

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
(повне найменування вищого навчального закладу)

Факультет Інфокомунікацій
(повне найменування інституту, факультету (відділення))

Кафедра Інформаційно-мережної інженерії
(повна назва кафедри (предметної, циклової комісії))

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)

(рівень вищої освіти)

Amazon Lex – створення чат-бота, обробка природної мови

(тема)

Виконав: студент 2 курсу групи ІМІм-20-1

Спеціальність 172 Телекомунікації та

радіотехніка

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційно-мережна

інженерія

(повна назва освітньої програми)

Рижов О.О.

(прізвище та ініціали)

Керівник доц. Кривенко С.А.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Безрук В.М.

(прізвище, ініціали)

2021 р.

Не містить відомостей, заборонених
до відкритого публікування

Керівник _____ /*С.А. Кривенко*

Студент _____ / *О.О. Рижов*

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

(повне найменування вищого навчального закладу)

Факультет Інфокомунікацій

Кафедра Інформаційно-мережної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 172 Телекомунікації та радіотехніка

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційно-мережна інженерія

(повна назва)

ЗАТВЕРДЖУЮ

Зав. кафедри «ІМІ»

_____ проф. Безрук В.М.

(підпис)

« 08 » _____ листопада 2021 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Рижову Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Amazon Lex – створення чат-бота, обробка природної мови

керівник роботи доц., к.т.н. Кривенко Станіслав Анатолійович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ВНЗ від « 08 » _____ листопада 2021 р. № 1674 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13.12. 2021 р.

3. Вихідні дані до роботи Провести аналіз доступних сервісів хмарних обчислень для застосування в технологіях машинного навчання. Основні послуги, що надаються хмарними системами. Мови реалізації моделі – Python3, HTML.

4. Перелік запитань, що необхідно опрацювати в роботі _____

Вступ

1 Огляд обробки природної мови

2 Аналіз сервісів для обробки природної мови

3 Модель створення середовища для роботи чат-бота з обробкою

природної мови

4 Тестування моделі

Висновки

5. Перелік графічного матеріалу із зазначенням креслень, схем, плакатів, комп'ютерних ілюстрацій слайди презентації в форматі Power Point (назва роботи, мета роботи, зміст, Що таке обробка природньої мови, Amazon Transcribe, Amazon Polly, Amazon Translate, Amazon Comprehend, Amazon Lex, Модель середовища для чат-бота, Інтерфейс взаємодії з ботом висновки)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Ознайомлення із завданням. Уточнення ТЗ.	08.11.21	
2	Підбір літератури за темою роботи.	09.11.21- 11.11.21	
3	Виконання розділу 1	12.11.21- 17.11.21	
4	Виконання розділу 2	18.11.21- 22.11.21	
5	Виконання розділу 3	23.11.21- 28.11.21	
6	Виконання розділу 4	29.11.21- 5.12.21	
7	Висновки	05.12.21- 09.12.21	
8	Оформлення презентаційного матеріалу, підготовка до захисту у ЕК	10.12.21- 13.12.21	

Дата видачі завдання 08 листопада 2021 р.

Студент _____ Рижов О.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ доц. Кривенко С.А.
(підпис) (посада, прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка випускної магістерської кваліфікаційної роботи містить: 96 стор., 64 рис., 25 джерел, 3 додатки.

AMAZON LEX, ОБРОБКА ПРИРОДНОЇ МОВИ, МАШИННЕ НАВЧАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ, AWS.

Об'єкт дослідження – Amazon Lex.

Предмет дослідження – можливості застосування Amazon Lex, взаємодія людини з чат-ботами.

Мета роботи – розробка моделі для створення чат-бота на основі сервісу Amazon Lex, з яким користувачі можуть взаємодіяти, щоб записатися на прийом до стоматолога та дослідження його можливостей.

Завдання роботи – вивчення та аналіз: проблем, випадків використання та сервісів Amazon для обробки природної мови, моделей створення маркерів та розробки функцій; визначення й розглядання: обробки природної мови, попередньої обробки тексту, Amazon Transcribe, Amazon Polly, Amazon Translate, Amazon Comprehend, Amazon Lex, наміру; створення: середовища для роботи чат-бота з обробкою природної мови, чат-бота на основі сервісу Amazon Lex, функції AWS Lambda, сегмента Amazon S3.

ABSTRACT

The explanatory note of the final master's qualification work contains: 96 pages, 64 figures, 25 sources, 3 app.

AMAZON LEX, NATURAL LANGUAGE PROCESSING, MACHINE LEARNING, ARTIFICIAL INTELLIGENCE, AWS.

The object of research is Amazon Lex.

The subject of research – the possibility of using the Amazon Lex, human interaction with chatbots.

The purpose of the work is developing a model to create a chatbot based on the Amazon Lex service, with which users can interact to make an appointment with a dentist and explore its capabilities.

The task of the work is to study and analyze: problems, cases of use and services of Amazon for natural language processing, models for creating markers and developing functions; identified and considered: natural language processing, word processing, Amazon Transcribe, Amazon Polly, Amazon Translate, Amazon Comprehend, Amazon Lex, intention; creation: environments for chatbot with natural language processing, chatbot based on Amazon Lex service, AWS Lambda functions, Amazon S3 segment.

ЗМІСТ

Перелік скорочень, умовних позначень, символів, одиниць і термінів	8
Вступ	9
1 Огляд обробки природної мови	14
1.1 Що таке Natural Language Processing	14
1.2 Проблеми, з якими стикається NLP	15
1.3 Випадки використання NLP	17
1.4 Потік обробки природної мови	17
1.4.1 Попередня обробка тексту	18
1.5 Моделі створення маркерів та розробки функцій	19
1.6 Висновки до першого розділу	23
2 Аналіз сервісів для обробки природної мови	25
2.1 Сервіс Amazon Transcribe	25
2.2 Сервіс Amazon Polly	26
2.3 Сервіс Amazon Translate	28
2.4 Сервіс Amazon Comprehend	30
2.5 Сервіс Amazon Lex	31
2.6 Висновки до другого розділу	33
3 Модель створення середовища для роботи чат-бота з обробкою природної мови	34
3.1 Створення бота за допомогою Amazon Lex	35
3.2 Тестування створеного бота	38
3.3 Створення функції AWS Lambda	39

3.4	Перевірка функції Lambda	42
3.5	Оновлення намірів бота.....	43
3.6	Висновки до третього розділу.....	45
4	Тестування моделі	46
4.1	Побудова та тестування нашого бота	46
4.2	Налаштування пулу ідентифікаційних даних Amazon Cognito.....	48
4.3	Змінення ролей IAM, для доступу до Amazon Lex.....	51
4.4	Створення сегмента Amazon S3.....	53
4.5	Оновлення та тестування демонстраційного файлу	55
4.6	Висновки до четвертого розділу.....	60
	Висновки	62
	Список літератури.....	65
	Додаток А Слайди презентації	68
	Додаток Б Код	75
	Додаток В Публікації за темою роботи	92

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

AGI – Artificial General Intelligence – сильний штучний інтелект

AI – Artificial Intelligence – штучний інтелект

Amazon S3 – Amazon Simple Storage Service

API – Application Programming Interface – інтерфейс прикладного програмування

ARN – Amazon Resource Name – назва ресурсу Amazon

AWS – Amazon Web Services – дочірня компанія Amazon.com, що надає платформу хмарних обчислень в оренду приватним особам, компаніям та урядам на основі платної підписки

HIPAA – Health Insurance Portability and Accountability Act – Закон США про мобільність та підзвітність медичного страхування

HTML – HyperText Markup Language – мова розмітки гіпертексту

IAM – Identity and Access Management – управління ідентифікацією та доступом

ML – Machine Learning – машинне навчання

NER – Named Entity Recognition – розпізнавання іменованих сутностей

NLP – Natural Language Processing – обробка природної мови

NLTK – Natural Language Toolkit – Набір інструментів природної мови

PCI DSS – Payment Card Industry Data Security Standard – Стандарту безпеки даних індустрії платіжних карток

SSML – Speech Synthesis Markup Language – мова розмітки синтезу мовлення

TF-IDF – Term Frequency and Inverse Document Frequency – частота термінів та інверсна частота документа

URL – Uniform Resource Locator – єдиний вказівник на ресурс

ВСТУП

На сьогоднішній день обробка природної мови (NLP) є основною темою, яка досліджується та розробляється останнє десятиліття. Здатність аналізувати, обчислювати та розуміти просту людську мову за допомогою комп'ютерів і машинного навчання була впроваджена в різні програми та платформи. У цій роботі розглядається, як чат-боти в основному використовують такі параметри, як сутності та наміри, щоб узгоджувати синтаксис і виконувати обробку природною мовою. Поява віртуальних помічників у споживчих технологіях змусила великі компанії працювати на власних хмарних платформах.

Завдяки технології штучного інтелекту, що розвивається, останнім часом чат-боти стають розумнішими та швидшими. Чат-боти, як правило, доступні цілодобово, забезпечуючи постійну підтримку та послуги. Проблема розробки інтелектуального чат-бота все ще існує з моменту появи штучного інтелекту. Функціональність чат-ботів може варіюватися від коротких розмов, орієнтованих на бізнес, до триваліших розмов на основі медичних заходів. Однак головна роль, яку повинні відігравати чат-боти, полягає в розумінні людських висловлювань, щоб правильно реагувати.

Метою кваліфікаційної роботи є розробка моделі для створення чат-бота на основі сервісу Amazon Lex, з яким користувачі можуть взаємодіяти, щоб записатися на прийом до стоматолога та дослідження його можливостей.

Завдання роботи полягає у вивченні та аналізі: проблем, випадків використання та сервісів Amazon для обробки природної мови, моделей створення маркерів та розробки функцій; визначені й розгляданні: обробки природної мови, попередньої обробки тексту, Amazon Transcribe, Amazon Polly, Amazon Translate, Amazon Comprehend, Amazon Lex, наміру; створенні: середовища для роботи чат-бота з обробкою природної мови, чат-бота на основі сервісу Amazon Lex, функції AWS Lambda, сегмента Amazon S3.

Кваліфікаційна робота складається з чотирьох розділів, 64 рисунків, 3 додатків та 25 джерел літератури.

У першому розділі розглядається обробка природної мови, її проблеми та випадки використання. Обробка природної мови також відома як NLP (Natural Language Processing). NLP розробляє обчислювальні алгоритми для автоматичного аналізу та представлення людської мови. Оцінюючи структуру мови, системи машинного навчання ML можуть обробляти великі набори слів, фраз і речень. Машинне навчання ML – це підмножина більш широкої галузі інформатики, яка відома як штучний інтелект AI. AI – це створення будівельних машин, які можуть виконувати завдання, які зазвичай виконує людина. У сучасній культурі штучний інтелект з'являється у фільмах або художніх творах. Ви можете згадати деякі приклади штучного інтелекту у науково-фантастичних фільмах або телешоу, які контролюють майбутній світ або діють розумно самостійно – іноді з негативними наслідками для суспільства або оточуючих людей. Ці застосування AI починалися як комп'ютерні агенти, які сприймали їхнє середовище та вживали заходів для досягнення певної мети. Однак, для деяких із цих вигаданих прикладів штучного інтелекту їх дії не були результатом, якого спочатку передбачали їх творці. Інші вигадані приклади штучного інтелекту більш доброякісні або позитивні: вони краще працюють з людством, але вони також мають більш загальний характер. Такі види загального штучного інтелекту є прикладами сильного штучного загального інтелекту AGI. Вони мають здатність вивчати або розуміти будь-яке завдання, яке може зрозуміти людина.

Проблеми штучного інтелекту зазвичай охоплюють багато сфер дослідження: обробка природної мови, міркування, представлення знань, навчання, сприйняття та взаємодія з фізичним середовищем. AI ще не є реальністю, якщо ви не живете в симуляції. Однак з кожним роком це стає ближчим у кожній із цих областей.

Можливо, ви також читали чи бачили коментарі до етики створення AI. Не всі погляди позитивні – можливо частково через страх перед вигаданими

штучними інтелектами, які знищили людей або використовували їх як джерела енергії. Деякі також можуть розглянути ризик масового безробіття, оскільки розумна машина може працювати безперервно.

Обробка природної мови NLP – це ще одна сфера машинного навчання із збільшенням її використання. Наприклад, Amazon Alexa (або будь-який інший голосовий помічник) використовує NLP, щоб спробувати відповісти на ваші запитання. NLP – це мова, а також письмовий текст.

NLP використовується в багатьох додатках, таких як:

Боти чату або колл-центру – автоматизовані системи для отримання балансу в банку або замовлення їжі в ресторані.

Інструменти перекладу – перетворення тексту між мовами або програми, які можуть перекладати меню в режимі реального часу.

Переклад голосу в текст – перетворення вимовлених слів у текст. Може використовуватися для включення автоматичних субтитрів.

Аналіз настроїв – дозволяє аналізувати почуття коментарів у оглядах продуктів, музики та фільмів. Ці почуття можна використати, щоб оцінити аудиторію фільму.

NLP – це широкий термін для загального набору ділових або обчислювальних проблем, які можна вирішити за допомогою машинного навчання.

Після аналізу NLP в розділі розглядається попередня обробка тексту.

У другому розділі аналізуються досягнення компанії Amazon, розглядаються основні сервіси, які використовуються для обробки природної мови. Amazon Transcribe може автоматично перетворювати розмовну мову на текст, його використовують для медичної транскрипції, субтитрів для відео, маркування потокового вмісту та для моніторингу колл-центрів. Amazon Polly може перетворювати письмовий текст у розмовну мову, наприклад, для виробництва служб новин, систем навчання мові, систем навігації та виробництва анімацій. Amazon Translate може створювати переклади між мовами в режимі реального часу, який можна використовувати для створення

міжнародних веб-сайтів, локалізації програмного забезпечення, розробки багатомовних чат-ботів та міжнародного управління медіа. Amazon Comprehend автоматизує багато випадків використання NLP, які розглядаються в цій роботі, наприклад, для аналізу юридичних та медичних документів, виявлення фінансового шахрайства, масштабного аналізу мобільних додатків або позначення вмісту для аналізу та керування. І, нарешті, Amazon Lex може створити інтерфейс, схожий на людину, який може взаємодіяти за допомогою голосу та тексту, щоб задавати запитання, отримувати відповіді чи виконувати завдання, наприклад, створення інтерфейсів для управління запасами та програм продажів, інтерактивної допомоги, обслуговування клієнтів або запитів до баз даних людською мовою.

У третьому розділі розглядається синтез моделі середовища для роботи чат-бота з обробкою природньої мови на основі Amazon Lex. Описано виконання наступних кроків високого рівня: створення і тестування бота за допомогою плану зустрічі Amazon Lex ScheduleAppointment; створення і тестування функцію AWS Lambda для виконання завдань ініціації, перевірки та виконання; налаштування наміру MakeAppointment для використання функції Lambda як гачка коду. Кожен бот, який створюється в Amazon Lex, має намір. Намір – це дія, яку виконує бот.

Четвертий розділ присвячений тестуванню моделі для чат-бота, в тому числі, за допомогою веб-сторінки розміщеної на Amazon S3. Розглядається збирання Amazon Lex та Lambda в одне середовище, перевірення бота, для доведення, що він виконує лямбда-функцію. Далі, тестується публікація бота, щоб можна було перевірити його виклик з веб-сторінки. Найпростіший спосіб провести тест – створити статичну веб-сторінку та розмістити її на Amazon S3. Перевіряється чи викликається API Amazon Lex з веб-сторінки для завантаження нашого бота, завдяки чому відвідувачі веб-сторінки можуть потім взаємодіяти з нею. Тестується налаштування пулу ідентифікаційних даних Amazon Cognito та створення відповідних ролей (одна роль закінчується на Auth_Role, а інша на Unauth_Role), для забезпечення безпеки на цій сторінці.

Веб-сторінці, на якій розміщено нашого бота, потрібен дозвіл на доступ до Amazon Lex. Для цього двом ролям, створеним у пулі ідентифікаційних даних Amazon Cognito, потрібні дозволи на доступ до Amazon Lex. Перевіряється налаштування AWS Identity and Access Management (IAM), щоб надати ці дозволи ролям пулу ідентифікаційних даних. Після налаштування дозволів безпеки і створення сегмента S3 для розміщення нашої веб-сторінки, тестується оновлений демонстраційний файл HTML, і перевіряється щоб він використовував пул ідентифікаційних даних Amazon Cognito. Фінальне тестування перевіряє взаємодію зі створеним в магістерській роботі ботом. На сторінці з розділом Статичний хостинг веб-сайтів за вибраним URL-адресом відкривається створена веб-сторінка.

1 ОГЛЯД ОБРОБКИ ПРИРОДНОЇ МОВИ

Обробка природної мови також відома як NLP (Natural Language Processing). Перш ніж розповісти про NLP, ми розглянемо приклад, як NLP використовується в Amazon Alexa (Рис. 1.1).



Рисунок 1.1 – Використання NLP в Alexa

Amazon Alexa, Alexa працює завдяки тому, що пристрій, наприклад Amazon Echo, записує наше слово. Запис нашого мовлення надсилається на сервери Amazon для ефективнішого аналізу. Amazon розбиває нашу фразу на окремі звуки. Потім він підключається до бази даних, що містить вимову різних слів, щоб знайти слова, які найбільше відповідають поєднанню окремих звуків. Amazon визначає важливі слова, щоб зрозуміти завдання та виконувати відповідні функції. Наприклад, якщо Alexa помітить такі слова, як надворі або температура, це відкриє навик Погода Alexa. Потім сервери Amazon надсилають інформацію назад на наш пристрій, і Alexa говорить [1,2].

1.1 Що таке Natural Language Processing

NLP розробляє обчислювальні алгоритми для автоматичного аналізу та представлення людської мови. Оцінюючи структуру мови, системи машинного навчання ML можуть обробляти великі набори слів, фраз і речень. Машинне навчання ML – це підмножина більш широкої галузі інформатики, яка відома як

штучний інтелект AI. AI – це створення будівельних машин, які можуть виконувати завдання, які зазвичай виконує людина. NLP – це широкий термін для загального набору ділових або обчислювальних проблем, які можна вирішити за допомогою машинного навчання [3,4,5].

Однак системи NLP передували машинному навчанню, наприклад, для перетворення мовлення в текст на старих мобільних телефонах, які були перед смартфонами, використовували NLP, а також програми зчитування з екрана. Зараз багато систем NLP використовують певну форму машинного навчання.

NLP представляє ієрархічну структуру мови (Рис. 1.2).

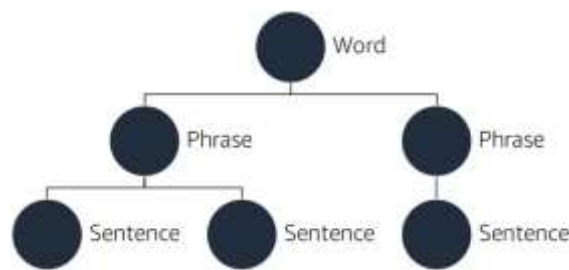


Рисунок 1.2 – Структура мови NLP

Слова знаходяться на найнижчому рівні в ієрархії. Група слів, складає словосполучення. На наступному рівні фрази утворюють речення, а речення передають ідеї.

Системи NLP стискаються з кількома значними проблемами. Далі ми розглянемо ці проблеми.

1.2 Проблеми, з якими стикається NLP

Перш за все, мова не точна. Слова можуть мати різні значення залежно від інших слів, які їх оточують. Це відомо як контекст. Часто одні й ті ж слова чи фрази можуть мати кілька значень. Наприклад, розглянемо термін погода (weather), ми можемо бути «under the weather», який в англійській мові має розмовне значення, що ми хворіємо. Однак «there is wonderful weather outside», а це означає, що на вулиці хороші погодні умови. Фраза «Oh, really?» може

передавати здивування, незгоду чи багато інших значень, залежно від контексту та звороту.

На рис. 1.3 наведено деякі з основних проблем для NLP.

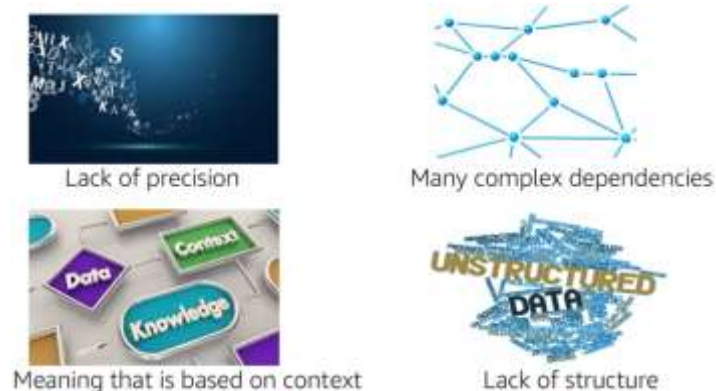


Рисунок 1.3 – Основні проблеми NLP

Однією з проблем є виявлення структури тексту. Одне з перших завдань будь-якої програми NLP – розбити текст на змістовні одиниці, такі як слова, фрази та речення [5].

Ще одна проблема – маркування даних. Після того, як система перетворить текст у дані, вона повинна застосувати мітки, які представляють різні частини мови. Кожна мова потребує різної схеми маркування, щоб відповідати граматиці мови.

NLP також стикається з проблемою представлення контексту. Оскільки значення слова може сильно залежати від контексту, будь-якій системі NLP потрібен спосіб його представлення. Це велика проблема, оскільки існує багато контекстів, і важко перетворити контекст у форму, яку можуть зрозуміти комп'ютери [3,4,8].

Не менш важливою проблемою є застосування граматики. Хоча граматика визначає структуру мови, застосування граматики є невимовно великим за обсягом. Боротьба з різницею в тому, як люди використовують мову, є серйозною проблемою для систем NLP. Саме тут машинне навчання може мати великий вплив.

1.3 Випадки використання NLP

NLP можна застосовувати до широкого кола проблем.

На рис. 1.4 зображено випадки використання NLP.



Рисунок 1.4 – Випадки використання NLP

Деякі з найбільш поширених додатків включають:

- Пошукові програми, такі як Google або Bing;
- Інтерфейси людини-машини, наприклад Alexa;
- Аналіз настроїв для маркетингових або політичних кампаній;
- Соціальні дослідження, засновані на аналізі ЗМІ;
- Чат-боти для імітації людської мови в програмах.

1.4 Потік обробки природної мови

Ми можемо застосувати конвеєр розробки машинного навчання, який зображено на рис. 1.5.

Перше завдання – сформулювати проблему, потім зібрати та позначити дані. Для NLP збір даних складається з розбиття текстів на значущі підмножини та позначення множин. Функціональна інженерія є основною частиною програм NLP. Цей процес ускладнюється, коли ми маємо справу з дуже неправильним або неструктурованим текстом. Наприклад, ми створюємо

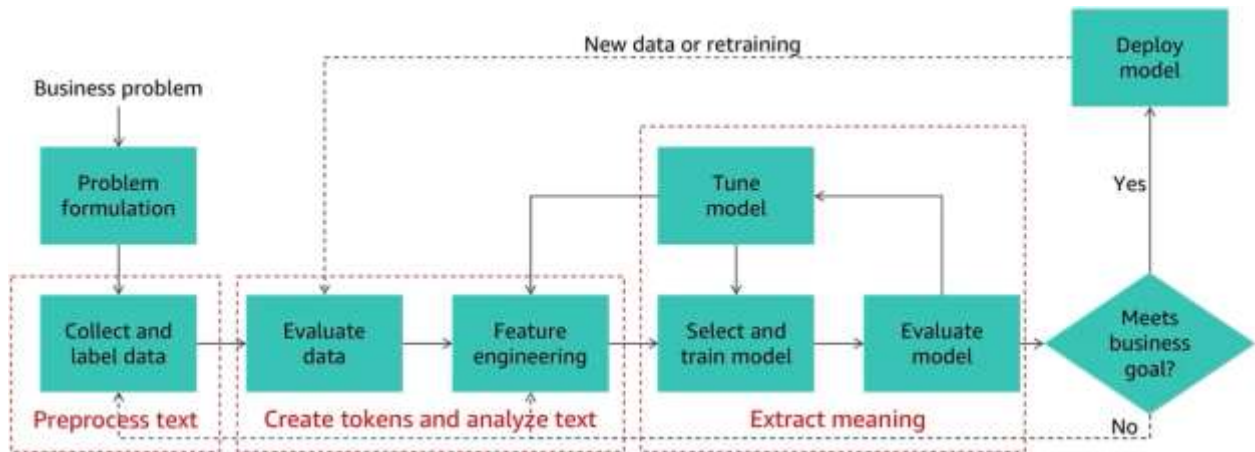


Рисунок 1.5 – Конвеєр розробки машинного навчання

програму для класифікації документів. Ми повинні вміти розрізняти слова зі спільними термінами, але різними значеннями. Маркування даних у домені NLP іноді також називають тегом. У процесі маркування ми призначаємо окремі текстові рядки різним частинам мови. Існують спеціалізовані інструменти, які можна використовувати для NLP-маркування [6].

1.4.1 Попередня обробка тексту

На рис 1.6 наведено приклад попередньої обробки тексту. Для попередньої обробки доступні кілька бібліотек та інструментів (наприклад, NLTK для Python) [7].

Перше завдання програми NLP – перетворити текст у дані, щоб його можна було проаналізувати. Ми перетворюємо текст, видаляючи з введеного тексту слова, які не потрібні для аналізу. У прикладі (рис. 1.6) слова *this* і *is* видаляються, щоб залишити зразок тексту фрази [4].

Після видалення стоп-слів ми можемо нормалізувати текст, перетворивши схожі слова в загальну форму. Наприклад, слова *run*, *runner*, *ran*, та *running* – це різні форми слова *run*. Ми можемо нормалізувати всі екземпляри цих слів у блоці тексту, використовуючи ці процеси стеммінгу та лематизації. Лематизація об'єднує різні форми слова в один термін. Лематизація версій

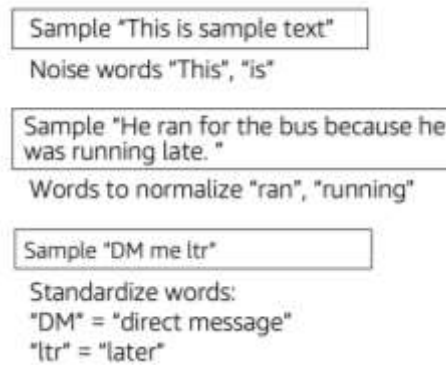


Рисунок 1.6 – Зразок попередньої обробки тексту

слова *run*, згрупувала б усі екземпляри цих форм в один термін *run*. З іншого боку, стеммінг видаляє символи, які алгоритм стеммінгу вважає непотрібними. Стеммінг може не працювати зі словом *run* як прикладом, оскільки форма *ran* може не розпізнаватися як форма слова *run* [9,10].

Після нормалізації тексту ми можемо стандартизувати його, видаливши слова, яких немає у словнику, який ми використовуємо для аналізу. Наприклад, ми можемо видалити акроніми, сленг і спеціальні символи.

Загальні етапи попередньої обробки:

- Усунення стоп-слів;
- Нормалізування схожого тексту;
- Стандартизування нерозпізаного тексту.

Інші етапи попередньої обробки

- Кодування;
- Перевірка орфографії та граматики.

Набір інструментів природної мови (Natural Language Toolkit), також відомий як NLTK. Їх бібліотека Python надає функції, які можна використовувати для видалення стоп-слів, нормалізації та стандартизації тексту [7,8].

1.5 Моделі створення маркерів та розробки функцій

Ще одним першим кроком у створенні системи NLP є перетворення

тексту в колекцію даних, наприклад у DataFrame. Усі бібліотеки NLP допомагають у цьому процесі. У прикладі, зображеного на рис. 1.7, показано використання функції токенизації слова з бібліотеки NLTK.

```
from nltk.tokenize import word_tokenize
text = "this is some sample text."
Print(word_tokenize(text))

Output: ['this', 'is', 'some', 'sample', 'text', '.']
```

Рисунок 1.7 – Зразок коду токена

Після того, як ми очистили текст і завантажили його у DataFrame, ми можемо застосувати одну з моделей NLP для створення функцій. Ось кілька поширених моделей.

Перша модель відома як мішок слів. Це проста модель для визначення частоти слів у документі. Модель створює ключ для кожного слова. Значення ключа – це кількість разів, коли це слово зустрічається в документі.

Друга модель – це частота термінів та інверсна частота документа, яка також відома як TF-IDF (Term Frequency and Inverse Document Frequency). Частота термінів – це кількість разів, коли слово зустрічається в документі. Обернена частота документа – це кількість разів, коли слово зустрічається в групі документів.

Ці два значення використовуються разом для обчислення ваги для слів. Слова, які часто зустрічаються в багатьох документах, мають меншу вагу. В області NLP було створено багато моделей. У прикладі, зображеного на рис. 1.8, показана модель, яку називають мішок слів.

Мішок слів, являє собою векторну модель. Векторні моделі перетворюють кожне речення або фразу у вектор, який є математичним об'єктом, який записує як спрямованість, так і величину.

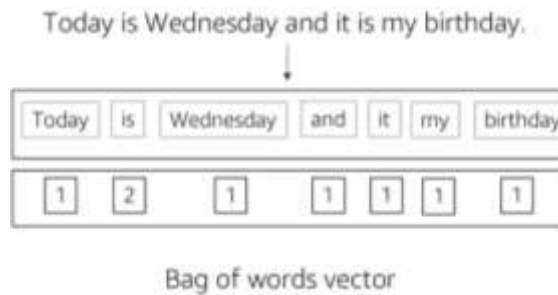


Рисунок 1.8 – Приклад моделі NLP: мішок слів

У прикладі на рис. 1.8 просте речення перетворюється на вектор, де записується частота кожного слова. Слово *is* має значення 2, оскільки воно зустрічається в реченні двічі.

Мішок слів часто використовується для класифікації документів за різними категоріями. Він також використовується для отримання атрибутів, які входять до програм NLP, наприклад, в аналізі настроїв [6,11].

Існує три широкі категорії аналізу тексту, зображених на рис.1.9.

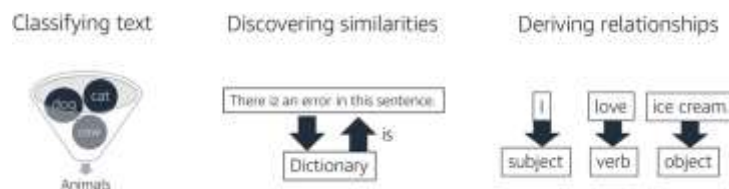


Рисунок 1.9 – Категорії аналізу тексту

По-перше, класифікація тексту – ця категорія аналізу подібна до інших систем класифікації, які ми розглядали раніше. Текст надає вхідні дані для процесу, який витягує функції. Потім ми надсилаємо функції за допомогою алгоритму машинного навчання, який взаємодіє з моделлю класифікатора щоб зробити висновок про класифікацію. Ми можемо використовувати бібліотеку NLTK Python для створення системи класифікації.

По-друге, виявлення подібності – зіставлення тексту має багато застосувань. Наприклад, автоматичне виправлення орфографії та перевірка граматики засновані на відповідності тексту. Часто використовується алгоритм редагування відстані (також відомий як відстань Левенштейна).

По-третє, виведення відносин – ми можемо вивести відносини між різними словами або фразами в тексті, використовуючи процес, який називається роздільною здатністю кореференцій. Кореферентність – це явище, яке позначає відношення між назвами (знаковими одиницями), компонентами висловлювання, при якому кілька компонентів відсилають до одного об'єкта позамовної дійсності (референта). Кілька систем NLP надають бібліотеки Python для виведення відносин [12,13,14].

Однією з найбільших проблем для NLP є те, як описати контекст для тексту. Розглянемо приклад такої проблеми, коли користувач шукає термін планшет, зображеного на рис 1.10.



Рисунок 1.10 – Проблема контексту на прикладі слова tablet

Оскільки слово планшет має принаймні два різних значення, пошукова система повинна знати, яке значення має на увазі користувач. Термін можна уточнити далі (наприклад, додавши інший термін, наприклад медицина або комп'ютери). Якщо це не кваліфікується іншими термінами, то більшість пошукових систем покладаються на найбільш часто використовуваний контекст.

Процес вилучення сутностей відомий як розпізнавання іменованих сутностей (Named Entity Recognition, NER). Модель NER має наступні функції:

По-перше, вона може ідентифікувати фрази-іменники за допомогою діаграм залежностей і тегів частини мови.

По-друге NER може класифікувати фрази, використовуючи алгоритм

класифікації, такий як Word2vec.

Також, ця модель може розрізняти сутності, використовуючи графік знань.

На рис 1.11 наведено приклад використання NER для вилучення сутностей, Титаніка та Північної Атлантики з тексту.

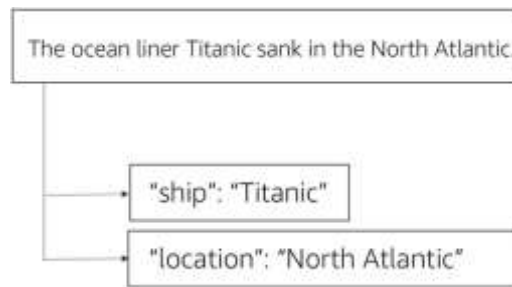


Рисунок 1.11 – Іменовані сутності

Після того як названі сутності витягнуто, ми можемо використовувати графік знань, щоб витягти значення. Графік знань поєднує знання предмета з машинним навчанням, щоб отримати значення. Механізм рекомендацій Amazon є прикладом графіка знань.

1.6 Висновки до першого розділу

Обробка природної мови також відома як NLP (Natural Language Processing). NLP розробляє обчислювальні алгоритми для автоматичного аналізу та представлення людської мови. Оцінюючи структуру мови, системи машинного навчання ML можуть обробляти великі набори слів, фраз і речень. Машинне навчання ML – це підмножина більш широкої галузі інформатики, яка відома як штучний інтелект AI. AI – це створення будівельних машин, які можуть виконувати завдання, які зазвичай виконує людина.

Одними з основних варіантів використання NLP є аналіз пошукових запитів, взаємодія людини з машиною, а також маркетингові та соціальні дослідження.

Обробка природної мови NLP – це ще одна сфера машинного навчання із збільшенням її використання. Наприклад, Amazon Alexa (або будь -який інший голосовий помічник) використовує NLP, щоб спробувати відповісти на ваші запитання. NLP – це мова, а також письмовий текст.

Після аналізу NLP в розділі розглянуто попередню обробку тексту.

2 АНАЛІЗ СЕРВІСІВ ДЛЯ ОБРОБКИ ПРИРОДНОЇ МОВИ

У цьому розділі ми розглянемо п'ять керованих служб машинного навчання, які можна використовувати для різних випадків використання. Ці сервіси спрощують процес створення програми машинного навчання.

2.1 Сервіс Amazon Transcribe

Amazon Transcribe – це повністю керована служба, яка використовує передові технології машинного навчання для розпізнавання мовлення в аудіофайлах і його транскрибування в текст. Ми можемо використовувати Amazon Transcribe для перетворення аудіо в текст і для створення програм, які включають вміст у аудіофайли [15,16].

Основні функції Amazon Transcribe:

- Розпізнання записаних голосів;
- Перетворення потокового аудіо в текст;
- Налаштування спеціалізованих словників;
- Інтеграція із програмами за допомогою WebSockets;
- Створювання субтитрів для кількох мов у режимі реального часу.

Amazon Transcribe може розпізнавати певні голоси в аудіофайлі, а ми можемо створити індивідуальний словник для термінів, які спеціалізуються на певному домені. Ми також можемо додати службу транскрипції до своїх програм шляхом інтеграції з WebSockets. WebSockets надають інтерфейс для доступу до Інтернету, який забезпечує двостороннє спілкування між програмою та Amazon Transcribe.

Деякі з найбільш поширених випадків використання Amazon Transcribe наведені на рис. 2.1 та включають:



Рисунок 2.1 – Випадки використання Amazon Transcribe

По-перше, медична транскрипція – медичні працівники можуть записувати свої нотатки, а Amazon Transcribe може записувати їх умовні нотатки у вигляді тексту.

По-друге, субтитри відео – організації, що займаються виробництвом відео, можуть автоматично створювати субтитри з відео. Це також можна зробити в режимі реального часу для додавання субтитрів (Closed Captioning, CC).

Крім того, маркування потокового вмісту – медіа-компанії можуть фіксувати та позначати вміст, а потім передавати його в Amazon Comprehend для подальшого аналізу.

Також, моніторинг колл-центру клієнтів – компанії можуть записувати дзвінки з обслуговування клієнтів або продажів і транскрибувати їх, а потім аналізувати результати для навчання або стратегічних можливостей

2.2 Сервіс Amazon Polly

Amazon Polly – це керований сервіс, який перетворює текст у реалістичне мовлення. Amazon Polly підтримує кілька мов і включає різні реалістичні голоси [16,17].

Основні функції Amazon Polly:

- Створення голосу з формату звичайного тексту або мови розмітки

синтезу мовлення (Speech Synthesis Markup Language, SSML);

- Створення вихідних даних у кількох аудіоформатах;
- Пропонує політику оплати за використання та використовує інфраструктуру AWS, щоб утримувати низькі витрати.

Amazon Polly може перетворювати текст у реальну мову. Ми можемо вводити або звичайні текстові файли, або файл, відформатований мовою розмітки синтезу мовлення або SSML. SSML – це мова розмітки, яка використовується для надання спеціальних інструкцій щодо звучання мови. Наприклад, якщо ми хочемо ввести паузу в потоці мовлення, ми можемо додати тег SSML, який вказує Amazon Polly робити паузу між двома словами.

Ми також можемо виводити мовлення з Amazon Polly у формати аудіопотоку MP3, Vorbis та імпульсно-кодової модуляції (PCM).

Amazon Polly має різноманітні програми. Поширені випадки використання Amazon Polly включають мобільні додатки (наприклад, програми для читання новин), ігри, платформи електронного навчання та програми доступності для людей із вадами зору.

Amazon Polly придатний для використання з регульованими робочими навантаженнями відповідно до Закону США про мобільність та підзвітність медичного страхування від 1996 року (Health Insurance Portability and Accountability Act, HIPAA) і Стандарту безпеки даних індустрії платіжних карток (Payment Card Industry Data Security Standard, PCI DSS).

Деякі з найбільш поширених випадків використання Amazon Polly наведені на рис. 2.2.

По-перше, виробництво служб новин – головні новинні компанії використовують Amazon Polly для створення вокального контенту безпосередньо зі своїх написаних історій.

По-друге, системи мовного навчання. Компанії, що навчають мов, використовують Amazon Polly для створення систем для вивчення нової мови.



Рисунок 2.2 – Випадки використання Amazon Polly

Також, системи навігації – Amazon Polly вбудовано в інтерфейси програмування прикладних програм (API), щоб розробники могли додавати голос до своїх програм на основі геоданих.

Крім того, виробництво анімації – аніматори використовують Amazon Polly, щоб додавати голоси своїм персонажам.

2.3 Сервіс Amazon Translate

Amazon Translate – це повністю керована служба перекладу тексту, яка використовує передові технології машинного навчання для забезпечення високоякісного перекладу на вимогу [16,18,20].

Основні функції Amazon Translate:

- Розвиток багатомовного користувацького досвіду для програм;
- Переклад документів кількома мовами;
- Аналіз вхідного тексту на декількох мовах

За допомогою Amazon Translate ми можемо створити багатомовний досвід у своїх програмах. Ми можемо створювати системи для читання документів однією мовою, а потім відтворювати або зберігати їх іншою мовою. Ми також можемо використовувати його як частину системи аналізу документів.

Amazon Translate повністю інтегрований з іншими сервісами машинного

навчання, такими як Amazon Comprehend, Amazon Transcribe та Amazon Polly. За допомогою цієї інтеграції ми можемо витягувати названі сутності, настрої та ключові фрази, інтегрувавши їх із Amazon Comprehend, створювати багатомовні субтитри за допомогою Amazon Transcribe та озвучувати перекладений вміст за допомогою Amazon Polly [11,18].

Деякі з найбільш поширених випадків використання Amazon Translate наведені на рис. 2.3.



Рисунок 2.3 – Випадки використання Amazon Translate

Перший варіант використання – створення міжнародних веб-сайтів. Ми можемо використовувати Amazon Translate для швидкої глобалізації своїх веб-сайтів.

Інший варіант використання – локалізація програмного забезпечення. Локалізація є великою вартістю для всього програмного забезпечення, орієнтованого на глобальну аудиторію. Amazon Translate може скоротити час розробки програмного забезпечення та значно зменшити витрати на локалізацію програмного забезпечення.

Amazon Translate також можна використовувати для розробки багатомовних чат-ботів. Чат-боти використовуються для створення більш людського інтерфейсу для програм. За допомогою Amazon Translate ми можемо створити чат-бота, який розмовляє кількома мовами [18,19].

Останнім прикладом використання є міжнародне управління медіа. Компанії, які керують медіа для глобальної аудиторії, використовують Amazon Translate, щоб зменшити витрати на локалізацію.

2.4 Сервіс Amazon Comprehend

Amazon Comprehend використовує NLP для отримання інформації про вміст документів. Він розвиває розуміння, розпізнаючи сутності, ключові фрази, мову, почуття та інші загальні елементи в документі [16,20,21,22].

Основні функції Amazon Comprehend:

- Витягнення ключових об'єктів з документа, наприклад людей або локацій;
- Визначення мови, яка використовується в документі;
- Визначення настроїв, таких як позитивні, негативні, нейтральні чи змішані, які виражені в документі;
- Визначення частину мови для окремих слів у документі.

Amazon Comprehend реалізує багато методів NLP, які ми розглянули в першому розділі цієї роботи. Ми можемо витягувати ключові сутності, виконувати аналіз настроїв і позначати слова частинами мови [11,21,22].

Деякі з найбільш поширених випадків використання Amazon Comprehend наведені на рис. 2.4.

Перший випадок – аналіз юридичних та медичних документів. Юридичні страхові та медичні організації використовують Amazon Comprehend для виконання багатьох функцій NLP, які ми розглянули в першому розділі цієї роботи.

Виявлення фінансового шахрайства – ще один варіант використання Amazon Comprehend. Банківські, фінансові та інші установи використовують його для вивчення дуже великих наборів даних про фінансові операції, щоб виявити шахрайство та знайти модель незаконних операцій.



Рисунок 2.4 – Випадки використання Amazon Comprehend

Інше використання – для масштабного аналізу мобільних додатків. Розробники мобільних додатків використовують Amazon Comprehend, щоб шукати закономірності використання своїх програм, щоб вони могли розробляти покращення.

Також, його можна використовувати для керування вмістом. Медіа та інші контент-компанії можуть використовувати Amazon Comprehend для позначення вмісту для аналізу та керування.

2.5 Сервіс Amazon Lex

Amazon Lex – це служба AWS для створення розмовних інтерфейсів для програм за допомогою голосу та тексту. З Amazon Lex той самий механізм розмови, який працює на Amazon Alexa, тепер доступний будь-якому розробнику [16,19,20,23,24].

Основні функції Amazon Lex:

- Створення чат-бота, який може взаємодіяти за допомогою голосу та тексту, щоб задавати запитання, отримувати відповіді чи виконувати завдання;
- Автоматичне масштабування свого чат-бота за допомогою AWS Lambda;
- Зберігання лог-файлів розмов для аналізу.

За допомогою Amazon Lex ми можемо додати інтерфейс людської мови до своїх програм. Amazon Lex дозволяє використовувати той самий механізм розмови, що й Amazon Alexa. Ми можемо автоматично збільшити потужність свого рішення Amazon Lex, створивши функції AWS Lambda для масштабування на вимогу. Крім того, ми можемо зберігати файли журналів розмов для подальшого аналізу.

Деякі з найбільш поширених випадків використання Amazon Lex наведені на рис. 2.5.



Рисунок 2.5 – Випадки використання Amazon Lex

Перший варіант використання – це створення інтерфейсів для управління запасами та продажів. Голосові інтерфейси стають все більш поширеними. Компанії використовували Amazon Lex для додавання чат-ботів до своїх інвентаризацій і програм продажу [23,24,25].

Amazon Lex також можна використовувати для розробки інтерактивної допомоги. Поєднуючи Amazon Lex з іншими послугами машинного навчання, клієнти створюють більш складну допомогу для багатьох різних галузей.

Інше використання Amazon Lex – створення інтерфейсів обслуговування клієнтів. Голосові програми, схожі на людину, швидко стають нормою для багатьох програм обслуговування клієнтів. Amazon Lex може скоротити час, необхідний для розробки цих чат-ботів, і підвищити їх якість.

Останній приклад використання – запити до баз даних людською мовою.

Amazon Lex поєднується з іншими службами баз даних AWS для створення складних програм для аналізу даних із мовним інтерфейсом, схожим на людину.

2.6 Висновки до другого розділу

Amazon Transcribe може автоматично перетворювати розмовну мову на текст, його використовують для медичної транскрипції, субтитрів для відео, маркування потокового вмісту та для моніторингу колл-центрів.

Amazon Polly може перетворювати письмовий текст у розмовну мову, наприклад, для виробництва служб новин, систем навчання мові, систем навігації та виробництва анімацій.

Amazon Translate може створювати переклади між мовами в режимі реального часу, який можна використовувати для створення міжнародних веб-сайтів, локалізації програмного забезпечення, розробки багатомовних чат-ботів та міжнародного управління медіа.

Amazon Comprehend автоматизує багато випадків використання NLP, які розглядаються в цій роботі, наприклад, для аналізу юридичних та медичних документів, виявлення фінансового шахрайства, масштабного аналізу мобільних додатків або позначення вмісту для аналізу та керування.

І, нарешті, Amazon Lex може створити інтерфейс, схожий на людину, який може взаємодіяти за допомогою голосу та тексту, щоб задавати запитання, отримувати відповіді чи виконувати завдання, наприклад, створення інтерфейсів для управління запасами та програм продажів, інтерактивної допомоги, обслуговування клієнтів або запитів до баз даних людською мовою.

3 МОДЕЛЬ СТВОРЕННЯ СЕРЕДОВИЩА ДЛЯ РОБОТИ ЧАТ-БОТА З ОБРОБКОЮ ПРИРОДНОЇ МОВИ

Обробка природної мови є поширеним випадком використання машинного навчання. В даній роботі було розглянуто, як використовувати Amazon Lex для створення чат-бота, з яким користувачі можуть взаємодіяти, щоб записатися на прийом до стоматолога.

Наступна діаграма ілюструє середовище, яке ми створили в цій роботі (рис. 3.1)

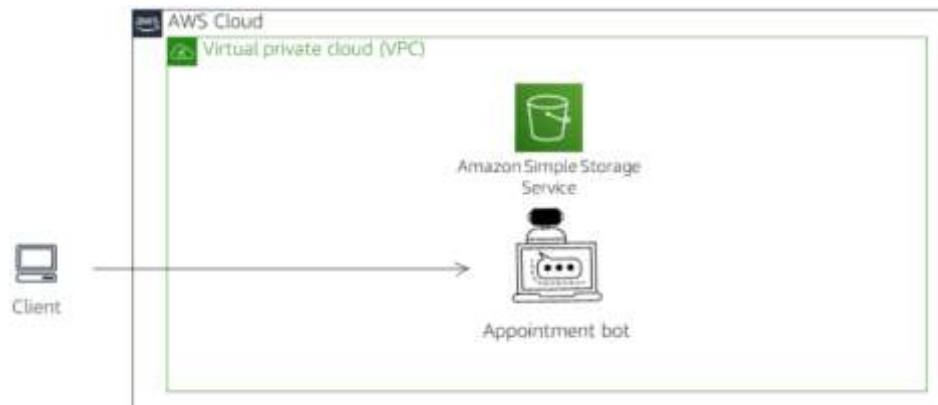


Рисунок 3.1 – Середовище, яке буде створене

Ми виконали наступні кроки високого рівня:

- Створили і протестували бота за допомогою плану зустрічі Amazon Lex ScheduleAppointment;
- Створили і протестували функцію AWS Lambda для виконання завдань ініціалізації, перевірки та виконання;
- Налаштували намір MakeAppointment для використання функції Lambda як гачка коду;
- Перевірили бота на консолі та через статичну веб-сторінку, розміщену на Amazon Simple Storage Service (Amazon S3).

3.1 Створення бота за допомогою Amazon Lex

В цій частині роботи ми створили бота за допомогою Консолі керування AWS. Для цього виконали наступні кроки:

- На консолі керування AWS у меню Служби вибрали Amazon Lex. Відкрилась початкова сторінка цього сервісу, на якій знаходився короткий опис сервісу та кнопка для початку роботи, зображена на рис. 3.1;

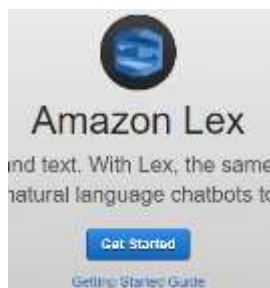


Рисунок 3.1 – Кнопка для початку роботи з Amazon Lex

- На сторінці Amazon Lex натиснули почати. З'явився початковий інтерфейс, зображений на рис. 3.2;

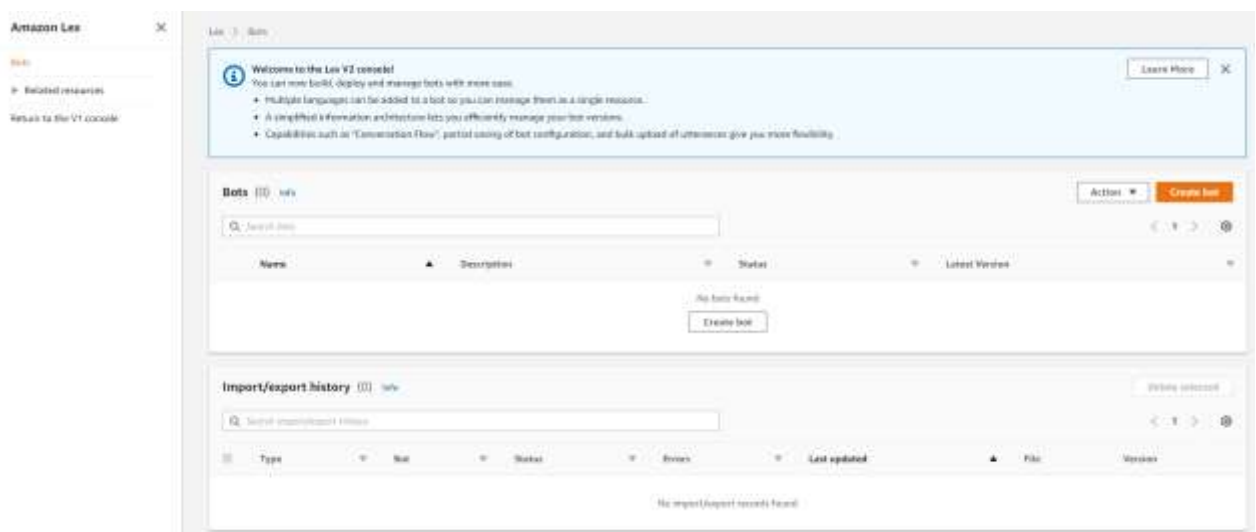


Рисунок 3.2 – Початковий інтерфейс консолі V2

- У лівій навігаційній панелі вибрали Повернутись до консолі V1, інтерфейс змінився та виглядав, як показано на рис. 3.3;

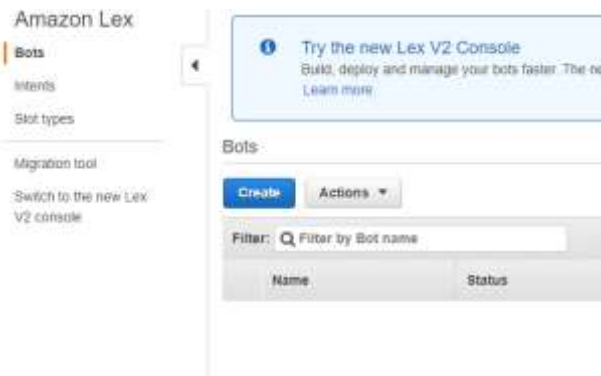


Рисунок 3.3 – Початковий інтерфейс консолі V1

- Вибрали створити, відкрився інтерфейс створення бота, зображений на рис. 3.4;

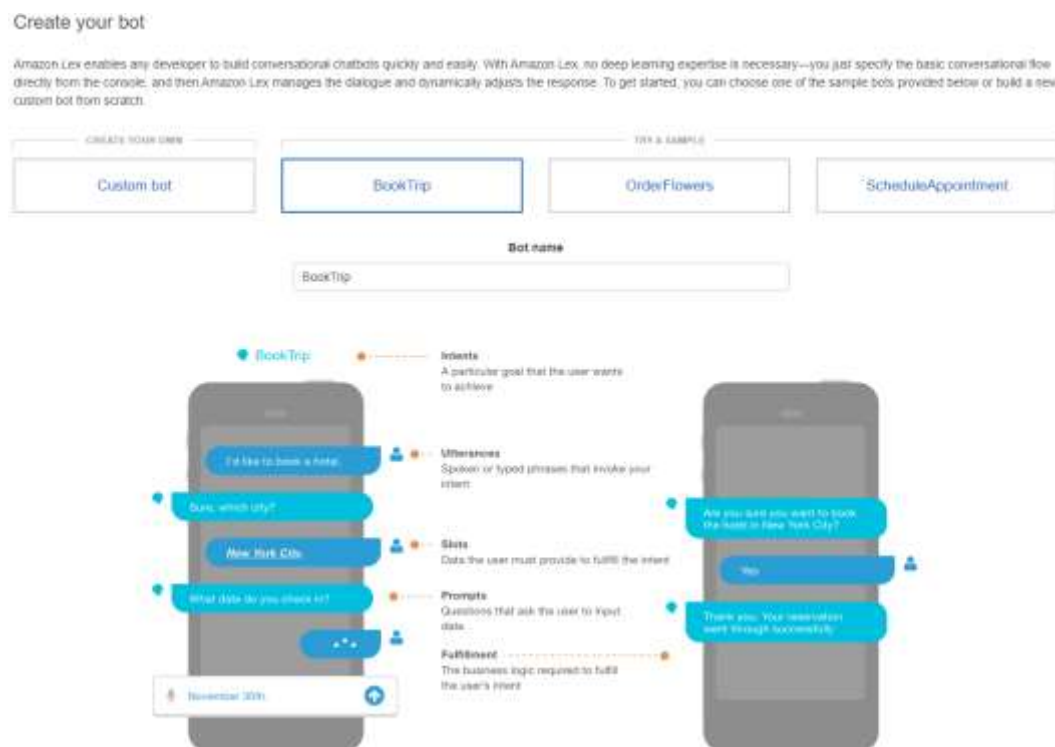


Рисунок 3.4 – Інтерфейс створення бота у Amazon Lex

- На сторінці створення бота вибрали план ScheduleAppointment, інтерфейс змінився відповідно, як показано на рис. 3.5;

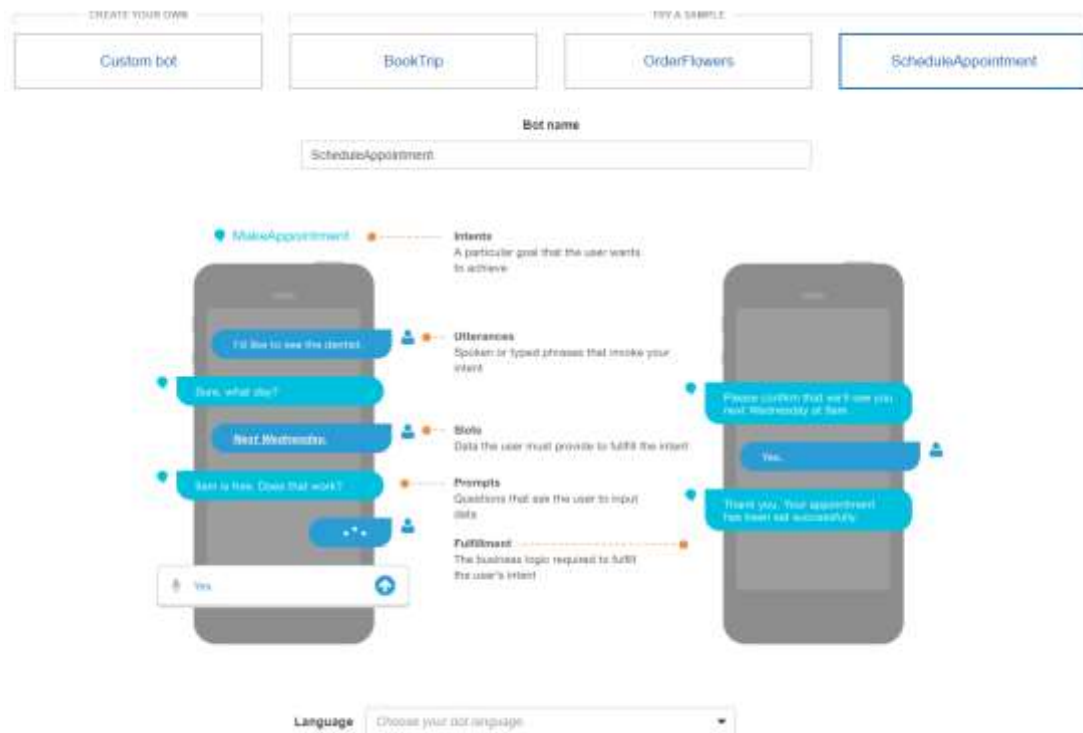


Рисунок 3.5 – Інтерфейс створення бота для створення зустрічей

- У розділі мова обрали англійську (США), як зображено на рис 3.6;

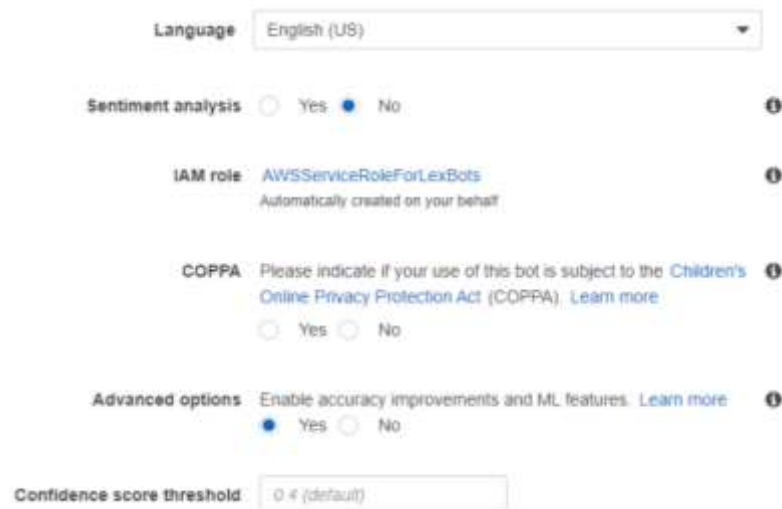


Рисунок 3.6 – Початкові налаштування бота

- Для розділу COPPA вибрали Ні (рис. 3.7);

COPPA Please indicate if your use of this bot is subject to the [Children's Online Privacy Protection Act \(COPPA\)](#). [Learn more](#)

☐ Yes ☒ No

Рисунок 3.7 – Вимкнення COPPA для нашого бота

- Натиснули створити, відкрився інтерфейс, на якому можна протестувати нашого бота та опублікувати в разі успішного тестування, зображений на рис. 3.8.

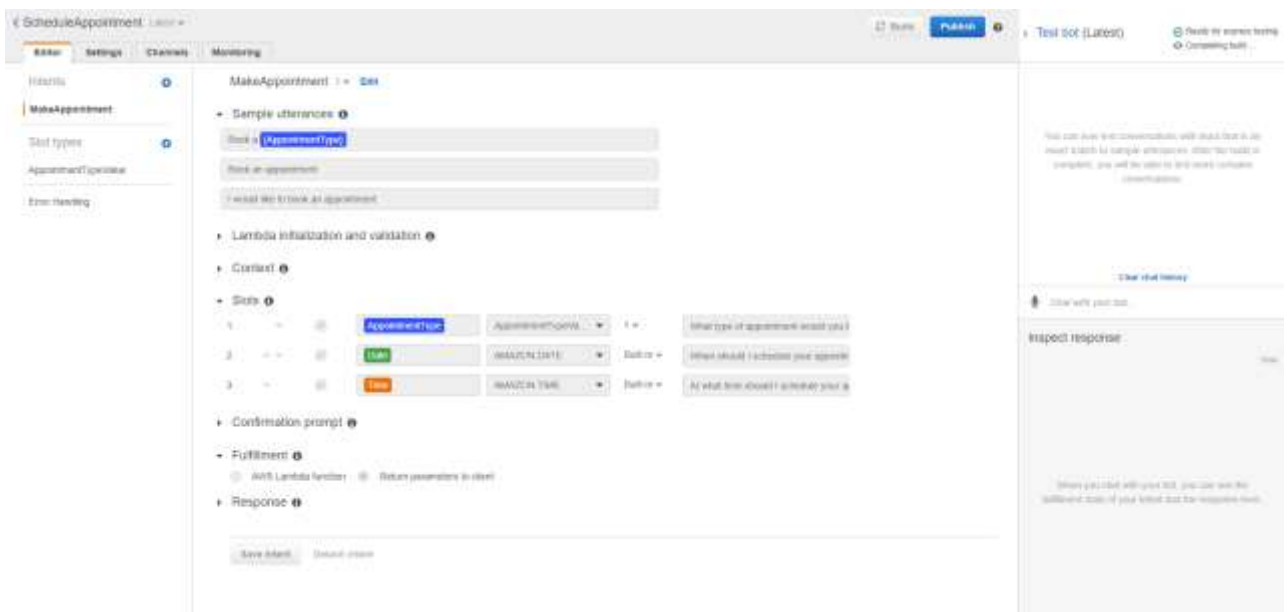


Рисунок 3.8 – Інтерфейс тестування та публікування бота

3.2 Тестування створеного бота

Тепер ми перевірили свого бота за допомогою тестового вікна на консолі. Після того, як ми побачили статус про завершення збірки, перевірили статус нашого бота, ввівши такі значення:

- I would like to make an appointment;
- A root canal;
- 5/1/2020;
- 4:00 PM;
- Yes.

Отримали підтвердження, зображене на рис. 3.9.

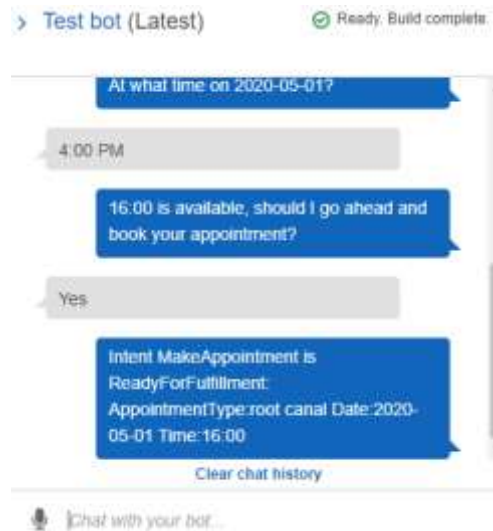


Рисунок 3.9 – Відповідь з підтвердженням запису від створеного бота

Також переглянули відповідь для перевірки, зображену на рис. 3.10.

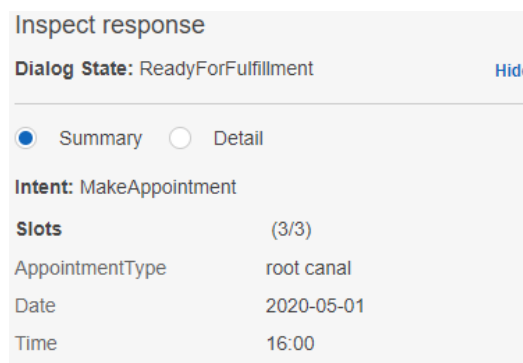


Рисунок 3.10 – Відповідь для перевірки коректності даних занесених ботом

3.3 Створення функції AWS Lambda

В даній частині роботи ми створили функцію AWS Lambda за допомогою Консолі керування AWS. Для цього виконали наступні кроки:

- На консолі керування AWS у меню Служби обрали Lambda;
- Натиснули створити функцію;

- Вибрали вкладку Використовувати план, відкрилися відповідні креслення, зображені на рис. 3.11;

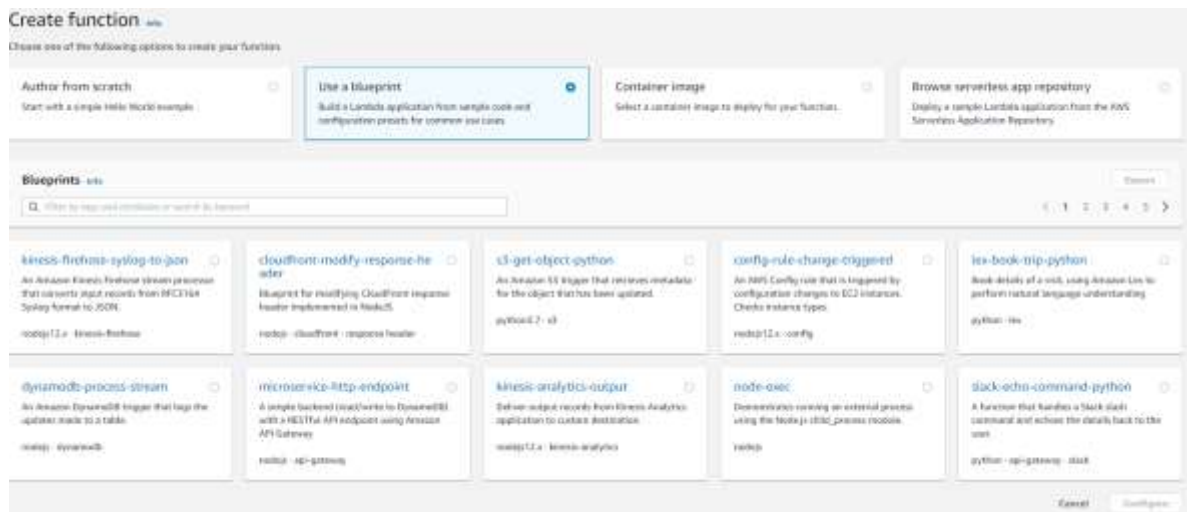


Рисунок 3.11 – Інтерфейс вибору креслень для створення лямбда-функції

- У вікні пошуку відфільтрували креслення Amazon Lex, ввівши Lex, як зображено на рис. 3.12;

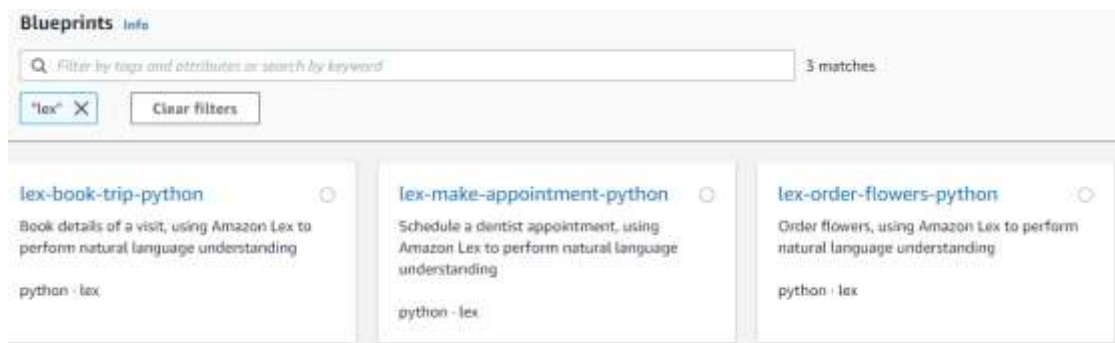


Рисунок 3.12 – Фільтр креслень за ключем “lex”

- Відкрили план lex-make-appointment-python, вибравши його, відкрився інтерфейс конфігурації для обраного плану функції, зображеного на рис 3.13;

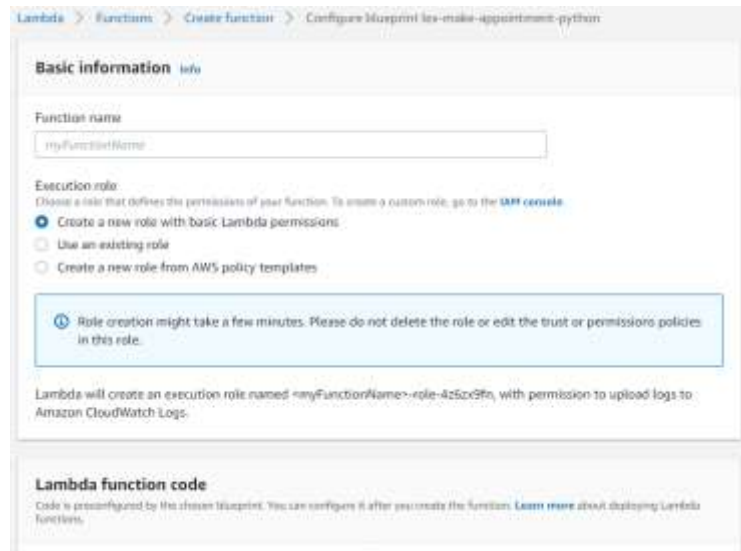


Рисунок 3.13 – Інтерфейс конфігурації для плану lex-make-appointment-python

- Ввели ім'я функції MakeAppointmentCodeHook;
- Для ролі виконання створили нову роль із шаблонів політики AWS;
- Для імені ролі ввели myLexrole, тепер інтерфейс виглядав, як зображено на рис. 3.14;

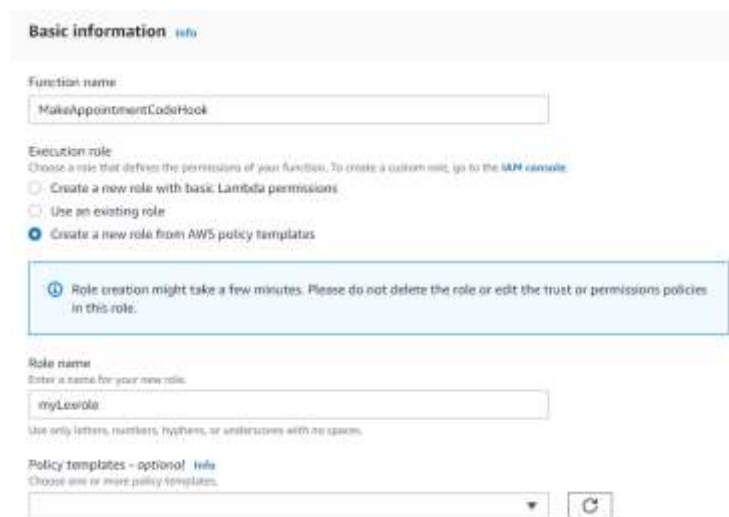


Рисунок 3.14 – Інтерфейс конфігурації після введення даних

- Натиснули створити функцію.

Коли функція була готова вона з'явилася у вікні Lambda Designer, зображена на рис. 3.15.



Рисунок 3.17 – Результат виконання тестування функції Lambda

Також можна переглянути деталі тесту, розгорнувши розділ Деталі, як це продемонстровано на рис. 3.17.

3.5 Оновлення намірів бота

Кожен бот, якого ми створюємо в Amazon Lex, має намір. Намір – це дія, яку виконує бот. У цій частині роботи ми оновили наміри використовуючи створену раніше функцію AWS Lambda.

- На консолі керування AWS у меню Служби вибрали Amazon Lex;
- У списку ботів обрали ScheduleAppointment, якого ми створили на початку;
- Розгорнули розділ «Ініціалізація та перевірка Lambda»;
- Вибрали редагувати;
- Отримали повідомлення про те, що ми збираємося дати Amazon Lex дозвіл на виклик нашої функції Lambda, зображене на рис. 3.18. Натиснули Ок;



Рисунок 3.18 – Повідомлення про дозвіл Amazon Lex викликати Lambda

- Обрали опцію «гачок» коду ініціалізації та перевірки;
- У спадному меню обрали MakeAppointmentCodeHook, зображеному на рис. 3.19.

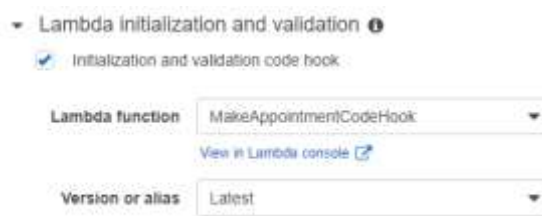


Рисунок 3.19 – Налаштування гачку коду у розділі ініціалізації

Окрім додавання коду для ініціалізації бота, ми також додали код для виконання запиту.

- У розділі «Виконання» вибрали перемикач функції AWS Lambda;
- У спадному списку функцій вибрали MakeAppointmentCodeHook, зображеному на рис. 3.20;

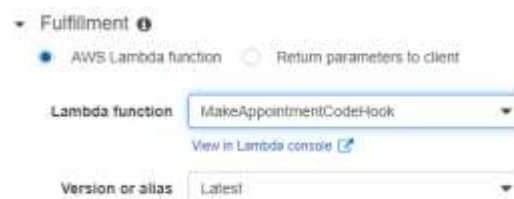


Рисунок 3.20 – Налаштування гачку коду у розділі виконання

- Зберегли намір натиснувши відповідну кнопку.

3.6 Висновки до третього розділу

Було розглянуто синтез моделі середовища для роботи чат-бота з обробкою природньої мови на основі Amazon Lex. Виконано наступні кроки високого рівня: створення і тестування бота за допомогою плану зустрічі Amazon Lex ScheduleAppointment; створення і тестування функцію AWS Lambda для виконання завдань ініціалізації, перевірки та виконання; налаштування наміру MakeAppointment для використання функції Lambda як гачка коду.

Кожен бот, який створюється в Amazon Lex, має намір. Намір – це дія, яку виконує бот.

4 ТЕСТУВАННЯ МОДЕЛІ

4.1 Побудова та тестування нашого бота

Тепер ми перевірили нашого бота, щоб переконатися, що він виконує лямбда-функцію.

- Вибрали Зібрати;

Нам було запропоновано повідомлення про те, що ми можемо продовжувати редагувати свого бота, поки він збирається, зображене на рис. 4.1.



Рисунок 4.1 – Повідомлення про можливість редагування бота під час збирання

- Вибрали Зібрати;

Після завершення збірки ми отримали підтвердження, що збірка нашого бота пройшла успішно, зображене на рис. 4.2.

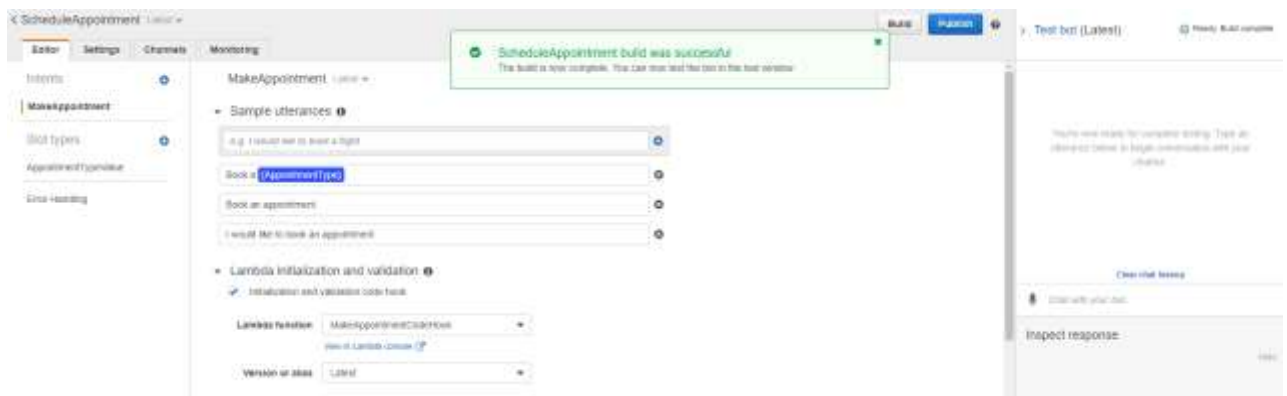


Рисунок 4.2 – Успішна збірка бота

Перевірили бота ввівши наступні дані:

- Make an appointment;
- root canal;
- Вибрали один із часів, які відображаються;
- yes.

Отримали підтвердження, зображене на рис. 4.3.

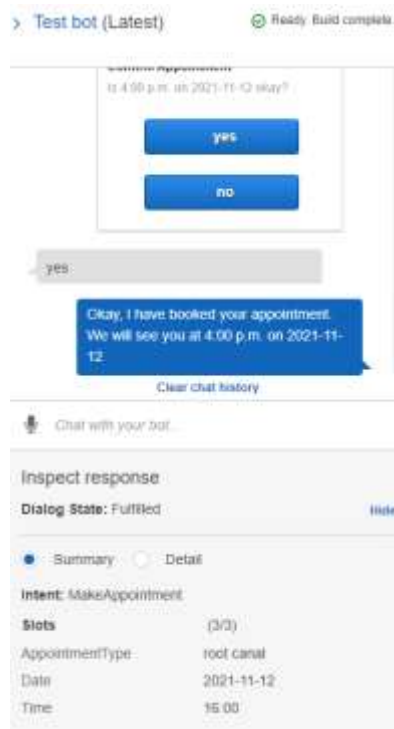


Рисунок 4.3 – Відповідь для перевірки оновленого бота

Тепер, коли у нас була робоча версія нашого бота, ми опублікували його, щоб можна було перевірити його, викликавши його з веб-сторінки.

- Натиснули опублікувати;
- Для псевдоніма нашого бота ввели унікальне ім'я, rzhvbot (рис. 4.4). Цей псевдонім нам знадобився в подальшому;
- Натиснули опублікувати;

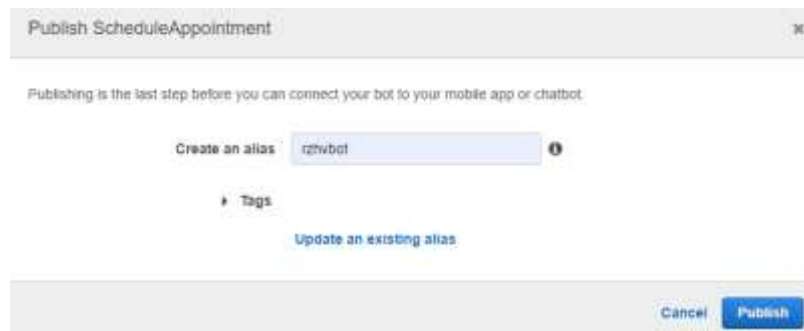


Рисунок 4.4 – Присвоєння псевдоніма для бота

Після того, як бот завершив публікацію, ми отримали сповіщення про підтвердження, зображене на рис. 4.5.

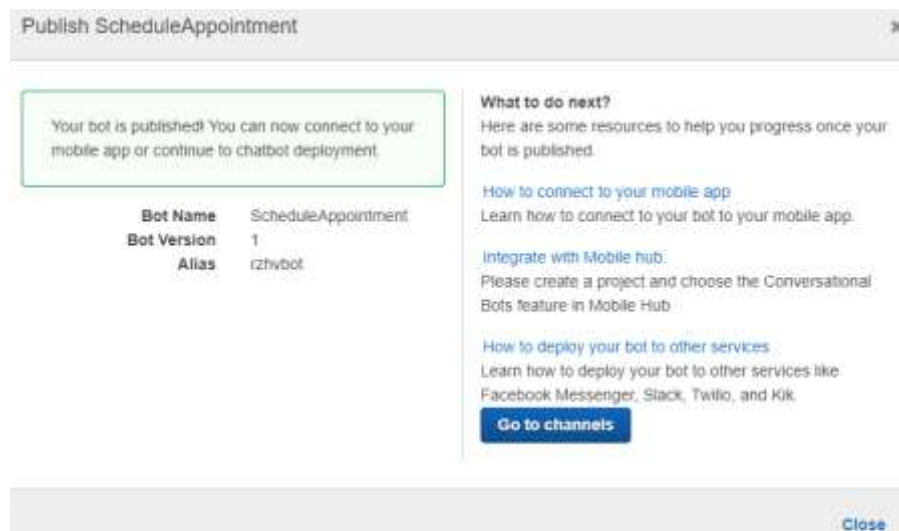


Рисунок 4.5 – Сповіщення про підтвердження публікації бота

- Закрили сповіщення Publish ScheduleAppointment, вибравши Закрити.

Тепер, коли наш бот працював, настав час перевірити його, запустивши з веб-сторінки. Найпростіший спосіб провести тест – створити статичну веб-сторінку та розмістити її на Amazon S3. Веб-сторінка викличе API Amazon Lex для завантаження нашого бота. Відвідувачі веб-сторінки можуть потім взаємодіяти з нею.

4.2 Налаштування пулу ідентифікаційних даних Amazon Cognito

Тепер ми налаштували веб-сторінку для тестування нашого бота для зустрічі. Ця веб-сторінка була розміщена в Amazon S3 як статична веб-сторінка. Щоб забезпечити безпеку на цій сторінці, спочатку потрібно було налаштувати пул ідентифікаційних даних Amazon Cognito.

Для налаштування пулу ідентифікаційних даних, ми:

- На консолі керування AWS у меню Служби вибрали Cognito;
- На інтерфейсі обрали Manage Identity Pools, зображеному на рис. 4.6;



Рисунок 4.6 – Інтерфейс вибору виду пулу

- На сторінці майстра «Початок роботи» для імені пулу ідентифікаційних даних ввели: myidentitypool;
- Вибрали Увімкнути доступ до неавтентифікованих ідентифікаторів, зображений на рис. 4.7 та натиснули Створити пул;

 The image shows a screenshot of the 'Create new identity pool' wizard in the Amazon Cognito console. The title is 'Create new identity pool'. Below the title, there is a note: 'Identity pools are used to store and user identities. To create a new identity pool, enter a unique name.' There is a text input field for 'Identity pool name' with the value 'myidentitypool' and a green checkmark icon. Below the input field, there is a hint: 'Example: My App Name'. The wizard has several sections: 'Unauthenticated identities', 'Authentication flow settings', and 'Authentication providers'. The 'Unauthenticated identities' section is expanded, showing a checkbox for 'Enable access to unauthenticated identities' which is checked. Below this, there is a note: 'Enabling this action means that anyone with internet access can be granted AWS credentials. Unauthenticated identities are typically users who do not log in to your application. Typically, the permissions that you assign for unauthenticated identities should be more restrictive than those for authenticated identities.' The 'Authentication flow settings' section is also expanded, showing a note about the authentication process and a checkbox for 'Allow Basic (Classic) Flow' which is unchecked. The 'Authentication providers' section is also expanded. At the bottom right, there are 'Cancel' and 'Create Pool' buttons.

Рисунок 4.7 – Інтерфейс створення ідентифікаційного пулу

- Розгорнули розділ Переглянути деталі;
- Записали назви двох ролей (одна роль закінчиться на Auth_Role, а інша роль закінчиться на Unauth_Role), зображених на рис. 4.8. Пізніше в роботі нам знадобляться обидві ці назви ролей;

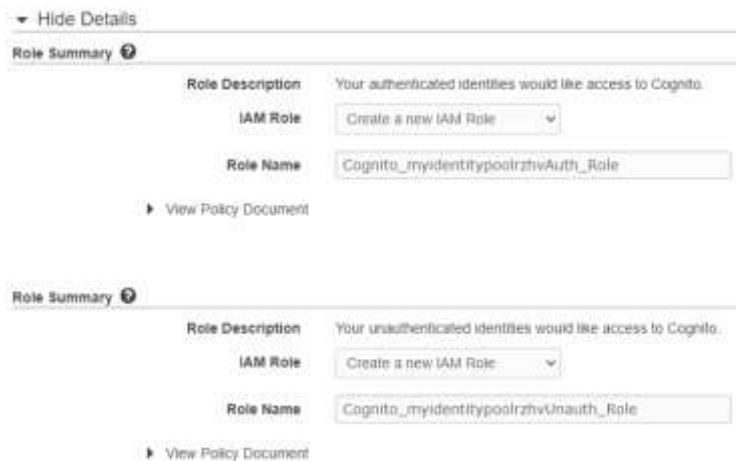


Рисунок 4.8 – Інтерфейс створення ролей для створеного пулу

- Натиснули дозволити;
- Записали ідентифікатор IdentityPoolID, оскільки цей ідентифікатор нам знадобиться пізніше, зображений на рис. 4.9.

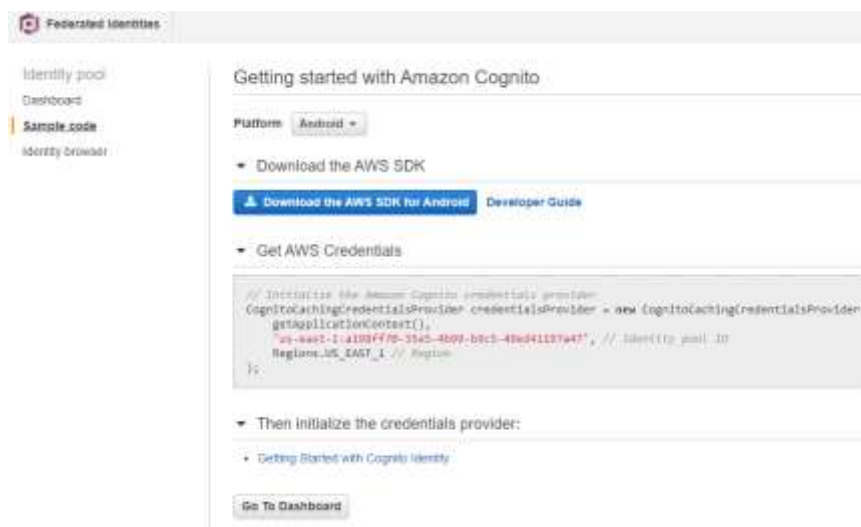


Рисунок 4.9 – Інтерфейс видачі ідентифікатора

Наш ідентифікатор відповідно "us-east-1:a198ff70-35e5-4b99-b9c5-49ed41197a47".

4.3 Змінення ролей IAM, для доступу до Amazon Lex

Веб-сторінці, на якій розміщено нашого бота, потрібно було дозволити доступ до Amazon Lex. Для цього двом ролям, створеним у пулі ідентифікаційних даних Amazon Cognito, були потрібні дозволи на доступ до Amazon Lex. Ми повинні були налаштувати AWS Identity and Access Management (IAM), щоб надати ці дозволи ролям пулу ідентифікаційних даних. Для цього зробили наступні кроки:

- На консолі керування AWS у меню Служби вибрали IAM;
- Натиснули Ролі;
- Ввели кілька перших символів імені Auth_Role, яке ми зазначили в підрозділі 4.2, зображеному на рис.4.10;
- У списку результатів пошуку вибрали роль, яку ми створили;
- Натиснули Вкласти політики;
- У вікні пошуку ввели AmazonLex;



Рисунок 4.10 – Інтерфейс пошуку ролей для IAM

- Обрали політики AmazonLexReadOnly та AmazonLexRunBotsOnly, зображені на рис 4.11;



Рисунок 4.11 – Інтерфейс обирання політик для ролі IAM

- Натиснули Вкласти політику. Отримали сторінку, зображену на рис 4.12;



Рисунок 4.12 – Інтерфейс дозволів для ролі Auth_Role

- Повторили той самий процес, щоб приєднати дві політики до Unauth_Role. Отримали сторінку, зображену на рис. 4.13;



Рисунок 4.13 – Інтерфейс дозволів для ролі Unauth_Role

4.4 Створення сегмента Amazon S3

Далі, після того, як ми налаштували дозволи безпеки, ми повинні були створити сегмент S3 для розміщення нашої веб-сторінки.

- Спочатку створили два наступні файли веб-сторінки у локальному каталозі файлів.
 - index.html
 - error.html

Файл index.html містить скрипт, який завантажить нашого бота.

- На консолі керування AWS у меню Служби вибрали S3;
- На сторінці Amazon S3 вибрали Створити сегмент;
- Ввели назву свого сегмента. Оскільки всі назви сегментів S3 мають бути унікальними, ввели lexrzhvbot;
- Натиснули створити сегмент, отримали його у списку сегментів, зображеному на рис. 4.14;



Рисунок 4.14 – Створений сегмент у Amazon S3

- На сторінці Сегменти вибрали сегмент, який ми щойно створили, відкрився інтерфейс взаємодії з сегментом, зображений на рис. 4.15;

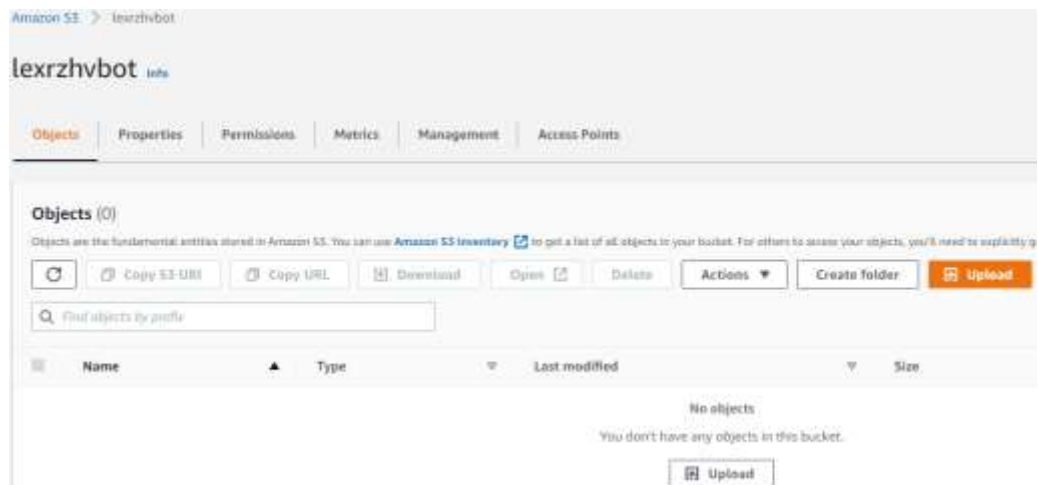


Рисунок 4.15 – Інтерфейс створеного сегменту

- На сторінці огляду сегмента вибрали Завантажити;
- На сторінці Завантаження вибрали Додати файли;
- Перейшли до розташування файлів index.html та error.html, які ми створили раніше;
- Вибрали обидва файли, а потім натиснули Завантажити;
- Після завершення завантаження повернулися на сторінку огляду сегмента, вибравши Вихід;

В результаті отримали сегмент з двома файлами, зображених на рис. 4.16.

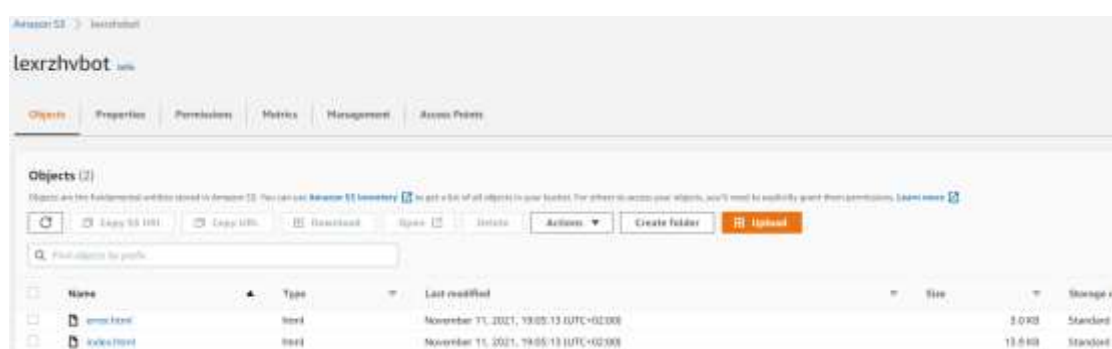


Рисунок 4.16 – Інтерфейс створеного сегменту після виконаних маніпуляцій

- На вкладці Властивості прокрутили вниз до розділу Статичний хостинг веб-сайтів і натиснули Редагувати;
- Вибрали Enable;

- Для документа «Індекс» ввели: index.html;
- Для сторінки помилки ввели: error.html;

Сторінка редагування після цього мала вигляд, зображений на рис. 4.17.

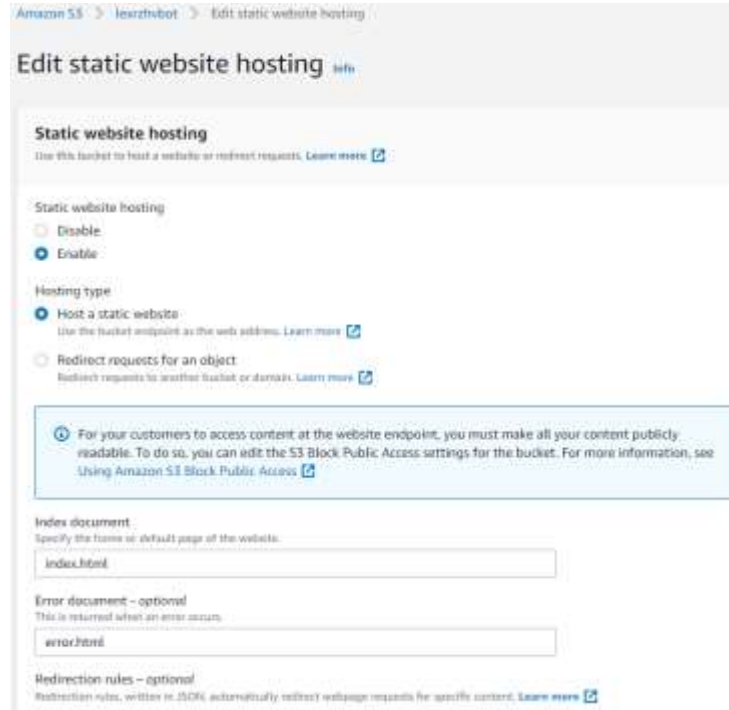


Рисунок 4.17 – Сторінка редагування статичного хостингу веб-сайтів

- Натиснули зберегти зміни.

4.5 Оновлення та тестування демонстраційного файлу

Тепер ми оновили демонстраційний файл HTML, щоб він використовував пул ідентифікаційних даних Amazon Cognito, який ми створили в підрозділі 3.7.

Використали текстовий редактор, щоб внести наступні зміни до HTML-сторінки.

- У рядку 144 сценарію додали IdentityPoolId для пулу ідентифікаційних даних, який ми створили в підрозділі 4.2 (рис. 4.18);

```
144 IdentityPoolId: 'us-east-1:a198ff70-35e5-4b99-b9c5-49ed41197a47',
```

Рисунок 4.18 – Рядок 144 сценарію index.html

- У рядках 185 і 186 сценарію додали botAlias і botName для нашого бота, зображено на рис. 4.19;

```
185 botAlias: 'rzhvbot',
186 botName: 'ScheduleAppointment',
```

Рисунок 4.19 – Рядки 185 та 186 сценарію index.html

- Зберегли оновлену сторінку HTML локально;
- Вибрали вкладку Об'єкти для сегмента, який ми створили в підрозділі 4.4;
- Вибрали Завантажити та завантажили файл HTML, який ми відредагували;
- Вибрали Вихід, щоб закрити сторінку завантаження, отримали сегмент з оновленим файлом, зображений на рис. 4.10;
- Вибрали вкладку Дозволи;
- У розділі Блокувати публічний доступ вибрали Редагувати;

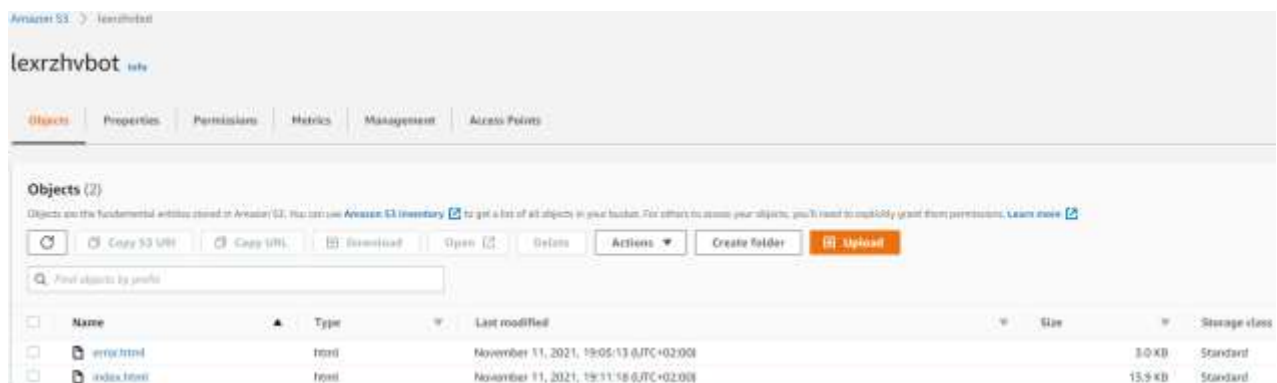


Рисунок 4.10 – Оновлений сегмент S3

- Зняли такі параметри (рис. 4.11):
 - Блокувати весь публічний доступ;
 - Блокувати загальнодоступний доступ до сегментів і об'єктів, наданих за допомогою нових політик загальнодоступного сегмента або точки доступу;

- Блокувати загальнодоступний і міжобліковий доступ до сегментів і об'єктів за допомогою будь-яких політик загальнодоступного сегмента або точки доступу.



Рисунок 4.11 – Сторінка редагування Блокування публічного доступу

- Вибрали Зберегти зміни;
- У діалоговому вікні підтвердження ввели Confirm, а потім вибрали Підтвердити (рис. 4.12);



Рисунок 4.12 – Вікно для підтвердження змін

- У розділі Політика сегмента вибрали Редагувати;
- Щоб надати загальнодоступний доступ для читання веб-сторінки, скопіювали наведену нижче політику сегмента. У редакторі політики сегмента вставили політику, зображену на рис. 4.13.

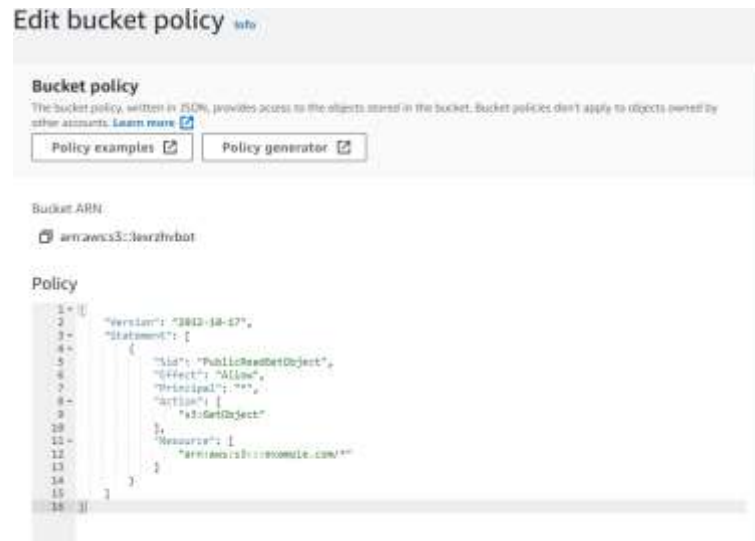


Рисунок 4.13 – Сторінка редактора політики сегмента

- У розділі Resource політики відредагували значення імені ресурсу Amazon (ARN), замінивши example.com на ім'я нашого сегмента, зображену на рис. 4.14;

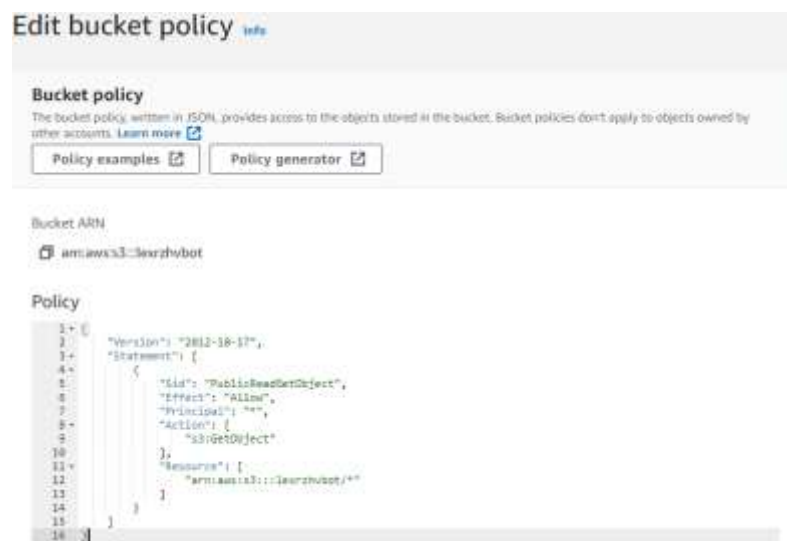


Рисунок 4.14 – Сторінка редактора політики сегмента зі зміненим ресурсом

- Вибрали Зберегти зміни. Ми отримали попередження, що сегмент має відкритий доступ, зображене на рис 4.15;



Рисунок 4.15 – Попередження про відкритий доступ до сегмента

- Вибрали вкладку Властивості;
- Прокрутили вниз до розділу Статичний хостинг веб-сайтів, зображену на рис. 4.16;



Рисунок 4.16 – Сторінка з розділом Статичний хостинг веб-сайтів

- Унизу розділу Статичний хостинг веб-сайтів вибрали URL-адресу;
- Відкрилася наша веб-сторінка, і ми змогли взаємодіяти зі своїм ботом, як це робили в консолі керування AWS, зображена на рис. 4.17.



Рисунок 4.17 – Інтерфейс взаємодії з ботом через веб-сторінку

4.6 Висновки до четвертого розділу

Було зібрано Amazon Lex та Lambda в одне середовище, перевірено бота, для доведення, що він виконує лямбда-функцію. Далі, відбулося тестування публікації бота, щоб можна було перевірити його виклик з веб-сторінки. Найпростіший спосіб провести тест – створити статичну веб-сторінку та розмістити її на Amazon S3. Перевірено виклик API Amazon Lex з веб-сторінки для завантаження нашого бота, завдяки чому відвідувачі веб-сторінки змогли взаємодіяти з нею. Було протестовано налаштування пулу ідентифікаційних даних Amazon Cognito та створено відповідні ролі (одна роль закінчується на Auth_Role, а інша на Unauth_Role), для забезпечення безпеки на цій сторінці.

Після чого веб-сторінці, на якій розміщено нашого бота, потрібен дозвіл на доступ до Amazon Lex. Для цього двом ролям, створеним у пулі ідентифікаційних даних Amazon Cognito, потрібні дозволи на доступ до Amazon Lex. Перевірили налаштування AWS Identity and Access Management (IAM) та надали відповідні дозволи ролям пулу ідентифікаційних даних. Після налаштування дозволів безпеки і створення сегмента S3 для розміщення нашої веб-сторінки, відбулося тестування оновленого демонстраційного файлу

HTML, і перевірка використання пулу ідентифікаційних даних Amazon Cognito. У фінальному тестуванні перевірили взаємодію зі створеним в магістерській роботі ботом. На сторінці з розділом Статичний хостинг веб-сайтів за вибраним URL-адресом відкривається створена веб-сторінка.

ВИСНОВКИ

Завдання на кваліфікаційну роботу магістра виконане.

У першому розділі було розглянуто обробку природної мови, її проблеми та випадки використання. Обробка природної мови також відома як NLP (Natural Language Processing). NLP розробляє обчислювальні алгоритми для автоматичного аналізу та представлення людської мови. Оцінюючи структуру мови, системи машинного навчання ML можуть обробляти великі набори слів, фраз і речень. Машинне навчання ML – це підмножина більш широкої галузі інформатики, яка відома як штучний інтелект AI. AI – це створення будівельних машин, які можуть виконувати завдання, які зазвичай виконує людина.

Проблеми штучного інтелекту зазвичай охоплюють багато сфер дослідження: обробка природної мови, міркування, представлення знань, навчання, сприйняття та взаємодія з фізичним середовищем. AI ще не є реальністю, якщо ви не живете в симуляції. Однак з кожним роком це стає ближчим у кожній із цих областей.

Обробка природної мови NLP – це ще одна сфера машинного навчання із збільшенням її використання. Наприклад, Amazon Alexa (або будь-який інший голосовий помічник) використовує NLP, щоб спробувати відповісти на ваші запитання. NLP – це мова, а також письмовий текст.

Після аналізу NLP в розділі розглянуто попередню обробку тексту.

У другому розділі проаналізовані досягнення компанії Amazon, розглянуті основні сервіси, які використовуються для обробки природної мови. Amazon Transcribe може автоматично перетворювати розмовну мову на текст, його використовують для медичної транскрипції, субтитрів для відео, маркування потокового вмісту та для моніторингу колл-центрів. Amazon Polly може перетворювати письмовий текст у розмовну мову, наприклад, для виробництва служб новин, систем навчання мові, систем навігації та

виробництва анімацій. Amazon Translate може створювати переклади між мовами в режимі реального часу, який можна використовувати для створення міжнародних веб-сайтів, локалізації програмного забезпечення, розробки багатомовних чат-ботів та міжнародного управління медіа. Amazon Comprehend автоматизує багато випадків використання NLP, які розглядаються в цій роботі, наприклад, для аналізу юридичних та медичних документів, виявлення фінансового шахрайства, масштабного аналізу мобільних додатків або позначення вмісту для аналізу та керування. І, нарешті, Amazon Lex може створити інтерфейс, схожий на людину, який може взаємодіяти за допомогою голосу та тексту, щоб задавати запитання, отримувати відповіді чи виконувати завдання, наприклад, створення інтерфейсів для управління запасами та програм продажів, інтерактивної допомоги, обслуговування клієнтів або запитів до баз даних людською мовою.

У третьому розділі було розглянуто синтез моделі середовища для роботи чат-бота з обробкою природньої мови на основі Amazon Lex. Виконано наступні кроки високого рівня: створення і тестування бота за допомогою плану зустрічі Amazon Lex ScheduleAppointment; створення і тестування функцію AWS Lambda для виконання завдань ініціації, перевірки та виконання; налаштування наміру MakeAppointment для використання функції Lambda як гачка коду. Кожен бот, який створюється в Amazon Lex, має намір. Намір – це дія, яку виконує бот.

Четвертий розділ присвячений тестуванню моделі для чат-бота, в тому числі, за допомогою веб-сторінки розміщеної на Amazon S3. Було зібрано Amazon Lex та Lambda в одне середовище, перевірено бота, для доведення, що він виконує лямбда-функцію. Далі, відбулося тестування публікації бота, щоб можна було перевірити його виклик з веб-сторінки. Найпростіший спосіб провести тест – створити статичну веб-сторінку та розмістити її на Amazon S3. Перевірено виклик API Amazon Lex з веб-сторінки для завантаження нашого бота, завдяки чому відвідувачі веб-сторінки змогли взаємодіяти з нею. Було протестовано налаштування пулу ідентифікаційних даних Amazon Cognito та

створено відповідні ролі (одна роль закінчується на Auth_Role, а інша на Unauth_Role), для забезпечення безпеки на цій сторінці.

Після чого веб-сторінці, на якій розміщено нашого бота, потрібен дозвіл на доступ до Amazon Lex. Для цього двом ролям, створеним у пулі ідентифікаційних даних Amazon Cognito, потрібні дозволи на доступ до Amazon Lex. Перевірили налаштування AWS Identity and Access Management (IAM) та надали відповідні дозволи ролям пулу ідентифікаційних даних. Після налаштування дозволів безпеки і створення сегмента S3 для розміщення нашої веб-сторінки, відбулося тестування оновленого демонстраційного файлу HTML, і перевірка використання пулу ідентифікаційних даних Amazon Cognito. У фінальному тестуванні перевірили взаємодію зі створеним в магістерській роботі ботом. На сторінці з розділом Статичний хостинг веб-сайтів за вибраним URL-адресом відкривається створена веб-сторінка.

СПИСОК ЛІТЕРАТУРИ

- [1] What exactly is Alexa? Where does she come from? And how does she work? [Online]. Available: <https://www.digitaltrends.com/home/what-is-amazons-alexa-and-what-can-it-do/>
- [2] Sadavarte, Sanket Sanjay, and Eliane Bodanese, “Pregnancy Companion Chatbot Using Alexa and Amazon Web Services”, IEEE Pune Section International Conference (PuneCon), pp. 1- 5. IEEE, 2019.
- [3] Gobinda G. Chowdhury. “Natural Language Processing”. In: Annual Review of Information Science and Technology, 2003. – С. 51-89.
- [4] J. Le. (2018). “The 7 NLP Techniques That Will Change How You Communicate in the Future (Part I)” [Online]. Available: <https://heartbeat.fritz.ai/the-7-nlp-techniques-that-will-change-how-youcommunicate-in-the-future-part-i-f0114b2f0497/>
- [5] Ted Briscoe. Introduction to Linguistics for Natural Language Processing. Tech. rep. 2013, pp. 1–37.
- [6] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu. Deep Learning Based Text Classification: A Comprehensive Review // 2005.
- [7] NLTK Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.nltk.org/>
- [8] Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.
- [9] Jurafsky D, Martin J H (2002) Speech and Language Processing – An Intro to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson Education Asia, ISBN 81-7808-594-1
- [10] R. Socher, “Recursive deep learning for natural language processing and computer vision” Stanford University, 2014, pp. 8-120.
- [11] Daniel Jurafsky. Classification: Naive Bayes, Logistic Regression, Sentiment /

- Daniel Jurafsky, James H. Martin // *Speech and Language Processing* / Daniel Jurafsky, James H. Martin., 2015. – С. 1–28.
- [12] Бенгфорт Бенджамин, Билбро Ребекка, Охеда Тони Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка. – СПб.: Питер, 2019. – 368 с.
- [13] Хобсон Лейн, Ханнес Хапке, Коул Ховард Обработка естественного языка в действии. – СПб.: Питер, 2020. – 576 с
- [14] Manaswi N.K. (2018) Developing Chatbots. In: *Deep Learning with Applications Using Python*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3516-4_11
- [15] What is Amazon Transcribe? [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.aws.amazon.com/transcribe/latest/dg/transcribe-what-is.html>
- [16] Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. Автоматическая обработка текстов на естественном языке и анализ данных: навч. посіб. – М.: НИУ ВШЭ, 2017. – 269 с.
- [17] What is Amazon Polly? [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.aws.amazon.com/polly/latest/dg/what-is.html>
- [18] What is Amazon Translate? [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.aws.amazon.com/translate/latest/dg/what-is.html>
- [19] Klopfenstein, L., Delpriori, S., Malatini, S., Bogliolo, A.: The rise of bots: a survey of conversational interfaces, patterns, and paradigms. In: *Proceedings of the 2017 Conference on Designing Interactive Systems*, pp. 555–565. Association for Computing Machinery (2017).
- [20] Poibeau T., Saggion H., Piskorski J., Yangarber R. Multi-source, Multilingual Information Extraction and Summarization. *Theory and Applications of Natural Language Processing*. Springer, Berlin, Heidelberg, 2013. – 257 с.
- [21] What is Amazon Comprehend? [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.aws.amazon.com/comprehend/latest/dg/what-is.html>
- [22] M. Bates (1995). “Models of natural language understanding” [Online].

Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC40721/>

- [23] What is Amazon Lex? [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.aws.amazon.com/lex/latest/dg/what-is.html>
- [24] Abdul-Kader, S., & Woods, J. (2015). Survey on Chatbot Design Techniques in Speech Conversation Systems. *International Journal of Advanced Computer Science and Applications*, 6(7). <http://doi.org/10.14569/ijacsa.2015.060712>
- [25] J. Chai and J.Lin, “The role of natural language conversational interface in online sales: a case study,” *International Journal of Speech Technology.*, vol. 4, pp. 285–295, Nov. 2001.