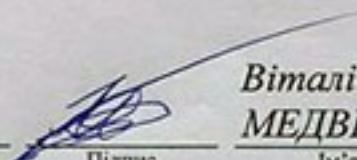
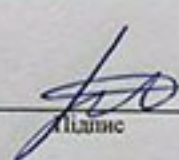


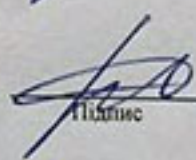
КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему Метод генерації відповідей з доповнюючим інформаційним пошуком
для допоміжної комунікації


Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 2 курсу, група КНм-23-1
Курс, група виконавця  Віталій
МЕДВЕДЧУК
Підпис Ім'я, ПРІЗВИЩЕ

Керівник: к.т.н., доцент кафедри КН
Науковий ступінь, посада  Руслан
БАГРІЙ
Підпис Ім'я, ПРІЗВИЩЕ

Нормоконтроль: к.т.н., доцент кафедри КН
Науковий ступінь, посада  Руслан
БАГРІЙ
Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор
Підпис  Олександр
БАРМАК
Підпис Ім'я, ПРІЗВИЩЕ

16 грудня 2024 р.

Факультет інформаційних технологій

Кафедра комп'ютерних наук

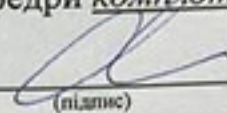
Освітній ступінь магістр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук



(підпис)

д.т.н., професор Олександр БАРМАК

« 01 » вересня 2024 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

1. Тема кваліфікаційної роботи магістра: «Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації»

2. Завдання видано студенту Віталію МЕДВЕДЧУКУ

(Ім'я, ПРІЗВИЩЕ)

3. Керівник роботи доцент кафедри КН Руслан БАГРІЙ

(Ім'я, ПРІЗВИЩЕ)

4. Затверджені наказом університету від «26» 08 2024 р. № 60

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

Мета кваліфікаційної роботи магістра – покращення комунікації для людей з обмеженими можливостями мовлення за допомогою генерації відповідей з доповнюючим інформаційним пошуком. Для досягнення поставленої мети визначено наступні задачі: провести огляд сучасних підходів до генерації відповідей з використанням мовних моделей та технологій доповнюючого інформаційного пошуку (RAG); розробити метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації; підготувати релевантні текстові дані для покращення точності генерації відповідей з доповнюючим інформаційним пошуком; розробити програмну реалізацію системи комунікації між людьми з використанням генерації відповідей з доповнюючим інформаційним пошуком; провести тестування взаємодії системи комунікації з користувачами та проаналізувати коректність його відповідей

Реферат

Кваліфікаційна робота магістра присвячена розробці методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації.

Актуальність теми. Сучасне суспільство дедалі більше орієнтується на технологічні засоби для подолання бар'єрів комунікації, особливо для людей з вадами слуху та мови. Ці групи населення часто стикаються з труднощами у спілкуванні через обмежений доступ до зрозумілих та ефективних методів передачі інформації. Традиційні методи комунікації не завжди забезпечують швидке та гнучке надання необхідних відповідей у різних соціальних контекстах, що може призводити до непорозумінь та соціальної ізоляції.

Метод генерації відповідей з доповнюючим інформаційним пошуком є актуальним, оскільки він пропонує технологічний підхід для підвищення якості та швидкості комунікації. Використання інформаційного пошуку дозволяє системі надавати більш точні та змістовні відповіді, збагачені актуальними даними з різних джерел. Це особливо важливо для ситуацій, коли необхідно швидко знайти або уточнити інформацію під час діалогу.

Крім того, цей метод підвищує автономність людей з вадами слуху та мовлення, дозволяючи їм ефективно брати участь у соціальній, професійній та освітній діяльності. Впровадження такого підходу може значно полегшити інтеграцію цих людей у суспільство, зменшити їх залежність від сторонньої допомоги та підвищити загальний рівень інклюзивності.

Таким чином, розробка методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації є перспективною темою дослідження, яка може зробити значний внесок у покращення якості життя та комунікаційних можливостей для людей з обмеженими можливостями.

Мета і задачі роботи. Метою кваліфікаційної роботи магістра є покращення комунікації для людей з обмеженими можливостями мовлення за допомогою генерації відповідей з доповнюючим інформаційним пошуком.

Для досягнення поставленої мети визначено наступні задачі:

- провести огляд сучасних підходів до генерації відповідей з використанням мовних моделей та технологій доповнюючого інформаційного пошуку (RAG);
- розробити метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації;
- підготувати релевантні текстові дані для покращення точності генерації відповідей з доповнюючим інформаційним пошуком;
- розробити програмну реалізацію системи комунікації між людьми з використанням генерації відповідей з доповнюючим інформаційним пошуком;
- провести тестування взаємодії системи комунікації з користувачами та проаналізувати коректність його відповідей.

Об’єкт дослідження – процес генерації текстових відповідей з доповнюючим інформаційним пошуком.

Предмет дослідження – методи збору та аналізу інформації, великі мовні моделі та векторні бази даних для систем комунікації, орієнтованих на людей з обмеженими можливостями мовлення.

Методи дослідження, застосовані для вирішення поставлених завдань: для генерації відповідей – використовується метод генерації тексту за допомогою великих мовних моделей; для доповнюючого інформаційного пошуку – застосовується метод векторного пошуку в базі даних з використанням алгоритмів подібності; для інтеграції мовної моделі з інформаційним пошуком – застосовується Retrieval-Augmented Generation, який поєднує генерацію тексту із результатами пошуку у базі знань для забезпечення більш точних та інформативних відповідей.

Наукова новизна одержаних результатів. В результаті проведеної роботи удосконалено метод генерації відповідей для допоміжної комунікації за рахунок використання генеративної мовної моделі з інформаційним пошуком з урахуванням зовнішнього контексту, що дало можливість покращити комунікації для людей з обмеженими можливостями мовлення завдяки генерації релевантних варіантів відповідей.

Апробація результатів кваліфікаційної роботи магістра та публікації.

Основні положення і результати роботи опубліковані в збірнику наукових праць – Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації / Медведчук В.Ю., Багрій Р.О., Скрипник Т.К. // Збірник наукових праць за матеріалами Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук – 2024» Хмельницький, 2024.

Також, результати роботи опубліковані у науковому журналі – Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації / Медведчук В.Ю., Багрій Р.О., Скрипник Т.К., Мазурець О.В., Монастирська Д.С. // Науковий журнал «Вісник Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2024, №6 (Довідка з редакції).

Структура та обсяг роботи. Кваліфікаційна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 41 найменування та 4 додатків. Загальний обсяг кваліфікаційної роботи магістра становить 83 сторінки, з них 73 сторінки основного тексту та 36 сторінок додатків. У роботі наведено 33 рисунка.

Ключові слова: допоміжна комунікація, генерація відповідей, доповнюючий інформаційний пошук, великі мовні моделі, Retrieval-Augmented Generation, пошукові алгоритми, текстові відповіді, контекст.

Зміст

Перелік скорочень	4
Вступ	5
Розділ 1 Дослідження методів генерації відповідей для забезпечення допоміжної комунікації	8
1.1 Аналіз предметної області.....	8
1.2 Огляд технології великих мовних моделей та технології RAG	12
1.3 Огляд існуючого програмного забезпечення	19
1.4 Постановка задачі.....	23
Розділ 2 Розробка методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації.....	24
2.1 Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації	24
2.2 Архітектура методу генерації тексту з інформаційним пошуком	25
2.2.1 Індексція даних.....	26
2.2.2 Пошук інформації	28
2.2.3 Генерація відповіді	29
2.3 Метод векторного вбудовування.....	31
2.4 Налаштування технології RAG для генерації відповідей.....	36
Висновки до розділу 2	39
Розділ 3 Реалізація методу генерації відповідей для допоміжної комунікації.....	41
3.1 Опис платформи, технологій та бібліотек.....	41
3.2 Опис компонентів системи комунікації.....	45
3.3 Опис побутових ситуацій та формування датасету.....	52
Висновки до розділу 3	54
Розділ 4 Дослідження методу пошуково-доповненої генерації варіантів відповідей	55
4.1 Опис застосунку для дослідження системи комунікації.....	55
4.2 Тестування системи комунікації з використанням генерації відповідей з доповнюючим інформаційним пошуком.....	59

4.3 Визначення ефективності системи комунікації.....	73
Висновки до розділу 4	76
Загальні висновки.....	78
Перелік посилань.....	80
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
ВООЗ	Всесвітня організація охорони здоров'я
AAC	Augmentative and Alternative Communication
ШІ	Штучний інтелект
DL	Deep Learning
RL	Reinforcement Learning
LLM	Large Language Model
RAG	Retrieval-Augmented Generation
API	Application Programming Interface
GPT	Generative Pre-trained Transformer
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
AI	Artificial Intelligence
GenAI	Generative Artificial Intelligence
NLP	Natural Language Processing
RLHF	Reinforcement Learning from Human Feedback

Вступ

Актуальність теми. Сучасне суспільство дедалі більше орієнтується на технологічні засоби для подолання бар'єрів комунікації, особливо для людей з вадами слуху та мови. Ці групи населення часто стикаються з труднощами у спілкуванні через обмежений доступ до зрозумілих та ефективних методів передачі інформації. Традиційні методи комунікації не завжди забезпечують швидке та гнучке надання необхідних відповідей у різних соціальних контекстах, що може призводити до непорозумінь та соціальної ізоляції.

Метод генерації відповідей з доповнюючим інформаційним пошуком є актуальним, оскільки він пропонує технологічний підхід для підвищення якості та швидкості комунікації. Використання інформаційного пошуку дозволяє системі надавати більш точні та змістовні відповіді, збагачені актуальними даними з різних джерел. Це особливо важливо для ситуацій, коли необхідно швидко знайти або уточнити інформацію під час діалогу.

Крім того, цей метод підвищує автономність людей з вадами слуху та мовлення, дозволяючи їм ефективно брати участь у соціальній, професійній та освітній діяльності. Впровадження такого підходу може значно полегшити інтеграцію цих людей у суспільство, зменшити їх залежність від сторонньої допомоги та підвищити загальний рівень інклюзивності.

Таким чином, розробка методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації є перспективною темою дослідження, яка може зробити значний внесок у покращення якості життя та комунікаційних можливостей для людей з обмеженими можливостями.

Мета і задачі роботи. Метою кваліфікаційної роботи магістра є покращення комунікації для людей з обмеженими можливостями мовлення за допомогою генерації відповідей з доповнюючим інформаційним пошуком.

Для досягнення поставленої мети визначено наступні задачі:

- провести огляд сучасних підходів до генерації відповідей з використанням мовних моделей та технологій доповнюючого інформаційного пошуку (RAG);
- розробити метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації;
- підготувати релевантні текстові дані для покращення точності генерації відповідей з доповнюючим інформаційним пошуком;
- розробити програмну реалізацію системи комунікації між людьми з використанням генерації відповідей з доповнюючим інформаційним пошуком;
- провести тестування взаємодії системи комунікації з користувачами та проаналізувати коректність його відповідей.

Об’єкт дослідження – процес генерації текстових відповідей з доповнюючим інформаційним пошуком.

Предмет дослідження – методи збору та аналізу інформації, великі мовні моделі та векторні бази даних для систем комунікації, орієнтованих на людей з обмеженими можливостями мовлення.

Методи дослідження, застосовані для вирішення поставлених завдань: для генерації відповідей – використовується метод генерації тексту за допомогою великих мовних моделей; для доповнюючого інформаційного пошуку – застосовується метод векторного пошуку в базі даних з використанням алгоритмів подібності; для інтеграції мовної моделі з інформаційним пошуком – застосовується Retrieval-Augmented Generation, який поєднує генерацію тексту із результатами пошуку у базі знань для забезпечення більш точних та інформативних відповідей.

Наукова новизна одержаних результатів. В результаті проведеної роботи удосконалено метод генерації відповідей для допоміжної комунікації за рахунок використання генеративної мовної моделі з інформаційним пошуком з урахуванням зовнішнього контексту, що дало можливість покращити комунікації для людей з обмеженими можливостями мовлення завдяки генерації релевантних варіантів відповідей.

Апробація результатів кваліфікаційної роботи магістра та публікації.

Основні положення і результати роботи опубліковані в збірнику наукових праць – Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації / Медведчук В.Ю., Багрій Р.О., Скрипник Т.К. // Збірник наукових праць за матеріалами Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук – 2024» Хмельницький, 2024.

Також, результати роботи опубліковані у науковому журналі – Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації / Медведчук В.Ю., Багрій Р.О., Скрипник Т.К., Мазурець О.В., Монастирська Д.С. // Науковий журнал «Вісник Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2024, №6 (Довідка з редакції).

Структура та обсяг роботи. Кваліфікаційна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 41 найменування та 4 додатків. Загальний обсяг кваліфікаційної роботи магістра становить 83 сторінки, з них 73 сторінки основного тексту та 36 сторінок додатків. У роботі наведено 33 рисунка.

Розділ 1 Дослідження методів генерації відповідей для забезпечення допоміжної комунікації

1.1 Аналіз предметної області

Спілкування є необхідністю для кожної людини, але деякі мають вроджені чи набуті вади, через які не можуть виразити свої думки та відчуття. Згідно з оцінками ВООЗ, близько 430 мільйонів людей у світі мають значні порушення слуху чи мови. Більшість з них вивчають мову жестів, але жести можуть варіюватися в залежності від країни. Жестова мова – це спеціальний вид письма, що дозволяє позначати слова та літери, але її не знає більшість людей [1].

Порушення слуху можуть бути частковими чи повними, виникати через інфекції, шум, медикаменти чи травми. Глухота буває вродженою чи набутою, а порушення слуху можуть бути кондуктивними (зміна рухливості барабанної перетинки) чи нейросенсорними (пошкодження внутрішнього вуха чи слухового нерва). Важливими причинами є старіння, інфекційні хвороби та надмірний шум [2].

Тим часом порушення мови, як-от алалія, афазія, глухонімота та мутизм, часто не класифікуються, а людей називають «німими». Алалія – це відсутність або недорозвиток мовлення, афазія – порушення через ураження мозку, глухонімота – вроджена або набута втрата слуху з відсутністю мовлення, а мутизм – відсутність відповіді на запитання при збереженні здатності до розуміння мови [3].

Зважаючи на велику кількість людей з вадами слуху та мови, суспільство повинно сприймати їх як рівних. Для полегшення їхнього життя розроблено інноваційні технології, такі як слухові апарати, субтитрування, програми для альтернативної комунікації, смарт-годинники та відеодзвінки з перекладом на жестову мову. Ці технології сприяють соціалізації та інтеграції людей з вадами, знижуючи бар'єри у повсякденному житті.

Однак, наявність таких порушень слуху та мови не повинна ставати бар'єром для повноцінної участі в суспільному житті. Оскільки традиційні методи комунікації можуть бути обмеженими для багатьох людей з вадами, з'являється потреба у

використанні альтернативних способів вираження думок та почуттів. Одним із таких способів є доповнювальна та альтернативна комунікація (ААС), яка відкриває нові можливості для тих, хто має серйозні порушення мовлення, дозволяючи їм знову знаходити шлях до взаєморозуміння та соціалізації.

Підсилювальна та альтернативна комунікація (ААС) відіграє вирішальну роль у покращенні комунікації для людей із серйозними порушеннями мовлення. Для багатьох людей здатність спілкуватися є фундаментальною для встановлення зв'язків з іншими та здійснення повсякденної діяльності. Методи ААС можуть бути трансформаційними, дозволяючи людям перейти від ізоляції та залежності до активної участі в соціальному та сімейному контексті, таким чином сприяючи стосункам і спільному розумінню [4].

З розвитком технологій та потребою в ще більш ефективних способах комунікації, нейронні підходи стали важливим інструментом для удосконалення процесів взаємодії. Якщо традиційні методи ААС вже забезпечують значну підтримку для людей із серйозними порушеннями мовлення, то застосування глибоких нейронних мереж відкриває нові можливості для ще більш точного та швидкого генерування відповідей, що сприяє ще більшій інтеграції та зручності в комунікації.

Нейронні підходи з'явилися як потужний засіб для генерації відповідей, спираючись на можливості навчання представлення глибоких нейронних мереж. Ці методи можна навчати наскрізним способом, сприяючи бездоганній інтеграції різних компонентів у діалогові системи, включаючи розпізнавання мовлення, відстеження стану та вивчення політики [5][6]. Диференційованість цих моделей дозволяє оптимізувати за допомогою градієнтних методів, підвищуючи їх ефективність у динамічних розмовних контекстах [6].

Методи формування відповідей у допоміжній комунікації, засновані на генерації, включають створення відповідей слово за словом, що забезпечує високий ступінь гнучкості в діалозі. Ці методи використовують архітектури глибокого навчання, де нейронні мережі кодують дані користувача та знання в безперервні

векторні представлення, виконують міркування в нейронному просторі, а потім декодують згенеровану відповідь назад у природну мову [5][6]. Цей підхід є корисним, оскільки він може створювати нові відповіді, адаптовані до певного контексту розмови.

Навпаки, методи, засновані на пошуку, генерують відповіді шляхом вибору з попередньо визначеного пулу відповідей, зазвичай створених на основі існуючих людських взаємодій. Цей підхід пом'якшує м'якість, яка часто зустрічається у згенерованих відповідях, надаючи більш консервативне рішення для генерації відповідей[5]. Хоча методи пошуку можуть обмежити різноманітність відповідей через їх фіксовану природу, вони зазвичай використовуються в комерційних програмах завдяки своїй ефективності та надійності в обробці типових запитів користувачів [6].

Гібридні методи поєднують сильні сторони методів, заснованих на генерації та пошуку. Ці системи використовують здатність глибокого навчання вивчати представлення конкретних завдань, одночасно включаючи символічні підходи для кращої інтерпретації та ефективності виконання. Наприклад, гібридна система може використовувати модель на основі генерації для складних запитів і модель пошуку для більш простих взаємодій, таким чином оптимізуючи загальну продуктивність у допоміжних комунікаційних завданнях [5] [6].

Методи навчання з підкріпленням (RL), такі як Q-навчання та підходи до градієнта політики, також досліджувалися для генерації відповідей у діалогових системах. Ці методи зосереджені на оптимізації політик на основі відгуків користувачів системи, що робить їх придатними для обробки складних багатоетапних розмов [5] [6]. Однак методи RL можуть бути ресурсомісткими, оскільки вони часто вимагають великих даних про взаємодії для вивчення ефективних стратегій реагування [6].

III залишається в авангарді технологічних досягнень, пропонуючи значні вдосконалення комунікаційних засобів. Генеративний штучний інтелект, прикладом якого є такі додатки, як ChatGPT, використовує обробку природної мови для

створення тексту та залучення користувачів до розмови, піднімаючи критичні питання про природу контенту, створеного людиною та машиною. У майбутньому розвитку штучного інтелекту очікується, що його здатність імітувати людську взаємодію покращиться, сприяючи більш інтуїтивно зрозумілим методам спілкування, які можуть допомогти людям у різних контекстах, таких як обслуговування клієнтів і особиста допомога.

Розробка зрозумілого штучного інтелекту має вирішальне значення для забезпечення прозорості та довіри до автоматизованих засобів зв'язку. Ця технологія не тільки дає результати, але й пояснює аргументацію своїх висновків. Ця можливість має важливе значення для підвищення довіри користувачів, особливо коли системи штучного інтелекту використовуються в чутливих сферах, таких як охорона здоров'я та освіта, де точна та надійна комунікація є життєво важливою.

Просторові обчислення представляють ще один рубіж у комунікаційних технологіях, дозволяючи користувачам взаємодіяти з цифровим вмістом у фізичному просторі за допомогою передових датчиків та інтерфейсів. Ця інновація покращує спільну діяльність, забезпечуючи захоплюючий досвід віртуальної освіти, ділових зустрічей і маркетингових кампаній. Інтегруючи просторові обчислення в комунікаційні стратегії, організації можуть створювати більш привабливе та інтерактивне середовище, яке сприяє кращому обміну інформацією та командній роботі [7].

Комунікаційна модель Шеннона-Вівера служить теоретичною основою для аналізу комунікаційних процесів, у тому числі тих, що беруть участь у формуванні відповідей на допоміжну комунікацію. Ця модель ідентифікує критичні компоненти, такі як відправник, повідомлення та отримувач, і підкреслює важливість зворотного зв'язку в циклі зв'язку. Незважаючи на структурований підхід, лінійність моделі піддається критиці за недостатнє вирішення складнощів зворотного зв'язку, особливо в динамічних комунікаційних середовищах [8]. У прикладних умовах розуміння цих моделей може сприяти розробці діалогових систем, спрямованих на покращення залучення користувачів та ефективності спілкування. Майбутні дослідження у сфері

генерації відповідей для допоміжної комунікації повинні зосередитися на вдосконаленні методологій, що використовуються в системах багаточергового діалогу. Оскільки поточні моделі, як правило, демонструють складність через глибоку нейронну архітектуру, існує нагальна потреба в простіших, але ефективних структурах, які можуть використовувати контекст розмови для покращеного формування відповідей. Вивчаючи допоміжні завдання, які покращують розуміння контексту, дослідники можуть розробляти моделі, які зберігають високу точність у застосунках у реальному світі. Наприклад, оптимізація моделей за допомогою таких завдань, як відновлення порядку слів або відновлення порядку висловлювань, показала перспективу в досягненні кращих локальних оптимумів у завданнях генерації відповідей.

Потенціал додатків штучного інтелекту (AI) і глибокого навчання (DL) у доповнювальних і альтернативних комунікаційних системах (AAC) вимагає подальшого вивчення. Ці технології можуть значно розширити можливості пристроїв AAC, покращивши їх здатність розуміти та передбачати заплановану комунікацію користувачів, зрештою роблячи взаємодію швидшою та природнішою для людей з проблемами спілкування, такими як аутизм. Механізми безперервного навчання можуть бути інтегровані в ці системи для адаптації до вподобань користувача з часом, тим самим покращуючи загальний досвід спілкування [9].

1.2 Огляд технології великих мовних моделей та технології RAG

Великі мовні моделі (Large Language Models, LLM) – це складні системи штучного інтелекту, спеціально розроблені для обробки, генерації та інтерпретації мовлення людей [10]. Вони знаходять застосування в ряді завдань, що варіюються від перекладу, до текстової генерації, відповіді на запитання, аналізу емоцій, автоматизації обслуговування клієнтів та багатьох інших. Їхнє визнання зростає завдяки здатності виконувати складні завдання, раніше доступні лише людині. LLM стали основою для розробки численних сучасних технологій, зокрема чат-ботів,

систем рекомендацій, пошукових механізмів і автоматизованих інструментів. Вони інтегруються в платформи соціальних мереж, освітні сервіси, медичні системи та бізнес-додатки.

LLM – це нейронні мережі, що навчаються на величезних обсягах текстових даних з метою розпізнавання та створення зв'язного тексту. В їх основі лежать сучасні алгоритми, такі як трансформери, які забезпечують ефективну обробку контексту.

Історія розвитку великих мовних моделей (LLM) почалася з використання базових підходів, таких як n-грамні моделі і методи, засновані на частотності слів. Ці ранні алгоритми мали обмежену здатність обробляти текст, оскільки не враховували складні мовні зв'язки і мали вузький контекст. Згодом, з розвитком нейронних мереж, зокрема рекурентних нейронних мереж (RNN) і моделей з довготривалою короткочасною пам'яттю (LSTM), з'явилася можливість краще враховувати послідовність слів і контекст у текстах.

Револьюційний поворот в еволюції LLM відбувся у 2017 році, коли Google представив свою роботу під назвою "Увага - це все, що вам потрібно", яка вводила архітектуру трансформерів. Цей новаторський підхід витіснив рекурентні нейронні мережі й став фундаментом для сучасних моделей, таких як GPT (Генеративний Перенавчений Трансформер), що був створений OpenAI і вражає своєю неймовірною здатністю генерувати текст; BERT (Двобічні Кодувальні Представлення з Трансформерів), розроблений Google, що акцентує увагу на глибокому контекстному розумінні тексту; та LLaMA (Велика Мовна Модель Meta AI), ініціатива Meta, спрямована на оптимізацію використання обчислювальних ресурсів [11].

Основними принципами функціонування таких моделей є трансформери, які складають основну архітектуру для обробки тексту. Механізм уваги допомагає визначити найбільш значущі елементи тексту, що сприяє кращому розумінню контексту. Навчання на великих обсягах текстових даних дозволяє моделям адаптуватися до різноманітних стилів та тем, підвищуючи їх ефективність у широкому спектрі задач.

Моделі великих мовних моделей (LLM) мають широкий спектр застосувань у різних сферах. У бізнесі вони використовуються для автоматизації обслуговування клієнтів, зокрема в чат-ботах, що здатні швидко і ефективно відповідати на запити, а також для створення звітів та аналізу великих обсягів текстової інформації, що дозволяє зекономити час і ресурси. У медицині LLM можуть допомогти в інтерпретації клінічних записів, аналізуючи медичні документи, та в підтримці діагностики, надаючи лікарям корисні рекомендації на основі даних про пацієнта.

В освіті ці моделі стають потужним інструментом для індивідуалізованого навчання, створюючи персоналізовані матеріали, що відповідають рівню знань учня. Крім того, LLM використовуються для автоматизованого оцінювання робіт, що значно полегшує процес перевірки завдань у великих навчальних установах.

У креативних сферах вони відіграють важливу роль у генерації текстів, написанні сценаріїв, а також у створенні художніх виробів, допомагаючи авторам швидше розвивати ідеї та реалізовувати творчі проєкти. В наукових дослідженнях LLM дозволяють обробляти величезні масиви даних, що є важливим для нових відкриттів і формулювання гіпотез, а також для пошуку нових ідей на основі вже існуючих знань [12].

Великі мовні моделі (LLM) мають багато переваг, які роблять їх потужним інструментом для обробки тексту та вирішення різноманітних задач. Однією з основних переваг є їх виняткове розуміння контексту та мовних зв'язків, що дозволяє більш точно інтерпретувати та генерувати текст. Це дозволяє моделі ефективно працювати з великими обсягами інформації, враховуючи складні взаємозв'язки між словами та фразами. Крім того, LLM володіють високою адаптивністю: вони можуть бути налаштовані на виконання різних завдань, таких як автоматичний переклад, створення контенту або аналіз текстів. Моделі здатні працювати в реальному часі, що відкриває можливості для інтерактивних додатків, таких як чат-боти та персоналізовані помічники.

Однак, незважаючи на всі переваги, існують і суттєві мінуси, пов'язані з використанням великих мовних моделей. Одним із найбільших недоліків є висока

вартість обчислень та велике споживання енергії, оскільки для тренування таких моделей потрібні значні обчислювальні ресурси. Це може бути проблемою з точки зору як економічних, так і екологічних витрат. Крім того, якщо навчальні дані містять стереотипи чи упередження, моделі можуть їх передавати в результатах своєї роботи, що призводить до непередбачуваних або неприязних висловлювань. Ще одним важливим аспектом є можливість генерації недостовірної інформації: навіть найпотужніші моделі не завжди можуть гарантувати точність і достовірність створеного контенту. Окрім того, великі мовні моделі не мають свідомого розуміння тексту, яке властиве людині, і їхній "розум" обмежений лише математичними взаємозв'язками між словами, що означає, що вони не завжди можуть зрозуміти глибину або емоційний контекст деяких висловлювань.

Можна очікувати поглибленої інтеграції LLM у повсякденність, покращення їхньої продуктивності, а також розширення функціональних можливостей шляхом поєднання з іншими інноваціями, наприклад, робототехнікою чи доповненою реальністю. Великі мовні моделі виступають потужним чинником, що змінює сучасний світ, надаючи нові можливості для автоматизації, навчання та комунікації. Їх потенціал дозволяє значно прискорити численні процеси, проте це супроводжується ризиками, такими як етичні виклики та великий витратний характер ресурсів.

Retrieval-Augmented Generation (RAG) – це інноваційний підхід у сфері штучного інтелекту, що поєднує потужності генеративних моделей, таких як GPT, із механізмами інформаційного пошуку. Основна концепція полягає у використанні широких зовнішніх баз даних для поповнення відповідей генеративної моделі найсвіжішими даними. Це відкриває можливості для розв'язання завдань, які вимагають високої точності, актуальності та орієнтації на конкретні факти. У нашу епоху, коли інформації стає все більше, доступ до релевантних даних є ключовим для ефективного вирішення численних задач. RAG не лише споживає знання, навчання моделі, а й інтегрує живу, актуальну інформацію, що робить цю технологію надзвичайно цінною для бізнесу, науки та користувачів [13].

Архітектура RAG (Retrieval-Augmented Generation) поєднує два основні компоненти. Перший з них – це процес пошуку (Retrieval), який відповідає за знаходження релевантної інформації з зовнішніх джерел, таких як бази знань, документи чи Інтернет. Другий компонент – це генерація (Generation), який формує відповіді, спираючись на отриману інформацію та власні знання моделі. Ця комбінація дозволяє системі не лише працювати з великою кількістю зовнішніх даних, але й ефективно генерувати текстові відповіді, що максимально відповідають заданому контексту. Механізм функціонування системи починається з того, що користувач формулює запит. Після цього система активує пошуковий механізм для знаходження відповідних документів чи даних. Отримані матеріали передаються генеративній моделі, яка обробляє їх, комбінуючи з власними знаннями, і на основі цього створює відповідь [13].

Система RAG знаходить широке застосування в різних сферах, де потрібне ефективне оброблення та інтеграція великої кількості інформації. У бізнес-аналітиці RAG активно використовується для автоматизації створення звітів та пошуку важливої інформації, необхідної для ухвалення стратегічних рішень. Завдяки своїй здатності швидко обробляти великі обсяги даних, система допомагає виявляти ключові фактори, які можуть впливати на бізнес-процеси.

У наукових дослідженнях RAG дозволяє ефективно обробляти великі масиви даних, знаходити специфічні дослідження чи факти, що значно пришвидшує процес аналізу та допомагає вченим зосереджуватися на найважливіших аспектах. У галузі юриспруденції ця система забезпечує швидкий доступ до юридичних консультацій або допомагає знаходити відповідні законодавчі норми для розв'язання конкретних правових питань.

В освіті RAG використовується для створення адаптивних навчальних матеріалів, які постійно оновлюються на основі актуальної інформації, що дозволяє студентам та учням отримувати найсучасніші знання у своїх навчальних сферах. У медицині система допомагає інтегрувати дані з різних джерел для підтримки

діагностики та надання точних рекомендацій, що може бути критично важливим у прийнятті рішень щодо лікування пацієнтів.

Крім того, у сфері клієнтського обслуговування RAG використовується для відповіді на складні запити, що вимагають доступу до специфічної інформації. Це дозволяє швидко і точно вирішувати питання клієнтів, підвищуючи ефективність роботи служби підтримки та полегшуючи процес обслуговування [14].

Однією з основних переваг системи RAG є її актуальність. Вона забезпечує відповіді на запити, використовуючи найновішу доступну інформацію, що дає змогу користувачам отримувати актуальні та точні дані в реальному часі. Ця система також демонструє значну гнучкість, оскільки може адаптуватися до різних доменів і специфічних завдань, надаючи рішення в таких сферах, як бізнес, наука, медицина чи юриспруденція. Завдяки цьому, RAG стає потужним інструментом для розв'язання різноманітних проблем. Крім того, система підвищує точність генеративних моделей, оскільки базується на релевантних даних, що дозволяє знижувати ймовірність помилок і робити відповіді більш достовірними. Масштабованість також є важливою перевагою: RAG може інтегруватися з великими базами знань, що забезпечує доступ до величезних обсягів інформації та дозволяє системі обробляти значні дані.

Водночас, система має й деякі недоліки. Одним із них є можливість появи шуму в результатах, що полягає в тому, що система може вибирати нерелевантну інформацію під час пошуку, що призводить до неточних або неповних відповідей. Крім того, RAG залежить від якості даних: якщо бази знань, з якими вона працює, мають помилки або недостовірну інформацію, це може призвести до спотворених відповідей. Ще одним значущим недоліком є витратність ресурсів – процес інтеграції пошукової частини з генерацією потребує великих обчислювальних потужностей, що може бути дорогим для користувачів з обмеженими ресурсами. Нарешті, проблема безпеки також є важливою: доступ до конфіденційних даних може становити ризики, якщо система не має належних механізмів захисту [15].

RAG розширює горизонти генеративних моделей, надаючи їм можливість не тільки бути креативними, а й дуже інформованими. Це є особливо важливим у

ситуаціях, де актуальність і точність інформації грають вирішальну роль. Подальше вдосконалення стратегій фільтрації інформації, оптимізація витрат на обробку, а також інтеграція з різноманітними доменними базами знань може перетворити RAG на ключову складову екосистеми штучного інтелекту.

Синергія великих мовних моделей (LLM) та RAG відкриває нові можливості для розвитку технологій штучного інтелекту. Об'єднання цих двох підходів дозволяє вирішити одну з основних проблем LLM – обмеженість навчального корпусу. Завдяки інтеграції з RAG, LLM перетворюється на інструмент не лише для генерації тексту, але й для ефективної інтерпретації специфічних інформаційних даних, що значно розширює його функціональні можливості.

LLM здатна також вдосконалювати пошукові запити, формулюючи їх більш контекстуально, що дозволяє системі ефективно взаємодіяти з механізмами пошуку в RAG. Це особливо важливо, оскільки використання природної мови робить пошуковий процес більш точним і релевантним. Поєднання з RAG дозволяє створювати масштабовані системи, які можуть без обмежень отримувати доступ до зовнішніх джерел даних. Таке рішення є менш ресурсоємним, ніж використання великих монолітних мовних моделей, оскільки дозволяє додавати нову інформацію через зовнішні ресурси, замість того, щоб проводити постійне повторне навчання моделі. Використання RAG також допомагає знизити витрати, пов'язані з навчанням LLM, оскільки нові дані можуть бути інтегровані через зовнішні джерела, а не за рахунок додаткових обчислювальних потужностей для повторного тренування моделі.

Ці технології активно використовуються в різних сферах. У пошукових системах, таких як Google або Bing, інтеграція LLM і RAG дозволяє надавати користувачам більш точні й обґрунтовані відповіді. У корпоративних додатках, що базуються на RAG, активно обробляються великі обсяги даних для прийняття рішень. В освітніх платформах ці технології надають студентам доступ до актуальної інформації у зручному для навчання форматі.

Однак, існують і певні виклики при співіснуванні LLM і RAG. Якість отриманих результатів значною мірою залежить від надійності зовнішніх джерел, що може вплинути на точність відповідей. Інтеграція цих двох систем вимагає складних механізмів індексації та вибору даних для забезпечення ефективності пошуку. Крім того, паралельне використання LLM і RAG може призводити до обчислювальних затримок, що знижує продуктивність [16].

Загалом, поєднання LLM і RAG дозволяє створювати більш ефективні, сучасні та масштабовані системи. LLM забезпечує гнучкість і інноваційність, а RAG надає доступ до зовнішніх джерел інформації, підвищуючи точність і актуальність даних. В цих технологіях лежить основа розвитку сучасного штучного інтелекту, особливо в сферах, де важливими є фактична точність і оновленість даних.

1.3 Огляд існуючого програмного забезпечення

Proloquo2Go – це інноваційний додаток, призначений для альтернативної та додаткової комунікації, що активно допомагає особам, які не можуть висловлюватися усно, передавати свої думки й емоції через символи й текст. З потужною бібліотекою, що налічує понад 27,000 знаків, цей додаток створює можливості для людей різного рівня грамотності ефективно комунікувати, спираючись на зрозумілі та виразні графічні зображення. Завдяки впровадженню словника Crescendo™, Proloquo2Go підтримує мовний розвиток, полегшуючи користувачам плавний перехід від простих слів до ускладнених речень [17].



Рисунок 1.1 – Інтерфейс додатку Proloquo2Go

Однозначним плюсом програми є її багатомовність, адже вона доступна на чотирьох мовах: англійській, іспанській, французькій та нідерландській, що дозволяє одночасно використовувати кілька мов. Користувачі можуть персоналізувати словниковий запас і інтерфейс додатку на свій смак, додаючи зображення родичів або друзів на кнопки, що перетворює спілкування на більш особистісне. Окрім цього, додаток надає різноманітні рішення для доступності, пропонуючи 23 попередньо налаштовані макети кнопок – від 9 до 144 кнопок на сторінці, що дозволяє налаштовувати його відповідно до фізичних особливостей користувачів.

Проте, незважаючи на численні переваги, Proloquo2Go не позбавлений недоліків. Його висока вартість у 249.99 доларів може стати значним бар'єром для деяких користувачів. Також, хоча можливості налаштування програми є дуже широкими, новим користувачам або тим, хто не володіє технічними навичками, може бути важко впоратися з цим. Ще один серйозний мінус – це прив'язка до продукції Apple, адже додаток доступний лише на iPad, iPhone та Apple Watch, що суттєво обмежує його доступність для тих, хто користується альтернативними платформами.

В загальному, Proloquo2Go є потужним інструментом комунікації, що пропонує великий спектр функціоналу для людей з мовними бар'єрами. Попри певні недоліки, такі як висока ціна й складність налаштування, його адаптивність, підтримка мовного розвитку і доступність роблять його важливим ресурсом для багатьох.

Speech Assistant AAC – це текстово-мовний додаток, розроблений для осіб з мовленнєвими розладами, такими як афазія, МНД/БАС, аутизм, інсульт або церебральний параліч. Ця програма дозволяє користувачам організовувати категорії та фрази, які легко розміщуються на кнопках, що дає можливість формувати повідомлення для візуального або усного озвучення. Функція історії повідомлень забезпечує швидкий доступ до раніше введених фраз, оптимізуючи процес спілкування. Користувачі здатні налаштовувати розміри кнопок, текст і кольорові схеми, що робить додаток максимально вигідним для їхніх потреб.

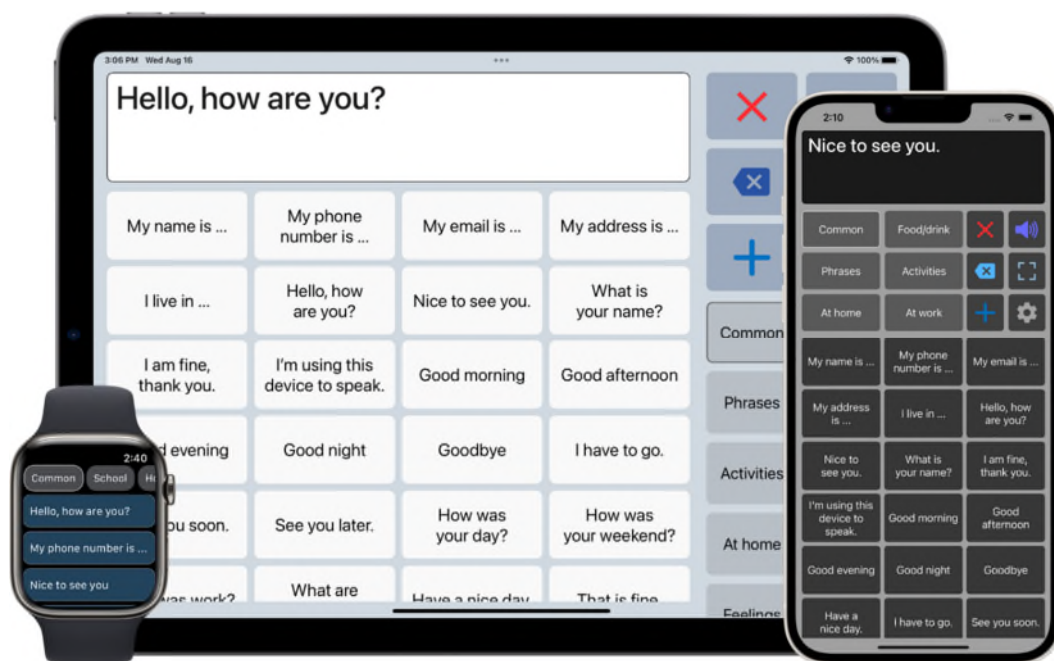


Рисунок 1.2 – Інтерфейс додатку Speech Assistant AAC

Крім того, повноекранний режим дає можливість бачити повідомлення з великим шрифтом, що особливо важливо в шумному оточенні. Додаток підтримує функцію запису голосу або користування текстово-мовним синтезатором,

встановленим на пристрої, і є багатомовним, що робить його доступним для користувачів з різних країн. Додаток також може інтегруватися з іншими пристроями, підтримуючи Bluetooth-клавіатури і можливість створення ярликів для швидкого доступу до функцій. Користувачі можуть резервувати дані через iCloud, Mail або AirDrop, що підвищує зручність програми [18].

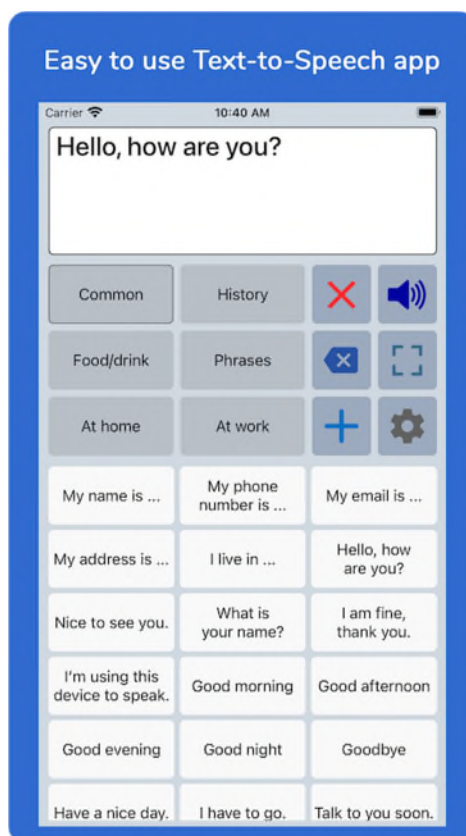


Рисунок 1.3 – Приклад роботи додатку Speech Assistant AAC

Проте, не можна не відзначити й недоліки цього додатку. Безкоштовна версія має обмежені можливості, і багато корисних функцій доступні лише в платній версії. Якість синтезованого мовлення залежить від голосів, установ яких є на користувацькому пристрої, що може вплинути на враження від користування. Більш того, додаток не завжди підтримує рідкісні мови, що може стати проблемою для користувачів з особливими мовними вимогами.

Отже, можна підсумувати, що Speech Assistant AAC – це потужний інструмент для осіб з порушеннями мовлення, який пропонує широкий спектр можливостей для

комунікації. Хоча існують певні обмеження, такі як відносна бідність безкоштовної версії та ризик залежності від якості TTS голосів, програма забезпечує простоту використання, налаштування та доступність на багатьох платформах.

1.4 Постановка задачі

Мета кваліфікаційної роботи магістра – покращення комунікації для людей з обмеженими можливостями мовлення за допомогою генерації відповідей з доповнюючим інформаційним пошуком.

Для досягнення поставленої мети необхідно реалізувати виконання наступних задач:

- провести огляд сучасних підходів до генерації відповідей з використанням мовних моделей та технологій доповнюючого інформаційного пошуку (RAG);
- розробити метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації;
- підготувати релевантні текстові дані для покращення точності генерації відповідей з доповнюючим інформаційним пошуком;
- розробити програмну реалізацію системи комунікації між людьми з використанням генерації відповідей з доповнюючим інформаційним пошуком;
- провести тестування взаємодії системи комунікації з користувачами та проаналізувати коректність його відповідей.

Розділ 2 Розробка методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації.

2.1 Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації

Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації базується на інтеграції інформації з релевантних джерел та врахуванні контексту спілкування, що робить його ефективним і зручним для людей з вадами слуху чи мовлення. Основна ідея полягає у створенні методу, що дозволить отримувати індивідуалізовані відповіді на запити, спираючись на завантажені документи, історію чату, запитання та потреби користувача.

На рисунку 2.1 представлено схему методу та її опис.



Рисунок 2.1 – Схема методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації

Вхідними даними методу генерації відповідей для допоміжної комунікації є текстовий запит користувача, контекст із завантажених документів, таких як профіль користувача або інші дані, а також історія попередньої взаємодії, яка включає повідомлення чату. Ці дані забезпечують необхідну інформацію для формування релевантної відповіді в рамках допоміжної комунікації.

На Етапі 1 виконується завантаження контексту, де відбувається імпорт документів у базу даних. Документи розбиваються на смислові частини, що дозволяє створити унікальні ідентифікатори для кожного фрагмента тексту. Потім генеруються ембедінги тексту, які допомагають у подальшому пошуку схожих даних. Цей процес дозволяє структурувати дані для зберігання та подальшого використання при генерації відповідей.

Етап 2 передбачає пошук релевантної інформації. Використовуючи ембедінги, метод здійснює пошук у базі даних найбільш схожих текстових фрагментів, що відповідають запиту користувача. Це дає змогу сформувати контекст для створення відповіді, який враховує дані з історії чату та профілю користувача.

На Етапі 3 відбувається формування відповіді. Метод будує динамічну підказку із урахуванням наданого контексту та історії чату. Генерується кілька варіантів відповіді, що можуть бути надані користувачеві. Вибір варіантів відповіді на цьому етапі дає можливість надати користувачеві різні варіанти, з яких він може вибрати найбільш відповідний.

Вихідними даними методу є набір варіантів відповідей, які адаптовані до запиту користувача. Окрім цього, відбувається оновлення бази даних із релевантною інформацією, яка може бути використана в наступних сеансах комунікації, а також збереження історії взаємодії, що забезпечує можливість кращого персоналізованого спілкування.

2.2 Архітектура методу генерації тексту з інформаційним пошуком

Основною метою методу генерації відповідей з доповнюючим інформаційним пошуком є надання точних, релевантних та значущих відповідей на запитання

користувачів, зокрема для людей з вадами слуху та мови. Цей метод дозволяє ефективно поєднувати генерацію тексту за допомогою великої мовної моделі та пошук відповідної інформації в документах або базах даних, що забезпечує високу точність і адаптивність відповідей до конкретних запитів. Метод базується на трьох основних етапах: індексація даних, пошук інформації та генерація відповіді на основі знайденої інформації. Ці етапи взаємодіють між собою, що забезпечує високу точність і релевантність відповідей.

Процес роботи методу зображено на рисунку 2.2

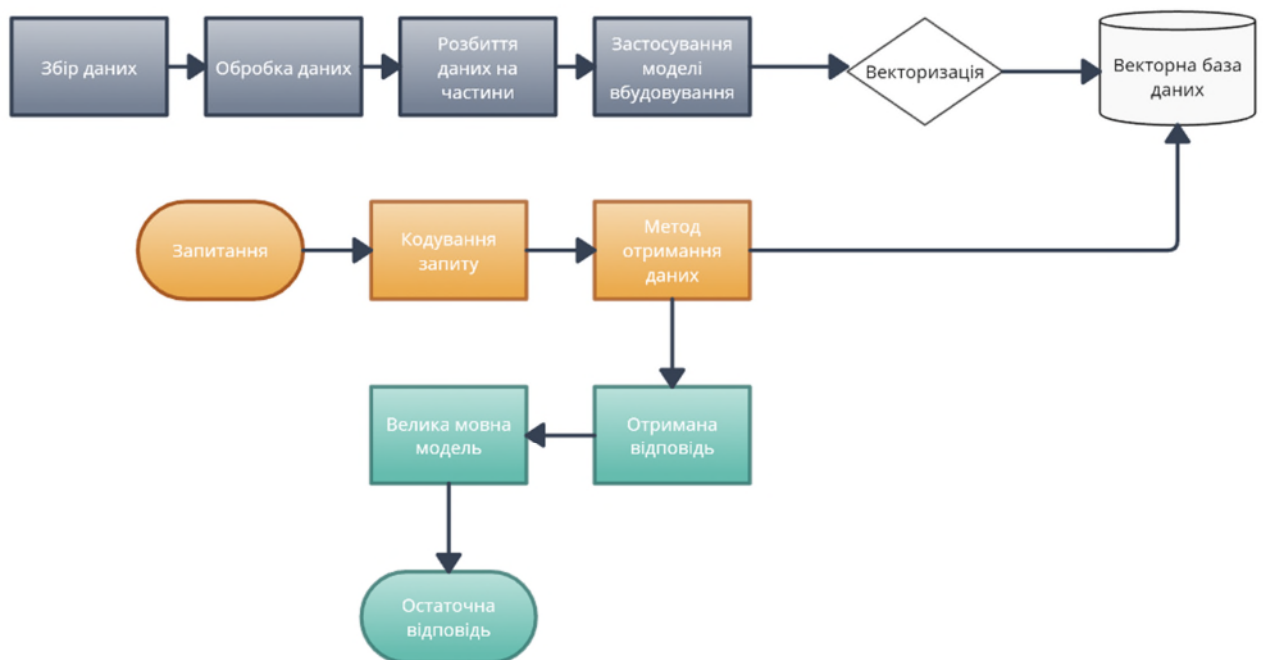


Рисунок 2.2 – Схема роботи методу генерації тексту з інформаційним пошуком

2.2.1 Індекссація даних

Індекссація даних забезпечує обробку даних для зберігання в базі даних у вигляді векторних представлень. Це дозволяє системі знаходити та зіставляти інформацію, використовуючи семантичну подібність, а не простий текстовий збіг.

Процес індекссації даних зображено на рисунку 2.3

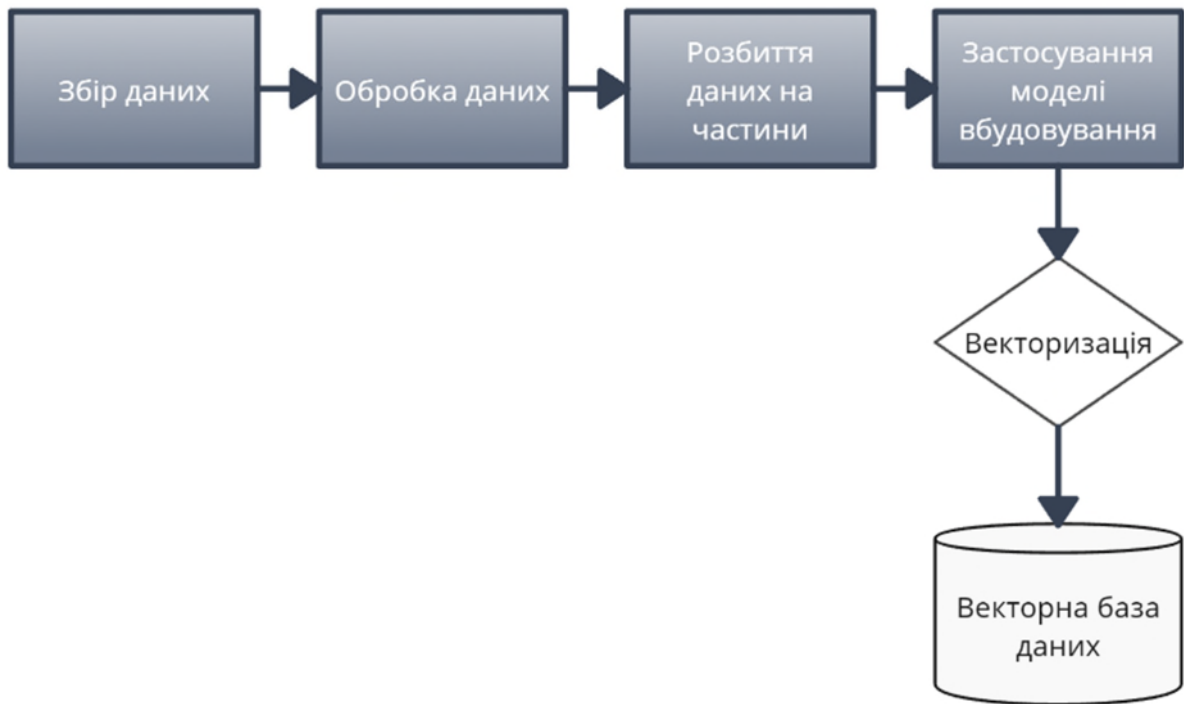


Рисунок 2.3 – Процес індексації даних

Збір даних може охоплювати не лише документи, а й різноманітні джерела, такі як веб-сайти, бази даних, API або навіть зображення. Вибір інструментів для збору інформації залежить від джерела та потреби в забезпеченні цілісності та відповідності запитам користувачів.

На етапі обробки даних важливо очистити інформацію від зайвих елементів та неструктурованих даних. Застосовуються різні техніки, зокрема нормалізація тексту, видалення стоп-слів та вирівнювання формату. Крім того, необхідно виявити та видалити дублікати, які можуть вплинути на ефективність пошуку.

Щоб полегшити пошук та покращити результати, великі текстові блоки або документи розбиваються на менші частини – наприклад, речення чи абзаци. Ці частини можуть бути автономними або взаємопов'язаними залежно від контексту.

Для трансформації тексту в багатовимірні вектори, які відображають зміст, застосовуються моделі вбудовування. Вони можуть бути натреновані на конкретних

задачах або використовуватися універсальні моделі, які надають високоякісні ембедінги.

Векторизація дозволяє створити семантичне відображення тексту, що дає змогу системам порівнювати фрагменти тексту через вимірювання відстані між векторами, наприклад, за допомогою косинусної схожості. Чим ближчі два вектори, тим подібніші їх змісти.

Після генерування векторів вони зберігаються в спеціалізованих векторних базах даних. Ці бази даних оптимізовані для швидкого пошуку та масштабування, забезпечуючи ефективний семантичний пошук і надання релевантних результатів у реальному часі.

2.2.2 Пошук інформації

Даний етап відповідає за пошук релевантної інформації, використовуючи векторні представлення. Він дозволяє отримати потрібні документи на основі запиту користувача.

Процес пошуку інформації зображено на рисунку 2.4

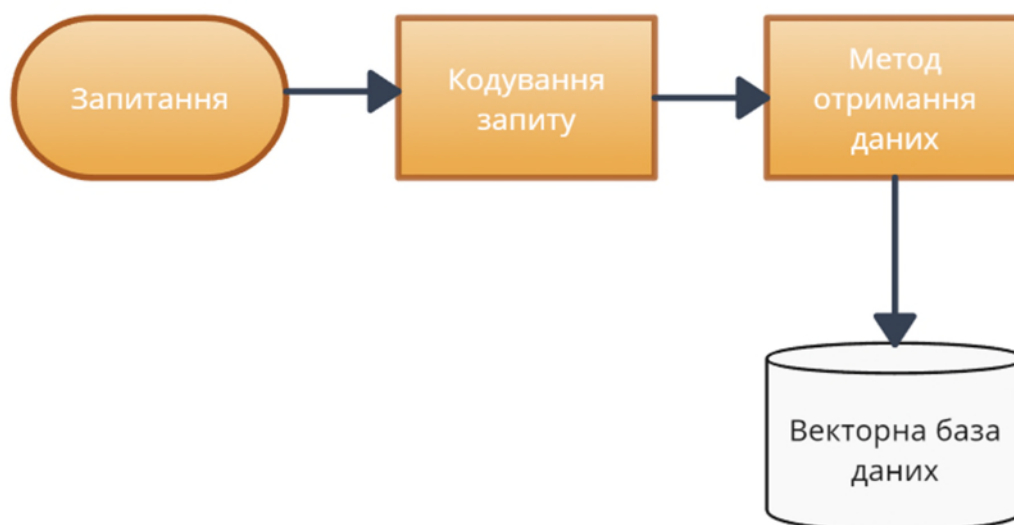


Рисунок 2.4 – Процес пошуку інформації

Користувач вводить запит, який може бути як простим питанням, так і складним пошуковим запитом. Часто в запитах є ключові слова або контекст, які допомагають сформулювати точні критерії для подальшого пошуку векторів.

Запит перетворюється на вектор за допомогою спеціалізованих кодувальних моделей, які обробляють текст і відображають його зміст у вигляді багатовимірних векторів. Це дозволяє виявляти схожість між запитом і іншими текстами в базі даних на глибшому рівні, а не лише через точний збіг слів.

Для пошуку найбільш релевантних результатів використовуються методи семантичного пошуку, що порівнюють вектори за допомогою відстані або подібності між ними. Алгоритми, як-от максимальна маржинальна релевантність, не лише підвищують точність, але й покращують різноманітність відповідей, мінімізуючи дублювання інформації у результатах пошуку.

Векторні бази даних, оптимізовані для швидкого пошуку, дозволяють працювати з великими обсягами даних. Вони використовують різні індексаційні структури для ефективного знаходження найбільш релевантних документів, що знижує затримки та покращує час відгуку системи.

2.2.3 Генерація відповіді

Генерація відповіді обробляє знайдену інформацію та створює кінцеву відповідь за допомогою генеративних моделей. Його мета – зробити результат зручним для сприйняття користувачем.

Процес генерації відповіді зображено на рисунку 2.5



Рисунок 2.5 – Процес генерації відповіді

На цьому етапі система отримує фрагменти інформації з бази даних, знайдені під час пошуку. Ці фрагменти можуть містити текст, уривки з документів чи інші дані. Важливо, щоб ці частини були достатньо детальними та містили необхідний контекст для формування змістовної відповіді.

Велика мовна модель (LLM) відповідає за генерацію відповіді, інтегруючи знайдену інформацію в контекст запиту користувача. Модель не лише надає прямі відповіді, але й може додавати пояснення, логічні висновки або структурувати відповідь так, щоб вона ставала зрозумілою та корисною. Для цього використовуються різні стратегії, зокрема переформулювання, підсумки або аналітичні висновки.

Остаточна відповідь, що отримує користувач, об'єднує знайдені дані та логічні висновки, сформовані моделлю. Це може бути текстова відповідь, яка надає прямий

відгук на запит, або короткий підсумок, що висвітлює основні моменти. Важливо, щоб відповідь залишалась релевантною і легкою для сприйняття користувачем.

2.3 Метод векторного вбудовування

Векторне вбудовування відіграє ключову роль в архітектурі сучасних методів генерації тексту з інформаційним пошуком, забезпечуючи ефективне перетворення неструктурованих даних у компактні та семантично насичені вектори. У процесі індексації даних векторизація дозволяє створювати базу даних, яка зберігає інформацію у форматі, оптимізованому для швидкого доступу та точного пошуку. На етапі пошуку векторна база виступає основою для порівняння закодованих запитів з векторизованими фрагментами даних, забезпечуючи релевантність отриманих відповідей. Інтеграція векторного підходу з мовними моделями у фінальній генерації відповіді дозволяє отримувати якісні, контекстуально точні результати. Таким чином, векторні вбудовування є важливим інструментом, що поєднує індексацію, пошук і генерацію в єдиний ефективний процес.

Векторне вбудовування (embedding) – це метод у машинному навчанні, який перетворює об'єкти, такі як тексти, зображення чи слова, у числові вектори. Ці вектори зберігають семантичні та контекстуальні властивості об'єктів, маючи при цьому меншу розмірність порівняно з оригінальними даними. Це дозволяє працювати з необробленими даними, такими як текст, за допомогою математичних моделей. Кожен елемент (наприклад, слово або речення) відображається у вигляді високорозмірного вектора, де розмірність визначається кількістю характеристик, що описують цей елемент. Цей підхід є основою для багатьох сучасних застосувань у сферах, як-от пошук, класифікація та генерування тексту, а також у додатках для обробки зображень і звуків [19].

Процес роботи методу векторного вбудовування зображено на рисунку 2.6

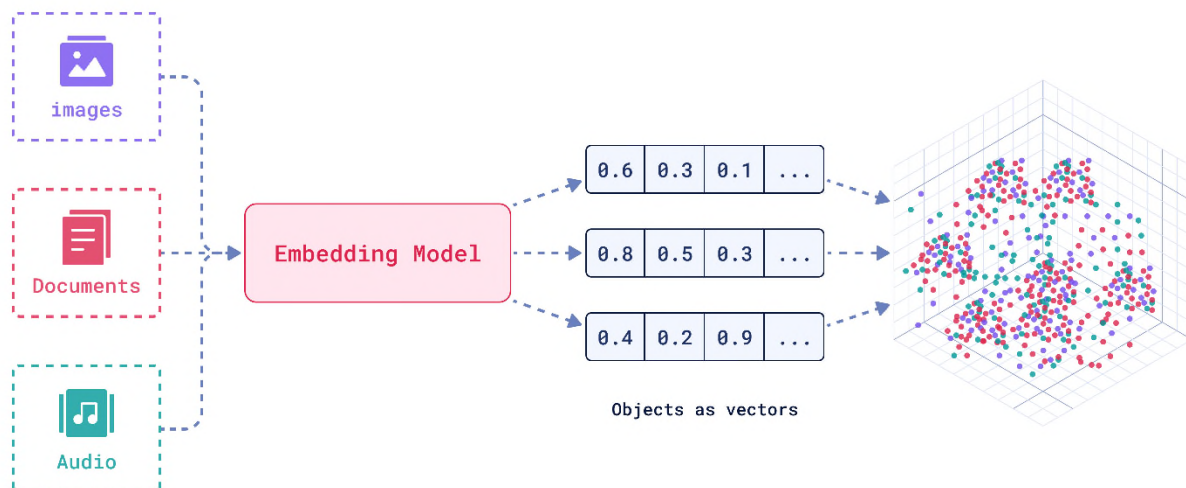


Рисунок 2.6 – Процес роботи методу векторного вбудовування [19]

Векторне вбудовування забезпечує математичне представлення тексту, яке дозволяє ефективно обробляти інформацію на комп'ютерах. Кожне слово або фраза перетворюється у вектор – набір чисел, що визначають його місце в багатовимірному просторі. Наприклад, за допомогою векторних вбудовувань можна обчислити подібність між словами. Слова "машина" і "транспортний засіб" будуть знаходитись поруч один з одним, так само як і "кіт", "собака", "тварина" та "домашня тварина". Але ці слова будуть розділені на дві групи: слова, пов'язані з тваринами, та слова, пов'язані з транспортними засобами, які будуть розташовані в різних координатах, оскільки не мають нічого спільного між собою [20].

На рисунку 2.7 зображено приклад розбиття слів на групи за типами.

Vector Embeddings: Words as Points in 3D Space

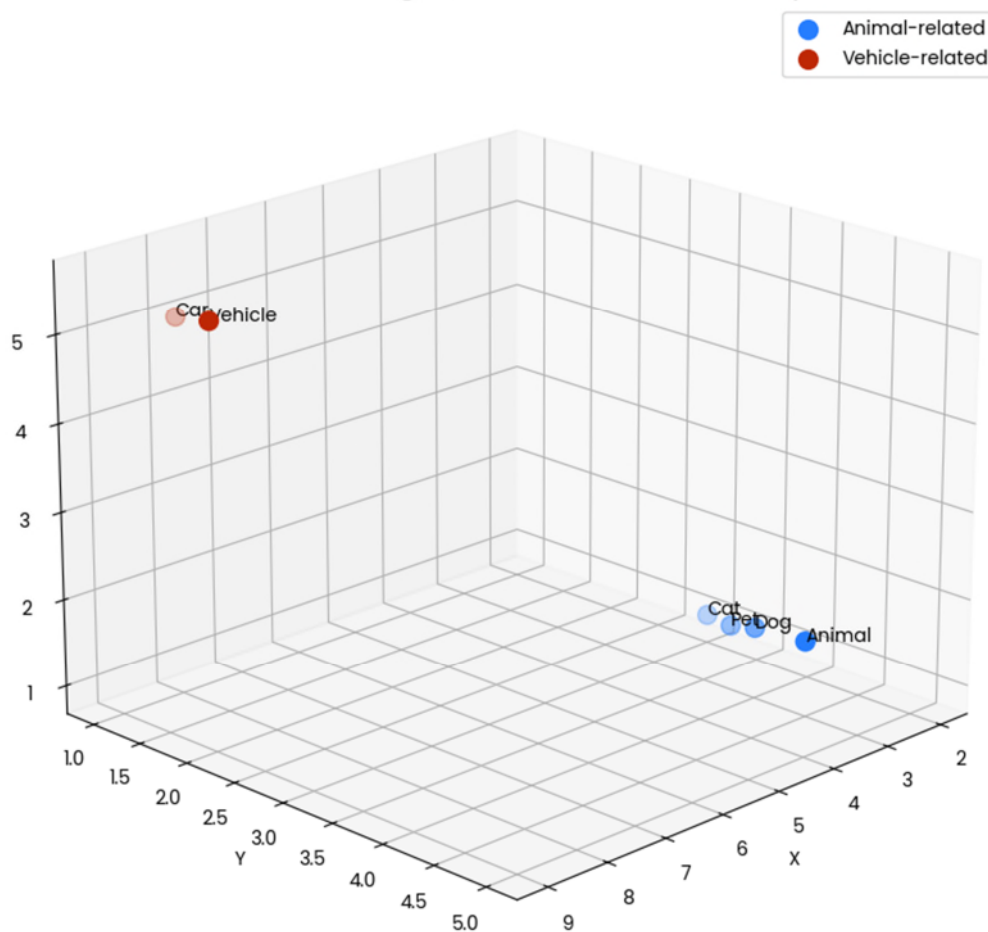


Рисунок 2.7 – Два кластери слів у тривимірному просторі

Вбудовування векторів дозволяють алгоритмам машинного навчання виявляти закономірності в даних і виконувати завдання, такі як аналіз настроїв, переклад мови, системи рекомендацій та інші програми. Різні додатки використовують різні типи векторних вбудовувань, кожне з яких має свою специфічну мету.

Вбудовування слів – це тип векторних вбудовувань, що перетворює слова на числові вектори в безперервному просторі. Вони відображають семантичне значення та контекстуальні зв'язки між словами, що дозволяє моделям машинного навчання розуміти мову та вирішувати завдання обробки природної мови.

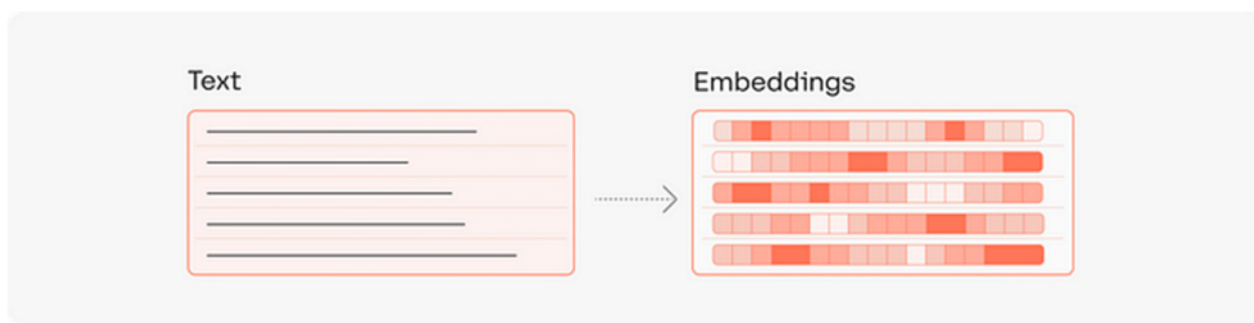


Рисунок 2.8 – Процес вбудовування слів

Вбудовування слів активно використовуються в таких програмах, як аналіз настроїв, моделювання мови та класифікація текстів.

Вбудовування зображень передбачають перетворення візуальних даних у числові вектори, що вловлюють важливі характеристики зображень. Вони застосовуються для завдань, таких як класифікація зображень, виявлення об'єктів і пошук зображень за змістом.

Вбудовування документів розширюють ідею вбудовування слів, перетворюючи цілі документи в числові вектори. Вони фіксують загальний зміст і контекст текстів, що дозволяє алгоритмам машинного навчання краще порівнювати великі об'єми інформації.

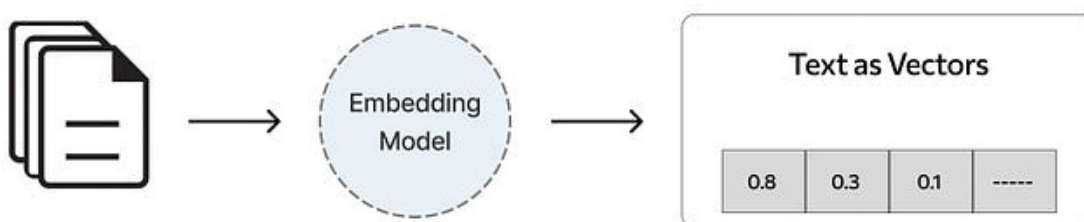


Рисунок 2.9 – Процес вбудовування документів

Крім того, існують доменно-спеціальні вбудовування, адаптовані для конкретних типів даних і програм. Наприклад, вбудовування профілів користувачів допомагає фіксувати їхні вподобання, вбудовування продуктів використовується в системах рекомендацій для обчислення подібності між товарами, вбудовування аудіо допомагають обробляти мовлення, а вбудовування соціальних мереж дозволяють

аналізувати соціальні зв'язки. Кожен тип вбудовування розроблений з урахуванням специфіки конкретної галузі, що дає змогу моделям машинного навчання вирішувати спеціалізовані завдання та проводити детальний аналіз [21].

Векторні вбудовування мають кілька важливих переваг. Вони дозволяють ефективно обробляти великі обсяги даних, скорочуючи їх розмір і перетворюючи складні структури в компактні вектори, що значно економить пам'ять і обчислювальні ресурси. Крім того, цей метод універсальний і гнучкий, оскільки його можна застосовувати до різних типів даних, таких як текст, зображення чи звуки, забезпечуючи узагальнене представлення і допомагаючи моделювати семантичні залежності між об'єктами. Векторні вбудовування також покращують результати пошуку та рекомендацій, адже подібні об'єкти мають схожі вектори, що підвищує точність запитів і систем рекомендацій.

Проте є і недоліки цього підходу. Створення ефективних моделей векторних вбудовувань потребує налаштування численних параметрів, що може бути ресурсозатратним і займати чимало часу. Перетворення складних об'єктів у вектори також несе ризик втрати важливої інформації, що може бути критичним для точного аналізу. Крім того, якість вхідних даних має велике значення: низька якість даних може погіршити точність векторних представлень, що робить систему чутливою до помилок і невідповідностей. Загалом, хоча векторні вбудовування значно підвищують ефективність багатьох алгоритмів машинного навчання, вони вимагають ретельної підготовки та аналізу для уникнення можливих труднощів.

Векторні вбудовування зробили революцію в тому, як машини розуміють і обробляють складні дані, забезпечивши нове покоління програм ШІ. Від розробки мовних моделей, що стоять за чат-ботами, до створення складних систем розпізнавання зображень, векторні вбудовування є основою багатьох проривів ШІ. Інтеграція векторних можливостей у основні системи керування даними обіцяє зробити ці потужні методи більш доступними та ефективними для підприємств будь-якого розміру. Незалежно від того, чи є ви розробником, фахівцем із обробки даних

чи бізнес-лідером, розуміння та використання векторних вбудовувань буде ключовим для того, щоб залишатися в авангарді інновацій ШІ [22].

2.4 Налаштування технології RAG для генерації відповідей

RAG є інноваційним підходом до генерації тексту, що поєднує в собі можливість великих мовних моделей із механізмами доповнюючого інформаційного пошуку. Налаштування технології RAG для системи допоміжної комунікації передбачає створення цілісного процесу, який забезпечує ефективний пошук релевантних даних та їх інтеграцію у відповіді, що генеруються [23]. Це дозволяє підвищити точність, актуальність та інформативність відповідей для користувачів, які мають обмеження мовлення чи слуху. Основна мета цього налаштування – забезпечити можливість надання користувачеві не просто узагальнених відповідей, а таких, що враховують конкретний контекст, історію взаємодії та актуальну інформацію з бази даних.

Prompt Engineering є важливою складовою налаштування технології RAG. Він передбачає оптимізацію формулювання інструкцій для великої мовної моделі, щоб забезпечити генерацію релевантних і точних відповідей. У контексті допоміжної комунікації правильне застосування Prompt Engineering дозволяє адаптувати відповіді до індивідуальних потреб користувачів, враховуючи їхні запити, контекст взаємодії та наявні дані. Це сприяє зменшенню кількості некоректних результатів і підвищенню ефективності комунікації, особливо для людей з вадами мовлення чи слуху [24].

Інструкції – це вхідні дані, які надсилаються до великої мовної моделі (LLM) через платформу генеративного штучного інтелекту (GenAI), як-от ChatGPT OpenAI або Microsoft Copilot. Інструкція може бути визначена як запитання, команда, твердження, зразок коду або інша форма тексту. Деякі LLM також підтримують нетекстові інструкції, включаючи зображення та аудіофайли. Після надсилання

вхідних даних платформа застосовує їх до LLM, який використовує вхідні дані як основу для створення відповідної відповіді.

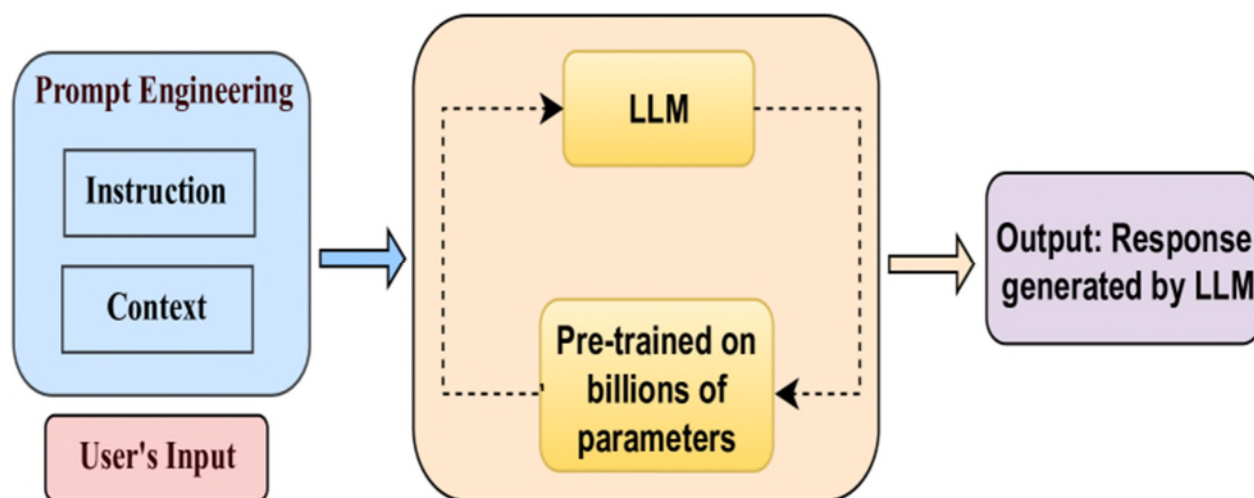


Рисунок 2.10 – Схема роботи моделі з інструкцією[24]

Модель здатна генерувати різноманітні результати в залежності від формулювання інструкції. Ця інструкція може містити як кілька слів, так і цілі абзаци, включаючи процитовані фрагменти або зображення для ілюстрації. Основна мета полягає у наданні моделі необхідних даних, що дозволяє отримувати точні і доцільні результати. Вона може представляти собою продовження попереднього запиту до LLM, наприклад, з проханням надати додаткову інформацію або уточнення для поліпшення відповіді [25].

Переваги таких підказок полягають у можливості забезпечення точних відповідей, що вказують моделі на наміри користувача, для створення релевантного контенту. Це суттєво підвищує продуктивність, дозволяючи користувачам оперативніше справлятися з завданнями та економити ресурси. Якісні інструкції значно збагачують досвід взаємодії з системами, що веде до більшого задоволення користувачів. Вони допомагають приймати зважені рішення, скорочуючи кількість ітерацій під час співпраці, що позитивно позначається як на особистій ефективності, так і на загальних процесах у організації. Крім того, такі інструкції вносять вклад в

розширення творчих можливостей, генеруючи нові ідеї та погляди, які могли б залишитися непоміченими.

Інструкції можна адаптувати відповідно до специфічних потреб користувачів. Наприклад, маркетологи можуть модифікувати стиль та тон відповіді, створюючи унікальний контент для своїх кампаній, а викладачі можуть налаштовувати навчальний процес під потреби студентів, надаючи своєчасне зворотне зв'язок [26].

Щоб модель могла генерувати корисні, точні та всебічні відповіді, інструкція повинна містити чіткі директиви для LLM. Однак варто зазначити, що сама інструкція є лише частиною величезної системи. Модель також активно використовує технології обробки природної мови та алгоритми глибокого навчання, щоби перевіряти та інтерпретувати дані, які вводять користувачі.

Для генерації коректних, значущих та релевантних відповідей необхідно використати інструкцію, яка розділена на основну інструкцію, контекст, формування відповіді.

Основна інструкція визначає завдання моделі, що полягає в наступному. Необхідно надати відповідь на запитання співрозмовника так, як це могла б зробити людина, інформація про яку міститься у контексті. У відповідях слід використовувати лише ключові слова без додаткових уточнень, описів чи пояснень. Якщо відповідь міститься у контексті, потрібно надати її. Якщо такої відповіді немає, бажано сформулювати кілька варіантів відповідей, що базуються на контексті, історії чату та самому запитанні.

Контекст інструкції включає в себе:

1. Контекст (інформація про людину, якщо є).
2. Історія чату (попереднє спілкування).
3. Запитання.

У контексті використовується інформація з документів, якщо вони наявні, історії чату та запитання. Підсумовуючи, необхідно зазначити, що в контексті інструкції застосовується інформація, отримана від користувача та його співрозмовника.

Формування відповіді уточнює, як повинні виглядати кінцеві варіанти відповіді, а саме – необхідно надати 4 короткі варіанти відповіді. Відповіді повинні:

1. Бути чіткими, короткими, простими і зрозумілими.
2. Уникати будь-яких роздумів, пояснень, уточнень чи рекомендацій.
3. Бути максимально лаконічними та зрозумілими, як у природному діалозі.

Таким чином, інструкція виконує функцію своєрідного «напрямого механізму», який забезпечує мовній моделі необхідну структуру для створення відповіді. Правильно налаштована інструкція дозволяє системі генерувати тексти, які не лише відповідають на запит користувача, але й враховують усі особливості контексту та стилю комунікації. Це підвищує загальну якість взаємодії та робить систему більш ефективною у допоміжній комунікації.

Висновки до розділу 2

Метод генерації відповідей з доповнюючим інформаційним пошуком є ефективним інструментом для персоналізованої комунікації з людьми з вадами слуху чи мовлення. Завдяки інтеграції даних з профілю користувача та історії чату, метод враховує контекст взаємодії, що дозволяє генерувати точні та індивідуалізовані відповіді. Поетапне завантаження контексту, пошук схожих даних та створення кількох варіантів відповідей сприяють зручній і гнучкій комунікації, покращуючи досвід користувача і адаптацію системи.

Архітектура методу включає індексацію даних, пошук релевантної інформації та генерацію відповідей, що підвищує точність та ефективність взаємодії. Індиксація створює семантичну базу для точного пошуку, а векторні представлення забезпечують глибше розуміння запиту. Генерація відповіді на основі знайденої інформації гарантує змістовність і адаптивність, що робить цей підхід ідеальним для допоміжної комунікації.

Векторне вбудовування є важливим компонентом цього методу, перетворюючи неструктуровані дані в семантично насичені вектори. Це покращує індексацію та точність пошуку, що є критичним для обробки природної мови. Завдяки

векторизації, система може досягати високої релевантності відповідей, забезпечуючи контекстуальну точність. Однак, для оптимальних результатів важливо налаштувати параметри моделей і забезпечити якість вхідних даних.

Налаштування технології RAG для генерації відповідей у системах допоміжної комунікації дозволяє значно підвищити точність, релевантність та інформативність відповідей для користувачів з мовленнєвими чи слуховими обмеженнями. Оптимізація процесів через Prompt Engineering забезпечує гнучке формулювання інструкцій, що адаптуються до індивідуальних потреб та контексту спілкування. Поєднання можливостей великих мовних моделей та механізмів інформаційного пошуку створює інтерактивну та ефективну комунікацію, де відповіді враховують історію взаємодії та актуальну інформацію. Правильне налаштування інструкцій та їхнього контексту сприяє швидшій та більш якісній генерації відповідей, що підвищує загальну ефективність системи та задоволення користувачів.

Розділ 3 Реалізація методу генерації відповідей для допоміжної комунікації

3.1 Опис платформи, технологій та бібліотек

Розроблена платформа є інтерактивною системою для допоміжної комунікації, що інтегрує генерацію відповідей із додатковим інформаційним пошуком. Вона орієнтована на людей з вадами мовлення та слуху, надаючи можливість ефективно комунікувати завдяки адаптивному інтерфейсу та індивідуалізованим відповідям. Основною метою платформи є створення інструменту, що підтримує соціальну інтеграцію та знижує бар'єри у спілкуванні.

Архітектурний підхід платформи базується на модульній структурі, яка складається з трьох взаємопов'язаних компонентів, кожен із яких виконує свою специфічну функцію.

Першим ключовим елементом є фронтенд, що відповідає за графічний інтерфейс. Ця частина забезпечує користувачам інтуїтивно зрозумілий доступ до функціоналу системи. Завдяки зручному дизайну, користувачі можуть легко вводити запити, переглядати отримані відповіді та здійснювати управління документами.

Другий компонент – це бекенд, який виконує основні обчислювальні операції та керує обробкою запитів. У цій частині реалізуються функції генерації відповідей, пошуку інформації та інтеграції з великою мовною моделлю та базою знань. Бекенд слугує "розумовим центром" платформи, забезпечуючи швидкість і точність її роботи.

Третім важливим елементом є база даних, яка відіграє роль сховища даних. Вона містить індексовані документи та зберігає історію запитів, що дозволяє виконувати контекстний пошук. Завдяки цьому платформа може ефективно аналізувати попередню взаємодію з користувачами та надавати більш релевантні результати. Усі ці компоненти взаємодіють між собою за допомогою чітко визначених інтерфейсів. Такий підхід дозволяє забезпечити високу продуктивність системи, її масштабованість та стабільність навіть за значного навантаження.

Для реалізації методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації було використано наступні інструменти:

1. CustomTkinter;
2. Chroma;
3. PyPDF;
4. Ollama з моделями aya-expanse:8b та nomic-embed-text;
5. LangChain для реалізації Retrieval-Augmented Generation.

Для розробки використовується мова програмування Python.

На рисунку 3.1 зображено інструменти, що використані для реалізації.

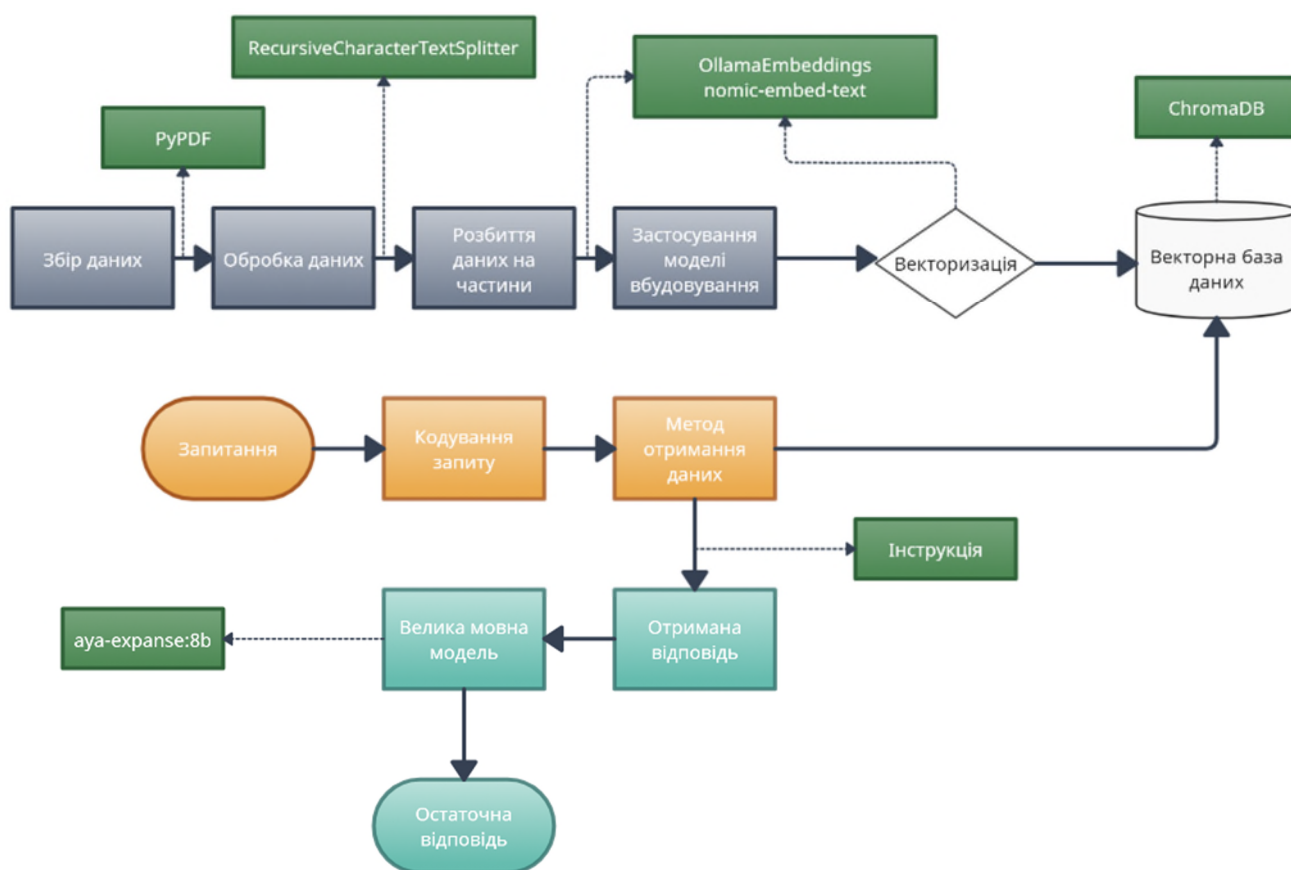


Рисунок 3.1 – Схема роботи методу генерації тексту з інформаційним пошуком з використанням інструментів для реалізації

Технологічна основа платформи базується на сучасних інструментах і підходах, що забезпечують зручність взаємодії, швидкість обробки запитів і точність

відповідей. Для розробки інтерфейсу користувача використано CustomTkinter, бібліотеку для створення графічних інтерфейсів, яка підтримує адаптивний дизайн, включаючи темну тему, та дозволяє реалізувати сучасні елементи управління. Завдяки цьому інтерфейс стає інтуїтивно зрозумілим і зручним для різних категорій користувачів [27].

У ролі бази знань застосовується Chroma – інструмент для роботи з векторними представленнями даних. Ця технологія дозволяє перетворювати текстові документи у векторні формати, що забезпечує швидкий пошук інформації та ефективну інтеграцію з мовними моделями. Таке рішення ідеально підходить для зберігання великого обсягу даних та їхнього використання в аналітичних задачах [28].

Мовні моделі відіграють ключову роль у функціонуванні системи. У платформу інтегровано кілька мовних моделей для забезпечення високої точності та контекстуальності відповідей. Однією з таких моделей є `omic-embed-text`, яка відповідає за ефективне векторне подання тексту. Ця модель використовується для створення точних векторних представлень документів, що допомагає швидко знаходити необхідну інформацію в контексті запитів користувачів. Вона сприяє точному відображенню змісту та покращенню ефективності пошуку, оскільки враховує не лише конкретні ключові слова, а й загальний контекст документа, що підвищує релевантність результатів пошуку [29].

Також в платформу інтегрована модель `aya-expanse:8b`, що є потужною локальною мовною моделлю. `aya-expanse:8b` – це модель, оптимізована для виконання складних запитів та формування детальних і контекстуальних відповідей, з урахуванням попередніх взаємодій із користувачем. Вона має великий обсяг параметрів (8 мільярдів), що дозволяє їй працювати з великими обсягами тексту та виконувати обчислення на високій швидкості, навіть без необхідності постійного підключення до зовнішніх серверів. Така локальна інтеграція моделі допомагає значно знизити затримки у відповіді, підвищуючи тим самим швидкість взаємодії. Основними функціями моделі `aya-expanse:8b` являються - генерація тексту, адаптація до контексту, оптимізація запитів, швидкодія та безпека, інтеграція з RAG. Завдяки

своїм потужним обчислювальним можливостям і високій точності обробки запитів, аya-expanse:8b забезпечує стабільну та ефективну роботу системи, дозволяючи платформі надавати користувачам швидкі, точні та персоналізовані відповіді [30]. Обидві моделі працюють через додаток Ollama, що забезпечує зручне локальне розгортання та ефективне управління моделями без необхідності постійного підключення до зовнішніх серверів. Такий підхід підвищує безпеку та швидкодію системи [31].

Додатково в платформі реалізована технологія Retrieval-Augmented Generation (RAG), яка комбінує пошук інформації у базі знань із генерацією тексту. Це дозволяє забезпечити не лише високу точність відповідей, але й їхню актуальність, оскільки модель використовує релевантну інформацію з бази знань для створення контекстно обґрунтованих текстів [32]. Для реалізації механізму Retrieval-Augmented Generation (RAG) використовується бібліотека LangChain, яка забезпечує зручний спосіб інтеграції пошуку та генерації тексту. LangChain дозволяє ефективно комбінувати мовні моделі з базою знань, забезпечуючи динамічний доступ до релевантної інформації та створення контекстуальних відповідей. Це спрощує процес налаштування та розширення функціоналу платформи [33]. Усі ці компоненти гармонійно інтегровані, створюючи технологічну основу, яка забезпечує ефективну взаємодію користувача з системою та надійне виконання її функцій.

Платформа використовує сучасні технології для обробки даних, що забезпечують ефективність роботи з інформацією та її безпеку. Однією з ключових функцій є підтримка роботи з документами у форматі PDF. Завантажені файли автоматично індексуються та додаються до бази знань, що дозволяє швидко знаходити потрібну інформацію. Для цього застосовується методологія текстового поділу, яка передбачає розбиття документів на логічно структуровані частини. Такий підхід покращує індексацію та прискорює процес пошуку даних [34]. Для роботи з PDF-файлами платформа використовує бібліотеку PyPDF, яка забезпечує зручний інструментарій для зчитування, обробки та аналізу вмісту документів. PyPDF дозволяє коректно працювати з текстом, навіть якщо він має складну структуру,

наприклад, розділи, таблиці чи зображення. Це значно спрощує процес автоматизації індексації PDF-документів [35].

Основою інфраструктури розробки є мова програмування Python, яка вважається однією з найпопулярніших у світі завдяки своїй універсальності, простоті та багатому набору бібліотек. Python забезпечує високу гнучкість у реалізації різноманітних функцій платформи та дозволяє легко інтегрувати її з іншими технологіями. Завдяки широкій спільноті розробників, платформа має доступ до перевірених рішень і може швидко адаптуватися до нових потреб [36].

Особливу увагу приділено безпеці даних. Уся інформація зберігається локально, що значно знижує ризики витоку даних і забезпечує конфіденційність для користувачів. Такий підхід робить платформу особливо актуальною для використання у сферах, де приватність є критично важливою.

Завдяки інтеграції сучасних технологій і ефективній організації інфраструктури, платформа створює надійне середовище для роботи з інформацією, підтримуючи користувачів у вирішенні їхніх задач. Це дозволяє їй бути потужним інструментом, орієнтованим на забезпечення якісної допомоги у комунікації та пошуку даних.

3.2 Опис компонентів системи комунікації

Система має на меті допомогти людям з вадами слуху та мови у спілкуванні шляхом автоматичної генерації відповідей на запити. Вона забезпечує підтримку в обміні інформацією, використовуючи текстові документи, що дає змогу користувачам з різними потребами легко взаємодіяти з іншими. Завдяки вбудованій мовній моделі система здатна розуміти запити і генерувати відповіді, адаптуючи їх до конкретного контексту. На рисунку 3.2 зображено основні модулі та компоненти системи комунікації, які забезпечують ефективне функціонування платформи.

значно підвищує її інформативність і дозволяє системі надавати точніші відповіді. Окрім цього, інтерфейс пропонує інструменти для управління документами. Користувач може видаляти вже завантажені файли або оновлювати базу знань відповідно до змін у даних. Такий комплексний підхід до організації графічного інтерфейсу гарантує, що платформа буде не лише функціональною, але й зручною у використанні, дозволяючи максимально спростити процес роботи з інформацією та забезпечити високу ефективність виконання задач.

Модуль обробки запитів відіграє ключову роль у забезпеченні коректної роботи платформи, гарантуючи правильне трактування даних, які вводять користувачі. Взаємодія починається з прийому текстових запитів, які користувачі вводять через зручний графічний інтерфейс. Цей процес організований таким чином, щоб мінімізувати можливість помилок і забезпечити зрозумілий механізм взаємодії. Одразу після введення тексту система виконує попередню нормалізацію запиту. Це означає, що всі непотрібні символи, такі як зайві пробіли, спеціальні знаки чи інші форматовальні помилки, видаляються або виправляються. Така обробка дозволяє привести запит до єдиного стандарту, що сприяє підвищенню точності подальшого аналізу [38]. Наступним етапом є верифікація даних. Система перевіряє, чи відповідає введений текст певним встановленим правилам, таким як мінімальна довжина, коректна структура або допустимий формат. Ця перевірка допомагає уникнути некоректної інтерпретації запиту, забезпечуючи стабільність роботи платформи. Поєднання цих етапів дозволяє модулю обробки запитів ефективно адаптуватися до різних типів користувацьких даних, забезпечуючи не лише правильну інтерпретацію, але й швидку та надійну обробку запитів для отримання відповідей [39].

База знань слугує ключовим компонентом платформи, забезпечуючи ефективний інформаційний пошук і доступ до даних. Її основна функція полягає у зберіганні та організації завантажених документів, що формують основу для подальшого аналізу та відповіді на запити. Коли користувач додає PDF-файл, система автоматично виконує обробку цього документа. Текст із файлу розбивається на структуровані блоки, що сприяє зручній індексації. Завдяки цьому база знань стає не

просто сховищем інформації, а оптимізованим середовищем для швидкого пошуку даних.

Особливістю цієї бази є використання векторних представлень документів. Векторизація текстових документів відбувається за допомогою сучасних методів машинного навчання, що дозволяють перетворити текст в числові вектори. Це значно покращує швидкість і точність пошуку релевантної інформації. Це означає, що текст кодується у вигляді багатовимірних математичних векторів. Такий підхід дозволяє системі проводити складний пошук не лише за ключовими словами, але й з урахуванням контексту, семантичного значення та схожості даних. Завдяки векторним моделям пошук стає надзвичайно швидким і точним, навіть при великих обсягах інформації. Поєднання автоматичної індексації та векторного кодування забезпечує базі знань високу функціональність і адаптивність до різних типів даних. Вона не лише зберігає інформацію, але й інтегрує її в роботу платформи, дозволяючи швидко та точно відповідати на складні запити користувачів [40].

Модуль інформаційного пошуку є невід'ємною частиною платформи, який забезпечує швидке й ефективне знаходження потрібної інформації. Його робота базується на використанні сучасних алгоритмів, що дозволяють обробляти великі обсяги даних з високою точністю. Основою пошукових механізмів є векторні представлення текстів, які дозволяють аналізувати не лише окремі ключові слова, але й семантичну схожість між запитом користувача та збереженими даними. Такий підхід забезпечує високу релевантність результатів навіть для складних запитів, що потребують врахування контексту.

Особливістю модуля є інтеграція пошуку за ключовими словами. Він поєднує швидкий аналіз текстів із урахуванням історії чату та наявних документів у базі знань. Це дозволяє враховувати контекст взаємодії з користувачем, надаючи результати, які найбільше відповідають його потребам. Завдяки цьому платформа здатна забезпечити не лише точність, але й актуальність знайденої інформації. Інтеграція таких підходів до інформаційного пошуку робить платформу універсальним інструментом для роботи з великими масивами даних, дозволяючи користувачам отримувати швидкі й

обґрунтовані відповіді на свої запити. Система оптимізує пошук не лише за точними ключовими словами, але й враховує контекст запиту користувача. Зокрема, використовуються алгоритми семантичного пошуку, які дозволяють знаходити найбільш релевантні документи, навіть якщо запит не містить точних термінів, присутніх у тексті. Це дозволяє здійснювати пошук, враховуючи значення слів та їх контекст в рамках запиту. Модуль пошуку працює як "розумний навігатор", який ефективно поєднує сучасні технології обробки тексту з потребами користувачів.

Модуль генерації відповідей є центральним елементом платформи, який відповідає за створення текстових результатів, максимально релевантних до запитів користувача. Після генерування кількох варіантів відповіді, система надає користувачу можливість вибрати той, який найбільше відповідає його потребам. Вибір відповіді базується на фактори, як-от емоційний відтінок, точність інформації та відповідність стилю комунікації. Користувач може вибрати з декількох варіантів, що дає йому більше контролю над процесом комунікації. Його робота тісно інтегрована з інформаційним пошуком, що дозволяє йому враховувати як знайдені документи, так і контекст, у якому поставлено запит. Завдяки цьому система формує відповіді, які не лише містять необхідну інформацію, але й відповідають логіці та стилю взаємодії з користувачем.

Важливим аспектом функціонування модуля є можливість генерації кількох варіантів відповідей. Модель створює тексти, що відрізняються змістом, стилем або емоційним забарвленням. Це дозволяє користувачеві обрати той варіант, який найкраще відповідає його потребам або ситуації. Наприклад, одна відповідь може бути лаконічною та нейтральною, тоді як інша – детальнішою, з елементами емоційної підтримки чи уточненнями. Такий підхід забезпечує більшу гнучкість у використанні платформи, роблячи її інструментом, який адаптується до різних потреб користувача. Модуль генерації відповідей, поєднуючи можливості штучного інтелекту з релевантною інформацією з бази знань, гарантує не лише якісні, але й персоналізовані результати, що значно покращує загальний досвід взаємодії з платформою. Після обробки запиту, система використовує велику мовну модель для

генерації відповідей, яка може адаптуватися до конкретних ситуацій. Відповіді генеруються з урахуванням емоційного контексту та інтонації запиту. Модель також враховує історію попередніх запитів користувача, що дозволяє створювати більш персоналізовані та точні відповіді.

Модуль пам'яті чату є важливим елементом, який забезпечує збереження і використання контексту діалогу протягом усієї взаємодії з користувачем. Це дозволяє створювати більш природне та органічне спілкування, оскільки система може опиратися на попередні запити та відповіді для кращого розуміння поточної ситуації. Вся історія запитів і відповідей зберігається в локальній пам'яті, що гарантує швидкий доступ до інформації і дозволяє системі зберігати логіку розмови навіть через кілька етапів взаємодії.

Важливою функцією цього модуля є підтримка послідовності діалогу. Завдяки збереженню контексту попередніх запитів, система здатна генерувати узгоджені відповіді, що враховують не тільки поточний запит, а й інформацію з попередніх частин розмови. Це створює відчуття безперервності діалогу та підвищує точність і доречність відповідей, оскільки вони стають не просто реакцією на одиничний запит, а відображенням загальної логіки взаємодії. Збереження контексту не лише покращує ефективність системи, а й значно підвищує зручність користувачів, дозволяючи їм мати більш зв'язне та інтуїтивно зрозуміле спілкування з платформою. Після вибору найбільш доречного варіанту, система надає остаточну відповідь користувачу. Всі відповіді зберігаються для подальшого використання, що дозволяє удосконалювати модель з часом, враховуючи відгуки користувачів. Всі дані зберігаються відповідно до стандартів безпеки та конфіденційності.

Адміністративні функції системи спрямовані на забезпечення актуальності та організованості даних, що є важливим для безперервної і ефективної роботи платформи. Одним з основних аспектів цієї функціональності є оновлення бази даних, яке включає завантаження нових документів та їх інтеграцію в існуючу інформаційну структуру. При цьому також відбувається перерахунок індексів, що дозволяє підтримувати точність і швидкість пошуку, а також актуалізувати всі зв'язки між

різними елементами інформації в базі. Ще одним важливим елементом адміністративних функцій є очищення пам'яті чату. Цей процес дає можливість користувачам за їхнім бажанням видаляти історію запитів і відповідей, що зберігаються у локальній пам'яті. Така опція є корисною з точки зору безпеки та конфіденційності, дозволяючи користувачам контролювати, які дані зберігаються в системі. Вона також дає змогу зберігати систему в оптимальному стані, не допускаючи накопичення зайвих або застарілих даних. Ці адміністративні функції сприяють не лише покращенню ефективності роботи системи, але й гарантують зручність і безпеку для користувачів, забезпечуючи належну організацію та захист даних. Система має великий потенціал у забезпеченні комунікації для людей з вадами слуху та мови. Вона дозволяє створювати інклюзивне середовище, де кожен може взаємодіяти з іншими без бар'єрів. Завдяки використанню передових технологій, таких як векторизація та великі мовні моделі, система пропонує високу точність та швидкість відповіді. Крім того, врахування емоційного контексту допомагає зробити комунікацію більш природною.

Безпека даних є однією з основних пріоритетів при розробці платформи, і для цього передбачено кілька важливих заходів, які гарантують конфіденційність та захист користувацької інформації. Одним із ключових підходів є локальне зберігання всіх даних. Це означає, що вся інформація, що генерується або обробляється під час взаємодії з платформою, зберігається безпосередньо на пристрої користувача. Такий підхід значно знижує ризик витоків даних, оскільки немає потреби в централізованому сховищі або передачі чутливої інформації через інтернет. Додатково, для підвищення рівня захисту, всі дані шифруються перед збереженням. Шифрування забезпечує надійний захист від несанкціонованого доступу, навіть якщо дані потраплять у чужі руки. Таким чином, зашифровані дані стають непізнаваними без відповідного ключа для розшифрування, що створює додатковий бар'єр для потенційних загроз. Ці заходи спрямовані на забезпечення максимального рівня безпеки, дозволяючи користувачам бути впевненими у тому, що їхні дані захищені на

кожному етапі взаємодії з платформою, незалежно від того, чи йдеться про особисті запити, історію чату чи завантажені документи [41].

Ця структура забезпечує ефективну взаємодію між користувачем і системою, дозволяючи адаптувати платформу до потреб кожного користувача та надавати релевантні та зрозумілі відповіді.

3.3 Опис побутових ситуацій та формування датасету

Система, розроблена для допомоги людям із вадами слуху та мовлення, дозволяє полегшити їхню комунікацію в повсякденних ситуаціях, які можуть виникати в кафе, ресторані, на роботі, у кав'ярні, лікарні чи магазині. Наприклад, якщо співрозмовник запитує про улюблений напій користувача в кав'ярні, програма аналізує інформацію з документа, наданого користувачем, та пропонує кілька відповідей, таких як "Мені подобається капучино" або "Я зазвичай п'ю чай".

У магазині, якщо співрозмовник хоче уточнити, які продукти варто купити, система може запропонувати відповіді, посилаючись на харчові уподобання користувача, вказані у документі: наприклад, "Я віддаю перевагу свіжим овочам" або "Купи, будь ласка, щось із молочного".

У лікарні система може допомогти відповісти на запитання лікаря про історію хвороби або ліки, які користувач приймає, запропонувавши відповіді, як-от "Я приймаю ліки для нормалізації тиску" або "У мене немає алергій".

На роботі, якщо колега питає про поточний стан завдання, система генерує варіанти відповідей, що враховують специфіку роботи користувача, наприклад, "Я закінчую звіт і надішлю його до кінця дня" або "Ми ще обговорюємо це з командою".

У ресторані, коли офіціант цікавиться, чи потрібні додаткові сервіси, система допомагає сформулювати відповідь: "Можна, будь ласка, додатковий набір приборів?" або "Все чудово, дякую!".

Датасет системи складається з історії запитів, контексту запитання, інструкції та завантажених документів, що включають інформацію про користувача. Для

формування персональної інформації про користувача, що виступатиме базою для датасету та використовуватиметься системою для коректних персоналізованих відповідей створено план формування.

План формування включає в себе інформацію про те, що документ має бути сформований таким чином, щоб містити основні дані про особу та її життя. Спочатку зазначається загальна інформація, яка включає повне ім'я, дату та місце народження. Далі йде розділ, що стосується освіти, в якому вказуються навчальні заклади, роки навчання, спеціальності та отримані ступені, а також додаткові курси і сертифікати. Інформація про кар'єру повинна містити посади, організації та періоди роботи. Наступний розділ присвячений ключовим досягненням, де слід вказати професійні здобутки та будь-які визнання, нагороди чи дипломи. Потім описується сім'я та особисте життя, включаючи сімейний стан, інформацію про чоловіка або дружину, кількість та вік дітей, їхні інтереси, а також наявність домашніх улюбленців і хобі. Інформація про здоров'я повинна включати хронічні захворювання, вади та засоби адаптації, алергії, а також інші аспекти, що мають значення. Розділ особистих інтересів відображає улюблені книги, фільми, страви, напої, музику, а також бренди, які подобаються особі. Контактна інформація містить місце проживання, електронну пошту та номер телефону. У додатковій інформації повинні бути відображені активність в громадських або благодійних організаціях та особисті цілі або мрії. Розділ характер описує основні риси характеру, особистісні якості, що впливають на взаємодію з іншими, а також підхід до вирішення складних ситуацій. Крім того, в документі вказуються вподобання в покупках, такі як основні категорії продуктів в продуктовому магазині, медикаменти чи засоби догляду з аптеки, улюблені страви та напої в ресторанах, кафе та кав'ярнях. Останній розділ присвячений побуту, в якому зазначаються улюблені страви для кожного члена сім'ї, а також особливі побутові звички та вподобання.

За допомогою штучного інтелекту за планом формування генерується профіль користувача з інформацією для різних побутових ситуацій. Документ завантажується у систему після чого за інформацією з документу формується базовий датасет. В

результаті, користувач може сподіватися на ефективну і зручну допомогу у комунікації, що максимально враховує його індивідуальні вподобання та обставини.

Висновки до розділу 3

Платформа має модульну архітектуру, що включає фронтенд (для інтерфейсу користувача), бекенд (для обробки запитів і генерації відповідей) і базу даних (для зберігання інформації). Вона використовує сучасні технології, такі як CustomTkinter для інтерфейсу, Chroma для роботи з векторними даними та мовні моделі для точних і контекстуальних відповідей. Інтеграція Retrieval-Augmented Generation (RAG) дозволяє поєднувати пошук і генерацію тексту для високої точності відповідей. Платформа також підтримує безпеку даних, зберігаючи інформацію локально, що важливо для конфіденційності користувачів.

Інтерфейс є простим і інтуїтивно зрозумілим, дозволяючи користувачам швидко вводити запити та взаємодіяти з системою. База знань використовує векторні представлення документів для швидкого та точного пошуку інформації. Модуль генерації відповідей формує кілька варіантів відповідей на запит користувача, враховуючи контекст і емоційну забарвленість. Система також зберігає контекст діалогу, що дозволяє підтримувати безперервне і логічне спілкування.

Система для людей із вадами слуху та мовлення ефективно допомагає в повсякденній комунікації, адаптуючи відповіді до різних ситуацій, таких як в кафе, ресторані, на роботі або в лікарні. Використовуючи персоналізовані дані, система генерує відповідні варіанти реакцій на запитання, враховуючи індивідуальні переваги користувача, що сприяє полегшенню взаємодії.

Розділ 4 Дослідження методу пошуково-доповненої генерації варіантів відповідей

4.1 Опис застосунку для дослідження системи комунікації

Інтерфейс застосунку, наведений на рисунку 4.1, призначений для управління системою комунікації.

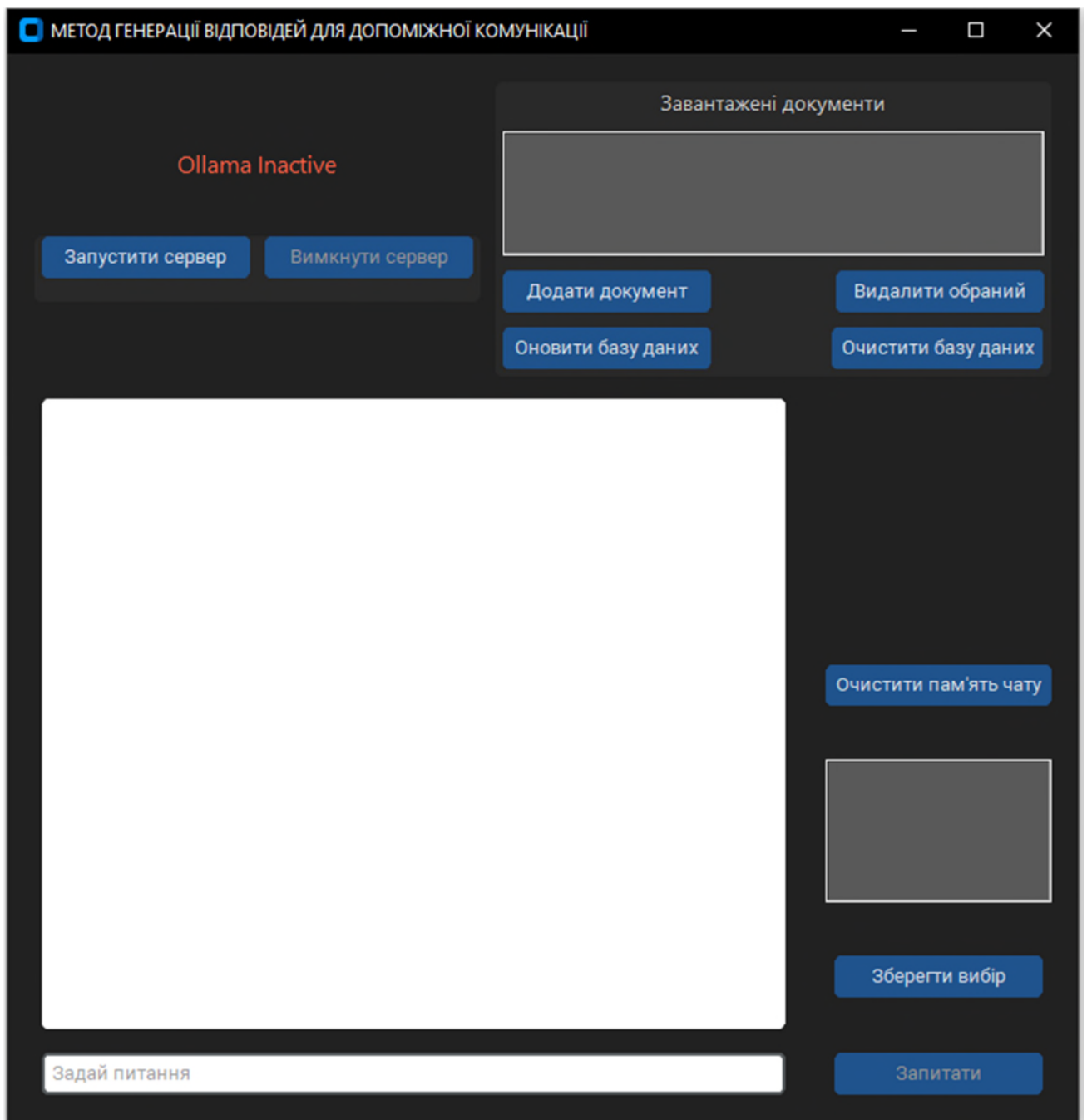


Рисунок 4.1 – Інтерфейс застосунку

Інтерфейс застосунку включає в себе наступні ключові частини:

В лівій верхній частині інтерфейсу розташовано індикатор, що відображає активність сервера. Для запуску сервера достатньо натиснути кнопку «Запустити сервер» та кнопка «Вимкнути сервер», що відповідає за його вимкнення.

В правій верхній частині інтерфейсу розташовано список завантажених документів та ряд кнопок для роботи з документами. Кнопки «Додати документ» та «Видалити обраний» відповідають за додавання та видалення документу з папки. Кнопка «Оновити базу даних» відповідає за додавання, оновлення чи зміну бази даних, де інформація з бази даних розбивається на чанки. Кнопка «Очистити базу даних» видаляє всю інформацію з бази даних.

В лівій нижній частині інтерфейсу розташовано поле в якому відображатимуться запитання, варіанти відповіді, остаточні відповіді, якщо підсумувати, то дане поле буде відображати історію чату. Також нижче розташоване поле для вводу запитання користувачем.

В правій нижній частині інтерфейсу розташована кнопка «Очистити пам'ять чату», що видалить весь чат не тільки в полі відображення, а і в файлів JSON, в який записується вся історія чату. Під даною кнопкою розташований список, в якому відображаються варіанти відповіді, що з'являються в даному списку лише після того, як система згенерує варіанти відповіді, в даному списку обирається варіант відповіді, після чого потрібно натиснути на кнопку «Зберегти вибір» під списком. Цим самим система запам'ятає вибір та передасть його в чат, після чого буде використовувати для наступних відповідей. Останньою кнопкою є «Запитати», що надсилає запит системі, після того, як користувач заповнить поле для вводу запитання.

Система допомагає людині із вадами слуху або мовлення спілкуватися, пропонуючи кілька відповідей на запитання співрозмовника. Наприклад, якщо співрозмовник запитує: "Привіт, що ти любиш пити?", програма, спираючись на інформацію в датасеті, може запропонувати такі варіанти відповідей, які зображено на рисунку 4.2

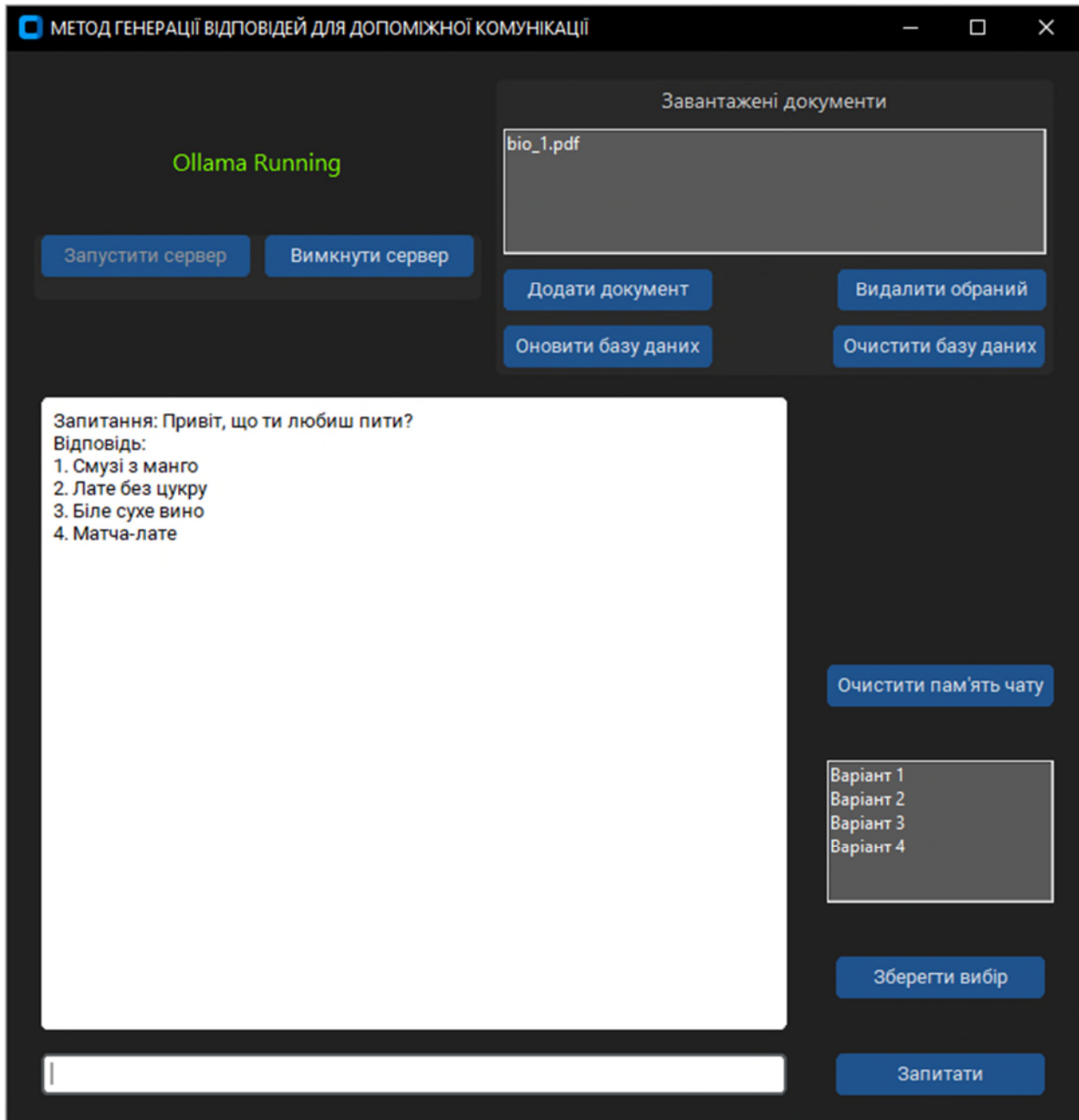


Рисунок 4.2 – Варіанти відповіді на запитання

Кожна відповідь враховує персональні дані користувача, зазначені у документі, а також контекст розмови. Інший приклад: запит "Яке кіно можеш порадити для перегляду?" може викликати відповіді зображені на рисунку 4.3

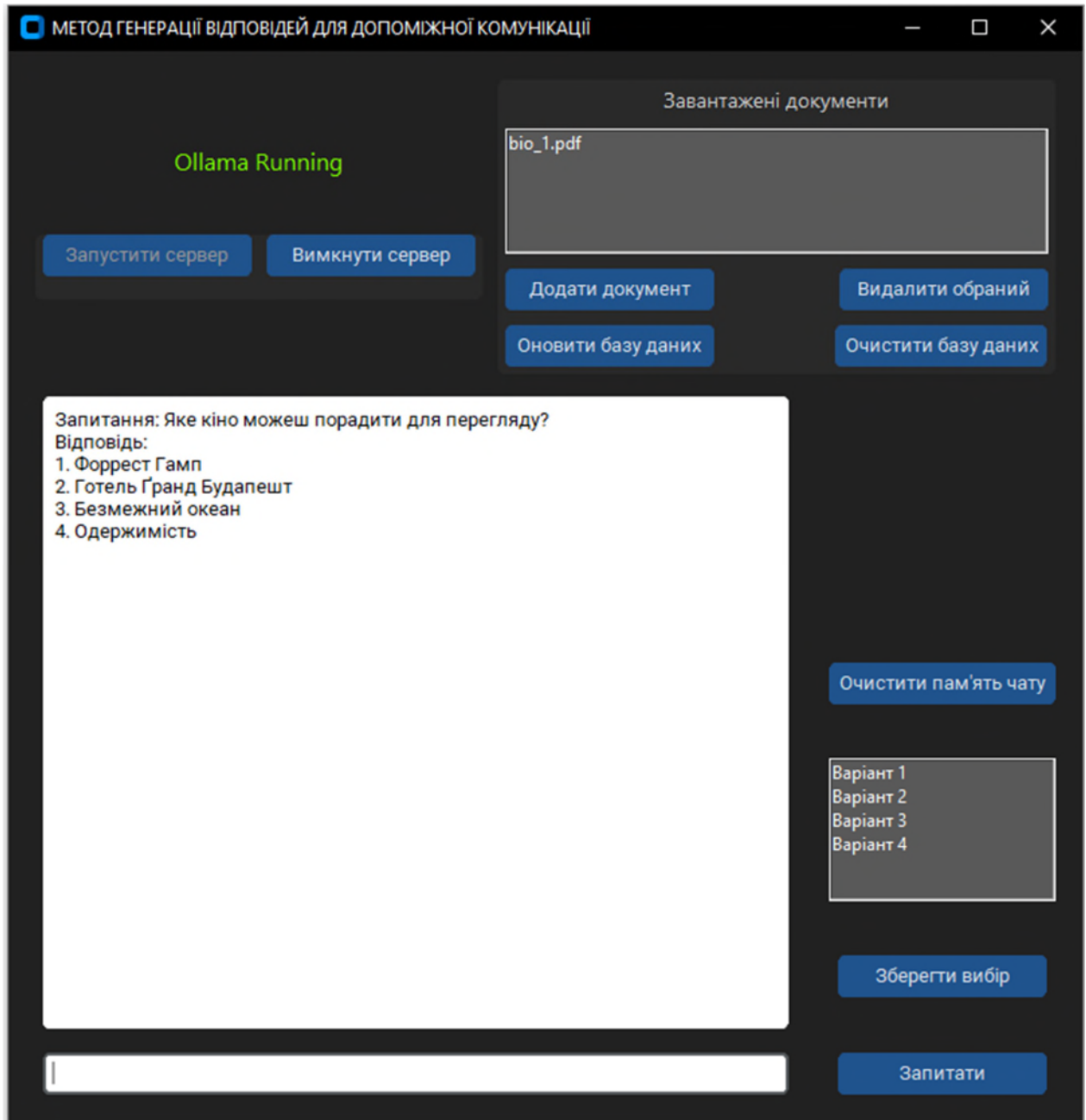


Рисунок 4.3 – Варіанти відповіді на запитання

Система використовує історію чату, контекст запитання та профіль користувача, щоб створити найбільш релевантні варіанти відповіді. Користувач обирає один із варіантів і передає його співрозмовнику, забезпечуючи точність та людяність у спілкуванні. Цей підхід робить програму зручною, адаптивною та орієнтованою на реальні потреби користувачів із комунікаційними бар'єрами.

4.2 Тестування системи комунікації з використанням генерації відповідей з доповнюючим інформаційним пошуком

Для тестування системи комунікації з використанням генерації відповідей з доповнюючим інформаційним пошуком використаємо критерії оцінки, що враховують наступні аспекти:

1. Точність відповідей – критерій оцінює, наскільки правильно система зрозуміла ваше запитання та надала відповідь, що відповідає його змісту та вимогам;
2. Варіативність відповідей – критерій відображає кількість запропонованих варіантів відповідей, їх різноманітність, якість та відповідність запиту користувача;
3. Контекстність відповідей – критерій оцінює здатність системи враховувати не тільки попередні повідомлення у діалозі, але й додаткові джерела інформації, особливо у разі запитів, що є частиною тривалої розмови або потребують специфічних знань про користувача;
4. Значущість відповідей – критерій оцінює, наскільки корисною, релевантною та важливою є отримана відповідь в контексті вашого запиту.

Для тестування системи комунікації будуть створені імітації діалогів на теми «Відвідування ресторану», «Покупки в продуктовому магазині» та «Побут». Імітації діалогів включатимуть конкретні запитання адресовані системі, яка надаватиме варіанти відповіді на запитання для відтворення діалогу. Для порівняння система комунікації буде протестована окремо з використанням методу генерації відповідей з інформаційним пошуком, так і без його використання. Для коректного порівняння запитання під час тестування на однакові теми дублюватимуться – спочатку у варіанті з використанням методу генерації з документом, а потім без нього. Зміст документу описується в розділі формування датасету.

Тестування з використанням документу №1. Розглянемо імітацію діалогу на тему «Відвідування ресторану». Процес діалогу продемонстрований на рисунках, що зображені нижче.

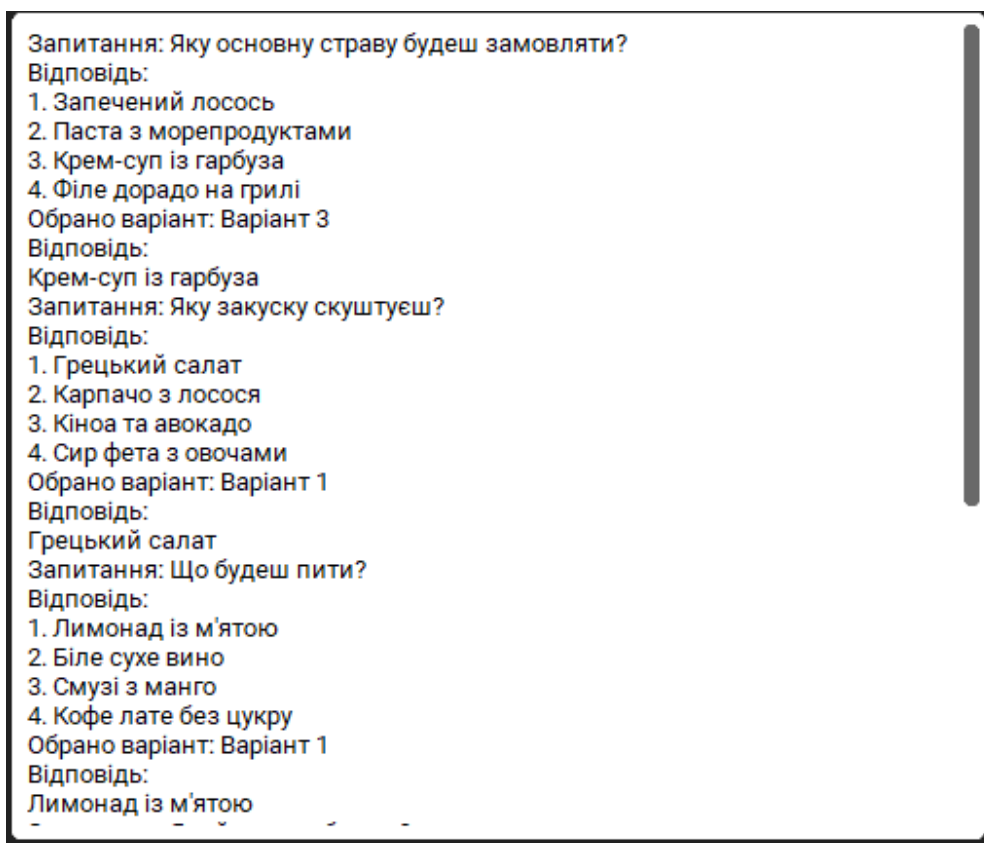


Рисунок 4.4 – Початок діалогу на тему «Відвідування ресторану»

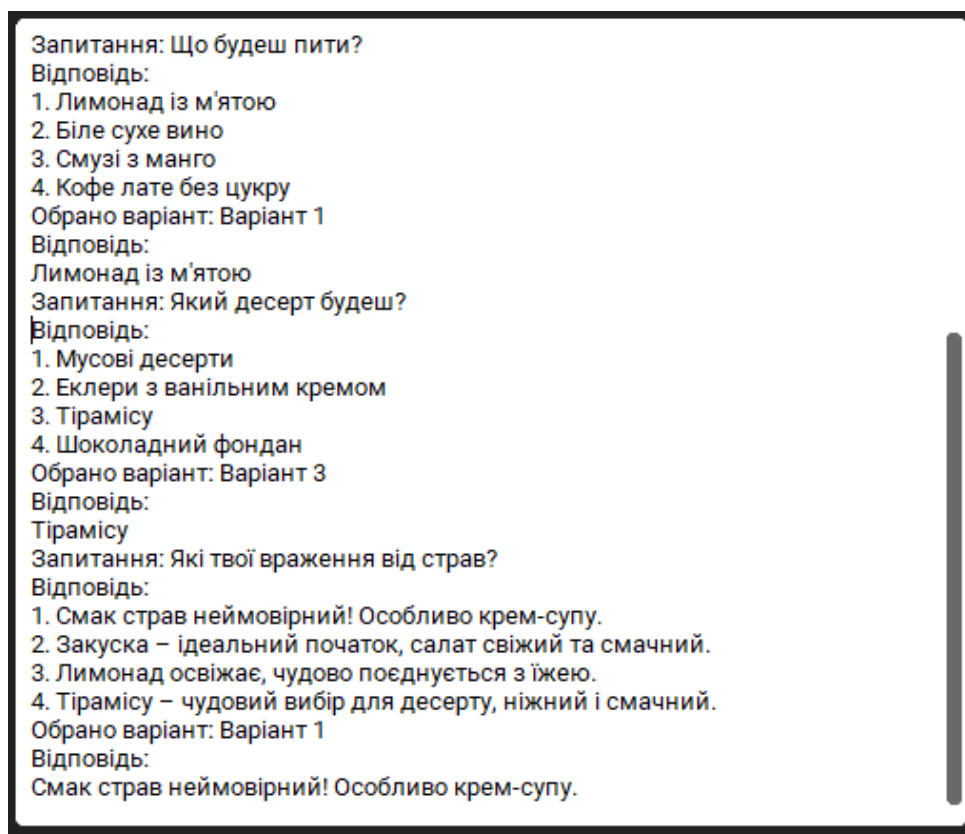


Рисунок 4.5 – Завершення діалогу на тему «Відвідування ресторану»

Оцінювання ефективності роботи методу генерації відповідей за попередньо сформованими критеріями наведено в таблиці.

Таблиця 4.1 – Оцінювання комунікаційної системи на тему «Відвідування ресторану».

Критерії оцінки	Опис отриманого результату	Оцінка
Точність відповідей	Всі відповіді чітко відповідають на запитання, без зайвих уточнень. Вони точно відповідають на запит про страви, напої та враження, і відповідають стилю та темі діалогу.	5/5
Варіативність відповідей	Система надала чотири варіанти відповіді для кожного запитання, які різняться між собою і можуть підходити для різних ситуацій. Варіанти відповідають інтересам та вподобанням людини.	5/5
Контекстність відповідей	Відповіді враховують попередні вибори та інтереси людини, зокрема її улюблені страви та напої.	5/5
Значущість відповідей	Відповіді є релевантними та корисними для вибору страв, напоїв, десертів, з урахуванням смакових вподобань та теми діалогу.	5/5

Тестування на тему «Відвідування ресторану» набрало 20 балів із 20 можливих і вважається прекрасним результатом. Всі варіанти відповіді є точними, різноманітними та релевантними, враховують контекст та особливості вибору людини.

Тестування з використанням документу №2. Розглянемо імітацію діалогу на тему «Покупки у продуктовому магазині». Процес діалогу продемонстрований на рисунках, що зображені нижче.

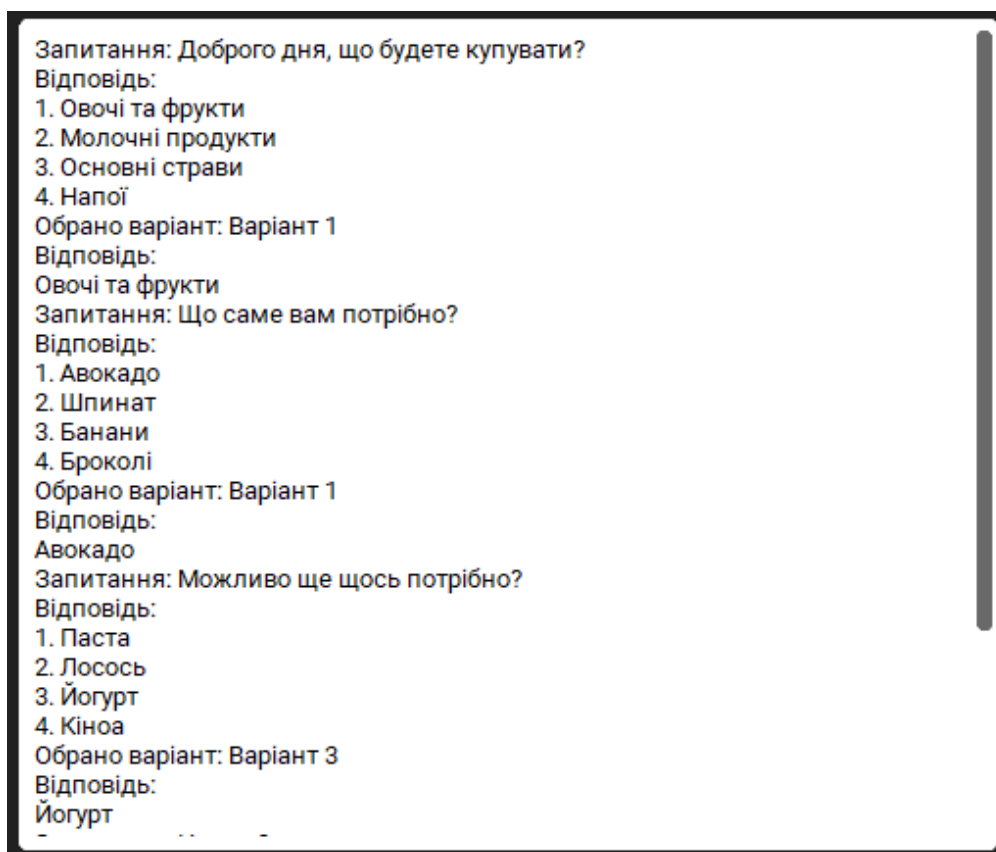


Рисунок 4.6 – Початок діалогу на тему «Покупки в продуктовому магазині»

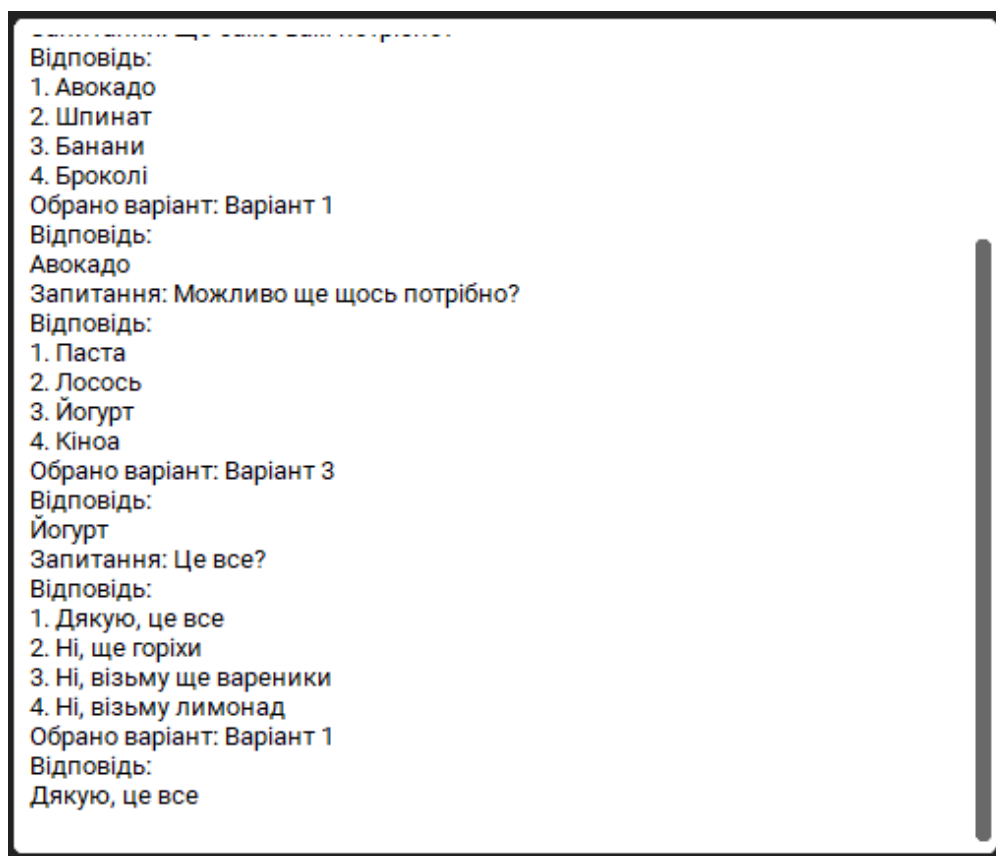


Рисунок 4.7 – Завершення діалогу на тему «Покупки в продуктовому магазині»

Оцінювання ефективності роботи методу генерації відповідей за попередньо сформованими критеріями наведено в таблиці.

Таблиця 4.2 – Оцінювання комунікаційної системи на тему «Покупки в продуктовому магазині».

Критерії оцінки	Опис отриманого результату	Оцінка
Точність відповідей	Відповіді чітко відповідають на запитання, не виходять за межі заданої теми покупок. Зокрема, система надає правильні варіанти щодо вибору продуктів, що підходять до запитань.	5/5
Варіативність відповідей	Варіанти відповіді різноманітні, з кількома можливостями на кожне запитання. Всі варіанти відповідають темі, зберігаючи рівень лаконічності та конкретності.	5/5
Контекстність відповідей	Система використовує попередні варіанти, зокрема те, що людина обирала серед певних продуктів, але варіанти для кожного нового питання не завжди максимально коректно відображають її звички, оскільки в профілі є конкретні вподобання в їжі.	4/5
Значущість відповідей	Відповіді є релевантними та важливими для теми покупки, проте вони могли б бути точнішими з огляду на специфічні вподобання.	4/5

Тестування на тему «Покупки в продуктовому магазині» набрало 18 балів із 20 можливих і вважається гарним результатом. Загалом діалог отримав високу оцінку за точність та варіативність відповідей, але є певні моменти, де можна було б краще врахувати особливості покупки згідно з інформацією про людину.

Тестування з використанням документу №3. Розглянемо імітацію діалогу на тему «Побут». Процес діалогу продемонстрований на рисунках, що зображені нижче.

Запитання: Що готуватимемо на вечерю?
Відповідь:
1. Запечений лосось з кіноа.
2. Грецький салат і паста.
3. Крем-суп та стейк.
4. Українські вареники з сиром.
Обрано варіант: Варіант 1
Відповідь:
Запечений лосось з кіноа.
Запитання: Як щодо страв для дітей?
Відповідь:
1. Домашній борщ з варениками.
2. Грецький салат та сендвічі.
3. Томатна паста.
4. Піца або бургери.
Обрано варіант: Варіант 2
Відповідь:
Грецький салат та сендвічі.
Запитання: Вдома є всі інгредієнти для цих страв?
Відповідь:
1. Так, всі інгредієнти є.
2. Тільки деякі відсутні.
3. Потрібно купити кілька продуктів.
4. Не достатньо овочів.
Обрано варіант: Варіант 4
Відповідь:
Не достатньо овочів.

Рисунок 4.8 – Початок діалогу на тему «Побут»

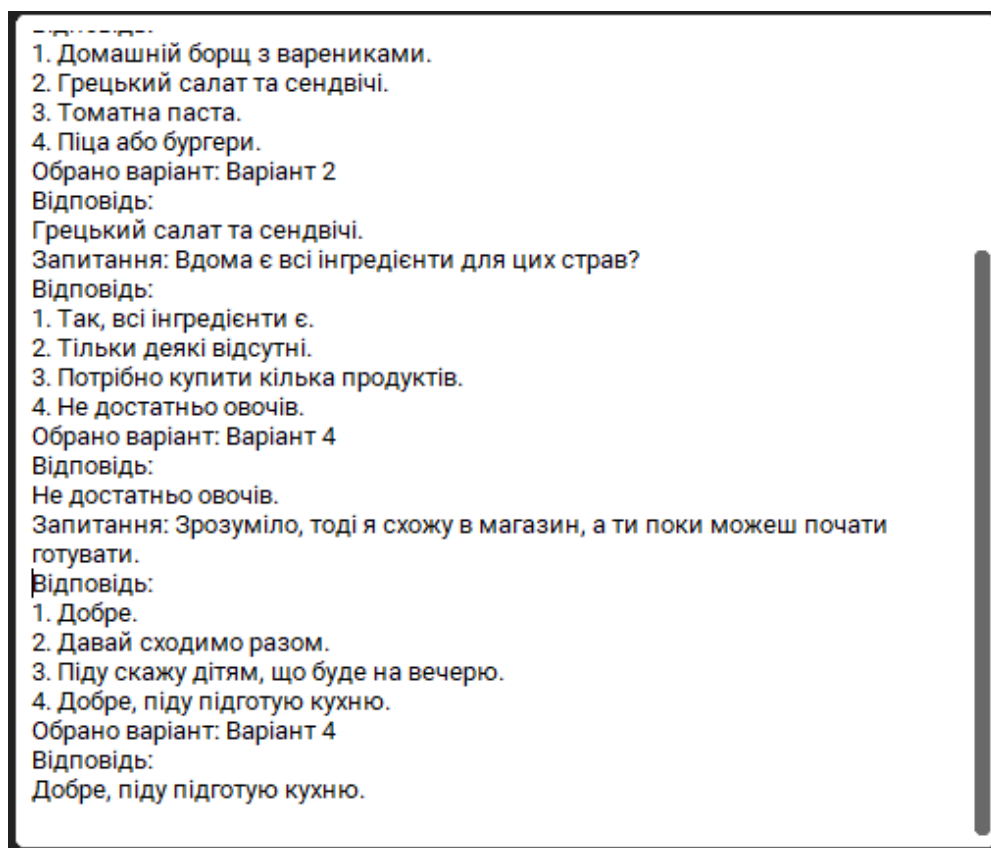


Рисунок 4.9 – Завершення діалогу на тему «Побут»

Оцінювання ефективності роботи методу генерації відповідей за попередньо сформованими критеріями наведено в таблиці.

Таблиця 4.3 – Оцінювання комунікаційної системи на тему «Побут».

Критерії оцінки	Опис отриманого результату	Оцінка
Точність відповідей	Відповіді відповідають на запитання точно та чітко. Пропозиції є релевантними у контексті побутової теми. Вони відповідають реальним уподобанням.	5/5
Варіативність відповідей	Варіанти відповіді різноманітні, з кількома можливостями на кожне запитання. Всі варіанти відповідають темі, зберігаючи рівень лаконічності та конкретності.	5/5

Продовження таблиці 4.3

Контекстність відповідей	Система використовує попередні варіанти, зокрема те, що людина обирала серед певних продуктів, але варіанти для кожного нового питання не завжди максимально коректно відображають її звички, оскільки в профілі є конкретні вподобання в їжі.	4/5
Значущість відповідей	Відповіді є релевантними та важливими для теми покупки, проте вони могли б бути точнішими з огляду на специфічні вподобання.	5/5

Тестування на тему «Побут» набрало 19 балів із 20 можливих і вважається гарним результатом. Діалог добре відображає побутову ситуацію з урахуванням інформації про родину, і відповіді точні, варіативні та релевантні.

Тестування без використання документу №1. Розглянемо імітацію діалогу на тему «Відвідування ресторану». Процес діалогу продемонстрований на рисунках, що зображені нижче.

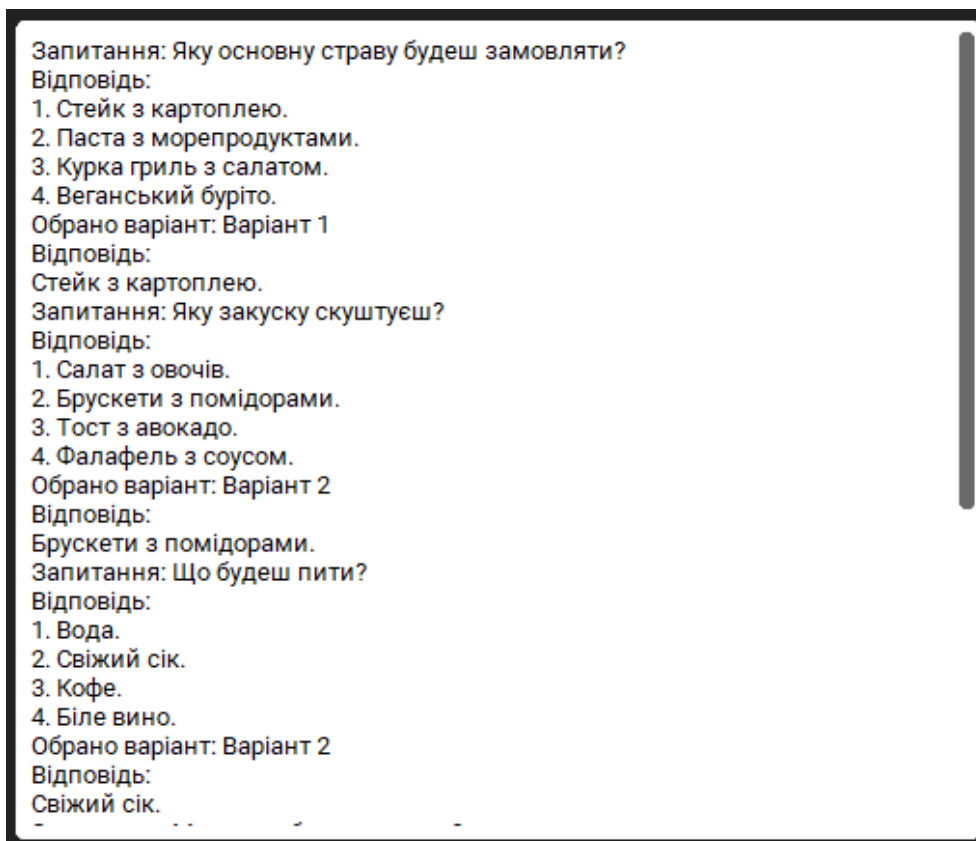


Рисунок 4.10 – Початок діалогу на тему «Відвідування ресторану»

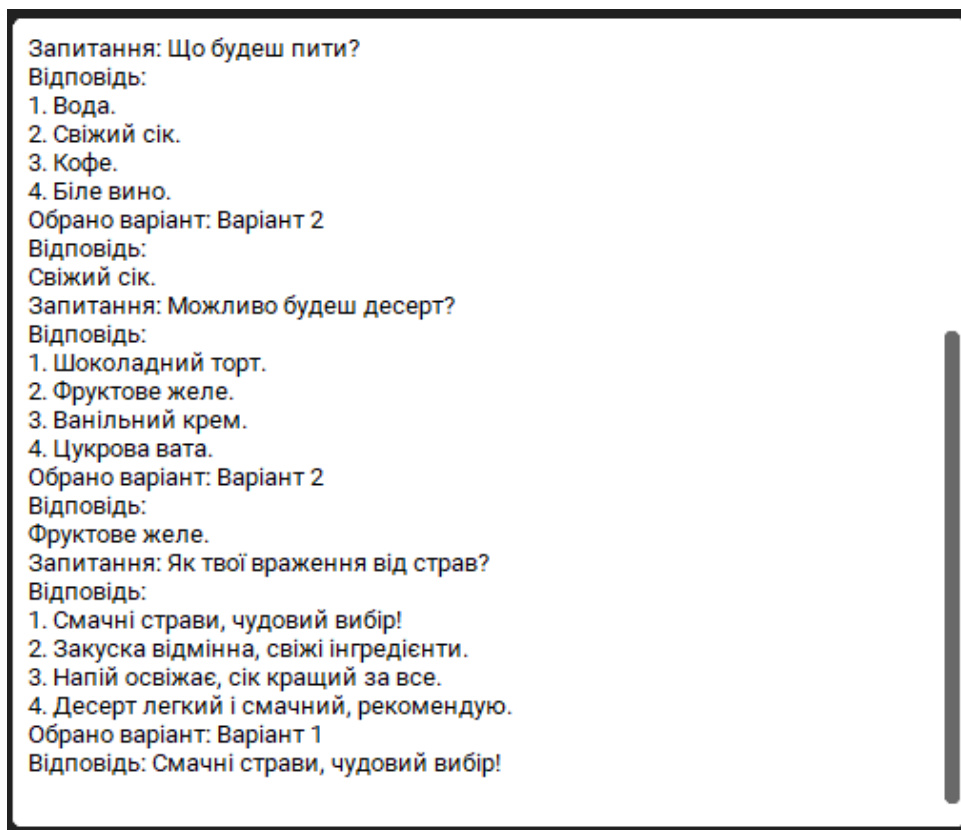


Рисунок 4.11 – Завершення діалогу на тему «Відвідування ресторану»

Оцінювання ефективності роботи методу генерації відповідей за попередньо сформованими критеріями наведено в таблиці.

Таблиця 4.4 – Оцінювання комунікаційної системи на тему «Відвідування ресторану».

Критерії оцінки	Опис отриманого результату	Оцінка
Точність відповідей	Відповіді чітко відповідають на запитання, без зайвих уточнень. Вони точно відповідають на запитання про страви та напої, що є в контексті відвідування ресторану.	5/5
Варіативність відповідей	Відповіді мають достатню різноманітність: кілька варіантів на кожне питання, що дозволяє людині обирати відповідь, яка найкраще підходить.	5/5
Контекстність відповідей	Оскільки в діалозі немає інформації про людину (контекст відсутній), відповідей, які враховують індивідуальні особливості, немає.	1/5
Значущість відповідей	Всі відповіді релевантні до запитів, і відповідають на запитання в рамках відвідування ресторану. Кожна відповідь коротка та точно відповідає на запитання.	5/5

Тестування на тему «Відвідування ресторану» набрало 16 балів із 20 можливих і вважається добрим результатом. Система надає точні, варіативні та значущі відповіді, але відсутність контексту обмежує персоналізацію.

Тестування без використання документу №2. Розглянемо імітацію діалогу на тему «Покупки у продуктовому магазині». Процес діалогу продемонстрований на рисунках, що зображені нижче.

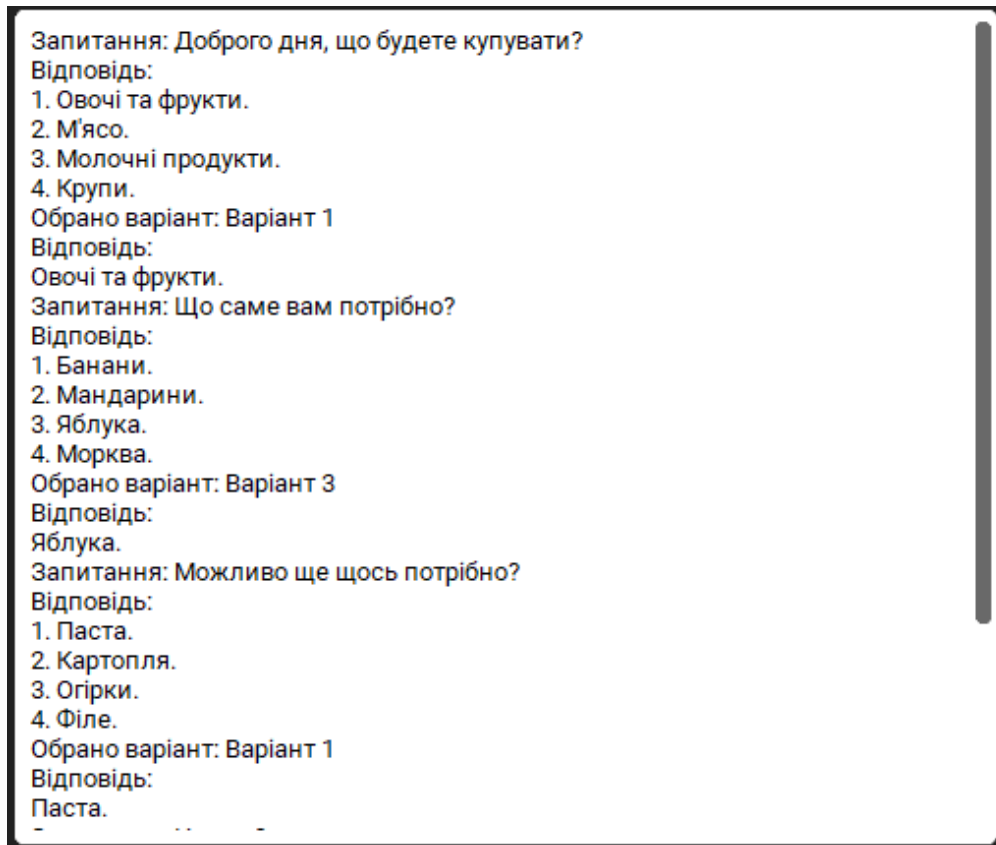


Рисунок 4.12 – Початок діалогу на тему «Покупки в продуктовому магазині»

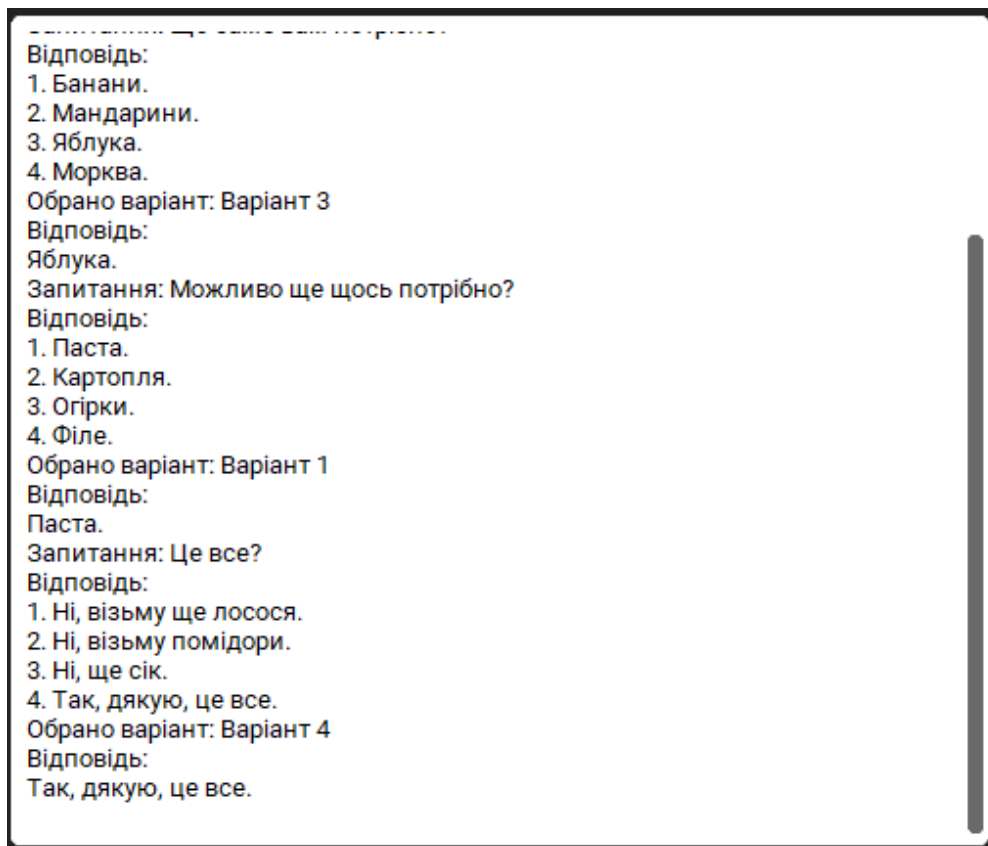


Рисунок 4.13 – Завершення діалогу на тему «Покупки в продуктовому магазині»

Оцінювання ефективності роботи методу генерації відповідей за попередньо сформованими критеріями наведено в таблиці.

Таблиця 4.5 – Оцінювання комунікаційної системи на тему «Покупки в продуктовому магазині».

Критерії оцінки	Опис отриманого результату	Оцінка
Точність відповідей	Відповіді чітко відповідають на запитання, не відходять від теми покупок у магазині. Вони коректно визначають предмети, які людина може купувати.	5/5
Варіативність відповідей	Відповіді мають достатню різноманітність, адже пропонуються варіанти на кожне запитання, що дозволяє людині вибирати найбільш підходящий варіант для себе.	4/5
Контекстність відповідей	Враховуючи, що немає контексту про людину, система не може персоналізувати відповіді. Але загалом відповіді підтримують логіку діалогу.	1/5
Значущість відповідей	Відповіді є чіткими і корисними для діалогу, вони актуальні до запитань, що задаються, і відповідають на кожне з них без зайвих уточнень.	4/5

Тестування на тему «Покупки в продуктовому магазині» набрало 14 балів із 20 можливих і вважається добрим результатом. Загалом система добре виконує завдання в межах доступної інформації, однак для поліпшення результату варто додати можливість персоналізації відповідей, враховуючи контекст людини.

Тестування без використанням документу №3. Розглянемо імітацію діалогу на тему «Побут». Процес діалогу продемонстрований на рисунках, що зображені нижче.

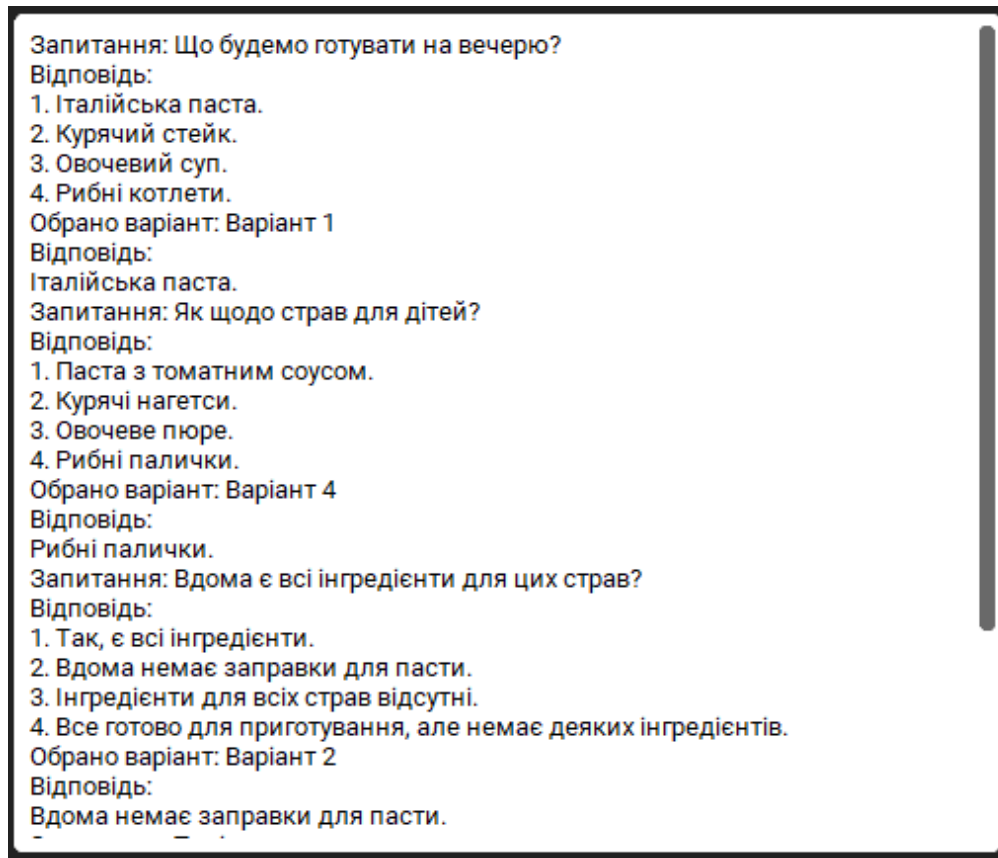


Рисунок 4.14 – Початок діалогу на тему «Побут»

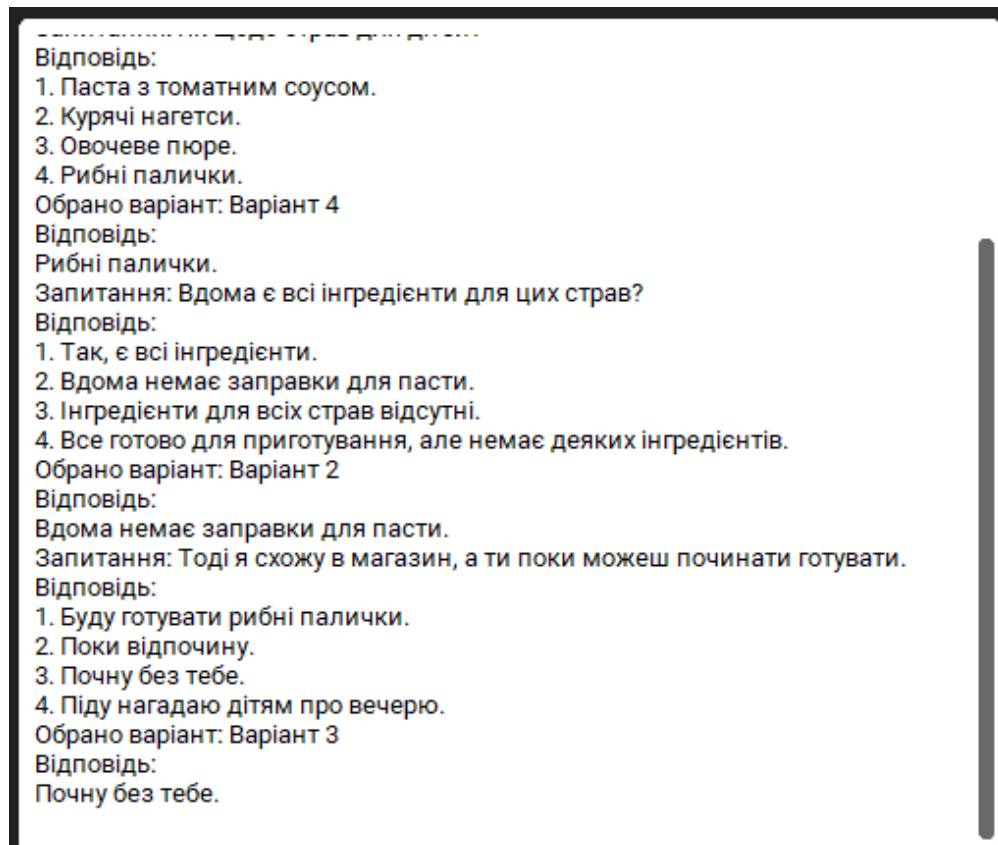


Рисунок 4.15 – Завершення діалогу на тему «Побут»

Оцінювання ефективності роботи методу генерації відповідей за попередньо сформованими критеріями наведено в таблиці.

Таблиця 4.6 – Оцінювання комунікаційної системи на тему «Побут».

Критерії оцінки	Опис отриманого результату	Оцінка
Точність відповідей	Відповіді чітко відповідають на запитання, не відходять від основної теми побутового діалогу. Вони коректно визначають варіанти страв та дій, пов'язаних з готуванням.	5/5
Варіативність відповідей	Відповіді мають достатню різноманітність, пропонуються різні варіанти страв для дорослих і дітей, варіанти дій у відповідь на питання. Це надає вибір для людини.	5/5
Контекстність відповідей	Враховуючи, що немає контексту про людину, система не може персоналізувати відповіді. Проте відповіді логічно підтримують хід діалогу, відповідаючи на актуальні питання.	1/5
Значущість відповідей	Відповіді є корисними та релевантними для діалогу, вони відповідають на кожне запитання і сприяють розвитку подій у побутовому контексті.	4/5

Тестування на тему «Побут» набрало 15 балів із 20 можливих і вважається добрим результатом. Система надає точні, варіативні та значущі відповіді, проте відсутність контексту про людину знижує її персоналізацію.

Таким чином система пройшла тестування, як з використанням інформації про людину з попередньо завантаженого документу, так і без нього. Тести імітували ситуації, в яких людина може опинитись кожного дня.

4.3 Визначення ефективності системи комунікації

Ефективність системи комунікації – це міра відповідності між запланованими цілями та фактичними результатами, отриманими під час тестування за встановленими критеріями. Вона відображає, наскільки успішно система задовольняє потреби користувачів та досягає поставлених завдань у визначених умовах використання.

Після проведення ряду тестувань, проведено дослідження для аналізу ефективності системи комунікації. На рисунку 4.16 продемонстрована статистика балів кожного тестування з використанням доповнюючого пошуку.

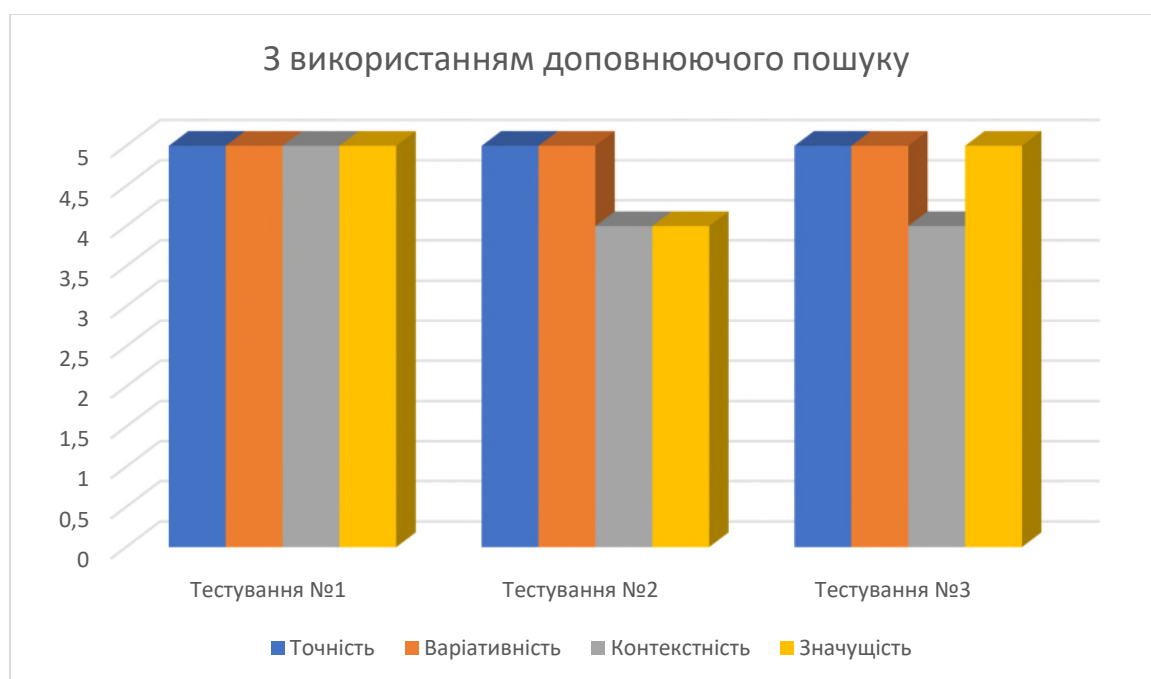


Рисунок 4.16 – Діаграма тестування з використанням доповнюючого пошуку

На рисунку 4.17 продемонстрована статистика балів кожного тестування без використання доповнюючого пошуку.

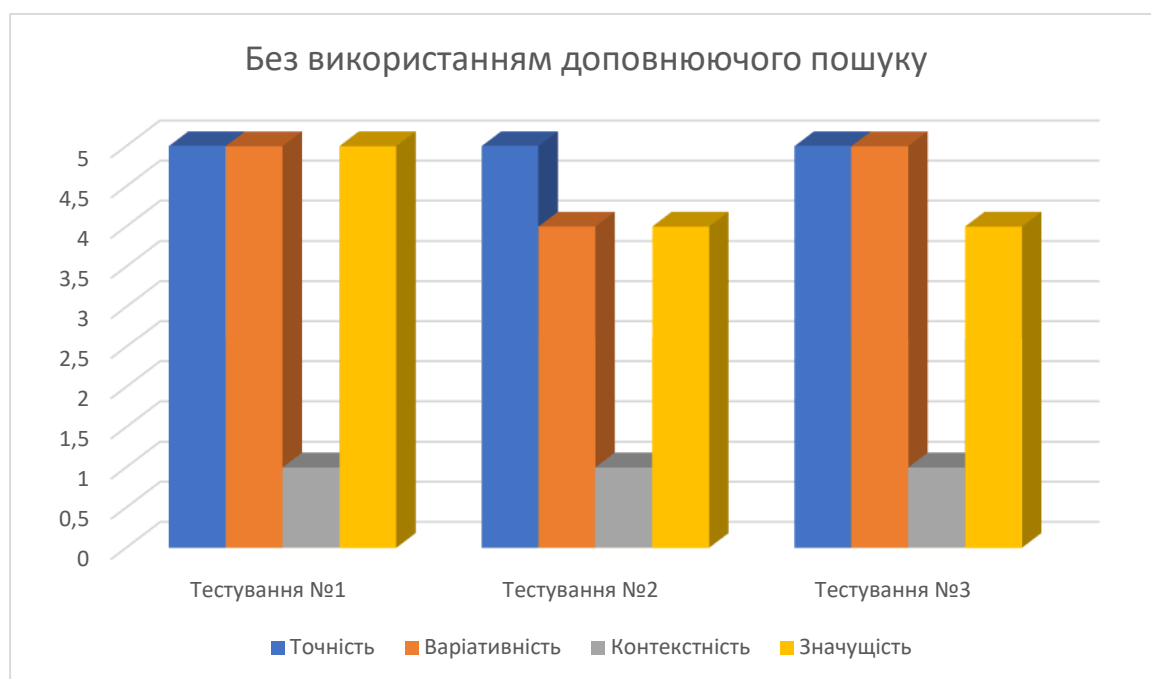


Рисунок 4.17 – Діаграма тестування без використання доповнюючого пошуку

Після проходження тестування з та без використання доповнюючого пошуку, система комунікації отримала 57 балів із 60 можливих та 45 балів із 60 можливих відповідно.

Успішність тестувань можна оцінити за наведеними результатами:

1. Тестування на тему «Відвідування ресторану» з використанням доповнюючого пошуку – 100%;
2. Тестування на тему «Покупки у продуктовому магазині» з використанням доповнюючого пошуку – 90%;
3. Тестування на тему «Побут» з використанням доповнюючого пошуку – 95%;
4. Тестування на тему «Відвідування ресторану» без використання доповнюючого пошуку – 80%;
5. Тестування на тему «Покупки у продуктовому магазині» без використання доповнюючого пошуку – 70%;
6. Тестування на тему «Побут» без використання доповнюючого пошуку – 75%.

Середнє значення ефективності системи комунікації з використанням доповнюючого пошуку сягає 95% із 100% можливих – це прийнятний результат для системи комунікації, коли без використання доповнюючого пошуку лише – 75% із 100% можливих. В основному система комунікації проявила себе досить гарно у всіх тестуваннях, хоча іноді були і недоліки. Можна сказати, що система успішно впоралась із своїми задачами.

На рисунку 4.18 представлено діаграму, що демонструє статистику по кожному з критеріїв, заданих для оцінки системи комунікації.

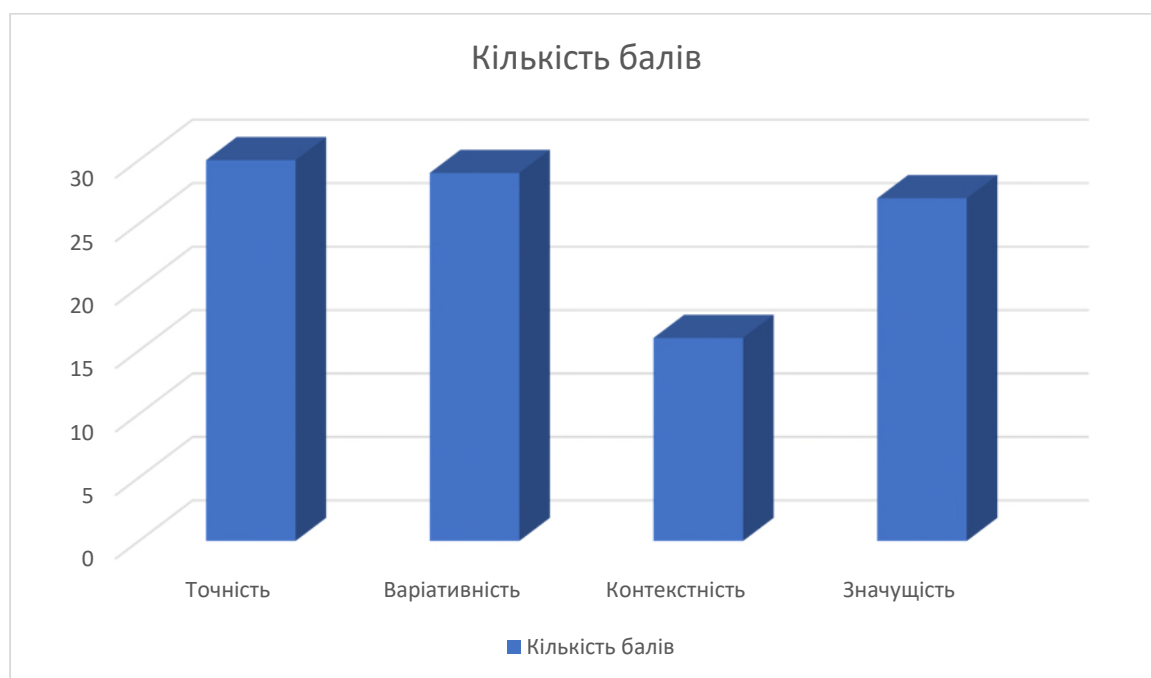


Рисунок 4.18 – Діаграма статистики кожного з критеріїв оцінки

Після проходження тестування з та без використання доповнюючого пошуку, система комунікації отримувала оцінку по кожному з критеріїв.

Успішність критеріїв за результатами тестувань можна оцінити нижче:

1. Точність відповідей – 100%;
2. Варіативність відповідей – 96,67%;
3. Контекстність відповідей – 53,33%;
4. Значущість відповідей – 90%.

Середня оцінка критеріїв системи комунікації сягає 85% із 100% можливих – це прийнятний результат для системи комунікації. Найгірший результат показав критерій – «Контекстність відповідей», його результат 53,33%, даний критерій визначає наскільки відповіді відповідають контексту, так як проводились дві групи тестувань з та без використання доповнюючого пошуку, коли доповнюючий пошук не використовувався контекст отримував оцінку в один бал, адже система надавала відповіді, що враховували історію чату, але самого контексту не було, тобто надати відповіді, що змогли б задовільнити користувача системі надзвичайно важко.

В результаті тестувань система комунікації з використанням генерації відповідей та доповнюючого інформаційного пошуку для допоміжної комунікації продемонструвала високий рівень ефективності. Оцінка за критеріями системи також підтвердила її високу ефективність. Загалом, система успішно виконала поставлені завдання, хоча й потребує подальшого вдосконалення.

Висновки до розділу 4

Інтерфейс запропонованого застосунку для людей із вадами слуху чи мовлення спрощує комунікацію, надаючи кілька варіантів відповідей на запитання співрозмовників. Система адаптується до особистих даних користувача і контексту розмови, що дозволяє пропонувати найбільш релевантні та персоналізовані варіанти відповідей. Такий підхід забезпечує точність, людяність і зручність у спілкуванні, створюючи ефективний інструмент для користувачів з комунікаційними бар'єрами.

Проведено тестування системи комунікації за наступними критеріями: точність відповідей, варіативність відповідей, контекстність відповідей, значущість відповідей. Було проведено по три тестування з використанням документу з описом людини, та без використання документу. Тестування були проведені на теми «Відвідування ресторану», «Покупки у продуктовому магазині», «Побут».

Як результат дослідження система комунікації отримала 57 балів із 60 можливих з використанням інформації про людину та 45 із 60 можливих без

використання, що складає 95% та 75% відповідно, що в середньому складає 85%. Найкращі результати система продемонструвала на тему «Відвідування ресторану» з використанням документу з інформацією про людину – 100%, а найгірші – на тему «Покупки у продуктовому магазині» без використання документу з інформацією про людину – 70%. Найкращі результати система продемонструвала за критерієм «Точність відповідей» - 100%, а найгірші за критерієм – «Контекстність відповідей» - 53,33%, даний критерій потрапив у скрутне становище через проведення тестів без використання документу з інформацією про людину, це сильно зменшило оцінки за критерієм.

Загальні висновки

Кваліфікаційна робота магістра розв'язує науково-технічну задачу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації. За виконання роботи були поставлені й вирішені наступні задачі:

1. Проведено огляд сучасних підходів до генерації відповідей з використанням мовних моделей та технологій доповнюючого інформаційного пошуку (RAG).
2. Розроблено метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації.
3. Підготовлено релевантні текстові дані для покращення точності генерації відповідей з доповнюючим інформаційним пошуком.
4. Розроблено програмну реалізацію системи комунікації між людьми з використанням генерації відповідей з доповнюючим інформаційним пошуком.
5. Проведено тестування взаємодії системи комунікації з користувачами та проаналізувати коректність його відповідей.

Проведено тестування розробленого методу за допомогою імітацій діалогів на теми «Відвідування ресторану», «Покупки в продуктовому магазині» та «Побут». Тестування проводилися з використанням персоналізованої інформації про користувача та без неї. Для діалогу «Відвідування ресторану» успішність з та без інформації становить – 100% та 80% відповідно, для «Покупки в продуктовому магазині» успішність з та без інформації становить – 90% та 70% відповідно, «Побут» успішність з та без інформації становить – 95% та 75% відповідно.

Запропонований метод має значний потенціал для підвищення ефективності допоміжної комунікації для осіб із вадами мовлення або слуху. Його практичне застосування сприяє автоматизації комунікації у різних життєвих ситуаціях, що допомагає користувачам краще інтегруватися у суспільство. Завдяки доповнюючому пошуку система дозволяє персоналізувати відповіді, враховуючи індивідуальні

особливості та контекст діалогу. Крім того, метод оптимізує процес комунікації, забезпечуючи точність та різноманітність відповідей.

Наукова новизна роботи полягає у поєднанні генеративних мовних моделей із доповнюючим інформаційним пошуком (RAG) для вирішення завдань допоміжної комунікації. У межах дослідження була розроблена архітектура методу, що використовує індексацію даних, пошукові алгоритми та великі мовні моделі. Генерація відповідей була вдосконалена завдяки інтеграції зовнішнього інформаційного пошуку, що забезпечує вищий рівень контекстності та значущості відповідей. Також технологія адаптована до персоналізованих потреб користувачів, що робить цей метод ефективним для широкого спектра побутових завдань.

Отримані результати підтверджують перспективність методу для подальшого розвитку технологій допоміжної комунікації та його впровадження у практичні системи.

Перелік посилань

1. Жестова мова [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Жестова_мова
2. Вади слуху [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Вади_слуху
3. Види і причини мовленнєвих порушень [Електронний ресурс]. – Режим доступу: <https://tirc.do.zt.ua/vydy-i-prychyny-movlennevyh-porushen/>
4. Prakhar Gupta, Harsh Jhamtani, and Jeffrey Bigham : Target-Guided Dialogue Response Generation Using Commonsense and Data Augmentation. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 1301–1317, Seattle, United States. Association for Computational Linguistics. (2022) URL: <https://aclanthology.org/2022.findings-naacl.97/>
5. Balda, E.R., Behboodi, A., Mathar, R. : Adversarial Examples in Deep Neural Networks: An Overview. In: Pedrycz, W., Chen, SM. (eds) Deep Learning: Algorithms and Applications. Studies in Computational Intelligence, vol 865. Springer, Cham. (2020) https://doi.org/10.1007/978-3-030-31760-7_2
6. 13 Emerging Trends in Information Technology for 2023 [Електронний ресурс]. – Режим доступу: <https://connect.comptia.org/blog/emerging-trends-in-information-technology>
7. What Are AAC Devices? The Ultimate Guide on Pros, Cons, Types and Costs [Електронний ресурс]. – Режим доступу: <https://i-autism.com/what-are-aac-devices/>
8. The evolution of communications in the last 25 years [Електронний ресурс]. – Режим доступу: <https://medium.com/@GemmaRubio/the-last-25-years-of-communications-evolution-ebbc487eb4d0>
9. Велика мовна модель [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Велика_мовна_модель
10. Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, Mi Zhang :

Efficient Large Language Models: A Survey (2024) URL: <https://openreview.net/forum?id=bsCCJHbO8A>

11. 7 Large Language Model (LLM) Use Cases and Applications: How AI is Reshaping Industries [Электронный ресурс]. – Режим доступа: <https://collabnix.com/7-top-large-language-model-llm-use-cases-and-applications-how-ai-is-reshaping-industries/>

12. What Is Retrieval-Augmented Generation (RAG)? [Электронный ресурс]. – Режим доступа: <https://www.oracle.com/artificial-intelligence/generative-ai/retrieval-augmented-generation-rag/>

13. Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela : Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (2020) URL: <https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf>

14. Asrifa Sultana, Shivani Valecha : A Class of Quasi-Variational Inequalities with Unbounded Constraint Maps: Existence Results and Applications (2023) URL: <https://arxiv.org/abs/2301.05482>

15. Xiaoyu Cheng : Robust Data-Driven Decisions Under Model Uncertainty (2022) URL: <https://arxiv.org/abs/2205.04573>

16. Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, Haofen Wang : Retrieval-Augmented Generation for Large Language Models: A Survey (2023) URL: <https://arxiv.org/abs/2312.10997>

17. Proloquo2Go [Электронный ресурс]. – Режим доступа: <https://www.assistiveware.com/products/proloquo2go>

18. Speech Assistant AAC App [Электронный ресурс]. – Режим доступа: <https://speechassistant.app/>

19. What are Vector Embeddings? - Revolutionize Your Search Experience [Электронный ресурс]. – Режим доступа: <https://qdrant.tech/articles/what-are-embeddings/>

20. What Are Vector Embeddings? An Intuitive Explanation [Электронный ресурс]. – Режим доступа: <https://www.datacamp.com/blog/vector-embedding>
21. Decoding Vector Embeddings: Empowering AI with Data Representation [Электронный ресурс]. – Режим доступа: <https://phaneendragn.medium.com/decoding-vector-embeddings-empowering-ai-with-data-representation-27a92a26dbff>
22. What Are Vector Embeddings? Everything You Need to Know [Электронный ресурс]. – Режим доступа: <https://www.intersystems.com/resources/what-are-vector-embeddings-everything-you-need-to-know/>
23. What is retrieval-augmented generation? [Электронный ресурс]. – Режим доступа: <https://research.ibm.com/blog/retrieval-augmented-generation-RAG>
24. Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, Aman Chadha : A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications (2024) URL: <https://arxiv.org/html/2402.07927v1>
25. Prompt Engineering vs Fine-tuning vs RAG [Электронный ресурс]. – Режим доступа: <https://myscale.com/blog/prompt-engineering-vs-finetuning-vs-rag/>
26. What is an AI prompt? [Электронный ресурс]. – Режим доступа: <https://www.techtarget.com/searchenterpriseai/definition/AI-prompt>
27. CustomTkinter: A modern and easy-to-use GUI package for Python. [Электронный ресурс]. – Режим доступа: <https://customtkinter.tomschimansky.com>
28. Chroma: The simplest way to build search systems. [Электронный ресурс]. – Режим доступа: <https://www.trychroma.com/>
29. Nomic Embed Text v1 [Электронный ресурс]. – Режим доступа: <https://www.nomic.ai/blog/posts/nomic-embed-text-v1>
30. AYA Expanse: Connecting our world. [Электронный ресурс]. – Режим доступа: <https://cohere.com/blog/aya-expanse-connecting-our-world>
31. Ollama [Электронный ресурс]. – Режим доступа: <https://www.ollama.com>
32. What Is Retrieval-Augmented Generation (RAG)? [Электронный ресурс]. – Режим доступа: <https://www.oracle.com/artificial-intelligence/generative-ai/retrieval-augmented-generation-rag/>

33. Langchain: Building and deploying large language models. [Електронний ресурс]. – Режим доступу: <https://www.langchain.com>
34. RecursiveCharacterTextSplitter [Електронний ресурс]. – Режим доступу: https://python.langchain.com/v0.2/api_reference/text_splitters/character/langchain_text_splitters.character.RecursiveCharacterTextSplitter.html
35. PyPDF Documentation [Електронний ресурс]. – Режим доступу: <https://pypdf.readthedocs.io/en/stable/>
36. Python [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Python>
37. Досвід користувача – User Experience [Електронний ресурс]. – Режим доступу: <https://wezom.com.ua/ua/blog/user-experience>
38. Що таке обробка природної мови (Natural Language Processing, NLP)? [Електронний ресурс]. – Режим доступу: <https://thetransmitted.com/adlucem/shho-take-obrobka-prirodnoyi-movu-natural-language-processing-nlp/>
39. What is Data Verification? [Електронний ресурс]. – Режим доступу: <https://knowledge.melissa.com/en-gb/what-is-data-verification>
40. Вкладання слів [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Вкладання_слів
41. What is encryption? [Електронний ресурс]. – Режим доступу: <https://www.cloudflare.com/learning/ssl/what-is-encryption/>

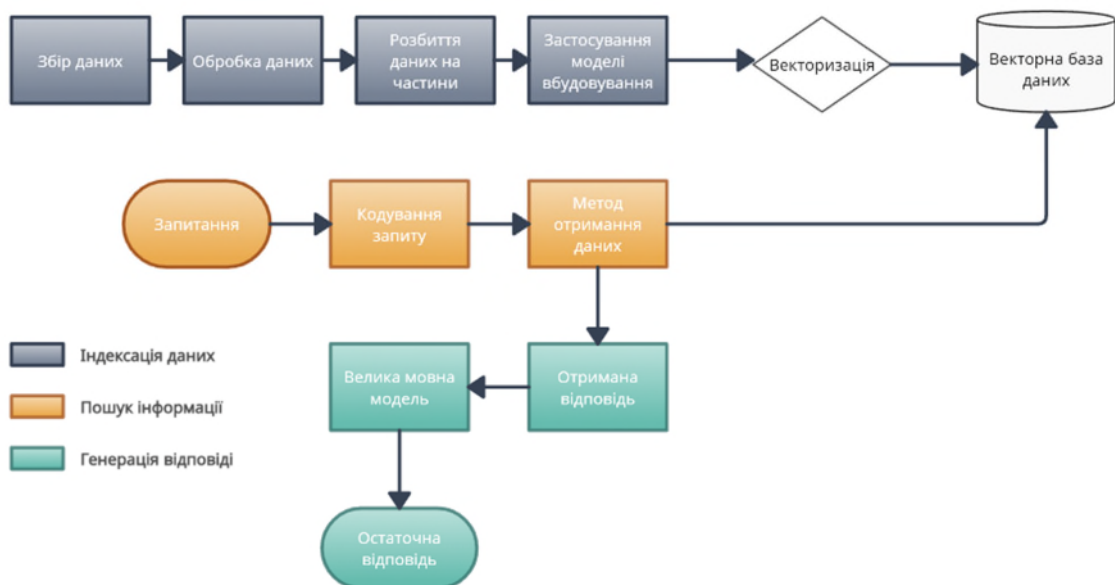
ДОДАТКИ

Додаток А

Схема методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації



Схема роботи методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації



Додаток Б

Світлини наукових публікацій, виконаних при роботі над кваліфікаційною роботою магістра

Перелік наукових публікацій:

Збірник наукових праць – Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації / Медведчук В.Ю., Багрій Р.О., Скрипник Т.К. // Збірник наукових праць за матеріалами Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук – 2024» Хмельницький, 2024.

Науковий журнал – Також, результати роботи опубліковані у науковому журналі – Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації / Медведчук В.Ю., Багрій Р.О., Скрипник Т.К., Мазурець О.В., Монастирська Д.С. // Науковий журнал «Вісник Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2024.

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XVI Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2024»

15-16 листопада 2024

Хмельницький 2024

УДК 004.4

Медведчук В.Ю., Багрій Р.О., Скрипник Т.К.

*Хмельницький національний університет***МЕТОД ГЕНЕРАЦІЇ ВІДПОВІДЕЙ ДЛЯ ДОПОМІЖНОЇ КОМУНІКАЦІЇ**

Проблема комунікації є надзвичайно важливою в сучасному світі, де велика кількість людей стикається з труднощами в спілкуванні через вади слуху чи мовлення. Виявлення потреб таких людей та розробка рішень для полегшення комунікації має вирішальне значення для їхнього соціального включення та доступу до інформації. Зроблено огляд існуючих методів генерації відповідей для допоміжної комунікації та запропоновано метод для створення ефективних комунікаційних інструментів, здатних адаптуватися до потреб користувачів.

The problem of communication is extremely important in today's world, where a large number of people face difficulties in communication due to hearing or speech impairments. Identifying the needs of such people and developing solutions to facilitate communication is crucial for their social inclusion and access to information. An overview of existing response generation methods for assistive communication is made and a method for creating effective communication tools capable of adapting to user needs is proposed.

Комунікація є однією з найважливіших потреб людини, але для багатьох людей з вадами слуху чи мовлення це стає великою проблемою. За даними ВООЗ, близько 430 мільйонів людей у світі мають значні порушення слуху чи мовлення, що ускладнює їхню соціальну інтеграцію та повсякденне життя [1]. Допомога людям з вадами слуху та мовлення є важливим кроком до забезпечення їхнього повноцінного доступу до спілкування та інформації, сприяючи їхній соціальній і професійній інтеграції. Штучний інтелект відкриває нові можливості, дозволяючи створювати додатки для підтримки комунікації людей з такими вадами. Перспективним підходом є поєднання генеративних моделей з інформаційним пошуком, що дозволяє швидко знаходити необхідну інформацію та адаптувати відповіді до потреб користувачів [2].

Допоміжна комунікація — це підхід, що використовує різноманітні засоби та методи для полегшення спілкування осіб, які мають труднощі в усному чи письмовому спілкуванні. Допоміжна комунікація використовує різні засоби, зокрема жестову мову, комунікаційні пристрої, картки із зображеннями та текстові додатки, для полегшення спілкування осіб з порушеннями слуху, мовлення або когнітивними обмеженнями. Сучасні мобільні додатки та спеціалізовані пристрої

забезпечують інтерактивне й зручне спілкування, враховуючи індивідуальні потреби користувачів [3].

У системах автоматизованої допоміжної комунікації використовують різні методи генерації відповідей, кожен з яких має свої переваги та обмеження.

Методи на основі правил базуються на заздалегідь визначених умовах, що дозволяє отримати точні й передбачувані відповіді. Цей підхід особливо підходить для задач, де важлива стабільність і контрольованість результатів, наприклад, в інформаційних системах з обмеженими варіантами відповідей. Однак, такі методи мають низьку гнучкість, не підходять для адаптації до нових запитів і важко масштабуються для складніших сценаріїв [4].

Методи на основі шаблонів використовують готові шаблони, які дають можливість швидко формувати осмислені відповіді. Це корисно для чат-ботів, які обробляють рутинні запити, оскільки їхня реалізація проста та економить ресурси. Проте ці методи мають обмежену адаптивність, не здатні створювати оригінальні відповіді та з часом втрачають ефективність при розширенні кількості запитів [5].

Методи пошуку відповіді зосереджені на виборі найбільш підходящих відповідей із бази даних або текстового корпусу. Вони добре працюють у випадках, коли є велика кількість даних, оскільки дозволяють швидко знайти відповіді на повторювані питання, але сильно залежать від якості й обсягу бази знань. Такий підхід обмежує здатність системи адаптуватися до нових запитів, оскільки вона не може створювати нові відповіді [6].

Методи на основі великих мовних моделей (LLM), таких як GPT чи BERT, використовують нейронні мережі для створення унікальних відповідей залежно від контексту запиту. Ці моделі забезпечують високу гнучкість і можуть адаптуватися до різноманітних запитів, оскільки здатні "навчатися" на великих масивах даних, а також легко масштабуються. Незважаючи на це, LLM є ресурсоемними, потребують значних обчислювальних ресурсів і чутливі до якості даних для тренування, що може призвести до появи непередбачуваних або неточних відповідей при недостатньому навчанні [7].

Метою роботи є розробка методу генерації відповідей для допоміжної комунікації. Робота запропонованого методу полягає в генерації відповідей для допоміжної комунікації шляхом інтеграції великих мовних моделей (LLM) з механізмами інформаційного пошуку для адаптації відповідей до контексту і потреб користувача, забезпечуючи їхню релевантність, точність та інформаційну насиченість. На рисунку 1 представлено схему роботи методу.

Етап 1 – Отримання даних. На цьому початковому етапі відбувається отримання даних з різних джерел. Це можуть бути документи, статті, чи інші форми вмісту, які слугуватимуть основою для подальшої роботи. Дана інформація збирається і зберігається для подальшого оброблення.

Етап 2 – Індексція. Отримані дані індексуються, що дозволяє швидко їх знаходити під час запитів. Індексція передбачає структурування даних у такий спосіб, щоб забезпечити ефективний доступ до них. Це може включати створення словників, баз даних або інших структур даних.

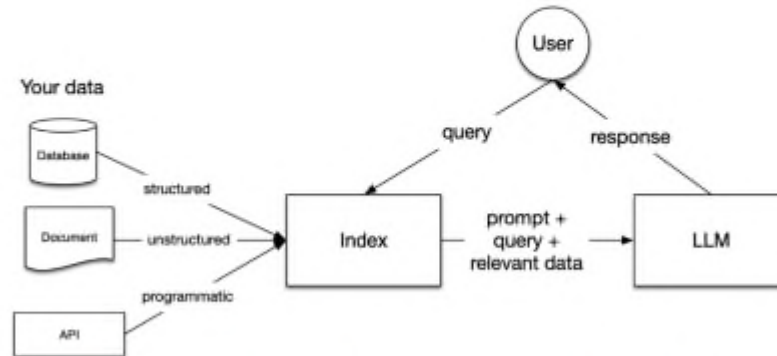


Рисунок 1 – Схема роботи методу генерації відповідей

Етап 3 – Запит. Коли користувач формулює запит, система аналізує його, щоб знайти відповідні дані з індексованих джерел. Це може включати обробку природної мови для кращого розуміння намірів користувача і вибору найбільш релевантних результатів.

Етап 4 – Генерація зразка. На цьому етапі система формує зразок відповіді, використовуючи інформацію, отриману з індексованих даних. Це може включати генерацію тексту, який надає користувачу корисну інформацію, базуючись на знайдених даних.

Етап 5 – Генерація відповіді. Фінальний етап передбачає формування остаточної відповіді на запит користувача. Це може включати об'єднання згенерованих зразків і даних, що були отримані на попередніх етапах, для створення змістовної і зрозумілої відповіді.

Етап 6 – Висновок. Після генерації відповіді, система може надавати рекомендації або додаткові ресурси, що допомагають користувачу краще зрозуміти тему або задовольнити його потреби. Це завершує процес, але система може також зберігати отриману інформацію для покращення майбутніх запитів.

Отже, запропонований метод генерації відповідей із доповнюючим інформаційним пошуком є ефективним інструментом для покращення комунікації людей з порушеннями мовлення або слуху. Він дозволяє автоматично генерувати відповіді з використанням додатковою інформації, що значно підвищує точність і

релевантність відповідей. Інтеграція великих мовних моделей із системами пошуку дозволяє створити гнучку систему, здатну адаптуватися до нових запитів та оновлювати базу знань для підтримки актуальної інформації. Це сприятиме покращенню комунікації та соціальної взаємодії людей з особливими потребами.

Перелік посилань

1. Вади слуху URL: https://uk.wikipedia.org/wiki/Вади_слуху
2. Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, Suranga Nanayakkara; Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering. Transactions of the Association for Computational Linguistics 2023
3. "Everyone Deserves AAC": Preliminary Study of the Experiences of Speaking Autistic Adults Who Use Augmentative and Alternative Communication Amy L. Donaldson , endever* corbin and Jamie McCoy
4. Eftimov T, Koroušić Seljak B, Korošec P (2017) A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. PLoS ONE
5. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean; Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems, 2013.
6. Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, Suranga Nanayakkara; Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering. Transactions of the Association for Computational Linguistics 2023
7. Zhicheng Guo, Sijie Cheng, Yile Wang, Peng Li, Yang Liu; Retrieval-Augmented Generation for Large Language Models: A Survey.

ISSN 2307-5732
DOI 10.31891/2307-5732

Науковий журнал



ВІСНИК

Хмельницького національного
університету

Технічні науки

Довідка: ВХНУ ТН 25/11/24

Видання: Вісник Хмельницького національного університету. Технічні науки

Категорія фаховості видання: фахове видання України, у якому можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора наук, кандидата наук та ступеня доктора філософії, категорії «Б» філософії, категорії «Б» (наказ МОН №1643 від 28.12.2019, наказ МОН №409 від 17.03.2020).

Напрямок – технічні науки за спеціальностями – 101, 121, 122, 123, 124, 125, 141, 151, 161, 172, 181, 182 (28.12.2019), спеціальності – 131, 132, 133 (17.03.2020)

Назва статті: МЕТОД ГЕНЕРАЦІЇ ВІДПОВІДЕЙ З ДОПОВНЮЮЧИМИ ІНФОРМАЦІЙНИМ ПОШУКОМ ДЛЯ ДОПОМІЖНОЇ КОМУНІКАЦІЇ

Автори: Медведчук В.Ю., Багрий Р.О., Скрипник Т.К., Мазурець О.В.Ю., Монастирська Д.С. (Хмельницький національний університет)

Номер, у який прийнято статтю: №6 до друку орієнтовно буде рекомендовано до 30 грудня 2024 року.

25.11.2024

Начальник відділу
інтелектуальної власності та трансферу технологій Ю.В.Кравчик



УДК 004.8

DOI:

МЕДВЕДЧУК ВІТАЛІЙ

Хмельницький національний університет

ORCID ID: [0009-0005-9661-3251](https://orcid.org/0009-0005-9661-3251)e-mail: medvedchuk.vitalii@gmail.com

БАГРІЙ РУСЛАН

Хмельницький національний університет

ORCID ID: [0000-0001-5219-1185](https://orcid.org/0000-0001-5219-1185)e-mail: bahriiro@khmnu.edu.ua

СКРИПНИК ТЕТЯНА

Хмельницький національний університет

ORCID ID: [0000-0002-8531-5348](https://orcid.org/0000-0002-8531-5348)e-mail: tkskripnik1970@gmail.com

МАЗУРЕЦЬ ОЛЕКСАНДР

Хмельницький національний університет

ORCID ID: [0000-0002-8900-0650](https://orcid.org/0000-0002-8900-0650)e-mail: exc.chong@gmail.com

МОНАСТІРСЬКА ДАР'Я

Хмельницький національний університет

e-mail: monkadasha@gmail.com

МЕТОД ГЕНЕРАЦІЇ ВІДПОВІДЕЙ З ДОПОВНЮЮЧИМ ІНФОРМАЦІЙНИМ ПОШУКОМ ДЛЯ ДОПОМІЖНОЇ КОМУНІКАЦІЇ

Проблема покращення комунікації для людей з обмеженими можливостями мовлення є надзвичайно важливою у сучасному суспільстві, де технології можуть значно полегшити взаємодію та інтеграцію таких осіб у соціальні процеси. Традиційні методи спілкування, такі як жестова мова чи письмове повідомлення, часто не забезпечують необхідної точності та швидкості комунікації, що призводить до бар'єрів у повсякденному житті. Сучасні технології, такі як великі мовні моделі (LLM) і технологія доповнюючого інформаційного пошуку (RAG), можуть значно покращити ці процеси. Вони дозволяють автоматично генерувати індивідуалізовані текстові відповіді, враховуючи не тільки запит користувача, а й контекст, що робить комунікацію більш точною і швидкою.

У статті пропонується метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації, що інтегрує релевантну інформацію з різних джерел, зокрема, історії чату та профілю користувача. Це дозволяє генерувати відповіді, які краще відповідають індивідуальним потребам кожного користувача. Метод передбачає три етапи: завантаження контексту, пошук релевантної інформації та генерацію кількох варіантів відповіді. Особливістю запропонованого методу є здатність враховувати широкий контекст і потреби користувача, що забезпечує високу точність і персоналізацію відповіді.

Проведені експерименти показали високу ефективність методу, зокрема в оцінці точності відповідей, що варіюється в межах 85-95%. Це дозволяє значно покращити комунікацію для людей з обмеженими можливостями мовлення, знижуючи соціальні бар'єри та покращуючи якість їхнього повсякденного життя.

Ключові слова: допоміжна комунікація, генерація відповідей, доповнюючий інформаційний пошук, великі мовні моделі, Retrieval-Augmented Generation, пошукові алгоритми, текстові відповіді, контекст.

Vitaliy MEDVEDCHUK, Ruslan BAHRII, Tetiana SKRYPNYK, Oleksandr MAZURETS, Daria MONASTYRSKA
Khmelnitskyi National University

METHOD OF RESPONSE GENERATION WITH RETRIEVAL-AUGMENTED INFORMATION SEARCH FOR ASSISTIVE COMMUNICATION

The issue of improving communication for people with speech impairments is critically important in modern society, where technology can greatly facilitate interaction and integration of such individuals into social processes. Traditional communication methods, such as sign language or written messages, often fail to provide the necessary accuracy and speed of communication, creating barriers in everyday life. Modern technologies, such as large language models (LLMs) and retrieval-augmented generation (RAG) technology, can significantly enhance these processes. They enable the automatic generation of individualized text responses, considering not only the user's query but also the context, making communication more accurate and faster.

The article proposes a method for response generation with retrieval-augmented information search for assistive communication, which integrates relevant information from various sources, including chat history and user profiles. This allows for generating responses that better address the individual needs of each user. The method consists of three stages: context loading, retrieval of relevant information, and generation of multiple response options. A distinctive feature of the proposed method is its ability to consider extensive context and user needs, ensuring high accuracy and personalization of the response.

Experiments conducted demonstrated the method's high efficiency, particularly in the accuracy of responses, which ranged between 85% and 95%. This significantly improves communication for people with speech impairments, reducing social barriers and enhancing the quality of their daily lives.

Keywords: assistive communication, response generation, retrieval-augmented information search, large language models, Retrieval-Augmented Generation, search algorithms, text responses, context.

Постановка проблеми

Люди з порушеннями мовлення стикаються зі значними бар'єрами у повсякденній комунікації, що часто призводить до соціальної ізоляції та обмеження їхньої участі у громадському житті [1]. Augmentative and Alternative Communication (AAC) – це поширений спосіб комунікації, який використовується для поповнення або заміни мовлення або письма для тих, хто має обмеження у виробництві або розумінні мовлення [2]. Традиційні методи допоміжної комунікації включають жести, малюнки, текстові таблиці та пристрої для створення письмових або голосових повідомлень. Однак ці методи не завжди забезпечують достатню швидкість і точність спілкування, особливо у складних соціальних контекстах або під час динамічних діалогів [3]. Низькотехнологічні засоби, такі як жести чи символи, обмежені в можливості адаптації до індивідуальних потреб користувача та контексту взаємодії. Високотехнологічні AAC-рішення, що включають програмні додатки для смартфонів і планшетів зі синтезованою мовою, покращують ситуацію, але потребують подальшого вдосконалення для підвищення ефективності комунікації [4].

Сучасні технології, такі як великі мовні моделі і доповнюючий інформаційний пошук, відкривають нові можливості для AAC-систем. Великі мовні моделі, здатні генерувати текстові відповіді на основі глибокого аналізу контексту, що дозволяє адаптувати відповіді до потреб користувача та історії взаємодії. Інтеграція технології RAG дозволяє залучати актуальну інформацію з зовнішніх джерел, що підвищує точність і змістовність відповідей у реальному часі [5]. Дослідження показують, що застосування LLM у AAC-системах покращує якість комунікації, знижує соціальні бар'єри та забезпечує гнучкість у різних соціальних контекстах.

Персоналізовані текстові відповіді – це відповіді, які адаптуються до індивідуальних особливостей користувача, зокрема до історії попередніх діалогів, потреб та контексту запиту. У процесі взаємодії людини з особливими потребами з системою комунікації користувач ініціює запит через інтерфейс системи, використовуючи текст, символи або жести. Система завантажує контекст, аналізуючи попередні діалоги та профіль користувача, а потім здійснює пошук релевантної інформації для уточнення відповіді. Після цього генерується персоналізована текстова відповідь, яка враховує як знайдені дані, так і індивідуальні особливості користувача. Саме на етапі генерації відповіді відбувається персоналізація з урахуванням усієї доступної

інформації.

Ефективність таких рішень підтверджена науковими дослідженнями, які вказують на зростання автономності користувачів завдяки персоналізованим і швидким відповідям, що генеруються мовними моделями [6]. Таким чином, необхідно розробити метод, який використовує великі мовні моделі та технології доповнюючого пошуку для створення точних і персоналізованих відповідей. Це дозволить значно покращити комунікацію для людей з обмеженими можливостями мовлення та сприяти їхній соціальній інтеграції.

Аналіз останніх джерел

Великі мовні моделі (Large Language Models) є сучасними системами штучного інтелекту, здатними генерувати текстові відповіді завдяки навчанню на величезних обсягах даних. Розвиток LLM почався зі створення моделей на основі трансформерів, таких як BERT, GPT та інших подібних архітектур. Модель GPT (Generative Pre-trained Transformer), розроблена OpenAI, демонструє здатність генерувати тексти з глибоким контекстним розумінням, що робить її ефективною для застосування у допоміжних комунікаційних системах [7].

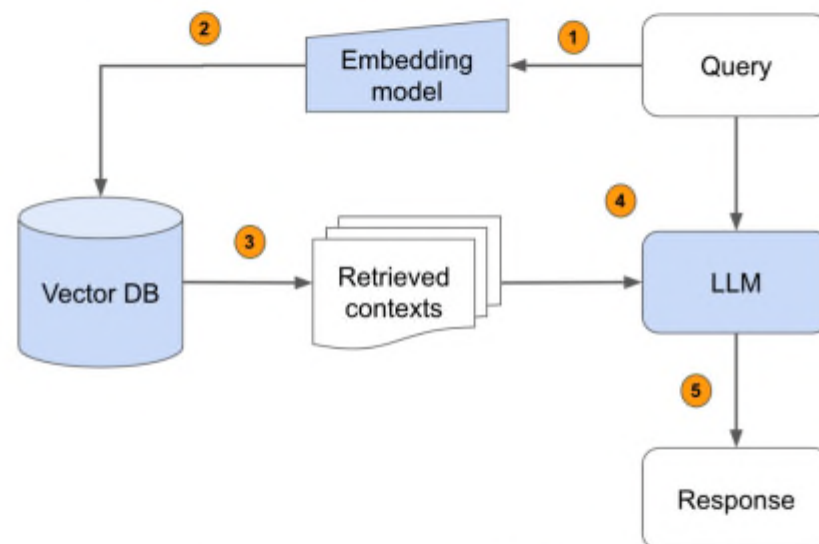


Рисунок 1 – Схема інтеграції LLM з системою RAG

Обробка запиту починається з введення запиту, який може бути питанням, підказкою або будь-яким іншим введенням, на яке мовна модель повинна відповісти. Потім запит передається до моделі вбудовування, яка перетворює його на вектор - числове представлення, яке може бути зрозумілим і обробленим системою. Цей вектор запиту використовується для пошуку у векторній базі даних, яка містить попередньо обчислені вектори потенційних контекстів, що можуть бути використані моделлю для генерації відповіді. Система отримує найбільш релевантні контексти на основі того, наскільки близько їхні вектори відповідають вектору запиту. Отримані контексти передаються до великої мовної моделі, яка використовує цю інформацію для генерації обґрунтованої та точної відповіді. LLM враховує як оригінальний запит, так і отримані контексти для створення всебічної та релевантної відповіді, синтезуючи інформацію з контекстів, щоб забезпечити, що відповідь не тільки базується на попередніх знаннях, але й доповнена конкретними деталями з отриманих даних. Нарешті, LLM видає відповідь, яка тепер інформована зовнішніми даними, отриманими в процесі, що робить її більш точною та детальною.

У роботі розглядається метод застосування Retrieval-Augmented Generation (RAG) разом із великими мовними моделями (LLM) для створення налаштованих мовних рішень. Основна увага приділена забезпеченню точності та індивідуалізації відповідей на основі конкретних запитів користувачів. Використовується підхід, де інформація спочатку отримується із зовнішніх джерел даних, а потім інтегрується LLM для генерування відповідей, релевантних до контексту. Цей підхід дозволяє користувачам AAC отримувати відповіді, які адаптовані до їхніх унікальних потреб і контекстів. Наприклад, люди з комунікативними порушеннями можуть швидко отримати підказки або тексти, які відповідають їхній конкретній ситуації. Це покращує якість спілкування та робить його більш природним [6].

Стаття описує інтеграцію методів глибокого навчання та нейронних мереж для текстового генерування, яке базується на специфічному контексті. Підхід RAG дозволяє системі виконувати пошук інформації у великих наборах даних перед генерацією відповідей LLM. Ця комбінація підвищує точність та відповідність створених текстів до потреб користувача. Такий підхід може бути використаний у системах допоміжної комунікації для автоматичного підбору фраз, які підходять до конкретної ситуації. Це особливо корисно для створення комунікативних підказок або фраз на основі специфічного контексту, наприклад, під час спілкування на певні теми чи у конкретних соціальних сценаріях [8].

У цій статті представлено підхід до адаптивної комунікації, де Retrieval-Augmented Generation використовується для покращення продуктивності мовних моделей. RAG забезпечує пошук актуальної інформації, що потім інтегрується LLM для створення адаптованих текстів. Автори досліджують, як цей метод підвищує здатність моделей адаптувати відповіді до конкретних потреб користувачів. Цей підхід може допомогти у створенні персоналізованих систем комунікації для користувачів із порушеннями мовлення. Наприклад, системи можуть швидко підбирати найбільш відповідні фрази або тексти для конкретної ситуації, підвищуючи швидкість та точність комунікації. Це дозволяє користувачам легше та ефективніше висловлювати свої думки та потреби [9].

Таким чином, застосування великих мовних моделей та технології RAG є перспективним напрямком для вдосконалення систем допоміжної комунікації. Це дозволяє створювати адаптивні, персоналізовані та високоточні відповіді, що значно покращує якість життя людей з обмеженими можливостями мовлення.

Метою роботи є: покращення комунікації для людей з обмеженими можливостями мовлення за допомогою генерації відповідей з доповнюючим інформаційним пошуком. Метод повинен підвищити швидкість, точність та гнучкість спілкування для людей з обмеженими можливостями мовлення. Потрібно здійснити аналіз ефективності роботи запропонованого методу.

Виклад основного матеріалу

Метод дозволяє ефективно поєднувати генерацію тексту за допомогою великої мовної моделі та пошук відповідної інформації в документах або базах даних, що забезпечує високу точність і адаптивність відповідей до конкретних запитів. Метод базується на трьох основних етапах: індексація даних, пошук інформації та генерація відповіді на основі знайденої інформації. Ці етапи взаємодіють між собою, що забезпечує високу точність і релевантність відповідей. На рисунку 2 зображено процес роботи методу.

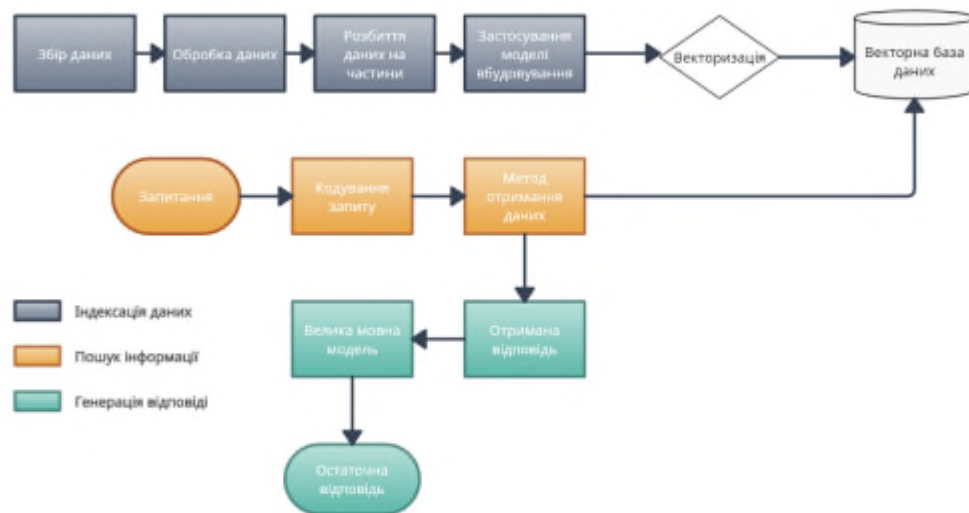


Рисунок 2 – Схема роботи методу генерації відповіді

Індексація даних забезпечує обробку даних для зберігання в базі даних у вигляді векторних представлень. Це дозволяє системі знаходити та зіставляти інформацію, використовуючи семантичну подібність, а не простий текстовий збіг.

Збір даних може охоплювати не лише документи, а й різноманітні джерела, такі як веб-сайти, бази даних, API або навіть зображення. Вибір інструментів для збору інформації залежить від джерела та потреби в забезпеченні цілісності та відповідності запитам користувачів.

На етапі обробки даних важливо очистити інформацію від зайвих елементів та неструктурованих даних. Застосовуються різні техніки, зокрема нормалізація тексту, видалення стоп-слів та вирівнювання формату. Крім того, необхідно виявити та видалити дублікати, які можуть вплинути на ефективність пошуку.

Щоб полегшити пошук та покращити результати, великі текстові блоки або документи розбиваються на менші частини – наприклад, речення чи абзаци. Ці частини можуть бути автономними або взаємопов'язаними залежно від контексту.

Для трансформації тексту в багатовимірні вектори, які відображають зміст, застосовуються моделі вбудовування. Вони можуть бути натреновані на конкретних задачах або використовуватися універсальні моделі, які надають високоякісні ембедінги.

Векторизація дозволяє створити семантичне відображення тексту, що дає змогу системам порівнювати фрагменти тексту через вимірювання відстані між векторами, наприклад, за допомогою косинусної схожості. Чим ближчі два вектори, тим подібніші їх змісти.

Після генерування векторів вони зберігаються в спеціалізованих векторних базах даних. Ці бази даних оптимізовані для швидкого пошуку та масштабування, забезпечуючи ефективний семантичний пошук і надання релевантних результатів у реальному часі.

Пошук інформації відповідає за пошук релевантної інформації, використовуючи векторні представлення. Він дозволяє отримати потрібні документи на основі запиту користувача.

Користувач вводить запит, який може бути як простим питанням, так і складним пошуковим запитом. Часто в запитах є ключові слова або контекст, які допомагають сформулювати точні критерії для подальшого пошуку векторів.

Запит перетворюється на вектор за допомогою спеціалізованих кодувальних моделей, які оброблюють текст і відображають його зміст у вигляді багатовимірних векторів. Це дозволяє виявляти схожість між запитом і іншими текстами в базі даних на глибшому рівні, а не лише через точний збіг слів.

Для пошуку найбільш релевантних результатів використовуються методи семантичного пошуку, що порівнюють вектори за допомогою відстані або подібності між ними. Алгоритми, як-от максимальна маржинальна релевантність, не лише підвищують точність, але й покращують різноманітність відповідей, мінімізуючи дублювання інформації у результатах пошуку.

Векторні бази даних, оптимізовані для швидкого пошуку, дозволяють працювати з великими обсягами даних. Вони використовують різні індексаційні структури для ефективного знаходження найбільш релевантних документів, що знижує затримки та покращує час відгуку системи.

Генерація відповіді обробляє знайдену інформацію та створює кінцеву відповідь за допомогою генеративних моделей. Її мета – зробити результат зручним для сприйняття користувачем.

На цьому етапі система отримує фрагменти інформації з бази даних, знайдені під час пошуку. Ці фрагменти можуть містити текст, уривки з документів чи інші дані. Важливо, щоб ці частини були достатньо детальними та містили необхідний контекст для формування змістовної відповіді.

Велика мовна модель (LLM) відповідає за генерацію відповіді, інтегруючи знайдену інформацію в контекст запиту користувача. Модель не лише надає прямі відповіді, але й може додавати пояснення, логічні висновки або структурувати відповідь так, щоб вона ставала зрозумілою та корисною. Для цього використовуються різні стратегії, зокрема переформулювання, підсумки або аналітичні висновки.

Prompt Engineering є важливою складовою методу генерації відповідей із доповнючим інформаційним пошуком. Оптимізація інструкцій для великої мовної моделі (LLM) дозволяє досягти високої точності, адаптивності та релевантності відповідей. Інструкція для LLM складається з трьох ключових елементів: основної інструкції, контексту та формування відповіді.

Основна інструкція визначає завдання моделі, що полягає в наступному. Необхідно надати відповідь на запитання співрозмовника так, як це могла б зробити людина, інформація про яку міститься у контексті. У відповідях слід використовувати лише ключові слова без додаткових уточнень, описів чи пояснень. Якщо відповідь міститься у контексті, потрібно надати її. Якщо такої відповіді немає, бажано сформулювати кілька варіантів відповідей, що базуються на контексті, історії чату та самому запитанні.

Контекст інструкції включає в себе: контекст (інформацію про людину, якщо є), історію чату (попереднє спілкування), запитання. У контексті використовується інформація з документів, якщо вони наявні, історії чату та запитання. Підсумовуючи, необхідно зазначити, що в контексті інструкції застосовується інформація, отримана від користувача та його співрозмовника.

Формування відповіді уточнює, як повинні виглядати кінцеві варіанти відповіді, а саме – необхідно надати 4 короткі варіанти відповіді. Відповіді повинні:

1. Бути чіткими, короткими, простими і зрозумілими.
2. Уникати будь-яких роздумів, пояснень, уточнень чи рекомендацій.
3. Бути максимально лаконічними та зрозумілими, як у природному діалозі.

Адаптація інструкцій до потреб користувача забезпечує зменшення кількості некоректних результатів та підвищує ефективність комунікації, що є критичним для систем допоміжної комунікації.

Остаточна відповідь, що отримує користувач, об'єднує знайдені дані та логічні висновки, сформовані моделлю. Це може бути текстова відповідь, яка надає прямий відгук на запит, або короткий підсумок, що висвітлює основні моменти. Важливо, щоб відповідь залишалась релевантною і легкою для сприйняття користувачем.

Програмна реалізація методу

Для реалізації методу генерації відповідей з доповнючим інформаційним пошуком для допоміжної комунікації було використано ряд інструментів та технологій, що забезпечують високу ефективність та точність

системі.

Основною мовою програмування, що використана для розробки, є Python. Ця мова є однією з найпопулярніших завдяки своїй універсальності, простоті та багатому набору бібліотек, що забезпечує гнучкість у реалізації різноманітних функцій платформи. Вона дозволяє ефективно інтегрувати різні компоненти системи, що сприяє швидкому розвитку та адаптації до нових вимог [10].

Для створення графічного інтерфейсу користувача застосовувалася бібліотека CustomTkinter. Вона дозволяє створювати адаптивні інтерфейси з підтримкою темної теми, що робить взаємодію з платформою зручною та інтуїтивно зрозумілою для користувачів з різними рівнями підготовки [11].

Для роботи з векторними поданнями даних використано Chroma. Цей інструмент дозволяє перетворювати текстові документи у векторні формати, що забезпечує швидкий пошук релевантної інформації та інтеграцію її з мовними моделями. Chroma є основою для створення бази знань, що забезпечує ефективний доступ до необхідної інформації [12].

Бібліотека PyPDF була використана для автоматичної індексації та обробки PDF-документів. Вона дозволяє ефективно працювати з текстами в складних документах, таких як розділи, таблиці або зображення, що забезпечує точність та повноту даних для подальшої обробки [13].

Моделі, використовувані через Ollama, зокрема `aya-expanse:8b` та `omic-embed-text`, відіграють важливу роль у генерації відповідей. Модель `aya-expanse:8b` відповідає за створення точних та контекстуальних відповідей, аналізуючи поточний стан чату та враховуючи специфічні потреби користувача [14]. Модель `omic-embed-text` відповідає за інтеграцію зовнішніх знань у процес генерації, що дозволяє забезпечити відповідь, яка не лише точна, але й актуальна з точки зору зовнішніх джерел [15].

Для реалізації технології Retrieval-Augmented Generation (RAG) використовувалася бібліотека LangChain. Вона дозволяє ефективно поєднувати пошук релевантної інформації з баз даних та генерацію текстових відповідей, що робить відповіді більш точними і відповідними до контексту запиту користувача [16].

Всі ці інструменти інтегровані в єдину систему, що забезпечує високу точність і швидкість взаємодії з користувачем. Це дозволяє генерувати персоналізовані та релевантні відповіді, значно покращуючи комунікацію для людей з обмеженими можливостями мовлення.

Аналіз ефективності запропонованого методу

Ефективність системи комунікації – це міра відповідності між запланованими цілями та фактичними результатами, отриманими під час тестування за встановленими критеріями. Вона відображає наскільки успішно система задовольняє потреби користувачів та досягає поставлених завдань у визначених умовах використання.

Для тестування системи комунікації з використанням генерації відповідей з доповнюючим інформаційним пошуком використаємо критерії оцінки, що враховують наступні аспекти: Точність відповідей – цей критерій оцінює, наскільки правильно система зрозуміла ваше запитання та надала відповідь, що відповідає його змісту та вимогам; Варіативність відповідей – цей критерій відображає кількість запропонованих варіантів відповідей, їх різноманітність, якість та відповідність запиту користувача; Контекстність відповідей – цей критерій оцінює здатність системи враховувати не тільки попередні повідомлення у діалозі, але й додаткові джерела інформації, особливо у разі запитів, що є частиною тривалої розмови або потребують специфічних знань про користувача; Значущість відповідей – цей критерій оцінює наскільки корисною, релевантною та важливою є отримана відповідь в контексті вашого запиту.

Для тестування системи комунікації будуть створені імітації діалогів на теми «Відвідування ресторану», «Покупки в продуктовому магазині» та «Побут». Імітації діалогів включатимуть конкретні запитання адресовані системі, яка надаватиме варіанти відповіді на запитання для відтворення діалогу. Для порівняння система комунікації буде протестована окремо з використанням методу генерації відповідей з інформаційним пошуком, так і без його використання. Для коректного порівняння запитання під час тестування на однакові теми дублюватимуться – спочатку у варіанті з використанням методу генерації з документом, а потім без нього.

Після проведення ряду тестувань, проведено дослідження для аналізу ефективності системи комунікації.

На рисунку 3 продемонстрована статистика балів кожного тестування з використанням доповнюючого пошуку.

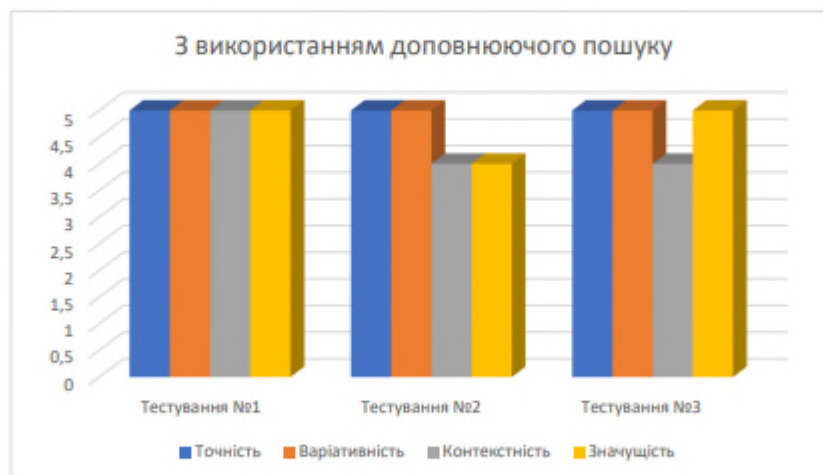


Рисунок 3 – Діаграма тестування з використанням доповнюючого пошуку

На рисунку 4 продемонстрована статистика балів кожного тестування без використання доповнюючого пошуку.

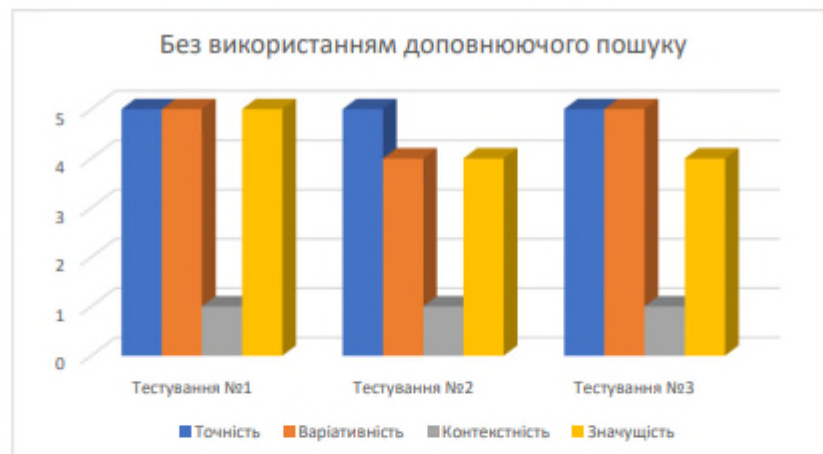


Рисунок 4 – Діаграма тестування без використання доповнюючого пошуку

Після проходження тестування з та без використання доповнюючого пошуку, система комунікації отримала 57 балів із 60 можливих та 45 балів із 60 можливих відповідно. Успішність тестувань можна оцінити за наведеними результатами:

1. Тестування на тему «Відвідування ресторану» з використанням доповнюючого пошуку – 100%;
2. Тестування на тему «Покупки у продуктовому магазині» з використанням доповнюючого пошуку – 90%;
3. Тестування на тему «Побут» з використанням доповнюючого пошуку – 95%;

4. Тестування на тему «Відвідування ресторану» без використання доповнюючого пошуку – 80%;
5. Тестування на тему «Покупки у продуктовому магазині» без використання доповнюючого пошуку – 70%;
6. Тестування на тему «Побут» без використання доповнюючого пошуку – 75%.

Середнє значення ефективності системи комунікації з використанням доповнюючого пошуку сягає 95% із 100% можливих – це прийнятний результат для системи комунікації, коли без використання доповнюючого пошуку лише – 75% із 100% можливих. В основному система комунікації проявила себе досить гарно у всіх тестуваннях, хоча іноді були і недоліки. Можна сказати, що система успішно впоралась із своїми задачами.

На рисунку 5 представлено діаграму, що демонструє статистику по кожному з критеріїв, заданих для оцінки системи комунікації.

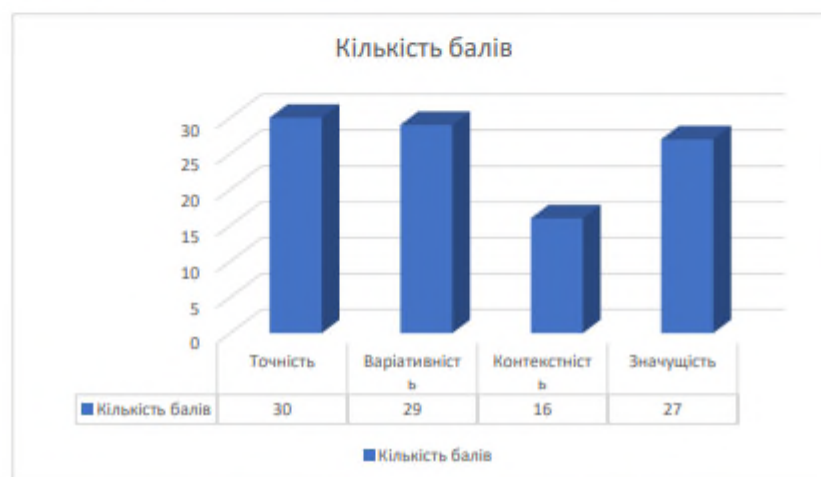


Рисунок 5 – Діаграма статистики кожного з критеріїв оцінки

Після проходження тестування з та без використання доповнюючого пошуку, система комунікації отримувала оцінку по кожному з критеріїв. Успішність критеріїв за результатами тестувань можна оцінити нижче:

1. Точність відповідей – 100%;
2. Варіативність відповідей – 96,67%;
3. Контекстність відповідей – 53,33%;
4. Значущість відповідей – 90%.

Середня оцінка критеріїв системи комунікації сягає 85% із 100% можливих – це прийнятний результат для системи комунікації.

В результаті тестувань система комунікації з використанням генерації відповідей та доповнюючого інформаційного пошуку для допоміжної комунікації продемонструвала високий рівень ефективності. Оцінка за критеріями системи також підтвердила її високу ефективність. Загалом, система успішно виконала поставлені завдання, хоча й потребує подальшого вдосконалення.

Висновки

У результаті проведеної роботи покращено метод генерації відповідей з доповнючим інформаційним пошуком для допоміжної комунікації, який ефективно підвищує швидкість, точність і гнучкість спілкування для людей з обмеженими можливостями мовлення. Використання таких передових технологій, як великі мовні

моделі (LLM) та технологія Retrieval-Augmented Generation (RAG), дозволяє значно покращити комунікацію шляхом інтеграції актуальної інформації з різних джерел, що робить відповіді більш релевантними і змістовними.

Під час тестування системи з використанням методу доповнюючого пошуку було виявлено високий рівень ефективності. Система продемонструвала середній результат 95% ефективності, що свідчить про її здатність забезпечити точні і варіативні відповіді на різні запити користувачів. У порівнянні з традиційними методами без доповнюючого пошуку, ефективність системи без використання цієї технології складала лише 75%. Це підтверджує важливість інтеграції додаткового пошуку для забезпечення більш високої якості комунікації.

Крім того, система продемонструвала відмінні результати за критерієм точності (100%) та варіативності відповідей (96,67%). Хоча критерій контекстності відповіді виявив певні недоліки (53,33%), загальний рівень значущості відповідей (90%) підтверджує, що система може бути ефективно використана в реальних умовах. Результати критерію контекстності притерпіли сильного зниження в тестуваннях без доповнюючого пошуку, через що загальна оцінка також знизилась.

Таким чином, запропонований метод генерації відповідей з доповнюючим інформаційним пошуком має значний потенціал для полегшення комунікації для людей з вадами слуху та мовлення, а також для підвищення їх автономії в повсякденному житті. Подальші дослідження і вдосконалення технології дозволять ще більше покращити якість спілкування та зменшити бар'єри у взаємодії між людьми з різними можливостями.

Література

1. Shining a light on Augmentative and Alternative Communication [Електронний ресурс]. – Режим доступу: https://www.communicationmatters.org.uk/wp-content/uploads/2019/01/2013_Shining_a_Light_on_AAC.pdf
2. Augmentative and Alternative Communication (AAC) [Електронний ресурс] // American Speech-Language-Hearing Association | ASHA. – Режим доступу: <https://www.asha.org/public/speech/disorders/AAC/>
3. Chris Klein : Communication and Developing Relationships for People Who Use Augmentative and Alternative Communication, Assistive Technology Outcomes and Benefits, Volume 11, Summer 2017, pp.58-65 URL: https://www.atia.org/wp-content/uploads/2017/11/ATOB_ATOBN1V11_ART-5.pdf
4. Allen, A. A., Schlosser, R. W., Brock, K. L., & Shane, H. C. : The effectiveness of aided augmented input techniques for persons with developmental disabilities: a systematic review. *Augmentative and Alternative Communication*, 33(3), 149–159. (2017) URL: <https://doi.org/10.1080/07434618.2017.1338752>
5. Florianne Rademaker, Anke de Boer, Elisa Kupers, Alexander Minnaert: Applying the Contact Theory in Inclusive Education: A Systematic Review on the Impact of Contact and Information on the Social Participation of Students With Disabilities (2020) URL: <https://doi.org/10.3389/feduc.2020.602414>
6. Sayantan Pal, Souvik Das, Rohini K. Srihari, Jeffery Higginbotham, Jenna Bizovi : Empowering AAC Users: A Systematic Integration of Personal Narratives with Conversational AI (2024) URL: <https://doi.org/10.18653/v1/2024.customnlp4u-1.2>
7. Zhibo Chu, Zichong Wang, Chengming Li, Ruifeng Xu, Shiwen Ni, Xi Feng, Xiping Hu, Min Yang, Wenbin Zhang : History, Development, and Principles of Large Language Models—An Introductory Survey (2024) URL: <https://arxiv.org/abs/2402.06853>
8. Byun, J.; Kim, B.; Cha, K.-A.; Lee, E. : Design and Implementation of an Interactive Question-Answering System with Retrieval-Augmented Generation for Personalized Databases (2024) URL: <https://doi.org/10.3390/app14177995>
9. Nicholas Thomas Walker, Stefan Ultes, Pierre Lison : Retrieval-Augmented Neural Response Generation Using Logical Reasoning and Relevance Scoring (2023) URL: <https://arxiv.org/html/2310.13566>
10. Python [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Python>
11. CustomTkinter: A modern and easy-to-use GUI package for Python. [Електронний ресурс]. – Режим доступу: <https://customtkinter.tomschimansky.com>
12. Chroma: The simplest way to build search systems. [Електронний ресурс]. – Режим доступу: <https://www.trychroma.com/>

13. PyPDF Documentation [Электронный ресурс]. – Режим доступа: <https://pypdf.readthedocs.io/en/stable/>
14. AYA Expanse: Connecting our world. [Электронный ресурс]. – Режим доступа: <https://cohere.com/blog/aya-expanse-connecting-our-world>
15. Nomic Embed Text v1 [Электронный ресурс]. – Режим доступа: <https://www.nomic.ai/blog/posts/nomic-embed-text-v1>
16. Langchain: Building and deploying large language models. [Электронный ресурс]. – Режим доступа: <https://www.langchain.com>

References

1. Shining a light on Augmentative and Alternative Communication [Elektronnyi resurs]. – Rezhym dostupu: https://www.communicationmatters.org.uk/wp-content/uploads/2019/01/2013_Shining_a_Light_on_AAC.pdf
2. Augmentative and Alternative Communication (AAC) [Elektronnyi resurs] // American Speech-Language-Hearing Association | ASHA. – Rezhym dostupu: <https://www.asha.org/public/speech/disorders/AAC/>
3. Chris Klein : Communication and Developing Relationships for People Who Use Augmentative and Alternative Communication, Assistive Technology Outcomes and Benefits, Volume 11, Summer 2017, pp.58-65 URL: https://www.atia.org/wp-content/uploads/2017/11/ATOB_ATOBN1V11_ART-5.pdf
4. Allen, A. A., Schlosser, R. W., Brock, K. L., & Shane, H. C. : The effectiveness of aided augmented input techniques for persons with developmental disabilities: a systematic review. *Augmentative and Alternative Communication*, 33(3), 149–159. (2017) URL: <https://doi.org/10.1080/07434618.2017.1338752>
5. Florianne Rademaker, Anke de Boer, Elisa Kupers, Alexander Minnaert: Applying the Contact Theory in Inclusive Education: A Systematic Review on the Impact of Contact and Information on the Social Participation of Students With Disabilities (2020) URL: <https://doi.org/10.3389/feduc.2020.602414>
6. Sayantan Pal, Souvik Das, Rohini K. Srihari, Jeffery Higginbotham, Jenna Bizovi : Empowering AAC Users: A Systematic Integration of Personal Narratives with Conversational AI (2024) URL: <https://doi.org/10.18653/v1/2024.customnlp4u-1.2>
7. Zhibo Chu, Zichong Wang, Chengming Li, Ruifeng Xu, Shiwen Ni, Xi Feng, Xiping Hu, Min Yang, Wenbin Zhang : History, Development, and Principles of Large Language Models—An Introductory Survey (2024) URL: <https://arxiv.org/abs/2402.06853>
8. Byun, J.; Kim, B.; Cha, K.-A.; Lee, E. : Design and Implementation of an Interactive Question-Answering System with Retrieval-Augmented Generation for Personalized Databases (2024) URL: <https://doi.org/10.3390/app14177995>
9. Nicholas Thomas Walker, Stefan Ultes, Pierre Lison : Retrieval-Augmented Neural Response Generation Using Logical Reasoning and Relevance Scoring (2023) URL: <https://arxiv.org/html/2310.13566>
10. Python [Elektronnyi resurs]. – Rezhym dostupu: <https://uk.wikipedia.org/wiki/Python>
11. CustomTkinter: A modern and easy-to-use GUI package for Python. [Elektronnyi resurs]. – Rezhym dostupu: <https://customtkinter.tomschimansky.com>
12. Chroma: The simplest way to build search systems. [Elektronnyi resurs]. – Rezhym dostupu: <https://www.trychroma.com/>
13. PyPDF Documentation [Elektronnyi resurs]. – Rezhym dostupu: <https://pypdf.readthedocs.io/en/stable/>
14. AYA Expanse: Connecting our world. [Elektronnyi resurs]. – Rezhym dostupu: <https://cohere.com/blog/aya-expanse-connecting-our-world>
15. Nomic Embed Text v1 [Elektronnyi resurs]. – Rezhym dostupu: <https://www.nomic.ai/blog/posts/nomic-embed-text-v1>
16. Langchain: Building and deploying large language models. [Elektronnyi resurs]. – Rezhym dostupu: <https://www.langchain.com>

Додаток В

Програмні коди

```

Лістинг файлу app.py
import customtkinter as ctk
from tkinter import filedialog, messagebox, Listbox
import subprocess
import re
import os
from populate_database import main as populate_database, clear_database
from query_data import query_rag
from chat_memory import clear_memory, add_message_to_memory, load_memory

class App():
    def __init__(self):
        self.root = ctk.CTk()
        self.center_window()
        ctk.set_appearance_mode("dark")
        ctk.set_default_color_theme("dark-blue")

        self.root.title("МЕТОД ГЕНЕРАЦІЇ ВІДПОВІДЕЙ ДЛЯ ДОПОМІЖНОЇ КОМУНІКАЦІЇ")
        self.root.geometry("720x720")

        self.mainframe = ctk.CTkFrame(self.root)
        self.mainframe.pack(fill="both", expand=True)

        self.mainframe.grid_columnconfigure((0,11), weight=1)
        self.mainframe.grid_rowconfigure((0,19), weight=1)

        self.option_texts = {}

        self.status_label = ctk.CTkLabel(self.mainframe, text="Ollama Inactive",
font=("Segoe UI", 16), text_color="tomato")
        self.status_label.grid(row=3, column=1, pady=5, padx=5, sticky="ns",
columnspan=4)

        self.button_frame = ctk.CTkFrame(self.mainframe)
        self.button_frame.grid(row=5, column=1, pady=5, padx=5, sticky="ns",
columnspan=4)

        self.run_button = ctk.CTkButton(self.button_frame, text="Запустити сервер",
command=self.run_ollama)
        self.run_button.grid(row=0, column=0, padx=5, sticky="w")

        self.stop_button = ctk.CTkButton(self.button_frame, text="Вимкнути сервер",
command=self.stop_ollama, state="disabled")
        self.stop_button.grid(row=0, column=1, padx=5, sticky="e")

        self.docs_frame = ctk.CTkFrame(self.mainframe)
        self.docs_frame.grid(row=1, column=7, pady=5, padx=5, sticky="ns",
columnspan=4, rowspan=8)

        self.docs_label = ctk.CTkLabel(self.docs_frame, text="Завантажені документи",
font=("Segoe UI", 14))
        self.docs_label.grid(row=0, column=0)

        self.docs_listbox = Listbox(self.docs_frame, height=5, width=60,

```

```

background="gray35", foreground="white")
    self.docs_listbox.grid(row=1, column=0, pady=5, padx=5)

    self.add_doc_button = ctk.CTkButton(self.docs_frame, text="Додати документ",
command=self.add_document)
    self.add_doc_button.grid(row=2, column=0, pady=5, padx=5, sticky="w")

    self.remove_doc_button = ctk.CTkButton(self.docs_frame, text="Видалити
обраний", command=self.remove_document)
    self.remove_doc_button.grid(row=2, column=0, pady=5, padx=5, sticky="e")

    self.update_chroma_button = ctk.CTkButton(self.docs_frame, text="Оновити базу
даних", command=self.update_chroma)
    self.update_chroma_button.grid(row=3, column=0, pady=5, padx=5, sticky="w")

    self.clear_chroma_button = ctk.CTkButton(self.docs_frame, text="Очистити базу
даних", command=self.clear_chroma)
    self.clear_chroma_button.grid(row=3, column=0, pady=5, padx=5, sticky="e")

    self.clear_button = ctk.CTkButton(self.mainframe, text="Очистити пам'ять
чату", command=self.clear_chat)
    self.clear_button.grid(row=13, column=9, pady=5, padx=5, columnspan=2)

    self.response_text = ctk.CTkTextbox(self.mainframe, wrap="word", width=500,
height=425, fg_color="white", text_color="black")
    self.response_text.grid(row=10, column=1, pady=10, padx=10, rowspan=7,
columnspan=7, sticky="nsew")

    self.prompt_entry = ctk.CTkEntry(self.mainframe, placeholder_text="Задай
питання", width=500, fg_color="white", text_color="black")
    self.prompt_entry.grid(row=18, column=1, pady=5, padx=5, columnspan=7)

    self.ask_button = ctk.CTkButton(self.mainframe, text="Запитати",
command=self.ask_llm, state="disabled")
    self.ask_button.grid(row=18, column=9, pady=5, padx=5, columnspan=2)

    self.options_listbox = Listbox(self.mainframe, height=4, background="gray35",
foreground="white")
    self.options_listbox.grid(row=14, column=9, pady=5, padx=5, rowspan=2,
columnspan=2, sticky="nsew")

    self.save_choice_button = ctk.CTkButton(self.mainframe, text="Зберегти
вибір", command=self.save_user_choice)
    self.save_choice_button.grid(row=16, column=9, pady=5, padx=5, columnspan=2)

    self.load_chat_history()

    self.server_process = None
    self.load_documents()
    self.root.mainloop()

def run_ollama(self):
    if not self.server_process:
        command = [r"C:\Users\medve\AppData\Local\Programs\Ollama\ollama.exe",
"serve"]
        self.server_process = subprocess.Popen(command)
        self.status_label.configure(text="Ollama Running", text_color="lawn
green")
        self.run_button.configure(state="disabled")
        self.stop_button.configure(state="normal")
        self.ask_button.configure(state="normal")

def stop_ollama(self):

```

```

    if self.server_process:
        subprocess.run(["powershell", "-Command", "Get-Process | Where-Object
{$_ .ProcessName -like '*ollama*'} | Stop-Process"], shell=True)
        self.server_process = None
        self.status_label.configure(text="Ollama Inactive", text_color="tomato")
        self.run_button.configure(state="normal")
        self.stop_button.configure(state="disabled")
        self.ask_button.configure(state="disabled")

def ask_llm(self):
    prompt = self.prompt_entry.get()
    if re.match(r"^[a-zA-ЯиІіієЄrґa-zA-Z0-9\s.,!?' ]+$", prompt):

        self.load_chat_history()

        result = query_rag(prompt)

        cleaned_result = self.clean_text(result)

        self.options_listbox.delete(0, ctk.END)
        self.option_texts.clear()

        options = re.findall(r'\d+\.\s+(.*?)(?=\n\d+\.\s|$)',
cleaned_result.strip(), re.DOTALL)

        for i, option in enumerate(options):
            option_label = f"Варіант {i + 1}"
            self.option_texts[option_label] = option
            self.options_listbox.insert(ctk.END, option_label)

        self.response_text.insert(ctk.END, f"Запитання:
{prompt}\nВідповідь:\n{cleaned_result}\n")

        self.response_text.see(ctk.END)

        self.prompt_entry.delete(0, ctk.END)

def save_user_choice(self):
    selected_index = self.options_listbox.curselection()
    if selected_index:
        chosen_label = self.options_listbox.get(selected_index)
        chosen_option = self.option_texts.get(chosen_label)

        if chosen_option:
            add_message_to_memory(" ", f"Обрано варіант: {chosen_label}")
            add_message_to_memory("Відповідь", chosen_option)

            self.response_text.insert(ctk.END,
                                     f"Обрано варіант:
{chosen_label}\nВідповідь: {chosen_option}\n\n")
            messagebox.showinfo("Успіх", "Твій вибір було збережено!")
            self.options_listbox.delete(0, ctk.END)
        else:
            messagebox.showwarning("Попередження", "Оберіть варіант перш ніж
зберігати.")

def load_chat_history(self):
    self.response_text.delete("1.0", ctk.END)
    chat_history = load_memory()
    for message in chat_history:
        role = message["role"]
        content = message["content"]
        cleaned_content = self.clean_text(content)

```

```

        if role == " ":
            self.response_text.insert(ctk.END, f"{cleaned_content}")
        elif role == "Запитання":
            self.response_text.insert(ctk.END, f"{role}: {cleaned_content}")
        elif role == "Відповідь":
            self.response_text.insert(ctk.END, f"\n{role}:
\n{cleaned_content}\n")

    def clear_chat(self):
        clear_memory()
        self.response_text.delete("1.0", ctk.END)
        self.options_listbox.delete(0, ctk.END)
        messagebox.showinfo("Успіх", "Пам'ять чату очищено!")

    def load_documents(self):
        data_folder = 'data'
        self.docs_listbox.delete(0, ctk.END)
        if os.path.exists(data_folder):
            for doc in os.listdir(data_folder):
                self.docs_listbox.insert(ctk.END, doc)

    def add_document(self):
        file_path = filedialog.askopenfilename(filetypes=[("PDF files", "*.pdf")])
        if file_path:
            data_folder = 'data'
            os.makedirs(data_folder, exist_ok=True)
            destination = os.path.join(data_folder, os.path.basename(file_path))
            with open(file_path, 'rb') as fsrc, open(destination, 'wb') as fdst:
                fdst.write(fsrc.read())
            self.load_documents()

    def remove_document(self):
        selected_index = self.docs_listbox.curselection()
        if selected_index:
            selected_doc = self.docs_listbox.get(selected_index)
            os.remove(os.path.join('data', selected_doc))
            self.load_documents()

    def update_chroma(self):
        populate_database()
        messagebox.showinfo("Успіх", "База даних успішно оновлена!")

    def clear_chroma(self):
        clear_database()
        messagebox.showinfo("Успіх", "База даних успішно очищена від обраного
документу!")

    def center_window(self, width=720, height=720):
        screen_width = self.root.winfo_screenwidth()
        screen_height = self.root.winfo_screenheight()

        x = (screen_width // 2) - (width // 2)
        y = (screen_height // 2) - (height // 2)

        self.root.geometry(f"{width}x{height}+{x}+{y}")

    def clean_text(self, text):
        return re.sub(r'\n+', '\n', text).strip()

if __name__ == "__main__":
    App()

```

Лістинг файлу chat_memory.py

```
import json
import os

MEMORY_FILE = "chat_memory.json"

def load_memory():
    if os.path.exists(MEMORY_FILE):
        with open(MEMORY_FILE, "r", encoding="utf-8") as f:
            return json.load(f)
    return []

def save_memory(history):
    with open(MEMORY_FILE, "w", encoding="utf-8") as f:
        json.dump(history, f, ensure_ascii=False, indent=4)

def add_message_to_memory(role, content):
    history = load_memory()
    history.append({"role": role, "content": content})
    save_memory(history)

def clear_memory():
    if os.path.exists(MEMORY_FILE):
        os.remove(MEMORY_FILE)
```

Лістинг файлу get_embedding_function.py

```
from langchain_community.embeddings.ollama import OllamaEmbeddings

def get_embedding_function():
    embeddings = OllamaEmbeddings(model="nomic-embed-text")
    return embeddings
```

Лістинг файлу populate_database.py

```
import argparse
import os
import shutil
from langchain_community.document_loaders import PyPDFDirectoryLoader
from langchain_text_splitters import RecursiveCharacterTextSplitter
from langchain.schema.document import Document
from get_embedding_function import get_embedding_function
from langchain_chroma import Chroma

CHROMA_PATH = "chroma"
DATA_PATH = "data"

def main():

    parser = argparse.ArgumentParser()
    parser.add_argument("--reset", action="store_true", help="Reset the database.")
    args = parser.parse_args()
    if args.reset:
        print("Clearing Database")
        clear_database()

    documents = load_documents()
    chunks = split_documents(documents)
    add_to_chroma(chunks)

def load_documents():
    document_loader = PyPDFDirectoryLoader(DATA_PATH)
    return document_loader.load()
```

```

def split_documents(documents: list[Document]):
    text_splitter = RecursiveCharacterTextSplitter(
        chunk_size=800,
        chunk_overlap=80,
        length_function=len,
        is_separator_regex=False,
    )
    return text_splitter.split_documents(documents)

def add_to_chroma(chunks: list[Document]):
    db = Chroma(
        persist_directory=CHROMA_PATH, embedding_function=get_embedding_function()
    )

    chunks_with_ids = calculate_chunk_ids(chunks)

    existing_items = db.get(include=[])
    existing_ids = set(existing_items["ids"])
    print(f"Number of existing documents in DB: {len(existing_ids)}")

    new_chunks = []
    for chunk in chunks_with_ids:
        if chunk.metadata["id"] not in existing_ids:
            new_chunks.append(chunk)

    if len(new_chunks):
        print(f"Adding new documents: {len(new_chunks)}")
        new_chunk_ids = [chunk.metadata["id"] for chunk in new_chunks]
        db.add_documents(new_chunks, ids=new_chunk_ids)
    else:
        print("No new documents to add")

def calculate_chunk_ids(chunks):
    last_page_id = None
    current_chunk_index = 0

    for chunk in chunks:
        source = chunk.metadata.get("source")
        page = chunk.metadata.get("page")
        current_page_id = f"{source}:{page}"

        if current_page_id == last_page_id:
            current_chunk_index += 1
        else:
            current_chunk_index = 0

        chunk_id = f"{current_page_id}:{current_chunk_index}"
        last_page_id = current_page_id

        chunk.metadata["id"] = chunk_id

    return chunks

def clear_database():
    if os.path.exists(CHROMA_PATH):
        shutil.rmtree(CHROMA_PATH)

```

```

if __name__ == "__main__":
    main()
    documents = load_documents()
    print(len(documents))

```

Лістинг файлу query_data.py

```

import argparse
from langchain_chroma import Chroma
from langchain.prompts import ChatPromptTemplate
from langchain_community.llms.ollama import Ollama
from get_embedding_function import get_embedding_function
from chat_memory import load_memory, add_message_to_memory

```

```
CHROMA_PATH = "chroma"
```

```
PROMPT_TEMPLATE = """
```

```

Твоя задача - надати відповідь на запитання співрозмовника так, як це могла б зробити людина, інформація про яку міститься у контексті. Відповіді мають містити лише ключові слова без додаткових уточнень, описів чи пояснень. Якщо відповідь можна отримати безпосередньо з контексту, надай її. Якщо такої відповіді немає, сформулюй кілька варіантів відповідей, які базуються на контексті, історії чату та самому запитанні.

```

```
Відповіді повинні:
```

```

Бути чіткими, короткими, простими і зрозумілими.
Уникати будь-яких роздумів, уточнень чи рекомендацій.
Бути лаконічними, як у природному діалозі.

```

```
---
```

```

Контекст (інформація про людину, якщо є):
{context}

```

```

Історія чату (попереднє спілкування):
{chat_history}

```

```

Запитання:
{question}

```

```
---
```

```

Дай 4 короткі варіанти відповіді. Відповіді мають бути максимально лаконічними та зрозумілими.
Уникай зайвих пояснень чи уточнень.

```

```
"""
```

```

def query_rag(query_text: str, save_to_memory: bool = True):
    chat_history = "\n".join([f"{msg['role']}: {msg['content']}" for msg in load_memory()])

    embedding_function = get_embedding_function()
    db = Chroma(persist_directory=CHROMA_PATH, embedding_function=embedding_function)

    results = db.similarity_search_with_score(query_text, k=5)
    context_text = "\n\n---\n\n".join([doc.page_content for doc, _score in results])

    if not context_text:
        context_text = "Контексту немає."

```

```
prompt_template = ChatPromptTemplate.from_template(PROMPT_TEMPLATE)
prompt = prompt_template.format(
    context=context_text,
    question=query_text,
    chat_history=chat_history
)

model = Ollama(model="aya-explore:8b")
response_text = model.invoke(prompt)

if save_to_memory:
    add_message_to_memory("Запитання", query_text)
    add_message_to_memory("Відповідь", response_text)
    add_message_to_memory(" ", "")

return response_text

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Запит до моделі RAG")
    parser.add_argument("--query", type=str, help="Ваш запит")
    parser.add_argument("--add_info", type=str, help="Додати нову інформацію в історію")
    args = parser.parse_args()
```

Додаток Г

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Метод генерації відповідей з доповнюючим
інформаційним пошуком для допоміжної комунікації

Виконав:
Студент II курсу, група КНм-23-1
В.Ю. Медведчук

Керівник:
к.т.н., доцент кафедри КН
Р.О. Багрій

Актуальність

Люди з порушеннями мовлення стикаються зі значними бар'єрами у повсякденній комунікації, що часто призводить до соціальної ізоляції та обмеження їхньої участі у громадському житті. Традиційні методи допоміжної комунікації включають жести, малюнки, текстові таблиці та пристрої для створення письмових або голосових повідомлень. Однак ці методи не завжди забезпечують достатню швидкість і точність спілкування, особливо у складних соціальних контекстах або під час динамічних діалогів. Низькотехнологічні засоби, такі як жести чи символи, обмежені в можливості адаптації до індивідуальних потреб користувача та контексту взаємодії. Високотехнологічні ААС-рішення, що включають програмні додатки для смартфонів і планшетів зі синтезованою мовою, покращують ситуацію, але потребують подальшого вдосконалення для підвищення ефективності комунікації.

Сучасні технології, такі як великі мовні моделі і доповнюючий інформаційний пошук, відкривають нові можливості для ААС-систем. Великі мовні моделі, здатні генерувати текстові відповіді на основі глибокого аналізу контексту, що дозволяє адаптувати відповіді до потреб користувача та історії взаємодії. Інтеграція технології RAG дозволяє залучати актуальну інформацію з зовнішніх джерел, що підвищує точність і змістовність відповідей у реальному часі.

Мета та постановки задачі

Метою кваліфікаційної роботи магістра є покращення комунікації для людей з обмеженими можливостями мовлення за допомогою генерації відповідей з доповнючим інформаційним пошуком.

Для досягнення поставленої мети необхідно реалізувати виконання наступних задач:

- Провести огляд сучасних підходів до генерації відповідей з використанням мовних моделей та технологій доповнючого інформаційного пошуку (RAG);
- Розробити метод генерації відповідей з доповнючим інформаційним пошуком для допоміжної комунікації;
- Підготувати релевантні текстові дані для покращення точності генерації відповідей з доповнючим інформаційним пошуком;
- Розробити програмну реалізацію системи комунікації між людьми з використанням генерації відповідей з доповнючим інформаційним пошуком;
- Провести тестування взаємодії системи комунікації з користувачами та проаналізувати коректність його відповідей.

Об'єкт та предмет дослідження

Об'єкт дослідження – процес генерації текстових відповідей з доповнючим інформаційним пошуком.

Предмет дослідження – методи збору та аналізу інформації, великі мовні моделі та векторні бази даних для систем комунікації, орієнтованих на людей з обмеженими можливостями мовлення.

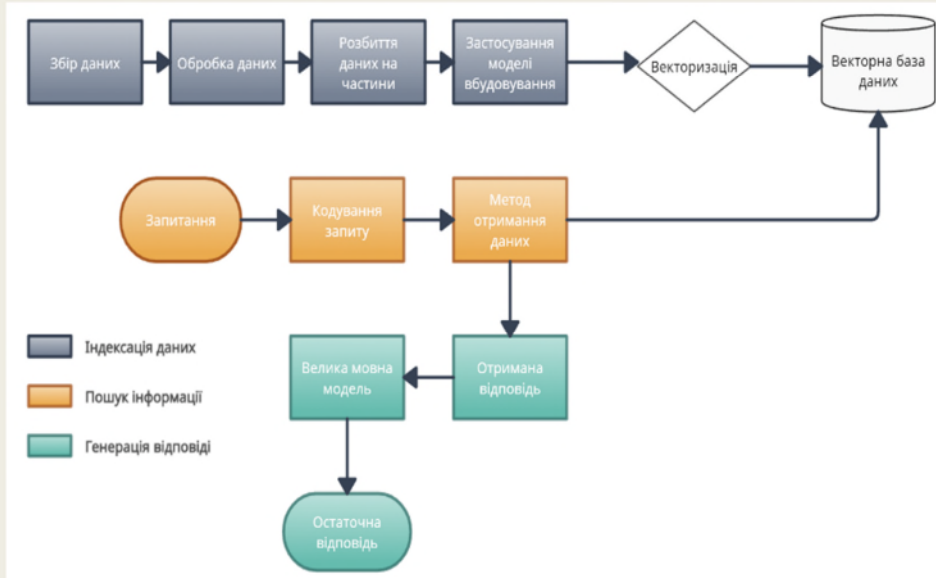
Наукова новизна

В результаті проведеної роботи удосконалено метод генерації відповідей для допоміжної комунікації за рахунок використання генеративної мовної моделі з інформаційним пошуком з урахуванням зовнішнього контексту, що дало можливість покращити комунікації для людей з обмеженими можливостями мовлення завдяки генерації релевантних варіантів відповідей.

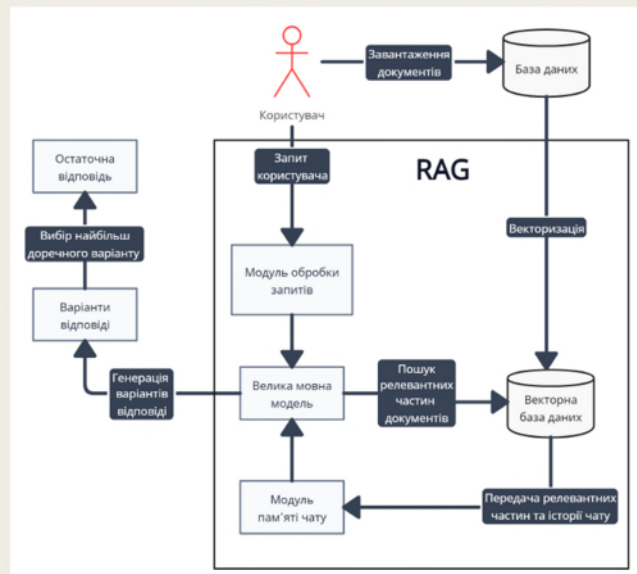
Схема методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації



Схема роботи методу генерації відповідей



Компоненти системи комунікації



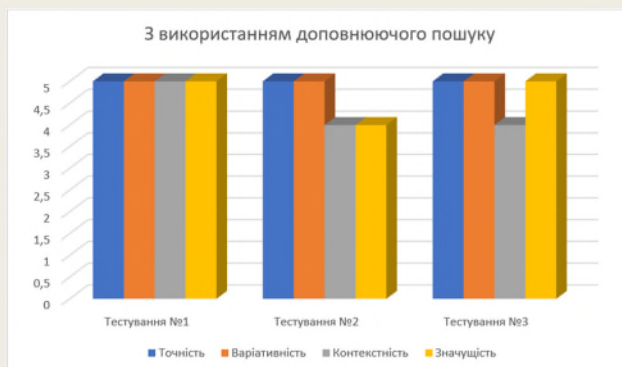
Prompt Engineering

Prompt Engineering є важливою складовою методу генерації відповідей із доповнюючим інформаційним пошуком. Оптимізація інструкцій для великої мовної моделі дозволяє досягти високої точності, адаптивності та релевантності відповідей. У контексті допоміжної комунікації правильне застосування Prompt Engineering дозволяє адаптувати відповіді до індивідуальних потреб користувачів, враховуючи їхні запити, контекст взаємодії та наявні дані. Це сприяє зменшенню кількості некоректних результатів і підвищенню ефективності комунікації, особливо для людей з вадами мовлення чи слуху. Інструкція для LLM складається з трьох ключових елементів:

- Основна інструкція – визначає загальне завдання та підхід до надання відповіді;
- Контекст інструкції – визначає джерела інформації, що використовуються для формування відповіді;
- Формування відповіді – визначає вимоги до структури та стилю кінцевих відповідей.

Таким чином, інструкція виконує функцію своєрідного «напрямого механізму», який забезпечує мовній моделі необхідну структуру для створення відповіді.

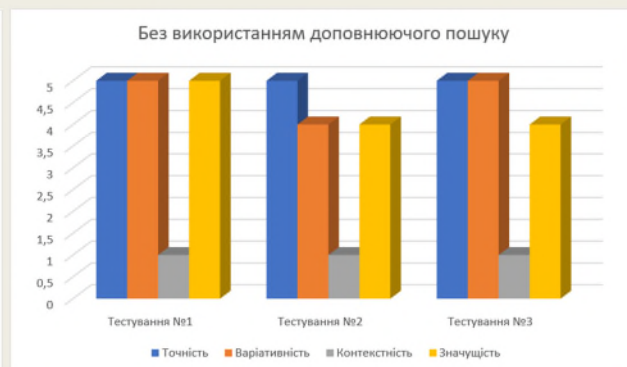
Визначення ефективності системи комунікації



Тестування №1 на тему «Відвідування ресторану» з використанням доповнюючого пошуку – 100%;

Тестування №2 на тему «Покупки у продуктовому магазині» з використанням доповнюючого пошуку – 90%;

Тестування №3 на тему «Побут» з використанням доповнюючого пошуку – 95%.

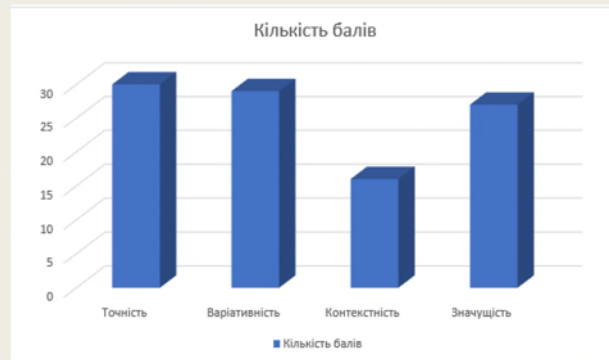


Тестування №1 на тему «Відвідування ресторану» без використання доповнюючого пошуку – 80%;

Тестування №2 на тему «Покупки у продуктовому магазині» без використання доповнюючого пошуку – 70%;

Тестування №3 на тему «Побут» без використання доповнюючого пошуку – 75%.

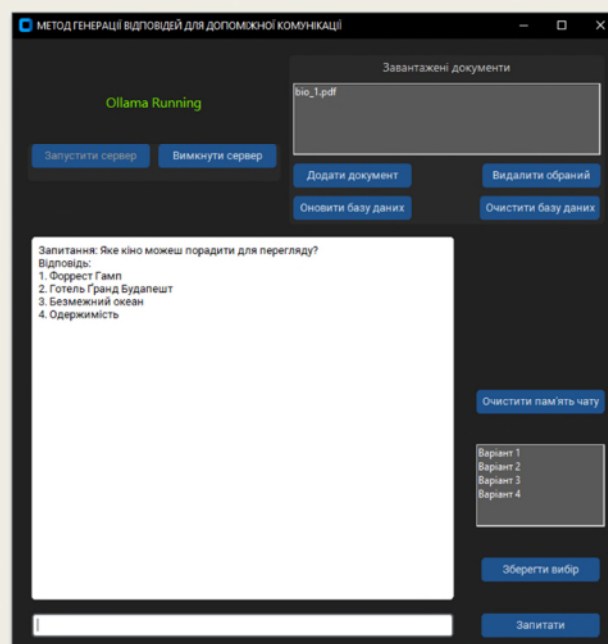
Визначення ефективності системи комунікації



Успішність критеріїв за результатами тестувань:

1. Точність відповідей – 100%;
2. Варіативність відповідей – 96,67%;
3. Контекстність відповідей – 53,33%;
4. Значущість відповідей – 90%.

Приклад роботи системи



Висновки

Кваліфікаційна робота магістра розв'язує науково-технічну задачу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації.

Проведено тестування розробленого методу за допомогою імітацій діалогів на теми «Відвідування ресторану», «Покупки в продуктовому магазині» та «Побут». Тестування проводилися з використанням персоналізованої інформації про користувача та без неї. Для діалогу «Відвідування ресторану» успішність з та без інформації становить – 100% та 80% відповідно, для «Покупки в продуктовому магазині» успішність з та без інформації становить – 90% та 70% відповідно, «Побут» успішність з та без інформації становить – 95% та 75% відповідно.

Запропонований метод має значний потенціал для підвищення ефективності допоміжної комунікації для осіб із вадами мовлення або слуху. Його практичне застосування сприяє автоматизації комунікації у різних життєвих ситуаціях, що допомагає користувачам краще інтегруватися у суспільство. Завдяки доповнюючому пошуку система дозволяє персоналізувати відповіді, враховуючи індивідуальні особливості та контекст діалогу. Крім того, метод оптимізує процес комунікації, забезпечуючи точність та різноманітність відповідей.

Наукова новизна роботи полягає у поєднанні генеративних мовних моделей із доповнюючим інформаційним пошуком (RAG) для вирішення завдань допоміжної комунікації. У межах дослідження була розроблена архітектура методу, що використовує індексацію даних, пошукові алгоритми та великі мовні моделі. Генерація відповідей була вдосконалена завдяки інтеграції зовнішнього інформаційного пошуку, що забезпечує вищий рівень контекстності та значущості відповідей. Також технологія адаптована до персоналізованих потреб користувачів, що робить цей метод ефективним для широкого спектра побутових завдань.

Отримані результати підтверджують перспективність методу для подальшого розвитку технологій допоміжної комунікації та його впровадження у практичні системи.

ДЯКУЮ ЗА УВАГУ!

Anti-Plagiarism v-15.258 Educational

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 8%**

ID: 159242 Назва: КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА на тему Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації Додано в БД: 2024-12-14 Автора: Віталій МЕДВЕДЧУК Керівники: Руслан БАГРІЙ Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	96814	1400	2413 (2%)	35 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Віталій МЕДВЕДЧУК

Співавтор:

Назва: Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації

Науковий керівник: Руслан БАГРІЙ, к.т.н., доцент

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1:1.4%

Коефіцієнт подібності 2:0.8%

Мікропробіли: 0

Заміна букв: 0

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2024-12-14 18:39:09.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

Дата

14.12.2024

експерт

Пешковський С.Р.

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА ДО ЗАХИСТУ ЗА РЕЗУЛЬТАТАМИ АНАЛІЗУ ЗВІТУ ПОДІБНОСТІ

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

Назва: Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації

Автор: Віталій МЕДВЕДЧУК

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доц. Руслан БАГРІЙ

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

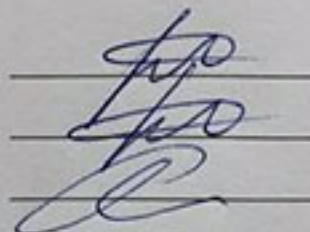
- 1) за програмою Anti-Plagiarism виявлені 1% є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення.
- 2) За програмою StrikePlagiarism виявлені за коефіцієнтом подібності КП1 – 1.4%, за коефіцієнтом подібності КП2 – 0,8%, що є запозиченнями, які розміщені в розділах аналізу існуючих технологій та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 1% та 1,4% і 0,8% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КН



Руслан БАГРІЙ

Руслан БАГРІЙ

Олександр БАРМАК



ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

зр. КНм-23-1 Віталія МЕДВЕДЧУКА за темою: «Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації»

1. Актуальність теми

Актуальність теми кваліфікаційної роботи обумовлена необхідністю вдосконалення засобів комунікації для людей з обмеженими можливостями мовлення. Запропонований метод генерації відповідей з доповнюючим інформаційним пошуком дозволяє значно підвищити якість та швидкість спілкування, сприяючи інклюзивності у суспільстві. Використання сучасних технологій, таких як великі мовні моделі та алгоритми Retrieval-Augmented Generation (RAG), забезпечує створення релевантних і контекстно збагачених відповідей, що особливо важливо для підвищення автономності людей із вадами мовлення.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Тема роботи повністю відповідає предметній області спеціальності 122 – Комп'ютерні науки та загальним вимогам до кваліфікаційної роботи магістра. Об'єктом дослідження є процес генерації текстових відповідей з доповнюючим інформаційним пошуком, а предметом – методи збору та аналізу даних, використання великих мовних моделей та векторних баз даних для допоміжної комунікації.

3. Професійні та особистісні якості магістранта

Віталій МЕДВЕДЧУК у процесі роботи продемонстрував високий рівень знань і навичок у галузі комп'ютерних наук. Студент відзначився відповідальністю, цілеспрямованістю та здатністю самостійно вирішувати складні задачі дослідження.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Робота виконана самостійно, академічного плагіату не виявлено, стосовно всіх запозичень наведено відповідні посилання на джерела.

5. Наукова новизна та оригінальність запропонованих підходів

Отримані такі результати: вдосконалено метод генерації відповідей для допоміжної комунікації шляхом інтеграції великих мовних моделей з доповнюючим інформаційним пошуком. Запропонована архітектура системи забезпечує високий рівень контекстності,

точності та адаптивності відповідей. Отримані результати оприлюднені на XVI всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2024», 15-16 листопада 2024 р., м. Хмельницький, Україна, доповідь на тему «Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації».

6. Ступінь оволодіння методами дослідження

Студент продемонстрував впевнене володіння сучасними методами дослідження, такими як векторний пошук, алгоритми подібності та методи генерації тексту на основі великих мовних моделей.

7. Повнота та якість розкриття теми роботи

Мета роботи повністю розкрита. Отримані результати підтверджують наукову обґрунтованість положень, а також досягнення всіх поставлених завдань.

8. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу

Робота відзначається логічною структурою, послідовністю викладу та високим рівнем аргументованості. Використаний стиль відповідає сучасним стандартам наукового письма, що забезпечує легкість сприйняття матеріалу.

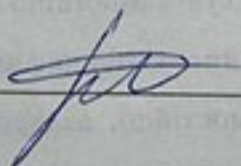
9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Розроблений метод має значний потенціал для практичного застосування у створенні систем допоміжної комунікації для людей з вадами мовлення чи слуху. Система може використовуватися у різних соціальних, професійних та освітніх контекстах, сприяючи інтеграції цих людей у суспільство.

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

Кваліфікаційну роботу магістра Віталія МЕДВЕДЧУКА рекомендовано до захисту. Вважаю, що робота заслуговує на оцінку «відмінно».

Науковий керівник _____



к.т.н., доц. Руслан Багрій



ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

гр. КНм-23-1 Віталія МЕДВЕДЧУКА за темою: Метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації

1. Актуальність обраної теми

Розробка методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації дозволяє значно покращити якість спілкування для людей з обмеженими можливостями мовлення. Запропонований метод допомагає усунути бар'єри комунікації та мінімізує залежність від сторонньої допомоги. Тому робота, виконана автором, є актуальною та перспективною для підвищення інклюзивності та покращення якості життя людей з особливими потребами.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Тема кваліфікаційної роботи повністю відповідає спеціальності 122 «Комп'ютерні науки». Виконані завдання для допоміжної комунікації відповідають сучасним напрямкам комп'ютерних наук. Робота відповідає вимогам до наукових досліджень і має практичне значення для покращення комунікації людей з обмеженими можливостями мовлення.

3. Повнота розкриття мети та завдань дослідження

В роботі автор повністю розкриває мету дослідження та поставленні в межах теми завдання.

4. Наявність наукової новизни

У кваліфікаційній роботі представлена наукова новизна та інновації, що відповідають спеціальності 122 «Комп'ютерні науки». Запропоновано метод генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації, який дозволяє підвищити точність та релевантність відповідей. Отримані результати мають наукове та практичне значення й були оприлюднені на науково-практичній конференції «Актуальні проблеми комп'ютерних наук – 2024» та у «Віснику Хмельницького національного університету».

5. Зміст кожного розділу роботи

Робота складається з чотирьох розділів.

У першому розділі виконано аналіз предметної області та сучасних підходів до генерації відповідей для допоміжної комунікації, зокрема технологій великих мовних моделей та Retrieval-Augmented Generation.

Другий розділ присвячено розробці методу генерації відповідей з доповнюючим інформаційним пошуком для допоміжної комунікації. Описано архітектуру методу, етапи індексації даних, пошуку інформації та генерації відповідей.

У третьому розділі детально розглянуто реалізацію системи комунікації на основі розробленого методу. Описано вибір платформ, технологій, бібліотек та компоненти системи.

У четвертому розділі проведено тестування системи комунікації та дослідження ефективності методу генерації відповідей. Проаналізовано результати тестування та коректність роботи системи з користувачами.

6. Ступінь розкриття теми роботи

Тема кваліфікаційної роботи повною мірою розкрита та обгрунтована. Проведено детальний аналіз актуальності, сучасних підходів і відомих досліджень у сфері генерації відповідей з доповнюючим інформаційним пошуком. Поставлені завдання виконані в повному обсязі, а результати дослідження проаналізовані та підтверджені тестуванням системи, що демонструє ефективність запропонованого методу.

7. Якість оформлення кваліфікаційної роботи

Оформлення роботи відповідає необхідним вимогам та стандартам, які ставляться до кваліфікаційних робіт. Текст написаний чіткою літературною мовою, а структура роботи відповідає встановленим нормам.

8. Недоліки кваліфікаційної роботи

Разом з тим, у роботі є певні недоліки. Зокрема, не розглянуто можливість застосування запропонованого методу на різноманітних наборах даних з іншими мовами чи діалектами. Також варто було б більш детально описати процес інтеграції запропонованої системи з реальними комунікаційними платформами.

Проте ці недоліки не впливають на загальну позитивну оцінку роботи..

9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота.

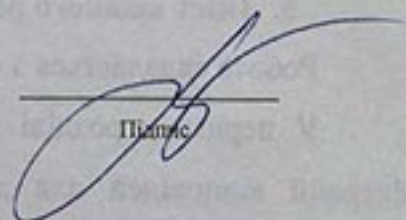
Враховуючи високий рівень виконання роботи, відповідність поставленим завданням та вимогам, кваліфікаційна робота може бути допущена до захисту.

Рекомендована оцінка – «Відмінно».

Опонент (прізвище, імя, по батькові, посада, місце роботи)

Говорухинко Т.О., декан фін ХНУ

«16» 12 2024 р.


Підпис