Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

Кваліфікаційна робота

за освітнім ступенем «бакалавр»

на тему:

РОЗРОБКА ЧАТ-БОТА З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

Виконав:
здобувач IV курсу
групи ІПЗ-41
спеціальності 121 «Інженерія програмного забезпечення»
Арешин Р.С.

Науковий керівник:
к.т.н., доц. Гаврилюк В.І.

Рівне – 2024

ЗМІСТ

ВСТУП 3
РОЗДІЛ 1. ЧАТ-БОТИ 6
1.1. Визначення і типи 6
1.2. Оброблення мови 10
1.3. Готові моделі чат ботів. Їх переваги та недоліки 11
1.4. Історія розвитку чат-ботів зі штучним інтелектом 13
1.5. Використання чат-ботів зі штучним інтелектом у різних галузях 14
РОЗДІЛ 2. ЗАСТОСУВАННЯ NLP ТА МАШИННЕ НАВЧАННЯ 17
2.1. NLP(Natural Language Processing) засіб для оброблення мови 17
2.2. Машине-навчання 18
2.3. Машине навчання під час використання готових моделей 19
2.4. Порівняння методів машинного навчання 20
2.5. Виклики та проблеми в застосуванні NLP 22
2.6. Використання Node.js для серверної частини чат-бота 23
2.7. Використання AngularJS для розробки інтерфейсу чат-бота 24
2.8. Вплив технологій на продуктивність військових операцій 25
РОЗДІЛ 3. РОЗРОБКА ЧАТ-БОТУ 27
3.1. Підготовка середовища та версії засобів 27
3.2. Початок реалізації 27
3.3. Розробка серверної частини 28
3.4. Додавання функціоналу, для тренування моделі, через стороні готові моделі 31
3.4. Розробка графічної частини 33
3.5. Використання зворотного зв'язку для покращення моделі 37
ВИСНОВКИ 39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 41

ВСТУП

Актуальність роботи: В сучасному світі використання штучного інтелекту і чат-ботів набуває все більшого значення в різних сферах. Чат-боти стали важливим інструментом для спілкування з користувачами в онлайн-середовищі, і вони можуть виконувати різноманітні завдання, від відповідей на запитання користувачів до навчання інформаційному пошуку та підтримки клієнтів.

Навчання чат-бота на основі даних має великий потенціал для покращення якості обслуговування, ефективності спілкування та збільшення задоволеності клієнтів. Збір та аналіз даних користувачів дозволить чат-боту стати більш персоналізованим і адаптивним до потреб кожного конкретного користувача.

Історична довідка: Виникнення чат-ботів відбувалося етапами, починаючи з далеких часів і здобуваючи значну популярність у сучасному інтернет-середовищі. Перші спроби створення програм для імітації розумової діяльності людини належать серединам XX століття, коли розробники експериментували із створенням автоматизованих систем для взаємодії з користувачами.

Одним з ключових кроків у розвитку чат-ботів було створення ELIZA в 1966 році доктором Джозефом Вайзенбаумом. ELIZA використовувала простий алгоритм відповіді на питання та є одним із попередників сучасних чат-ботів. У 1990-х роках із розвитком штучного інтелекту та обчислювальних технологій з'явилися більш складні чат-боти, що мали здатність навчання на основі взаємодії з користувачами.

Мета дослідження:

Метою цього дослідження є розробка алгоритму навчання чат-бота з використанням машинного навчання, спеціально адаптованого до вимог військової сфери. Здійснюючи навчання на великому обсязі військових даних, дослідження спрямоване на створення інтелектуального агента, здатного

ефективно функціонувати в реальному часі, враховуючи стратегічні та тактичні аспекти операцій.

Підходи до навчання та технічні аспекти забезпечення безпеки і стійкості чат-бота в умовах військових викликів будуть вивчені з метою розробки надійного та швидкореагуючого інтелектуального агента, адаптованого до динаміки військових операцій.

Об’єкт дослідження: Об’єктом даного дослідження є розробка чат-бота з використанням машинного навчання на платформі Node.js, спеціально адаптованого для військової сфери. Головною метою є створення інтелектуального агента, здатного ефективно спілкуватися та виконувати завдання в умовах військових операцій.

Предмет дослідження: У цьому дослідженні предметом є вивчення та оптимізація використання машинного навчання в роботі чат-бота, спеціально призначеного для військової діяльності на платформі Node.js. Основна увага буде приділена аналізу інформації, яку надають користувачі чат-боту під час взаємодії, зокрема в контексті військових запитань та вимог.

В рамках поставленої мети, дослідження вирішить наступні завдання:

- Розширений збір та аналіз інформації від користувачів чат-боту, із спеціальним акцентом на військовий контекст.
- Розробка та вдосконалення моделі машинного навчання, що навчатиметься на основі зібраних даних користувачів, із врахуванням військових сценаріїв та запитань.
- Впровадження навченої моделі для покращення функціональності чат-бота, що сприятиме наданню більш точних та релевантних відповідей на запитання військових користувачів.

Це дослідження буде спрямоване на вдосконалення ефективності чат-бота відповідно до конкретних вимог та потреб військової сфери, роблячи акцент на аналізі відповідей та взаємодії у військових контекстах.

Цілі та завдання дослідження:

1. Аналіз військових даних користувачів чат-боту:

- Збір та аналіз інформації від військових користувачів.
- Визначення ключових тем і запитань.

2. Розробка моделі машинного навчання:
   - Створення моделі на основі військових даних користувачів.
   - Удосконалення моделі для відповідей на військові запитання.
3. Оптимізація роботи чат-бота:
   - Використання моделі для поліпшення відповідей чат-бота військовим користувачам.
   - Забезпечення оптимальної відповіді на запитання військового контексту.
4. Забезпечення безпеки та конфіденційності:
   - Вивчення технічних аспектів забезпечення стійкості та конфіденційності військових даних.
   - Розробка та впровадження заходів безпеки в обміні військовою інформацією.
5. Ефективна військова взаємодія:
   - Забезпечення високої ефективності та відповіді чат-бота на військові вимоги.
   - Вдосконалення алгоритмів взаємодії з урахуванням особливостей військового середовища.

Структура роботи. Робота складається з трьох розділів, дев’ятнадцяти підрозділів, висновків та списку використаних джерел. Загальний обсяг роботи становить 41 сторінки.

РОЗДІЛ 1. ЧАТ-БОТИ

1.1. Визначення і типи

Чат-бот є програмою-співрозмовником, яка емулює людське спілкування через текст або голос. Ці програми автоматизують завдання за заданим алгоритмом, ведучи діалог з користувачем, виконуючи його запити або відповідаючи на питання. Перші програми, що імітували спілкування людей, з'явилися ще у 1966 році, наприклад, віртуальний співрозмовник Elisa досить автентично пародіював діалог із психотерапевтом.

Зі зростанням популярності месенджерів у 2010-х чат-боти отримали новий імпульс. Більшість з них працює на платформах відомих месенджерів, таких як Facebook Messenger, Telegram, Viber, Skype, Slack, і можуть функціонувати як окремі додатки або бути вбудованими у функціонал пошукових систем .

Чат-боти застосовуються у різних галузях, таких як електронна комерція, колл-центри та ігрова індустрія. Зазвичай використання чат-ботів в областях електронної комерції та колл-центрів обмежено вузькою спеціалізацією і не може включати широкий спектр спілкування з людьми.

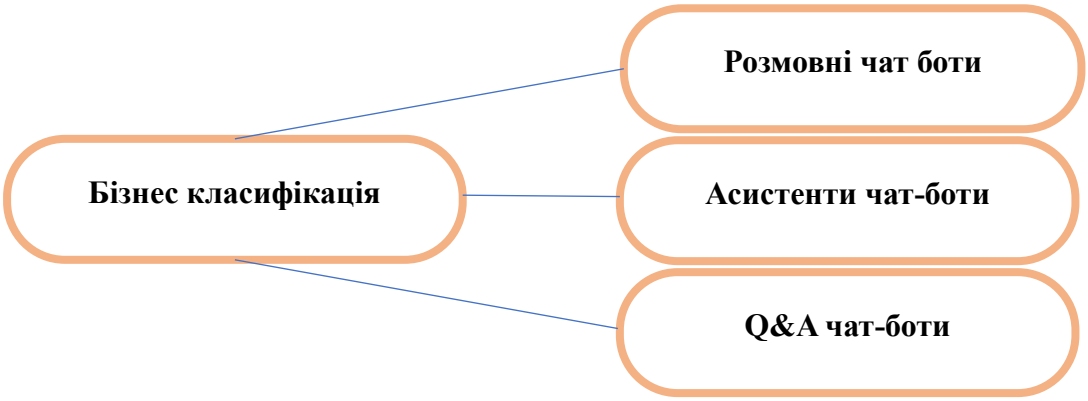
Компанії можуть має різне розуміння того, як класифікувати чат-ботів. Але можна виділити два типи класифікації: бізнес-класифікація чат-бот додатків і класифікація чат-бот додатків за технічним типом. На рисунку 1.1 представлена діаграма бізнес-класифікації чат-ботів.

Розглянемо кожен тип докладніше:

1. **Розмовні чат-боти**: Створені для неструктурованого спілкування, схожого на розмову з людиною, і не мають конкретної мети.

2. **Чат-боти помічники**: Мають конкретно визначену мету і витягують дані з відповідей користувача, необхідні для досягнення цієї мети. Вони можуть використовуватися як заміна або помічники у заповненні веб-форм, наприклад, отримання банківської виписки або оформлення іпотеки онлайн.

3. **Q&A (questions and answers).** Чат-боти призначені для надання простих відповідей відповідно до принципу "1 запитання - 1 відповідь". Вони можуть служити альтернативою розділам часто задаваних питань (FAQ) на різних веб-сайтах.



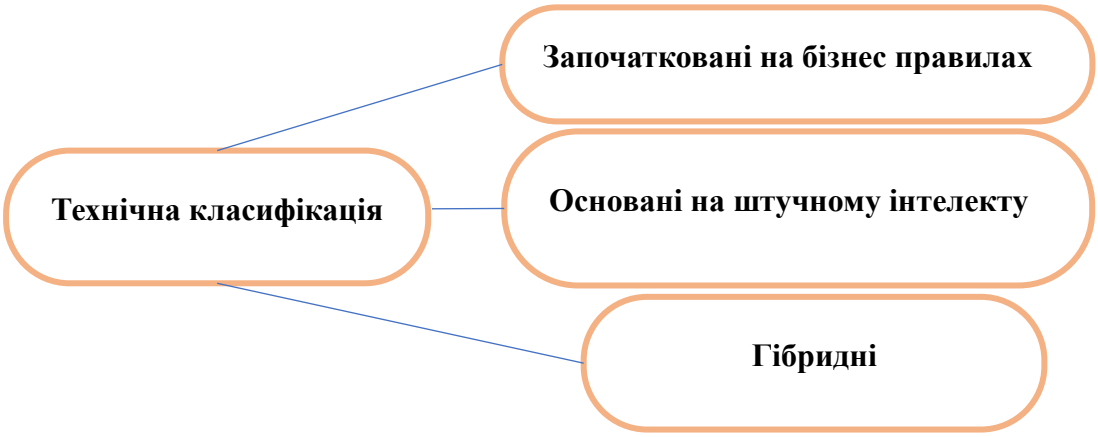


Рисунок 1.1. Бізнес-класифікація чат-бот додатків.

Рисунок 1.2. Технічні типи чат-ботів.

Розглянемо кожен із типів докладніше:

1. **Засновані на бізнес-правилах**. Вони мають деревоподібну структуру діалогу, де розмова з користувачем відбувається за заздалегідь визначеним маршрутом, встановленим розробником. Користувач може приймати рішення під час розмови, але завжди дотримуючись визначеного шляху. Зазвичай такі чат-боти уникають питань, що вимагають відповідей у вільній формі, натомість містять велику кількість кнопок як альтернативу.

2. **Засновані на штучному інтелекті**. Ці чат-боти повністю побудовані на штучному інтелекті (NLP, NLU, NN тощо). На відміну від чат-ботів, заснованих на бізнес-правилах, вони не мають заздалегідь визначеного маршруту розмови. Натомість шлях діалогу визначається на основі тренувальних даних, використаних для навчання моделі машинного навчання. Чат-боти цього типу приймають рішення про запитання та відповіді на основі минулих діалогів, що призводить до потреби у великих наборах даних для "розумного" спілкування. На даний момент існують лише прототипи таких чат-ботів.

3. **Гібридні**. Це поєднання перших двох типів. Гібридні чат-боти ведуть розмову за заздалегідь визначеним маршрутом, але використовують штучний інтелект для розпізнавання намірів користувача та вилучення цінних даних з повідомлень (наприклад, ім'я, дата тощо). Цей тип чат-ботів найбільш поширений у комерційних додатках.

Схему взаємодії користувача та чат-бота можна коротко описати так:

Спочатку користувач адресує свій запит до будь-якого з доступних йому каналів. За запитом стоїть певний намір, інтент, тобто. бажання отримати відповідь на запитання, отримати послугу, товар чи будь-який контент, наприклад музику або відео. Як канали можуть виступати розумні пристрої, помічники, вбудовані в пристрої або мобільні телефони, звичний дзвінок на номер телефону, месенджери або веб-чати.

Далі може знадобитися додаткова обробка або конвертація формату повідомлення. Діалогові платформи завжди працюють із текстом, тоді як ряд каналів передбачають голосове спілкування. За цю конвертацію відповідають платформи ASR (розпізнавання мовлення), TTS (синтез мовлення), інтеграційна система з телефонією. Окремі канали, наприклад, месенджери або помічник Siri в мобільному телефоні, дозволяють поєднувати візуальні інтерактивні елементи (наприклад, кнопки або картки товарів, на які можна натиснути) та природну мову. Для роботи з ними потрібна інтеграція з відповідними API.

Запит, перетворений на текст, надходить у серверну частину. Її завдання – зрозуміти сенс сказаного, вловити інтент користувача і ефективно обробити його, віддавши результат. Для цього серверна частина використовує безліч технологій, таких як нормалізація тексту, морфологічний аналіз, аналіз семантичної близькості сказаного, ранжування гіпотез, виділення іменованих сутностей і, нарешті, формування запитів вже машинною мовою, через сукупність API до зовнішніх баз даних та інформаційних систем. Прикладом таких зовнішніх систем може бути Google Translate, Adobe Photoshop, системи, бази контенту або сервіси, такі як Netflix або Google Play Music. Отримавши дані, діалогова платформа генерує відповідь - текст, голосове повідомлення (за допомогою TTS), включає стрімінг контенту або повідомляє про чинність (наприклад, розміщення замовлення в електронному магазині). Якщо в початковому запиті даних для прийняття рішень щодо подальшої дії недостатньо, засіб NLU ініціює уточнюючий діалог, щоб отримати всі параметри, що бракують, і зняти невизначеність.

Основний цикл обробки запиту клієнта складається з наступних подій та дій:

1. Клієнт надає запит в додаток , відповідно отримує графічне підтвердження запиту.

2. Додаток первино обробляє запит від користувача, та розставляє у відповідні комірки тим самим формуючи запит для серверу.

3. Запит додатку (разом із контекстом) відправляється на обробку , якщо запит був вдало оброблений в додатку (наприклад графічна кнопка) – його обробляє сервер, якщо ж це текстовий запит користувача тоді він переходить в NLU-модуль, де на основі тексту встановлюється інтент (тег\посилання на інформацію\дію).

4.Сервер на основі отриманих вказівок виконує їх(пошук в БД, генерація графіку, генерація запитів на інші ресурси , то що).

5. Виконання скриптів (бізнес-логіки) у відповідності до заданого сценарію та отриманих або згенерованих данних серверу.

6. Генерація текстової відповіді з використанням макропідстановок і функцій узгодження слів природною мовою, та надсилання відповіді в додаток.

7. Надсилання графічної (текст в чаті) відповіді клієнту.

1.2. Оброблення мови

Управління напрямом діалогу є ключовою складовою системи, яка визначає контекст сказаного та встановлює зв'язок із попередніми та наступними висловлюваннями. Цей процес робить фразу більш контекстуалізованою, враховуючи момент її вираження, виконавця, а також інші дані, що можуть бути передані системі разом із запитом, такі як місцезнаходження користувача. Деякі системи також використовують процес наповнення контексту (slot filling), який дозволяє отримувати необхідні дані з фрази користувача, контексту попередніх висловлювань або явно запитувати користувача за відсутності необхідної

Найскладнішим етапом роботи чат-боту являється процес розбору висловлювання(мовлення) клієнта. Цей процес називається NLP - Natural Language Processing, розуміння сенсу запиту.

У найпростішому вигляді процес «розуміння» мови складається з таких основних етапів:

- попередня обробка тексту;
- перебірка тексту з наявною мовною моделлю;
- встановлення інтенту у разі виявлення схожості.

І тут виникає найбільша проблема цього підходу – заповненість мовної моделі, її ергономіка та семантичне наповнення.

Архітектура системи чат-бота може реалізувати наступний підхід до обробки запиту природною мовою:

1. Розбивка тексту на окремі слова.
2. Виправлення помилок (зберігаючи обидва варіанти тексту).

3. Поповнення тексту морфологічними ознаками – визначення лем (нормальної форми слів) та граматичних категорій (грамем).
4. Розширення запиту за допомогою словників синонімів.
5. Додавання інформації про значущість окремих слів (їх ваги).
6. Розширення запиту синтаксичним деревом.
7. Розширення запиту з урахуванням кореферентності (розпізнавання займенників).
8. Визначення іменованих сутностей.
9. Класифікація запиту за допомогою двох підходів (можуть використовуватися одночасно): на основі прикладів фраз та алгоритмів машинного навчання, а також на основі формальних правил (шаблонів).
10. Ранжування гіпотез класифікації з урахуванням поточного контексту розмови.
11. Заповнення інформаційних "слотів" – параметрів запиту, переданих у фразі користувача.

1.3. Готові моделі чат ботів. Їх переваги та недоліки

На сьогоднішній день існує безліч готових моделей для створення чат-ботів, таких як Dialogflow від Google, Watson Assistant від IBM, Microsoft Bot Framework, Rasa та інші. Ці моделі надають розробникам інструменти для швидкого впровадження інтелектуальних агентів у різні проекти, включаючи військові. Вони забезпечують функціональні можливості для розуміння та обробки природної мови, інтеграції з різними платформами та налаштування діалогів.

Переваги використання готових моделей у військових проектах:

1. **Швидкість впровадження**: Готові моделі дозволяють значно скоротити час на розробку і впровадження чат-бота. Вони вже мають

попередньо налаштовані функціональності, що дозволяє швидко запустити чат-бота у використання.

2. **Випробувана якість**: Такі моделі вже були протестовані і випробувані в реальних умовах. Це дозволяє уникнути багатьох потенційних помилок, які могли б виникнути при розробці з нуля.

3. **Підтримка та оновлення**: Готові рішення часто підтримуються командами розробників, які забезпечують регулярні оновлення та покращення. Це означає, що чат-бот завжди буде відповідати сучасним вимогам і технологіям.

4. **Масштабованість**: Багато готових моделей мають можливість масштабування, що дозволяє легко адаптувати їх під зростаючі вимоги військових операцій.

5. **Економічна ефективність**: Використання готових моделей може бути більш економічно ефективним, оскільки розробка з нуля часто вимагає значних ресурсів і часу.

Недоліки використання готових моделей у військових проектах:

1. **Обмежена налаштовуваність**: Готові моделі можуть мати обмежені можливості для кастомізації. Це може бути проблемою, якщо ваші вимоги специфічні і відрізняються від стандартних функціональностей, що пропонуються моделлю. Військові проекти часто вимагають унікальних рішень, які не завжди легко реалізувати в межах готової моделі.

2. **Залежність від постачальника**: Використання готових моделей створює залежність від постачальника. У випадку, якщо постачальник припиняє підтримку або змінює умови користування, це може створити проблеми для вашого проекту.

3. **Безпека і конфіденційність**: Готові рішення можуть мати проблеми з безпекою та конфіденційністю даних, особливо якщо вони обробляють чутливу військову інформацію. Важливо ретельно перевіряти ці аспекти перед впровадженням.

4. **Відсутність унікальності**: Використання стандартних рішень може призвести до того, що ваш чат-бот буде схожим на інші, що знижує його унікальність і потенційно впливає на військові операції.

5. **Обмежена інтеграція**: Деякі готові моделі можуть мати обмеження щодо інтеграції з іншими військовими системами та сервісами, що може ускладнити їх використання в більш складних військових інфраструктурах.

Ці переваги та недоліки слід враховувати при виборі готових моделей для військових проектів, особливо якщо мова йде про специфічні сфери, де вимоги до безпеки та налаштовуваності можуть бути дуже високими.

1.4. Історія розвитку чат-ботів зі штучним інтелектом

Розвиток технологій чат-ботів, зокрема чат-ботів зі штучним інтелектом, можна поділити на кілька ключових етапів. Початок був у 1960-х роках, коли Джозеф Вайценбаум у 1966 році в MIT створив першого чат-бота, відомого як ELIZA. ELIZA була здатна симулювати розмову з психотерапевтом, використовуючи прості алгоритми розпізнавання ключових слів і шаблонів.

У 1970-1980-ті роки з'являються перші інтерактивні комп'ютерні ігри, де використовуються елементи чат-ботів. Також розвиваються експертні системи, здатні вести діалоги у вузькоспеціалізованих областях.

З розвитком Інтернету в 1990-х роках чат-боти починають використовуватись для онлайн-підтримки користувачів. В цей час з'являється відомий бот ALICE (Artificial Linguistic Internet Computer Entity), який базувався на технології AIML (Artificial Intelligence Markup Language).

У 2000-х роках з'являються перші чат-боти зі штучним інтелектом, які використовують машинне навчання для поліпшення своїх відповідей. Прикладом є бот SmarterChild, який був доступний через платформи AOL Instant Messenger і MSN Messenger.

Зі зростанням потужностей обчислювальної техніки та розвитком глибокого навчання у 2010-х роках, чат-боти зі штучним інтелектом стають все більш складними і точними. В цей час з'являються такі системи, як Siri від Apple, Google Assistant, Amazon Alexa та Microsoft Cortana. Ці чат-боти використовують технології обробки природної мови (NLP) та глибокого навчання для розпізнавання голосу, розуміння контексту та надання релевантних відповідей.

Сучасні чат-боти у 2020-х роках, такі як GPT-3 від OpenAI, використовують передові алгоритми машинного навчання та величезні обсяги даних для забезпечення високої якості взаємодії з користувачами. Вони здатні вести більш природні та глибокі розмови, адаптуватись до стилю спілкування користувачів та надавати більш точні і релевантні відповіді.

1.5. Використання чат-ботів зі штучним інтелектом у різних галузях

Чат-боти зі штучним інтелектом знайшли широке застосування у багатьох галузях завдяки своїй здатності автоматизувати комунікаційні процеси, підвищувати ефективність роботи та забезпечувати високий рівень обслуговування клієнтів.

В охороні здоров'я чат-боти зі штучним інтелектом допомагають у плануванні прийомів до лікарів, надають інформацію про медичні послуги, моніторять стан здоров'я пацієнтів і надають рекомендації щодо здорового способу життя. Прикладом є **HealthTap** та **Babylon Health**, які здійснюють попередню діагностику на основі симптомів і рекомендують подальші дії.

У фінансовому секторі банки та фінансові установи використовують чат-ботів зі штучним інтелектом для надання інформації про рахунки, здійснення транзакцій, підтримки клієнтів у режимі 24/7 та запобігання шахрайству. Чат-боти, такі як **Erica** від Bank of America та **Cleo**, аналізують фінансові дані, надають персоналізовані рекомендації щодо інвестицій та допомагають у вирішенні фінансових питань.

В освіті чат-боти зі штучним інтелектом допомагають студентам з пошуком навчальних матеріалів, відповідають на питання щодо курсових робіт, а також надають підтримку під час дистанційного навчання. Прикладом є **Duolingo**, який адаптується до стилю навчання кожного студента, надає індивідуальні рекомендації та оцінює прогрес у навчанні.

В електронній комерції чат-боти зі штучним інтелектом використовуються для допомоги покупцям у виборі товарів, надання інформації про замовлення, здійснення продажів та підтримки клієнтів. **Shopify's Kit** та **H&M's Ada** аналізують поведінку користувачів, надають персоналізовані рекомендації та покращують загальний досвід покупок.

У сфері підтримки клієнтів багато компаній використовують чат-ботів зі штучним інтелектом для автоматизації служби підтримки, що дозволяє швидко відповідати на запити клієнтів, вирішувати проблеми та надавати необхідну інформацію. Прикладом є **Zendesk's Answer Bot** та **IBM Watson Assistant**, які можуть обробляти великий обсяг запитів одночасно, що підвищує ефективність роботи служб підтримки.

В туризмі і готельному бізнесі чат-боти зі штучним інтелектом допомагають з бронюванням номерів, надають інформацію про місцеві визначні пам'ятки, ресторани та послуги, а також підтримують клієнтів під час подорожей. **Booking.com's Booking Assistant** та **Expedia's Virtual Agent** надають персоналізовані рекомендації на основі уподобань користувачів та покращують загальний досвід подорожей.

У медіа та розвагах чат-боти зі штучним інтелектом використовуються для взаємодії з аудиторією, надання рекомендацій щодо контенту, організації конкурсів та проведення опитувань. **Spotify's Chatbot** та **Disney's Zootopia Chatbot** аналізують вподобання користувачів, надають персоналізований контент та підвищують рівень залучення аудиторії.

У військовій сфері чат-боти зі штучним інтелектом використовуються для різних завдань, включаючи координацію операцій, надання оперативної інформації та підтримку зв'язку між підрозділами. Військові чат-боти можуть допомагати у проведенні розвідки, аналізувати дані з різних джерел, надавати прогнози щодо можливих загроз та підтримувати прийняття рішень у реальному часі. Вони також можуть бути інтегровані в системи управління бойовими діями для покращення ефективності та швидкості реагування.

Ці приклади демонструють широкі можливості використання чат-ботів зі штучним інтелектом у різних сферах, що сприяє підвищенню ефективності роботи, зниженню витрат та покращенню якості обслуговування клієнтів.

РОЗДІЛ 2. ЗАСТОСУВАННЯ NLP ТА МАШИННЕ НАВЧАННЯ

2.1. NLP(Natural Language Processing) засіб для оброблення мови

Natural Language Processing (NLP) визначає новий рівень інтеракції в сучасних чат-ботах, надаючи їм здатність розуміти та взаємодіяти з природною мовою користувачів. Завдяки NLP, чат-боти можуть не лише розуміти окремі слова, але й враховувати контекст розмови, забезпечуючи більш точні та змістовні відповіді.

Основна перевага NLP полягає у можливості визначення намірів користувачів. Чат-боти, використовуючи NLP, можуть ідентифікувати запитання та потреби користувачів, реагуючи на різні форми висловлення та варіації відповідей. Це особливо важливо в умовах, коли одна й та сама інформація може бути виражена по-різному залежно від контексту розмови або мовних особливостей користувача.

Крім того, NLP дозволяє чат-ботам обробляти природну мову на більш глибокому рівні. Вони можуть розпізнавати граматичні структури, виявляти семантичні зв’язки між словами та фразами, що допомагає краще розуміти значення запитів користувачів. Це також включає можливість виявлення емоційного забарвлення повідомлень, що може бути корисним для створення більш людяного та чуйного інтерфейсу.

Застосування NLP допомагає чат-ботам постійно оновлювати свій словниковий запас, враховуючи нові терміни та вирази, що допомагає підтримувати актуальність мовлення. Це особливо важливо у динамічних сферах, де нові терміни можуть з'являтися швидко, наприклад, у технічних галузях або популярній культурі.

Сучасні моделі NLP, такі як BERT, GPT-3 та інші, використовують передові методи глибокого навчання, що дозволяє їм аналізувати величезні обсяги

текстових даних та вчитися на них. Це забезпечує високу точність розпізнавання мови та генерування відповідей, що є ключовим для успішної взаємодії з користувачами.

Інтеграція NLP з іншими технологіями, такими як машинне навчання та нейронні мережі, дозволяє створювати чат-ботів, які можуть адаптуватися до стилю спілкування кожного користувача, надавати персоналізовані рекомендації та забезпечувати більш інтерактивний досвід. Це підвищує задоволеність користувачів та ефективність використання чат-ботів у різних галузях, включаючи бізнес, освіту, охорону здоров'я та багато інших.

Загалом, застосування NLP у розробці чат-ботів дозволяє створювати більш розумні та адаптивні системи, здатні ефективно взаємодіяти з користувачами, розуміти їхні потреби та надавати якісні та релевантні відповіді. Це відкриває нові можливості для впровадження чат-ботів у різні сфери, сприяючи підвищенню ефективності комунікації та задоволеності користувачів.

2.2. Машине-навчання

Машинне навчання (МН) стало необхідною складовою для подальшого розвитку та вдосконалення чат-ботів. У вашій курсовій роботі розгляньте наступні аспекти, пов'язані з використанням МН у чат-ботах:

1. **Навчання з Нагляду** - Моделі чат-ботів можуть бути навчені на основі попередньо позначених даних, враховуючи правильні відповіді та взаємодії для підвищення ефективності.
2. **Не навчання з Нагляду** - Машинне навчання дозволяє чат-ботам аналізувати непозначені дані, виявляти закономірності та вдосконалювати свої навички.

3. **Пів наглядне Навчання** - Застосування обох типів навчання дозволяє чат-ботам пристосовуватися до різних сценаріїв та отримувати більш глибоке розуміння користувальницьких потреб.
4. **Прогресивне Навчання** - Чат-боти можуть вдосконалювати свої навички через час, здобуваючи додатковий досвід та оновлюючи свою модель на основі нових даних.
5. **Розпізнавання Патернів** - МН дозволяє чат-ботам розпізнавати та аналізувати патерни в поведінці користувачів для покращення інтеракцій.
6. **Оптимізація Рішень** - Чат-боти можуть застосовувати МН для оптимізації прийняття рішень, адаптуючи їх до змінюючись умов та вимог користувачів.

Ці аспекти МН визначають високий потенціал для розвитку інтелектуальних та динамічних чат-ботів, що можуть ефективно взаємодіяти з користувачами та адаптуватися до різних сценаріїв взаємодії.

2.3. Машине навчання під час використання готових моделей

Машинне навчання відіграє ключову роль у функціонуванні чат-ботів, особливо при використанні готових моделей. Готові моделі, такі як Dialogflow від Google, Watson Assistant від IBM, Microsoft Bot Framework та інші, надають розробникам вже навчений інструментарій для швидкого впровадження інтелектуальних агентів у різні проекти, включаючи військові.

Однією з основних переваг використання готових моделей є наявність великого набору попередньо оброблених даних. Ці моделі мають вже вбудовану базову функціональність та знання, які були зібрані і систематизовані на основі великої кількості інформації з різних джерел. Це дозволяє значно скоротити час на навчання, оскільки базова функціональність вже налаштована, і розробникам залишається лише додати специфічні дані для досягнення потрібного рівня точності.

Використання готових моделей дозволяє адаптувати їх до специфічних потреб користувачів, наприклад, у військових проектах, де необхідна адаптація до

специфічної термінології або сценаріїв. Ці моделі також підтримують можливість автоматичного вдосконалення на основі нових даних, що дозволяє чат-ботам покращувати свої відповіді з часом, адаптуючись до змін у вимогах і поведінці користувачів.

Крім того, використання готових моделей може бути економічно вигідним, оскільки не потребує значних витрат на розробку з нуля. Це особливо важливо для військових проектів з обмеженим бюджетом. Готові моделі можуть бути навчени на основі попередньо позначених даних, що забезпечує правильні відповіді та взаємодії для підвищення ефективності чат-бота у військових умовах. Вони також дозволяють чат-ботам пристосовуватися до різних сценаріїв і отримувати більш глибоке розуміння користувальницьких потреб у військових операціях, вдосконалюючи свої навички з часом і розпізнаючи патерни в поведінці користувачів.

Важливою перевагою використання готових моделей є можливість покращення локальної моделі. Інтеграція з готовою моделлю дозволяє отримати доступ до великого обсягу знань та передових технологій, які важко або неможливо розробити самостійно. Це значно підвищує ефективність і точність локальної моделі, оскільки вона може використовувати переваги вже існуючих рішень та адаптувати їх до специфічних потреб військових проектів.

Таким чином, використання готових моделей для машинного навчання у військових проектах надає значні переваги, такі як прискорене впровадження, економія коштів і можливість автоматичного вдосконалення. Однак важливо враховувати специфічні вимоги безпеки та конфіденційності при роботі з чутливими військовими даними

2.4. Порівняння методів машинного навчання

Існують різні методи машинного навчання, кожен з яких має свої особливості, переваги та недоліки. До основних методів належать наглядове навчання, безнаглядове навчання та напівнаглядове навчання. У чат-ботах ці методи

використовуються для поліпшення точності та ефективності взаємодії з користувачами.

Наглядове навчання передбачає використання позначених даних для тренування моделей. Це означає, що кожен вхідний приклад у тренувальному наборі супроводжується правильним виходом. Серед переваг наглядового навчання можна відзначити високу точність та прогнозованість, оскільки моделі на основі наглядового навчання зазвичай мають високу точність завдяки використанню позначених даних. Однак, цей метод має і недоліки, такі як потреба у великій кількості позначених даних та висока вартість позначення великих обсягів даних. У чат-ботах наглядове навчання використовується для визначення інтенцій та класифікації тексту, що допомагає розпізнавати наміри користувачів на основі попередньо позначених прикладів.

Безнаглядове навчання використовує непозначені дані для виявлення прихованих структур у даних. Мета полягає у виявленні патернів і зв'язків без попередньо відомих відповідей. Перевагами цього методу є менші вимоги до даних, оскільки він не потребує позначених даних, що значно знижує витрати на збір та обробку. Крім того, безнаглядове навчання може виявляти приховані структури та патерни у даних, які важко побачити за допомогою наглядового навчання. Недоліками є низька точність порівняно з наглядовим навчанням та складність інтерпретації результатів. У чат-ботах безнаглядове навчання використовується для кластеризації запитів користувачів та виявлення аномалій.

Напівнаглядове навчання є комбінацією наглядового та безнаглядового навчання. Використовує як позначені, так і непозначені дані для тренування моделей. Це дозволяє зменшити потребу у великій кількості позначених даних, ефективно використовуючи невелику кількість позначених даних разом з великою кількістю непозначених даних. Напівнаглядове навчання часто досягає вищої точності, ніж безнаглядове навчання, завдяки використанню позначених даних. Однак, цей метод вимагає складних алгоритмів і методів для

ефективного використання обох типів даних. У чат-ботах напівнаглядове навчання використовується для поліпшення моделей на великому обсязі непозначених даних з мінімальною кількістю позначених даних, а також для генерації нових позначених даних на основі існуючих непозначених даних.

2.5. Виклики та проблеми в застосуванні NLP

Застосування обробки природної мови (NLP) у чат-ботах стикається з низкою викликів і проблем, які можуть впливати на точність і ефективність взаємодії з користувачами. До основних проблем належать розуміння контексту, багатомовність, ідіоматичні вирази та інші.

Однією з найскладніших задач NLP є розуміння контексту. Важливо не лише визначити, що сказано, але й зрозуміти, що малося на увазі в конкретній ситуації. Проблеми виникають через відсутність контексту у коротких або незавершених повідомленнях та множинні значення слів і фраз, які можуть мати кілька значень залежно від контексту. Для подолання цих проблем використовуються глибокі нейронні мережі, такі як BERT або GPT-3, які здатні враховувати контекст при обробці тексту. Також важливо враховувати історію розмови для кращого розуміння поточного контексту.

Обробка природної мови різними мовами є складною задачею через різні граматичні структури, лексичний склад та інші особливості. Проблеми виникають через різні структури речень у різних мовах та обмежений обсяг навчальних даних для деяких мов. Для подолання цих проблем використовуються багатомовні моделі, такі як mBERT, які здатні обробляти текст на різних мовах, а також машинний переклад для конвертації тексту на одну мову перед обробкою.

Ідіоматичні вирази та фразеологізми можуть бути важкими для розуміння моделями NLP, оскільки їх значення не випливає з значення окремих слів. Проблеми виникають через невідповідність буквального значення ідіом та варіативність ідіоматичних виразів у різних регіонах. Для подолання цих

проблем використовуються контекстні моделі, які враховують контекст при обробці тексту, а також розширення словника для включення ідіом у навчальні дані та їх пояснення для моделей.

Використання різних методів машинного навчання та обробки природної мови у чат-ботах дозволяє підвищити точність та ефективність їх роботи. Проте ці методи також стикаються з низкою викликів, які вимагають подальших досліджень та вдосконалень для досягнення кращих результатів у розумінні та обробці людської мови.

2.6. Використання Node.js для серверної частини чат-бота

Node.js - це відкрите середовище для виконання JavaScript-коду на сервері. Воно побудоване на двигуні V8 від Google Chrome та відоме своєю асинхронною та неблокуючою архітектурою, що робить його ідеальним для створення ефективних та масштабованих серверних додатків.

Основні характеристики Node.js включають асинхронність та неблокуючу обробку подій, що дозволяє обробляти багато запитів одночасно. Це важливо для чат-ботів з великою кількістю користувачів. Використання JavaScript на стороні сервера спрощує розробку та дозволяє зменшити розрив між клієнтським та серверним кодом. Модульність та пакетний менеджер npm дозволяють розділяти код на поновлювані компоненти, а платформонезалежність підтримує роботу на різних операційних системах.

Node.js є вибором для розробки серверної частини чат-ботів завдяки таким перевагам:

- **Асинхронна взаємодія**: дозволяє обробляти багато запитів одночасно, що важливо для реалізації реального часу та інтерактивних чат-ботів.
- **Легка інтеграція з WebSocket**: WebSocket підтримується за замовчуванням, сприяючи миттєвій взаємодії з користувачами через чат-бот.

- **Швидкість розробки та простота використання**: використання JavaScript та велика кількість доступних модулів прискорюють процес розробки.
- **Ефективне використання ресурсів**: забезпечує ефективне використання ресурсів сервера, що важливо для масштабування чат-ботів.

Таким чином, Node.js є потужним інструментом для розробки серверної частини чат-ботів, який забезпечує ефективність та зручність в розробці веб-додатків в реальному часі.

2.7. Використання AngularJS для розробки інтерфейсу чат-бота

Поєднання AngularJS та Node.js відкриває перед розробниками широкі можливості для створення чат-ботів, які не лише ефективно взаємодіють з користувачами, але й мають динамічний та інтуїтивно зрозумілий інтерфейс. Обидві технології відомі своєю асинхронною обробкою та здатністю працювати в режимі реального часу, що робить їх ідеальним вибором для створення повноцінних чат-ботів.

Асинхронна взаємодія та реальний час - обидві технології володіють потужними механізмами асинхронного програмування, що сприяє ефективній та миттєвій взаємодії чат-бота з користувачами в режимі реального часу. Модульність та розширення - AngularJS забезпечує легку реалізацію модульної архітектури із компонентами, що легко розширюються. Це, спільно з Node.js, дозволяє легко додавати та розвивати функціонал чат-бота. Спільна мова – JavaScript - обидві технології використовують JavaScript, створюючи консистентне середовище для розробки та полегшуючи взаємодію між фронтендом та бекендом. Інтерактивний адміністративний інтерфейс - AngularJS може служити як інструмент для створення адміністративного інтерфейсу чат-бота, де адміністратор може ефективно налаштовувати та управляти роботою бота. Легка інтеграція та керування залежностями - Node.js разом із пакетним менеджером npm забезпечує легку інтеграцію різних

бібліотек та сервісів, а AngularJS дозволяє ефективно керувати клієнтським кодом та залежностями.

Поєднання цих двох технологій відкриває нові перспективи для створення чат-ботів, які не лише вражають своєю функціональністю, але й приваблюють візуально та забезпечують зручне управління.

2.8. Вплив технологій на продуктивність військових операцій

Впровадження нових технологій, таких як машинне навчання та обробка природної мови (NLP), суттєво впливає на ефективність і продуктивність військових операцій. Завдяки цим технологіям можна автоматизувати багато процесів, забезпечуючи швидкий доступ до інформації, підвищуючи точність рішень та знижуючи ризики.

Машинне навчання дозволяє аналізувати великі обсяги даних у реальному часі, виявляти тенденції та аномалії, що можуть свідчити про потенційні загрози. Це забезпечує швидке реагування на змінювані умови на полі бою, дозволяючи військовим підрозділам діяти більш оперативно і ефективно. Впровадження машинного навчання у військові системи управління дає змогу оптимізувати логістику, управління ресурсами та планування операцій.

NLP, у свою чергу, покращує комунікацію та взаємодію між військовими підрозділами, забезпечуючи розуміння і аналіз природної мови. Чат-боти зі штучним інтелектом можуть допомагати в обробці запитів, надавати інформацію про місцевість, погодні умови, стан військової техніки тощо. Вони також можуть використовуватися для навчання та тренувань, забезпечуючи інтерактивні навчальні програми та підтримуючи безперервний процес навчання.

Загалом, впровадження машинного навчання та NLP у військових операціях підвищує оперативність, ефективність та координацію дій військового персоналу, що сприяє успішному виконанню місій та знижує ризики.

РОЗДІЛ 3. РОЗРОБКА ЧАТ-БОТУ

3.1. Підготовка середовища та версії засобів

Найважливішим кроком для створення будь якого застосунку є налаштування програмного середовища , та встановлення залежностей.

Для реалізації проекту використані наступні версії засобів :

1. Visual Studio Code 1.85.1v - в якості самого середовища.
2. Node.js 18.15.0v.
3. AngularJS 1.8.2'v.
4. Node-NLP 4.27.0v.
5. Socket.IO 4.7.4v.
6. SQLLite 5.1.7v.

Цей список повністю відповідає актуальності коду на даний момент , і є важливим розуміння що з оновленням засобів, можуть змінитися як властивості об’єктів, так і їх методи.

3.2. Початок реалізації

Для реалізації використано клієнт-серверну архітектуру. На клієнті реалізується ввід інформації та отримання відповіді від сервера. На сервері обробку введеної інформації, її зберігання та генерації відповідей на основі розробленої моделі.

За основу комунікації між візуальною частиною (FE) та серверною частиною (BE) ми будемо використовувати сокети (WebSocket) , для комунікації в реальному часі , а також аутентифікації користувача через простий алгоритм з використанням ідентифікатора сокета (socket.id). Отримані дані ми будемо обробляти за допомогою NLP (Natural Language Processing) , отримані

результати відправляти для користувача, а також зберігати до відповідної таблиці, для подальшого машинного тренування.

Подана схема може бути відображена наступним чином:

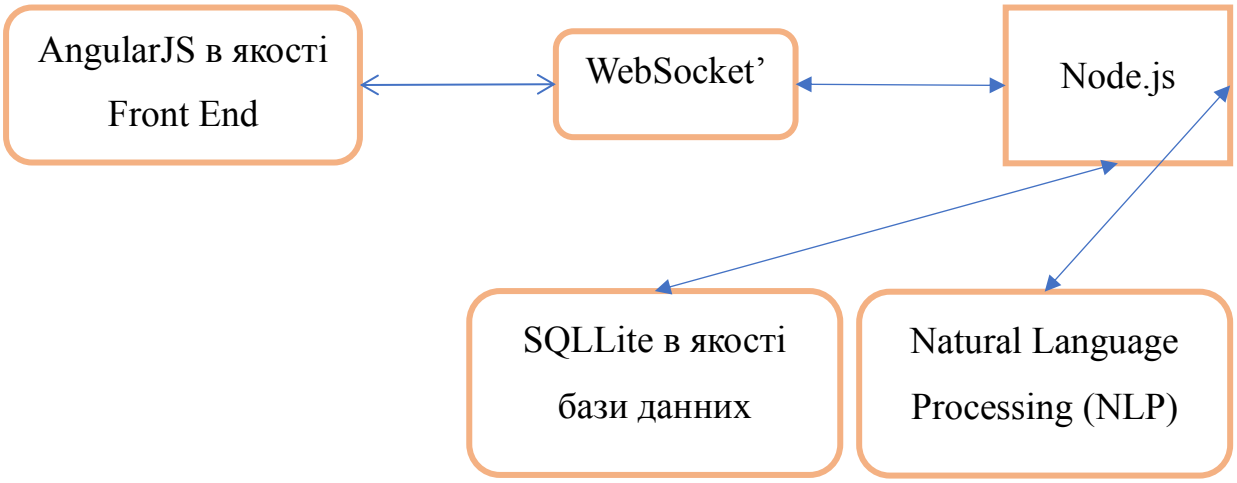


Рисунок 1.3.Схема комунікації між засобами.

3.3. Розробка серверної частини

Оскільки вибрано серверне рішення на основі Node.js, що включає в себе інтеграцію з пакетним менеджером NPM (Node Package Manager), розпочнемо з використання цього інструменту. Для початку встановимо необхідні пакети, зокрема:

npm install sqlite3 – база данних;

npm install node_nlp – інтерпритація NLP для node.js;

npm install express – фреймворк , котрий в нашому випадку буде основою для налаштувань серверу сокетів;

npm install socketio – відповідно пакет з сокетами;

npm install cors – і в моєму випадку пакет для використання CORS(Cross-Origin Resource Sharing), щоб не розділяти по доменах сервер і застосунок;

Для функціонування пакетів, відповідно потрібно наявність самого Node.js та його інструмент NPM. Після встановлення необхідних пакетів, створюємо новий файл для виконання, в даному випадку назву файлу вибрано як “botchat.js” та приступаємо до створення самого коду.

Після успішного підключення необхідних пакетів та налаштування серверу та веб-сокетів, розглянемо кілька можливих сценаріїв написання коду. З моєї точки зору, найоптимальнішим варіантом є почати зі створення запиту до таблиці, яка виступає основним джерелом даних. Для зберігання інформації використовується таблиця з чотирма стовпцями: Id (порядковий номер запису), data (місце для ймовірних запитань користувача), teg (інтент, розглядуваний як мітка або ключове слово для відповіді) та answer (інформація, яку бот повинен повернути відповідно до питання).

Цю таблицю можна відобразити наступним чином :

Id	data	teg	answer
1	Привіт	вітання	Доброго дня
2	Бажаю міцного	вітання_військ	І вам бажаю міцного

Таблиця 1. Таблиця бази данних

Наступним кроком буде задання для мовної моделі інформації отриману з бази даних, для її машинного навчання :

```
const manager = new NlpManager({languages: ['uk']});
db.all('SELECT id, data, teg, answer FROM chatbot_ai', [], (err, rows) => {
    if (err) {
        throw err;
    }
    rows.forEach(row => {
        const { Id, data, teg, answer } = row;
        manager.addDocument('uk',data,teg);
        manager.addAnswer('uk',teg,answer);
    });
});
```

Тут потрібно звернути увагу на два методи в об’єкту maganer котрий оголосили, а саме: addDocument для додавання очікуваних питань та інтент, addAnswer для додавання відповіді по відповідному інтенту.

Тепер важливим етапом є тренування моделі, на основі отриманих даних :

```
manager.train().then(async () => {
    await manager.save();
    console.log('Модель успішно тренована та збережена');
});
```

Після тренування моделі, створюється готова мовна модель для використання, проте ще не створено функціоналу для комунікації з користувачем.

Для комунікації з застосунком для прийняття і відображення інформації, використовується з’єднання сокетів :

```
socket.on('message', async(data)=>{
    try{
        await manager.train();
        await manager.save();
        let response=await manager.process('uk', data);
        if(response.answer)
        {
            socket.emit('message', response.answer);
        }
        else socket.emit('message', "Я не розумію вас, спробуйте перефразувати своє питання");
    }catch(error){
    socket.emit('message', 'Виникла помилка при обробці повідомлення');
    }
});
```

Для навчання використовується наступний підхід:

```
socket.on('study', (data)=>{
```

```
manager.addDocument('uk', data.question, data.teg);
manager.addAnswer('uk', data.teg, data.answer);
db.run('INSERT INTO chatbot_ai (data, teg, answer) VALUES(?, ?, ?)', [data.question, data.teg, data.answer], function(err){
if(err){
return console.error(err.message);
}
console.log(`Значенняз ID$ {this.lastID} було успішно додано до таблиці.`);
});
```

Запропонована реалізація навчання та опрацювання запитів є досить простою. Кожен новий запит служить для тренування моделі, і після цього отримується результат відповідно до запиту користувача. Тренування в даному випадку полягає в додаванні користувачем інформації як у мовну модель, так і у базу даних.

3.4. Додавання функціоналу, для тренування моделі, через стороні готові моделі

Однією з важливих особливостей розробки ефективних чат-ботів є можливість інтеграції з готовими моделями для покращення навчання і продуктивності. Використання сторонніх готових моделей, таких як GPT-3.5, дозволяє значно підвищити точність і ефективність чат-бота завдяки доступу до великої кількості вже оброблених даних та передових алгоритмів.

Опис процесу тренування локальної моделі за допомогою готової моделі

Основна ідея полягає в тому, щоб використовувати відповіді, отримані від сторонньої моделі, для покращення локальної моделі. Нижче наведено опис цього процесу.

Ініціалізація локальної моделі: На початку, ми ініціалізуємо локальну модель та завантажуємо дані з бази даних. Це дозволяє моделі мати базовий набір даних для початкового тренування.

```
const manager = new NlpManager({ languages: ['uk'] });
```

```
const query = 'SELECT id, data, answer FROM chatbot_ai';


db.all(query, [], (err, rows) => {
  rows.forEach(row => {
    manager.addDocument('uk', row.data, row.data); // Використовуємо запит як тег
    manager.addAnswer('uk', row.data, row.answer);
  });
  manager.train().then(async () => {
    await manager.save();
    console.log('Модель успішно тренована та збережена');
  });
});
```

Обробка запитів від користувачів: Коли користувач надсилає запит, локальна модель намагається знайти відповідь. Якщо відповідь знайдена, вона надсилається користувачеві.

```
let response = await manager.process('uk', data);
if (response.answer) {
  socket.emit('message', response.answer);
} else {
  // Звернення до сторонньої моделі
}
```

Звернення до готової моделі: Якщо локальна модель не може знайти відповідь, запит надсилається до сторонньої готової моделі (GPT-3.5). Відповідь від сторонньої моделі потім використовується для навчання локальної моделі.

```
const gptResponse = await getChatGptResponse(data, userId);
if (gptResponse) {
  socket.emit('message', gptResponse);
  // Навчаємо локальну модель на основі відповіді GPT
}
```

Навчання локальної моделі: Відповідь від сторонньої моделі додається до локальної моделі як новий документ та відповідь. Потім модель тренується і зберігається для подальшого використання.

```
manager.addDocument('uk', data, data); // Використовуємо запит як тег
manager.addAnswer('uk', data, gptResponse);
await manager.train();
await manager.save();
console.log('Локальна модель успішно оновлена на основі відповіді GPT');
```

Оновлення на основі зворотного зв'язку: Локальна модель також може бути оновлена на основі зворотного зв'язку від користувачів. Користувачі можуть повідомляти про неправильні відповіді, і локальна модель буде оновлена відповідно до правильної інформації.

```
socket.on('incorrect_response', async (data) => {
manager.removeAnswer('uk', data.question, data.incorrect_answer);
manager.addAnswer('uk', data.question, data.correct_answer);
await manager.train();
await manager.save();
console.log('Модель успішно оновлена на основі зворотного зв\'язку');
});
```

Інтеграція з готовими моделями дозволяє значно підвищити ефективність роботи чат-ботів, забезпечуючи точні та релевантні відповіді на специфічні запити користувачів. Використання великого обсягу даних та передових алгоритмів, наданих сторонніми моделями, дозволяє швидко навчати локальну модель та покращувати її продуктивність.

3.4. Розробка графічної частини

Для розробки графічної частини використовується AngularJS, який дозволяє зручно та швидко організувати візуалізацію нашого чату. Використання

налаштованого набору стилів та готового застосунку дозволяє зручно інтегрувати код.

Спочатку необхідно створити верстку, основними компонентами якої будуть:

- Поле для відображення тексту від користувача та відповіді від серверу.
- Поле для набору тексту.
- Механізм для відправлення набраного тексту.

Ці компоненти стануть основою для зручної та ефективної інтеракції з чат-ботом, а AngularJS спростить їхню інтеграцію та візуалізацію.

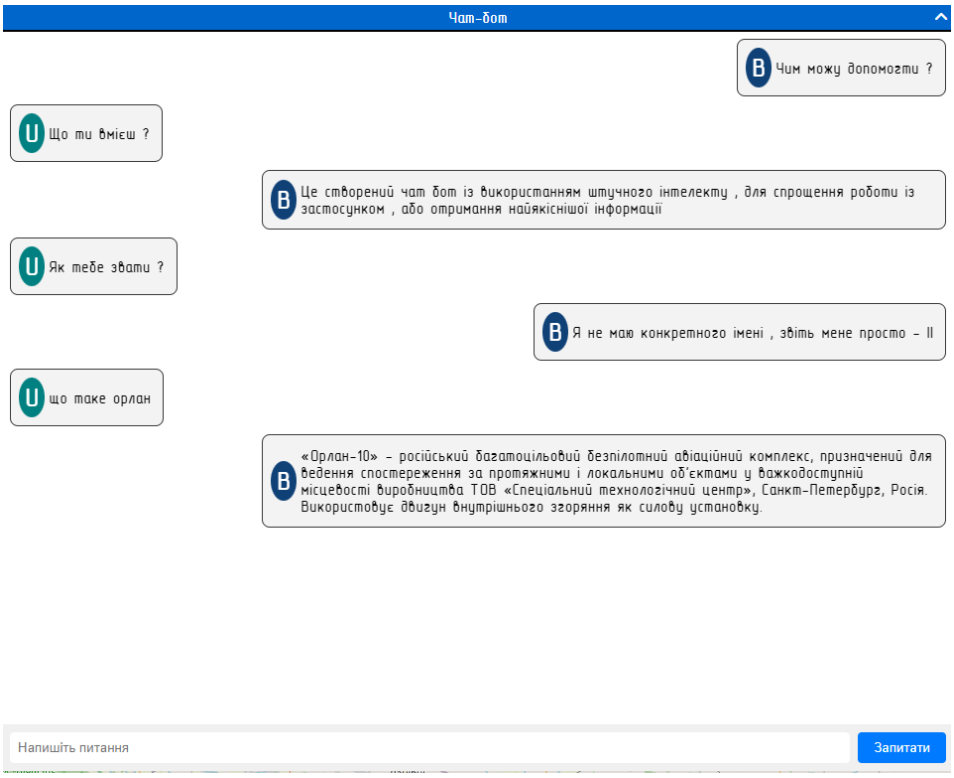


Рисунок 1.3.Приклад оформлення чат-бота.

Після реалізації верстки необхідно організувати відправлення, та прийом інформації від серверу, а також формування запиту на додавання нової інформації від користувача, весь діалог користувача і серверу зберігається на боці застосунку в об'єкті botmessages , налаштування чату в botchat :

that.chatbot=

{

```
open:function(){
        if(!socketbot||!socketbot.connected){
                socketbot=io;
                socketbot.on('message', (data) => { that.model.botmessages.push ( {
        username: 'B', message: data});});
        }
},
send:function(data){
        if(that.model.botchat.stady==0)
        {
                if(that.model.botchat.message==='Будемонавчатись'){
                that.model.botmessages.push({username:'U',message:that.model.botchat.
        message});
                that.model.botmessages.push({username:'B',message:"Подайте
        питання"});
                that.model.botchat.stady++;
        }
else{
        if(that.model.botchat.message.length>0){
                that.model.botmessages.push({username:'U',message:that.model.botchat.
        message});
        }
        if(socketbot&&socketbot.connected){
                socketbot.emit('message',that.model.botchat.message);
        }
}
}else{
        if(that.model.botchat.stady==1){
        that.model.botmessages.push({username:'U',message:that.model.botchat.messa
ge});
```

```
        that.model.botmessages.push({username:'B',message:"Подайте відповідь"});
        that.model.botchat.stady++;
        that.model.stadybotquest.question=that.model.botchat.message;
}
else if(that.model.botchat.stady==2){
        that.model.botmessages.push({username:'U',message:that.model.botchat.messa
ge});
        that.model.botmessages.push({username:'B',message:"Подайте ключо
веслово"});
        that.model.botchat.stady++;
        that.model.stadybotquest.answer=that.model.botchat.message;
}
else if(that.model.botchat.stady==3){
        that.model.botmessages.push({username:'U',message:that.model.botchat.messa
ge});
        that.model.botmessages.push({username:'B',message:"Взято до
опрацювання"});
        that.model.botchat.stady=0;
        that.model.stadybotquest.teg=that.model.botchat.message;
        socketbot.emit('study',that.model.stadybotquest);
}
}
that.model.botchat.message="";
}
};
```

Встановлюється підключення через сокети та прослуховування відповіді від сервера. У разі отримання відповіді, додається текст до об'єкту та візуально виводиться на сторінці. Реалізація машиного навчання у відправці повідомленя заключається в перевірці наміру користувача за ключовою фразою "Будемо

навчатись". Якщо такої фрази не виявлено, повідомлення буде направлено для обробки на сервер.

Якщо отримано фразу для навчання, проходять три етапи: питання, де користувач має представити очікуване питання, відповідь – очікувана відповідь від сервера на питання, ключове слово – інтент. У даному випадку дозволяється користувачу вибрати його самостійно, що є коректним з точки зору соціальної інженерії.

3.5. Використання зворотного зв'язку для покращення моделі

Зворотний зв'язок від користувачів є критично важливим для покращення роботи чат-бота. Збирання та аналіз зворотного зв'язку дозволяє виявити слабкі місця у роботі бота, а також підвищити точність та релевантність відповідей. Зворотний зв'язок може бути зібраний різними способами, включаючи оцінювання відповідей, коментарі та автоматичний аналіз історії розмов.

Для забезпечення ефективного збору зворотного зв'язку в чаті використовуються ключові слова. Коли користувач отримує некоректну відповідь, він може використати фразу "Спробуй ще" для повторного запиту або "Невірно" для надання коректної інформації. Ці відгуки автоматично зберігаються в базі даних разом із запитанням, некоректною відповіддю та, якщо є, правильною відповіддю, наданою користувачем.

Якщо користувач не надає правильної відповіді, система звертається до зовнішнього джерела, такого як ChatGPT, для отримання правильної відповіді. Отримана відповідь також зберігається в базі даних.

Зворотний зв'язок аналізується та обробляється для визначення, що саме потребує корекції. Невірна відповідь видаляється з моделі чат-бота, а правильна відповідь додається. Після цього модель повторно тренується на нових даних для покращення точності відповідей у майбутньому.

Після оновлення моделі користувач отримує повідомлення про те, що його зворотний зв'язок був врахований, і правильна відповідь була додана до бази знань. Це дозволяє постійно покращувати якість роботи чат-бота, роблячи його більш точним і корисним для користувачів. Зворотний зв'язок від користувачів допомагає швидко ідентифікувати та виправляти помилки, що підвищує загальну ефективність системи.

ВИСНОВКИ

У цій кваліфікаційній роботі було досліджено та розроблено чат-бот з використанням технологій машинного навчання та обробки природної мови (NLP). Основні результати дослідження дозволяють зробити наступні висновки. Було створено чат-бот, який успішно інтегрує технології машинного навчання та NLP для забезпечення високої точності та релевантності відповідей. Використання наглядового, безнаглядового та напівнаглядового навчання дозволило оптимізувати процес навчання моделі, що забезпечило її адаптивність та ефективність.

Впровадження сучасних технологій машинного навчання та NLP значно підвищило продуктивність та ефективність роботи чат-бота. Асинхронна архітектура Node.js та використання AngularJS для розробки інтерфейсу дозволили створити інтерактивний та зручний для користувачів додаток. Це сприяє швидшій обробці запитів та забезпечує миттєву взаємодію з користувачами.

Система зворотного зв'язку дозволила постійно вдосконалювати модель чат-бота на основі відгуків користувачів. Це забезпечує актуальність відповідей та покращує користувацький досвід. Можливість інтеграції нових даних та оновлення моделі підвищує адаптивність чат-бота до змінних умов і потреб користувачів. Розроблений чат-бот має велике практичне значення, оскільки може бути використаний у різних галузях, включаючи підтримку клієнтів, освітні програми, охорону здоров'я та військові проекти. Його впровадження дозволяє знизити навантаження на людські ресурси, підвищити ефективність обслуговування та забезпечити доступність інформації в режимі реального часу.

Подальше вдосконалення алгоритмів машинного навчання дозволить підвищити точність і швидкість роботи чат-ботів, а також забезпечити їхню адаптивність до нових умов. Розширення функціональності чат-бота,

включаючи можливість інтеграції з іншими системами, такими як CRM та ERP, а також розвиток можливостей для більш складних інтерактивних сценаріїв, сприятиме покращенню якості обслуговування.

Розробка та впровадження багатомовної підтримки дозволить розширити аудиторію користувачів та забезпечити високий рівень обслуговування у глобальному масштабі. Інтеграція аналізу емоційного забарвлення повідомлень користувачів може підвищити якість взаємодії та зробити чат-бот більш чуйним та людяним. Проведення польових випробувань та впровадження розробленого чат-бота у реальних умовах дозволить оцінити його ефективність та виявити можливі проблеми у практичному застосуванні. Це забезпечить отримання додаткових даних для подальшого вдосконалення моделі.

Таким чином, результати дослідження показують високу ефективність розробленого чат-бота та підтверджують перспективність використання технологій машинного навчання та NLP у різних галузях. Подальші дослідження та впровадження нових технологій дозволять ще більше покращити якість та функціональність таких систем.

.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ляшенко А.І. Інформаційні технології та системи штучного інтелекту: навчальний посібник. Київ: Центр учбової літератури, 2017. – 320 с.
2. Сисоєв О.О. Машинне навчання та обробка даних: підручник. Харків: ХНУРЕ, 2019. – 280 с.
3. Ігнатов О.І., Губенко І.М., Антонюк І.М. Системи штучного інтелекту: навчальний посібник. Львів: Видавництво Львівської політехніки, 2018. – 240 с.
4. Ковальчук В.М. Технології обробки природної мови: навчальний посібник. Київ: НТУУ "КПІ", 2020. – 300 с.
5. Литвин В.В. Інтелектуальні системи: теорія та практика: монографія. Тернопіль: ТНЕУ, 2021. – 350 с.
6. Лапчук О.Ю., Павленко М.М. Основи обробки природної мови: навчальний посібник. Київ: КНУ, 2021. – 270 с.
7. Малєєв Ю.М. Інтелектуальні системи: алгоритми та методи: навчальний посібник. Львів: Видавництво Львівської політехніки, 2020. – 320 с.
8. A brief history of Chatbots. [Електронний ресурс] https://chatbotslife.com/a-brief-history-of-chatbots-d5a8689cf52f
9. Telegram API Documentation. [Електронний ресурс] https://core.telegram.org/
10. Bird S., Klein E., Loper E. Natural Language Processing with Python. Sebastopol: O’Reilly, 2009. 504 p.
11. Jurafsky D., Martin J. H. Speech and Language Processing. New Jersey: Prentice Hall, 2023. 628 p. [Електронний ресурс] https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf
12. ChatGPT by OpenAI. [Електронний ресурс] https://chat.openai.com/
13. Raschka S. Python Machine Learning. 2nd Edition. Packt Publishing, 2017.
14. Lane H., Howard C., Hapke H. Natural Language Processing in Action. Manning Publications, 2019.
15. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2nd Edition. O'Reilly Media, 2019.

16. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. 4th Edition. Pearson, 2020.
17. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016.
18. Raj S. Building Chatbots with Python: Using Natural Language Processing and Machine Learning. Apress, 2019.
19. Nichol A. Chatbots: An Introduction. O'Reilly Media, 2017.
20. Casciaro M. Node.js Design Patterns. 3rd Edition. Packt Publishing, 2020.