Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

РОЗРОБКА ЧАТ-БОТУ В TELEGRAM З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки» 122

Керівник курсової роботи
с.в. Борозенний С.О.

_____

(підпис)

«__» _____ 2023 р.

Виконав студент КН-3
Білогрудов Д.В.
«__» _____ 2023 р.

Київ 2023

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ
Зав. кафедри мультимедійних систем
доцент, к.ф-м.н.
Жежерун О.П.
_____
(підпис)
«__» _____ 2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

Студента Білогрудова Даніїла Вячеславовича факультету інформатики 3 курсу

Тема: Розробка чат-боту в Telegram з використанням штучного інтелекту

Зміст ТЧ до курсової роботи:

Індивідуальне завдання
Календарний план виконання роботи
Вступ
1 Використання NLP у чат-ботах
2 Інструменти розробки
3 Опис реалізації чат-боту
Результати
Висновок
Список використаних джерел

Дата видачі «__» _____ 2023 р. Керівник Борозенний С.О.
_____
(підпис)
Завдання отримав _____
(підпис)

Тема: Розробка чат-боту в Telegram з використанням штучного інтелекту.

Календарний план виконання роботи:

№ п/п	Назва етапу	Термін виконання	Примітка
1.	Отримання завдання на курсову роботу	27.10.2022	
2.	Пошук літератури за темою роботи	10.01.2023	
3.	Ознайомлення з літературою	15.02.2023	
4.	Ознайомлення з фреймворком aiogram для розробки Telegram чат-боту	01.03.2023	
5.	Ознайомлення з бібліотеками NLTK, SpaCy і HuggingFace Transformers для обробки природного тексту	15.03.2023	
6.	Розробка чат-боту і інтеграція моделей NLP у чат-бот	01.04.2023	
7.	Написання текстової частини до курсової роботи	01.05.2023	
8.	Надання керівнику роботу для перевірки	08.05.2023	
9.	Коригування виконаної роботи	10.05.2023	
10.	Остаточне оформлення роботи і слайдів	12.05.2023	
11.	Подання роботи на кафедру для перевірки на плагіат	15.05.2023	
12.	Захист курсової роботи	22.05.2023	

Білогрудов Д.В. _____

Борозенний С.О. _____

«__» _____

ЗМІСТ

АНОТАЦІЯ

Ця робота присвячена розробці чат-боту для виконання запитів користувачів засобами штучного інтелекту, а саме обробки природної мови (NLP). У роботі розглядається такі методи обробки природної мови, як класифікація наміру (intent recognition), розпізнавання іменованих сутностей (NER), машинний переклад (machine translation) і генерація тексту (NLG), а також інструменти для використання цих методів. До того ж, розбираються відомі бібліотеки для розробки чат-ботів у месенджері Telegram і інтеграція методів NLP у чат-бот.

Ключові слова: чат-бот, Telegram, штучний інтелект, обробка природної мови.

ВСТУП

Актуальність теми та практичне значення.

Наразі чат-боти з можливостями штучного інтелекту переживають пік популярності. За даними новинної агенції Reuters, чат-бот ChatGPT, який використовує велику мовну модель GPT-4, розроблену компанією OpenAI, налічує 100 мільйонів користувачів щоденно всього за півроку після релізу, що робить ChatGPT найшвидше зростаючим додатком в історії [1].

Перевагами чат-ботів є те, що вони не вимагають ніяких додаткових дій від користувача для отримання інформації. Користування чат-ботом нагадує переписку з реальною людиною, а уся важлива інформація для формування відповіді автоматично витягується із запиту. Тим більше, для того, щоб почати користуватись чат-ботом, зазвичай, не треба встановлювати ніяке додаткове ПЗ, а достатньо відкрити месенджер або посилання на веб-сторінку.

Ці переваги роблять чат-боти привабливими також і для бізнесу. За даними видання Forbes, використання компанією чат-ботів може призвести до покращення клієнтського досвіду, скорочення витрат і підвищення ефективності процесів [2]. На відміну від традиційних методів технічної підтримки користувачів, де усі запити обробляються вручну співробітниками, тепер більшість запитів можуть бути оброблені автоматично чат-ботом, без втручання людини.

Також варто звернути увагу на обрану платформу для розробки чат-боту – Telegram. Станом на 2023 рік, Telegram знаходиться на 4 місці серед найуживаніших месенджерів у світі, налічуючи 700 мільйонів користувачів щомісячно [3]. Популярність платформи сприятиме збільшенню цільової аудиторії та кращій інтеграції чат-бота у робочий процес потенційного користувача.

Мета і завдання роботи.

Мета – дослідження різних методів обробки природної мови та їхнє використання у створенні чат-боту. Для досягнення мети були поставлені наступні завдання:

1. Розглянути основні методи обробки природної мови та їхню роль у обробці запитів користувачів.
2. Дослідити основні інструменти розробки чат-ботів для месенджеру Telegram та розробити чат-бот, здатний приймати вхідні повідомлення користувачів.
3. Розробити моделі для NLP або використати готові рішення для певної групи задач з подальшою інтеграцією їх у чат-бот.

Об’єкт і предмет дослідження.

Об’єкт дослідження – методи обробки природної мови, класифікація наміру, розпізнавання іменованих сутностей, машинний переклад та генерація тексту, які використовуються в розробленому Telegram чат-боті.

Предмет дослідження – розробка Telegram чат-боту з використанням штучного інтелекту та методів обробки природної мови для виконання запитів користувачів.

Структура роботи.

Курсова робота складається з анотації, вступу, трьох розділів, висновку, списку прийнятих скорочень і списку використаної літератури.

Перший розділ курсової містить загальні відомості про чат-боти, історію їхнього використання, а також відомості про обробку природної мови і її методи

– класифікація наміру, машинний переклад, розпізнавання іменних сутностей і генерація природного тексту.

Другий розділ присвячений аналізу інструментів для розробки NLP моделей і чат-ботів, та теоретичні відомості, які лягли в основу реалізації.

Третій розділ зосереджений на обґрунтуванні вибору тієї чи іншої технології для розробки чат-боту, опис деталей реалізації й аналіз отриманих результатів.

РОЗДІЛ 1. ВИКОРИСТАННЯ NLP У ЧАТ-БОТАХ

1.1 Чат-боти

Чат-бот – це комп'ютер, програма, алгоритм або штучний інтелект, який взаємодіє з людиною [4]. Чат-боти можуть бути створені для різних цілей, таких як обслуговування клієнтів, пошук інформації, персональна допомога та для розважальних цілей. Вони можуть бути інтегровані в месенджери, веб-сайти або мобільні додатки, дозволяючи користувачам взаємодіяти з ними за допомогою текстових повідомлень або голосових команд.

Однією з ключових переваг чат-ботів є їхня здатність надавати миттєві та персоналізовані відповіді на запити користувачів цілодобово, без необхідності втручання людини. Це, наприклад, дозволяє компаніям автоматизувати процеси технічної підтримки користувачів і тим самим заощадити кошти на додатковий персонал і підвищити рівень задоволеності клієнтів.

1.1.1 Принцип роботи чат-ботів

Чат-боти зазвичай створюються в такий спосіб, щоб зробити у користувача ілюзію спілкування з реальною людиною. Це досягається надаванням відповідей у розмовній манері. Принцип роботи чат-ботів можна поділити на дві головні дії:

1. Обробка запиту користувача для правильного розуміння контексту розмови і предмету запиту. Однією з найголовніших речей на цьому етапі вважається розпізнання наміру (intent recognition) [5].
2. Надання коректної відповіді, яка має відповідати запиту. Зазвичай, відповідь формується за допомогою алгоритмів генерування природної мови.

У залежності від принципу реалізації та типу введення розмови, чат-боти поділяються на два види [2]:

– Декларативні чат-боти – їх зазвичай називають «інтерактивним довідником». На основі заготовлених відповідей такі чат-боти можуть вести структуровані розмови та видавати користувачу певну інформацію про ресурс чи продукт.
– Предикативні (прогнозуючі) чат-боти – їх також називають «віртуальними асистентами». За допомогою алгоритмів штучного інтелекту, вони можуть надавати персоналізовані відповіді, базуючись на попередніх запитах.

1.1.2 Історія чат-ботів

Історію чат-ботів заведено відстежувати від питання «Чи можуть машини думати?» у статті «Обчислювальна техніка і інтелект» фундатора сучасної інформатики Алана Тюрінга [7]. У цій статті автор запропонував тест, пізніше названий на його честь, який полягає у відповіді на питання «Чи може машина імітувати людський інтелект настільки добре, щоб обдурити людину, змусивши її повірити, що вона спілкується з іншою людиною?». Саме тест Тюрінга популяризував ідею чат-ботів.

Однією з найстаріших програм, яка по праву може вважатися чат-ботом, це ELIZA, розроблена Лабораторією штучного інтелекту у Массачусетському технологічному інституті [4]. Принцип роботи чат-боту ELIZA полягає у перефразуванні запитів користувача («Я почуваюся сумно.» – «Чому ви почуваєтесь сумно?»). ELIZA стала однією з найвпливовіших програм у розвитку технології чат-ботів, спричинивши поштовх до бурхливого розвинення цієї галузі, а також до написання багатьох наукових статей про неї і впливу на витвори масової культури.

З роками розвиток технологій штучного інтелекту (далі – AI) дозволив створення більш розвинених чат-ботів, особливо на початку 2000-х років. Прогрес у сфері AI також перетнувся зі зростанням кількості месенджерів і соціальних мереж, що створило ідеальне підґрунтя для створення чат-ботів для масового використання. Як приклад чат-бота з тієї епохи можна назвати SmarterChild, який був інтегрований у AOL Instant Messenger [6]. SmarterChild був здатний розуміти команди природною мовою і міг виконувати широкий спектр завдань, таких як надання новин, прогнозу погоди чи надання інформації про фондовий ринок.

1.1.3 Сучасний стан чат-ботів

Сьогодні чат-боти здатні виконувати широкий спектр завдань – від простих, як-от прогноз погоди чи планування зустрічей, до більш складних, як діагностика захворювань чи надання фінансових порад. Як доказ їхньої поширеності, можна навести приклад чат-ботів Google Assistant і Apple Siri, які інтегровані за замовчуванням у дві найрозповсюдженіші мобільні операційні системи у світі – Android і iOS.

Справжній фурор здійснив випущений у 2022 році чат-бот ChatGPT компанії OpenAI [1]. Цей чат-бот використовує велику мовну модель GPT (generative pre-trained transformer – генеративний попередньо-навчений трансформер), навчену на величезних обсягах тексту різними мовами і здатну генерувати продовження тексту, наданого у запиті. ChatGPT привернув до себе увагу головним чином завдяки своїм детальним і чітким відповідям у багатьох галузях знань, включаючи генерацію коду [8].

1.2 Обробка природної мови

Обробка природної мови (NLP) – це галузь комп'ютерних наук, яка займається розумінням та генерацією людської мови комп'ютерами. Її головна

мета – забезпечити здатність комп'ютерів аналізувати, розуміти та генерувати мову з такою ж виразністю та ефективністю, як і людина. NLP є ключовим елементом у розробці чат-ботів, адже саме від коректності алгоритмів обробки мови залежить здатність чат-бота розуміти запити користувачів та надавати на них відповіді в зрозумілій формі.

1.2.1 Методи NLP

Задачі NLP зазвичай поділяються на дві великі підкатегорії. Першою з них є розуміння природної мови (NLU). NLU спрямоване на обробку не тільки синтаксичної структури мови, але й на вилучення з неї семантичного змісту. У свою чергу, під NLU підпадають наступні методи [9]:

- Розпізнавання іменованих сутностей (NER) – це метод, який використовується для вилучення та класифікації іменованих об'єктів з текстових даних за заздалегідь визначеними категоріями, такі як люди, організації, локації або дати.
- Класифікація тексту – цей метод передбачає розподіл тексту на різні класи або категорії, наприклад, спам чи не спам, позитивні чи негативні настрої або теми, такі як спорт чи розваги.
- Вилучення ключових слів – цей метод полягає у визначенні та вилученні найбільш релевантних слів або фраз з тексту.

Другою підкатегорію NLP є генерування природної мови (NLG). Головним завданням NLG є створення тексту, який б здавався написаним людиною і який б був легким для розуміння тією ж самою людиною. Серед методів NLU можна виділити наступні [9]:

- Підсумовування тексту – цей метод передбачає створення коротшої версії тексту зі збереженням найважливішої інформації.

– Генерування діалогу – цей метод передбачає створення розмови між комп'ютером і людиною, яка виглядала б природною і захоплюючою.

Також слід зазначити, що перед використанням будь-якого тексту у алгоритмах NLP, він має пройти через попередню обробку. Для цього найчастіше використовують наступні методи:

– Нормалізація тексту – цей метод передбачає перетворення тексту в стандартний формат, використовуючи такі прийоми, як токенізація, видалення стоп-слів, лематизація тощо.
– Маркування частин мови (part-of-speech tagging) – це процес віднесення кожного слова в тексті до певної частини мови (як-от іменник, займенник, дієслово тощо).

1.2.2 Основні проблеми NLP

Незважаючи на те, що NLP зазнала значного прогресу за останні роки, все одно залишається багато проблем і задач, з якими поточні проблеми мають складнощі. До таких відносяться [10]:

– Двозначність – наприклад, «терен» як рослина і «терен» як територія. У NER ця проблема постає при стиканні з топонімами з однаковою назвою – Лондон, Великобританія і Лондон, Канада.
– Розпізнавання іронії і сарказму – фраза «це було дуже цікаво» може вжитися у саркастичному тоні і набути протилежного сенсу.
– Орфографічні помилки – текст з помилками може вплинути на точність мовної моделі у навчанні чи розпізнаванні.

Щоб подолати ці виклики, розробники досліджують нові методи, такі як глибинне навчання, трансферне навчання та доповнення даних, а також

включають в моделі NLP більше контекстних і специфічних знань для відповідного домену.

1.2.3 Використання NLP на практиці

NLP має численні практичні застосування в багатьох галузях. Як вже було зазначено вище, однією з найпоширеніших галузей застосування NLP є чат-боти і віртуальні асистенти. Також можна відмітити застосування NLP у наступних сферах: машинний переклад (у поєднанні з нейронними мережами), аналіз настроїв користувачів у соціальних мережах (використовується компаніями для оцінки реакції на свій продукт, або ж політичними силами), інформаційний пошук (отримання потрібної інформації з великого корпусу тексту за ключовими словами), підсумовування тексту тощо.

Загалом, ці практичні застосування демонструють потенціал NLP для автоматизації та оптимізації різних завдань і процесів, що робить її важливою технологією для цифрової епохи.

РОЗДІЛ 2. ІНСТРУМЕНТИ РОЗРОБКИ

2.1 Платформи для підтримки Telegram Bot API

Для розробки будь-якого застосунку (боту) для месенджеру Telegram застосовується прикладний програмний інтерфейс (API) для взаємодії з системою повідомлень – Telegram Bot API, наданий розробниками месенджера [11]. Але взаємодіяти з API напряму неоптимально, бо це вимагає обробку низькорівневих HTTP-запитів і ручну реалізацію різних функцій, тому, як альтернативу, краще застосовувати інструменти, які надають розробникам спрощені методи роботи з повідомленнями, командами та іншим функціоналом ботів.

2.1.1 Бібліотеки для Python

Один із способів надсилання запитів до Telegram Bot API – це використання високорівневих інтерфейсів, які надають зручні абстракції та інструменти для взаємодії з API. Надалі будуть розглянуті такі інтерфейси саме мовою Python (аргументація чому саме Python найбільше пасує для цілей розробки чат-боту буде надана пізніше у роботі).

Серед найпоширеніших бібліотек мовою Python для Telegram Bot API можна виділити pyTelegramBotAPI [12] і фреймворк aiogram [13] (який, до речі, був створений українським розробником). Обидві платформи надають схожі базові можливості, такі як обробка вхідних повідомлень за типом повідомлення (текст, зображення, команда тощо), надсилання повідомлень у чат і підтримка інтерактивних повідомлень. Цього функціоналу має бути достатньо для головної мети чат-боту – отримування вхідних повідомлень, обробка їх засобами NLP і надання користувачу кінцевої відповіді.

Головна відмінність і, напевно, перевага фреймворку aiogram це повна підтримка асинхронності. Це означає паралельне виконання декількох завдань, що робить бібліотеку більш ефективною для обробки декількох вхідних запитів і виконання операцій вводу/виводу, та асинхронну взаємодію з зовнішніми API.

2.1.2 Хмарні платформи

Альтернативою до написання коду чат-бота власноруч можуть бути такі сервіси, як Google Dialogflow, Amazon Lex і IBM Watson Assistant – хмарні платформи, які пропонують можливості NLU, що дозволяють розробникам створювати та розгортати чат-боти та віртуальних асистентів на основі AI. Ці сервіси надають інструменти та інфраструктуру для обробки мови, розпізнавання намірів, управління діалогами та інтеграції з різними сервісами (такими, як Telegram), що полегшує розробку складних розмовних агентів без необхідності створювати все з нуля. Вони також пропонують такі функції, як вбудовані моделі машинного навчання, розпізнавання голосу, аналіз настроїв і багатомовну підтримку, що дозволяє розробникам створювати інтелектуальні та контекстно-залежні програми чат-ботів.

2.2 Підготовка тексту для подальшої обробки

Використання необробленого тексту безпосередньо при вирішенні завдань NLP може обмежувати ефективність аналізу. Сирий текст зачасту містить надлишкову інформацію, наприклад, розділові знаки, спеціальні символи, стоп-слова тощо. Попередня обробка допомагає видалити або обробити ці елементи, дозволяючи зосередитися на змістовному контенті.

2.2.1 Нормалізація

Нормалізація тексту – це процес перетворення текстових даних у стандартизований формат. Він передбачає застосування набору правил або методів для очищення, стандартизації та підготовки текстових даних для подальшого аналізу або задач NLP.

Для нормалізації тексту засобами мови Python найчастіше використовуються бібліотеки NLTK і SpaCy. Нижче наведений приклад коду, який приймає вхідний текст і нормалізує його:

```
nlp = spacy.load('en_core_web_sm')
doc = nlp(text.lower())

tokens = [token.text for token in doc if token.is_alpha]
normalized_tokens = [nltk.stem.lemmatizer.lemmatize(token) for token
    in tokens if token not in stop_words]
```

Нормалізація тексту відбувається у два етапи – токенізація і лемматизація. Для токенізації створюється пайплайн на основі моделі “en_core_web_sm” з бібліотеки SpaCy [14], який обробляє вхідний текст для подальшого витягування з нього токенів. Далі за допомогою лемматизатора з бібліотеки NLTK [15] кожен токен скорочується до базової словникової форми (“running” – “run”). Таким чином, текст нормалізований і готовий до подальшого використання.

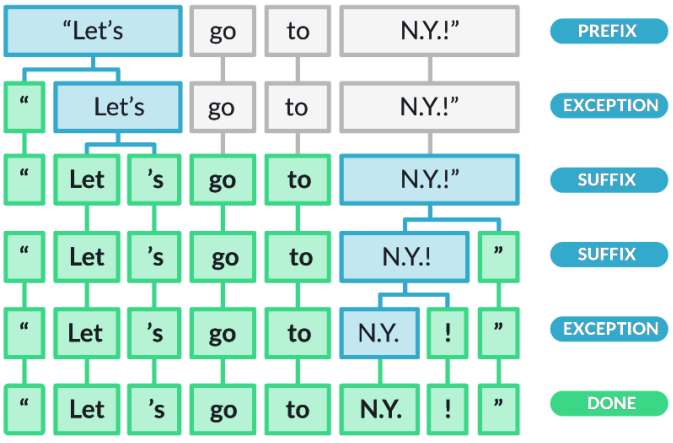

Рисунок 1. Процес токенізації засобами бібліотеки SpaCy [14]

2.2.2 Векторизація

Векторизація тексту – це процес перетворення текстових даних у числове представлення, яке зрозуміле алгоритмами машинного навчання і може бути оброблене ними. Він передбачає перетворення тексту на числові вектори або матриці, де кожен вимір представляє певну особливість або характеристику тексту. Існує багато способів векторизації тексту, найпоширеніші з яких – Bag-Of-Words, TF-IDF і Word Embeddings.

Бібліотека для машинного навчання scikit-learn [16] надає вбудований клас TfidfVectorizer() для подальшого використання векторизованого тексту у моделях машинного навчання.

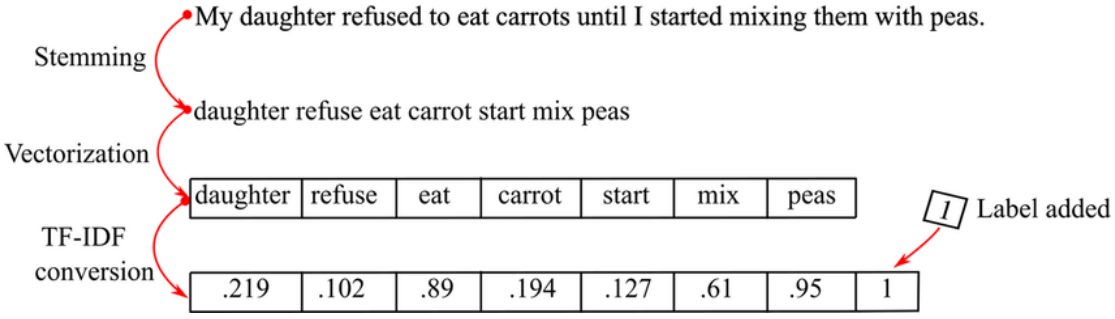

Рисунок 2. Принцип роботи векторизації тексту [17]

2.3 Моделі для класифікації

Класифікація тексту – це одне з фундаментальних завдань NLP, яке передбачає автоматичне присвоєння попередньо визначених категорій або міток текстовим документам на основі їхнього змісту. Метою класифікації текстів є розробка моделей, які можуть точно і автоматично класифікувати текстові документи за наперед визначеними категоріями або класами.

Для цілей даної роботи, буде розглянуто два типи класифікаторів – класифікатор наміру і класифікатор мови.

2.3.1 Класифікація наміру

Мета класифікації наміру – визначення наміру або задуму, що стоїть за певним текстовим фрагментом. Він спрямований на розуміння основної мети або наміру, який виражений користувачем у письмовій формі.

У контексті розробки чат-боту, було обрано три наміри для розпізнавання – «прогноз погоди», «пошук зображення» і «скорочення тексту». Для навчання був використаний набір з 1000 повідомлень, пов’язаних з пошуком інформації і які підпадають під один з трьох намірів. Для навчання був реалізований пайплайн для машинного навчання, який складається з двох класів бібліотеки scikit-learn: вищеописаний TfidfVectorizer() і одна з моделей класифікації. Нижче наведені результати тестування моделі класифікації наміру з різними класифікаторами.

Перевага використання пайплайну полягає в тому, що він дозволяє безперешкодно інтегрувати кілька етапів обробки та алгоритмів машинного навчання. Він спрощує робочий процес, інкапсулюючи етапи векторизації і класифікації в єдиний об'єкт, що полегшує навчання і розгортання моделі класифікації тексту.

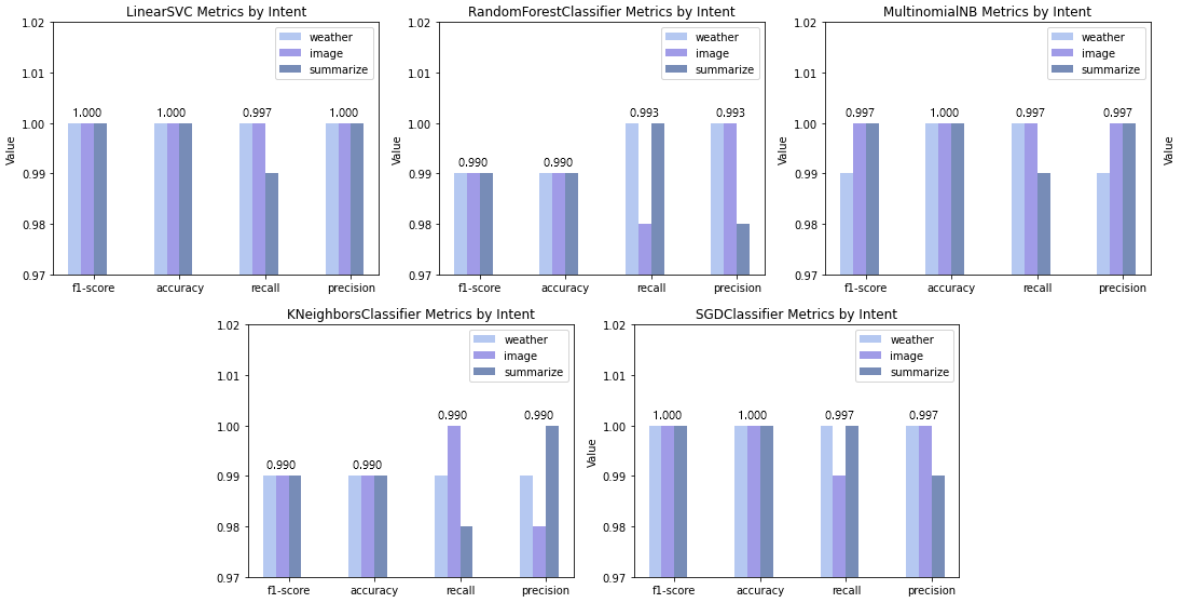

Рисунок 3. Метрики різних типів класифікаторів для розпізнавання наміру

2.3.2 Класифікація мови

Класифікація мови – це завдання автоматичного визначення мови, якою написано текст або документ. Як одна з основних проблем в NLP, вона відіграє вирішальну роль у різних застосунках, які передбачають багатомовну обробку даних.

Принцип роботи класифікатору мови ідентичний до класифікатора наміру. Відмінність тільки полягає у виборі тренувальних даних – був використаний датасет з 3 мільйонами повідомлень різними мовами. Для цілей розробки даного чат-бота, було обрано шість мов: англійська, українська, польська, іспанська, німецька і французька.

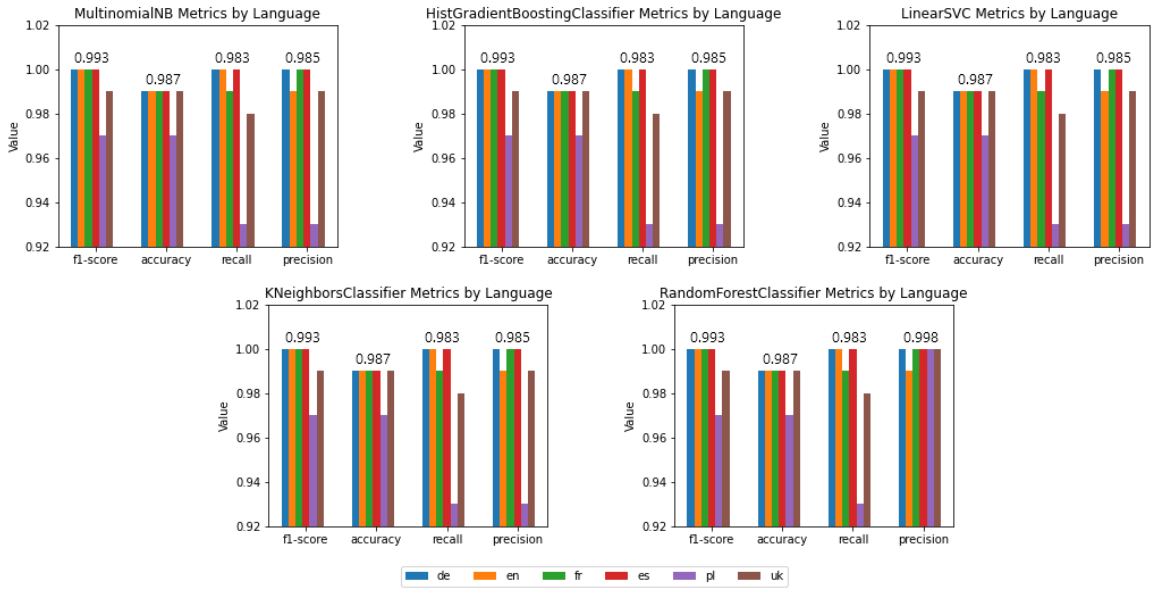

Рисунок 4. Метрики різних типів класифікаторів для розпізнавання мови

2.4 Методи машинного перекладу

Машинний переклад – це автоматичний переклад тексту з однієї мови на іншу за допомогою комп'ютерних алгоритмів. Для розробки чат-боту машинний переклад може бути альтернативою для тренування моделей NLP для кожної підтримуваної мови окремо – замість цього пропонується перекладати вхідне

повідомлення для подальшої обробки однією визначеною мовою (наприклад, англійською), і видавати перекладачу результат перекладений на початкову мову запиту.

Для реалізації машинного перекладу у застосунку можна використовувати сторонні API – Google Translation API і DeepL API. Що DeepL, що Google Translate пропонують різні функції та можливості персоналізації, такі як визначення мови, переклад тексту та документів, підтримка різних моделей перекладу тощо.

2.5 Методи NLU

Розуміння природної мови (NLU) займається машинним розумінням вхідного тексту. Методи NLU зможуть дозволити чат-боту розуміти та інтерпретувати вхідні дані користувача таким чином, щоб полегшити змістовну взаємодію, і спрямовані на вилучення ключової інформації з виразів природною мовою. До методів NLU можна віднести вищеописане розпізнавання наміру, але для вдосконалення можливостей боту слід запровадити додаткові методи розуміння мови.

2.5.1 Розпізнавання іменованих сутностей

Розпізнавання іменованих сутностей (NER) фокусується на ідентифікації та класифікації іменованих об'єктів у тексті, таких як імена людей, організацій, місцезнаходження, дати, числові значення тощо. Точно розпізнаючи та виділяючи ці сутності, чат-бот може глибше розуміти запити користувачів і надавати більш контекстуально релевантні відповіді.

Наприклад, користувач надсилає чат-боту запит «Яка буде погода в Києві через три дні?». Враховуючи, що намір запиту вже розпізнаний («прогноз погоди), далі алгоритм NER має витягти наступну інформацію: місцеположення

(«Київ») і дату, за яку надати прогноз «через три дні». Для цього знадобляться вже згадана бібліотека SpaCy, бібліотека для вилучення назв міст і країн flashgeotext [18] і бібліотека для парсингу дат dateparser [19].

Власне процес вилучення іменованих сутностей розбивається на два етапи – вилучення локації і вилучення дати. Для вилучення локації був використаний власний пул даних для пошуку на основі класу LookupData() з бібліотеки flashgeotext на основі датасету з назвами і синонімами до назв усіх населених пунктів з населенням більше 10 тисяч. Для вилучення дати спочатку з запиту дістаються усі токени, пов'язані з часом і датами, потім за допомогою функції parse бібліотеки dateparser надана дата перетворюється на її представлення у форматі POSIX.

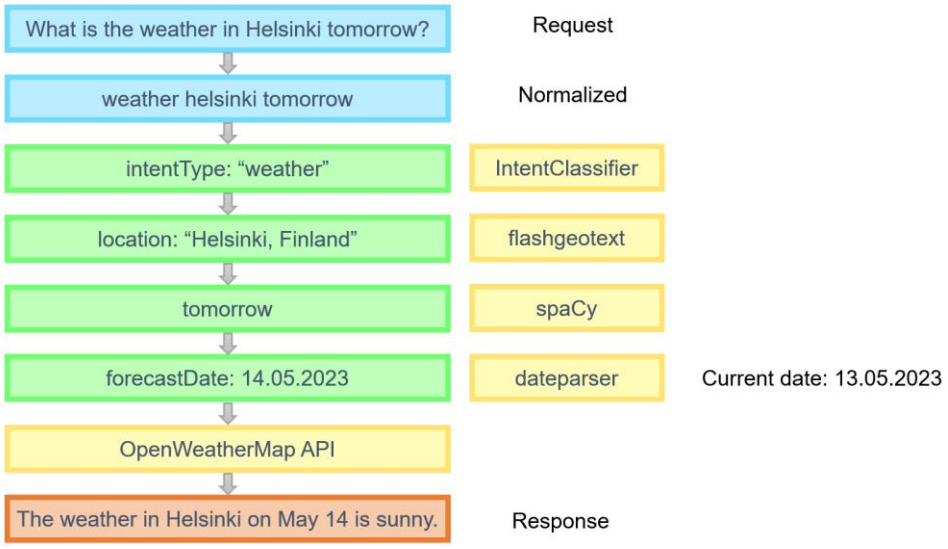

Рисунок 5. Вилучення іменованих сутностей з запиту

2.5.2 Вилучення ключових слів

Вилучення ключових слів – це метод, який використовується для визначення та вилучення важливих або релевантних слів чи фраз із заданого тексту. Він спрямований на визначення найбільш значущих ключових слів, які представляють основну тему тексту або речення.

У контексті чат-боту вилучення ключових слів важливе, наприклад, у випадку, коли користувач надає запит, пов’язаний з пошуком даних (загальної інформації, зображень, тощо) про певний об’єкт чи тематику. Для реалізації цього функціоналу можна використати маркування частин мови (part-of-speech tagging), яке за замовчуванням наявне у бібліотеці SpaCy. Робиться припущення, що запит розподіляється на дві групи слів – тип запиту і тема запиту («Знайди зображення червоної автівки» – «знайди зображення» / «червоної автівки»). Запит токенізується, потім з нього витягуються іменникові фрагменти зі значущими частинами мови (іменники, дієслова, прикметники), і наприкінці повертається іменниковий фрагмент, пов’язаний саме з темою запиту (зазвичай, останній у реченні).

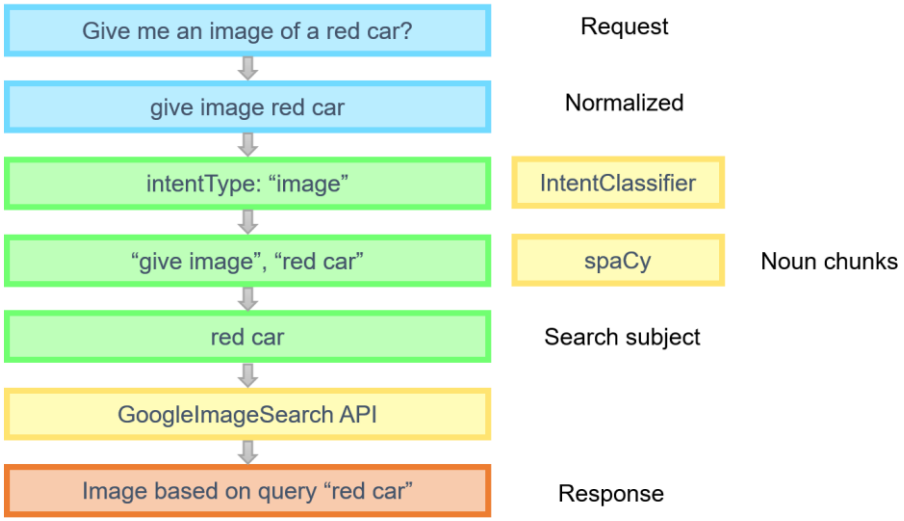

Рисунок 6. Вилучення теми пошуку з запиту

2.6 Методи NLG

Генерація природної мови (NLG) фокусується на створенні тексту, подібного на текст природою мовою на основі попередньо визначених правил, шаблонів або алгоритмів машинного навчання. Методи NLG є важливими для розробки діалогових агентів, чат-ботів та інших систем штучного інтелекту, які можуть спілкуватися з користувачами у більш природній та людській манері.

Як один з методів NLG, можна взяти приклад генерації стислого переказу тексту. Генерація стислого переказу тексту – це процес створення стислого та зв'язного резюме заданого тексту, що фіксує його основні ідеї, ключові моменти та важливі деталі. Він полягає в тому, щоб скоротити довгий документ, наприклад, статтю, новину або документ, до коротшої версії, зберігаючи при цьому його зміст і актуальність. Існує два основних підходи до конспектування тексту:

- Витяжне підсумовування (extractive summarization): короткий переказ створюється шляхом вибору та вилучення найважливіших речень або уривків з оригінального тексту.
- Абстрактне підсумовування (abstractive summarization): передбачає розуміння змісту тексту та створення зв'язного короткого переказу шляхом перефразування та переосмислення змісту.

Найбільш раціональний підхід для генерації підсумку тексту – поєднання двох з цих підходів. Першій підхід може бути реалізований завдяки бібліотекам NLTK і Spacy. Для цієї реалізації створюється таблиця частотності слів у тексті, на основі якої в свою чергу будується список балів речень. Речення з найбільшими балами обираються для фінального підсумку. Другий підхід можна реалізувати за допомогою пайплайну summarize з бібліотеки HuggingFace Transformers на основі моделі DistilBart CNN [20].

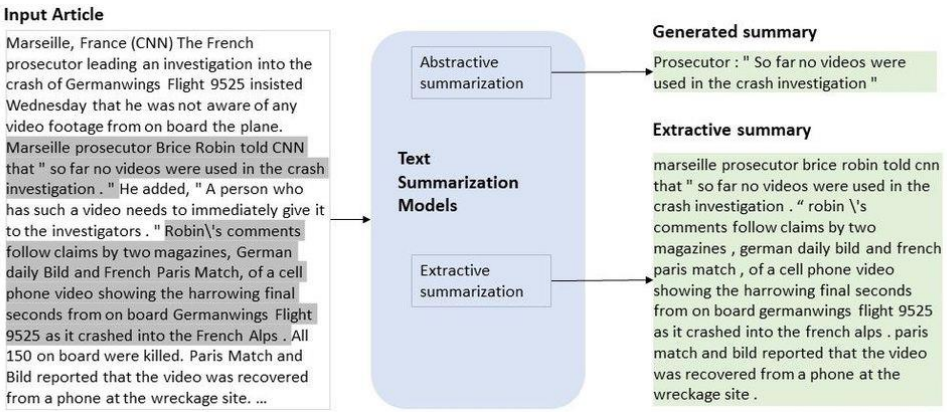

Рисунок 7. Принцип роботи генерації підсумку тексту моделлю DistilBart [21]

РОЗДІЛ 3. ОПИС РЕАЛІЗАЦІЇ ЧАТ-БОТУ

3.1 Аналіз технічного завдання

Технічне завдання полягає в розробці чат-боту для месенджеру Telegram з використанням методів AI і NLP. За мету було взяти розробити бот-«помічник», який зможе допомагати користувачу з деякими повсякденними задачами. Було обрано три таких задачі:

- Пошук прогнозу погоди у певній локації на визначену дату.
- Пошук зображень в мережі за запитом.
- Створення короткого переказу наданого тексту.

Для реалізації цього функціоналу завдання було розбито на наступні підзадачі:

- Створення та навчання моделі для розпізнавання наміру повідомлення, в залежності від теми запиту (прогноз погоди, пошук зображень, короткий переказ).
- Створення та навчання моделі для розпізнавання мови запиту.
- Розробка алгоритму для розпізнавання іменованих сутностей і витягування інформації про місцеположення і дату з запиту про пошук прогнозу погоди.
- Розробка алгоритму для витягування ключових слів з запиту про пошук зображення.
- Розробка моделі для генерації короткого переказу тексту.
- Створення чат-боту за допомогою Telegram Bot API, який прийматиме вхідні повідомлення користувачів для подальшої обробки.
- Інтеграція попередньо розроблених моделей у чат-бот.

Для навчання моделей мають бути зібрані датасети з відповідними даними (вхідні повідомлення з маркуванням наміру і мови), або використані вже існуючі моделі (з бібліотек Spacy і HuggingFace Transformers).

3.2 Пояснення вибору використаних технологій

3.2.1 Основа чат-боту

Для розробки чат-боту була обрана мова програмування Python. Python вважається найбільш влучним вибором для цілей розробки застосунків з AI та NLP завдяки своїй багатій екосистемі бібліотек та фреймворків, присвячених цим галузям. Як фреймворк для розробки Telegram боту був обраний aiogram, тому, що він підтримує асинхронність, активно розробляється та підтримується спільнотою. На його основі додається основний функціонал чат-боту.

Код чат-боту виконується за допомогою диспетчера, який запущений в асинхронному циклу обробки подій. До диспетчера підключені обробники вхідних повідомлень, які надсилають відповіді на команди («/start», «/help»), отримують текст повідомлення для подальшої обробки або обробляють взаємодію з надісланим повідомленням (обрання мови в повідомленні, надісланому після команди «/help»).

3.2.2 Сторонні бібліотеки

Для підтримки методів NLP використовуються вже вищеописані бібліотеки scikit-learn, NLTK, SpaCy і HuggingFace Transformers. Також, для впровадження додаткового функціоналу, були використані вбудовані бібліотеки logging (логування вхідних і вихідних повідомлень), json (зчитування датасетів у форматі JSON для навчання моделей), requests (запити до API Google Image Search і OpenWeatherMap), datetime (обробка об’єктів для зберігання дат).

Із сторонніх бібліотек також були використані pandas (попередня обробка датасетів для використання у навчанні моделей класифікації), joblib (зберігання навчених моделей у файлах) і deepl.

Модуль deepl – це обгортка над DeepL API. DeepL – це сервіс онлайн-перекладу, який використовує алгоритми глибокого навчання для надання перекладів різними мовами [22]. Він використовує передові алгоритми та нейронні мережі для аналізу та розуміння контексту, синтаксису та семантики речень, що дозволяє йому створювати високоякісні переклади з підвищеною точністю та вільністю.

3.2.3 Вибір класифікаторів

На основі результатів тестування класифікаторів наміру і мови були обрані наступні алгоритми класифікаторів з бібліотеки scikit-learn – LinearSVC для класифікатора наміру і RandomForestClassifier для класифікатора мови. Ефективність роботи класифікаторів оцінювалась за наступними метриками:

- Точність (accuracy): відношення true positives до загальної суми.
- Повнота (recall): відношення true positives до суми true positives і false negatives.
- Влучність (precision): відношення true positives до суми true positives і false positives.
- F1-score: середнє гармонійне значення recall і precision.

LinearSVC та RandomForestClassifier - відомі алгоритми класифікації. LinearSVC знаходить найкращу гіперплощину для розділення класів у високовимірному просторі, тоді як RandomForestClassifier створює ліс дерев рішень та агрегує їхні прогнози. LinearSVC фокусується на лінійному розділенні, тоді як RandomForestClassifier обробляє як лінійні, так і нелінійні зв'язки.

3.3 Структура застосунку

Чат-бот уявляє собою проект мовою Python, що складається з наступних частин:

- Функція main(), що містить вхідну точку для початку виконання диспетчера боту.
- Файл config.py, у якому міститься токен Telegram-боту і ключі для використовуваних API.
- Файл bot_commands.py, що містить основні методи для прийняття вхідних повідомлень і відправки повідомлень з кінцевим результатом користувачеві.
- Файл logger_setup.py, який відповідає за збереження логів у файл log.txt.
- Клас GenerateMessage, який відповідає за обробку вхідних повідомлень, а саме розпізнання наміру і виконання дій, в залежності від нього.
- Класи IntentClassifier і LanguageClassifier, що містять методи для навчання моделей, збереження їх у файли і завантаження з них, і використання моделей задля класифікації наміру/мови.
- Клас Translator, що містить методи для розпізнання мови за допомогою LanguageClassifier і перекладу тексту за допомогою DeepL API.
- Класи WeatherRecognition, ImageSearch і TextSummarizer, які містять методи для обробки відповідних повідомлень методами NLP і формування кінцевої відповіді за допомогою запитів до OpenWeatherMap API (для погоди), Google Image Search API (для зображень) і Hugging Face Transformers (для генерації короткого змісту тексту).
- Файли cities_data.json і languages.csv, які містять датасети для тренування моделей класифікації.
- Файл messages.json, який містить тексти попередньо підготовлених повідомлень (для команди «/start» і відповідей на некоректні запити).

Нижче наведена приблизна діаграма принципу роботи застосунку:

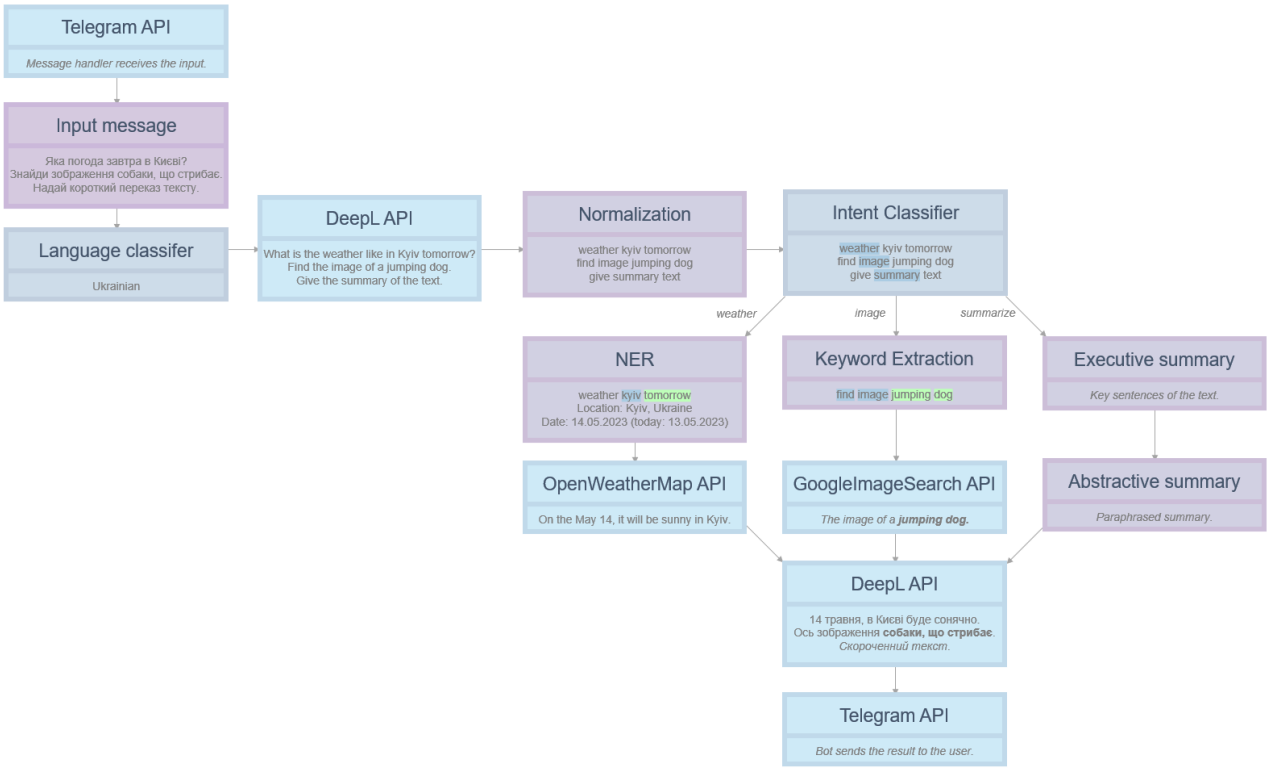

Рисунок 8. Принцип роботи застосунку

3.4 Демонстрація роботи чат-боту

При першому запуску чат-бота показується вікно з короткою інформацією про бот, і також надсилається повідомлення з пропозицією виконати команду «/help» для того, щоб дізнатися функціонал застосунку.

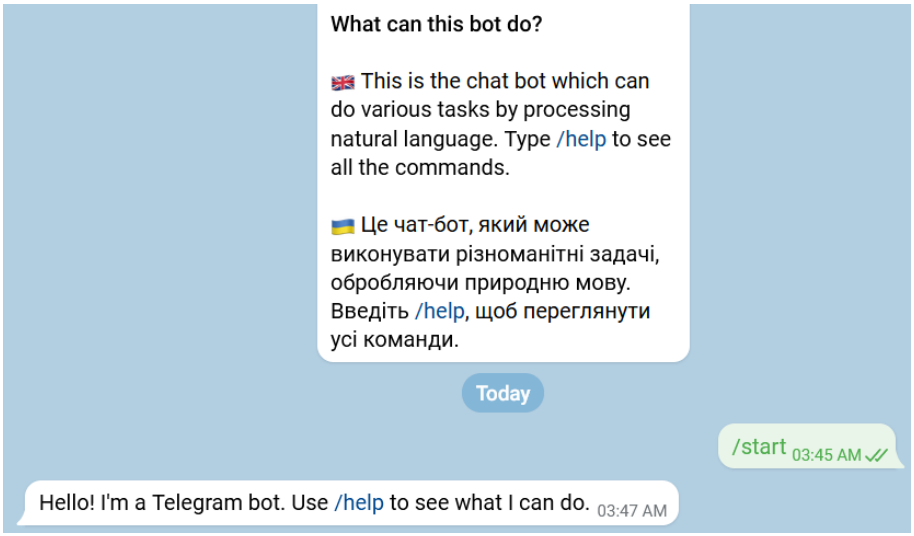

Рисунок 9. Вікно запуску чат-бота

При відправці команди «/help» користувачеві надсилається інтерактивне повідомлення з описом дій, які може виконувати чат-бот і прикладами запитів. При натисканні на кнопки під повідомленням, можна обрати мову відображення інформації з наявних.

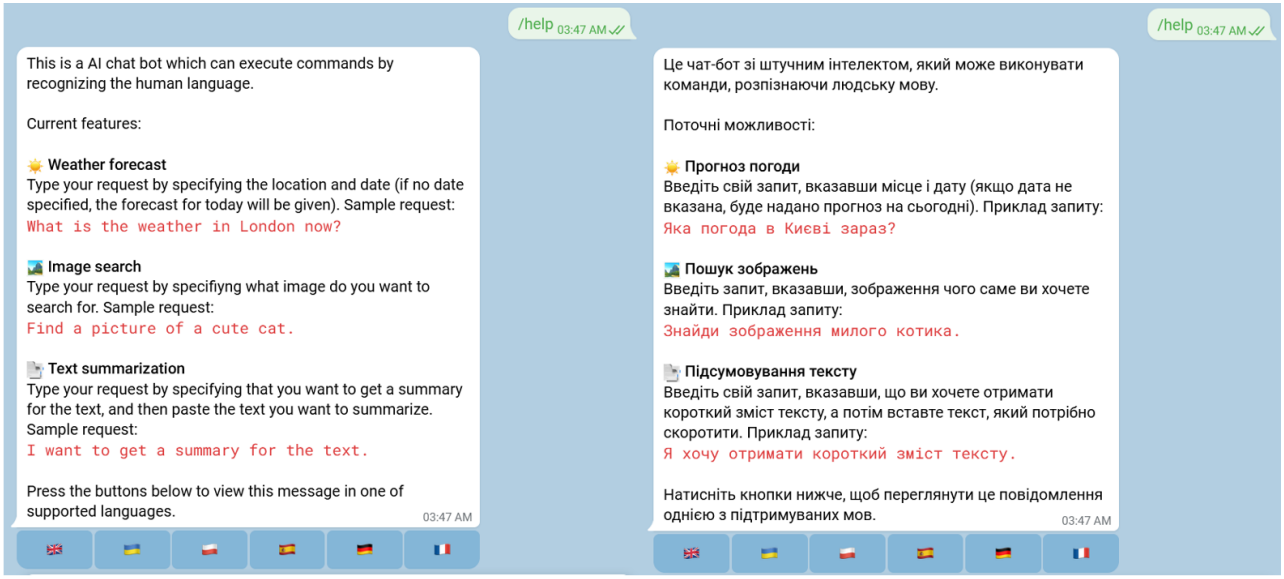

Рисунок 10. Приклад команди «/help» англійською і українською мовами

При відправці повідомлень, пов’язаних з темами запитів, які може обробити чат-бот, він надає відповідну інформацію. Як можна побачити на знімку екрана, при надані запитів, пов’язаних з погодою, чат-бот надає прогноз погоди, базуючись на населеному пункті, вказаному в запиті і даті (якщо дата не вказана, видається прогноз погоди на поточний день.

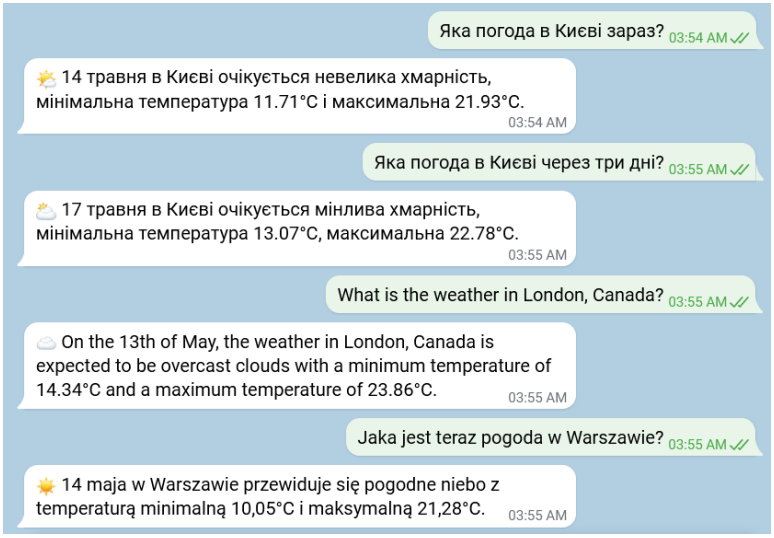

Рисунок 11. Приклади запитів, пов’язаних з прогнозом погоди різними мовами

Так само й обробляються запити, пов’язані з пошуком зображень. Користувач має вказати те, що він хоче знайти зображення (фото, картинку, малюнок тощо) і тему пошуку. Через деякий час, бот надсилає в чат відповідне зображення. Зображення вибираються випадковим чином з перших 10 результатів пошуку.

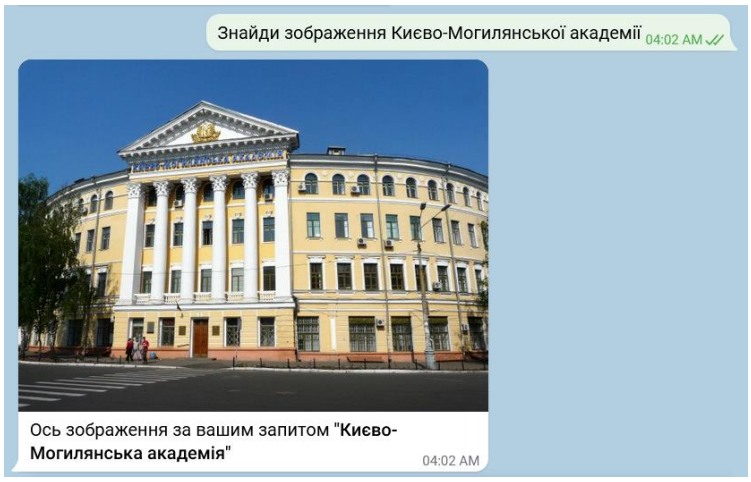

Рисунок 12. Приклад пошуку зображень за допомогою чат-боту

При запиті, пов’язаним з отриманням короткого змісту тексу, після початкового повідомлення, яке має містити намір про скорочення, бот попросить надіслати власне сам текст, якій потрібно скоротити. Після відправки тексту, через деякий час чат-бот надсилає в чат результат.

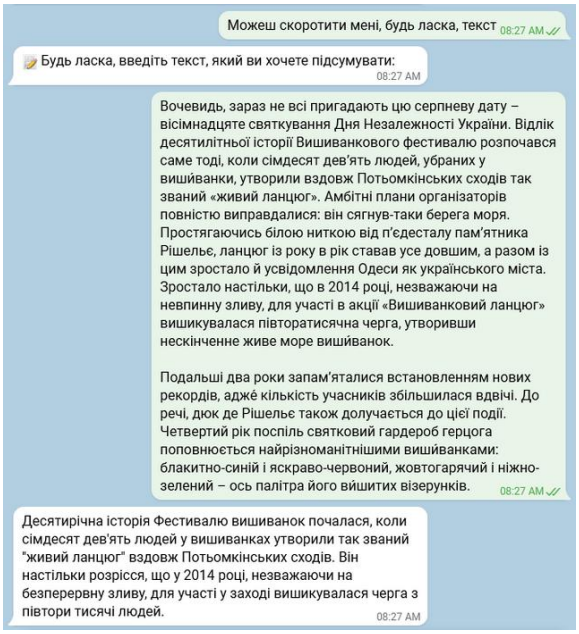

Рисунок 13. Приклад надання короткого змісту тексту чат-ботом

Також у чат-бота наявна обробка некоректних запитів. При надсиланні користувачем запитів, які не пов’язані із функціоналом чат-боту, або при надані некоректних даних, чат-бот відповість повідомленням, яке вказує на помилку у запиті користувача. Приклади некоректних запитів: запит з наміром, який чат-бот не може розпізнати, пошук зображення чи погоди за недійсною чи невказаною назвою або занадто короткий текст для скорочення (менше 500 символів).

Рисунок 14. Обробка некоректних запитів

ВИСНОВКИ

Під час написання курсової роботи, було проведено дослідження різних методів обробки природної мови, інструменти для їх застосування і інтеграції їх у застосунок. Також було досліджено історію розвитку чат-ботів і NLP, їхній сучасний стан і інструменти для реалізації алгоритмів NLP мовою Python. На основі досліджень була здійснена розробка застосунку – бот для платформи Telegram. Результатом став чат-бот, здатний приймати запити користувачів, обробляти їх належним чином і надавати підходящу відповідь за допомогою засобів розуміння і генерації природної мови.

Детально описані методи нормалізації і векторизації тексту, вилучення потрібної інформації за допомогою розпізнавання іменованих сутностей і ключових слів, розробка моделей машинного навчання для класифікації текстів і генерації тексту, інтеграції сторонніх API, а також етапи процесу розробки застосунку. Описано логіку роботи і приклади використання чат-боту.

В перспективі, бот може стати корисним помічником для користувачів завдяки своїй здатності розуміти природну мову і витягувати потрібну інформацію з неї без додаткових дій з боку користувача. Останнім часом, галузь чат-ботів демонструє швидкий розвиток та збільшений попит, тому розробка чат-боту з використанням штучного інтелекту на базі месенджеру Telegram є актуальною темою розробки та дослідження.

СПИСОК ПРИЙНЯТИХ СКОРОЧЕНЬ

NLP – обробка природної мови (natural language processing).

NLU – розуміння природної мови (natural language understanding).

NLG – генерування природної мови (natural language generation).

NER – розпізнавання іменованих сутностей (named entity recognition).

GPT – генеративний попередньо-навчений трансформер (generative pre-trained transformer).

AI – штучний інтелект (artificial intelligence).

API – прикладний програмний інтерфейс (application programming interface).

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Hu K. ChatGPT sets record for fastest-growing user base - analyst note. Reuters. 2023. URL: https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/.
2. Shweta. What is a chatbot? Everything you need to know. Forbes Advisor. 2022. URL: https://www.forbes.com/advisor/business/software/what-is-a-chatbot/.
3. Dixon S. Most popular messaging apps 2023 | Statista. Statista. 2023. URL: https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/.
4. Zemčík T. A brief history of chatbots. DEStech transactions on computer science and engineering. 2019. URL: https://doi.org/10.12783/dtcse/aicae2019/31439.
5. TELUS International. How do chatbots work? An introductory guide. TELUS international customer experience & digital solutions. URL: https://www.telusinternational.com/insights/digital-experience/article/the-future-of-chatbots.
6. Molnár G., Szüts Z. The role of chatbots in formal education. 2018. URL: https://ieeexplore.ieee.org/abstract/document/8524609.
7. Turing A. M. I. — Computing machinery and intelligence. Mind. 1950. Vol. LIX, no. 236. P. 433—460. URL: https://doi.org/10.1093/mind/lix.236.433.
8. Lock S. What is AI chatbot phenomenon ChatGPT and could it replace humans?. The Guardian. 2022. URL: https://www.theguardian.com/technology/2022/dec/05/what-is-ai-chatbot-phenomenon-chatgpt-and-could-it-replace-humans.
9. 10 NLP techniques every data scientist should know. ProjectPro. 2023. URL: https://www.projectpro.io/article/10-nlp-techniques-every-data-scientist-should-know/415#word-embeddings.

10. Roldós I. Major challenges of Natural Language Processing (NLP). MonkeyLearn Blog. 2020. URL: https://monkeylearn.com/blog/natural-language-processing-challenges/.
11. Telegram Bot API. Telegram APIs. 2023 URL: https://core.telegram.org/bots/api.
12. Welcome to pyTelegramBotAPI's documentation. pyTelegramBotAPI Documentation 4.10.0 documentation. 2022 URL: https://pytba.readthedocs.io/en/latest/index.html.
13. Welcome to aiogram's documentation. aiogram 2.25.1 documentation. 2022. URL: https://docs.aiogram.dev/en/latest/.
14. spaCy Usage Documentation. spaCy 101: Everything you need to know. URL: https://spacy.io/usage/spacy-101.
15. Bird S. NLTK documentation release 3.2.5. Readthedocs. 2017. URL: https://buildmedia.readthedocs.org/media/pdf/nltk/latest/nltk.pdf.
16. Scikit-learn documentation. DevDocs API Documentation. 2022. URL: https://devdocs.io/scikit_learn/.
17. Using a machine learning methodology to analyze Reddit posts regarding child feeding information / C. Donelson et al. Journal of child and family studies. 2021. Vol. 30, no. 5. P. 1290—1298. URL: https://doi.org/10.1007/s10826-021-01923-5.
18. Flashgeotext. Wayback Machine. 2022 URL: https://web.archive.org/web/20220327071211/https://flashgeotext.iwpnd.pw/#flashgeotext.
19. Dateparser - python parser for human readable dates. DateParser 1.1.2 documentation. 2014. URL: https://dateparser.readthedocs.io/en/latest/.
20. What is Summarization. Hugging Face — The AI community building the future. URL: https://huggingface.co/tasks/summarization.
21. Deng D. Bootstrap your text summarization solution with the latest release from NLP-recipes. Microsoft Tech Community. 2020. URL:

https://techcommunity.microsoft.com/t5/ai-customer-engineering-team/bootstrap-your-text-summarization-solution-with-the-latest/ba-p/1268809.

22. DeepL Translate. DeepL. URL: https://www.deepl.com/.