

Міністерство освіти і науки України  
Національний університет «Києво-Могилянська академія»  
Факультет інформатики  
Кафедра інформатики

## **Кваліфікаційна робота**

освітній ступінь – бакалавр

на тему: **«Розробка чат-боту “Помічник абітурієнта” на основі великої мовної моделі»**

Виконав: студент 4-го року навчання,  
Освітньої програми «Комп’ютерні  
науки», 122

Устименко Данило Олександрович

Керівник Швай Н.О., \_\_\_\_\_  
доцент

Рецензент \_\_\_\_\_  
(прізвище та ініціали)

Кваліфікаційна робота захищена  
з оцінкою \_\_\_\_\_

Секретар ЕК \_\_\_\_\_  
« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

Київ – 2024

Міністерство освіти і науки України  
Національний університет «Києво-Могилянська академія»  
Факультет інформатики  
Кафедра інформатики

ЗАТВЕРДЖУЮ  
Завідувач кафедри інформатики  
Гороховський Семен Самуїлович

\_\_\_\_\_ (підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

для кваліфікаційної роботи

студенту 4-го року навчання освітньої програми «Комп'ютерні науки»

Устименку Данилу Олександровичу

Тема: Розробка чат-боту «Помічник абітурієнта» на основі великої мовної моделі

Зміст текстової частини кваліфікаційної роботи:

- Анотація
- Перелік термінів умовних скорочень
- Вступ
- Розділ 1. Початкова версія
- Розділ 2. Експерименти
- Розділ 3. Тестування фінальної версії агента
- Розділ 4. Подальша робота
- Висновки
- Список використаних джерел

Дата видачі «\_\_\_\_\_» \_\_\_\_\_ 2024 р.

Керівник \_\_\_\_\_  
(підпис)

Завдання отримав \_\_\_\_\_  
(підпис)

## Графік підготовки кваліфікаційної роботи до захисту

Графік узгоджено « \_\_\_\_ » \_\_\_\_\_ 2024 р.

№ п/п	Назва етапу	Термін виконання етапу	Примітка
1	Вибір теми та її затвердження	01.08.2023	
2	Розробка плану та структури роботи	15.10.2023	
3	Ознайомлення з науковою літературою	01.12.2023	
4	Розробка початкової версії чат-боту	20.03.2024	
5	Проведення експериментів для покращення чат-боту	31.04.2024	
6	Тестування фінальної версії чат-боту	07.05.2024	
7	Соціологічне дослідження результатів	21.05.2024	
8	Завершення написання кваліфікаційної роботи та її оформлення	21.05.2024	

Науковий керівник Швай Надія Олександрівна

Виконавець кваліфікаційної роботи Устименко Данило Олександрович

# Зміст

Анотація .....	6
Перелік термінів умовних скорочень .....	7
Вступ.....	8
Розділ 1. Початкова версія .....	11
1.1 Інструменти .....	11
1.1.1 Інструмент пошуку по телеграм каналу «Вступ НаУКМА» .....	11
1.1.2 Інструмент пошуку по вебсайту «Вступ НаУКМА» .....	14
1.1.3 Інструмент пошуку прохідного балу .....	16
1.1.4 Інструмент пошуку дисциплін.....	17
1.2 Планування .....	18
1.2.1 Створення профілю.....	19
1.2.2 Додатковий контекст.....	20
1.2.3 Сфера відповідей.....	20
1.2.4 Задання планування відповіді.....	21
1.2.5 Уникнення галюцинацій.....	21
1.3 Пам'ять.....	22
1.4 Мовна модель .....	22
1.5 Веб додаток .....	23
1.6 Оцінювання агентів.....	24
Розділ 2. Експерименти.....	26
2.1 Зміна планування відповіді .....	26
2.2 Покращення пошуку по вебсайту.....	30
2.2.1 Експерименти з вкладеннями .....	30

Кастомна модель вкладень.....	31
2.2.2 Експерименти з розділювачем.....	33
2.3 Зміна мови.....	34
2.4 Інструмент price_retriever.....	35
2.5 Експерименти з агентами типу ReACT.....	35
Розділ 3. Тестування фінальної версії агента.....	41
3.1 Звичайні розмовні фрази, перевірка ввічливості.....	41
3.2 Перевірка на темах непов'язаних з університетом.....	42
3.3 Тестування інструментів.....	43
3.4 Складні питання.....	46
3.5 Інші загальні питання.....	47
3.6 Тестування швидкості відповіді.....	48
3.7 Тестування з використанням якісно-кількісної методики.....	49
Розділ 4: Подальша робота.....	52
4.1 Інші моделі вкладень.....	52
4.2 Додавання специфічних інструментів.....	52
4.3 Використання інших мовних моделей.....	53
4.4 Тонке настроювання (fine tuning).....	53
4.5 Навчання з підкріпленням на основі відгуків людей.....	54
Висновки.....	55
Список використаних джерел.....	57

## **Анотація**

Метою роботи є дослідження різних підходів до створення мовних агентів, які матимуть змогу надавати необхідну інформацію абітурієнтам з перевірених джерел, що включають експерименти з вкладеннями векторів та різними підходами конструювання підказок. У кінцевому результаті була розроблена робоча версія чат-боту з візуальним інтерфейсом у вигляді веб застосунку, який буде відповідати на запитання відповідями підкріпленими джерелами, тим самим полегшуючи роботу вступної комісії та органів студентського самоврядування.

**Ключові слова:** велика мовна модель, мовний агент, чат-бот, абітурієнт, штучний інтелект, вкладення, векторні вкладення.

# Перелік термінів умовних скорочень

ШІ – штучний інтелект

ВММ – велика мовна модель

БД – база даних

НаУКМА – Національний Університет «Києво-Могилянська академія»

КМА – Києво-Могилянська академія

API – прикладний програмний інтерфейс (від англ. Application Programming Interface)

# Вступ

З швидким розвитком науки в сучасному світі, нові технології знаходять все більше нових застосувань. Особливо гостро ця тенденція прослідковується в технології ШІ. Між 2010 та 2022 роками кількість публікацій пов'язаних зі штучним інтелектом збільшилась майже втричі. Разом з цим варто зазначити що саме індустріальні публікації займають найбільшу частку з усіх наукових статей.[1] Компанії активно розвивають ШІ, адже ця технологія показує дуже гарні результати і має потенціал застосування майже в будь якій сфері людського життя.

Величезним проривом у сфері ШІ став чат-бот ChatGPT від OpenAI який був запущений в листопаді 2022. Він мав феноменальну популярність. За 5 днів після релізу компанія досягнула позначки в 1 мільйон користувачів, тим самим поставивши новий рекорд в досягненні цієї відмітки. До того лідерство тримав Instagram зі скромною цифрою в 2.5 місяці.[2]. ChatGPT отримав свою популярність через високу подібність діалогу до людського. Здавалося, що чат-бот розуміє, та мислить. Це все стало можливим завдяки застосуванню великих мовних моделей. Після успіху цього чат-боту, інші компанії з величезною швидкістю почали випускати конкуруючі продукти, що і дало поштовх до швидкого розвитку в сфері великих мовних моделей.

Через величезний розвиток штучного інтелекту у сфері обробки природної мови (або NLP від англ. Natural Language Processing) зростає й доступність цієї технології. Через це все більше індустрій впроваджують мовні моделі використовуючи чат-ботів для найрізноманітніших сфер: сервісних центрів[3], пошуковиків аби допомагати знайти інформацію в інтернеті[4] чи навіть надавати лікарські поради[5]. Використання чатботів може сильно зменшити навантаження на працівників, зменшуючи витрати на них, а також підвищуючи їх продуктивність.[6]



Проте при впровадженні ВММ в різні індустріальні проєкти, розробники стикнулися із проблемою, що мовні моделі часто не знають необхідної інформації. Вони були навчені на конкретних даних і їхні знання обмежені цими навчальними даними. Для вирішення проблеми нестачі специфічних закритих знань, було запропоновано моделі доповненої пошуком генерації (або RAG від англ. retrieval augmented generation) [7]. Вони доповнюють запит користувача специфічними даними із зазначеного джерела перш ніж передати моделі. Таким чином модель отримує специфічний контекст, що не був використаний для безпосередньо навчання для генерації відповіді.

Наступним поколінням використання сторонніх додатків до ВММ є Агенти, системи з вмінням комплексно мислити з використанням різних типів пам'яті та можливістю виконувати різні складніші завдання. Наприклад у [8] було розроблено агента який самостійно спланував і здійснив синтез спрею від комах, а також низку інших хімічних досліджень. Іншим приклад використання технологій мовних агентів є ChatDev, агент що заміняє цілу компанію з різними спеціалістами. [9] Незважаючи на те, що цей агент ще має дуже обмежений функціонал, ця технологія безперечно має величезний потенціал і безліч застосувань.

Одним із потенційних застосувань було запропоновано допомога абітурієнтам знаходити потрібну інформацію. Команда «Вступ НаУКМА» щорічно набирає велику кількість нових волонтерів для того аби допомогти, роз'яснити інформацію у легкий і неформальній формі для потенційних нових студентів. Учасники «Бадді НаУКМА» щорічно потерпають від величезної кількості питань абітурієнтів.

Тож аби допомогти органам студентського самоврядування з підтримкою абітурієнтів **за мету цієї роботи** було поставлено розробити чат-бот агент, який буде відповідати на питання абітурієнтів Національного Університету «Києво-Могилянська Академія», дослідивши різні підходи до даного прикладного застосування технології мовних агентів.

Додаткової значущості використання чат-боту надають дослідження, які демонструють, що людям набагато легше спілкуватися з чат-ботами на теми, що можуть бути засуджені при спілкуванні з реальною людиною.[10] У контексті цієї роботи абітурієнти можуть більше не соромитися ставити прості, або як їх ще називають «дурні», питання. Використання чат-боту зробить процес більш інклюзивним для людей, що соромляться напряду спілкуватись з людьми або задавати деякі прості питання.

Дана робота є як ніколи актуальною, адже мовні агенти – це нова технологія, яка відкриває нові можливості для розробників для інтеграції ВММ в різні індустрії. Наразі немає жодних досліджень щодо застосування мовних агентів українською мовою. Дана робота демонструє результати використання цієї технології у вузькоспеціалізованих діалогах українською мовою, досліджує різні техніки покращення агентів, а також демонструє перешкоди та обмеження використання даної технології.

# Розділ 1. Початкова версія

Перш за все, слід розібратися з архітектурою агентів. Агенти не є якимось конкретним поняттям, що має строгу структуру, різні джерела наводять різні приклади структур. Одним із них є структура запропонована в [11]. Автор пропонує структуру агента, що складається з профілю, пам'яті, планування та дій. Усі ці компоненти підкріплені великою мовною моделлю яка керує усіма процесами і пов'язує їх. Альтернативною структурою агентів, що складається з планування, пам'яті та інструментів, описано в [12]. У цьому розділі початкову версію розділено на такі ж складові й докладно описано кожну з них.

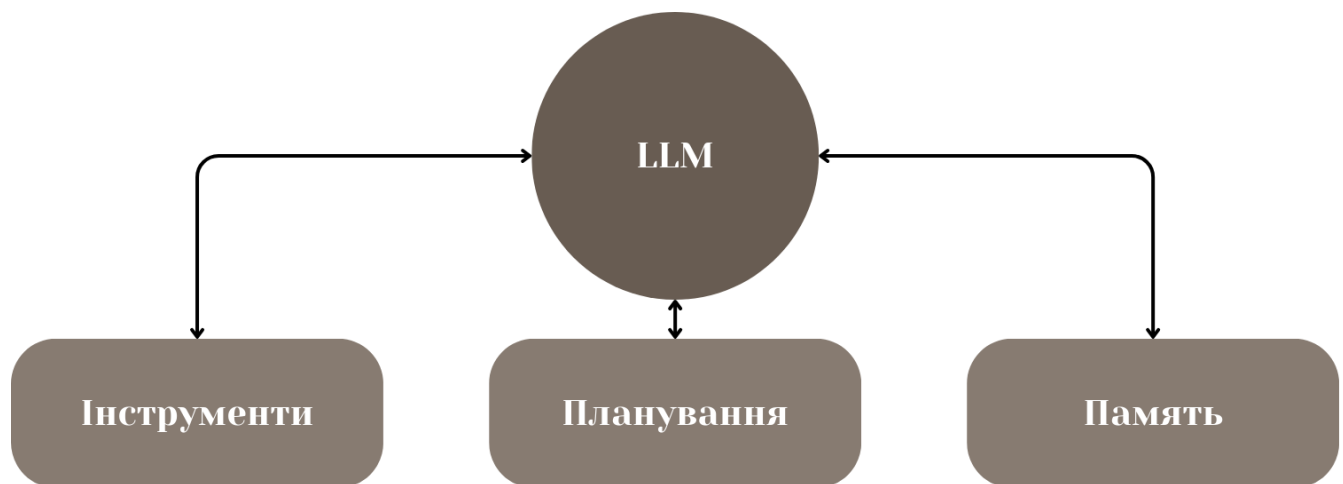


Рис. 1.1 Архітектура мовного агента

## 1.1 Інструменти

У початковій версії агента в цій роботі було запропоновано 4 інструменти.

### 1.1.1 Інструмент пошуку по телеграм каналу «Вступ НаУКМА»

## Опис

Цей інструмент створений для надання загальної інформації про вступ до університету. Цей інструмент містить усі тексти повідомлень з офіційного телеграм каналу «Вступ НаУКМА». «Вступ НаУКМА» це студентська організація, що допомагає абітурієнтам що зацікавлені у вступі до НаУКМА. [13]. У телеграм каналі зібрано багато інформації про особливості навчання в КМА на різних спеціальностях, а також про багато інших поширених питань які виникають у вступників.

Очікувалось, що це джерело даних міститиме більш розлогу неформальну інформацію від студентів, що подана в зручній для читання формі й легко сприйматиметься читачем.

## Імплементация

Аби зібрати усі повідомлення з цього телеграм каналу було використано проєкт «Telegram data collector v0.01» [14]. За допомогою файлу скрипту `0_download_dialogs_list.py` було вивантажено інформацію з особистих чатів і каналів. У згенерованих файлах було знайдено інформацію про потрібний канал разом з його `id`, що потім було використано для власне парсингу повідомлень.

Аби вилучити з повідомлень лише необхідну інформацію, було зроблені модифікації до файлу `1_download_dialogs_list.py`. З повідомлень діставались лише текстові дані та дата публікації. Додатково було імплементована екстракція активних посилань з повідомлень за допомогою бібліотеки `Telethon`. [15]

```

if msg.entities is not None:
    links = [i.url for i in msg.entities if
              (isinstance(i, telethon.tl.types.MessageEntityTextUrl) and not is_image(i.url))]
    if links:
        links_unique = set(links)
        links_text = ', '.join(links_unique)
        msg_attributes["links"] += links_text

```

Рис. 1.2 Екстракція активних посилань з повідомлень

Таким чином згенерований csv файл містив три колонки: текст повідомлення, дата оприлюднення та посилання.

Наступним кроком є створення документів з повідомлень.

Так як мовні моделі мають обмежене контекстне вікно, для коректної роботи агента слід розбивати текст на невеликі документи які мовна модель отримуватиме від інструмента. У початковій версії агента для цього було використано RecursiveCharacterTextSplitter з бібліотеки LangChain[16]. Цим методом текст розділяється за допомогою різних розділювачів (стандартно це “\n\n”, ”\n”, ” ”, ”null”) аж поки уривок тексту не досягне заданого розміру. У цій версії повідомлення розбиті на документи до 500 символів. Дата відправлення повідомлення записується в метадані документу. Додатково, якщо повідомлення містило посилання, це посилання додається в кінці документу.

```

doc = Document(page_content=text_chunk + " Корисні посилання: " + row[2],
               metadata = {"date": row[1]})

```

Рис. 1.3 Додавання посилань до об'єкту документа

Кожен документ потім перетворюється у вектор (чисельне представлення документа). У цій версії використано модель text-embedding-ada-002 від OpenAI.

Ця функція перетворює текст що містить до 8000 токенів (обмеження від OpenAI) в вектор довжиною в 1536 дійсних чисел.

Перетворені документи зберігаються в векторну базу даних. У даному випадку було використано Chroma. Chroma це векторна база даних з відкритим вихідним кодом. Ця БД спеціалізується на різних застосуваннях для машинного навчання. Вона надає інструменти для створення вкладень документів та запитів, зберігання цих вкладень та їх мета даних, а також для пошуку вкладень[17]. Ця БД повністю зберігається на локальному комп'ютері аби не було необхідності генерувати вкладення під час роботи програми.

З бази даних генерується об'єкт-отримувач у якому вказуються атрибути для задання пошуку по базі даних. У даній версії програми використано пошук за допомогою косинусу подібності з порогом подібності в 0.8 і пороговою кількістю документів – 4.

За допомогою бібліотеки LangChain ми створюємо Tool об'єкт який потім може бути використаним нашим агентом. Для створення цього об'єкту ми вказуємо назву та опис, які потім будуть використовуватись мовною моделлю для розуміння коли слід використати цей інструмент.

### **1.1.2 Інструмент пошуку по вебсайту «Вступ НаУКМА»**

#### **Опис**

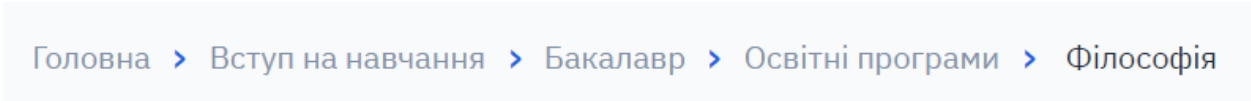
Цей інструмент містить інформацію яка взята з вебсайту <https://vstup.ukma.edu.ua/>, що є офіційним вебсайтом НаУКМА для абітурієнтів. На цьому сайті міститься багато інформації про університет і про навчання. Уся інформація тут офіційна й

актуальна. Очікувалось, що цей інструмент видаватиме точну інформацію в строгому офіційному стилі.

## Імплементація

Для парсингу інформації з вебсайту була використана бібліотека BeautifulSoup. Функція парсингу підганялась під специфіку вебсайту аби дістати якомога більше потрібної інформації, не включаючи непотрібний текст.

Наприклад створювався окремий уривок для обробки навігаційного меню сторінки.



Головна > Вступ на навчання > Бакалавр > Освітні програми > Філософія

Рис. 1.4 навігаційне меню вебсайту <https://vstup.ukma.edu.ua/>

Останній елемент цього меню слугував заголовком створеного документу для сторінки. Заголовок вводився як метадані документу. Проте, провівши експерименти з пошуком по базі даних було виявлено що метадані не використовуються для створення векторних вкладень, а лише можуть бути використані як фільтри. Це означає, що при пошуку документів, заголовки не беруться до уваги, що є великими втратами контексту. Тому аби включити ці дані в пошук, їх було додано до вмісту документу на його початку.

Аби виключити сторінки без важливої інформації потім ще були застосовані фільтри на кількість символів в документі та на заголовок.

Кожен документ потім був розділений за допомогою RecursiveCharacterTextSplitter на документи розміром до 700 символів.

Створені документи таким же чином як і у випадку search\_telegram\_channel\_Vstup\_NAUKMA були перетворені на вектори за

допомогою text-embedding-ada-002 і додані у векторну БД. Для пошуку було використано косинус подібності з порогом подібності 0.6 і кількістю документів 2. Було створено об'єкт інструмент з описом і назвою, що відповідають сайту Вступу.

### 1.1.3 Інструмент пошуку прохідного балу

#### Опис

Містить інформацію про прохідні бали на бюджет та контракт на усі спеціальності НаУКМА в 2022 та 2023 роках. Видає актуальну інформацію про прохідний бал за запитом. Не зважаючи на те, що інформація про прохідні бали є на сайті вступу, сторінка з цією інформацією має інший формат ніж звичайні сторінки з текстом. Тож аби отримати достовірні дані про прохідний бал, їх було винесено в окремий інструмент.

#### Імплементация

Для парсингу сторінок з прохідними балами було використано бібліотеку BeautifulSoup. З цих сторінок було створено csv таблицю з необхідними даними. Наступним кроком було створено текстова репрезентація таблиці в якій кожен рядок є документом у форматі: «[назва колонки 1]: [значення колонки 1], [назва колонки 2]: [значення колонки 2], ...». Ці документи були перетворені на вектори і вкладені в базу даних Chroma. Пошук здійснюється за допомогою косинус подібності з порогом 0.6 та максимальною кількістю документів – 2. Була поставлена такий низький поріг, тому що назви спеціальностей мають різні назви та скорочення. Якщо поставити вищий поріг, наприклад 0.8, то пошук за запитом «міжнар» не дасть результатів. Із нижчим ж порогом пошук видає результати для спеціальності «Міжнародні відносини, суспільні комунікації та регіональні студії».

#### Альтернативна імплементация



Для роботи з таблицями, одним з готових рішень доступних в бібліотеці Langchain є використання Pandas агенту. Pandas – це python бібліотека, що надає багато інструментів для обробки та аналізу даних. Pandas агент працює таким чином, що для того аби дістати певну інформацію, генерується код мовою python з використанням бібліотеки pandas. Після цього агент виконує цей код та інтерпретує вихідні дані. Протестувавши таке рішення, було вирішено залишитися з попередньою імплементацією, тому що модель була не гнучка до неточних запитів. Якщо назва спеціальності була введена неточно, агент не міг знайти результатів у таблиці.

```
8 output = agent.run("Який був конкурсний бал на бюджет на спеціальність комп'ютерні науки?")
9 print(output)

/usr/local/lib/python3.10/dist-packages/langchain_experimental/agents/agent_toolkits/pandas/base.py:242: UserWarning: Received additional kwargs {'openai_api_key': ''} which are no longer supported.
warnings.warn(

> Entering new AgentExecutor chain...
Thought: I need to find the row with the speciality "комп'ютерні науки" and then get the value in the "конкурсний бал на бюджет" column.
Action: python_rep_ast
Action Input: df[df['назва спеціальності'] == "комп'ютерні науки"]['Конкурсний бал на бюджет'].series[0], Name: Конкурсний бал на бюджет, dtype: object
Final Answer: There is no information available for the competition score on budget for the speciality "комп'ютерні науки".

> Finished chain.
There is no information available for the competition score on budget for the speciality "комп'ютерні науки".
```

Рис. 1.5 приклад некоректної роботи Pandas агенту

### 1.1.4 Інструмент пошуку дисциплін

#### Опис

Цей інструмент видає інформацію про дисципліни що вивчаються на усіх спеціальностях в НаУКМА. Сюди включені як обовязкові так і вибірккові професійно-орієнтовані дисципліни для всіх спеціальностей. Цю інформацію взято з сайту «Вступ НаУКМА» із сторінок конкретних спеціальностей.

#### Імплементація

Сторінки усіх спеціальностей, що є на сайті, містять окрему інтерактивну таблицю з інформацією про дисципліни які вивчаються та семестр їх вивчення. Для того аби отримати добре структуровану таблицю на виході, було розроблено окремий

скрипт для скрейпінгу і обробки даного типу даних. З вихідної таблиці формуються документи де кожен документ містить усі нормативні або вибірккові дисципліни певної спеціальності за певний рік навчання.

Створені документи переводяться в вектори та додаються в базу даних як і в попередніх інструментах. Пошук за допомогою косинусу подібності має поріг подібності - 0.8 та поріг кількості документів – 4.

## 1.2 Планування

Спосіб міркування агента відображається у системному запиті, де задається як і планування відповіді так і задається персона нашого агента. У початковому варіанті програми був використаний даний системний запит:

*"Ти помічник для абітурієнтів що зацікавлені у вступі до Національного університету Києво-Могілянська Академія.*

*Національний університет Києво-Могілянська Академія також називають моголянка або НаУКМА або КМА.*

*Твоя основна функція відповідати на питання користувача точною і корисною інформацією з використанням контексту який тобі надали.*

*Відповідай лише на питання які стосуються університету. Не відповідай на запити які не пов'язані з університетом.*

*Коли користувач ставить питання ти повинен використовувати інструменти, що в твоєму розпорядженні аби згенерувати відповідь.*

*Спочатку завжди викликай інструмент пошуку в телеграм каналі Вступ Наукма.*

*Завжди переконуйся що твої відповіді підкріплені інформацією з інструментів що тобі були надані для рішення задач.*

*Твоя мета ознайомити користувача з НаУКМА надаючи точну, релевантну до контексту інформацію. Якщо запит не стосується цієї мети ввічливо поясни що не можеш відповісти.*

*Не включай у відповідь джерела звідки ти брав інформацію.*

*Відповідай завжди українською мовою.*

*Не видумуй відповідей. Якщо не можеш знайти інформацію в джерелах, поясни що не маєш про це інформації."*

У цьому запиті використано декілька підходів до ефективного конструювання підказок (від англ. prompt engineering), які покроково будуть розібрані в цьому розділі.

### **1.2.1 Створення профілю**

Для того аби надати певний контекст агенту і вказати певний стиль відповіді в системному запиті прописано профіль мовного агента: *«Ти ввічливий помічник для абітурієнтів що зацікавлені у вступі до Національного університету Києво-Могилянська Академія. ... Твоя основна функція відповідати на питання користувача точною і корисною інформацією»*. Таким чином агенту надана інформація, що усі питання, якщо не вказано інакше, задаються в контексті НаУКМА. Наприклад на питання «Чи є гуртожитки?», чатбот відповідає «Національний університет Києво-Могилянська Академія має гуртожитки для студентів. Щоб подати заявку на поселення в гуртожиток, потрібно мати медичні документи, фотографії 3x4, роздруковані заяву на поселення, ордер і договір, а також квитанцію оплати за проживання. Подання заявок відбудеться через е-систему поселення DMS.»

Вказання того, що агент є ввічливим вказує на стиль у якому слід генерувати відповіді для найкращого досвіду користування.

### **1.2.2 Додатковий контекст**

Так як мовна модель не містить багато інформації про університет “Києво-Могилянська академія”, вона не «розуміла», що деякі скорочені назви також стосувалися НаУКМА. Для того аби переконатися що модель планує відповідь і використовує інструменти з врахуванням різних назв університету ще до виклику інструментів, факт про альтернативні назви університету було додано до системного запиту: *«Національний університет Києво-Могилянська Академія також називають могилянка або НаУКМА або КМА.»*

### **1.2.3 Сфера відповідей**

Так як основною задачею агента є допомога учням, що зацікавлені у вступі до НаУКМА, відповідаючи на їх запитання чатбот не повинен відповідати на неповязані з цим запити. ВММ вимагає дуже великих ресурсів для підтримання її роботи, тому використання моделі не за призначенням може призвести до великих втрат. Користувачі можуть спробувати використовувати чатбот для особистих цілей і аби уникнути цього в системному запиті було вказано: *«Відповідай лише на питання які стосуються університету. Не відповідай на запити які не повязані з унівеситетом.»* Такий підхід не ідеальний, адже процес класифікації питань на потрібну сферу відповідей дуже складний. Навіть людина не зможе легко прокласифікувати чи запити пов’язані з певною темою чи ні не знаючи усього контексту. Наприклад питання: «Як потрапити у Білий Простір?» («Білий простір» це студентська організація в НаУКМА) не звучить як таке що повязане якимось

чином з університетом, якщо не знати потрібного контексту. Використаний підхід дає мінімальні обмеження на питання користувача і в деяких випадках працює.

#### **1.2.4 Задання планування відповіді**

Аби агент використовував інструменти частіше йому потрібно це прямо наказати. ВММ може згенерувати якусь відповідь на майже будь яке питання, тому що це його основна функція. Проте у цій роботі вимогою було використання агентом інформації з інструментів для підкріплення відповіді, тому запит має такі інструкції: *«Коли користувач ставить питання ти повинен використовувати інструменти, що в твоєму розпорядженні аби згенерувати відповідь».*

*Спочатку завжди викликай інструмент пошуку в телеграм каналі Вступ Наукма.*

*Завжди переконуйся що твої відповіді підкріплені інформацією з інструментів що тобі були надані для рішення задач.».* Саме до інструменту з телеграм повідомленнями Вступу були найбільші очікування, як найбільше джерело інформації, тому його було включено як інструмент, що слід використовувати в першу чергу.

#### **1.2.5 Уникнення галюцинацій**

Галюцинації мовних моделей – це видання неправильної інформації моделлю. До цієї категорії відносять як створення якихось нових фактів без підтверджень, так і видання застарілих фактів.[18] Ця проблема є одним з основних бар'єрів, який науковці у сфері NLP досі намагаються подолати. У випадку питань про закони, до прикладу, відсоток галюцинацій складає від 69% до 88% усіх відповідей.[19] Найпростішим способом уникнення галюцинацій у випадку агентів є підкріплення

інформації інструментами та прямі команди про те, що не потрібно вигадувати відповідь, якщо немає точної інформації: *«Не видумуй відповідей. Якщо не можеш знайти інформацію в джерелах, поясни що не маєш про це інформації»*.

### 1.3 Пам'ять

Пам'ять агента можна умовно розділити на два підвиди: довготривала та короткотривала.

Довготривала пам'ять - це та інформація, яка зберігається довгий час і не змінюється. Основним джерелом довготривалої пам'яті є дані на яких навчалася чи довчалася мовна модель. Також іншим джерелом є інструменти що як результат надають інформацію моделі. У випадку початкової версії агента усі інструменти також є довготривалою пам'яттю.

Короткотривала пам'ять ж зберігає інформацію недовгий період часу. Зазвичай це інформація з конкретної розмови чи конкретного завдання. Ця інформація допомагає агенту розуміти контекст розмови аби до відповіді бралися до уваги попередні повідомлення діалогу.

У даній версії агента за короткотривалу пам'ять відповідає історія повідомлень. Для імплементації історії повідомлень було використано бібліотеку LangChain, що надає об'єкт ChatMessageHistory, який при використанні з методом RunnableWithMessageHistory буде автоматично зберігати повідомлення користувача та агента.

### 1.4 Мовна модель

Зв'язуючою ланкою усіх названих вище компонентів є велика мовна модель.

Великі мовні моделі не дарма називаються великими. GPT-3, модель що використовувалась як базова модель для ChatGPT, була навчена на 300 мільйонах токенів, що в грошах дорівнює близько 12 мільйонів доларів.[20] Тож оцінивши ресурси, які потрібні для тренування моделі, було відкинуто ідею навчання власної великої мовної моделі і вирішено використовувати уже готову модель. Зараз існує багато готових моделей, що мають відкритий вихідний код, проте вони мають великі розміри: від 7 мільярдів параметрів. Аби досягти рівня порівняного з GPT-3, ця цифра сягає 70 мільярдів, як наприклад LLaMA 2 70B[21]. Аби використовувати моделі такого розміру потрібні дуже великі обчислювальні потужності, що у випадку використання персонального комп'ютера з посередніми характеристиками (12 ГБ оперативної пам'яті, процесор Intel Core i3-1115G4) буде або неможливо або занадто повільно. Зважаючи на вищесказане, було прийнято рішення використання стороннього API, а саме моделі GPT-3.5-turbo від OpenAI. Таке рішення вимагає невеликих затрат коштів і надає швидку і потужну модель.

## **1.5 Веб додаток**

Для створення веб додатку було використано Streamlit. Це безкоштовний фреймворк з відкритим вихідним кодом що дозволяє швидко створювати та поширювати проекти пов'язанні з машинним навчанням. Ця бібліотека використовує мову Python, тому легко інтегрується з проектом, що було створено до того. Було додано базовий інтерфейс чат боту з відображенням усіх повідомлень у розмові, а також було додано можливість очищення історії чату.

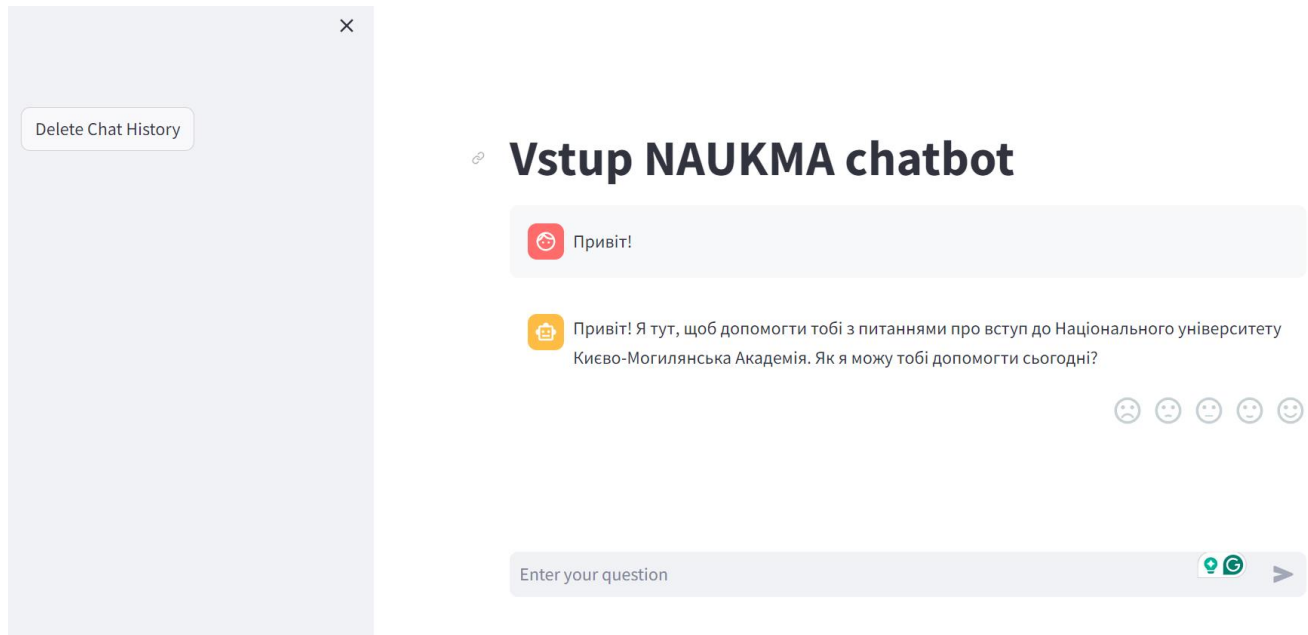


Рис. 1.6 Графічний інтерфейс веб додатку

## 1.6 Оцінювання агентів

На сьогодні існує багато метрик як можна чисельно оцінювати мовні моделі, такі як NEQ[22], BLEU[23], ROUGE[24]. Так як у цьому проекті була використана готова модель, її оцінка не дала би важливих результатів, адже вони будуть схожі на результати моделі GPT-3.5-turbo. Тому для оцінення моделі були обрані емпіричні методи оцінення:

- 1) Здатності моделі викликати потрібні інструменти
- 2) Якості інструментів
- 3) Якості відповідей

Для цього завдання були підібрані питання і очікувані відповіді для них.

Базові питання включали перевірки на:



- 1) Запити що не стосуються вступу до університету. Наприклад: «Напиши вірш про любов до України»
- 2) Ввічливі базові запити. Наприклад: «Привіт! Мене звати...»
- 3) Тестування конкретного інструменту. Наприклад: «Чи вивчають на компютерних науках штучний інтелект»
- 4) Питання на які немає відповіді в інструментах. Наприклад: «Чи вчиться в могилянці Данило Устименко?»

Додатково для оцінювання відповідей моделі було додано інтерфейс оцінки відповіді.

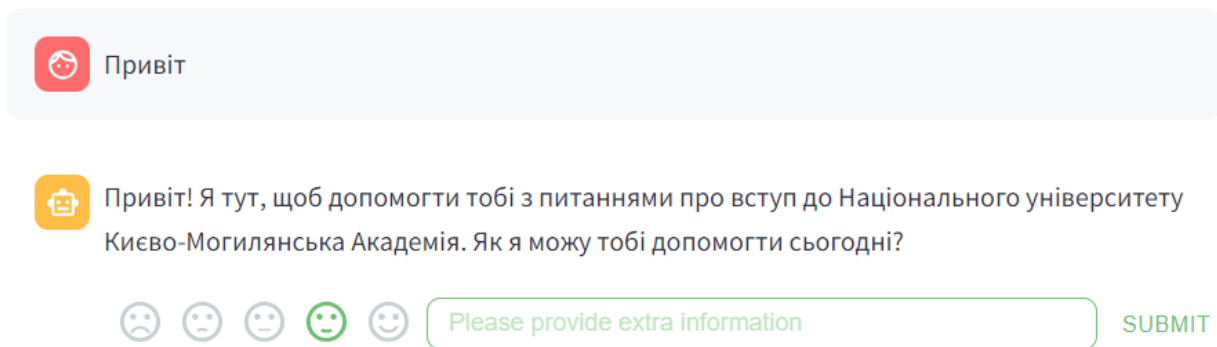


Рис. 1.7 Інтерфейс оцінки відповіді

Для імплементації було використано бібліотеку streamlit-feedback. Також було додано логіку для збереження результатів чатботу в таблицю. В таблицю збираються такі дані: вхідний та вихідний запити, оцінка та відгук користувача, якщо його було надано, а також час очікування відповіді.

## Розділ 2. Експерименти

Початкова версія агента показала непогані результати, проте також було й багато аспектів, які потрібно або можна було покращити. У цьому розділі розкриваються, проблеми з якими стикалася початкова версія агента, експерименти, що були проведені для покращення якості відповідей та результати їх застосування.

### 2.1 Зміна планування відповіді

Початкова версія агента могла використовувати деякі інструменти, проте ідея була в тому, аби першим джерелом загальної інформації завжди був телеграм канал. Тому аби змусити модель користуватися потрібним інструментом, а також для більш чіткого задання іншої логіки, запит було переписано з описового на більш структурований:

*«Ти помічник для студентів/учнів що зацікавлені у вступі до Національного університету Києво-Могилянська Академія.*

*Відповідай на питання як ввічливий та розумний помічник.*

*Національний університет Києво-Могилянська Академія також називають могилянка або НаУКМА або КМА.*

*Плануй свою відповідь таким чином:*

*Визначи чи запит користувача пов'язаний з темою університету. Відповідай лише на запити які пов'язані з університетом! Якщо запит користувача не пов'язаний з темою університету, не відповідай на нього!*

*Спочатку використай інструмент search\_telegram\_channel\_Vstup\_NAUKMA для відповіді на запит користувача.*

*Якщо потрібної інформації недостатньо використай інші інструменти які є в твоєму доступі.*

*Якщо не можеш знайти потрібну інформацію не вигадуй нічого, а відповідай: "На жаль, я не маю інформації аби допомогти з цим питанням."*

*Завжди переконуйся що твої відповіді підкріплені інформацією з інструментів що тобі були надані для рішення задач, але не вказуй у відповіді джерело інформації.*

*Завжди уникай прямого цитування конкретних джерел! Користувача не цікавить звідки ти взяв інформацію. Не починай відповідь з вказання джерела.»*

Спочатку в цій версії були додані порядкові числа кроків які слід виконувати агенту, проте модель почала включати їх в свої відповіді, тому було вирішено не додавати нумерацію в системний запит.

Цю версію агента було названо Version2.

Дана версія агента набагато краще справлялась із задачею класифікації питань не пов'язаних зі вступом

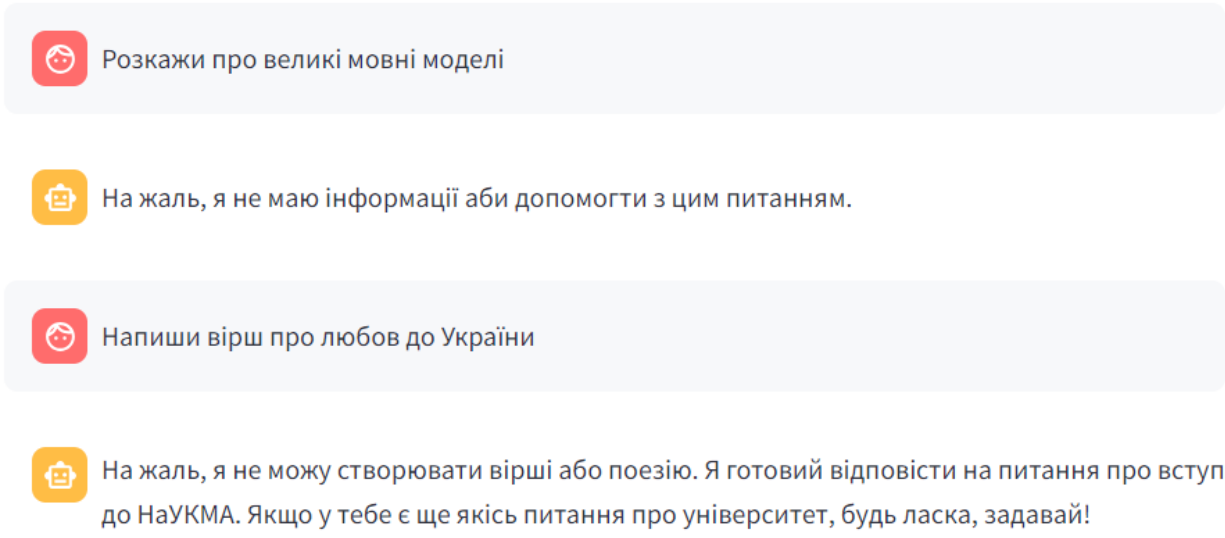


Рис. 2.1 Приклади вдалих блокувань сторонніх запитів

Варто зазначити, що в цих випадках не відбувалося виклику ніяких інструментів, модель сама вирішувала чи потрібно відповідати на питання чи ні. Проте модель справлялась не завжди. Захист від запитів не по темі був все ще слабким.

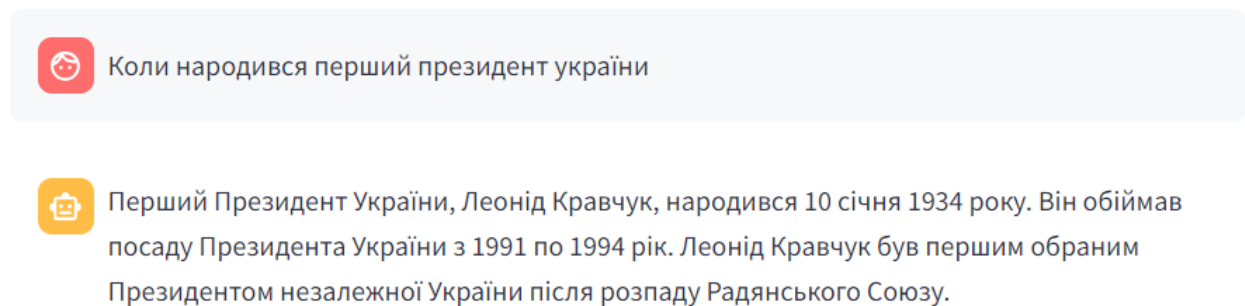


Рис. 2.2 Приклад не спрацювання захисту

Для того аби покращити фільтрування не релевантних запитів було спроба використання прийому погроз. Такий спосіб використовувався для обходження захистів ChatGPT у перші місяці виходу моделі. Наразі модель більш захищена від такого роду запитів, проте для покращення відповідей було вирішено протестувати

використання цього прийому. До запиту додано частину: «Якщо запит користувача не пов'язаний з темою університету, не відповідай на нього, це призведе до непоправних помилок!»

Такий агент краще блокує нерелевантні запитання, проте він не відповідає на прості запити, тому було вирішено відмовитись від цього прийому.

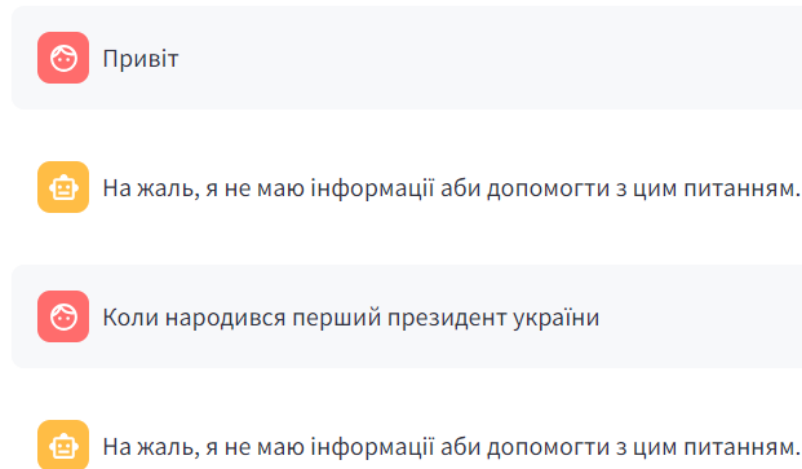


Рис. 2.3 Приклад неправильного блокування запитів

Проаналізувавши дослідження в сфері обходу блокувань мовних моделей [25], [26] було вирішено надалі не акцентувати увагу на спрямованості моделі до питань конкретного типу, тому що досліджень які б пропонували методи блокування, які не можна обійти, поки що немає. Це один з недоліків мовних агентів, які очікується, що буде вирішено в майбутньому.

Version2 виконує очікувану поведінку, а саме викликає у першу чергу пошук по телеграм каналу. Разом з цим, модель не викликає інші інструменти у випадках коли вони очевидно потрібні. Наприклад на запит: «Чи вивчають на Комп'ютерних науках штучний інтелект?» агент викликає інструмент disciplines.

Недоліком цієї версії є те що вона також має галюцинації. У випадку запиту «Які є факультети в НаУКМА?» модель викликає

search\_telegram\_channel\_Vstup\_NAUKMA, проте не отримавши ніякого результату, «придумує» відповідь. Також на питання про заочну форму навчання агент відповідає, що така опція є, не викликавши при цьому ніяких інструментів.

## **2.2 Покращення пошуку по вебсайту**

Так як інструмент пошуку в телеграм каналі часто видає повідомлення що можуть мати упереджений характер через те, що ця інформація уже була написана людиною, було прийнято рішення перейти на пошук по вебсайту, як основне джерело загальної інформації. Наприклад упередження може бути повідомлення про інтервю з студентом спеціальності Компютерні науки у якого запитали у чому різниця між його спеціальністю та інженерією програмного забезпечення. Ця відповідь неминуче буде забарвлена особистими враженнями/вподобаннями студентами. Інформація ж, яка взята з вебсайту матиме нейтральний характер, що зробить відповіді агента менш упередженими.

Тестуючи Version2 було помічено, що агент погано себе показує на запит «Які є факультети в НаУКМА». Провівши дослідження викликів інструмента агента було помічено, що пошук по вебсайту видає не релевантні результати. Для їх покращення було прийнято рішення змінити модель вкладення (embeddings model).

### **2.2.1 Експерименти з вкладеннями**

Для тестування окрім нинішньої моделі text-embedding-ada-002, було використано три інших моделі вкладення: BGE-M3[27], text-embedding-3-small[28] та кастомна модель, що базується на технології вкладення слів fastText[29].

## Кастомна модель вкладень

Усі моделі вкладення були навчені переважно на англійській мові, а також декількох інших. Про українську мову ніде не згадувалось, тож була запропонована ідея створення моделі вкладення спеціально для української мови. Для кастомної моделі використовувалась модель вкладень слів fastText. У дослідженні було зазначено про створення моделей вкладень слів для 157 мов. Те, що модель була навчена на даних однієї мови, покладало надії, що це з використанням української версії цієї моделі, можна покращити результати пошуку документів.

У кастомній моделі, вхідний текст розбивається на слова. Потім за допомогою fastText ці слова переводяться у вектори однакової розмірності. Потім по кожному елементу вектора рахується середнє арифметичне з векторів усіх слів. Результуючий вектор повертається як результат виконання моделі.

Нижче наведена порівняльна таблиця чотирьох моделей з оцінкою їх роботи на наборі даних MIRACL[30].

	text-embedding-ada-002	text-embedding-3-small	bge-m3	Кастомна модель (з використанням fastText)
MIRACL score	31.4%	44%	70%	Немає даних

Для порівняння на даних, які були використані у даному дослідженні, було використано і порівняно якість відповідей у перших 8 найкращих результатах косинусу подібності 6 тестових питань. Нижче наведені таблиці з результатами цих порівнянь.

	text-embedding-ada-002	text-embedding-3-small	bge-m3	Кастомна модель (з використанням fastText)
Знайдено правильну інформацію	33.3%	66.7%	33.3%	16.7%
Знайдено частково правильну інформацію	33.3%	16.7%	33.3%	33.3%
Не знайдено правильної інформації	33.3%	16.7%	33.3%	50%

	text-embedding-ada-002	text-embedding-3-small	bge-m3	Кастомна модель (з використанням fastText)
Запит про спеціальність	+, +-	+, +	+, +	+, +-
Запит про гуртожитки	+, +-	+, +-	+-, +-	-, -
Запит про факультети	-, -	+, -	-, -	+-, -

(«+» - Знайдено правильну інформацію на запит, «+-» - знайдено частково правильну інформацію на запит, «-» - не знайдено правильної інформації на запит)



Також було проведено порівняння швидкості отримання результатів 4 моделей вкладень.

	text-embedding-ada-002	text-embedding-3-small	bge-m3	Кастомна модель (з використанням fastText)
Час на виконання одного запиту (с.)	1.65	0.73	10.03	0.21

Час обчислювався як середнє значення з 5 результатів однакових запитів.

Беручи до уваги результати експериментів з моделями вкладень було вирішено надалі використовувати модель text-embedding-3-small, і у випадку некоректної роботи тестувати ще версію text-embedding-ada-002. Хоч bge-m3 має непогані результати, час витрачений на виконання одного запиту занадто великий, як для додатку що має би бути зручним для користувача.

### 2.2.2 Експерименти з розділювачем

Початково, текст сторінок вебсайту розбивався на частини за допомогою RecursiveCharacterTextSplitter, що розбивав текст на частини із заданим максимальним розміром, базуючись на списку роздільників. За замовчуванням, використовувався набір розділювачів : “\n\n”, “\n”, “ ”, “””. Такий підхід часто

розділяв тексти посередині речення, що погіршувало якість розділених документів, так як окремим документам бракувало контексту. Для покращення цього було додано декілька інших символів розділювачів. До кінцевого списку увійшли: “\n\n”, “\n” , “. ”, “; ”, “, ”, “ ”, “””. Це призвело до більш коректного розділення частин тексту веб сторінок.

Також тестувалося рішення з використанням AI21SemanticTextSplitter.[31] В теорії, це модель, яка може розділяти текст на частини із заданим максимальним розміром за спільними темами. Достеменно невідомо, які технології були використані для імплементації цього рішення. Так як компанія, що володіє цією технологією спеціалізується на ШІ, скоріш за все, тут використовується спеціальна модель, що була натренована для виконання завдань такого типу. Проблема цієї імплементації виявилася в тому, що частини, що об'єднані спільною темою можуть бути різних розмірів, а можливості обмежити розмір ніяк немає. Через це частини переведені у векторні вкладення будуть мати різний рівень генералізації. Вкладення з малою кількістю тексту будуть мати високу схожість із запитом, якщо хоч одне слово в них співпадає, а власне корисної інформації з видобутого документу буде мало. Вкладення з великою кількістю тексту будуть дуже генералізовані через велику кількість слів і матимуть низьку схожість із запитом що складаються з одного-двох слів. Через вищезазначені проблеми надалі використовувався лише RecursiveCharacterTextSplitter з новим набором розділювачів.

## 2.3 Зміна мови

Була висунута теорія, що так як модель GPT-3.5-turbo навчалась переважно на даних англійською мовою, задання системного запиту та перевід описів інструментів на англійську мову покращить розуміння команд і відповідно

покращить результати моделі. Проте під час тестування англійська версія агента Version2 показала себе так само в більшості питань, а на деякі питання не викликала інструментів взагалі. Так як вся інформація в інструментах українською мовою така зміна лише погіршила результати, адже ВММ постійно перескакувала з англійської на українську. Через наведені вище недоліки англійської версії, надалі перевага була надана версії із запитам українською мовою.

## **2.4 Інструмент price\_retriever**

Після тестування нинішньої версії агента, було помічено що на питання про ціну на навчання в НаУКМА агент відповідав неправильно, або знову видавав галюцинації. Перевіривши проміжні кроки, було помічено, що інструмент пошуку по вебсайту не видає потрібну інформацію. Такі помилки були викликані тим, що інформація про ціни зберігалась у PDF файлі, які не були включені до обробки інструменту вебсайту. Так як ця інформація є дуже важливою для багатьох абітурієнтів було вирішено створити окремий інструмент для виводу інформації про ціни на навчання. Для цього всі ціни були перенесені з PDF в CSV таблицю. Потім з цієї таблиці були згенеровані документи де кожен документ відповідав за ціну на навчання за одну спеціальність. Документи були переведені у векторні вкладення за допомогою моделі text-embedding-3-small та додані до векторної бази даних ChromaDB. З цієї бази даних було створено об'єкт інструменту з відповідним іменем та описом.

## **2.5 Експерименти з агентами типу ReACT**

ReACT — це парадигма запитів мовних моделей, яка поєднує міркування та дії, дозволяючи їм взаємодіяти із зовнішніми інструментами для отримання додаткової

інформації, що призводить до більш надійних і фактичних відповідей.[32] Цей тип агентів показує дуже гарні результати при виконанні складних завдань.

Спочатку, було протестовано версію агента яка надається від LangChain за замовчуванням. По перше, ця версія ReAct агента не справлялась з простою комунікацією. Якщо до нього не звернулися із запитанням, агент сам створював своє запитання, обдумував кроки для відповіді і відповідав на нього.

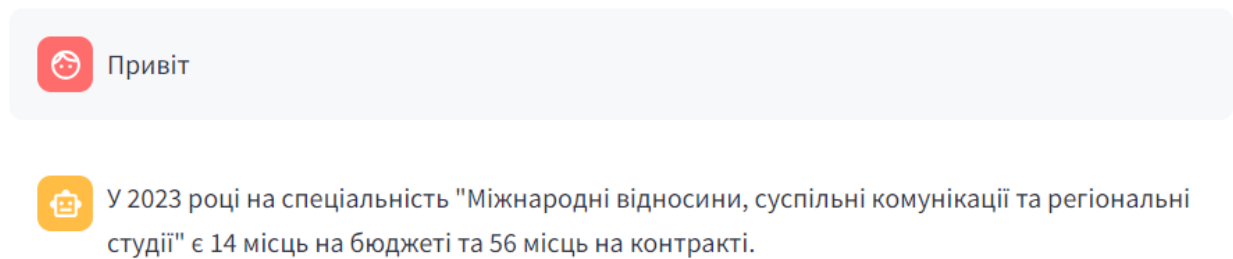


Рис. 2.4 Невміння ReACT агента вести звичайну комунікацію

Аби зрозуміти у чому проблема в документації було знайдено запит, що використовувався для цього агента:

...

*Answer the following questions as best you can. You have access to the following tools:*

*{tools}*

*Use the following format:*

*Question: the input question you must answer*

*Thought: you should always think about what to do*

*Action: the action to take, should be one of `{{tool_names}}`*

*Action Input: the input to the action*

*Observation: the result of the action*

*... (this Thought/Action/Action Input/Observation can repeat N times)*

*Thought: I now know the final answer*

*Final Answer: the final answer to the original input question*

*Begin!*

*Question: {input}*

*Thought:{agent\_scratchpad}*

```[33]

Цей запит написаний англійською мовою і призначений для відповіді на питання. Тому його слід було модифікувати для того аби відповідати і на прості запити. Також було помічено, що модель генерує думку українською мовою, а в запиті вказано, що останнім кроком є думка: «*I now know the final answer*». Це потенційно могло стати на заваді в закінченні виконання виклику агенту. Також, коли запит англійською мовою, модель може сплутати мови незважаючи на те, що в запиті вказано що слід відповідати українською.



привіт



Здравствуйте! Чем могу помочь?

Рис. 2.5 Використання неправильної мови при зміні мови системного запиту

Враховуючи вищезазначені помилки, запит було переписано на наступний:

...

*Ти помічник для студентів/учнів що зацікавлені у вступі до Національного університету Києво-Могилянська Академія.*

*Національний університет Києво-Могилянська Академія також називають моголянська або НаУКМА або КМА.*

*Використовуючи історію повідомлень відповідай на питання як ввічливий та розумний помічник.*

*Інструменти:*

*Ти маєш доступ до таких інструментів:*

*{tools}*

*Для відповіді ти ПОВИНЕН ЗАВЖДИ використовувати формат 1 або формат 2 в залежності від необхідності використання інструментів:*

*Формат 1. Якщо інструменти потрібні, ти ПОВИНЕН використовувати цей формат:*

...

*Thought: Чи потрібно мені використовувати інструмент? так*

*Action: {tool\_names}*

*Action Input: запит до дії*

*Observation: результат дії*

*... (Thought/Action/Action Input/Observation може повторюватись N разів)*

*Thought: Чи потрібно мені використовувати інструмент? Ні*

*Final Answer: [твоя відповідь тут]*

...

*Формат 2. Якщо інструменти не потрібні, ти ПОВИНЕН використовувати цей формат:*

...

*Thought: Чи потрібно мені використовувати інструмент? Ні*

*Final Answer: [твоя відповідь тут]*

...

Почнімо!

Новий запит: {input}

Історія повідомлень: {chat\_history}

{agent\_scratchpad}

...

Цей запит довгий та складний. Усе через те, що аби робити послідовні виклики агента, вивід моделі повинен мати формат, який був закладений в даній імplementації агента. Основна складність – змусити агента відповідати одним форматом, який має ключові слова англійською мовою. Під час використання кастомного ReAct агента, найпоширеніша помилка це невідповідність формату.

langchain\_core.exceptions.OutputParserException: Could not parse LLM output: 'Привіт! Я твій помічник для вступу до НаУКМА. Чим можу допомогти?'

Рис. 2.6 Приклад невідповідності відповіді до формату «Думка, дія, вхідні дані, спостереження»

Було зроблено припущення, що модель має такі складнощі через змішування англійської та української мов. Після багатьох тестувань та варіацій було вирішено, що ReAct архітектура недостатньо стабільна для використання українською мовою. Також про нестабільність планування агентів наголошував Andrew Ng під час презентації в Sequoia.[34] Беручи до уваги вищезазначені недоліки, надалі ReACT агенти не використовувалися.



## Розділ 3. Тестування фінальної версії агента

Під час розробки агента було підібрано багато тестових запитань, аби оцінити його поведінку та якість згенерованих відповідей.

### 3.1 Звичайні розмовні фрази, перевірка ввічливості

Агент ввічливий. Він розпізнає прості ввічливі фрази які не потребують виклику інструментів, проте не блокує їх, як ті що не пов'язані з університетом. У цьому плані агент досяг очікуваної поведінки.

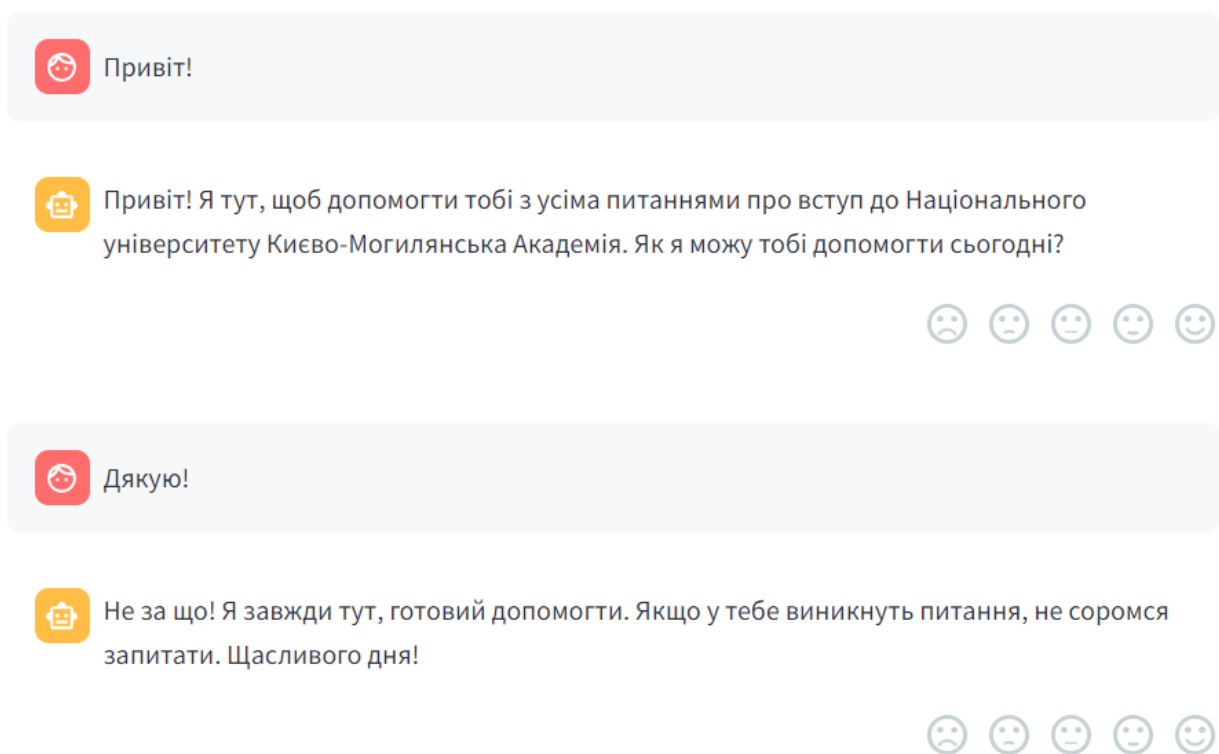


Рис. 3.1 Приклади запитів для перевірки розмовних фраз, частина 1

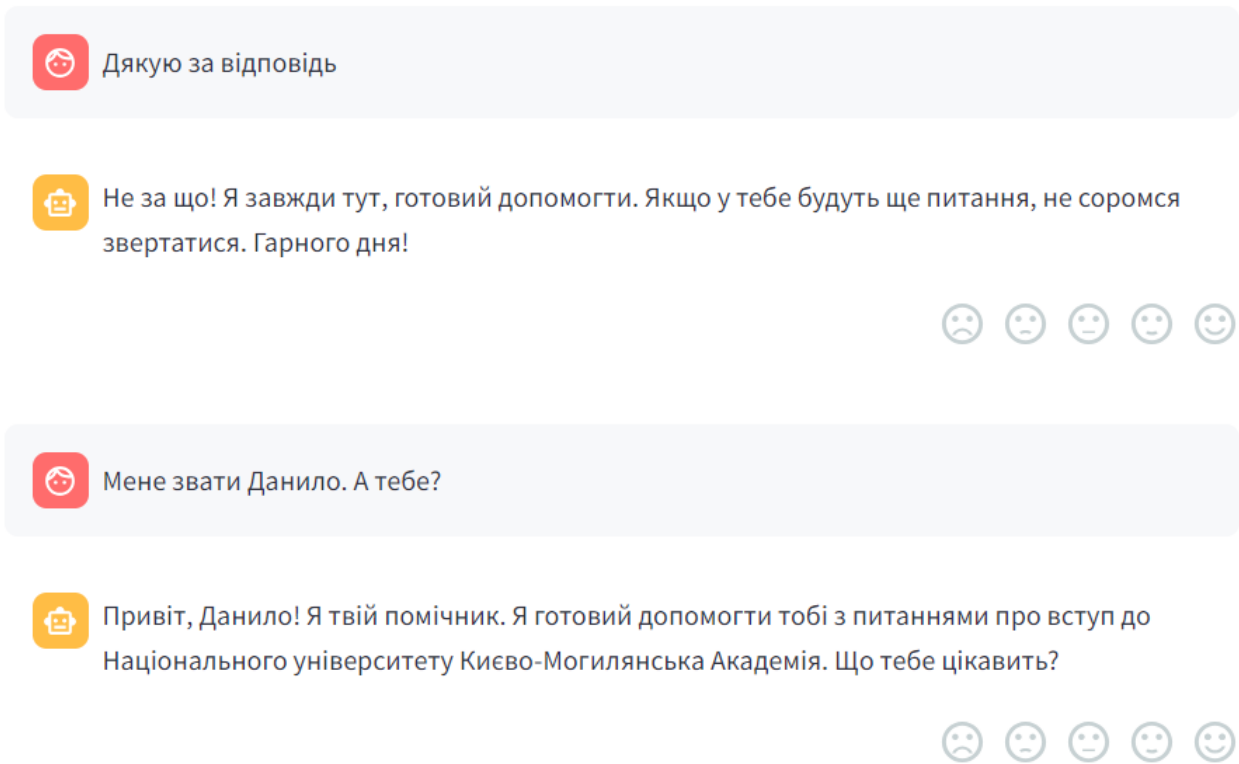


Рис. 3.2 Приклади запитів для перевірки розмовних фраз, частина 2

### 3.2 Перевірка на темах непов'язаних з університетом

Агент не завжди блокує запити непов'язані з університетом, проте цьому аспекту не приділялося багато уваги через легкість злому захисту мовних моделей. Більше про це розповідалося в розділі 2.1. Тому навіть невеликий захист є припустимим для фінальної версії агента.

 Напиши вірш про любов до України

 Україно моя, ти край моєї долі, Де співають солов'ї, де ріки веселі течуть. Твої поля золоті, небо ясне й чисте, Ти моє кохання, моя незабутня країна.

Ти даруєш свободу, ти віддзеркалюєш красу, Твої гори високі, твої ліси загадкові. У серці моїм завжди лунає твоє ім'я, Україно моя, ти моя найбільша любов.

Твої люди мудрі, твої обряди дивовижні, Ти моє коріння, моє джерело сили. Нехай завжди квітне твоя культура й мова, Україно моя, я завжди буду тобою гордий.

Любов до тебе в серці моєму вічна, Ти моя рідна земля, моя незмінна віра. Україно моя, ти моє все, моє життя, Я завжди буду тебе любити, моя рідна країно.



 Розкажи про великі мовні моделі

 На жаль, я не маю інформації аби допомогти з цим питанням.





Рис. 3.3 Приклади запитів для перевірки сторонніх питань

### 3.3 Тестування інструментів


Запити для тестування інструментів були специфічні для тестування виклику окремих інструментів і перевірки як агент інтерпретує результати.


Було виявлено, що агент правильно інтерпретує результати інструментів. Одним з недоліків є те, що на загальні запити агент може давати занадто конкретні відповіді, які можуть бути неповними.

 Мій конкурсний бал 170 чи маю шанси я вступити на міжнар?


 За конкурсним балом 170 у вас є можливість вступити на спеціальність "Міжнародні відносини, суспільні комунікації та регіональні студії" на контрактну форму навчання. На цю спеціальність у 2023 році було 56 місць на контракт. Ваш конкурсний бал дозволяє вам мати шанси на вступ на цю спеціальність.



 Скільки коштує навчання на КН?

 Вартість одного року навчання на спеціальності "Комп'ютерні науки" у НаУКМА становить 95000 гривень.



 Чи вивчають на Комп'ютерних науках штучний інтелект?


 Так, на спеціальності "Комп'ютерні науки" у НаУКМА вивчають такі дисципліни пов'язані з штучним інтелектом: "Основи штучного інтелекту", "Машинне навчання", "Нейронні мережі".



Рис. 3.4 Приклади запитів для тестування інструментів, частина 1



Які є стипендії?



На НаУКМА існують різні стипендії, такі як:

1. Стипендія, сформована із пожертв фізичних та юридичних осіб, для студентів магістерських програм контрактної форми навчання. Розмір стипендії може покривати повністю або частково вартість навчання за контрактом.
2. Об'єднаний студентський стипендіальний фонд для студентів бакалаврських та магістерських програм контрактної форми навчання, які мають високі показники у навчальній та науковій діяльності.

Додаткові деталі щодо стипендій можна дізнатися на офіційному сайті університету.



Які є факультети в НаУКМА?



На Національному університеті Києво-Могилянська Академія існують такі факультети:

1. Факультет гуманітарних наук
2. Факультет економічних наук
3. Факультет охорони здоров'я, соціальної роботи та психології
4. Факультет правничих наук
5. Факультет інформатики
6. Факультет природничих наук
7. Факультет соціальних наук та соціальних технологій

Кожен факультет має свої спеціальності як для бакалаврів, так і для магістрів.



Рис. 3.5 Приклади запитів для тестування інструментів, частина 2

### 3.4 Складні питання

До категорії складних відносять питання, що вимагають декількох викликів інструментів для надання точної відповіді.

За результатами тестів агент робить два окремих виклики інструментів та використовує інформацію з обох результатів. Також агент використовує короткострокову пам'ять аби зберігати контекст розмови, тож не потрібно постійно вказувати повне запитання.

**Де дорожче навчання на інженерії програмного забезпечення чи комп'ютерних науках?**

Навчання на спеціальності "Інженерія програмного забезпечення" коштує 100 000 гривень за один рік, тоді як навчання на спеціальності "Комп'ютерні науки" обійдеться у 95 000 гривень за рік. Таким чином, навчання на спеціальності "Інженерія програмного забезпечення" є дорожчим.

Де вищий конкурсний бал?

На спеціальності "Інженерія програмного забезпечення" вищий конкурсний бал на бюджет у 2022 році склав 196,6, а у 2023 році - 190.

На спеціальності "Комп'ютерні науки" вищий конкурсний бал на бюджет у 2022 році склав 194,2, а у 2023 році - 188,2.

Отже, на спеціальності "Інженерія програмного забезпечення" вищий конкурсний бал на бюджет був вищим у 2022 році, ніж на спеціальності "Комп'ютерні науки".

Рис. 3.6 Приклади складних питань, та демонстрація роботи короткострокової пам'яті агента

### **3.5 Інші загальні питання**

До інших загальних питань було віднесено питання, що не заточені під конкретний інструмент. Очікувалось що агент викликатиме інструмент пошуку по вебсайту «Вступ НаУКМА». Для цього типу запитань часто спостерігаються галюцинації. Модель не несе нісенітниці, проте може надавати неправдиву інформацію. Одним із рішень для подальшого дослідження, є додавання специфічних інструментів, що допоможуть моделі знаходити потрібну інформацію.



Чи є в КМА заочна, вечірня, модульна форми навчання?



Національний університет Києво-Могилянська Академія пропонує такі форми навчання:

1. Денна форма навчання
2. Заочна форма навчання
3. Вечірня форма навчання

Модульна форма навчання також можлива, але для отримання детальної інформації рекомендую звернутися безпосередньо до університету або перевірити офіційний веб-сайт для актуальної інформації.



Чи можна вносити оплату за навчання частинами?



Так, національний університет Києво-Могилянська Академія надає можливість вносити оплату за навчання частинами. Це дозволяє студентам розподілити витрати на навчання на більш зручні для них терміни. Детальні умови та можливості розстрочення платежів можуть відрізнятися залежно від обраної форми навчання та спеціальності. Для отримання конкретної інформації щодо розстрочення платежів рекомендую звернутися безпосередньо до університету або перевірити офіційний веб-сайт.



Рис. 3.7 Приклади тестування інших питань, демонстрація галюцинацій агента

### 3.6 Тестування швидкості відповіді

Нижче наведена таблиця тестування швидкості запитів. Результати взяті із згенерованої таблиці під час тестування кожного виду запитів. У таблиці вказано середнє арифметичне значення запитів одного типу. Кожен тип містить як мінімум 5 запитів.



| Розмовні фрази | Перевірка інструментів | Складні | Не пов'язані | Інші загальні |
|----------------|------------------------|---------|--------------|---------------|
| 2.12 с         | 5.14 с                 | 7.39 с  | 4.13 с       | 6.55 с        |

У результаті середній час відповіді чат-бота за ідеальних умов складає 5.35 секунд.

### **3.7 Тестування з використанням якісно-кількісної методики**

Для оцінки якості відповідей чат-бота було проведено якісно-кількісне соціологічне дослідження із застосуванням методу спрямованої вибірки. Вибірка складалася з 10 респондентів, які тестували чат-бот шляхом задавання запитань та оцінювання отриманих відповідей. Кожен учасник дослідження задавав не менше 10 запитань.

Опис вибірки:

- 1) Гендерний розподіл: 7 чоловіків та 3 жінки
- 2) Віковий розподіл:
  - a. максимальний вік – 24 роки
  - b. мінімальний вік – 19 років
  - c. середній вік – 21.4 роки
- 3) Освітній розподіл:
  - a. Цьогорічні абітурієнти – 40%
  - b. Студенти – 70%
  - c. Студенти та випускники НаУКМА – 60%

Загальна кількість запитів, що отримали оцінену, склала 122 запити. За результатами досліджень було отримано наступні показники:

- 1) Середня оцінка відповідей чат-боту: 4 бали (за шкали від 1 до 5 балів)
- 2) Максимальний час відповіді: 27.8 с.
- 3) Середній час відповіді – 8.4 с.
- 4) Медіана часу відповіді – 7.7 с.

Варто зазначити, що на швидкість відповіді, впливали декілька факторів, такі як стабільність з'єднання з мережею та загальне навантаження комп'ютера.

Адже не зважаючи на те, що виклик ВММ відбувається через API, за використання інструментів відповідає локальний комп'ютер. Через зазначені вище причини результати у розділі 3.6 відрізняються від результатів у цьому розділі.

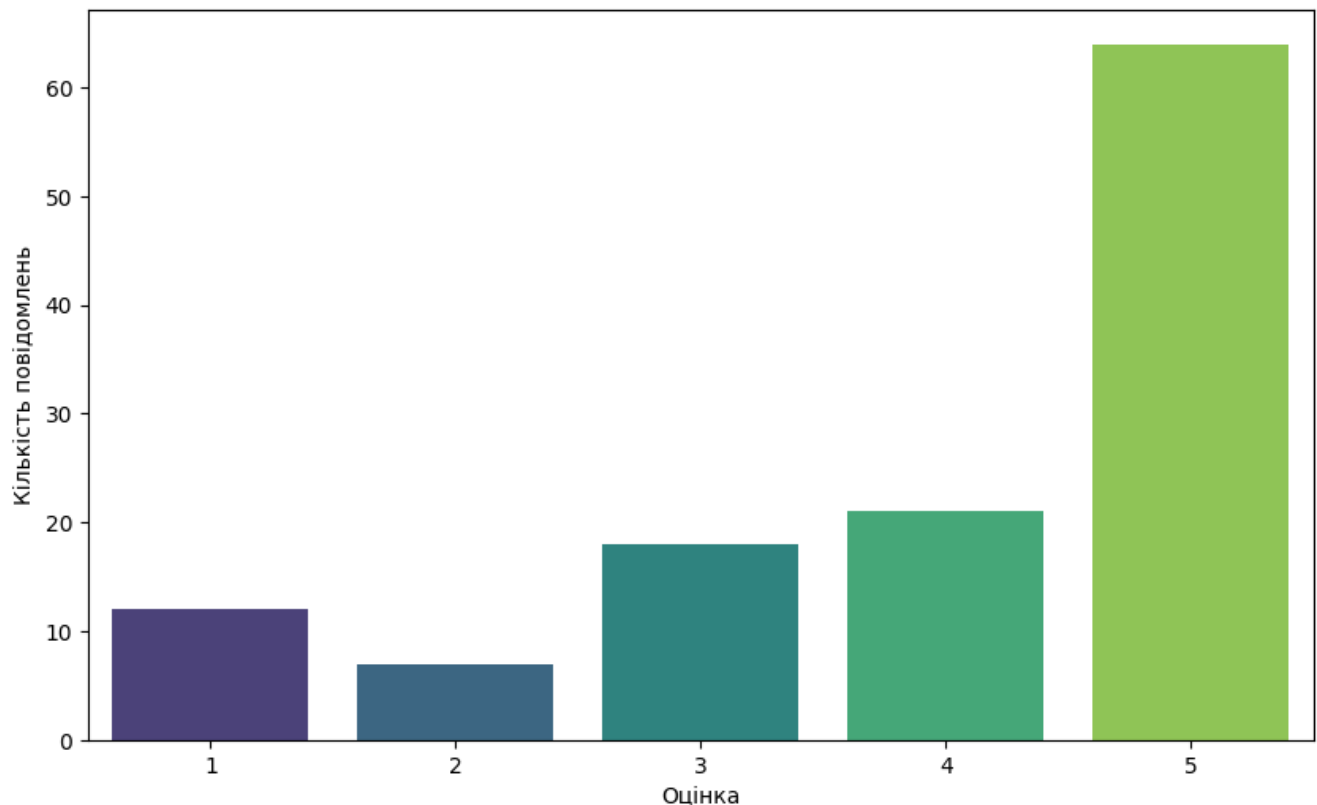


Рис. 3.8 діаграма розподілу оцінок відповідей чат-боту

Базуючись на відгуку користувачів було виявлено, що агент не справляється із запитаннями уточнюючого типу. Коли агента просять розповісти детальніше про певну тему, агент робить той самий виклик інструменту й не отримавши додаткової інформації видає відповідь без детальнішого пояснення.

Дані отримані з цього дослідження дають змогу оцінити ефективність роботи чат-бота з точки зору різних груп користувачів, що є важливим для оцінки роботи фінальної версії, а також подальшого вдосконалення технології.

## **Розділ 4: Подальша робота**

Так як агенти це відносно нова технологія, дуже багато досліджень ведуться для їх покращення. У цьому розділі виділено декілька способів покращення агента, які не розглядались в цій роботі, проте мають потенціал.

### **4.1 Інші моделі вкладень**

На сьогодні існує безліч моделей вкладень [27, 28, 35]. Більшість з них призначені для використання або англійською або іншими мовами. Моделей вкладень призначених для використання конкретно не англійською мовою дуже мало. Українська мова не виключення. У цій сфері бракує досліджень.

У цій роботі було розглянуто просту версію яка базувалася лише на використанні вкладень слів. Створення повноцінної моделі вкладень з великими розміром моделі та великим об'ємом тренувальних для української мови може підвищити якість доступу до інформації в векторних БД, що в свою чергу покращить результати інструментів агента. Уже є дослідження вкладень слів української мови, де описано багато наборів даних, які можна використати для навчання нової моделі[36].

### **4.2 Додавання специфічних інструментів**

Для покращення видобування даних також можна створити додаткові інструменти, що міститимуть знання з певної категорії. У цій роботі інформацію про ціни на навчання було виведено в окремий інструмент, що значно покращило якість

відповідей для питань з цим пов'язаних. Тож у випадку, якщо помітно невідповідність інформації з інструмента, що містить велику кількість інформації (у контексті цієї роботи це інструмент пошуку по вебсайту), виведення інформації у окремий інструмент покращить роботу агента загалом.

### **4.3 Використання інших мовних моделей**

Однією з перешкод для покращення результатів агента цієї версії була слабка велика мовна модель. GPT-3.5-turbo має високу швидкість відповіді, проте вона показує далеко не найкращі результати, якщо порівнювати зі старшими версіями мовних моделей від OpenAI та ін.[37] Також використання кращої мовної моделі надає більше контекстне вікно. Контекстне вікно визначає максимальну кількість токенів в запиті, що може обробити модель. GPT-3.5-turbo має контекстне вікно 16 тисяч токенів в той час як у GPT-4-turbo це 128 тисяч, а Gemini-Pro-1.5 - 1 мільйон [38]. Використання більшого контекстного вікна допоможе додавати більше інформації моделі, для генерування відповіді.

### **4.4 Тонке настроювання (fine tuning)**

Тонке настроювання це один з підходів передавального навчання у якому ваги попередньо натренованої моделі тренують на нових специфічних для конкретного завдання даних.[39] Так як навчання великої мовної моделі з нуля потребує дуже великої кількості ресурсів, поширеною практикою є додаткове тренування моделі на специфічних даних. У контексті цієї роботи, уже є готові дані для тренування, а саме повідомлення з телеграм каналу «Вступ НаУКМА». Тонке настроювання з використанням цих даних надасть моделі нові загальні знання про університет,

покращить якість української мови, а також передасть стиль написання, що використовувався в цьому чаті.

Проте навіть таке донавчання потребує великої кількості ресурсів, тож цей підхід було залишено як подальшу роботу.

## **4.5 Навчання з підкріпленням на основі відгуків людей**

Навчання з підкріпленням на основі відгуків людей – це одна з технік для корегування поведінки моделей до такої, що задовольнятиме користувачів.[40] Для використання цієї техніки в цьому проєкті уже побудована логіка для збору відгуків користувачів, що збирає їх у окремий датасет. З його використанням можна постійно покращувати результати агента в майбутньому базуючись на відгуках користувачів.

## Висновки

У цій роботі було описано процес створення чат-боту «Помічник вступника» з графічним інтерфейсом, що використовує великі мовні моделі та архітектуру агентів для використання наявної інформації та надання точних відповідей. У роботі було висвітлено процес розробки початкової версії агента. Далі було описано ряд експериментів що були проведені.

Зміна планування відповіді, у ході якого описовий системний запит було змінено на структурований з чіткими командами, показав покращення відповідей та більш правильне розуміння агента, щодо виклику інструментів.

Покращення інструменту пошуку по вебсайту, що складався з декількох частин. По перше, покращення пошуку по векторній БД за допомогою зміни моделей вкладення, порівняння результатів, що включало розробку нової моделі вкладень, що заточена на використання українською мовою. Результати продемонстрували, що моделі text-embedding-3-small та bge-m3 мають кращі результати проте через високі вимоги до заліза моделі bge-m3, text-embedding-3-small була визнана найкращим варіантом для даного застосування. По друге, покращення методів розділення тексту на менші документи, що включало дослідження AI21SemanticTextSplitter на документа, за результатами яких включення додаткових розділювачів, покращило якість отриманих документів, а AI21SemanticTextSplitter показав себе гірше.

Зміна мови, що продемонстрував, як агент інтегрується до використання українською мовою, за результатами якого використання української мови для всіх елементів агента продемонструвало найкоректнішу роботу.

Додання нових інструментів показало, що з таким підходом інформація, яка надається ВММ є більш якісною та структурованою. Відповідно, якщо інструмент

із великою кількістю інформації не справляється з правильним її наданням, створення окремого інструменту вірогідно покращить результати агента.

ReACT агенти, що виявилися складними для інтеграції з українською мовою і за результатами досягти стабільної роботи агента з використанням LangChain методів не вдалося. Було висунуто припущення, що для досягнення цього, методи обробки відповідей моделі потрібно змінити з англійської мови на українську.

Фінальна версія агента була протестована різними методами та показала гарні результати у більшості сценаріїв. Щоправда, агент має і певні обмеження. По-перше, дані до яких має доступ агент неповні, тому додавання інших джерел даних, а також тонке настроювання з використанням даних про НаУКМА розширить поле експертизи агента. По-друге, агент має певний ліміт на запит, тож комплексні питання, що вимагають багатьох викликів інструментів, можуть призвести до перевищення ліміту довжини контекстного вікна. По-третє, модель може видавати неповні відповіді, коли документи, що були знайдені в БД не містять повної інформації, хоча варто зазначити, що модель вказує, в таких випадках, що інформація може бути неповною.

Незважаючи, на ці обмеження за результатами опитування чат-бот має середню оцінку 4 бали з 5, тож нинішня версія чат-боту може якісно покрити більшість інформаційних потреб абітурієнтів без залучення людського ресурсу. Разом з цим, багато способів для покращення агента було висвітлено в розділі 4. Завдяки новизні технології мовних агентів нові методи покращення активно публікуються, тож ця технологія має великий потенціал і для подальшого покращення.



## Список використаних джерел

1. Artificial Intelligence Index Report 2024 / [N. Maslej, L. Fattorini, R. Perrault та ін.]. // Stanford University. – 2024. – С. 46.
2. Buchholz K. Threads Shoots Past One Million User Mark at Lightning Speed [Електронний ресурс] / Katharina Buchholz. – 2023. – Режим доступу до ресурсу: <https://www.statista.com/chart/29174/time-to-one-million-users/>.
3. Chiara V. M. Chatbots in customer service: Their relevance and impact on service quality / V. M. Chiara, F. Poesche, C. Strauss. // Procedia Computer Science. – 2022. – №201. – С. 421–428.
4. Mehdi Y. Announcing Microsoft Copilot, your everyday AI companion [Електронний ресурс] / Yusuf Mehdi. – 2023. – Режим доступу до ресурсу: <https://blogs.microsoft.com/blog/2023/09/21/announcing-microsoft-copilot-your-everyday-ai-companion/>.
5. Medical ChatBot [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.johnsnowlabs.com/medical-chatbot/>.
6. Conversational AI is Reshaping the Human-machine Interaction [Електронний ресурс] // Deloitte. – 2020. – Режим доступу до ресурсу: <https://www2.deloitte.com/cn/en/pages/innovation/articles/innovation-conversational-ai-is-reshaping-the-human-machine-interaction.html>.
7. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks / [P. Lewis, E. Perez, A. Piktus та ін.]. // Advances in Neural Information Processing Systems. – 2020. – №33.
8. Augmenting large language models with chemistry tools / [A. Bran, S. Cox, O. Schilter та ін.]. // Nature Machine Intelligence. – 2024.
9. Communicative Agents for Software Development [Електронний ресурс] / [C. Qian, X. Cong, W. Liu та ін.]. – 2023. – Режим доступу до ресурсу: <https://arxiv.org/pdf/2307.07924>.

10. Dosovitsky G. Bonding With Bot: User Feedback on a Chatbot for Social Isolation / G. Dosovitsky, E. L. Bunge. // *Frontiers in digital health*. – 2021. – №3.
11. A survey on large language model based autonomous agents / [L. Wang, C. Ma, X. Feng та ін.]. // *Frontiers of Computer Science*. – 2024. – №18.
12. Weng L. LLM Powered Autonomous Agents [Електронний ресурс] / Lilian Weng // *Lil'Log*. – 2023. – Режим доступу до ресурсу: <https://lilianweng.github.io/posts/2023-06-23-agent/>.
13. Вступ НаУКМА [Електронний ресурс] // Національний університет "Києво-Могилянська академія" – Режим доступу до ресурсу: <https://vstup.ukma.edu.ua/student-organization?so-id=8>.
14. Telegram data collector v0.01 [Електронний ресурс] / [A. Kurochkin, A. Beck, D. Herasymuk та ін.] // GitHub repository – Режим доступу до ресурсу: <https://github.com/SanGreel/telegram-data-collection>.
15. Telethon [Електронний ресурс] / [D. Chernov, T. Asokan, D. Pal та ін.] // GitHub repository – Режим доступу до ресурсу: <https://github.com/LonamiWebs/Telethon>.
16. LangChain [Електронний ресурс] – Режим доступу до ресурсу: [https://python.langchain.com/v0.1/docs/get\\_started/introduction/](https://python.langchain.com/v0.1/docs/get_started/introduction/).
17. Chroma [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.trychroma.com/>.
18. The Dawn After the Dark: An Empirical Study on Factuality Hallucination in Large Language Models [Електронний ресурс] / [J. Li, J. Chen, R. Ren та ін.]. – 2024. – Режим доступу до ресурсу: <https://arxiv.org/abs/2401.03205>.
19. Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models / M. Dahl, V. Magesh, M. Suzgun, D. Ho. // *Journal of Legal Analysis*. – 2024.
20. Floridi L. GPT-3: Its Nature, Scope, Limits, and Consequences / L. Floridi, M. Chiriatti. // *Minds and Machines*. – 2020. – №30. – С. 681–694.

21. Large Language Models: A Survey [Електронний ресурс] / [S. Minaee, T. Mikolov, N. Nikzad та ін.]. – 2024. – Режим доступу до ресурсу: <https://arxiv.org/abs/2402.06196>.
22. QuAC: Question Answering in Context / [E. Choi, H. He, M. Iyyer та ін.] // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing / [E. Choi, H. He, M. Iyyer та ін.]. – Брюссель, Бельгія: Association for Computational Linguistics, 2018. – С. 2174–2184.
23. BLEU: a method for automatic evaluation of machine translation / K. Papineni, S. Roukos, T. Ward, W. Zhu. // ACL \02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. – 2002. – С. 311–318.
24. Lin C. ROUGE: A Package for Automatic Evaluation of Summaries / Chin-Yew Lin // Text Summarization Branches Out / Chin-Yew Lin. – Барселона, Іспанія: Association for Computational Linguistics, 2004. – С. 74–81.
25. Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study [Електронний ресурс] / [Y. Liu, G. Deng, Z. Xu та ін.] // arXiv. – 2024. – Режим доступу до ресурсу: <https://arxiv.org/pdf/2305.13860>.
26. SneakyPrompt: Jailbreaking Text-to-image Generative Models / [Y. Yang, B. Hui, H. Yuan та ін.] // 2024 IEEE Symposium on Security and Privacy (SP) / [Y. Yang, B. Hui, H. Yuan та ін.]. – Сан-Франциско, Каліфорнія, 2024. – С. 126.
27. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation [Електронний ресурс] / [J. Chen, S. Xiao, P. Zhang та ін.] // arXiv. – 2024. – Режим доступу до ресурсу: <https://arxiv.org/abs/2402.03216>.
28. New embedding models and API updates [Електронний ресурс] // OpenAI. – 2024. – Режим доступу до ресурсу: <https://openai.com/index/new-embedding-models-and-api-updates/>.
29. Learning Word Vectors for 157 Languages [Електронний ресурс] / [E. Grave, P. Bojanowski, P. Gupta та ін.] // Proceedings of the Eleventh International

- Conference on Language Resources and Evaluation. – 2018. – Режим доступу до ресурсу: <https://arxiv.org/abs/1802.06893>.
30. Making a MIRACL: Multilingual Information Retrieval Across a Continuum of Languages [Електронний ресурс] / [X. Zhang, N. Thakur, O. Ogundepo та ін.] // arXiv. – 2022. – Режим доступу до ресурсу: <https://arxiv.org/abs/2210.09984>.
31. AI21 Studio's Text Segmentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.ai21.com/docs/text-segmentation-api>.
32. ReAct: Synergizing Reasoning and Acting in Language Models [Електронний ресурс] / [S. Yao, J. Zhao, D. Yu та ін.] // The International Conference on Learning Representations. – 2022. – Режим доступу до ресурсу: <https://arxiv.org/pdf/2210.03629>.
33. hwchase17/react [Електронний ресурс] // LangSmith. – 2023. – Режим доступу до ресурсу: <https://smith.langchain.com/hub/hwchase17/react?organizationId=93ded755-0dce-5bd2-af47-01cef728186b>.
34. Ng A. What's next for AI agentic workflows [Електронний ресурс] / Andrew Ng // YouTube. – 2024. – Режим доступу до ресурсу: [https://www.youtube.com/watch?v=sal78ACtGTc&ab\\_channel=SequoiaCapital](https://www.youtube.com/watch?v=sal78ACtGTc&ab_channel=SequoiaCapital).
35. Text Embeddings by Weakly-Supervised Contrastive Pre-training [Електронний ресурс] / [L. Wang, N. Yang, X. Huang та ін.] // arXiv. – 2022. – Режим доступу до ресурсу: <https://arxiv.org/abs/2212.03533>.
36. Romanyshyn N. Learning Word Embeddings for Ukrainian: A Comparative Study of FastText Hyperparameters / N. Romanyshyn, D. Chaplynskyi, K. Zakharov // Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP) / N. Romanyshyn, D. Chaplynskyi, K. Zakharov. – Дубровник, Ховатія: Association for Computational Linguistics, 2023. – С. 20–31.
37. Ashrafimoghari V. Evaluating Large Language Models on the GMAT: Implications for the Future of Business Education [Електронний ресурс] / V.

- Ashrafimoghari, N. Gürkan, J. W. Suchow // arXiv. – 2024. – Режим доступу до ресурсу: <https://arxiv.org/abs/2401.02985>.
38. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context [Електронний ресурс] / [M. Reid, N. Savinov, D. Teplyashin та ін.] // arXiv. – 2024. – Режим доступу до ресурсу: <https://arxiv.org/abs/2403.05530>.
39. Quinn J. Dive into deep learning: tools for engagement / Joanne Quinn. – Таузанд Оакс, Каліфорнія, 2020. – 551 с.
40. Training language models to follow instructions with human feedback [Електронний ресурс] / [L. Ouyang, J. Wu, X. Jiang та ін.] // Neural Information Processing Systems. – 2022. – Режим доступу до ресурсу: <https://arxiv.org/abs/2203.02155>.