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)

Методи синтезу надшвидкодіючих структур мовних систем штучного інтелекту

Виконав:
студент 2 курсу групи ІПЗм-21-3
Шульга В. В.
(прізвище, ініціали)

Спеціальність	121 – Інженерія програмного забезпечення
Тип програми	Освітньо-наукова
Керівник	проф. Четвериков Г. Г.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. Кафедри _____ З.В. Дудар

2023 р.

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 програма

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«___»_____ 202_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Шульзі Владиславу Володимировичу
_____
(прізвище, ім'я, по батькові)

1. Тема роботи «Методи синтезу надшвидкодіючих структур мовних систем штучного інтелекту» затверджена наказом університету від «___»_____ 202 __ р. №___
2. Термін подання студентом роботи до екзаменаційної комісії «__» _____ 202__ р.
3. Вихідні дані до роботи *структури даних, системи штучного інтелекту, синтез мовних структур, пояснювальна записка.*
4. Перелік питань, що потрібно опрацювати в роботі *мета роботи, аналіз предметної галузі і постановка задачі, дослідження методів синтезу мовних систем штучного інтелекту, зокрема надшвидкодіючих структур, вивчення можливостей їх використання у інформаційних системах.*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної області	25.02.2023	виконано
2	Постановка задачі	10.03.2023	виконано
3	Проведення дослідження	01.04.2023	виконано
4	Підготовка пояснювальної записки	01.05.2023	виконано
5	Підготовка презентації та доповіді	03.05.2023	виконано
6	Попередній захист	05.05.2023	виконано
7	Перевірка на академічний плагіат	05.05.2023	виконано
8	Нормоконтроль	10.05.2023	виконано
9	Рецензування	11.05.2023	виконано
10	Знесення диплома в електронний архів	12.05.2023	виконано
11	Допуск до захисту у зав.кафедри	12.05.2023	виконано
1	Аналіз предметної області	25.02.2023	виконано

Дата видачі завдання _____ 202__ р.

Студент _____
(підпис)

Керівник роботи _____ _____

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 68 с., 12 рис., 5 табл., 12 джерел.

ШТУЧНИЙ ІНТЕЛЕКТ, НЕЙРОННІ МЕРЕЖІ, МОВНІ МОДЕЛІ, СИНТЕЗ МОВИ, НАДШВИДКОДІЮЧІ СТРУКТУРИ, ГЕНЕРАТИВНІ МОДЕЛІ

Об’єктом дослідження є методи синтезу надшвидкодіючих структур мовних систем штучного інтелекту.

Метою роботи є дослідження методів впровадження синтезу людської мови у системах штучного інтелекту та нейронних мережах з урахуванням оптимальної швидкості.

Результатом роботи є порівняльне дослідження методів синтезу мовних моделей на основі генеративних нейронних мереж. Дослідження відбувається на системи, яка використовує штучний інтелект для роботи та спілкуванням з користувачами за допомогою обміну текстовими повідомленнями.

The explanatory note contains: 68 pages, 12 figures, 5 tables, 12 sources.

ARTIFICIAL INTELLIGENCE, NEURAL NETWORKS, LANGUAGE MODELS, LANGUAGE SYNTHESIS, HIGH-SPEED STRUCTURES, GENERATIVE MODELS

The object of research is methods of synthesis of ultra-fast structures language systems of artificial intelligence.

The method of work is the study of the methods of implementing the synthesis of human language in artificial intelligence systems and neural networks with optimal speed.

The result of the work is a comparative study of language model synthesis methods based on generative neural networks. The research takes place on systems that use artificial intelligence to operate and communicate with users through text messaging.

Умови публікації пояснювальної записки

Я,

Шульга Владислав Володимирович

(прізвище, ім’я, по батькові)

студент(ка) групи ІПЗм-21-3 здобувач вищої освіти на другому (магістерському) рівні

кафедра Програмної інженерії

(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему

Методи синтезу надшвидкодіючих структур мовних систем штучного інтелекту,

(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`

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ШІ – штучний інтелект

НСМ – надшвидкісні мовні системи

РНН – рекурентна нейронна мережа

ККМ – коефіцієнт кореляції Метьюза

BLEU – bilingual evaluation understudy

GLUE – General Language Understanding Evaluation

ROUGE – Recall-Oriented Understudy for Gisting Evaluation

GPU – Graphics processing unit

BERT – Bidirectional Encoder Representations from Transformers

ВСТУП

Штучний інтелект став важливою технологією 21-го століття, революціонізувавши різні сфери, починаючи від охорони здоров’я та фінансів до транспорту та сфери розваг.

Внаслідок розповсюдження ШІ, також зріс попит на мовні системи, які можуть розуміти, інтерпретувати та генерувати людську мову. Однак із ускладненням цих систем зросли і вимоги до обчислень, що призводить до збільшення часу обробки та затримки. Проблематика особливо гостро стоїть для додатків, які потребують відповідей у реальному часі, таких як голосові помічники та чат-боти.

Мовні системи, що працюють на основі штучного інтелекту, особливо важливі для забезпечення ефективного спілкування між людьми та машинами, що сприяє покращенню взаємодії з користувачем і підвищенню продуктивності. Однак розробка мовних систем, здатних обробляти величезні обсяги даних і надавати відповіді в реальному часі, може бути складним завданням.

Практичні застосування надшвидких мовних систем штучного інтелекту мають широкий спектр результатів різного ступеню ефективності. Наприклад, чат-боти та віртуальні помічники можуть скористатися перевагами таких систем, які можуть обробляти введені користувачем дані та генерувати відповіді в режимі реального часу.

Надшвидкісні мовні системи є критично важливим компонентом сучасних комунікаційних технологій, що забезпечує ефективне та ефективне спілкування між людьми та машинами. Ці підхои відіграють значну роль у різних сферах виключаючи ШІ, такі як робота зі зв’язком, фінансами і транспортом.

Наприклад, у телекомунікаційній галузі надшвидкісні мовні системи необхідні для розпізнавання голосу та транскрипції, що дозволяє клієнтам швидко й точно орієнтуватися в системах інтерактивного голосового відповіді. У фінансах НСМ можна використовувати для аналізу великих обсягів текстових даних, щоб визначити ринкові тенденції, настрої та потенційні ризики,

дозволяючи трейдерам та інвесторам швидко приймати обґрунтовані рішення. У транспорті ці системи можуть сприяти спілкуванню в режимі реального часу між водіями та диспетчерами, підвищуючи ефективність і безпеку транспортних мереж.

Розробка НСМ є складним завданням, яке потребує використання складних методів, таких як обробка природної мови, машинне навчання та стиснення даних. Щоб досягти високопродуктивних результатів із мінімальною затримкою, дослідники досліджували різні методи синтезу надшвидких структур мовних систем. Ці методи включають пошук архітектури, скорочення, квантування та дистиляцію знань, серед іншого.

Підсумовуючи, НСМ є важливим компонентом сучасних комунікаційних технологій і має широке застосування як у сфері ШІ, так і поза її межами. Використовуючи передові технології та методи, дослідники можуть розробляти ефективні та масштабовані мовні моделі, які можуть надавати відповіді в реальному часі та трансформувати різні галузі, зокрема телекомунікації, фінанси та транспорт

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Структура мовних систем

Мовні системи — це складні структури, які дозволяють людям спілкуватися один з одним. Ці системи складаються з різних компонентів, які працюють разом, щоб полегшити обмін інформацією. Структури мовних систем можна розділити на два типи: поверхневу структуру та глибинну структуру.

Поверхнева відноситься до фактичних слів і фраз, які використовуються в мові. Ці слова та фрази розташовані в певному порядку, щоб утворити речення та передати значення. Поверхнева структура мови може змінюватися залежно від таких факторів, як правила граматики, синтаксис і словниковий запас.

У письмовій формі поверхнева структура представлена такими символами, як літери та знаки пунктуації. Ці символи розташовані в певному порядку, щоб скласти слова, які потім об’єднуються, щоб сформувати речення та абзаци.

З іншого боку знаходиться глибинна структура, яка стосується основного значення або понять, що стоять за словами та фразами, що використовуються в мові. Це відноситься до таких речей, як контекст, у якому використовується мова, наміри мовця та культурне походження мовця та слухача. Глибинна структура є більш абстрактною, і її важче розрізнити, ніж структуру поверхні.

У системах природної мови[1] глибока структура часто визначається через невербальні ознаки, такі як тон голосу, вираз обличчя та мова тіла. У системах штучної мови глибока структура може бути реалізована за допомогою алгоритмів машинного навчання та методів обробки природної мови.

У деяких випадках одна глибинна структура може бути виражена кількома поверхневими, залежно від контексту та передбачуваного значення мовця. Наприклад, речення «я бачу вітрину з телескопом» може мати кілька поверхневих структур, залежно від того, чи «з телескопом» змінює «бачу» чи «вітрина». Глибинна структура речення залишається незмінною, але поверхнева структура може змінюватися.

Наприклад, програмне забезпечення для мовного перекладу використовує глибокий аналіз структури, щоб зрозуміти основне значення речення однією мовою, а потім перекласти його іншою мовою. Подібним чином чат-боти та голосові помічники використовують глибокий структурний аналіз, щоб зрозуміти введені користувачем дані та надати відповідні відповіді.

Реалізація мовних систем може приймати різні форми, починаючи від розмовної мови до письмової мови та мови жестів. Ці різні форми впровадження можуть мати різні структури[2], але всі вони виконують головну функцію — дозволяють людям спілкуватися між собою.

1.2 Аналіз актуальності реалізації надшвидкодіючих мовних систем, її стан та проблематика

Сфера мовних систем штучного інтелекту зазнала стрімкого зростання та розвитку в останні роки. МСН штучного інтелекту — це комп’ютерні системи, призначені для розуміння, генерування та керування природною мовою, і вони мають широкий спектр застосувань, включаючи чат-ботів, віртуальних помічників, машинний переклад, аналіз настроїв і класифікацію тексту.

Нещодавні досягнення в цій галузі призвели до створення дедалі складніших мовних моделей, таких як BERT, GPT-2-4. Ці мовні моделі продемонстрували чудову продуктивність у різних завданнях обробки природної мови, таких як генерація тексту, аналіз настроїв і відповіді на запитання.

Однак суттєвим недоліком цих мовних моделей є вимогливість в обчислювальному плані та потреба великого обсягу пам’яті для роботи, що обмежує їх застосування в програмах реального часу. Моделі з мільярдами параметрів вимагають десятків або сотень гігабайт пам’яті, що може стати серйозною проблемою для розгортання цих моделей на пристроях з обмеженою ємністю пам’яті або привести до зростання витрат на обслуговування та розгортання з використанням пристроїв зберігання інформації.

Іншим недоліком цих мовних моделей є їхня обмежена здатність узагальнювати нові завдання та домени. Незважаючи на те, що вони продемонстрували високу продуктивність у широкому діапазоні завдань, вони можуть мати труднощі, коли подаються нові або несподівані вхідні дані. Це особливо проблематично в реальних додатках, де користувачі можуть ставити несподівані або неочікувані запитання, і система повинна мати можливість надавати точні та відповідні відповіді.

Мовні моделі також викликають занепокоєння щодо їх впливу на навколишнє середовище. Обчислювальні ресурси, необхідні для навчання та використання цих моделей, мають значні витрати на енергію, що може сприяти зміні клімату та погіршенню навколишнього середовища за рахунок значної експлуатації природних ресурсів на навчання або розробку систем, енергоефективність яких може викликати сумніви.

Додатковим ризиком у сучасних НСМ є конфіденційність. Вона викликає занепокоєння у користувачів, коли йдеться про мовні моделі, оскільки вони часто вимагають великих обсягів персональних даних для ефективного навчання. Існує ризик того, що ці дані можуть бути використані в неетичних цілях, наприклад для ідентифікації окремих осіб або груп на основі їхньої мови чи особистої інформації. Щоб вирішити ці проблеми, розробники повинні надавати пріоритет захисту конфіденційності та прозорості в своїх методах збору та використання даних.

Упередженість — ще одна етична проблема при розробці мовних моделей ШІ. Системи, навчені на основі упереджених або нерепрезентативних даних, можуть увічнити та посилити існуючі суспільні упередження та стереотипи, що потенційно здатне призвести до шкоди та дискримінації маргіналізованих груп. Щоб забезпечити відповідальний і етичний розвиток технології штучного інтелекту, важливо визнавати та пом’якшувати упередженість у мовних моделях, включаючи регулярні аудити та тестування моделей на справедливість і неупередженість.

Підсумовуючи етичні та соціальні сторони систем, контроль надшвидких мовних моделей викликає занепокоєння щодо підзвітності та прозорості. У деяких випадках мовні моделі можуть приймати рішення, які мають значні наслідки, наприклад, у юридичному чи медичному контексті. Розробники повинні переконатися, що ці моделі є прозорими, зрозумілими та відповідальними за свої рішення, особливо в ситуаціях із високими ставками. Незважаючи на відносну новизну технологій ШІ, вже йдуть спроби розробити правила та рекомендації щодо розробки та розгортання технологій ШІ, які націлені на розв’язання етичних проблем. До цього відносяться пропозиція закону про підзвітність алгоритмів у США та закон про ШІ від Європейської комісії.

Щоб вирішити перелік проблем у сучасних НСМ на основі ШІ, зростає потреба в більш прозорих та ефективних моделях мови, які можуть обробляти дані в режимі реального часу. Це призвело до розробки нових методів і підходів для синтезу надшвидких структур мовних систем. Ці методи спрямовані на зменшення розміру та складності мовних моделей, зберігаючи або навіть покращуючи їх продуктивність.

1.3 Огляд мовних моделей. Різновидності та архітектура на основі нейронних мереж

Мовні моделі (рис. 1.1) — це обчислювальні моделі, які використовуються для обробки та розуміння людської мови. Існує кілька різних типів мовних моделей, але найпоширенішою архітектурою, яка використовується сьогодні, є архітектура нейронної мережі.

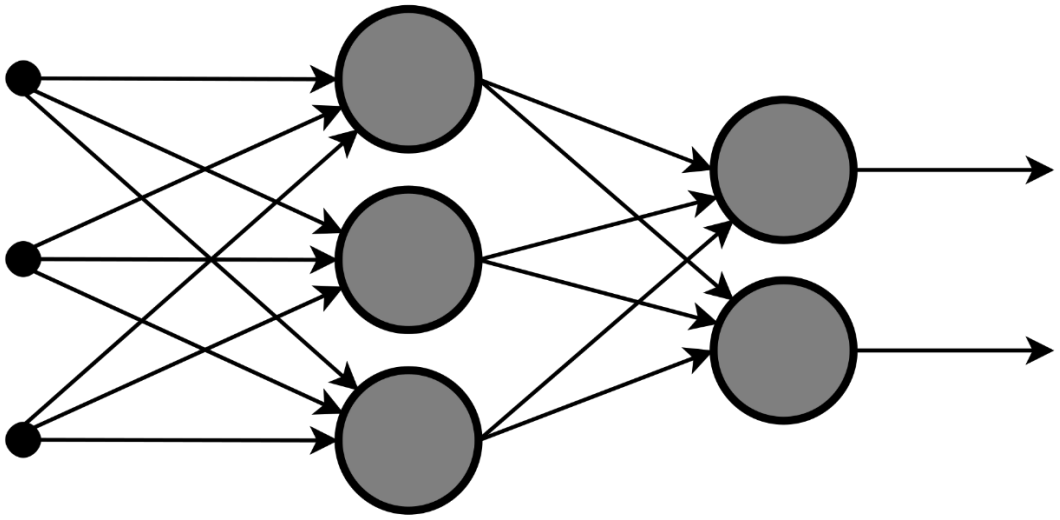

Рисунок 1.1 — Багаторівнева мовна модель

Архітектура нейронної мережі, яка використовується для мовних моделей, зазвичай є типом рекурентної нейронної мережі (РНН) або моделі на основі трансформатора. Ці моделі розроблені для того, щоб мати можливість аналізувати послідовності слів або інших мовних одиниць, і їх можна навчити передбачати, яке слово чи послідовність слів, ймовірно, буде наступним.

РНН — це тип нейронної мережі[3], яка може обробляти послідовності вхідних даних, передаючи інформацію від одного часового кроку до іншого. У контексті мовних моделей кожен часовий крок відповідає слову чи іншій мовній одиниці в послідовності тексту. РНН здатні вловлювати послідовні залежності між словами, що дозволяє їм моделювати структуру мови.

Основна ідея такої мовної моделі полягає в тому, щоб обробляти вхідні послідовності тексту по одному слову та використовувати контекст попередніх слів для прогнозування розподілу ймовірностей наступного слова в послідовності (рис 1.2). Архітектура РНН добре підходить для цього завдання, оскільки дозволяє моделі відстежувати попередні слова в послідовності та використовувати їх для кращих прогнозів.

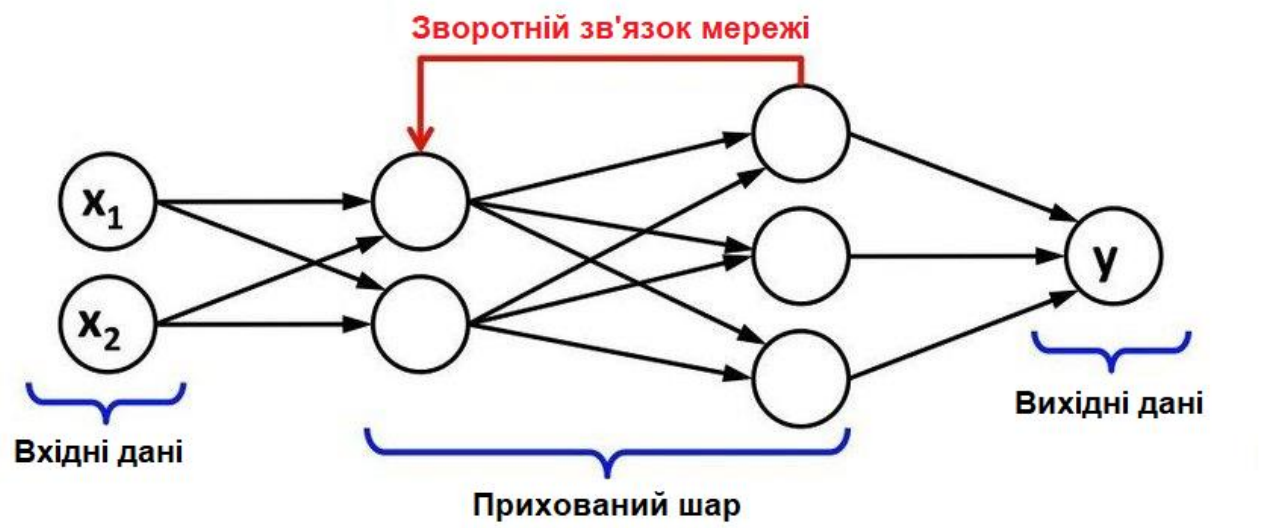

Рисунок 1.2 — Рекурентна нейронна мережа

У мовній моделі РНН вхідна послідовність зазвичай обробляється серією рівнів, які включають цикл зворотного зв’язку. Цикл зворотного зв'язку дозволяє моделі підтримувати прихований стан, який фіксує контекст попередніх слів у послідовності. На кожному кроці часу прихований стан оновлюється на основі поточного введеного слова та попереднього прихованого стану, і цей оновлений прихований стан потім використовується для прогнозування наступного слова в послідовності.

Навчання такої моделі зазвичай передбачає використання великого набору тексту для вивчення розподілу ймовірностей наступного слова в послідовності з урахуванням контексту попередніх слів. Це робиться шляхом мінімізації функції втрат, яка вимірює різницю між прогнозованим розподілом ймовірностей і справжнім розподілом ймовірностей.

Альтернативною є нейронна мережа прямого зв’язку — FNN (Feedforward neural network), яка складається з вхідного рівня, одного або кількох прихованих шарів і вихідного рівня (рис. 1.3). Її можна вважати більш спрощеною версією РНН, так як вона не має циклічної структури і загалом була розроблена раніше.

У цій моделі вхідний рівень отримує послідовність слів, які зазвичай представлені як одноразові вектори або вбудовування. Ці вхідні вектори потім перетворюються прихованими шарами, які навчаються витягувати релевантні

ознаки з вхідної послідовності. Рух інформації відбувається в одному напрямку, на відміну від РНН і позначений стрілками в бік вихідного шару.

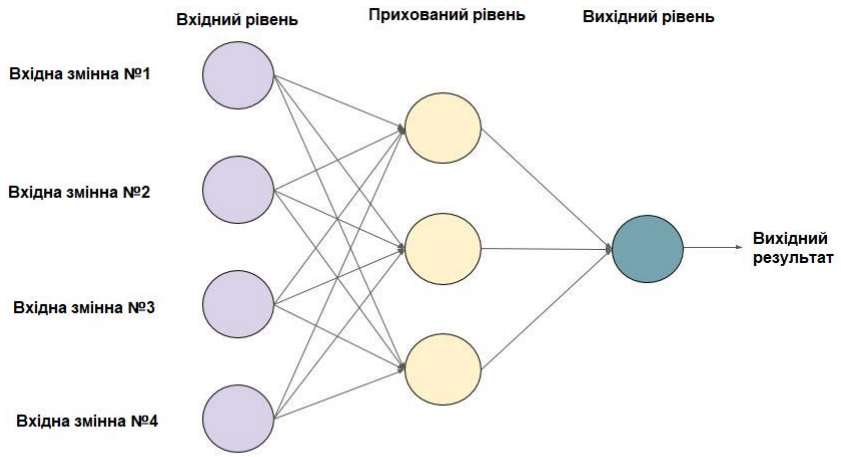

Рисунок 1.3 – Нейронна мережа прямого зв’язку з одним прихованим рівнем і 3 нейронами

Ще одним варіантом моделі на основі нейронної мережі є на базі трансормера[4]. Ці моделі широко використовуються в задачах обробки природної мови, таких як моделювання мови, класифікація тексту, машинний переклад і генерація тексту.

Модель трансформатора заснована на механізмі самоуважності, який дозволяє моделі звертати увагу на всі слова в реченні одночасно і вловлювати довгострокові залежності між ними. Це є значною відмінністю від РНН, яка обробляє послідовності одна за одною і може обробляти довгі запити досить повільно.

І РНН, і моделі на основі трансформаторів зазвичай навчаються за допомогою методики, яка називається навчанням під наглядом. Це передбачає передачу моделі послідовності слів, а потім навчання її передбаченню наступного слова в послідовності. Роблячи це на великому наборі даних, модель може навчитися генерувати правдоподібний текст, який нагадує людську мову.

1.4 Розгляд підходів до синтезу мови на основі алгоритмів

Під час синтезу мовних систем на основі алгоритмів, основна увага приділяється використанню обчислювальних моделей для створення та обробки мови. Ці методи можуть базуватися на різних техніках, таких як статистичні моделі, системи на основі правил, дерева рішень,.

Кінцева мета синтезу цього підходу полягає в тому, щоб дозволити машинам розуміти, обробляти та генерувати людську мову, яку можна використовувати в широкому діапазоні додатків, таких як обробка природної мови, чат-боти, машинний переклад і розпізнавання мовлення.

Використовуючи алгоритми та обчислювальні моделі, можна спроектувати мовні системи для роботи на надвисоких швидкостях, що дозволяє обробляти та генерувати мову в реальному часі. Це має так наслідки, як підвищення ефективності та результативності взаємодії людини та системи, а також для вдосконалення рівня обробки та синтезу природної мови.

Одним з прикладом такого методу є генетичні алгоритми (рис 1.4) — це тип оптимізації, який імітує процес природного відбору для заданої функції або набору параметрів. Коли справа доходить до синтезу мовних систем, генетичні алгоритми можна використовувати для оптимізації параметрів вже підготовленої мовної моделі.

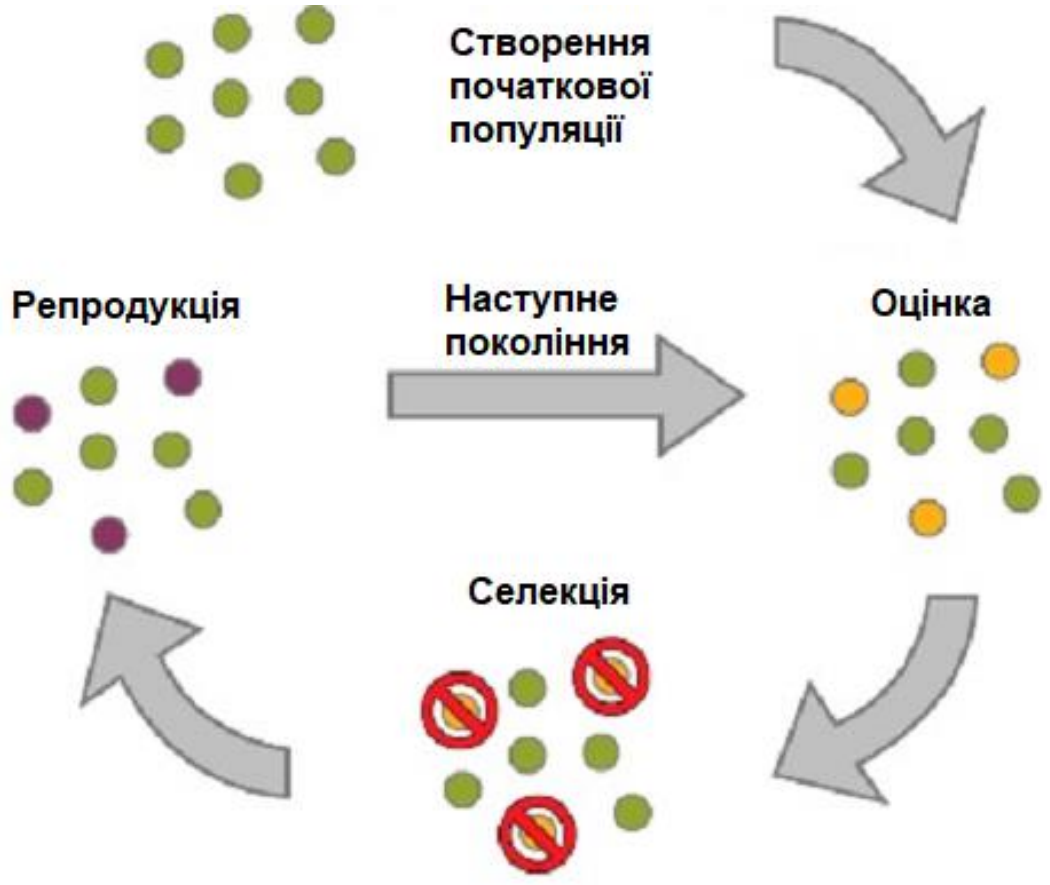

Рисунок 1.4 – Візуалізація генетичного алгоритму

Основний процес використання систем включає наступні етапи:

– визначення функції відповідності – функція використовується для оцінки продуктивності моделі;

– створення початкової сукупності – початкова сукупність складається з набору випадково згенерованих мовних моделей;

– застосування генетичних операторів (схрещування, мутація та відбір) – використовуються для створення нових мовних моделей із початкової популяції; схрещування передбачає поєднання двох батьківських сутностей для створення нової, а мутація передбачає внесення випадкових змін до одної моделі; відбір має за мету вибір найпридатніших сутностей популяції для використання в наступному поколінні;

– оцінка придатності нового покоління – функція придатності застосовується до нової популяції, щоб оцінити її ефективність і визначити, які з них слід використовувати для наступного покоління;

– повторення процесу – кроки 3 і 4 повторюються, доки не буде отримано задовільну модель мови.

Альтернативний до вищевказаного і достатньо розповсюджений метод — застосування Баєсових мереж. Це ймовірнісні графічні моделі, які представляють зв’язки між змінними та їхні залежності ймовірнісним способом.

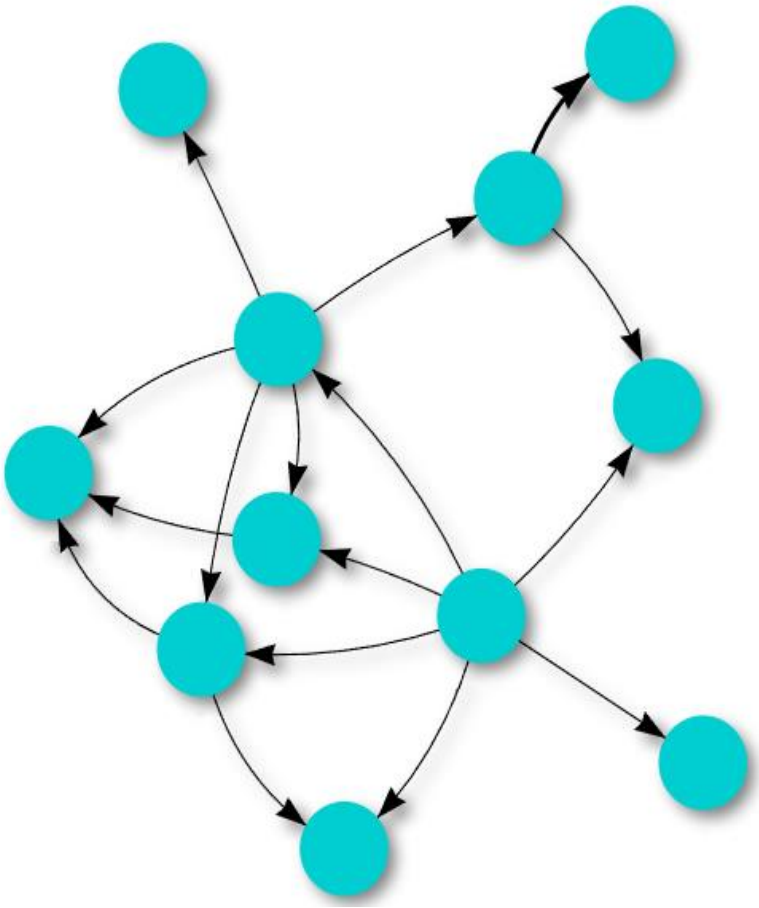

Рисунок 1.5 – Візуалізація Баєсової мережі

Коли справа доходить до синтезу мовних систем, Баєсові мережі можна використовувати для моделювання розподілу ймовірностей слів і фраз у даній мові, а також для виведення найбільш ймовірних слів або фраз за певних вхідних

даних. Процес використання мереж для синтезу мовних систем включає наступну послідовність дій:

– визначення змінних – змінні представляють слова та фрази даною мовою. Кожна змінна пов’язана з розподілом ймовірностей, який представляє ймовірність цієї змінної з урахуванням інших змінних у моделі;

– побудова мережі – створення сутності виконується шляхом визначення зв’язків між змінними в моделі. Зазвичай це робиться шляхом визначення набору таблиць умовної ймовірності, які визначають імовірність кожної змінної з урахуванням її батьківських змінних у мережі;

– вивчення параметрів – параметри моделі, які відповідають ймовірностям у таблицях умовної ймовірності, вивчаються з навчального набору даних, це передбачає оцінку ймовірностей на основі спостережених даних;

– висновок – коли параметри вивчені, Байєсова мережа може бути використана для висновку, з огляду на певні вхідні дані, мережа може бути використана для виведення найбільш імовірних слів або фраз, які слідують за цим введеним.

Використовуючи Байєсові мережі для моделювання розподілу ймовірностей слів і фраз у даній мові, можна створювати моделі, які можуть використовуватися в широкому діапазоні додатків, від розпізнавання мовлення та машинного перекладу до обробки природної мови та чат-ботів. . Байєсові мережі також можна використовувати в поєднанні з іншими методами, такими як генетичні алгоритми та системи на основі правил, для подальшого покращення продуктивності мовних моделей.

1.4 Задачі синтезу надшвидких структур мовних систем штучного інтелекту

Завдання синтезу надшвидких структур мовних систем штучного інтелекту полягає в розробці моделей, які можуть обробляти великі обсяги даних

і надавати відповіді в реальному часі з мінімальною затримкою. Метою цього завдання є вирішення проблеми розробки ефективних і масштабованих мовних систем, які можуть задовольнити вимоги сучасних програм.

Однією з головних проблем мовних систем штучного інтелекту є їхні обчислювальні вимоги, які можуть значно зрости в міру ускладнення моделей. Це може призвести до збільшення часу обробки та затримки, що ускладнить розробку мовних моделей, які можуть надавати відповіді в реальному часі. Крім того, оскільки попит на мовні системи ШІ продовжує зростати, потреба в моделях, які можуть швидко й ефективно обробляти дані, стає все більш критичною.

Обчислювальна потужність, необхідна для навчання мовних моделей штучного інтелекту, що може зайняти дуже багато часу та інтенсивних обчислень. Процес навчання зазвичай передбачає оптимізацію мільйонів параметрів за допомогою кількох ітерацій, що вимагає великих обсягів пам’яті та потужності обробки. У результаті розробка великомасштабних мовних моделей може бути надзвичайно дорогою, обмежуючи їх практичне застосування

2. ОГЛЯД НАУКОВОЇ ТА ПАТЕНТНОЇ ЛІТЕРАТУРИ

2.1 Огляд наукових конференцій

Під час аналізу предметної області дослідження, для отримання актуальної інформації щодо проблем та методів вирішення, було прийнято рішення звернутися наукових конференцій, які були проведені колегами з різних частин світу щодо відповідної теми.

Через схожу проблемну галузь до мовних моделей, можна звернутися до конференці з емпіричних методів обробки природної мови EMNLP (англ. Conference on Empirical Methods in Natural Language Processing) – вона присвячена аналізу, моделюванну та розумінну даних природної мови і за важливісю є однією з провідних у галузі обробки природної мови. EMNLP 2021, що відбулася в листопаді 2021 року, мала кілька релевантних доповідей, зокрема на ній розглядалися матеріали, які стосувалися наступних тем:

- швидкість мовних моделей;
- обробка великих відних даних;
- навчання та навчальні вибірки.

Підсумовуючи головні тези, автори роблять висновок, про те, що Ключовою проблемою при синтезі надшвидких структур мовних систем штучного інтелекту є компроміс між точністю та швидкістю. Одним із запропонованих рішень є використання методів дистиляції для передачі знань від більшої та повільнішої моделі вчителя до меншої та швидшої моделі учня. Наприклад, у документі, представленому на конференції, запропоновано до використання структуру дистиляції під назвою «S2S Distillation», яка забезпечує високі рівні стиснення з мінімальною втратою продуктивності.

S2S (Sequence-to-Sequence) Distillation — це тип дистиляції знань, який спеціально зосереджений на стисненні моделей послідовності (S2S), які широко використовуються в завданнях нейронного машинного перекладу (NMT). Основна відмінність між S2S Distillation та іншими типами дистиляції знань

полягає в специфічних техніках, які використовуються для стиснення та переміщення інформації від великої моделі вчителя в меншу модель учня.

Традиційні методи дистиляції (рис. 1.2) знань зазвичай передбачають навчання меншої моделі учня для імітації поведінки більшої моделі вчителя шляхом мінімізації різниці між їхніми результатами на наборі навчальних даних. Це передбачає порівняння різниці, або втрат після навчання, в роботі моделей (loss на рисунку) за допомогою використання функції softmax, обчислення ймовірностей прогнозів моделі вчителя, а потім використання цих ймовірностей як «м’яких цілей» для навчання моделі учня.

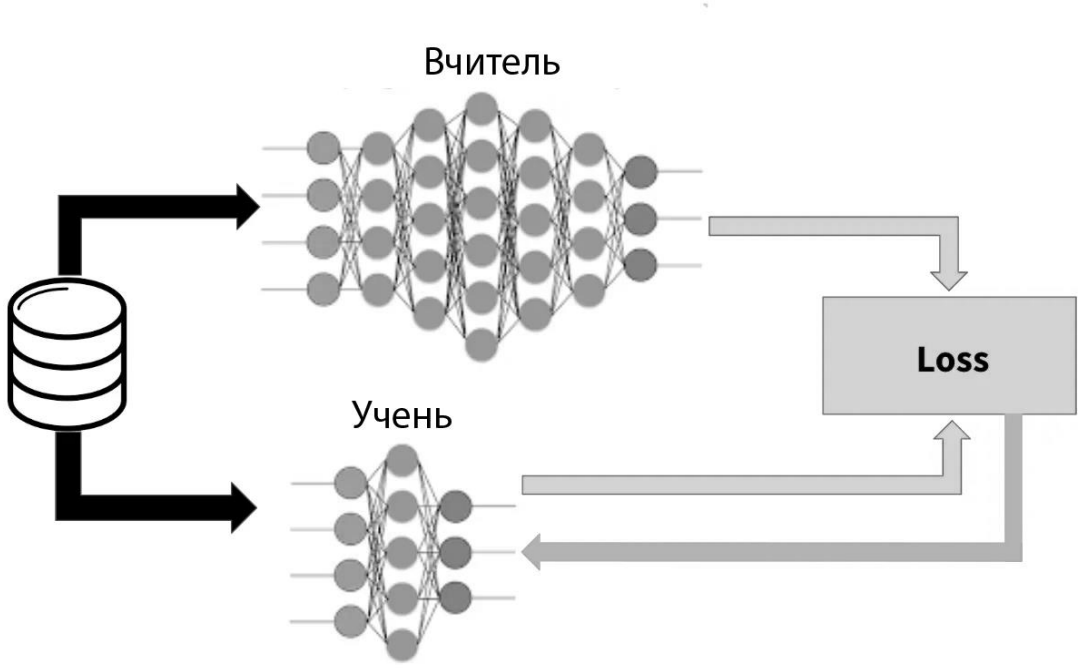

Рисунок 2.1 — Візуалізація дистиляції знань для моделі-вчителя та моделі-учня

На відміну від інших підходів, S2S дистиляція використовує техніку поділу на фрагменти[5], щоб розбивати вхідні та вихідні послідовності на менші фрагменти, які потім окремо обробляються моделлю студента. Це допомагає зменшити обчислювальні витрати на процес дистиляції, зберігаючи високу якість перекладу. Крім того, S2S дистиляція використовує спеціальний механізм

уваги для вирівнювання фрагментів між моделями викладача та студента, що ще більше покращує якість дистиляції.

Іншою релевантним заходом є конференція з систем на базі нейронних мереж та обробки інформації NeurIPS (англ. Conference on Neural Information Processing Systems) – спеціалізується на сфері машинного навчання, крім цього до неї також входять доповіді на тему когнітивної науку, психології, комп'ютерного зору, статистичної лінгвістики та теорії інформації. На ній також визначається схожа проблематика НСМ - ключовою проблемою при синтезі мовних систем штучного інтелекту є потреба в ефективних і масштабованих алгоритмах навчання.

Одним із запропонованих рішень для пришвидшення навчання є використання адаптивних алгоритмів оптимізації, які можуть динамічно регулювати темпи навчання або інші гіперпараметри під час навчання. Наприклад, у статті, представленій на NeurIPS 2021, запропоновано новий адаптивний оптимізатор RangerLARS, який забезпечує швидшу конвергенцію та кращу продуктивність у мовних завданнях.

RangerLARS (анлг. Ranger with LARS) — це алгоритм оптимізації для глибокого навчання, який поєднує дві методики, а саме оптимізатор Ranger і LARS (англ. Layer-wise Adaptive Rate Scaling).

Оптимізатор Ranger — це сучасний оптимізатор, який поєднує оптимізатор Rectified Adam (RAdam) і оптимізатор LookAhead. Було показано, що він покращує швидкість конвергенції та точність глибоких нейронних мереж у різноманітних завданнях, включаючи класифікацію зображень, виявлення об’єктів та мовне моделювання.

Оптимізатори — це алгоритми або методи, які використовуються для мінімізації функції помилок (функції втрат) або для максимізації ефективності виробництва. Вони можуть бути представлені у вигляді математичних функцій, які залежать від параметрів моделі, які можна вивчати, наприклад вагових коефіцієнтів і зміщень. Оптимізатори допомагають знати, як змінити ваги та швидкість навчання нейронної мережі, щоб зменшити втрати.

З іншого боку, LARS — це метод планування швидкості навчання, який динамічно регулює темп навчання кожного шару в глибокій нейронній мережі на основі локального градієнта та вагової норми. Було показано, що ця техніка покращує продуктивність узагальнення глибоких нейронних мереж і запобігає переобладнанню.

Поєднуючи обидва методи, RangerLARS має за мету вирішити наступні проблеми під час оптимізації:

– повільна конвергенція – за допомогою оптимізатора Ranger, може прискорити конвергенцію глибоких нейронних мереж, надаючи швидші та точніші оновлення параметрів моделі.

– погане узагальнення – використання LARS для коригування швидкості навчання на основі локального градієнта та норми ваги дає можливість запобігти переобладнанню та покращити ефективність узагальнення моделі.

– нестабільність – об’єднання всіх перелічених методів призводить до більш стабільного процесу оптимізації, що може зменшити дисперсію оновлень і підвищити надійність моделі.

2.2 Огляд наукових публікацій

У публікації «Few-Shot Learning for Language Generation with Generative Pretrained Transformer 3», яка була розглянута на конференції EMNLP 2021, автори розглядають метод дозованого (поодинокого) навчання[6] нейронних мереж (англ. few-shot). Документ представляє підхід до навчання для створення мови з використанням моделі GPT-3.

Проблематика доповіді, за словами авторів, відноситься до навчання мовних моделей з нуля, яке може займати багато часу та ресурсів, особливо для завдань з обмеженими навчальними даними.

Поодиноке навчання стосується практики подачі в модель машинного навчання дуже невеликої кількості навчальних даних, щоб керувати її

прогнозами, наприклад: кілька прикладів під час логічного висновку, на відміну від стандартних методів тонкого налаштування[7], які вимагають відносно великої кількості тренувальні дані для попередньо навченої моделі для точної адаптації до бажаного завдання.

Дозоване навчання складається з наступних частин (рис. 2.1):

– опис задачі – короткий опис того, що має робити модель,

– приклади – набір прикладів, які показують, що модель має передбачити

– запит (англ. prompt) – початок нового прикладу, який модель має завершити, згенерувавши відсутній текст

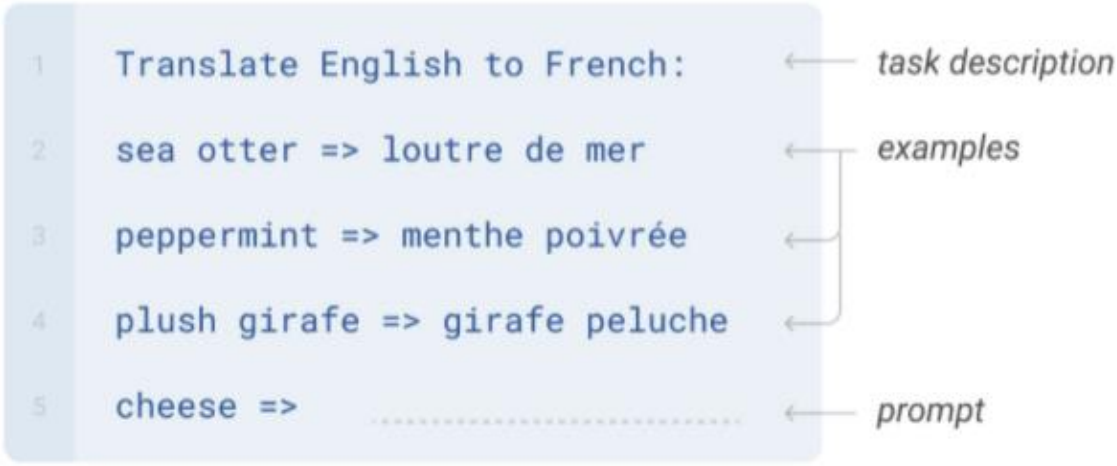

Рисунок 2.2 — Структура вхідних даних для методу few-shot навчання з метою перекладу з англійської на французьку мову

Створення невеликих складових (прикладів) може бути складною задачею, оскільки існує необхідність сформулювати «завдання», яке ви поставлене перед моделлю і яке вона повинна виконувати через них. Поширеною проблемою є те, що моделі, особливо малих розмірів, можуть бути більш чутливими до способу написання прикладів.

Підхід до оптимізації поодинокого навчання[8] полягає в тому, щоб вивчити загальне представлення для завдання, а потім навчити спеціалізовані класифікатори на основі цього представлення.

Відносно підходу one-shot (кількість прикладів для певного класу дорівнює одному) та zero-shot (схожий метод, коли машина навчають, як вчитися на даних без необхідності доступу до самих даних, на відміну від дозованого (англ. few-shot) підходу, не має точки зору під час передачі даних), неухильно покращується (рис. 2.2) точність разом із розміром моделі, продуктивність кількох кадрів зростає швидше, демонструючи, що більші моделі є більш досвідченим у навчанні в контексті.

Моделі однократного та кількакратного навчання можна підготувати швидше, оскільки вони потребують лише невеликої кількості позначених даних. Моделі навчання zero-shot можуть вимагати більше обчислювальних ресурсів і часу для навчання, оскільки їм потрібно вивчати зв’язки між різними класами на основі їхніх атрибутів.

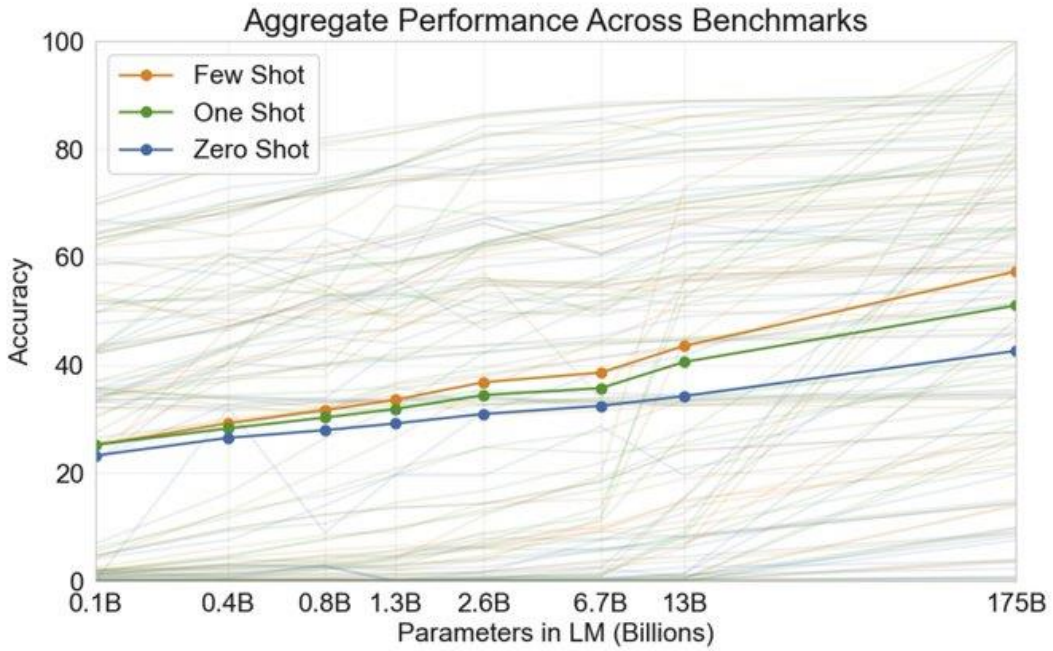

Рисунок 2.3 — Порівняння few-shot, one-shot та zero-shot підходів у навчанні моделі GPT-3

Few-shot техніка здебільшого використовується в комп’ютерному зорі, незважаючи на це, частина останніх мовних моделей (EleutherAI, GPT-Neo та

GPT-3), мають можливість використовувати цей підхід в обробці природної мови.

Щоб вирішити проблему навчання моделей з нуля, автори пропонують підхід до тренування підходами в кількома ітерацій, який налаштовує попередньо підготовлену модель GPT-3, використовуючи лише кілька прикладів. Автори оцінюють підхід на кількох контрольних наборах даних і демонструють, що він може генерувати високоякісний текст уже після навчання на кількох прикладах, що робить його корисним для завдань, які потребують синтезу тексту в режимі реального часу.

Як ще одну, варту уваги публікацію, через те, що у ній розглядаються актуальні проблеми швидкості навчання, можна визначити «Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism».

У цьому документі пропонується новий підхід до навчання надшвидких мовних моделей за допомогою паралелізму[9] моделей. Ключовою проблемою, яку автори розглядають у цій статті, є вимоги до пам’яті та обчислень для навчання великомасштабних мовних моделей. На момент публікації роботи, існуючі мовні моделі, такі як BERT і GPT-2, мають мільйони[10] або навіть мільярди параметрів, для навчання яких потрібна величезна кількість обчислювальних ресурсів. Це може ускладнити масштабування цих моделей для додатків у реальному часі або подальше покращення їх продуктивності.

Щоб вирішити цю проблему, пропонується новий підхід під назвою Megatron-LM, який використовує паралелізм моделей для розподілу вимог до обчислень і пам’яті між кількома GPU. Цей підхід дозволяє дослідникам тренувати надшвидкісні мовні моделі з мільярдами параметрів, використовуючи сотні чи навіть тисячі графічних процесорів.

Архітектура розроблена таким чином, щоб мати можливість до розпаралелювання, з кожним графічним процесором, відповідальним за підмножину параметрів моделі та обчислень. Це дозволяє дослідникам

тренувати модель на величезних обсягах даних, включаючи текстові корпуси з трильйонами токенів.

Автори оцінили цей підхід для ряду завдань обробки природної мови, включаючи моделювання мови, машинний переклад і класифікацію тексту. Вони виявили, що цей підхід дозволяє досягти значної продуктивності для всіх завдань і є значно швидшим і ефективнішим, ніж попередні методи.

Для ефективного навчання моделі, Megatron-LM використовує комбінацію паралелізму даних і конвеєрного паралелізму. Паралелізм даних передбачає поділ навчальних даних на менші пакети та їхню паралельну обробку на різних графічних процесорах. Конвеєрний паралелізм передбачає поділ моделі на етапи та паралельну обробку кожного етапу на різних графічних процесорах.

Архітектура Megatron-LM (рис. 2.4) складається з кількох шарів[11] трансформаторних блоків, причому кожен блок містить кілька центрів уваги та рівні прямого зв’язку. Вхідними даними моделі є послідовність токенів, а виходом є послідовність прогнозів для наступного токена в послідовності.

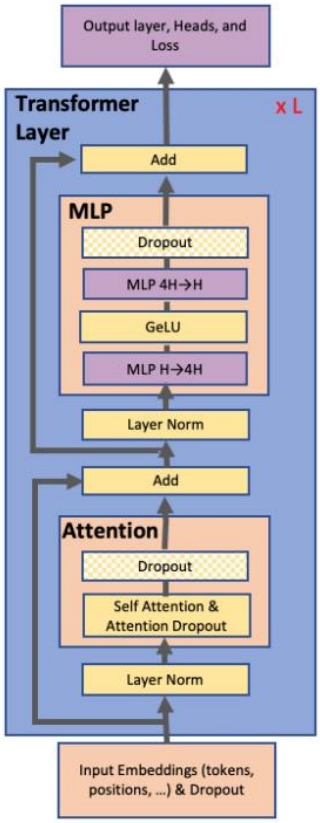

Рисунок 2.4 — Модифікована трансформерна архітектура

На рисунку 2.4 фіолетові блоки відповідають за повністю з’єднані шари, кожен блакитний блок представляє один шар трансформатора, який повторюється N разів.

Загалом, цей підхід є важливим внеском у область надшвидкого синтезу мовних моделей, оскільки він дозволяє навчати масивні мовні моделі більш ефективно та з кращою продуктивністю. Це може допомогти прискорити прогрес у обробці природної мови та створити нові програми для надшвидких мовних моделей у майбутньому

3. ПОСТАНОВКА ЗАДАЧІ

3.1 Науково-технічна задача

Магістерська кваліфікаційна робота має необхідність перед вирішенням списку задач. Одною із них є дослідження методів реалізації синтезу мови на прикладі штучного інтелекту, з обмеженою ресурсною базою. Також необхідно перевірити ефективність існуючих методів в залежності від їхньої архітектури. Важливою складовою роботи є проведення аналізу теоретичних відомостей та публікацій, які описують сучасний розвиток методів і технологій розпізнавання іменованих сутностей.

Метою цієї роботи є дослідження методів синтезу надшвидких структур мовних систем штучного інтелекту. До складу мети входить надання повного огляду останніх релевантних досягнень у сфері ШІ. Питання дослідження, на які прагне відповісти ця стаття, включають:

– які поточні обмеження НСМ та чому існує потреба в більш ефективних моделях мови?

– які різні методи синтезу НСМ штучного інтелекту та як вони порівнюються з точки зору ефективності та результативності?

– які потенційні реальні застосування надшвидких мовних моделей і як вони можуть покращити існуючі інструменти та системи обробки мови?

– які етичні та соціальні наслідки розробки надшвидких мовних моделей і як можна вирішити ці проблеми, щоб забезпечити відповідальний і етичний розвиток технології ШІ?

Щоб відповісти на ці дослідницькі запитання, у цій роботі розглядається відповідна література про мовні системи штучного інтелекту. У документі також аналізуються тематичні дослідження та застосування цих мовних моделей у реальному світі, щоб зрозуміти їхні потенційні переваги та обмеження. До експериментальної частини відноситься навчання та застосування різних технік оптимізації або розробки НСМ.

3.2 Предметна галузь дослідження

Під час проведення дослідження, головним предметом розробки застосунку, який буде базуватись, на основі боту або асистента, є генерація контенту – це ще одна сфера, де надшвидкісні мовні системи можуть бути корисними. Швидко й точно створюючи вміст, ці системи можна використовувати для автоматичного створення новинних статей, описів продуктів та інших типів вмісту.

Під час розробки застосунку, метою роботи буде реалізація генеративної нейронної мережі, яка буде здатна сприймати текстову інформацію і давати відповідні результати на основі неї. Під час цього буде проведено аналіз методів, метою яких є синтез мовних структур, дослідження їх швидкості, простоти або складності до підготовки, в тому числі й до навчання на можливих тестових вибірках.

Оскільки питання дослідження зосереджено на синтезі надшвидких структур мовних систем штучного інтелекту, має доцільність в оцінці НСМ на завданнях, пов’язаних з генерацією мови та її моделюванням.

3.3 Засоби проведення дослідження

На теперішній момент найбільш високу розповсюдженість серед методів у процесах синтезу мовних структур є нейронні мережі з застосуванням машинного навчання. Спираючись на це, реалізація буде спиратись на різновид генеративних мереж, а саме на підвид моделі BERT, яка базується на трансформерній архітектурі, використовує англійську мову та вперше була представлена у 2018 році, і з того часу займає місце умовного золотого стандарту для дослідників у цій галузі.

Словниковий запас моделі має розмір в 30 000 одиниць. Будь-яка лексема, якої немає в його цьому наборі, замінюється на невідоме значення. Також

розробка з проведенням попередньої підготовки одночасно щодо двох поставлених завдань:

– мовного моделювання - для прогнозування було відібрано 15% лексем, а метою навчання було спрогнозувати вибрану лексему з огляду на її контекст.

– передбачення наступного речення - маючи два проміжки тексту, модель передбачає, чи з’являться ці два проміжки послідовно в навчальному корпусі, та на основі навчання, враховує вагові коефіцієнти вірогідних варіантів продовження переданої послідовності.

У результаті виконаної підготовки, BERT вивчає приховані представлення слів і речень у контексті. Після оригінального навчання, модель можна налаштувати з меншими ресурсами на менших наборах даних, щоб оптимізувати продуктивність у конкретних завданнях, таких як завдання НЛП або класифікація тексту і завдання послідовного створення мови на основі послідовності (відповіді на запитання, відповіді у діалогах). Етап попереднього навчання значно дорожчий з точки зору обчислень, ніж тонке налаштування, тому як правило, варіації моделей після нього можна знайти у відкритому доступі.

Серед мов програмування, які могли бути використані для реалізації застосунку, було обрано Python. Його переваги над альтернативами, такими як JavaScript, C#, C++:

– простота освоєння, динамічна типізація та малі зусилля необхідні для початки розробки

– розповсюдженість серед наукової спільноти, не в останню чергу через свою простоту

– широке застосування для реалізації подібних до об’єкта дослідження проектів: ШІ, нейронні мережі, машинне навчання, робота з даними

– великий набір інструментів та фреймворків у відкритому доступі, високе місце в списку найбільш популярних мов у розробників

Відносно проведення експериментальних досліджень на конкретних моделях, було прийнято рішення використати набір інструментів від спільноти

Hugging Face, він надає велику кількість засобів з відкритим вихідним кодом для дослідження та розробки обробки природної мови, включаючи найсучасніші мовні моделі, такі як GPT-3, BERT, RoBERTa. До переліку причин, чому було використано саме цю бібліотеку входять наступні переваги:

– попередньо підготовлені моделі – наявний вибір з натренованих мовних моделей, які можна точно налаштувати під конкретні завдання, заощаджуючи час і зусилля при навчанні моделі з нуля, важливу роль цей пункт відіграє за умови обмежених ресурсів для повноцінної підготовки;

– простота використання - наявний зручний у використанні API для роботи з моделями, а також інтерфейс командного рядка для навчання власних моделей;

– вичерпна документація та підручники, що полегшує роботу з інструментами;

– підтримка спільноти - бібліотека має велике й активне співтовариство розробників і дослідників, які роблять внесок у розвиток і вдосконалення інструментів;

– інтеграція - інструменти були створені для легкого впровадження в існуючі робочі процеси та конвеєри з підтримкою багатьох мов програмування, фреймворків і платформ, відповідно це призводить до спрощення інтеграції набору засобів у існуюче дослідницьке чи виробниче середовище.

Через вищевказаний перелік переваг було прийнято рішення використати саме це рішення, але у середовищі розробників мовних моделей високою популярністю користуються також й інші інструменти, тому виконаємо короткий розгляд деяких з альтернатив:

– TensorFlow - це платформа машинного навчання з відкритим кодом, яка включає підтримку завдань обробки природньої мови, пропонує широкий спектр засобів для побудови та навчання мовних моделей, у тому числі попередньо навчених моделей, але ним може бути складніше користуватися, ніж інструментами Hugging Face;

– PyTorch - платформа машинного навчання з відкритим кодом, яка включає підтримку завдань, пов’язаних із синтезом мовних структур, цей засіб

має зручніший інтерфейс, ніж TensorFlow, і добре підходить для дослідників, які віддають перевагу більш гнучкій структурі;

– AllenNLP - бібліотека з відкритим кодом, створена на основі PyTorch, включає низку попередньо навчених моделей і інструментів для побудови та навчання з нуля, а також підтримку розширених функцій, таких як механізми концентрації уваги та багатозадачність навчання.

Зокрема, PyTorch також буде використано в якості допоміжного інструменту, так як він пропонує засоби низькорівневого контролю над моделями, які дозволяють більш точно проводити процедури навчання, які ще не підтримуються бібліотеками вищого рівня, включно з Hugging Face. Їх комбінація з метою покращити кінцеву результативність є розповсюдженим явищем під час розробки та дослідження НСМ.

4. ОПИС ТЕОРЕТИЧНИХ ДОСЛІДЖЕННЯ

4.1 Огляд методів оцінки мовних моделей

Оцінка продуктивності мовних моделей може використовувати різні методи, до прикладів найбільш розповсюджених технік та показників відносяться, перплексивність (англ. perplexity), точність, метрики BLUE та ROUGE, F-score.

Методи оцінки використовуються для визначення ефективності НСМ на етапі розробки. Ці техніки дозволяють розробникам оцінити якість моделі та встановити області для вдосконалення. Практичне застосування оцінювання залежить від конкретних цілей мовної моделі, що розробляється.

Одною з метрик для оцінки моделей є перплексивність (складність) – це міра того, наскільки добре розподіл імовірності або статистична модель прогнозує вибірку, використовувати для порівняння ймовірнісних моделей. Низька складність означає, що розподіл ймовірності добре передбачає вибірку. Воно визначається як середня негативна логарифмічна правдоподібність послідовності, зведена в ступінь. Якщо у нас є токенізована послідовність X = (x0, x1, … xt), тоді складність X розраховується за формулою:

$$\mathrm{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\}$$

Де, logpθ(xi| x<i) - це логарифм правдоподібності і-го маркера, залежно від попередніх маркерів x<i відповідно до нашої моделі. Важливим уточненням є те, що процедура токенізації безпосередньо впливає на складність моделі, що завжди слід враховувати під час порівняння різних моделей.

Іншим важливим показником є точність (ACC) – це відсоток правильних класифікацій, досягнутий навченою моделлю машинного навчання, тобто кількість правильних прогнозів, поділена на загальну кількість прогнозів у всіх

класах. Цей показник тісно пов’язаним з іншим, повнота (англ. recall) вимірює частку правильно передбачених позитивних випадків від усіх фактичних позитивних випадків (рис. 4.1).

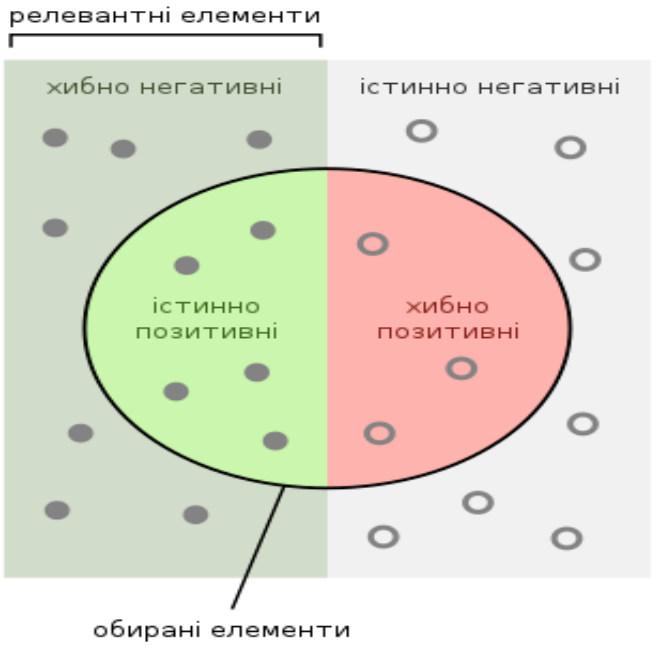

Рисунок 4.1 – Повнота та влучність на прикладі вибірки елементів

Іншими словами це кількість справжніх позитивних результатів, поділена на загальну кількість фактичних позитивних випадків, яка включає справжні позитивні та помилкові негативні результати. Точність дає відповідь на те як багато з обраних елементів є релевантними, повнота – як багато з релевантних елементів стають обраними (рис. 4.2).

Рисунок 4.2 – Порівняння повноти та влучності

Пов’язаною з двома попередніми значеннями є F-міра (F-score) - це одна з мір точності тесту. Обчислюється через влучність та повноту тесту, де влучність є числом правильно визначених позитивних результатів, поділеним на число всіх позитивних результатів, включно з визначеними неправильно, а повнота є числом правильно визначених позитивних результатів, поділеним на число всіх зразків, які повинно було бути визначено як позитивні.

Формула розрахунку F-міри, який використовує позитивний дійсний коефіцієнт β, де β вибрано таким чином, що відкликання вважається у β разів важливішим, ніж точність, має наступний вигляд:

$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}}.$$

BLEU — алгоритм для оцінки якості тексту, який було машинно перекладено з однієї природної мови на іншу. Якість вважається відповідністю між машинним і людським результатом. Чим ближче машинний переклад до професійного людського перекладу, тим він кращий – головна ідея BLEU. Алгоритм є одним з показників, який має високу кореляцію з людськими оцінками якості, і залишається розповсюдженим методом автоматизованої оцінки.

ROGUE – це набір показників, який використовується для оцінки якості текстових звітів шляхом їх порівняння з набором довідкових звітів. Він вимірює перекриття між зведеним машинним способом і еталонними підсумками з точки зору запам’ятовування, точності та оцінки F1. Оцінки ROUGE зазвичай повідомляються як ROUGE-1, ROUGE-2, ROUGE-L тощо, залежно від довжини n-грамів і/або найдовшої загальної довжини підпослідовності, яка використовується для порівняння.

Методи оцінювання використовуються протягом усього процесу розробки, щоб скерувати напрямок розробки та переконатися, що вона відповідає вимогам і цілям проекту. Оцінюючи продуктивність моделі через регулярні проміжки

часу, розробники мають можливість виявити проблеми на ранній стадії та внести необхідні коригування, щоб забезпечити відповідність моделі до поставлених перед проектом задач.

4.2 Методи розробки мовних моделей

Мета різних методів розробки мовних моделей полягає в тому, щоб покращити їх продуктивність у певному завданні, наприклад моделюванні мови, аналізі настроїв або машинному перекладі. Кінцевою метою є створення мовної моделі, яка може точно передбачити розподіл ймовірностей послідовності слів або створити зв’язний і осмислений текст.

Для досягнення цієї мети використовуються різні методи розробки. Наприклад, методи попередньої обробки даних, такі як токенізація та видалення стоп-слова, використовуються для зменшення кількості унікальних слів у корпусі та покращення здатності моделі узагальнювати нові дані. Методи розробки функцій можна використовувати для вибору та розробки відповідних функцій із текстових даних для покращення продуктивності моделі. Налаштування гіперпараметрів можна використовувати для вибору оптимального набору гіперпараметрів для моделі, тоді як методи регуляризації можуть використовуватися для запобігання переобладнанню та покращення здатності моделі узагальнювати нові дані.

Трансферне навчання може бути використане на попередньо підготовлених моделях для подібних завдань для покращення продуктивності НСМ. Ансамблеве навчання можна використовувати для поєднання кількох моделей для покращення продуктивності. Кінцевою метою всіх цих методів є створення мовної моделі, яка може точно передбачити розподіл ймовірностей послідовності слів або створити зв’язний і осмислений текст, який має відношення до конкретного завдання.

Крім цього, існує перелік методів, які можуть бути використані для синтезу надшвидких структур мовних систем ШІ. Ось деякі з них:

– паралелізм моделі —поділ великої моделі на кілька менших моделей, кожна з яких працює на окремому пристрої чи машині. Кожна менша модель може потім обробляти меншу частину даних, забезпечуючи швидшу обробку та зменшуючи вимоги до пам’яті системи.

– паралелізм даних – одна модель тренується на кількох пристроях, використовуючи різні частини набору даних одночасно. Це дозволяє істотно скоротити час, необхідний для навчання, і підвищити точність моделі;

– квантування – зниження точності вагових коефіцієнтів і активацій у моделі, таким чином зменшуючи вимоги до пам’яті та дозволяючи швидшу роботу;

– обрізка – техніка передбачає видалення непотрібних вантажів і з’єднань із моделі, зменшення розміру моделі та забезпечення швидшої обробки;

– дистиляція знань – ця техніка передбачає використання великої складної моделі (позначається учителем) для навчання меншої простішої моделі (учень) шляхом передачі знань, отриманих моделлю вчителя, моделі учня, як результат, це може призвести до значного зменшення розміру моделі та часу обробки при збереженні точності.

5. ОПИС ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

5.1 Навчання моделі

Для проведення експериментальних досліджень було обрано BERT, як основну модель. У порівнянні з іншими моделями, такими як GPT і Transformer-XL, BERT[12] розроблено для кодування двонаправленого контексту, що означає, що він враховує як попередні, так і наступні слова в реченні чи документі. Завдяки цьому BERT добре підходить для завдань, які потребують глибокого розуміння контексту тексту. Недоліком є те, що порівняно з GPT або Transformer-XL, модель гірше підходить для завдань, які вимагають генерувати текст, оскільки вони призначені для синтезу послідовностей слів.

Для цього дослідження, буде використано підтип моделі BERT – «bert-base-cased» який є меншою версією та містить 110 мільйонів параметрів, порівняно з оригінальними 340. Слово «cased» у назві перекладається як регістр або чутливий до регістру, тобто модель по-різному реагує на однакову інформацію, передану у різних комбінаціях малих та великих літер. Переваги вибору моделі на основі BERT:

– модель є попередньо-навченою, це означає, що для її роботи потрібне лише додаткове налаштування, за допомогою передачі даних, специфічних для вирішення поставленої задачі;

– широке застосування і високу популярність у спільноті дослідників, наявно багато ресурсів і попередньо натренованих моделей, які опубліковані у відкритому доступі;

– BERT – це високоточна мовна модель, при достатніх зусиллях при навчанні та попередньої підготовки, вона здатна досягати високої точності та продуктивності в метриках обробки природної мови.

Недоліки вибору BERT:

– модель є великою та дорогою за обчислювальними ресурсами, вона може бути повільною для навчання та важкою для розгортання в середовищах з обмеженими ресурсами;

– потреба у великій кількості даних для навчання, що може бути проблемою, якщо наявні обмеження доступу до даних;

– BERT є складною моделлю, що може ускладнити розуміння та інтерпретацію її прогнозів.

Порівнюючи bert-base-cased і BERT, обидві моделі є високоточними, кожен варіант можна точно налаштувати для широкого спектру завдань обробки природної мови. З точки зору продуктивності, BERT поступається через свій розмір і в 3 рази більшу кількість параметрів, але може бути більш точною через збільшені витрати на навчання. Тому, для виконання у практичній частині дослідження, з урахуванням обмежених розрахункових можливостей, було прийнято рішення використати bert-base-cased.

Як було зазначено раніше, обрана модель є попередньо навченою і публікується у відкритому доступі зі всіма параметрами. Вона базується на наборі даних англійської Вікіпедії та набору даних BookCorpus, який містить близько 800 мільйонів слів тексту з 11000 книг. Така модель була обрана через наступні чинники:

– широкий набір інформації, на базі якого наявна можливість до підготовки універсальної моделі;

– набір даних BookCorpus є загальнодоступним і публікується у відкритому вигляді.

Незважаючи на переваги, модель на основі цього набору даних має певні недоліки:

– можуть бути присутні упередження та неточності, через те, що дані взяті з джерел, які були створені людьми, зокрема, це проявляється через гендерну тенденційність, коли згенерований текст може схилятися до того, що певний набір професій притаманний одній статі;

– дані не є репрезентативними та можуть не бути і можуть не працювати добре на вузькій предметно-спеціальній галузі або зі спеціалізованою термінологією.

– набір даних наведено англійською, що обмежує застосовність моделі bert-base-cased до завдань лише однією мовою.

Як було вказано раніше, обрана модель є попередньо навченою, тому для проведення експериментів необхідно взяти набори даних для їх подальшої обробки. Одним із таких підходів є доробка або уточнення (англ. fine-tuning) моделі – процес, коли використовується спеціальний набір даних меншого розміру, ніж оригінальний, на якому виконувалось навчання. Як правило, він є специфічний для предметної галузі, і використовується для того щоб підготувати модель для конкретної задачі.

5.2 Перевірка моделі

Так як BERT не був розроблений для генерування тексту з нуля, а скоріше для попереднього навчання на великих обсягах текстових даних і тонкого налаштування конкретних завдань NLP, таких як класифікація тексту, відповідей на запитання та розпізнавання іменованих сутностей.

Коли справа доходить до синтезу тексту, модель можна використовувати для генерування тексту, даючи моделі певне підґрунтя у вигляді підказки або початкової послідовності слів, вже після цього є доцільним використання моделі для створення додаткового тексту на основі початкової підказки або інформації.

Якість створеного тексту залежить від низки факторів, включаючи розмір і якість навчальних даних, які використовуються для попереднього навчання моделі, конкретні підказки, надані моделі, і складність мови та синтаксису, що використовуються в створений текст.

Хоча BERT здатний генерувати текст, він не завжди може створювати зв’язні чи граматично правильні речення, а згенерований текст може потребувати значної постобробки, щоб бути придатним для використання.

Розглянемо підхід, при використанні маскованої строки, передана підказка має такий вигляд: "The capital of France, [MASK] contains the Eiffel Tower.". У

цьому випадку, символ маски позначений словом mask у дужках. Він може відрізнятися у різних моделях і повинен зчитуватись з відповідних налаштувань.

Виконавши передачу підказки до моделі та згенерувавши 10 результатів з найбільшими ваговими коефіцієнтами, які відповідають за ймовірність з якою модель вирішує доцільність застосування певного токену, отримуємо наступний результат, відсортований у порядку від найбільшого значення до найменшого:

- The capital of France, Paris, contains the Eiffel Tower.
- The capital of France, Lyon, contains the Eiffel Tower.
- The capital of France, Strasbourg, contains the Eiffel Tower.
- The capital of France, Versailles, contains the Eiffel Tower.
- The capital of France, Toulouse, contains the Eiffel Tower.
- The capital of France, Brussels, contains the Eiffel Tower.
- The capital of France, Metz, contains the Eiffel Tower.
- The capital of France, Bordeaux, contains the Eiffel Tower.
- The capital of France, Lille, contains the Eiffel Tower.
- The capital of France, Orléans, contains the Eiffel Tower.

Наведемо частину вихідного коду, який був використаний для доповнення тексту, у ньому застосовуються моделі з бібліотеки transformers та допоміжні засоби torch. Приклад:

```
# Кількість результатів
limit = 10
# Підказка з місцем для згенерованого тексту на місці спеціального
# токену mask_token
text = "The capital of France, " + tokenizer.mask_token + ",
contains the Eiffel Tower."
input = tokenizer.encode_plus(text, return_tensors = "pt")
mask_index = torch.where(input["input_ids"][0] ==
tokenizer.mask_token_id)
output = model(**input)
logits = output.logits
softmax = F.softmax(logits, dim = -1)
mask_word = softmax[0, mask_index, :]
```

```
# Розрахунок найбільш вірогідних результатів
top = torch.topk(mask_word, limit, dim = 1)[1][0]
for token in top:
   word = tokenizer.decode([token])
   generated = text.replace(tokenizer.mask_token, word)
   print(generated)
```

5.3 Оцінка моделі

Також виконаємо оцінку моделі за допомогою згаданих під час етапу теоретичних досліджень методів. Попередньо підготовлена модель залишається незмінною – bert-base-cased, однак буде використано інший набір даних – glue (General Language Understanding Evaluation). Всього цей датасет розділений на 9 частин (табл. 5.1), різні частини мають окреме призначення під час проведення тестування моделі

Таблиця 5.1 Структура BLUE

Назва фрагменту	Скорочене позначення	Призначення
The Corpus of Linguistic Acceptability	CoLA	КММ
The Stanford Sentiment Treebank	SST-2	Точність
Microsoft Research Paraphrase Corpus	MRPC	Точність, F1
Semantic Textual Similarity Benchmark	STS-B	КММ
Quora Question Pairs	QQP	Точність, F1
MultiNLI Matched	MNLI	ККМ
Question NLI	QNLI	Точність
Recognizing Textual Entailment	RTE	Точність

Winograd NLI	WNLI	Точність

У контексті цієї таблиці, є доцільним внести пояснення щодо коефіцієнту кореляції Метьюза (англ. Brian W. Matthews) – це статистичний показник, який використовується для оцінки ефективності моделей бінарної класифікації, включно з тими, які використовуються в задачах обробки природної мови. Він враховує істинні позитивні, хибно-позитивні, істинні негативні та хибно-негативні результати та забезпечує міру якості прогнозів моделі бінарної класифікації.

ККМ приймає значення від -1 до 1, де оцінка -1 вказує на повну розбіжність між прогнозами моделі та справжніми мітками, 0 вказує на те, що модель працює не краще, ніж випадкова, а 1 вказує на повну збіг між прогнози моделі та справжні мітки.

Через обмеження в обчислювальних можливостях, було прийнято рішення провести тести на предмет класифікації текстів з наступними фрагментами вищезгаданого набору:

– MRPC
– RTE
– WNLI

Отримуємо наступні результати:

Таблиця 5.2 Оцінка точності текстової класифікації

Фрагмент	Результати (точність, F1)	Витрачений час
MRPC	85.05, 89.54	01:47
RTE	66.43	2:06
WNLI	38.03	0:34

Ми отримуємо наведені у табл. 5.2 результати в наборі тесту для розробників із попередніми командами. Навчання виконується за даними,

організованими з допомогою ключа (англ. seed), тому при використанні однакової версії бібліотеки PyTorch (2.0.0 на момент роботи над дослідженням), результати будуть співпадати. Складнішою метрикою є час навчання, так як він залежить від графічного процесора і може значно відрізнятись як в гіршу, так і кращу сторони, вищевказані результати отримано з використанням 3060TI.

Також, як зазначають автори GLUE, для WNLI наявна різниця в розподілі для навчальної і тестової вибірки, через це можливі розбіжності між оцінками для цієї категорії. Поділ на навчання/розробку для WNLI є правильним, але виявляється дещо суперечливим: коли два приклади містять одне й те саме речення, це зазвичай означає, що вони матимуть протилежні мітки. Розподіл тренування та розробника може мати спільні речення, тому, якщо модель переобладнала навчальний набір, це може стати гіршим, ніж випадкова точність на WNLI на наборі розробника. Крім того, тестовий набір має інший розподіл міток, ніж набори тренувань і розробників.

До порівняння, наведемо приклад результатів для аналогічної моделі distilbert-base-uncased (відрізняється від bert-base-cased тим, що має менший розмір і, як виходить з назви, не чутлива до регістру тексту) взятий з таблиці результатів для оцінки за допомогою GLUE. Дані не мають інформацію про витрачений час, але припускаючи, що ця модель має меншу кількість параметрів, можна зробити висновок, що він буде менше, ніж в попередній таблиці.

Таблиця 5.3 Точність моделі distilbert-base-uncased

Фрагмент	Результати (точність, F1)	Витрачений час
MRPC	87.6, 83.1	n/a
RTE	54.1	n/a
WNLI	65.1	n/a

5.4 Оптимізація моделі

Під час виконання навчання було використано ряд способів для оптимізації моделі. Одним із них є деградація швидкості навчання. Суть цього підходу полягає в тому, що відбувається тренування мережі з високою швидкістю, а потім повільно зменшується/затухає до досягнення локальних мінімумів. Було емпірично доведено, що це допомагає як для оптимізації, так і узагальнення результату.

Розглянемо візуалізацію цього підходу, на рис. 5.1 синім кольором зображено процес навчання з постійною швидкістю. Кроки, які виконуються під час ітерацій до мінімумів, настільки шумні, що після певних повторень здається, що він блукає навколо мінімумів і насправді не збігається.

В той же час, підхід зі зменшенням швидкості навчання (позначено зеленою лінією), в кінцевому підсумку коливається в більш тісному регіон навколо мінімумів, а не відходить далеко від нього. Так як початкова швидкість навчання є великою, ми все ще маємо відносно високу швидкість, а через наближення до мінімальної швидкості навчання, кількість менш точних результатів, відповідно, також стає меншим.

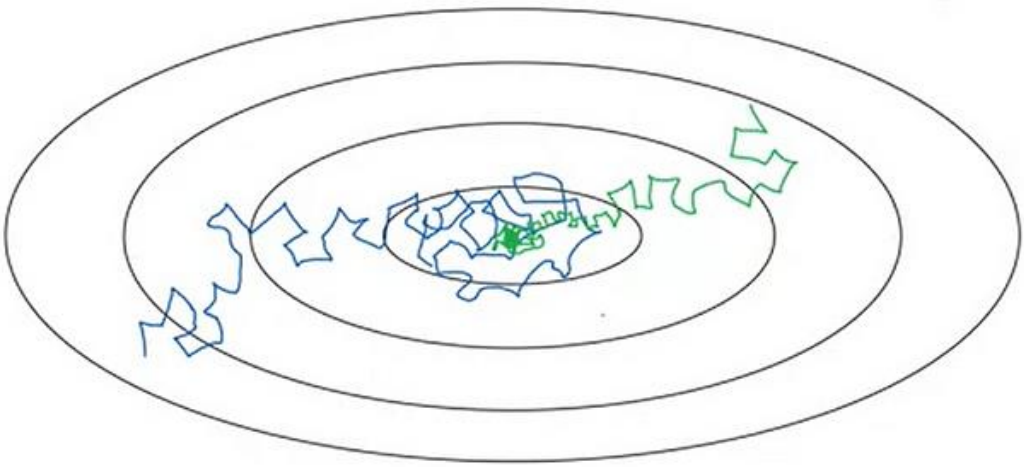

Рисунок 5.1 – Порівняння сталого (синій) та регресивного (зелений) темпу навчання

Так як бібліотека для тренування моделі за замовчування використовує лінійну формулу розрахунку швидкості навчання, було використано значення 2e-5 (зміна в 0.00002 після кожної ітерації)

Другим важливим методом було корегування епохи(epoch) моделі – гіперпараметра, який відповідає за те, скільки разів буде виконано тренування. Для того, щоб не перенасичувати мережу, важливо не ставити це значення занадто високим, тому для фрагментів RTE та WNLI, які є порівняно малими, було використано більше значення – 5, для MPRC – 3.

Наглядно порівняти результати можна, виконавши інверсію параметру епохи під час навчання з метою перевірити їх вплив на точність та час навчання моделі. Відповідно, для RTE та WNLI було застосовано значення, яке дорівнює 3, для більшої за розміром MPRC – 5. Отримані результати наведемо у таблиці 5.4.

Таблиця 5.4 Оцінка точності з інвертованими параметрами

Фрагмент	Результати (точність, F1)	Витрачений час
MRPC	85.05, 89.54	3:48
RTE	65.70	1:08
WNLI	33.8	0:20

Порівнюючи з попередніми значеннями (табл. 5.2), можемо побачити що точність та F-міра для MRPC не змінилися, проте в півтора рази зріс час навчання, приблизно на 2 хвилини, або в два рази. В інших випадках, для менших наборів RTE та WNLI , відбулося незначне зменшення витраченого часу, але також впала і результативність, на 1 та 5 відсотків відповідно.

Обґрунтування таких значень полягає в тому, що менші моделі мають менше параметрів, а це означає, що їм може знадобитися більше часу для навчання, щоб знайти оптимальне рішення. Крім того, менші моделі можуть бути

більш сприйнятливими до переобладнання, і більш тривале навчання може допомогти пом’якшити цю проблему.

Іншим використаним гіперпараметром є максимальна довжина послідовності навчання. Вона відноситься відноситься до обмежень на розмір послідовностей, які може прийняти модель. Зокрема, це гранична кількість токенів (слів або підслів), які вхідний текст може містити після токенізації. Якщо вхідна послідовність довша за це значення, вона буде скорочена або розбита на менші послідовності.

Через те, що очікувались порівняно не складні та короткі послідовності, таке значення було встановлено як 128. Вибір цього параметру важливий, оскільки він може вплинути на здатність моделі фіксувати відповідну інформацію у вхідному тексті.

Більша довжина може дозволити моделі обробляти довші вхідні послідовності, потенційно захоплюючи більше контексту та покращуючи продуктивність. Однак це також вимагає більше пам’яті та часу на обробку, а також може призвести до більш тривалого навчання та повільнішого висновку. І навпаки, менша довжина сприяє швидшому навчанню та виведенню, але може призвести до втрати важливої інформації.

Відносно оптимізації взаємодії процесу навчання на GPU, також було застосовано обмеження до кількості навчальних прикладів, які паралельно обробляються на пристрої під час навчання. Зокрема, це число прикладів, які завантажуються в пам’ять пристрою за один раз і обробляються за один прохід вперед і назад під час кожної ітерації навчання.

Через відносну актуальність відеокарти, на якій проводилось навчання (3060ti), цей гіперпараметр було встановлено у розмірі 32 одиниць і було прийнято рішення зменшити до 16 у випадку нестачі ресурсів або помилок під час навчання. Так як попереднє тестування пройшло успішно, кількість одночасних прикладів було залишена без змін.

Проведемо експериментальне дослідження та порівняємо попереднє значення – 32 з тим, яке потенційно має місце в застосуванні при більш

обмежених умовах – 16 паралельних значень, результат порівняння розглянемо у таблиці 5.5. Також зазначимо, що інші параметри, включаючи кількість епох залишається без змін і відповідає значенням представленим у таблиці 5.2 (3 для MRPC, 5 для RTE та WNLI).

Таблиця 5.5 – Порівняння швидкості навчання до максимальної кількості одночасно оброблюваних прикладів

Фрагмент	Попередні результати	Витрачений час	Зміна
MRPC	01:47	2:09	20%
RTE	1:08	1:25	25%
WNLI	0:20	0:35	70%

Експериментальним способом можемо підтвердити спад у швидкості навчання. Точність та F-міра не були вказані, так як зміни в налаштуваннях не відобразились на них, тому інформація з таблиці 5.2 залишається актуальною для нових результатів.

Варто зазначити, що одночасна кількість навчальних прикладів повинна налаштуватись на основі іншого згаданого вище параметру – максимальної довжини вхідної послідовності. Якщо послідовність має великий розмір, тренування з великою кількістю паралельних прикладів повинна бути зменшена, так як це може призвести до нестачі доступної пам’яті GPU.

Підсумовуючи, використання цих двох гіперпараметрів визначає певне компромісне рішення між тим, як швидко буде готуватися модель і наскільки точні результати вона зможе давати після цього. Найкращим підходом є проведення експериментів з різними комбінаціями для того щоб використати ресурси відеокарти у повному обсязі.

6. МОЖЛИВОСТІ ВПРОВАДЖЕННЯ У НАУКОВІЙ ТА ПРАКТИЧНІЙ ДІЯЛЬНОСТІ

Результати, отримані під час проведення дослідження, спираються на актуальну інформацію про НСМ, зокрема на варіанти їх реалізацій на базі генеративних нейронних мереж.

Важливим є уточнення про те, що сфера штучного інтелекту та обробки природної мови швидко розвивається, з’являються нові моделі та методики розробляються та регулярно публікуються. Особливого розвитку ця галузь отримала після успіху окремих моделей у 2022 році.

Практичне впровадження отриманих результатів є можливим для наступних варіантів, до складу яких входять:

– навчання мовних моделей;

– підготовка попередньо натренованих моделей під специфічну предметну галузь або конкретні задачі;

– оптимізація процесів розробки та впровадження в умовах обмежених ресурсів;

– оцінка точності моделей;

Дослідження може мати розвиток у майбутньому через розвиток у сфері та через появу різних варіацій мовних моделей, їх підходів до підготовки, донавчання та оцінки. Потенційним напрямком, який можна використати для роботи над подальшим аналізом є розробка моделей на інших мовах, які не мають такої популярності як англійська, яка займає місце золотого стандарту на даний момент, незважаючи на те, що у користувачів наявний попит на локалізовані реалізації мовних моделей, врахування культурних або географічних особливостей тощо.

ВИСНОВОК

Під час проведення дослідження на тему методів синтезу надшвидкодіючих структур мовних систем штучного інтелекту, було розглянуто актуальну інформацію про побудову моделей, націлених на вирішення поставлених проблем, до складу яких входить синтез та класифікація текстових послідовностей

Було розглянуто наукові роботи, які присвячені різного виду генеративним мережам, підходам до оптимізації, навчання та розробки. Через широке застосування моделей на базі трансформерної архітектури та їх актуальність, зокрема через поштовх до розвитку галузі від представників цієї категорії в останні роки, було прийнято рішення акцентувати увагу на цій категорії, зокрема провести експериментальну частину дослідження з одним із типів такої моделі – BERT.

Теоретична складова містила в собі аналіз фундаментальної інформації про методи розробки, оцінку та оптимізацію мовних моделей, найбільш розповсюджені метрики та розрахунки, за якими вони проводяться, та які проблеми вирішують

Частина практичного застосування отриманої дослідження включає в себе застосування отриманої інформації на прикладі підвиду мовної моделі BERT. Було проаналізовано результативність на прикладі точності класифікації тексту, з урахуванням раніше розглянутих метрик, швидкості навчання та підходів для оптимізації під час неї зі збереженням продуктивності.

Загалом, дослідження підкреслює важливість вибору відповідної мовної моделі та набору даних для певного завдання, а також ефективність тонкого налаштування попередньо навчених моделей для досягнення кращої продуктивності. Досягнення в області обробки природної мови є багатообіцяючими, і дають очікування подальшого прогресу в майбутньому.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Четвериков, Г. Г. Алгебрологічні та лексикографічні аспекти моделювання природної мови. Харків – 2014

2. М. Ф. Бондаренко, І. А. Ревенчук, Г. Г. Четвериков. Синтез швидкодіючих багатозначних структур мовних систем штучного інтелекту. Київ, 19-23 жовтня – 1998

3. Alex Graves. Generating sequences with recurrent neural networks. London, June – 2014

4. Ashish Vaswani, Noam Shazeer, Niki Parmar. Attention is all you need. Advances in neural information processing systems – 2017.

5. Deng Cai, Elman Mansimov, Yi-An Lai, Yixuan Su, Lei Shu, Yi Zhang. Measuring and Reducing Model Update Regression in Structured Prediction for NLP. February – 2022

6. Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, Jianfeng Gao. Few-shot Natural Language Generation for Task-Oriented Dialog. November – 2020.

7. Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Language Models are Few-Shot Learners. July – 2020.

8. Wasi Ahmad, Jianfeng Chi, Tu Le, Thomas Norton, Yuan Tian, Kai-Wei Chang. Intent Classification and Slot Filling for Privacy Policies. August – 2021

9. Philipp Schmid. Few-shot learning in practice: GPT-Neo and the HuggingFace Accelerated Inference API, June 2021. URL: https://huggingface.co/blog/few-shot-learning-gpt-neo-and-inference-api (дата звернення: 28.04.2023)

10. Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. July – 2019.

11. Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, Bryan Catanzaro. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. March – 2020.

12. E. Erdem, M. Kuyu, S. Yagcioglu, A. Frank, L. Parcalabescu, B. Plank, A. Babii, O. Turuta, A. Erdem, I. Calixto, E. Lloret, E.-S. Apostol, C.-O. Truica, B. Sandrih, A. Gatt, S. Martincic-Ipsic, G. Berend, and G. Korvel. 2022. Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning. Journal of Artificial Intelligence Research, in press.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

1. Четвериков, Г. Г. Алгебрологічні та лексикографічні аспекти моделювання природної мови. Харків – 2014

2. М. Ф. Бондаренко, І. А. Ревенчук, Г. Г. Четвериков. Синтез швидкодіючих багатозначних структур мовних систем штучного інтелекту. Київ, 19-23 жовтня – 1998

12. E. Erdem, M. Kuyu, S. Yagcioglu, A. Frank, L. Parcalabescu, B. Plank, A. Babii, O. Turuta, A. Erdem, I. Calixto, E. Lloret, E.-S. Apostol, C.-O. Truica, B. Sandrih, A. Gatt, S. Martincic-Ipsic, G. Berend, and G. Korvel. 2022. Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning. Journal of Artificial Intelligence Research, in press.