми, але він має і недолік – лише люди із встановленим цим додатком можуть спілкуватися на відстані.

Рисунок 1.3 – Приклади екранів мобільного додатку «Proloquo2Go AAC»

Розглянемо також можливості чат-бота зі штучним інтелектом, який вийшов у вільний для використання доступ, а саме «ChatGPT». Через те, що йому потрібно писати завдання, то використовувати його можливо лише для трансформації текстових речень у послідовність смайлів. Крім цього, ми можемо побачити з рисунка 1.4 те, що чат-бот перетворив текстові речення у послідовність смайлів, зробив препроцесінг вхідних речень, а сами видалив артиклі та граматичні конструкції часу в англійській мові.

Проте, на мою думку, на сьогодні він не навчаний розв’язувати таку задачу машинного перекладу, бо досить складно зрозуміти зміст речення, яке вийшло зі смайлів. Чат-бот лише зміг замінити декілька слів на емодзі,

які добре описують їх значення, це можуть і декілька вже розроблених нейронних мереж, які якраз навченні для того, щоб заміняти словосполучення або слова на емодзі.

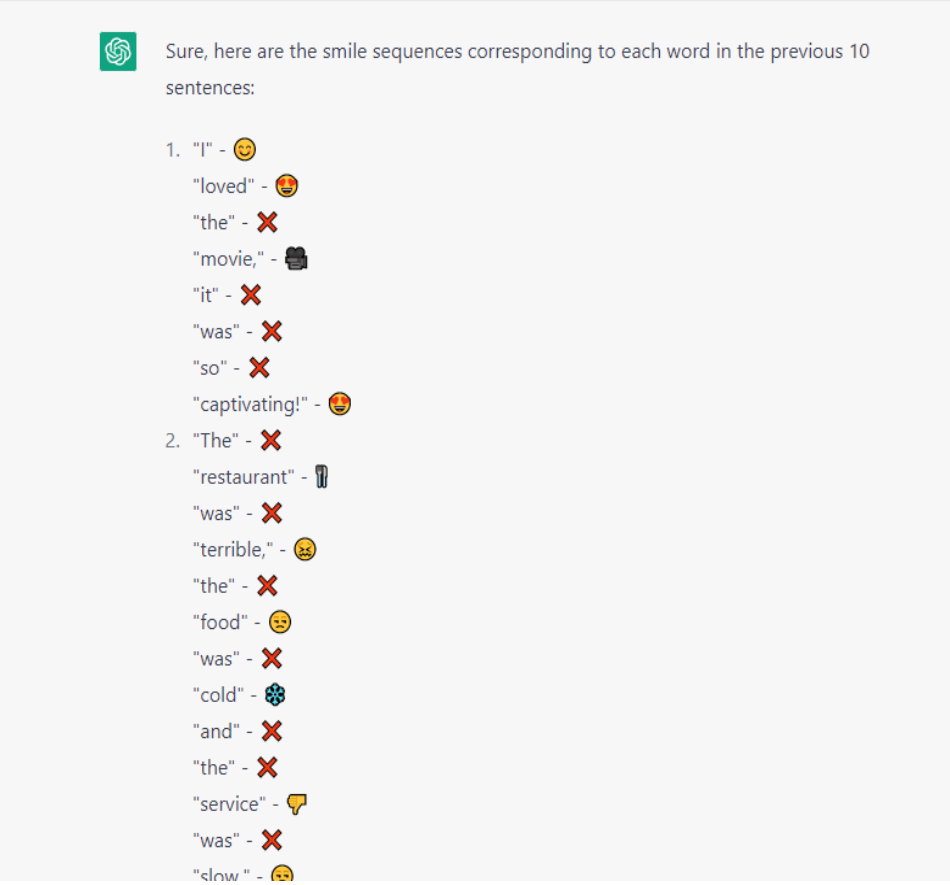


Рисунок 1.4 – Приклад трансформування текстового речення у послідовність смайлів штучним інтелектом «ChatGPT»

1.4 Постановка задачі

У рамках даної роботи необхідно провести дослідження можливих варіантів, моделей глибинного навчання, які підходять для розв’язку задачі по трансформування текстового речення англійською мовою у речення, яке складається з емодзі та відображає зміст текстового, та іншої задачі, яка дзеркальна попередній, тобто з послідовності смайлів у текстовий вигляд.

Ці обидві задачі є видом задачі машинного перекладу по перекладу послідовності слів з однієї мови в іншу. Тому пошук та аналіз існуючих методів розв’язку задачі машинного перекладу буде початковою точкою цього дослідження.

Крім пошуку методів розв’язку задачі необхідно буде визначити формат даних для навчання знайдених варіантів та подальшої взаємодії з ними. Тобто як відображати емодзі у вхідних даних.

Як результат дослідження, ми надаємо порівняльну характеристику якості перекладу послідовностей різної довжини досліджуваних моделей нейронних мереж, що дозволить визначити метод, який найбільше підходить до розв’язку задачі з таким форматом вхідних даних.

Для досягнення поставлених цілей необхідно виконати такі етапи:

– пошук та аналіз наявних наборів вхідних даних для машинного перекладу для використання у цьому дослідженні, створення власного набору для досліджування при необхідності;

– пошук та аналіз наукових статей, книг, доповідей, публікацій тощо, які описують підходи розв’язку схожих задач;

– визначення моделей та архітектур нейронних мереж, які підходять для подальшого дослідження та порівняння, на основі проаналізованих літературних джерел;

– описати математичні моделі досліджувальних архітектур та моделей нейронних мереж;

– провести програмну реалізацію досліджувальних моделей;

– визначити метрику для оцінювання якості розроблених нейронних мереж;

– провести перевірку якості реалізованих моделей;

– провести аналіз отриманих результатів, порівняти їх між собою та визначити найкращу модель, яка розв’язує поставленні задачі.

2 ДОСЛІДЖЕННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ МАШИННОГО ПЕРЕКЛАДУ

2.1 Визначення формату набору даних для нейронних мереж

Вхідними даними для нейронної мережі, яка трансформує текстові речення у речення зі смайлів будуть текстові речення, а вихідними – речення з індикаторів смайлів. Ці ж дані можна буде використати для навчання другої нейронної мережі, адже вона виконує трансформування але в інший бік. Тобто кожний смайл повинен мати власний текстовий ідентифікатор, який описує його значення, та відповідати слову або словосполученню (рисунок 2.1), адже в кожній мові є слова, мовні конструкції. Отже, перед генерацією даних для навчання та тестування спочатку необхідно визначити перелік смайлів, можна сказати створити окрему мову з них для використання у реченнях.

Рисунок 2.1 – Приклад візуалізації текстового речення у додатку «Proloquo2Go AAC»

Було зроблено пошук існуючого набору даних, який би складався з текстового речення та послідовності емодзі, які описують зміст текстового речення, але було знайдено тільки набори даних, які описують значення кожного емодзі окремо, що не підходить для використання у даному дослідженні. Тому було вирішено створити власний набір даних для навчання нейронних мереж у дослідженні.

Спочатку необідно було згенерувати текстові речення, для цього було використано декілька джерел. За допомогою чат-бот «ChatGPT» було згенеровано 1000 речень англійською мовою (рисноук 2.2), ще 200 речень було придумано самостійно, які можуть використовуватися у повсякденному житі. Було обрано для дослідження зробити невеликий набір даних, який орієнтований на спілкування людей, хворих афазією, в ресторані та лікарні.

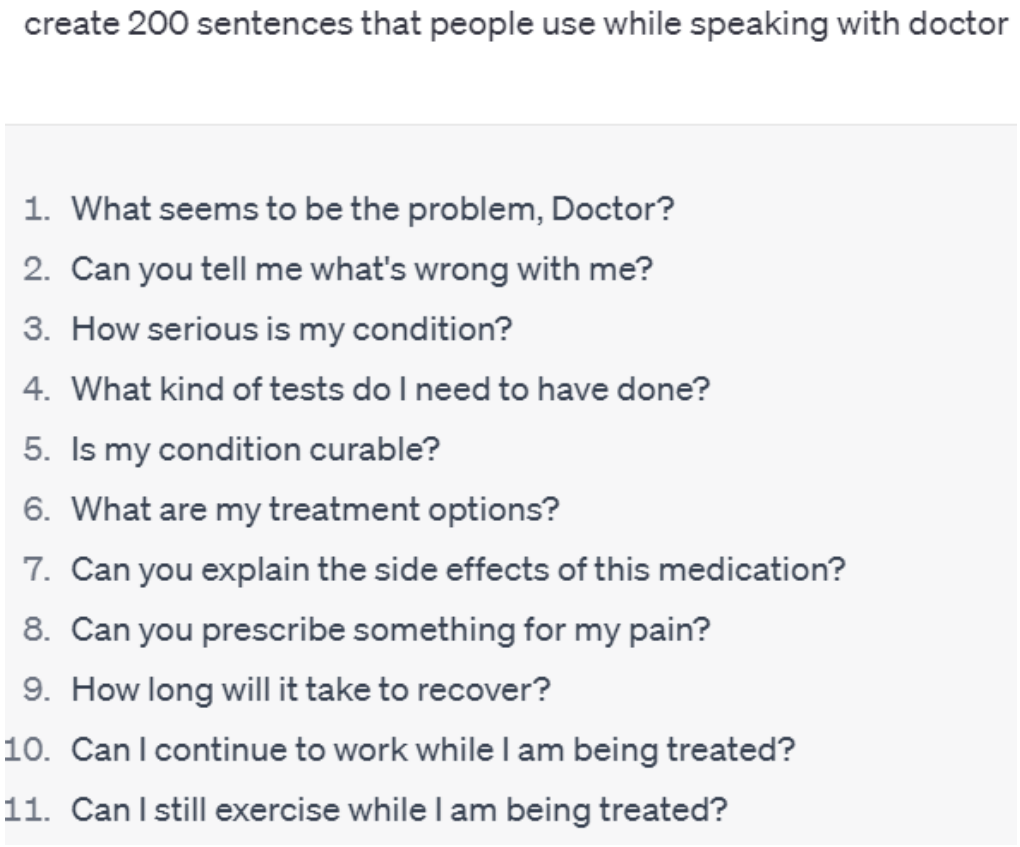
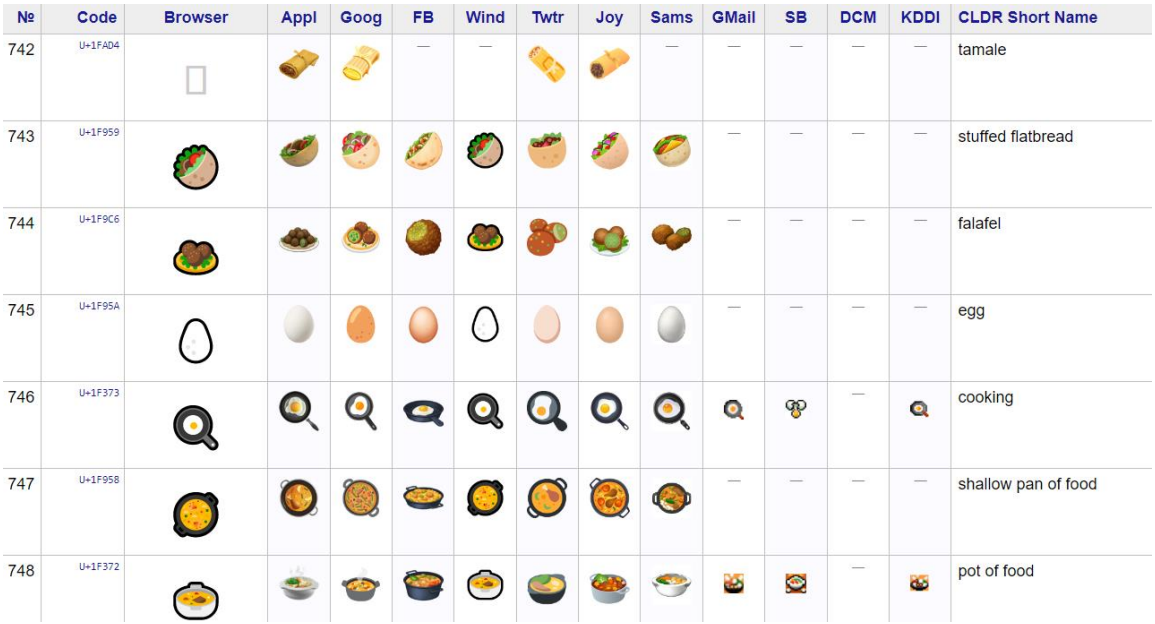


Рисунок 2.2 – Приклад згенерованих текстових речень

Наступним кроком є створення речень з смайлів, які б описували зміст текстового речення. Для цього було вирішено використовувати існуючі емодзі (рисунок 2.3), кожен з яких має унікальний юнікод для відображення у браузері чи в телефоні. Також до кожного смайла є короткий текстовий опис з одного слова чи словосполучення, було

вирішено, що у цьому дослідженні кожне цифрове зображення буде подано у вигляді текстового ідентифікатора, а опис смайла з декількох слів буде поданий як одне слово (рисунок 2.4). А трансформуванням текстового ідентифікатора у зображення та навпаки буде займатися інформаційна система, яка буде взаємодіяти з користувачем та використовувати нейронні мережі.



№	Code	Browser	Appl	Goog	FB	Wind	Twtr	Joy	Sams	GMail	SB	DCM	KDDI	CLDR Short Name
742	U+1FAD4				—	—			—	—	—	—	—	tamale
743	U+1F959									—	—	—	—	stuffed flatbread
744	U+1F9C6									—	—	—	—	falafel
745	U+1F95A									—	—	—	—	egg
746	U+1F373											—		cooking
747	U+1F958									—	—	—	—	shallow pan of food
748	U+1F372											—		pot of food

Рисунок 2.3 – Приклад наявних емодзі

Can I see the menu?	personstanding eyes openbook cook
Can you tell me what's wrong with me?	indexpointing speakinghead confoundedface personstanding questionred
How serious is my condition?	expressionlessface personstanding facewithmask dropofblood questionred
What kind of tests do I need to have done?	testtube personstanding personwalking questionred

Рисунок 2.4 – Приклад вхідного набору даних

Отже, було створено вхідні та вихідні речення для подальшого використання у цьому дослідженні. Було вирішено розбити створенні речення на тренувальні, які складають 850 речень, валідаційні – 150 та речення для тестування – 50.

2.2 Аналіз підходів розв’язку задачі машинного перекладу

Обидві задачі, для яких потрібно дослідити методи глибинного навчання, є задачами машинного перекладу, адже у контексті цих задач можна вважати, що емодзі – це існуюча звичайна мова, якою спілкуються люди. Тобто, необхідно перекласти речення з однієї мови на іншу та навпаки. Отже, математичним рішенням обох задач є знаходження найбільш ймовірного слова з іншої мови, яке відповідає слову з вхідної мови (формула 2.1) [5].

$$target = \text{argmax}_{target}\text{P}(\text{target}|\text{source}, \theta), \tag{2.1}$$

де target – вихідна мова;

source – вхідна мова;

θ – параметри моделі.

Розглянемо та проаналізуємо декілька методів для розв'язання задачі машинного перекладу. Одним з перших був метод статистичного машинного перекладу, основна ідея якого полягає у перекладі по словосполученнях для збільшення відповідності змісту вихідного речення до вхідного [6]. Метод статистичного машинного перекладу використовує n-грами [7], тобто може дивиться не тільки на отримане слово з найбільшою ймовірністю, а ще й розраховує ймовірність того, як часто це слово вживається у мові після попередніх слів, які вже були визначенні раніше. Отже, для розрахунку значення ймовірності послідовності необхідно перемножити ймовірності попередніх словосполучень (формула 2.2), після чого обрати слово, яке має найбільшу ймовірність у такій послідовності слів.

$$P_{i=1}^{n}(w^n) \approx \text{GP}(\text{w}_i|\text{w}_{i-1}) \approx \text{P}(\text{w}_1)\text{P}(\text{w}_2|\text{w}_1) \dots \text{P}(\text{w}_n|\text{w}_{n-1}). \tag{2.2}$$

Завдяки n-грамам така система обирає переклад слова, яке найбільше підходить по змісту, але такі системи займають багато пам’яті та часу для розрахування значень для довгих n-грам, а також переклад зазвичай робився через англійську мову, наприклад спочатку з української мови на англійську, а потім вже на іспанську, що призводило до збільшення затрат у часі у 2 рази.

Інший метод розв’язку полягає у використанні моделі «Seq2Seq», яка складається із двох частин: кодувальника та декодера [8]. Кодувальник збирає інформацію про речення вхідною мовою у вигляді вектора схованого стану та відправляє його у декодер. У свою ж чергу декодер використовує інформацію від кодувальника для того, щоб згенерувати вихідну послідовність користуючись результатом роботи від кодувальника. Зазвичай у моделях «Seq2Seq» використовують нейронні мережі, які побудовані на архітектурі рекурентних нейронних мереж. Найбільш відомі з них це звичайне рекурентна мережа (RNN), мережа з довгою короткочасною пам'яттю (LSTM) та мережа з замкнутими рекурентними одиницями (GRU), які відрізняються особливістю реалізації повторювального блока.

Рекурентні нейронні мережі допомагають у вирішені проблеми впливу контексту попереднього речення або його початку на кінцевий результат роботи мережі, вони можуть відстежувати залежності, які віддалені одна від одної (рисунок 2.6).

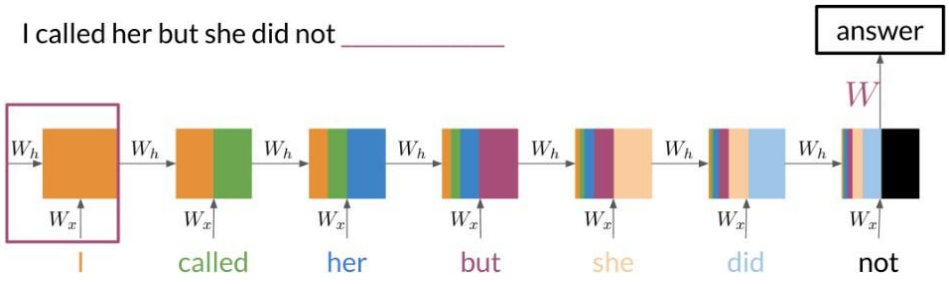


Рисунок 2.5 – Вплив слів на кінцевий результат зі збільшенням вхідної інформації в рекурентних нейронних мережах.

Коли ми вводимо більше інформації в модель, вплив попереднього слова на результат стає слабшим, але він все ще є, тобто RNN здатна фіксувати залежності та запам’ятовує попереднє слово, при цьому воно може знаходиться на початку речення чи абзацу [9].

Але у моделей рекурентної архітектури є декілька проблем, а саме:

– кодувальник може забувати початок довгих речень, що може призвести до не правильного перекладу із-за втрати контексту;

– жадібне декодування.

Для розв’язання проблеми жадібного декодування використовують підхід Beam Search [10]. Основна його ідея полягає у виборі декількох слів з найбільшими ймовірностями при перекладі замість одного. При перекладі слова ми обираємо не один результат з найбільшою ймовірністю, а беремо декілька можливих результатів, після чого для кожного такого слова ми генеруємо наступне, тобто отримуємо декілька гіпотез перекладу. Наступним кроком буде вибір декількох гіпотез з найбільшими значеннями ймовірності та продовжити переклад з обраними гіпотезами. Отже, використовуючи метод Beam Search при перекладі можна отримати послідовність, окремі слова якої мали не найбільше значення ймовірності.

Для розв’язку проблеми із втратою контексту до моделі Encoder-Decoder додають новий компонент моделі, який називається «Attention» або ж шар уваги (рисунок 2.6). Він допомагає декодеру визначитися зі словами, які потрібно брати до уваги при перекладі на кожному кроці, для кожного наступного слова.

Тобто при перекладі слова декодер звертається до шару уваги, передаючи йому значення свого прихованого стану, а шар уваги розраховує вектор контексту для цього прихованого стану використовуючи власну функцію та значення прихованих станів кодувальника. Після чого декодер оновлює свій стан на основі цього вектора контексту та генерує переклад слова, така взаємодія між

декодером та шаром уваги повторюється для генерації кожного наступного слова до завершення перекладу.

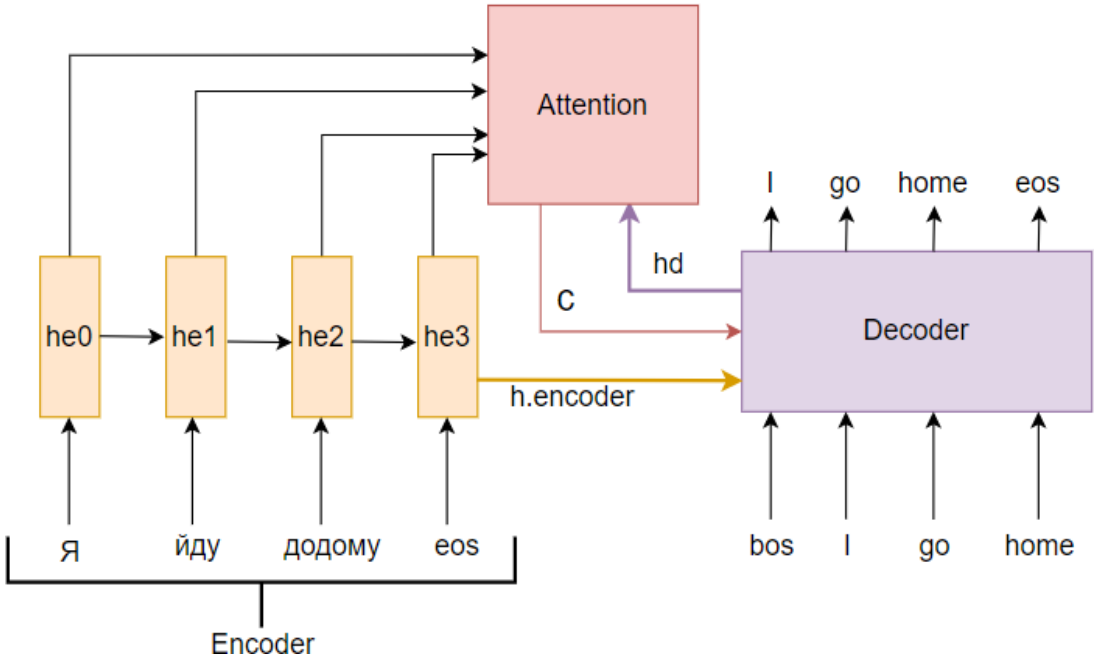


Рисунок 2.6 – Приклад взаємодії шару уваги із кодувальником тадекодером

Для розрахунку вектора контексту можна використовувати різні функції, нейронні мережі, але найбільш популярнимиє:

– множення кожного значень прихованого стану кодувальника та декодера;

– одношарова нейронна мережа.

Існує ще один спосіб розв’язування задач обробки природної мови, до яких відноситься й задача машинного перекладу. Цей метод полягає у використанні архітектури «Transformer», яка також має кодувальника та декодера, але побудована на використанні шарів уваги та не використовує архітектуру рекурентної нейронної мережі.

Ця архітектура має декілька переваг над моделями «Seq2Seq» з використанням рекурентних нейронних мереж:

– можливість паралельного кодування речення, що прискорює швидкість роботи моделі;

– збільшується точність перекладу, адже завдяки великої кількості шарів уваги модель має можливість звертати свою увагу на різні взаємозв’язки між словами у реченні та краще зберігає зміст великих речень;

– використовує менше пам’яті на довгих реченнях, адже моделі не потрібно запам'ятовувати усі слова речення;

– можливість паралельно розраховувати значення для внутрішнього шару уваги декодера.

Отже, існує багато моделей, які використовуються для розв’язку задачі машинного перекладу, у даній роботі необідно дослідити, чи підходять існуючі моделі для розв’язку поставленої задачі та проаналізувати, яка модель підходить найкраще для такого перекладу, тобто яка з них має більшу якість, відповідність перекладеного тексту до очікуваного.

3 НЕЙРОННИЙ МАШИННИЙ ПЕРЕКЛАДАЧ

3.1 Препроцесінг даних

Попередня обробка даних – це перший крок у створені моделі будь-якої моделі машинного навчання. Його основна мета полягає у тому, щоб підготови вхідні необроблені даних у формат, який очікує модель машинного навчання. У нашому випадку необхідно провести декілька кроків препроцесінгу вхідних даних.

Першим кроком для обох вхідних даних має бути токенізація речень, тобто перетворення речення на масив із чисел, де кожне число відповідає слову реченні. Адже для нейронної мережі вхідними даними мають бути числа тому, що вона виконує математичні операції. Для вхідних даних на англійській мові перед токенізацію необхідно видалити граматичні конструкції часу, бо неможливо показати цифровими зображеннями часові граматичні конструкції. Також непогано буде використати представлення півслів для токенізації наших речень, ця техніка дозволяє уникнути слів, які не входять у словниковий запас, дозволяючи частини слів представляти окремо, та дозволити токенайзеру групувати ці слова, коли потрібно.

Наступним кроком є фільтрація довгих речень, тобто становлення обмеження на кількість слів у реченні, щоб гарантувати, що у нас вистачить пам’яті для їх перекладу.

Після фільтрації довгих речень необхідно до кожного речення додати токен, який буде означати кінець речення. Це робиться для того, щоб модель розуміла, що вона закінчила переклад речення.

Останнім кроком є розбиття токенізованих речень за їх довжиною та приведення у однакову довжину, оскільки вхідні речення мають різну довжину. Основна ідея полягає у групувані маркерованих речень за довжиною та сегментом, це дозволяє не витрачати ресурси для приведення усіх речень до довжини найдовшого та прискорити навчання моделі.

3.2 Дослідження архітектур нейронних мереж для розв’язання задачі машинного перекладу

Запропоноване дослідження має на меті оцінити можливість використання існуючих підходів, які були представлені в розділі 3.2, для розв’язку задачі по перекладу з текстового речення у речення із емодзі та визначити, яка модель нейронної мережі видає переклад речення, який є найбільш точним до очікуваного. Було вирішено провести дослідження для двох рекурентних нейронних мереж з використанням шари уваги. Адже шар уваги збільшує вплив попередніх згенерованих слів при обчислені перекладу поточного слова у великих вхідних послідовностей у рекурентних нейронних мережах. Для дослідження було обрано нейрону мережу long short-term memory (LSTM) та gated recurrent units (GRU), які вирішують проблему зникаючого градієнту загальної рекурентної нейронної мережі (RNN) за допомогою затворів. Затвори – це невеликі нейронні мережі, кожна з яких має власні ваги та зміщення. Також було вирішено провести дослідження нейронної мережі з архітектурою трансформера для порівняння результату роботи нейронних мереж з різними архітектурами.

3.2.1 Дослідження рекурентних нейронних мереж для машинного перекладу

Почнемо з опису механізму уваги, тобто шару уваги, який використовується у обох моделей рекурентних нейронних мереж, які було обрано для дослідження. Цей механізм уваги дозволяє декодеру дивитися на усі зважені приховані стани послідовності кодувальника, що дозволяє більш точно створювати прогнози на відмінно від звичайних мереж архітектури кодувальник-декодер [11]. Для розрахунку значення уваги використовуються три вектори:

– вектор запиту (Q);

– вектор ключа (K);

– вектор значення (V).

Ці вектори створюються за допомогою множення ембеддингів матриці X, у якій кожний рядок матриці відповідає слову вхідного речення, на матриці ваг, які ми навчили. Було вирішено використовувати математичну модель уваги «Scaled Dot-Product Attention» (формула 3.1) для обчислення векторів, яка зображена на рисунку 3.1.

$$\text{Attention}(\text{Q}, \text{K}, \text{V}) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{3.1}$$

де Q, K та V це матриці запитів, ключів та значень відповідно;

d_k – розмірність ключів.

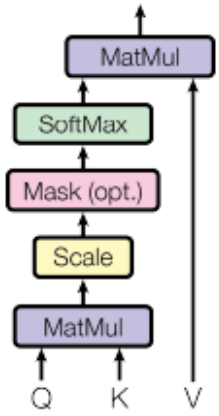


Рисунок 3.1 – Представлення реалізації уваги
«Scaled Dot-Product Attention»

Тепер перейдемо до дослідження моделі GRU з шаром уваги [12]. Звичайна мережа замкнутих нейронних одиниць на відмінно від стандартної RNN має два затвори, а саме затвори оновлення та збросу, або

ж забування. Затвор оновлення вирішує чи необхідно оновлювати стан комірки поточним значенням функції активації. Своєю чергою затвор зкидання вирішує чи важлива інформація попереднього стану комірки для майбутніх обчислень. На рисунку 3.2 зображено блок рекурентної нейронної мережі типу GRU.

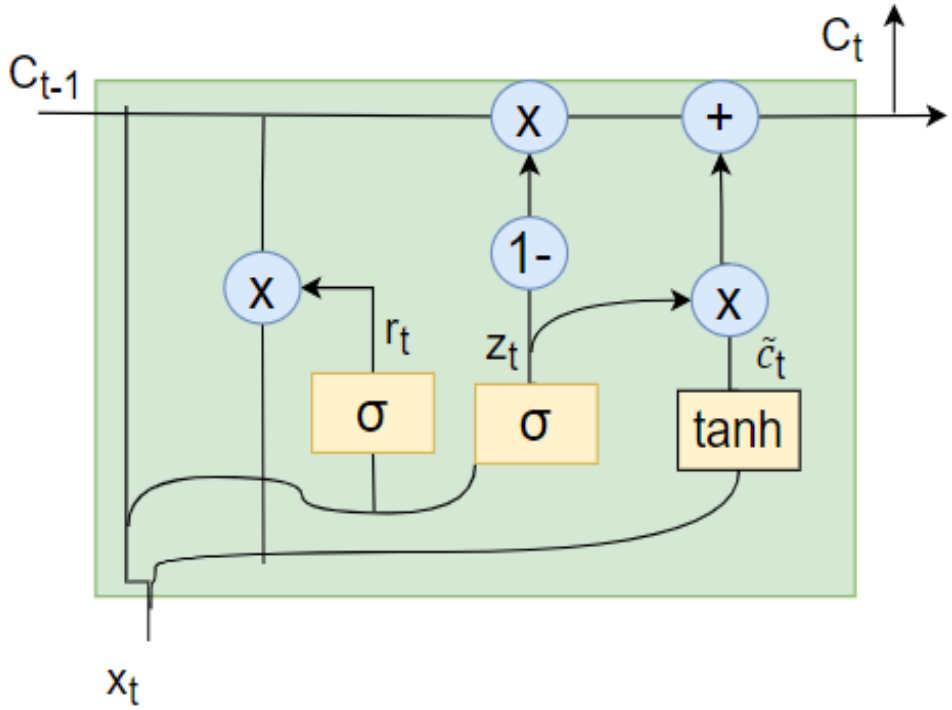


Рисунок 3.2 – Представлення блоку gated recurrent units

Ще однією особливість мережі GRU є те, що вона передає кінцевий стан комірки напряму як активація наступної комірки. Математичну модель цієї мережі можна записати за допомогою формул для обчислень значень затворів та комірки пам’яті (формули 3.2–3.5).

$$c_t = (1 - z_t) \circ z_{t-1} + z_t \circ \tilde{c}_t, \tag{3.2}$$

$$\tilde{c}_t = \tanh(W_{xc} x_t + W_{hc}(r_t \circ h_{t-1})), \tag{3.3}$$

$$r_t = \sigma(\mathrm{W_{xr}x}_t + \mathrm{W_{hr}c}_{t-1} + \mathrm{b}_r), \quad (3.4)$$

$$z_t = \sigma(\mathrm{W_{xz}x}_t + \mathrm{W_{hz}c}_{t-1} + \mathrm{b}_z), \quad (3.5)$$

де c_t – вихідний вектор;

$\tilde{\mathrm{c}}_t$ – вектор активаціїї кандидата;

z_t та r_t – це вихідні вектора завторів оновлення та збросу відповідно;

b_{r}, b_c, b_{z} – це ваги зміщення;

$\mathrm{W_{xr}}$, $\mathrm{W_{x}}c$, $\mathrm{W_{xz}}$ – це вхідні ваги;

$\mathrm{W_{hr}}$, $\mathrm{W_{h}}c$, $\mathrm{W_{hz}}$ – це рекурентні ваги.

У лістингу 3.1 приведено код для побудови рекурентної нейронної мережі, яка використовую блоки GRU у кодувальнику та декодері.

Лістинг 3.1 – Код створення моделі нейронної мережі з блоками GRU та шаром уваги

```
input_encoder = tl.Serial(
        tl.Embedding(vocab_size=input_vocab_size,
d_feature=d_model),
        [tl.GRU(n_units=d_model) for _ in range(n_encoder_layers)]
)

pre_attention_decoder = tl.Serial(
        tl.ShiftRight(mode=mode),
        tl.Embedding(vocab_size=target_vocab_size,
d_feature=d_model),
        tl.GRU(n_units=d_model)
)
model = tl.Serial(
      tl.Select([0, 1, 0, 1]),
      tl.Parallel(input_encoder, pre_attention_decoder),
      tl.Fn('PrepareAttentionInput', prepare_attention_input,
n_out=4),
```

Продовження лістингу 3.1

```
        tl.Residual(tl.AttentionQKV(d_model,
n_heads=n_attention_heads, dropout=attention_dropout, mode=mode)),
        tl.Select([0, 2]),
        [tl.GRU(n_units=d_model) for _ in range(n_decoder_layers)],
        tl.Dense(target_vocab_size),
        tl.LogSoftmax()
)
```

Нейронна мережа LSTM, блок якої представлено на рисунку 3.3, на відмінно від GRU має більше затворів, а саме три. Вона складається з вхідного затвору, вихідного та затвору забуття [12]. Завдяки цим затворам LSTM здатна взаємодіяти з станом пам'яті комірки, а саме видаляти непотрібну інформацію з неї, яку саму інформацію потрібно видалити вирішує затвор забуття. Він повертає числа від 0 до 1 для кожного стану комірки, де 0 – це повністю видалити, а 1 – навпаки. Вихідний же затвор відповідає за інформацію, яка передається наступному блоку нейронній мережі на вхід.

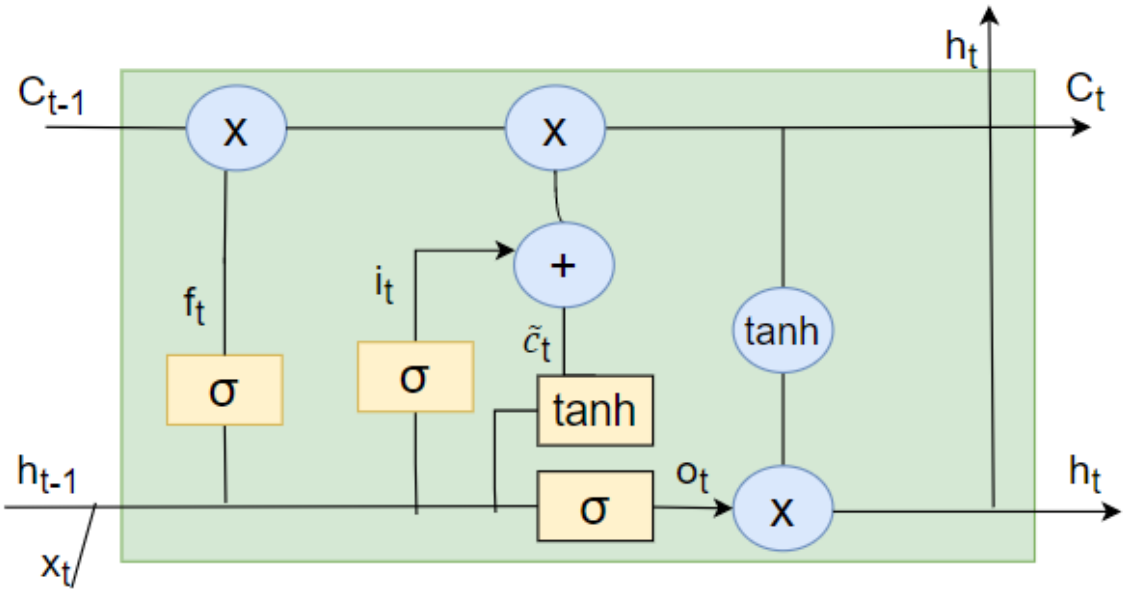


Рисунок 3.3 – Представлення блоку long short-term memory

Кожний затвор для розрахунку значень використовує сигмоїдну функцію, а значення нової інформації розраховується за допомогою гіперболічної функції активації. Математична модель LSTM складається з обчислення векторів для знаходження значень затворів та стану комірки (формули 3.6-3.11).

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + w_{co} \circ c_{t-1} + b_o), \quad (3.6)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + w_{cf} \circ c_{t-1} + b_f\right), \quad (3.7)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + w_{ci} \circ c_{t-1} + b_i), \quad (3.8)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t, \quad (3.9)$$

$$\tilde{c}_t = tahn(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \quad (3.10)$$

$$h_t = o_t \circ tahn(c_t), \quad (3.11)$$

де o_t, f_t, i_t – вектори активації вихідного завтора, завтора забуття та вхідного відповідно;

c_t та $\tilde{c}_t$ – вектори стану комірки та активаціїї стану комірки відповідно;

h_t – вектор прихованого стану або ж вихідний вектор блоку LSTM;

z_t та r_t – це вихідні вектора завторів оновлення та збросу;

b_o, b_f, b_i, b_c – це ваги зміщення;

W_{xo}, W_{xc}, W_{xf}, W_{xi} – це вхідні ваги;

W_{ho}, W_{hc}, W_{hf}, W_{hi} – це рекурентні ваги.

У лістингу 3.2 наведений код для створення нейронимим мережами, як можно побасити, що він майже ідентичний до лістингу 3.1, бо відрізняється тільки у використанні блоку LSTM замість GRU у кодувальнику та декодері

Лістинг 3.2 – Створення моделі нейронної мережі з блоками LSTM та з шаром уваги

```
input_encoder = tl.Serial(
        tl.Embedding(vocab_size=input_vocab_size,
d_feature=d_model),
        [tl.LSTM(n_units=d_model) for _ in range(n_encoder_layers)]
)
pre_attention_decoder = tl.Serial(
        tl.ShiftRight(mode=mode),
        tl.Embedding(vocab_size=target_vocab_size,
d_feature=d_model),
        tl. LSTM (n_units=d_model)
)
model = tl.Serial(
      tl.Select([0, 1, 0, 1]),
      tl.Parallel(input_encoder, pre_attention_decoder),
      tl.Fn('PrepareAttentionInput', prepare_attention_input,
n_out=4),
      tl.Residual(tl.AttentionQKV(d_model,
n_heads=n_attention_heads, dropout=attention_dropout, mode=mode)),
      tl.Select([0, 2]),
      [tl. LSTM (n_units=d_model) for _ in range(n_decoder_layers)],
      tl.Dense(target_vocab_size),
      tl.LogSoftmax()
)
```

Тобто архітектура досліджувальних нейронних мереж відрізняється тільки нейронними мережами, які використовуються у кодувальнику та декодері, та має вигляд зображений на рисунку 3.4. Як можна побачити ми можемо розпаралелити процес роботи кодувальника та декодера попередньої уваги, який створює активації для подальшого використання як запити на увагу.

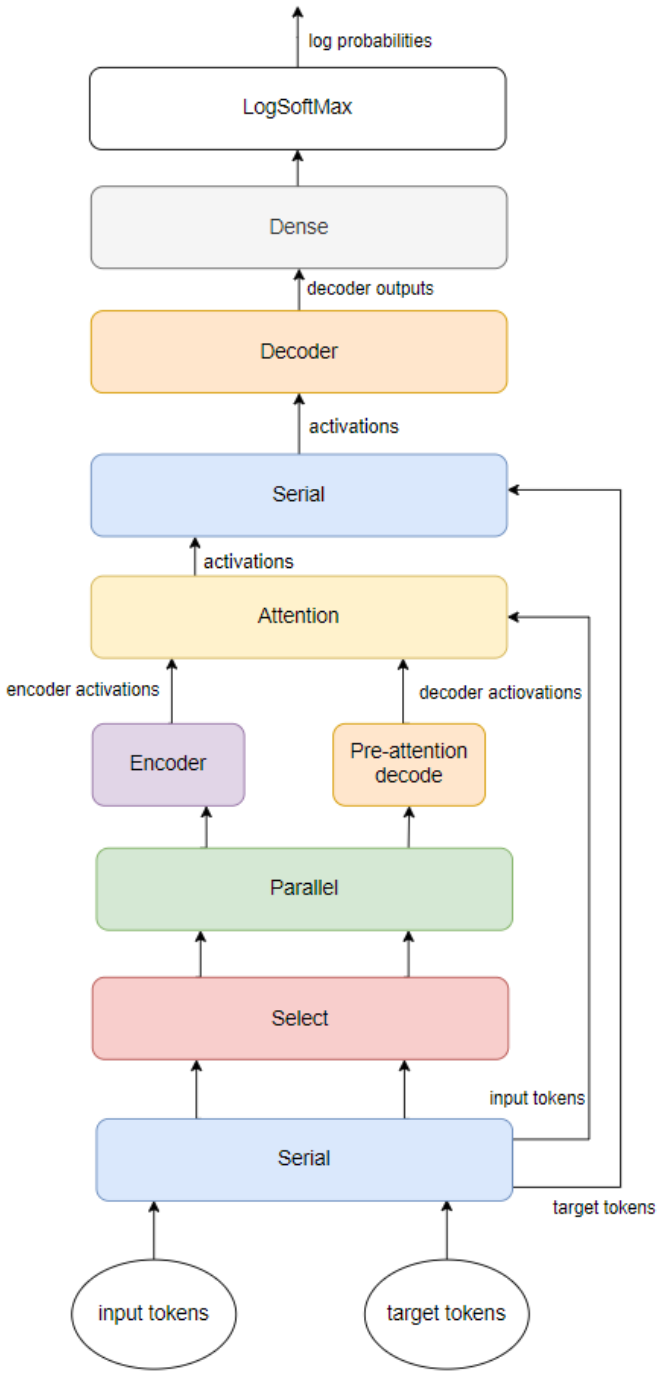


Рисунок 3.4 – Представлення архітектури створеної рекурентної нейронної мережі з шаром уваги

Отже, було досліджено два різні види рекурентних нейронних мереж, а саме GRU та LSTM, описано їх відмінності між собою та від звичайної рекурентної мережі. Наприкінці було побудовано дві моделі, які використовують механізм уваги.

3.2.2 Дослідження архітектури трансформера для машинного перекладу

Як було вже сказано трансформатори засновані на шарах уважності (рисунок 3.5) та не вимагають послідовних обчислень для кожного шару як у рекурентних нейронних мережах [13], [14]. Також у трансформерах крок градієнта залишається незмінним, який обчислюється від останнього виходу до першого входу, а у RNN цей крок збільшується із збільшенням довжини вхідних речень. Так як трансформер відноситься до моделей «Seq2Seq», то він також має кодувальника та декодера.

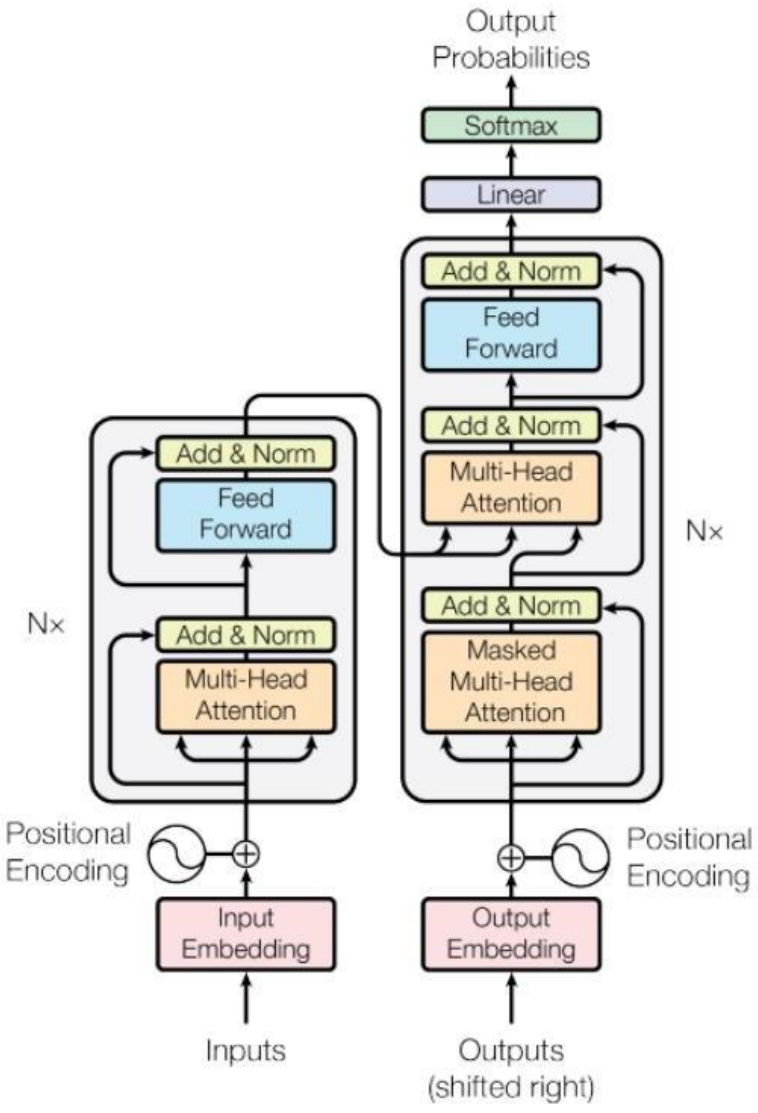


Рисунок 3.5 – Загальна архітектура нейронної мережі «Transformer»

Кожний шар кодувальника складається з двох підшарів, а саме механізму самоконтролю з кількома головками та мережею прямого зв'язку, яка позиційно повністю пов'язана. Перший підшар допомагає кодувальнику дивитися на інші слова речення під час кодування та відправляє результат у другий підшар. Структура декодера відрізняється від кодувальника тим, що має ще один підшар, який допомагає декодеру звернути увагу на вихідну інформацію від кодувальника. У кожному підшарі використовується однакова мережа прямого зв'язку, яка складається з двох лінійних перетворень та з активацією ReLU функцією між ними.

Інша особливість архітектури трансформера полягає у використані механізму multi-head attention (рисунок 3.6), тобто використовується не одна головка уваги, а декілька. При цьому обчислення значень кожної головки відбувається паралельно тому, що при побудові матриць запитів, ключів та значень використовуються різні ваги для кожної, після чого відбувається конкатенація отриманих значень від усіх головок уваги та множення на додаткову вагу кінцевої матриці.

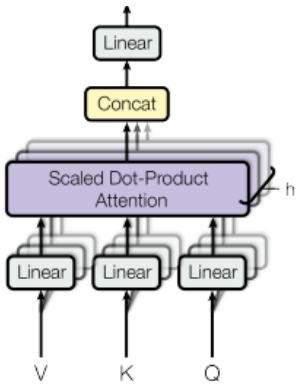


Рисунок 3.6 – Графічне зображення шару «Multi-head attention»

Отже, цей механізм дозволяє моделі одночасово звертати увагу на інформацію з різних підпросторів представлення вхідного речення при перекладі слів, тобто він надає інформацію про зв’язок між різними

словами речення. Крім того, мережа архітектури трансформера не страждає від проблеми з втратою контексту при перекладі довгих речень, бо обробляє речення в цілому, а не покладаються на приховані стани минулих степів для зберігання змісту речення.

3.3 Аналіз отриманих результатів

Для порівняння отриманих моделей було вирішено використовувати метрику Bilingual Evaluation Understudy (BLEU). Ця метрика використовується для автоматичного оцінювання тексту, який було перекладено машинним перекладачем. BLEU можна приймати значення від 0 до 1, де 0 означає низьку якість перекладу, тобто те, що отриманий переклад вхідної послідовності не збігається з очікуваним. А значення 1 означає, що результат машинного перекладу повністю збігається з очікуваним реченням. Математична модель метрики BLEU складається з двох частин, а саме: штрафу за стислість та перекриттям n-грам.

$$\mathrm{p}recision_i = \frac{\sum_{snt \in cand} \sum_{k \in snt} \min(m^k_{cand}, m^k_{ref})}{w^i_t}, \tag{3.12}$$

$$\mathrm{B}LEU = \min(1, \exp\left(1 - \frac{refL}{candL}\right))(\prod_{i=1}^{n} precision_i)^{1/n}, \tag{3.13}$$

де $cand$ – кількість і-грам в кандидаті, тобто в реченні, яке було отримане шляхом машинного перекладу;

m^i_{cand} – це кількість і-грам в кандидаті, що відповідає очікуваному перекладу;

m^i_{ref} – кількість і-грам в очікуваному перекладу;

w^i_t відповідає загальній кількості і-грам в кандидаті при перекладі;

$refL$ – довжина очікуваного речення;

$candL$ – довжина отриманого речення;

n – максимальна довжина n-грам.

На рисунку 3.7 представлена інтерпретації оцінки BLEU, яка було переведена у відсотки, тобто помножене на 100.

BLEU Score	Interpretation
< 10	Almost useless
10 - 19	Hard to get the gist
20 - 29	The gist is clear, but has significant grammatical errors
30 - 40	Understandable to good translations
40 - 50	High quality translations
50 - 60	Very high quality, adequate, and fluent translations
> 60	Quality often better than human

Рисунок 3.7 – Інтерптритація значень оцінки BLEU

Тобто метрика BLUE рахує кількість словосполучень різної довжини, так звані n-грами, які є в очікуваних реченнях та в результаті машинного перекладу. Більш довші n-грами використовують для визначення правильної послідовності слів у перекладі, можна сказати перевірку змісту отриманого речення.

При цьому ця метрика має декілька недоліків, про які слід пам'ятати порівнюючи моделі:

– погано працює при оцінюванні речень, які відображають зміст речення, але складаються з інших слів;

– не оцінює граматичний зміст речення, тобто якщо в результаті перекладу немає заперечення чи іншої частини мови, яка сильно змінює зміст речення, метрика лише зменшить кількість балів за те, що таких n-грам у перекладі не існує;

– відсутність службових слів, наприклад артиклів у англійській мові, однаково негативно впливає на оцінку як і відсутність або помилкове слово, яке впливає на зміст;

– токенізація речень кардинально впливає на оцінку, тобто якщо очікуване речення має іншу токенізацію від результату перекладу, то оцінка буде низькою, хоча при цьому текстовий зміст речень може бути однаковим.

Було визначено оцінку BLEU для трьох досліджувальних моделей для текстових послідовностей, довжина яких склала 10, 20 та 30 слів за допомогою 20 речень для кожної послідовності. Результати оцінки якості перекладу наведенні у таблицях 3.1 та 3.2.

Таблиця 3.1 – Результат оцінки перекладу моделями метрикою BLEU з англійських речень у речення з емодзі

	Оцінка BLEU		
	10 слів	20 слів	30 слів
GRU з механізмом уваги	23.85	21.9	19.27
LSTM з механізмом уваги	23.43	21.78	19.4
Transformer	26.62	25.78	24.3

Таблиця 3.2 – Результат оцінки перекладу моделями метрикою BLEU з речень у вигляді ідентифікаторів смайлів на англійську мову.

	Оцінка BLEU		
	10 слів	20 слів	30 слів
GRU з механізмом уваги	22.34	20.48	18.13
LSTM з механізмом уваги	22.06	20.5	18.21
Transformer	24.3	23.17	21.86

Слід сказати, що значення оцінки BLEU може збільшитися та досягти 30 відсотків та більше для моделей при збільшенні вхідних даних, адже у моделі буде більше інформації про різні взаємозв’язки слів та смайлів, що покращить якість перекладу. Також те, що для оцінки речень довжиною 10 слів було вирішено використати триграми, а в інших випадках 4-грами для обчлислення оцінки. Бо, якщо четверте та восьме слово відрізняються від таких слів в очікуваному реченні, то це дуже знизить оцінку якості перекладу, адже усі 4-грами будуть не вірні при довжині речення у 10 слів.

Як ми можемо побачити з двох таблиць, моделі більш якісно перекладають з англійської мови у послідовність ідентифікаторів смайлів ніж навпаки. У випадках коли довжина вхідної послідовності менша або дорівнює 20 словам усі моделі досліджувальних нейронних мереж мають оцінку BLEU вище ніж 20 відсотків, що можна інтерпретувати як переклад із зрозумілим змістом, проте з граматичними помилками. При довжині послідовності від 20 до 30 слів оцінка обох рекурентних нейрони мереж впала нижче 20 відсотків, а оцінка трансформера залишилася вище цього значення в обох випадках.

Моделі Рекурентні нейроні мережі з архітектурою LSTM та GRU з механізмом уваги показали майже однакові значення метрики BLEU, при цьому GRU має більше значення при довжині речень до 20 слів, а вже від 20 до 30 слів нейрона мережа LSTM перекладає трохи краще. Але при цьому GRU швидше та простіше навчається та швидше працює, бо має менше затворів на відмінно від LSTM. Тому, якщо порівнювати між ними, то кращим варіантом буде обрання нейронної мережі GRU тому, що різниця між оцінками якості перекладу дуже невелика, менше за 1 відсоток.

Отже, виходячи з результатів усі 3 моделі можна використовувати для розв’язку поставленої задачі по перекладу з англійського тексту у послідовність з емодзі із збереженням змісту, шляхом його візуалізації, та

навпаки зі смайлів у текстові речення англійською мовою. Проте кращим вибором буде обрання моделі з архітектурою трансформер, хоча його якість перекладу не дуже вища за метрикою BLEU від якості інших досліджувальних моделей, вона не настільки знижується при збільшенні довжини вхідного речення для перекладу.

ВИСНОВКИ

В результаті виконання науково-дослідної роботи було проведено пошук, аналіз наявних методів, які використовуються для розв’язання задачі машинного перекладу, розробка нейронних мереж, які виконують переклад з англійської мови у речення, яке складається лише з смайлів, та навпаки із збереженням контексту вхідного речення. Було проведено аналіз предметної галузі стосовно типів афазії, симптомів різних типів цього розладу, які можуть проявлятися у людей, хворих афазією.

Також було запропоновано ідею розробити нейронні мережі для вирішення проблеми соціалізації та спілкуванні людей з афазією, яка якраз полягає у створенні мереж для відображення змісту речень, який людина, хвора афазією, розуміє краще ніж звичайну мову. Було проведено аналіз реалізованих систем, які надають можливість спілкуватися за допомогою смайлів, але обидва додатка мають свої недоліки, а чат-бот «ChatGPT» на даний момент не може перетворювати текстові речення у смайли із збереженням контексту.

Проведено поверхневий аналіз існуючих методів розв’язку задачі машинного перекладу з яких було вирішено реалізувати нейронний машинний перекладач з використанням LSTM, GRU з механізмом уваги та з архітектурою трансформера. Окрім того, було створено власний набір вхідних даних, написано переклад текстових речень у вигляді ідентифікаторів смайлів, ці дані були використанні під час навчання та тестування якості отриманих моделей.

Для порівняння якості перекладу розроблених моделей було вирішено використовувати метрику BLEU, яка оцінює наскільки близьким вийшов переклад, перевіряючи співвідношення n-грам різних довжин між очікуваним реченням та результатом роботи нейронного машинного перекладача. Було проведено оцінку кожної моделі на однакових очікуваних реченнях довжиною десять, двадцять та тридцять слів.

Проаналізувавши отриманні результати, можна стверджувати, що при перекладі якість перекладу рекурентними нейронними мережами GRU та LSTM з шаром уваги відрізняється між собою менше на 0,1. А кращий результат продемонструвала модель, яка побудована з використанням архітектури трансформера, тому, що якість перекладу цієї моделі була більшою в усіх трьох випадків тестування для оцінки перекладу речень різних довжин.

Для подальшого удосконалення нейронних мереж варто створити власний набір смайлів, які б допомагали більш точно та якісно передати зміст речення. Окрім цього звичайно необхідно збільшити набір даних для навчання та тестування нейронної мережі хоча б для початку до 50 тисяч речень, що дозолить використовувати натреновану модель у різних життєвих випадках.

Отже, в результаті проведеного дослідження було розроблено декілька моделей нейронних мереж, які виконують машинний переклад, що дозволяє у майбутньому створити інформаційну систему, яка б взаємодіяла з користувачем та використовувала створені моделі для перекладу. Крім цього це дослідження дає змогу у майбутньому додати нові нейронні мережі, які будуть розпізнавати текст по камері чи по голосу та перетворювати у речення зі смайлів, використовуючи створені моделі. Ці нові функції зможуть вирішити ще більшу кількість повсякденних проблем людей, хворих афазією.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is aphasia? – types, causes and treatment. URL: https://www.nidcd.nih.gov/health/aphasia (date of access: 06.04.2023);

2. Aphasia: types, causes, symptoms & treatment. URL: https://my.clevelandclinic.org/health/diseases/5502-aphasia (date of access: 06.04.2023);

3. Ananya Ananth Rao, Prof Venkatesh S. Identification of Aphasia using natural language processing. Journal of University of Shanghai for Science and Technology. 2021. Vol. 23. P. 1737–1747.;

4. Jothi K. R., Sivaraju S. S., Yawalkar P. J. AI based Speech Language Therapy using Speech Quality Parameters for Aphasia Person. 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 5–7 November 2020. 2020. P. 1263–1271;

5. Jurafsky D., H. Martin J. Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition draft. 2023. P. 636. URL: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf (date of access: 08.04.2023);

6. Natural language processing (NLP) and its use in machine translation. URL: https://www.qblocks.cloud/blog/natural-language-processing-machine-translation (date of access: 11.04.2023);

7. Kapadia S. Language Models: N-Gram. URL: https://towardsdatascience.com/introduction-to-language-models-n-gram-e323081503d9 (date of access: 11.04.2023);

8. Sutskever I., Oriol V., Quoc V L. Sequence to Sequence Learning with Neural Networks. 2014. P. 9. URL: https://doi.org/10.48550/arXiv.1409.3215 (date of access: 12.04.2023);

9. Yaşar A. Neural Machine Translation: Inner Workings, Seq2Seq, and Transformers. URL: https://towardsdatascience.com/neural-machine-translation-

inner-workings-seq2seq-and-transformers-229faff5895b (date of access: 12.04.2023);

10. What is Beam Search? Explaining The Beam Search Algorithm | Width.ai. URL: https://www.width.ai/post/what-is-beam-search (date of access: 12.04.2023);

11. Overview of the Transformer-based Models for NLP Tasks / A. Gillioz et al. 2020 Federated Conference on Computer Science and Information Systems, 6–9 September 2020. 2020. URL: https://doi.org/10.15439/2020f20 (date of access: 13.04.2023);

12. Pedamallu H. RNN vs GRU vs LSTM. URL: https://medium.com/analytics-vidhya/rnn-vs-gru-vs-lstm-863b0b7b1573 (date of access: 15.04.2023);

13. Understanding LSTM Networks. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs (date of access: 15.04.2023);

14. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin. Attention is All you Need. 2017. P. 15. URL: https://doi.org/10.48550/arXiv.1706.03762 (date of access: 16.04.2023);

15. The Illustrated Transformer. URL: https://jalammar.github.io/illustrated-transformer/ (date of access: 16.04.2023);

16. Doshi K. Foundations of NLP Explained–Bleu Score and WER Metrics. URL: https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b (date of access: 17.04.2023);

17. LSTM vs. GRU for Arabic Machine Translation / N. Bensalah et al. Advances in Intelligent Systems and Computing. Cham, 2021. P. 156–165. URL: https://doi.org/10.1007/978-3-030-73689-7_16 (date of access: 17.04.2023);

18. Wolf T., Tunstall L., Werra L. v. Natural Language Processing with Transformers, Revised Edition. O'Reilly Media, Incorporated, 2022. 383 p.;

20. Kulkarni A., Shivananda A., Kulkarni A. Natural Language Processing Projects: Build Next-Generation NLP Applications Using AI Techniques. Apress L. P., 2021. 336 p.;

21. Trax Tutorials – Trax documentation. URL: https://trax-ml.readthedocs.io/en/latest/ (date of access: 28.04.2023).

ДОДАТОК А

Графічні матеріали

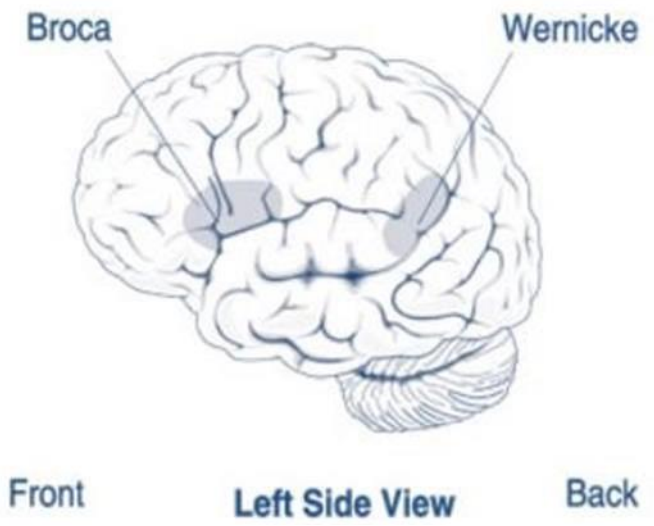


Рисунок А.1 – Ділянки мозку, уражені афазією Брока та Верніке

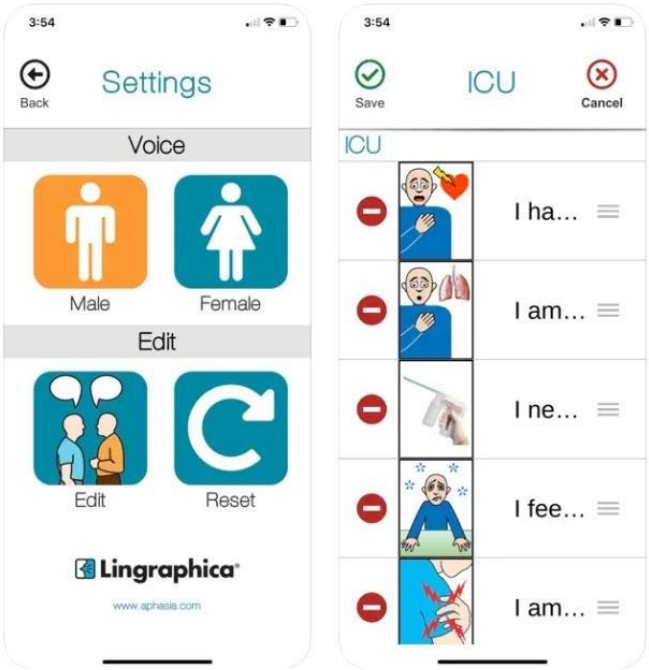


Рисунок А.2 – Приклади екранів мобільного додатку «SmallTalk Intensive Care» компанії Lingraphica

Рисунок А.3 – Приклади екранів мобільного додатку «Proloquo2Go AAC»

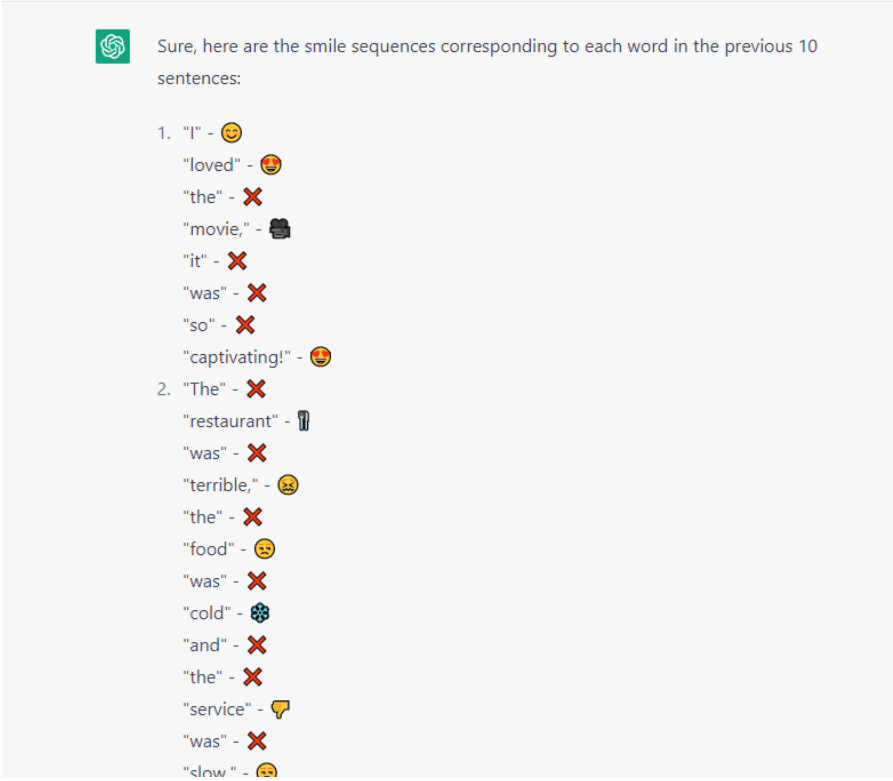


Рисунок А.4 – Приклад трансформування текстового речення у послідовність смайлів штучним інтелектом «ChatGPT»

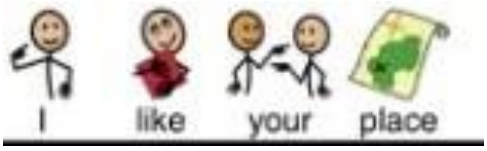


Рисунок А.5 – Приклад візуалізації текстового речення у додатку «Proloquo2Go AAC»

create 200 sentences that people use while speaking with doctor

1. What seems to be the problem, Doctor?
2. Can you tell me what's wrong with me?
3. How serious is my condition?
4. What kind of tests do I need to have done?
5. Is my condition curable?
6. What are my treatment options?
7. Can you explain the side effects of this medication?
8. Can you prescribe something for my pain?
9. How long will it take to recover?
10. Can I continue to work while I am being treated?
11. Can I still exercise while I am being treated?

Рисунок А.6 – Приклад згенерованих текстових речень

№	Code	Browser	Appl	Goog	FB	Wind	Twtr	Joy	Sams	GMail	SB	DCM	KDDI	CLDR Short Name
742	U+1FAD4				—	—			—	—	—	—	—	tamale
743	U+1F959									—	—	—	—	stuffed flatbread
744	U+1F9C6									—	—	—	—	falafel
745	U+1F95A									—	—	—	—	egg
746	U+1F373											—		cooking
747	U+1F958									—	—	—	—	shallow pan of food
748	U+1F372											—		pot of food

Рисунок А.7 – Приклад наявних емодзі

Can I see the menu?	personstanding eyes openbook cook
Can you tell me what's wrong with me?	indexpointing speakinghead confoundedface personstanding questionred
How serious is my condition?	expressionlessface personstanding facewithmask dropofblood questionred
What kind of tests do I need to have done?	testtube personstanding personwalking questionred

Рисунок А.8 – Приклад вхідного набору даних

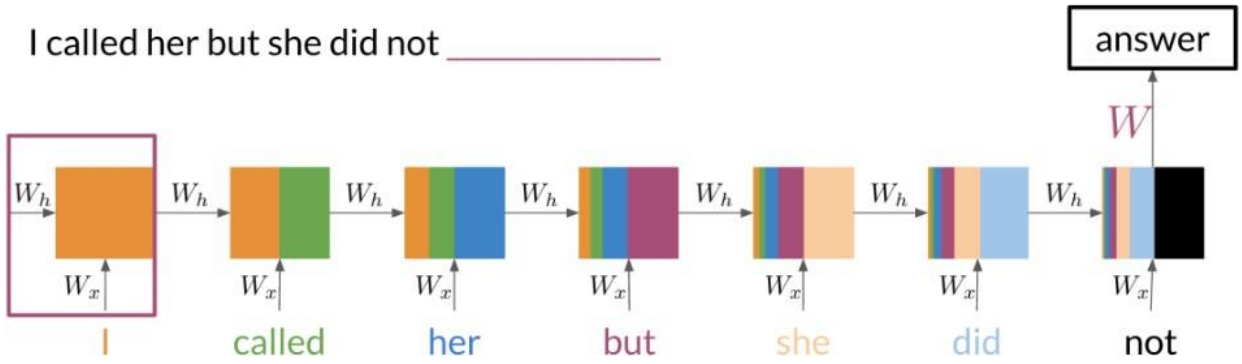


Рисунок А.9 – Вплив слів на кінцевий результат зі збільшенням вхідної інформації

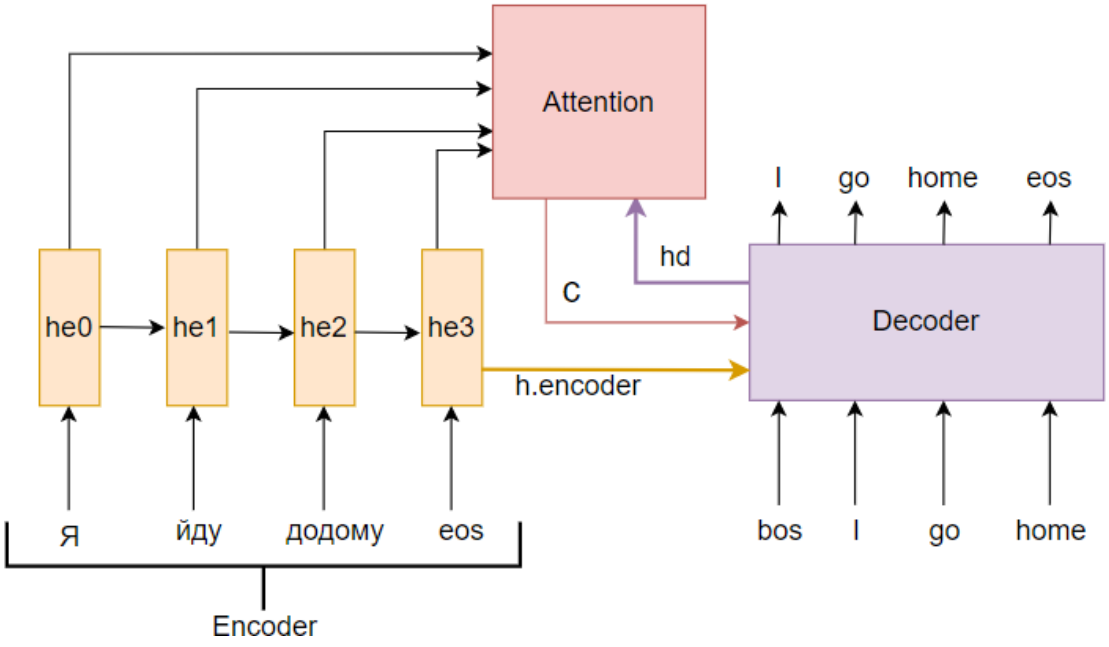


Рисунок А.10 – Приклад взаємодії шару уваги із кодувальником та декодером

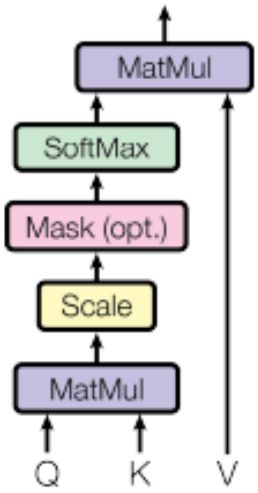


Рисунок А.11 – Представлення реалізації уваги «Scaled Dot-Product Attention»

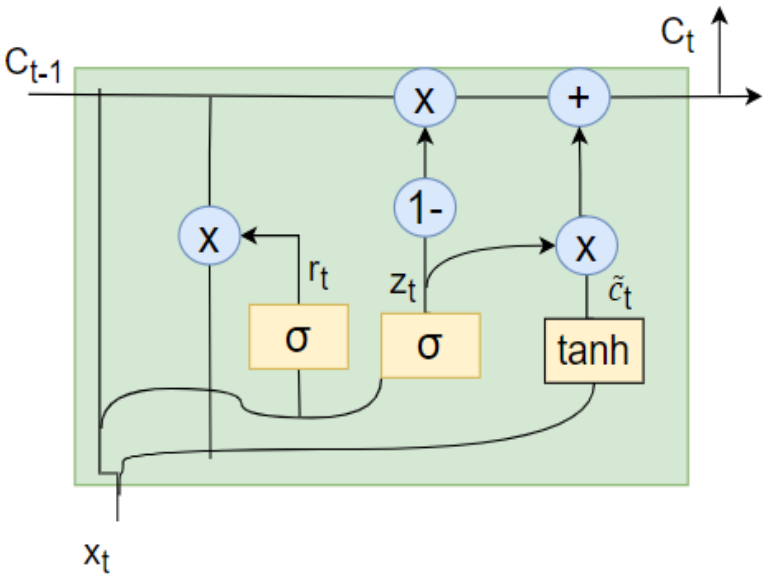


Рисунок А.12 – Представлення блоку gated recurrent units

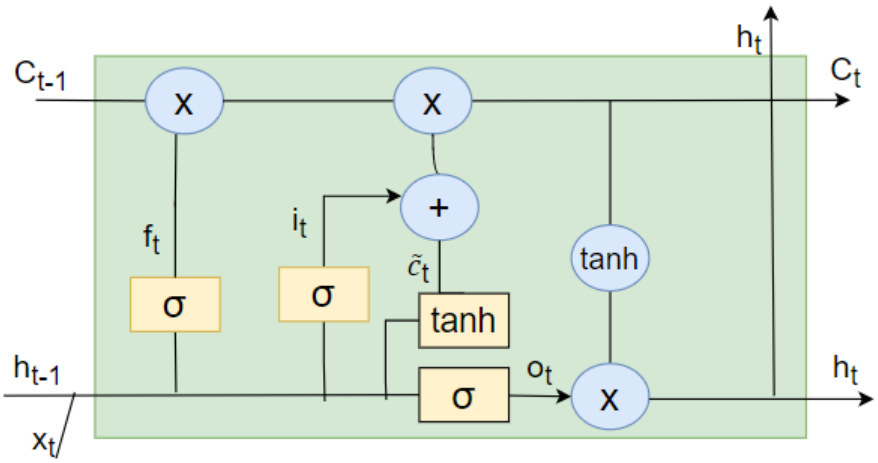


Рисунок А.13 – Представлення блоку long short-term memory

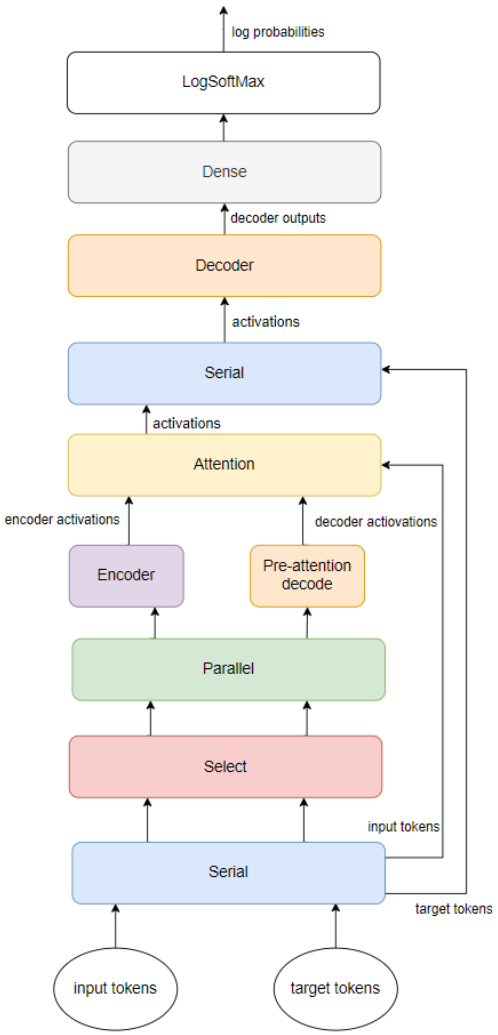


Рисунок А.14 – Представлення архітектури створеної рекурентної нейронної мережі з шаром уваги

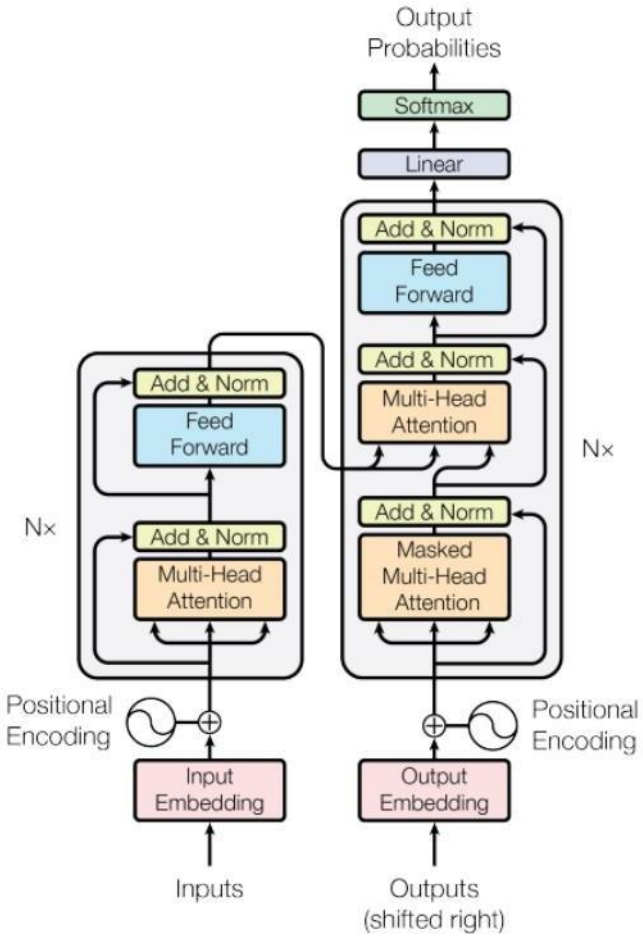


Рисунок А.15 – Загальна архітектура нейронної мережі «Transformer»

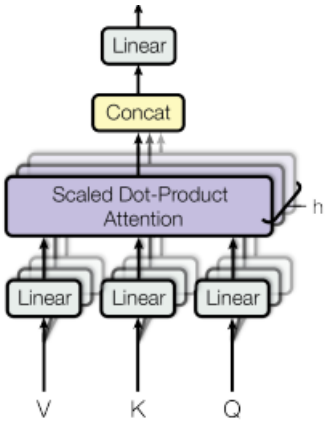


Рисунок А.16 – Графічне зображення шару «Multi-head attention»

BLEU Score	Interpretation
< 10	Almost useless
10 - 19	Hard to get the gist
20 - 29	The gist is clear, but has significant grammatical errors
30 - 40	Understandable to good translations
40 - 50	High quality translations
50 - 60	Very high quality, adequate, and fluent translations
> 60	Quality often better than human

Рисунок А.17 – Інтерптритація значень оцінки BLEU

ДОДАТОК Б

Текст програми

Код програмної реалізації моделей LSTM, GRU з механізмом уваги та трансформера для задачі машинного перекладу.

```
import numpy as np

import trax
from trax import layers as tl
from trax.fastmath import numpy as fastnp
from trax.supervised import training

EOS = 1
trainStream = loadDataForNN(path, isTrain=True)
evalStream = loadDataForNN(path, isTrain=False)
tokenizedTrainStream  =  trax.data.Tokenize(vocab_file="en.subword",
keys=[0])(trainStream)
tokenizedEvalTrainStream                                          =
trax.data.Tokenize(vocab_file="en.subword", keys=[0])(evalStream)

def appendEos(stream):
    for (inputs, targets) in stream:
        inputs_with_eos = list(inputs) + [EOS]
        targets_with_eos = list(targets) + [EOS]
        yield np.array(inputs_with_eos), np.array(targets_with_eos)

tokenizedTrainStream = appendEos(tokenizedTrainStream)
tokenizedEvalTrainStream = appendEos(tokenizedEvalTrainStream)
def filterByLength(stream):
    return  trax.data.FilterByLength(max_length=256,  length_keys=[0,
1])(stream)

tokenizedTrainStream = filterByLength(tokenizedTrainStream)
```

```
tokenizedEvalTrainStream = filterByLength(tokenizedEvalTrainStream)

def tokenize(input_str):
    inputs     =          next(trax.data.tokenize(iter([input_str]),
vocab_file="en.subword"))
    inputs = list(inputs) + [EOS]
    batch_inputs = np.reshape(np.array(inputs), [1, -1])

    return batch_inputs

def detokenize(tokens):
    tokens = list(np.squeeze(tokens))
    if EOS in tokens:
      tokens = tokens[:tokens.index(EOS)]
    return trax.data.detokenize(tokens, vocab_file="en.subword")

def bucketStream(stream):
    return trax.data.BucketByLength([8, 16, 32, 64, 128, 256], [128,
64, 32, 16, 8, 4, 2],
      length_keys=[0, 1])(stream)

tokenizedTrainStream = bucketStream(tokenizedTrainStream)
tokenizedEvalTrainStream = bucketStream(tokenizedEvalTrainStream)

def     prepareAttention(encoderActivations,     decoderActivations,
tokens):
    mask = tokens != 0
    mask    =    fastnp.reshape(mask,    (mask.shape[0],    1,    1,
mask.shape[1]))
    mask = mask + fastnp.zeros((1, 1, decoderActivations.shape[1],
1))
    return          decoderActivations,          encoderActivations,
encoderActivations, mask
```

```
def LSTMModel(input_vocab_size=500,
              target_vocab_size=500,
              d_model=512,
              n_encoder_layers=2,
              n_decoder_layers=2,
              n_attention_heads=3,
              mode='train'):
    inputEncoder = tl.Serial(
        tl.Embedding(vocab_size=input_vocab_size,
d_feature=d_model),
        [tl.LSTM(n_units=d_model) for _ in range(n_encoder_layers)]
    )
    preAttentionDecoder = tl.Serial(
        tl.ShiftRight(mode=mode),
        tl.Embedding(vocab_size=target_vocab_size,
d_feature=d_model),
        tl.LSTM(n_units=d_model)
    )
    return tl.Serial(
      tl.Select([0, 1, 0, 1]),
      tl.Parallel(inputEncoder, preAttentionDecoder),
      tl.Fn('PrepareAttention', prepareAttention, n_out=4),
      tl.Residual(tl.AttentionQKV(d_model,
n_heads=n_attention_heads, mode=mode)),
      tl.Select([0, 2]),
      [tl.LSTM(n_units=d_model) for _ in range(n_decoder_layers)],
      tl.Dense(target_vocab_size),
      tl.LogSoftmax()
    )
def GRUModel(input_vocab_size=500,
             target_vocab_size=500,
             d_model=512,
             n_encoder_layers=2,
             n_decoder_layers=2,
```

```
            n_attention_heads=3,
            mode='train'):
    inputEncoder = tl.Serial(
        tl.Embedding(vocab_size=input_vocab_size,
d_feature=d_model),
        [tl.GRU(n_units=d_model) for _ in range(n_encoder_layers)]
    )
    preAttentionDecoder = tl.Serial(
        tl.ShiftRight(mode=mode),
        tl.Embedding(vocab_size=target_vocab_size,
d_feature=d_model),
        tl.GRU(n_units=d_model)
    )
    return tl.Serial(
      tl.Select([0, 1, 0, 1]),
      tl.Parallel(inputEncoder, preAttentionDecoder),
      tl.Fn('PrepareAttention', prepareAttention, n_out=4),
      tl.Residual(tl.AttentionQKV(d_model,
n_heads=n_attention_heads, mode=mode)),
      tl.Select([0, 2]),
      [tl.GRU(n_units=d_model) for _ in range(n_decoder_layers)],
      tl.Dense(target_vocab_size),
      tl.LogSoftmax()
    )

def trainTask(stream):
    return training.TrainTask(
        labeled_data= stream,
        loss_layer= tl.CrossEntropyLoss(),
        optimizer= trax.optimizers.Adam(0.01),
        lr_schedule= trax.lr.warmup_and_rsqrt_decay(1000, 0.01),
        n_steps_per_checkpoint=10,
    )
```

```
def evalTask(stream):
    return training.TrainTask(
        labeled_data= stream,
        metrics=[tl.CrossEntropyLoss(), tl.Accuracy()]
    )
def train(model, trainStream, evalStream, outputFolder):
    return        training.Loop(model,        trainTask(trainStream),
eval_tasks=[evalTask(evalStream)], output_dir=outputFolder)

lSTMModelTrainLoop    =    train(LSTMModel(),    tokenizedTrainStream,
tokenizedEvalTrainStream, "/lstm")
GRUModelTrainLoop    =    train(GRUModel(),    tokenizedTrainStream,
tokenizedEvalTrainStream, "/gru")

from trax import models
transfromerModel = models.Transformer(
   input_vocab_size=500,
   d_model=512, d_ff=1024,
   n_heads=6, n_encoder_layers=4, n_decoder_layers=4,
   max_len=256, mode='train')
transformerModelTrainLoop          =          train(transfromerModel,
tokenizedTrainStream, tokenizedEvalTrainStream, "/lstm")
```

Код функції оцінки якості перекладу моделей

```
from sacrebleu.metrics import BLEU
import numpy as np
import pathlib

def load_data(path, isRefs=True):
  text = path.read_text(encoding='utf-8')
  lines = text.splitlines()
  return  [line  if  not  isRefs  else  line.split('\t')  for  line  in
lines]
```

```
def BLEU_score(refs, candidates, max_ngrams=4):
   bleu = BLEU(max_ngram_order=max_ngrams)
   return bleu.corpus_score(candidates, refs)

refs_10 = load_data(pathlib.Path("refs10.txt"))
candidates_10 = load_data(pathlib.Path("cands10.txt"), isRefs=False)
print("BLEU for sentances 10 lengths", BLEU_score(refs_10,
candidates_10, max_ngrams=3))

refs_20 = load_data(pathlib.Path("refs20.txt"))
candidates_20 = load_data(pathlib.Path("cands20.txt"), isRefs=False)
print("BLEU for sentances 20 lengths", BLEU_score(refs_20,
candidates_20))

refs_30 = load_data(pathlib.Path("refs30.txt"))
candidates_30 = load_data(pathlib.Path("cands30.txt"), isRefs=False)
print("BLEU for sentances 30 lengths", BLEU_score(refs_30,
candidates_30))
```

ДОДАТОК В

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	60 с.
	Інші документи	
2.	Презентаційні матеріали	__ плакатів

.	.	Прізвище та ініціали.	Підп.	Дата	
Розробив		Бєляєв В. С.			Дослідження та застосування методів глибинного навчання у задачах соціалізації людей з афазією
Перевірив		Рябова Н. В.			
Н.контр.		Малєєва І.А.			
Затв.		Філатов В.О.			

Шифр групи	Код напр./спец.
СШІм-21-2	122
ХНУРЕ кафедра ШІ	