

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ Інформаційних управляючих систем \_\_\_\_\_  
(повна назва)

**АТЕСТАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Дослідження методів розробки бот систем  
(тема)

Виконав:

студент 2 курсу, групи ІУСТм-18-1

Воловик С. М.

(прізвище, ініціали)

Спеціальність 122 Комп'ютерні  
науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі  
системи та технології

(повна назва освітньої програми)

Керівник д.т.н. проф. Левикін В.М.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

\_\_\_\_\_  
(підпис)

Петров К.Е.

(прізвище, ініціали)

2019р.

# Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
(повна назва)  
Кафедра \_\_\_\_\_ Інформаційних управляючих систем \_\_\_\_\_  
(повна назва)  
Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
Спеціальність \_\_\_\_\_ 122 – Комп'ютерні науки \_\_\_\_\_  
(код і повна назва)  
Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)  
Освітня програма \_\_\_\_\_ Інформаційні управляючі системи та технології \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2019 р.

## ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

Студенту \_\_\_\_\_ Воловику Сергію Миколайовичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження методів розробки бот-систем  
затверджена наказом університету від 31 жовтня 2019 р.
2. Термін подання студентом роботи до екзаменаційної комісії 17 грудня 2019 р.
3. Вихідні дані до роботи Науково-технічні публікації та інтернет джерела з тематики атестаційної роботи
4. Перелік питань, що потрібно опрацювати в роботі Вступ, Дослідження чат-ботів і постановка задачі, Дослідження чат-ботів, Дослідження сучасних бот-платформ, Постановка задачі дослідження, Дослідження класифікацій та можливих варіантів розробки чат-ботів, Дослідження класифікацій чат-ботів, Дослідження сервісів для створення ботів, Дослідження фреймворків для створення чат-бота, Складність створення чат-бота, Недоліки аналогів чат-ботів, Алгоритм розробки чат-бота, Дослідження штучної нейронної мережі та методів машинного навчання, їх застосування в бот розробці, Дослідження штучної нейронної мережі, Архітектура нейронних мереж, Навчання нейронної мережі, Глибоке навчання при розробці чат-бота, Дослідження моделі Послідовність до послідовності (Seq2Seq), Дослідження моделі Слово у вектор (Word2Vec), Рекомендації по використанню методу машинного навчання для вдосконалення роботи чат-бота, Розробка прикладної програми coffee bar чат-бота на основі розпізнавання ключових слів, меню / кнопок та контексту, Розпізнавання ключових слів, меню / кнопок та контексту, Інструменти та додаткові засоби для розробки програми, Побудова прикладної програми COFFEE BAR BOT, Висновки до розділу, Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина			

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз літератури та Інтернет-джерел	04.11.2019	
2	Постановка задачі	05.11.2019	
3	Обробка матеріалу	10.11.2019	
4	Дослідження чат-ботів	15.11.2019	
5	Дослідження класифікацій та можливих варіантів розробки чат-ботів	18.11.2019	
6	Дослідження штучної нейронної мережі та методів машинного навчання, їх застосування в бот розробці	22.11.2019	
7	Розробка прикладної програми coffee bar чат- бота на основі розпізнавання ключових слів, меню / кнопок та контексту	26.11.2019	
8	Написання пояснювальної записки	28.11.2019	
9	Підготовка презентації	09.12.2019	
10	Перевірка на плагіат	10.12.2019	
11	Нормоконтроль	10.12.2019	
12	Захист	17.12.2019	

Дата видачі завдання 4 листопада 2019 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ д.т.н. проф. Левикін В.М.  
(підпис) (посада, прізвище, ініціали)

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....	8
ВСТУП.....	9
1 ДОСЛІДЖЕННЯ ЧАТ-БОТІВ І ПОСТАНОВКА ЗАДАЧІ.....	11
1.1 Дослідження чат-ботів .....	11
1.2 Дослідження сучасних бот-платформ .....	15
1.3 Постановка задачі дослідження.....	17
2 ДОСЛІДЖЕННЯ КЛАСИФІКАЦІЙ ТА МОЖЛИВИХ ВАРІАНТІВ РОЗРОБКИ ЧАТ-БОТІВ .....	18
2.1 Дослідження класифікацій чат-ботів .....	18
2.2 Дослідження сервісів для створення ботів .....	22
2.3 Дослідження фреймворків для створення чат-бота.....	29
2.4 Складність створення чат-бота.....	31
2.5 Недоліки аналогів чат-ботів.....	33
2.6 Дослідження типового алгоритма розробки чат-бота .....	35
3 ДОСЛІДЖЕННЯ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ ТА МЕТОДІВ МАШИННОГО НАВЧАННЯ, ЇХ ЗАСТОСУВАННЯ В БОТ РОЗРОБЦІ .....	37
3.1 Дослідження штучної нейронної мережі .....	37
3.2 Архітектура нейронних мереж .....	41
3.4 Навчання нейронної мережі.....	42
3.5 Глибоке навчання при розробці чат-бота .....	46
3.6 Дослідження моделі Послідовність до послідовності (Seq2Seq).....	47
3.7 Дослідження моделі Слово у вектор (Word2Vec) .....	49
3.8 Рекомендації по використанню методу машинного навчання для вдосконалення роботи чат-бота.....	51

4 РОЗРОБКА ПРИКЛАДНОЇ ПРОГРАМИ COFFEE BAR ЧАТ-БОТА НА ОСНОВІ РОЗПІЗНАВАННЯ КЛЮЧОВИХ СЛІВ, МЕНЮ / КНОПОК ТА КОНТЕКСТУ .....	56
4.1 Розпізнавання ключових слів, меню / кнопок та контексту .....	56
4.2 Інструменти та додаткові засоби для розробки моделі .....	57
4.3 Побудова прикладної програми COFFEE BAR BOT .....	59
4.4 Висновки до розділу.....	63
ВИСНОВКИ .....	65
ПЕРЕЛІК ПОСИЛАНЬ .....	67
ДОДАТОК А .....	70
ДОДАТОК Б .....	71
ДОДАТОК В.....	73

## РЕФЕРАТ

Пояснювальна записка до атестаційної роботи включає: 90 сторінки, 20 рисунків, 2 таблиці, 28 джерел, 3 додатки.

АНАЛІЗ, БОТИ, ДОСЛІДЖЕННЯ, ОРГАНІЗАЦІЯ, КОРИСТУВАЧ, МЕССЕНДЖЕР, ПЛАТФОРМА, СЕРВІС.

Об'єктом дослідження є методи розробки бот-систем.

Метою роботи є проведення дослідження популярності сучасних платформ, месенджерів, визначення програмної платформи (фреймворк) та інтерфейс прикладних рішень і бібліотек API з великою кількістю доступних інструментів (функцій, класів, структур), які підходять для сучасної розробки бота.

В ході дослідження отримані такі результати – була проведена постановка задачі, що включає в себе:

- дослідити існуючі підходи для розробки бота;
- дослідити існуючі проблеми чат-ботів;
- визначення програмної платформи, а саме фреймворків та інтерфейс прикладних рішень і бібліотек API з великою кількістю доступних інструментів: функцій, класів, структур, які підходять для сучасної розробки бота;
- дослідити методи машинного навчання для вдосконалення ботів, та їх взаємодію з користувачем;
- розробка прикладної програми.

## ABSTRACT

Explanatory note to the attestation work includes: 90 pages,  
20 images, 2 tables, 28 sources, 3 applications.

ANALYSIS, BOTS, RESEARCH, ORGANIZATION, USER,  
MESSENGER, PLATFORM, SERVICE.

The object of the study is the methods of developing bot systems.

The purpose of the work is to analyze the popularity of modern platforms, messengers, determine the software platform (framework) and the interface of application solutions and API libraries with many available tools (functions, classes, structures) that are suitable for modern development of the bot.

In the course of the study the following results were obtained - the formulation of the research task was performed, which includes:

- investigate existing approaches for bot development;
- investigate existing problems of chatbots;
- definition of software platform, namely frameworks and interface of application solutions and API libraries with many available tools: functions, classes, structures that are suitable for modern development of the bot;
- investigate machine learning methods to improve bots, and their interaction with the user;
- development of an application program.

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БЗ – база даних

ПЗ – програмне забезпечення

ШІ – штучний інтелект

AIML – Artificial Intelligence Markup Language (Мова розмітки для штучного інтелекту)

A.L.I.C.E – Artificial Linguistic Internet Computer Entity (Штучно лінгвістична інтернет-комп'ютерна істота)

API – Application Programming Interface (Прикладний програмний інтерфейс)

B2B – Business to business (Бізнес для бізнесу)

CRM – Customer Relationship Management (Система управління взаємовідносинами з клієнтами)

HTTP – HyperText Transfer Protocol (Протокол передачі гіпертексту)

HTTPS – HyperText Transfer Protocol Secure (Захищений протокол передачі гіпертексту)

JSON – JavaScript Object Notation (Текстовий формат обміну даними між комп'ютерами)

NLP – Natural Language Processing (Нейролінгвістичне програмування)

XML – Extensible Markup Language (Розширювана мова розмітки)



## ВСТУП

В наш час технологій не стоять на одному місці: розумний дім, електрокари, хмарні технології, 3D друк, постійно з'являються нові технології та тренди, деякі з яких здатні серйозно вплинути на всю сферу ІТ. Відносно недавно всі говорили лише про додатки, але останнім часом набирає обертів розробка ботів, або чат-ботів скорочено від «робот-чат», яким пророкують велике майбутнє в комунікаціях та маркетингу.

То чому боти є актуальним, і навіть потрібними в наш час? Перш за все це дуже вигідно для бізнесу. Саме завдяки ботам можна оптимізувати витрати та зменшити навантаження на робочий персонал, їхня здатність виконувати понад вісімдесят відсотків задач, які існують у взаємодії між організацією та клієнтом, чи всередині компанії для підвищення якості послуг. По мірі того як боти стають розумнішими, задачі можуть ставати складнішими, і боти мають можливість навчатися, виконуючи більш складну роботу.

Термін бот походить від слова «робот». Скорочена назва вказує на сутність поняття – спеціальної утиліти, розробленої для виконання дій по заданому алгоритму і через призначені для людей інтерфейси. Хто такі боти в інтернеті? Це своєрідні імітатори дій живих людей, і іноді вони настільки складні, що їх неможливо відрізнити від реального користувача. Отримуючи певний запит, програми діють за заздалегідь продуманим сценарієм, правила якого незмінні.

Розумний бот призначений для виконання одноманітної і рекурсивної роботи. У порівнянні з людьми, він має прискорену реакцію на процеси і не втомлюється від монотонних дій. Використання ботів необхідно для виконання наступного виду операцій: однотипних, чергуються один з одним, швидких – з їх виконанням людина часто не здатна впоратися. Програма нового століття, яка несе безліч корисних функцій і допомагає людям в їх

роботі. Корисні утиліти всюди. Залежно від інтерфейсу, вони можуть бути різними. Основні причини, для чого потрібні боти: для виконання розважальних, пізнавальних, комерційних дій. Великого поширення сьогодні отримали чат-боти, які розмовляють з людиною на зрозумілій йому мові. На питання клієнтів бот дає заздалегідь запрограмовані відповіді. Такі утиліти активно використовуються в бізнесі: страхових компаніях, на інтернет-сторінках банків, інтернет-магазинах, лізингових компаніях та ін.

Об'єктом дослідження є методи розробки бот-систем.

Метою даної роботи є дослідження методів розробки бот систем. Для досягнення визначеної мети в магістерській атестаційній роботі пропонується вирішити такі задачі:

- дослідити існуючі підходи для розробки бота;
- дослідити існуючі проблеми чат-ботів;
- визначення програмної платформи, а саме фреймворків та інтерфейс прикладних рішень і бібліотек API з великою кількістю доступних інструментів: функцій, класів, структур, які підходять для сучасної розробки бота;
- дослідити методи машинного навчання для вдосконалення ботів, та їх взаємодію з користувачем
- розробка прикладної програми.

Робота виконана згідно з вимогами стандарту ДСТУ 3008-2016 [1] та методичними вказівками 2019 року щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 Комп'ютерні науки та освітньою програмою «Інформаційні управляючі системи та технології» [2].

## 1 ДОСЛІДЖЕННЯ ЧАТ-БОТІВ І ПОСТАНОВКА ЗАДАЧІ

### 1.1 Дослідження чат-ботів

Першими відомостями про бота можна вважати експеримент – а саме тест Тьюринга, опублікованого в 1950 році, який зводиться до того, що штучним інтелектом можна визнати програму, здатну вести бесіду подібну людині. У 1966 році професор МІТ Джозеф Вейценбаум написав програму ELIZA. Вона наслідувала мови стереотипного психотерапевта, постійно відповідаючи на репліки співрозмовника-людини зустрічними питаннями. І хоча спілкування було ілюзією, причому примітивною, Вайзенбаум був вражений тим, наскільки люди захоплювалися бесідою [3].

До 1990 року деревовидні набори правил, що лежали в основі ELIZA і інших подібних програм, стали настільки опрацьовані і складні, що тест Тьюринга з філософської концепції перетворився в реально проведене випробування. Була заснована щорічна премія AI Loebner. Тоді з'явилося і саме поняття «чат-бот». Його прийнято пов'язувати з Julia – електронної помічницею, розробленої Майклом молдинги в 1994 році. Джулія справлялася з імітацією спілкування набагато краще попередників, але все ще використовувала для вибору відповідних реплік ключові слова. Через рік був представлений бот A.L.I.C.E. (Artificial Linguistic Internet Computer Entity), який формулює відповідні фрази за допомогою аналізу евристичних шаблонів. Спілкування з A.L.I.C.E. вже нагадувало повноцінний діалог. Програма так і не пройшла тест Тьюринга, але вона три рази (у 2000, 2001, 2004 роках) визнавалася кращим чат-ботом конкурсу AI Loebner. Побити цей рекорд вдалося Mitsuku, розробці британця Стіва Уорсвіка, лише в 2018 році.

У 2006 році компанія IBM почала розробку суперкомп'ютера Watson, що володіє енциклопедичними знаннями і міг би давати відповіді на питання, задані вголос. Вже через чотири роки подібні рішення стали доступними для широкого загалу. Apple представила голосового асистента Siri (Speech

Interpretation and Recognition Interface), а потім з'явилися Google Now, Alexa від Amazon, Microsoft Cortana і Яндекс Аліса [3].

У той же час принципи і технології машинного навчання, що лежать в основі голосових помічників, перестали бути прерогативою корпорацій. З їх допомогою класичні чат-боти значно порозумнішали, стали досить надійні для комерційного використання і стали популярними у користувачів.

Якщо говорити про актуальність, то нам потрібно лише подивитися на статистику останніх 5 років на (рисунок 1.1).

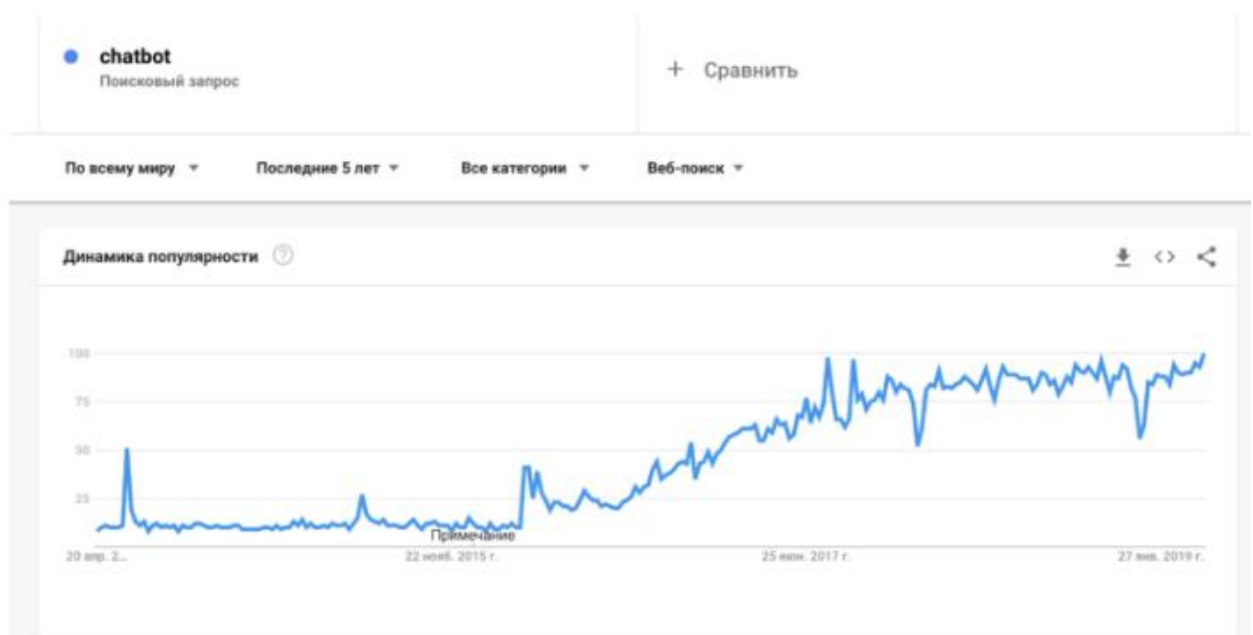


Рисунок 1.1 – Динаміка популярності чат-ботів на сьогодні (Джерело: Google Trends)

Сьогодні боти стають фактичним стандартом інтерфейсу для взаємодії з програмними сервісами. Це відбувається внаслідок широкого поширення платформ обміну повідомленнями (наприклад, Facebook Messenger для користувачів соціальної мережі і Slack для розробників), а частково завдяки прогресу в засобах розуміння природної мови, які підтримуються багатьма ботами. Ще одна рушійна сила – масове використання великих даних і алгоритмів машинного навчання: боти зручні як для користувача інтерфейс

для взаємодії з системами, що видають відповіді за результатами аналізу гігантських обсягів відомостей [4].

У великих софтверних компаніях визнають зручність ботів для інтеграції сервісів, каналів зв'язку і об'єднання користувачів. У Facebook, наприклад, хочуть поступово замінити ботами на платформі обміну повідомленнями Messenger всі програми, а в Microsoft заявляють, що операційна система майбутнього – це «діалог як платформа».

Про зростання популярності програм-роботів можна судити з розвитку сервісів на зразок Alexa, Siri, IBM Watson і Google Now. Існує також безліч ботів на платформах, які використовуються розробниками ПЗ для зв'язку з колегами і сервісами, таких як Slack, Microsoft Teams і HipChat [4].

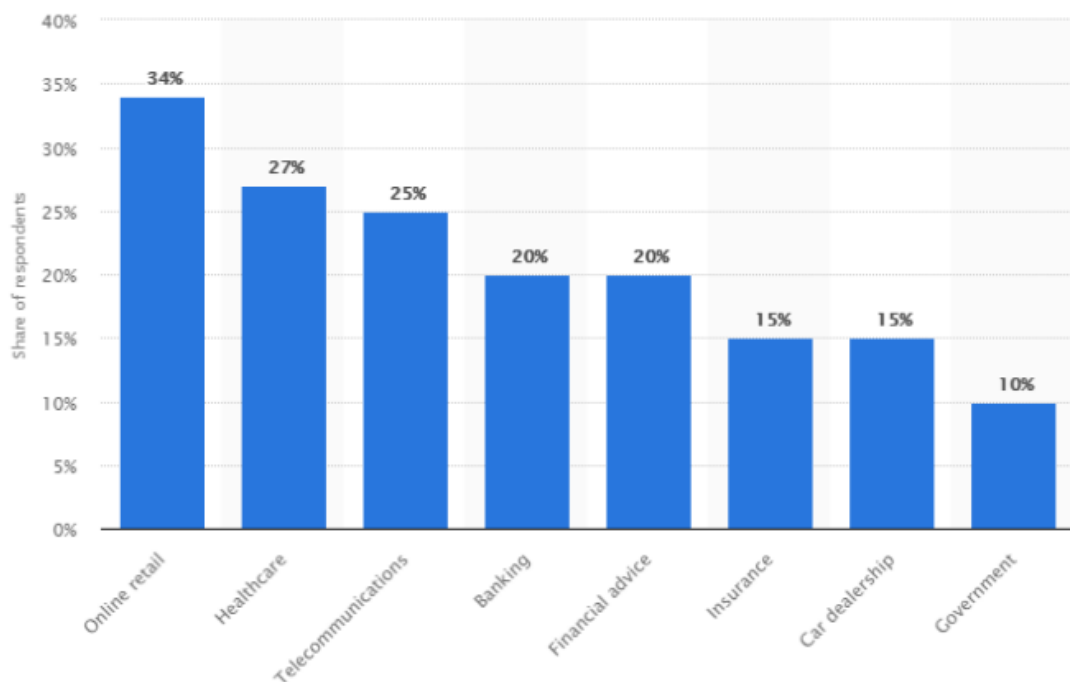


Рисунок 1.2 – Данні опитування Statista на 2017 рік [4]

За даними всесвітнього опитування Statista 2017 року, 34% респондентів заявили, що вони краще б відповідали на питання ШІ за допомогою чат-бота або віртуального помічника у випадку використання електронної комерції (рисунок 1.2).

Сьогодні активно створюються боти для масового користувача, багато часу приділяє програмам обміну повідомленнями. Такі боти нерідко стають альтернативою традиційним мобільних додатків.

Боти стають повсюдними – ми взаємодіємо з ними в автомобілі, вдома, на роботі і не тільки. Крім того, вони грають складну і все більш помітну роль в проектах розробки ПЗ. Накопичений на сьогодні досвід застосування ботів дозволяє оцінити як їх реальні практичні переваги, так і можливі мінуси. Перерахуємо принципи, на які варто орієнтуватися при створенні і використанні спамерських пошукових роботів [4].

Прості боти можна створювати з нуля і розміщувати на власних ресурсах, але коли потрібно спростити розробку та дистрибуцію, використовуються сторонні платформи. Кількість інструментів для розробки ботів швидко зростає, але є відмінність між платформами створення програм-роботів і платформами дистрибуції, на яких вони виконуються.

Такі компанії як Microsoft, Facebook, Telegram і інші виробники надають комплексні засоби розробки і дистрибуції програм-роботів. Інші компанії пропонують спеціалізовані ресурси для конкретних завдань, пов'язаних зі створенням і поширенням.

Платформи розробки ботів надають різні програмні засоби, фреймворки, інструментарії, інтерфейси програмування і додаткові можливості (сервіси обробки природної мови, пошуку та аналізу зображень та ін.). Вони можуть бути орієнтовані на конкретну платформу дистрибуції або розраховані на створення ботів, яких можна розгортати на багатьох платформах, зокрема Microsoft Bot Framework, Botkit і Pandorabots. Надані сервіси варіюються від документації і шаблонів коду до інтерфейсів створення ботів без програмування (наприклад Chatfuel). Навколо багатьох популярних платформ розробки сформувалися активні спільноти розробників, учасники яких можуть спілкуватися в відповідних онлайн-групах, обмінюючись досвідом і посиланнями на тематичні публікації, отримуючи консультації та відповіді на

питання, беручи участь в обговореннях. Деякі з таких спільнот, наприклад група Botmaker в Slack і форум Chatbot Magazine, стали осередком дискусій з широкого кола тем, що мають відношення до програм-роботів.

## 1.2 Дослідження сучасних бот-платформ

Якщо говорити про платформи, на яких використовують ботів, то більший відсоток належить саме месенджерам.

Що ж таке месенджер? Найчастіше, під поняттям месенджер мається на увазі програма або мобільний додаток, в якому можна обмінюватися повідомленнями з іншими користувачами. Перші месенджери з'явилися ще в 90-их роках, однак більшу популярність вони почали набирати тільки в еру пост ПК. Стрімкий розвиток мобільних пристроїв дозволило зробити їх доступними для людей з різним рівнем доходу. Зараз майже у кожної людини є смартфон під рукою. Найпопулярніші месенджери в Україні це WhatsApp, Viber і Telegram [4]. Розвиток ботів в месенджерах почався після того, як Telegram відкрив платформу для створення спеціальних акаунтів, обробних і відправляють повідомлення. На даний момент половина створених ботів засноване на Telegram. Зручність використання полягає в тому, що перш за все вони доступні як в самому месенджері, так і в соціальних мережах. А це означає, що в мобільних пристроях вони на 1 крок ближче до користувача, ніж мобільний додаток, і на 2 ніж веб-сайт. Більше не потрібно звертатися в різні джерела даних, щоб вирішити елементарну задачу [4].

Використання бота на даних платформах досить легке, потрібно лише вибрати його з каталогу, перейти за посиланням або знайти його по імені через пошук і вступити з ним в переписку. Для запуску бота вводиться команда /start (розпочати), або пропонується натиснути кнопку «старт» на віртуальній клавіатурі. Для роботи можуть використовуватись програмовані кнопки віртуальної клавіатури, за рахунок яких бот перетворюється в міні-додаток з інтуїтивно зрозумілим інтерфейсом. Далі бот відправить вам інформації про

себе, інструкції, а також список або виведе доступні команди або кнопки на екран девайсу.

Що ж стосується безпеки, то в рамках самого месенджера, боти не можуть зробити нічого поганого, тому що по своїй суті це ті ж акаунти користувачів, тільки управляються алгоритмічно. Як і звичайного користувача, обридлого бота можна видалити і заблокувати. Важливо, що бот не виступає ініціатором бесіди – це ви перший відправляєте йому запит, тому без вашого дозволу бот не може вас потурбувати.

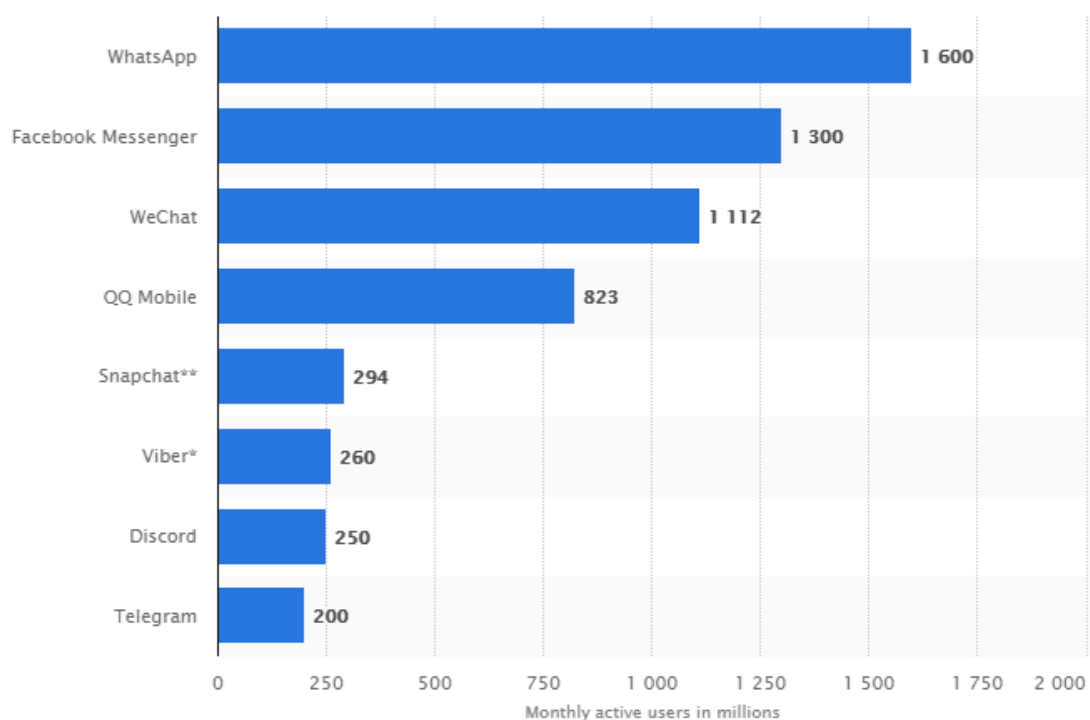


Рисунок 1.3 – Кількість користувачів в месенджерах за місяць [5]

Більшість свого часу люди проводять саме в месенджерах, підтвердження цьому є статистика активних користувачів за Липень 2019 року, взята з сайту statista.com (рисунок 1.3).

Звісно, потрібно розуміти, що всі ці платформи досить різні, але вони мають дещо спільне. Наприклад, досить легкий та інтуїтивно зрозумілий інтерфейс. У кожному із цього списку месенджері можна створити групові бесіди, виконувати аудіо та відео дзвінки, відправляти голосові повідомлення.



Згідно цієї статистики, можна впевнено говорити, що дані сервіси є важливою складовою взаємодії користувачів інтернет мережі.

### 1.3 Постановка задачі дослідження

Об'єктом дослідження є методи розробки бот-систем.

Метою роботи є проведення дослідження популярності сучасних платформ, месенджерів, визначення програмної платформи (фреймворк) та інтерфейс прикладних рішень і бібліотек API з великою кількістю доступних інструментів (функцій, класів, структур), які підходять для сучасної розробки бота.

В ході дослідження отримані такі результати – була проведена постановка задачі, що включає в себе:

- дослідити існуючі підходи для розробки бота;
- дослідити існуючі проблеми чат-ботів;
- визначення програмної платформи, а саме фреймворків та інтерфейс прикладних рішень і бібліотек API з великою кількістю доступних інструментів: функцій, класів, структур, які підходять для сучасної розробки бота;
- дослідити методи машинного навчання для вдосконалення ботів, та їх взаємодію з користувачем
- розробка прикладної програми.

## 2 ДОСЛІДЖЕННЯ КЛАСИФІКАЦІЙ ТА МОЖЛИВИХ ВАРІАНТІВ РОЗРОБКИ ЧАТ-БОТІВ

### 2.1 Дослідження класифікацій чат-ботів

Незважаючи на величезну кількість існуючих чат-ботів, їх усіх можна розділити на дві великі категорії в залежності від технології, що використовується для ведення інтелектуального діалогу: програми з обробкою текстів на природних мовах NLP (Natural Language Processing) та програми, засновані на формуваннях міркувань на основі конкретних випадків CBR (Cased-Based Reasoning).

Чат-боти з технологією обробки текстів на природних мовах починають аналіз реплік користувача з синтаксичного розбору, в результаті проведення якого є розбиття пропозицій на знайомі програмі нетермінальні і термінальні символи. Чат-боти з технологією NLP не мають бази даних з чітко заданими шаблонами відповідей, реакція на введення користувачем фрази кожен раз синтезується на підставі правил використовуваної граматики, структури раніше введеного користувачем тексту, а також знайдених в ньому ключових слів. Деякі з таких роботів не мають спочатку закладених граматик, а застосовують індуктивний метод виведення граматики, що дозволяє навчатися під час діалогу з користувачем і «налаштовуватися» під його манери.

Найвідомішим представником програм з формуванням міркувань [6] на основі конкретних випадків є програма A.L.I.C.E., створена в кінці 90-х рр. Р. Воллесем. Для написання A.L.I.C.E. була використана та спеціально розроблена мова програмування AIML. AIML є модифікованою версією мови XML.

Метою розробки мови AIML було забезпечення функцій, необхідних для вилучення та обробки знань, відповідних заданій шаблонної структури, а також

формування вихідних сигналів за схемою «стимул реакція». В даний час мова AIML, його специфікації і інтерпретатори для перекладу тексту програм на стандартні об'єктно-орієнтовані мови програмування безкоштовно поширюються на офіційному сайті A.L.I.C.E. AI Foundation. Порівняльна простота використовуваних технологій (A.L.I.C.E. заснована на принципі мінімалізму: «великий обсяг даних, маленький програмний код») і доступність інформації послужили причиною величезної популярності і широкого поширення ботів даного сімейства, а використовуємо принцип відкритих вихідних текстів дозволив програмістам впроваджувати елементи даної програми в процесі розробки своїх власних додатків, а також вносити свій вклад в розвиток і вдосконалення мови AIML. Для розширення бази знань A.L.I.C.E. застосовуються напіваавтоматичні методи [6].

У програмі закладені так звані, відповіді «за замовчуванням», застосовувані в тому випадку, якщо система не знайшла точно відповідного шаблону, або такі відповіді носять загальний характер і мають на меті змінити тему розмови на знайому програму. Успішність подібних маневрів залежить тільки від винахідливості розробників і від впертості користувача. Якщо відповіді, використовувати стандартну складені з урахуванням психологічних аспектів ведення діалогу і не повторюються занадто часто, то є ймовірність, що користувач навіть не помітить зміни теми розмови і тим більше не розпізнає це як «слабке місце» системи. У той же час програма зберігає всі репліки користувача, у відповідь на які були використані репліки «за замовчуванням», таким чином розробники чат-бота отримують доступ до логів розмов і можуть легко розширювати і вдосконалювати базу знань програми. Мова програмування AIML також дозволяє вбудовувати команди, написані на іншими мовами, таких як, наприклад, Javascript або C ++, що істотно збільшує область застосування програми і дозволяє вбудовувати її в самі різні додатки. Чат-боти сімейства A.L.I.C.E. можуть підтримувати діалог з декількома користувачами одночасно і мають структуру клієнт-сервер.

Нині спостерігається тенденція використання природно-мовних інтерфейсів, заснованих на технологіях чат-ботів, для практичних додатків: в центрах телефонного обслуговування; для автоматичних відповідей на email-запити; для доступу до баз даних і видачі запитуваної інформації віддаленим користувачам; для надання банківських послуг по телефону та ін. Однак застосування даних програм в системах електронної комерції поки що вкрай обмежена, і на даному етапі ці програми використовуються тільки в якості дослідних прототипів, які не мають широкого поширення.

Що до видів ботів, то на даний час є не офіційна класифікація: бот-автоматизатор, бот помічник, вбудований бот, бот який допомагає оформити замовлення, а також досить цікавий бот-збирач лідів.

Бот-автоматизатор – виконує за користувача кілька рутинних дій в автоматичному режимі, ніж економить йому час. Корисний для таких завдань, як відстеження посилок або розкладу транспорту, замовлення таксі, автоматична відправка листів за результатами подій в CRM [7].

Бот-помічник – вміє розпізнавати запити від клієнтів і давати на них відповіді, чи не залучаючи людини в процес. Якщо впирається в стелю своїх знань, перемикає клієнта на менеджера. Менеджер в свою чергу отримує повну історію звернення – з озвученими клієнтом персональними даними, питанням, історією спілкування. Такий бот «розвантажує» менеджерів, позбавляючи від рутини. Самонавчального, тому з часом може закривати все більше складних завдань. Може використовуватися, наприклад, службами підтримки для відповіді на найпопулярніші запити.

Вбудований бот – бот з елементами ШІ, йому можна «згодувати» великий обсяг даних, в якому він буде шукати інформацію краще і швидше людини. Корисний там, де потрібно орієнтуватися у величезних обсягах даних, наприклад, в документах страхових або юридичних фірм [8].

Бот, який допомагає оформити замовлення – цей бот може відшукати потрібний товар і оформити замовлення через інтерфейс месенджера швидше,

ніж це можна було б зробити через сайт. Економить час користувача за рахунок того, що виконує максимальну кількість завдань через інтерфейс чату, і перекидає в браузер для завершення оформлення замовлення (авторизації або оплати) автоматично, позбавляючи від необхідності щось шукати вручну. Корисний сервісів бронювання житла, квитків і т.д..

Бот-збирач лідів – збирає дані про клієнтів, зацікавлених в замовленні, уточнює у них персональні дані та інформацію про цікавить товар / послугу. Також береже менеджерів компанії від рутинної роботи. Корисний всюди, де потрібно збирати досить однотипні дані в користувачів.

Боти часто використовуються в робочих групах для автоматизації вирішення завдань, що зазвичай вимагають участі людини. Але повна відмова від взаємодії може обмежити творчі можливості і продуктивність людей. Тому ботів переважно використовувати для сприяння контактам і координації виконання робіт. Наприклад, боти можуть інформувати про хід виконання завдання, сповіщати про цілі групи, пов'язувати експертів з новачками і служити справі зміцнення довіри і командного духу. При цьому неякісна реалізація або надмірне використання ботів можуть привести до інформаційного перевантаження або зайвого кількості переривань.

В ідеалі взаємодія користувача з ботом має бути комфортним і відбуватися без запинок. Досягти цього можна шляхом ретельного планування схеми бесіди, особливо для діалогових спамерських пошукових робіт. Зокрема, бот міг би повторювати вводяться команди, щоб користувач знав, що його «почули» і правильно зрозуміли. Боти також повинні бути здатні визначати, коли діалог зайшов у глухий кут, і надавати підказки щодо можливого продовження взаємодії [8].

У ботів можуть бути елементи графічного інтерфейсу, що дозволяють прискорити взаємодію. Передбачивши глобальну перевірку введення ключових слів, що відносяться до навігації («допомога», «назад», «скасування», «почати заново», «вихід» і т. п.), можна уникнути створення

занадто «нетямущого» бота. Інструменти начебто BotMock допоможуть в моделюванні взаємодії користувача з ботом. Крім того, на багатьох платформах доступні готові протоколи спілкування, яких повинні дотримуватися боти, наприклад, в Facebook є набір нескладних порад по організації діалогів. Чат-боти розрізняються між собою залежно від завдань, які вони повинні вирішувати.

Для комерційного чат-боту, який потрібний для роботи в певному діалоговому просторі, темі, сфері відповідальності чат-бота, немає необхідності в використанні технології машинного навчання [9]. Для чат-ботів такого типу використовують технології базової обробки мови:

- виділення ключових слів або іменованих сутностей;
- сегментація (це розділення на речення);
- стемінг / лематизація.

А також технології створення регулярних граматик по типу вхідний патер - це шаблон відповідної реакції, приклад з AIML:

- масив (змінні) для вивчення або запам'ятовування контексту;
- регулярні вирази;
- рекурсія;
- цикли та інше.

Якщо є множина статистичних даних, то краще створити складнішого чат-бота по технології машинного навчання, а саме на методі глибинних нейронних мереж.

## 2.2 Дослідження сервісів для створення ботів

Найпростіший варіант – інтерактивна форма, найбільш складний – програма зі штучним інтелектом, що імітує співрозмовника, вміє підтримати розмову на будь-яку тему. Але як же створити чат-бота? На це питання є відповідь – використати сервіс для створення ботів, або запрограмувати.

Що до першого варіанту, то доцільно буде виділити декілька досить зручних сервісів:

- Flow XO – конструктор чат-ботів для Slack, Facebook messenger, Telegram, що не вимагає знання коду. Сервіс – простий у використанні, з інтуїтивно зрозумілим візуальним редактором. Дозволяє інтегрувати помічника більш ніж на 90 різних програм. Підтримується більшістю відомих платформ. Можливість використовувати готові шаблони і фільтри. Підключення до поштових баз даних для організації розсилок.

- Chatfuel – дозволяє створити і запустити чат-бота всього за 3 кроки: створити, інтегрувати з бажаним додатком, запустити.

Є шаблони ботів для деяких сфер діяльності - магазинів, видавництв, ріелторських компаній. Створити бота можна всього за 7 хвилин. Почати роботу можна безкоштовно, основні функції будуть доступні. Платні підписки надають ще більше можливостей. Ціна коригується залежно від кількості користувачів бота.

Chatfuel повідомляє, що 46% всіх пошукових робіт Facebook Messenger створені саме в цьому сервісі.

Але головним конкурентом Flow XO, Chatfuel є Dialogflow від корпорації Google. Стандартний тариф (Standard Edition) - безкоштовний, він охоплює більшість потреб розробників. Є платний тариф для підприємств (Enterprise Edition), які розраховується виходячи з потреб підприємства. Для новачків сайт пропонує готові шаблони різних тематик подорожі, харчування, роздрібна торгівля, розваги, угоди, підтримка клієнтів, пошук і т.д.

На сайті є повноцінне керівництво в тому, як почати роботу і зробити свого першого робота. Все описано дуже докладно, відмінне підмогою для тих, хто поняття не має, як проектуються онлайн-помічники. Dialogflow вживає машинне навчання, щоб роботи могли розпізнавати мову користувачів і

навчатися в процесі спілкування з ними, для наглядного вигляду в таблиці 2.1 відображається переваги та недоліки Dialogflow.

Таблиця 2.1 – Переваги та недоліки Dialogflow

Переваги	Недоліки
<p>За допомогою Api.ai (Dialogflow) можна моделювати великі та складні потоки, використовуючи Інтенти та Контексти.</p> <p>Заповнення слотів – це інтегрована функція. Добру частину логіки може вирішити чат-бот, який зменшує кодування на стороні сервера.</p> <p>Доступні домени – це специфікації, які можуть вирішувати декілька випадків загального використання та застосувань</p> <p>Ви можете навчити чат-NLP з прикладами в розділі «Навчання» (у бета-версії).</p> <p>Інтеграція одним клацанням миші з кількома платформами, такими як Facebook Messenger, Slack, Twitter та Telegram.</p>	<p>Неможливо заблокувати відповідність наміру, якщо присутній контекст.</p> <p>Навчальний розділ все ще знаходиться в бета-версії.</p> <p>Навіть якщо історії є потужною концепцією, є випадки, коли важко контролювати потік розмови, а бот схильний неправильно розуміти запити користувачів.</p>

А для створення ботів без використання сервісу, використовують API платформи. Наприклад, Bot API представляє собою HTTP-інтерфейс для



роботи з ботами в Telegram. Кожен бот – це спеціальний аккаунт, створений для автоматичного оброблення та відправлення повідомлень.

Існує два протилежних за логікою способи отримання оновлень від бота: long pulling – додаток автоматично опитує сервера Telegram на наявність будь-яких оновлень для бота [10]. За замовчуванням 100мс; webhook – сервера Telegram самі сповіщають додаток на сервері як тільки з'являться будь-які оновлення. Вхідні оновлення будуть зберігатися на сервері до тих пір, поки їх не оброблять, але не довше 24 годин. Незалежно від способу отримання оновлення, у відповідь відправляється об'єкт Update, серіалізовані в JSON [11].

Всі запити до Telegram Bot API повинні здійснюватися через HTTPS в такому вигляді: [https://api.telegram.org/bot<token>/НАЗВА\\_МЕТОДА](https://api.telegram.org/bot<token>/НАЗВА_МЕТОДА) [12].

Щоб створити досить швидко примітивного бота в Telegram досить часто використовують команди, які подаються візуально у інтерфейсі і для їх відправки достатньо просто натиснути по ній на екрані. Загальноприйнятою концепцією є написання фіксованих команд боту використовуючи символ «/» як префікс. Наприклад «/help» - команда реалізована для багатьох ботів, яка зазвичай виводить список доступних команд чи іншу необхідну користувачу інформацію для початку роботи із ботом.

На рисунку 2.1 наведено приклад відповіді на таку команду ботом BotFather, який дозволяє створити бота для месенджера Telegram [13].

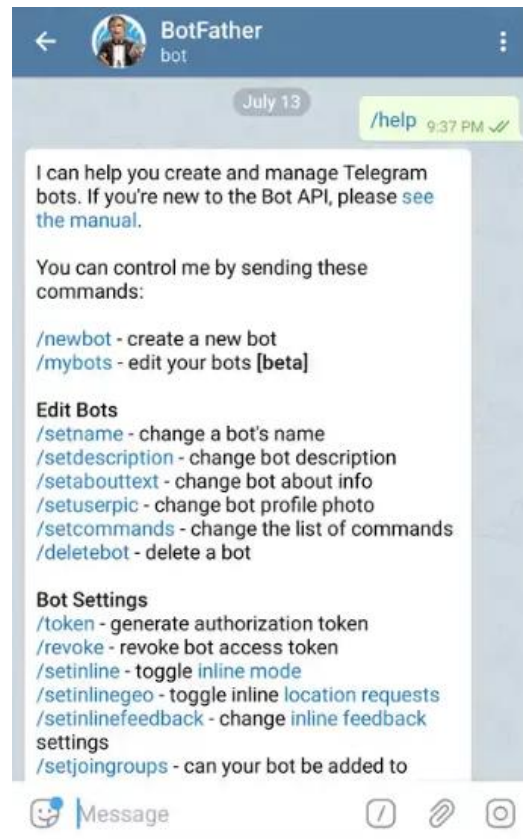


Рисунок 2.1 – Результат обробки команди «/help» ботом BotFather [13]

У месенджерах працює безліч чат-ботів для вирішення різноманітних завдань, на рисунках будуть відображені деякі з них .

Раурббот – бот «ПриватБанку» для переказу грошей в Telegram.

«Мої платежі» – бот для оплати комунальних послуг через Приватбанк в Facebook.

Зоряна – бот «Київстар» в Telegram, який консультує в питаннях зв'язку і допомагає визначитися з тарифами та послугами (рисунок 2.2).

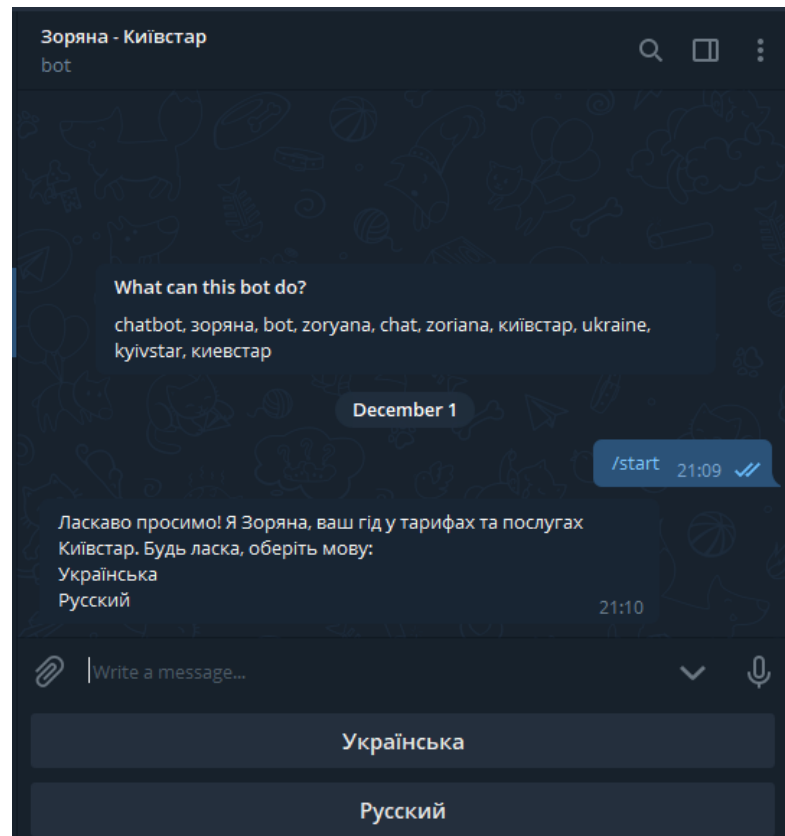


Рисунок 2.2 – Інтерфейс бота Зоряни компанії «Київстар» в Telegram

Whereisnrbot – неофіційний Telegram-бот, який відображає місце розташування посилки «Нова Пошта».

QTrackerBot – Telegram-бот, який відстежує посилки «Укрпошти», «Нова Пошта», «Делівері», «Міст-Експрес».

Paralegalbot – юридичний бот-реєстратор компаній (рисунок 2.3).

ParalegalBot є першим і єдиним рішенням, що дозволяє отримати документи для процесу без реєстрації і абсолютно безкоштовно. Віртуальний помічник

ParalegalBot дозволяє витратити менше часу на технічну частину написання заяв та клопотань, залишаючи більше для інтелектуальної діяльності юриста і адвокатської колегії.

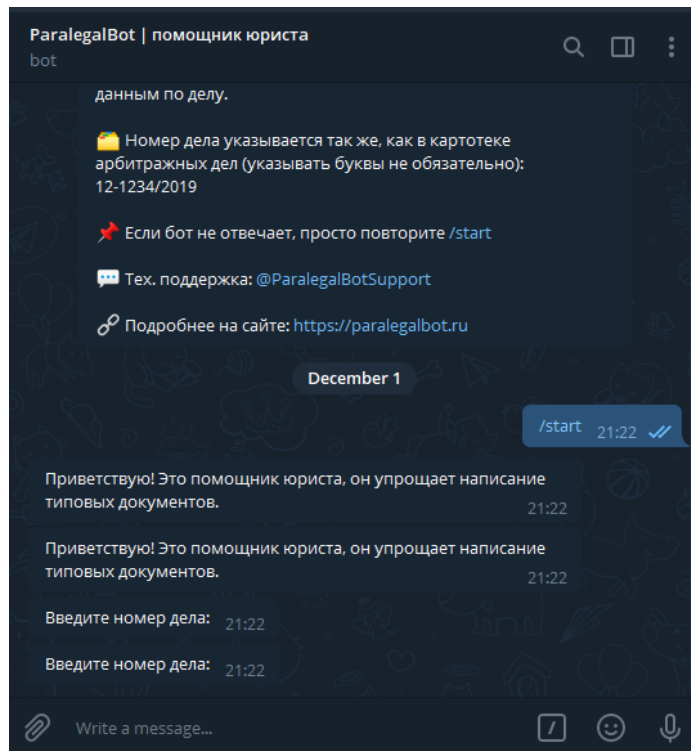


Рисунок 2.3 – Интерфейс бота Paralegalbot в Telegram

iGovBot – телеграм-бот, який шукає на порталі держпослуг iGov інформацію про компанії за назвою, кодом ОКПО або імені директора і повідомляє про зміни в тих компаніях, на які підписаний користувач.

OpenDataBot – мультиплатформенний бот, який моніторить реєстраційні дані українських компаній і судового реєстру.

То чому вигідно використовувати ботів?

По-перше, це дешевизна. Для компаній економічно вигідніше використовувати чат боти, ніж наймати людські ресурси. Як і люди, різні чат-бот можуть виконувати широке коло завдань від більш простого надання консультацій в США і здійснення покупок і замовлень замість клієнтів. Роблячи все це, чат-бот ніколи не втомлюються і не хворіють. Таким чином, вони не вимагають відплатних відпусток, лікарняних і доплат за роботу в нічні зміни на відміну від працівників людей.

По-друге, миттєва відповідь, відсутність черг. Чат-бот швидко виробляють точні відповіді. Менше, ніж секунди потрібні, щоб зіставити

запитувану інформацію з базою даних знань і згенерувати результат. Вони не витрачають зайвого часу на роздуми, пошуки, згадка і відкривання багатьох вкладок браузера. Більш того, чат боти можуть вести багато діалогів одночасно, тому клієнтам не потрібно чекати в черзі, що вони ненавидять робити. Швидке реагування на запит забезпечує чудовий користувацький досвід.

По-третє, доступність 24/7. Чат боти дозволяють компаніям постійно підтримувати зв'язок зі своїми клієнтами і не втрачати можливості отримувати прибуток. Для чат-бот форумі обідніх перерв і неробочих годин. Споживачі також раді, що можуть отримати допомогу в будь-який час доби, тижні і роки – саме тоді, коли вони її потребують, поки їхні запитання ще актуальні і інтерес не зник.

### 2.3 Дослідження фреймворків для створення чат-бота

API для створення чат-бота та послуг з їх використання надають такі іноземні компанії як Pandorabots і Synthetic Intelligence Network: перший надає доступ до свого API на базі досить простого XML-подібного скриптової мови AIML, який реалізує всі технології, необхідні для створення простих комерційних чат-ботів. Вартість використання API Pandotabots від 9\$ до 999\$ в залежності від кількості підтримуваних спамерських пошукових роботів Synthetic Intelligence Network пропонує мову SIML, концептуально схожий з AIML і створений на базі нього, а також свою власну середу розробки ботів - Syn Chatbot Studio під .NET.

Також досить відомими фреймворками для створення власних ботів є Errbot на Python і Hubot на CoffeeScript під Node.js. При цьому написані боти як для Errbot, так і Hubot можуть бути перенесені на Slack, Telegram, IRC і т.д.

Фреймворки для обробки природної мови. Для обробки природної мови на рівні, необхідному для створення чат-бот для полегшення роботи консультантів-людей в будь-якому сервісі, в базовому вигляді можуть підійти багато open-source технології. Подібний фреймворк також може бути створений на основі, наприклад, Apache OpenNLP. Ще не забуваємо про існуючі API, які забезпечують виділення сутностей і всяке подібне з прицілом на використання для чат-ботів – вже згадуваний тут api.ai і wit.ai. Обмежене застосування для реалізації чат-ботів може знайти інструментарій Yandex SpeechKit Cloud, що пропонує доступ як до мовних технологій, так і розпізнавання іменованих сутностей наступних типів: ПІБ, адреса, дата і час.

Чат-боти – це дуже перспективний напрямок. Їх активне використання в месенджерах і в якості цифрових асистентів в смартфонах з великою ймовірністю призведе до популяризації UX-парадигми messaging-as-an-interface. Уже зараз ключові гіганти Facebook, Google, Apple займаються розробкою чат-ботів персональних помічників.

Висновки про технологічні можливості:

Як створити групу чату бота, навіть з урахуванням використання останніх технологічних розробок, має на увазі дуже велике людське участь на всіх етапах розробки (в першу чергу для завдання шаблонів реакцій на призначене для користувача поведінка). Створення комерційного чат-бота не вимагає складних технологій – досить буде базових технологій обробки мови.

Існує достатня кількість фреймворків і API, які можуть бути використані для подальшого створення високоінтелектуальних чат-ботів.

Крім того, фреймворк для обробки мови для комерційного чат-бота можна, в принципі, створити самостійно на базі різних програмних бібліотек з відкритим вихідним кодом [14].

## 2.4 Складність створення чат-бота

Складність створення полягає в тому, що не дуже просто налагодити пошук виданої користувачем фрази по базі виразів та слів, які має бот. Для розмови на зрозумілій мові потрібна специфічна семантична пошукова машина по узагальненим шаблонам репліки, яка вміє розпізнати найбільш схожий шаблон, навіть якщо частина слів в запиті пропущені або замінені на інші слова. В цій пошуковій машині повинна бути вбудовано алгоритмічна морфологія, або як ще її називають обробка словозміни, також синтаксичний розбір, словники синонімів та інше [15].

Машина діалогу повинна здійснювати пошук відповідної репліки, пам'ятаючи контекст попередньої розмови. Але головною проблемою є покриття всієї різноманітності вхідних реплік, тому що людина може запитати що завгодно і в будь-якій формі.

Справа в тому, що у будь-яких пошукових машин є дві характеристики: точність і повнота. Для звичайних пошукових систем необхідна саме точність: важливо, щоб був вірно побудований список в першій десятці результатів, а повнота – то є то, скільки релевантних документів пошукова система витягнула з інтернету нікого не хвилює. У випадку ж з семантичним пошуком все зовсім навпаки. Точність у більшості ботів завжди на високому рівні – якщо вийшло розпізнати репліку, то відповідь буде точною. Що до повноти, то гарантувати розпізнавання великої частки вхідних фраз для такої пошукової системи досить складно, бо людина, яка вільно володіє мовою, може вигадати будь-які вхідні фрази, їх різноманітність така велика, що відповісти на всі дуже складно. Крім цього, потрібно враховувати, що бот протягом часу буде зустрічатися з «хвилею» нових питань, базованих на якихось подіях. В деяких випадках потік реплік можна спрогнозувати, але потрібно реагувати швидко.

Можливості роботів:

- можливість збору зворотного зв'язку;
- сучасні технології мовного процесингу дозволяють вести складно організований діалог з людиною;
- інтеграція з зовнішніми джерелами даних;
- впровадження в мобільні платформи;
- зручний і доброзичливий інтерфейс діалогових систем.
- рішення бізнес задач, таких як технічна підтримка, довідка, просування.

Відмінності автоматизованої комунікації від природної полягають:

- в рівні та методі розуміння віртуальним співрозмовником репліки співрозмовника;
- в загальній схемі і глибині всього діалогу;
- в характері, стилістиці і інші особливості формування відповіді.

Ці відмінності є наслідком природної обмеженості технологій NLP в порівнянні з мовною компетенцією людини наявністю у віртуального співрозмовника можливостей, які людині не доступні [16].

Обмеження на рівні аналізу вхідного тексту:

- семантично-лексичні дані;
- тематична специфіка робота;
- жаргон, неологізми і т.д.;
- помилки.

Важливий аспект, що потребує уваги під час розробки:

- тестування, що призводить до збою комунікації;
- діалогова поведінка віртуального співрозмовника;
- набір цілей співбесіди будь то розвага, чи консультація, чи рекомендація або навіть реклама;
- глибина володіння той чи іншою темою (предметною областю);
- варіативність, стилістична структура і інші мовні вимоги до відповідей віртуального співрозмовника;



- комунікативна організація діалогу.

Віртуальний співрозмовник повинен відповідати завжди та швидко, повинен володіти великою кількістю даних, не відволікатися, використовувати повторювані формулювання, підставляти дані, взяті з БД і сервісів, коректне комунікативне позиціонування: не треба намагатися імітувати людину, активне використання підказок і питань, коректна реакція на комунікативні проблеми: прохання переформулювати питання, чи не відхилятися від теми, побудова розмови відповідно до функціонального призначення конкретного бота.

## 2.5 Недоліки аналогів чат-ботів

Складно уявити технологію, у якій немає недоліків. Основні мінуси чат-ботів:

- багато хто не знає, хто такі віртуальні помічники;
- немає усталених майданчиків для розміщення;
- мало можливостей;
- простому користувачеві недоступні потужні боти;
- недосконалий інтерфейс.

Нерозуміння сарказму. Чат-боти призначені для взаємодії з людьми, проте останні, на відміну від ботів, схильні висловлюватися неоднозначно і часто вкладають у свої слова прихований сенс. Тому, незважаючи на те, що чат-боти прекрасно справляються з прямолінійними питаннями, генеруючи швидкі і точні відповіді, розпізнати сарказм, іронію і гумор складно навіть для ботів зі штучним інтелектом. Вони приймають всі слова за чисту монету і не вміють читати між рядків. Вчені невпинно працюють над вдосконаленням штучного інтелекту, намагаючись олюднити комп'ютерні програми, проте на даний момент їх можливості щодо цього аспекту залишають бажати кращого.

Погана імпровізація. Чат-бот працюють добре, поки розмова не виходить за межі запрограмованого алгоритму. Однак, розробникам неможливо передбачити мільярда можливих сценаріїв, тому, не шаблонна поведінка користувача збиває бота з пантелику. Намагаючись з'ясувати, що має на увазі співрозмовник, чат-бот задає навідні запитання. Якщо подальші відповіді клієнта залишаються бути незрозумілими, програма продовжує повторювати ті ж уточнюючі питання. Як результат, роздратований і незадоволений сервісом клієнт залишає сайт або додаток, так і не отримавши необхідної інформації.

Чи не для всіх видів бізнесу. Чат бот не є універсальним рішенням, і не для всіх видів бізнесу він доречний. Його використання може бути вдалим для компаній, що надають такі послуги, як таксі, доставка їжі, бронювання готельних номерів, продаж квитків, роздрібна торгівля і інші, де запити клієнтів в основному дуже передбачувані і шаблонізовані. Однак, багато B2B (Business to business) компаній потребують надавати ґрунтовні консультації з індивідуальним підходом і належним чином реагувати відповідно до ситуації на місці, щоб задовольнити клієнтів і завоювати їхню довіру. На жаль, чат боти не можуть забезпечити якісну службу підтримки клієнтів в таких випадках і не можуть стати повноцінною заміною персоналу люди для такого типу бізнесу.

Слід взяти до уваги такі нюанси, як:

- є складні системи фільтрів та налаштувань;
- є досить багато різноманітних сценаріїв використання продукту, з якими один бот не впорається;
- в складних питаннях все одно необхідно підключення до діалогу оператора;
- не використовує методи самонавчання;
- є сфери, в яких програма не впорається із запитом (наприклад, складна бізнес послуга).

## 2.6 Дослідження типового алгоритма розробки чат-бота

Алгоритм - це докладний набір інструкцій для виконання операції або вирішення проблеми. Технічно комп'ютери використовують алгоритми для виведення детальних інструкцій щодо виконання операції. З точки зору ефективності, різні алгоритми можуть легко і швидко виконувати операції або вирішувати проблеми.

Для побудови алгоритму спочатку потрібно дізнатись базову структуру побудови чат-бота його компоненти та їх взаємодію, для цього можна скористатися загальною схемою на рисунку 2.4.

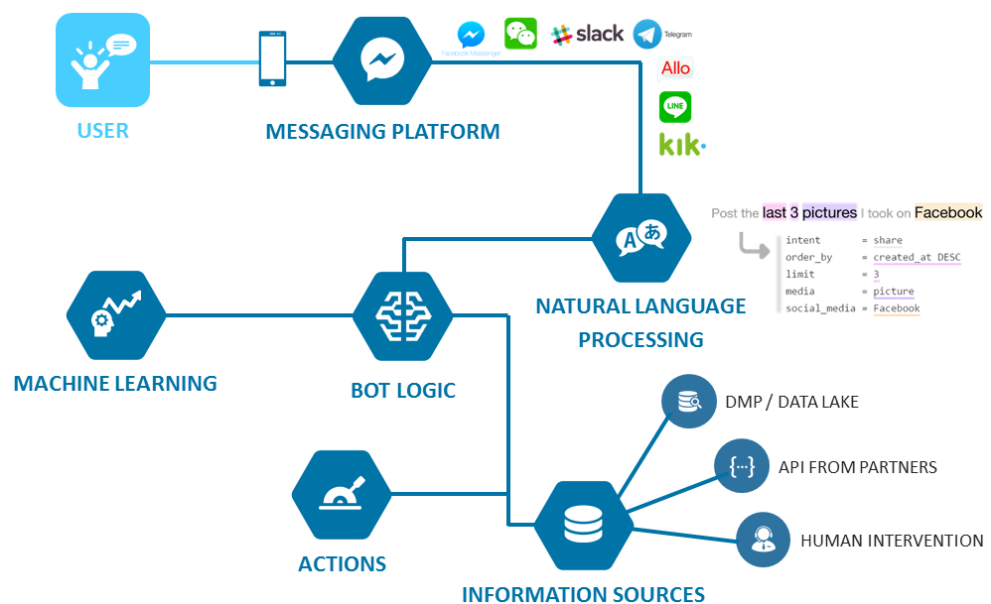


Рисунок 2.4 – Схема логіки чат-бота в месенджерах (Джерело: Hackernoon)

В алгоритмі розробки бота на першому кроці потрібно дізнатися яка платформа використовується користувачем, далі проводиться дослідження обраної платформи, перевірка чи є вже готова API, чи все ж потрібно запрограмувати самостійно (або переконатися в тому, що є вже готова API на

обраній мові програмування) більш детальноше можна подивитися в Додаткові Б.

Наступним кроком буде розглянуте питання – а чи потрібно використовувати нейронні мережі: готової бібліотеки або потрібно побудувати власну нейронну мережу, чи взагалі використовувати звичайні команди. Останній крок – це розробка головної функціональності бота більш детальноше можна подивитися в Додаткові Б. Результатом дослідження є необхідність вдосконалення типового алгоритму с цілю.

## 3 ДОСЛІДЖЕННЯ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ ТА МЕТОДІВ МАШИННОГО НАВЧАННЯ, ЇХ ЗАСТОСУВАННЯ В БОТ РОЗРОБЦІ

### 3.1 Дослідження штучної нейронної мережі

Штучні нейронні мережі були побудовані за принципом біологічних нейронних мереж, які представляють собою сукупність нервових клітин, які виконують певні фізіологічні функції. Складовим елементом нейронних мереж є нейрони. На рисунку 3.1 наведено структуру людського нейрону.

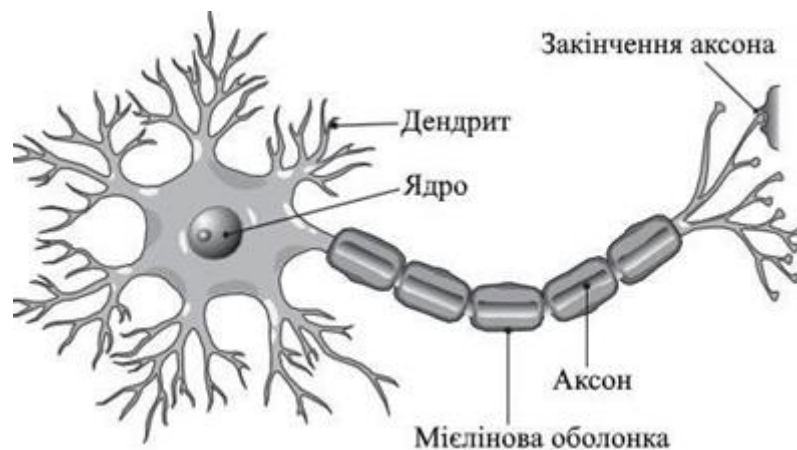


Рисунок 3.1 – Структура людського нейрону

Нейрон має декілька ключових функцій [17]:

- Функція отримання інформації – синапси є джерелами інформації;
- Інтегративна функція – на виході нейрону сигнал несе у собі інформацію про всі підсумовані в нейроні вхідні сигнали;
- Передавальна функція - переданий імпульс змушує медіатор на закінчення аксона передавати збудження наступному нейрону;
- Провідникова функція – інформація передається по аксону.

Синапсами називають фізичні зв'язки, які виводять сигнали від одного нейрона до входів інших. У штучних нейронних мережах кожне з'єднання характеризується певною вагою. Облігації, вага яких позитивний, називаються збудливими, а ті, чия вага негативними, є гальмівними. Вихід нейрона

називається аксоном. У штучній нейронній мережі штучний нейрон - це нелінійна функція, аргументом якої є лінійна комбінація всіх вхідних сигналів. Ця функція називається активацією. Результат функції активації відноситься до виходу нейрона. Набір таких нейронів називається штучною нейронною мережею.

Функція активації нейрона характеризує залежність сигналу на виході нейрона від суми сигналів на його входах. Зазвичай функція монотонно зростає і може знаходитися в діапазоні значень від -1 до 1 (гіперболічна дотична) або від 0 до 1 (сигмоїдна). Для деяких алгоритмів навчання функцію активації потрібно постійно диференціювати по всій числовій осі. Назва штучного нейрона може формуватися відповідно до його функції активації. Наприклад, назва "сигмоїдальний нейрон" вказує на те, що активаційна функція такого нейрона є сигмоїдальною.

До основних типів активаційних функцій належать:

- Порогова активаційна функція (функція Хевісайда):

$$f(x) = \begin{cases} 1 & \text{if } x \geq -w_0x_0 \\ 0 & \text{else} \end{cases} \quad (3.1)$$

- Сигмоїдальна активаційна функція:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$

- Гіперболічний тангенс:

$$\tanh(Ax) = \frac{e^{Ax} - e^{-Ax}}{e^{Ax} + e^{-Ax}} \quad (3.3)$$

Функцію Хевісайда не можна використовувати для алгоритму зворотного поширення помилки.

Перцептрон - тип штучного нейрона, який було розроблено Френком Розенблатом в 1950-их – 1960-их роках. Хоча у сучасних роботах частіше використовують сигмоїдальну модель штучного нейрона, щоб зрозуміти, як

працює сигмоїдальний нейрон, необхідно розглянути структуру і принцип роботи перцептрону. Перцептрон приймає на вхід певні значення  $x_1, x_2 \dots x_n$ , а на виході видає бінарний результат (рисунок 3.2). Розенблат запропонував використання, так званої, ваги ( $w_1, w_2, \dots w_n$ ) - чисел, що відображають важливість вкладу кожного із входів нейрона для кінцевого результату.

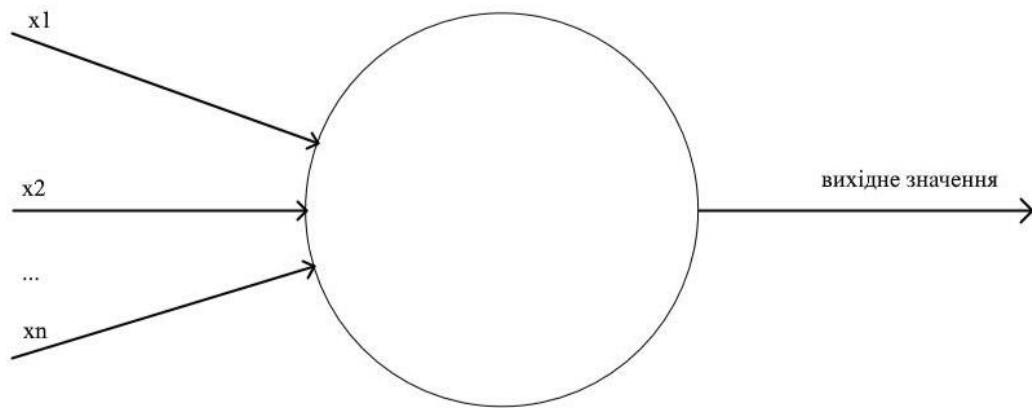


Рисунок 3.2 – Схема перцептрона [18]

Зважена сума входів порівнюється з граничним значенням і відповідно до цього на виході буде видано 0 або 1. Порогове значення є також важливим параметром нейрона.

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases} \quad (3.4)$$

Типові штучні нейронні мережі на базі таких перцептронів можна охарактеризувати такими властивостями: вони містять один прихований шар, у них відбувається пряме розповсюдження сигналу, у них порогова активаційна функція

Навчання перцептрону полягає в зміні його вагової матриці.

Існують 3 історично сформованих види перцептронів [19]:

- Одношаровий перцептрон, вхідні елементи якого безпосередньо пов'язані з вихідними за допомогою системи ваг. Є найпростішою мережею прямого поширення (feedforward network)

- Багатошаровий перцептрон Розенблатта у якому присутні додаткові приховані шари;
- Багатошаровий перцептрон Румельхарта у якому присутні додаткові приховані шари, а навчання проводиться за методом зворотного поширення помилки (backpropagation algorithm).

Якби певна невелика зміна ваги (або зміщення) викликала б невелику зміну на виході мережі, то бажаної поведінки нейронної мережі можна було б досягнути за допомогою простих модифікацій в процесі навчання. Проте навчання, якщо нейронна мережа складається з перцептронів, не так просто здійснити. Невелика зміна ваги або зміщення одного з перцептронів мережі може кардинально змінити його вихідне значення з 0 на 1. Тому незначні зміни значень одного з елементів мережі можуть призвести до труднощів у розумінні змін поведінки мережі. Оскільки завдання навчання нейронної мережі є завданням пошуку мінімуму функції помилки в певному просторі навчальних станів, то для його вирішення можуть застосовуватися стандартні методи теорії оптимізації. Для одношарового перцептрону з  $n$  входами і  $m$  виходами, пошук мінімуму функції необхідно буде здійснювати в  $nm$ -вимірному просторі [19].

Нейрони можна розділити на такі групи, в залежності від їх положення в мережі: вхідні нейрони, які приймають початковий вектор даних, проміжні нейрони, в яких відбуваються основні обчислювальні операції при навчанні та вихідні нейрони, які відображають результат роботи мережі.

Сигмоїдальні нейрони схожі на перцептрони, однак невеликі зміни в їх вагах і зсувах значно менше змінюють вихід нейрону. Цей факт дозволяє мережі, яка складається з сигмоїдальних нейронів ефективніше навчатися. На вхід сигмоїдального нейрону подаються будь-які значення від 0 до 1, так само як і у перцептрона. На виході ж видається не просто значення 0 чи 1, а певне число у проміжку між 0 і 1, оскільки активаційною функцією такого нейрону виступає сигмоїда, яка є нелінійною [19].



$$f(t) = \frac{1}{1+e^{-t}} \quad (3.5)$$

Чим більше  $\beta$  (параметр нахилу сигмоїдальної функції активації), тим більша крутизна графіка. При  $\beta \rightarrow \infty$  сигмоїда наблизатиметься до функції Хевісайда.

### 3.2 Архітектура нейронних мереж

Розглянемо завдання навчання мережі. Дано безліч тренувальних прикладів  $X$  мітками, які позначають очікуване вихідне значення мережі. Нейронні мережі визначають нелінійну гіпотезу  $hw, b(x)$  з параметрами  $w$  (weight - вага) і  $b$  (bias - зміщення) [19]. Нейронна мережа складена з безлічі простих нейронів так, що вихід одного з нейронів буде входом іншого (рисунок 3.3).

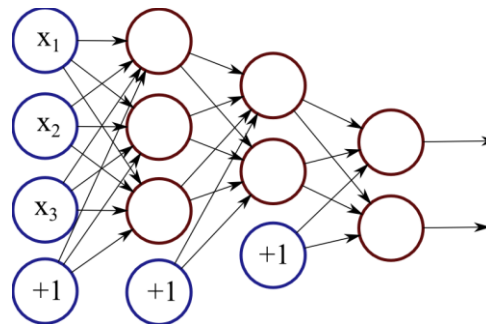


Рисунок 3.3 – Схема простої мережі прямого поширення [20]

Крайній лівий шар називається вхідним, а крайній правий шар — вихідним. Шар посередині є прихованим і називається так через те, що його значення не відстежуються на навчальних прикладах. Таким чином в даній мережі три елементи входу, три прихованих елемента і один вихідний елемент.

Нехай  $nl$  — кількість шарів в мережі (в даному випадку 3). Параметри мережі  $(w, b) = (w(1), b(1), w(2), b(2))$ . Результат застосування функції

активації (виходу) позначається  $a_i$  для кожного  $i$ -ого елемента [21].  
Отримаємо систему:

$$\begin{cases} a_1(2) = f(w_{11}(1)x_1 + w_{12}(1)x_2 + w_{13}(1)x_3 + b_1(1)), \\ a_2(2) = f(w_{21}(1)x_1 + w_{22}(1)x_2 + w_{23}(1)x_3 + b_2(1)) \end{cases} \quad (3.6)$$

Позначивши функцію суми через  $z$ , отримаємо у векторній формі:

$$\begin{cases} z^{(2)} = W^{(1)}x + b^{(1)} \\ a^{(2)} = f(z^{(2)}) \\ z^{(3)} = W^{(2)}a^{(2)} + b^{(2)} \\ h = f(z^{(3)}) \end{cases} \quad (3.7)$$

Тоді загальна формула системи матиме вигляд:

$$\begin{cases} z^{(l+1)} = W^{(l)}a^{(l)} + b^{(l)} \\ a^{(l+1)} = f(z^{(l+1)}) \end{cases} \quad (3.8)$$

Мережею прямого поширення називаються нейронні мережі, які використовують вихід одного шару в якості вхідних даних для наступного шару.

### 3.4 Навчання нейронної мережі

Спершу необхідно дати визначення загальним поняттям перед дослідженням принципів навчання нейронних мереж.

Епоха – прямий і зворотний прохід по усіх тренувальних прикладах.

Розмір партії (batch) – кількість тренувальних прикладів для однієї ітерації прямого і зворотного проходів.

Кількість ітерацій – кількість проходів, при цьому кожен прохід використовує партії (batch). Одним проходом вважається сума прямих проходів та зворотних проходів по цій партії. Наприклад, маючи 1000

тренувальних прикладів, розмір партії рівний 500, буде потрібно дві ітерації, щоб завершити одну епоху.

З математичної точки зору, навчання нейронних мереж – багатопараметричне завдання нелінійної оптимізації.

Алгоритм зворотнього поширення помилки визначає стратегію підбору ваг багат шарової мережі із застосуванням градієнтних методів оптимізації.

Оскільки цільова функція, яка зазвичай визначається як сума квадратичних різниць між фактичними і очікуваними вихідними значеннями, є безперервною, градієнтні методи оптимізації є високоефективними при навчанні мережі. За такого методу навчання багат шарової нейронної мережі необхідно обчислити вектор градієнта щодо параметрів всіх шарів мережі.

Нехай є кінцевий набір тренувальних даних  $m$  прикладів. Для навчання нейронної мережі застосуємо пакетний градієнтний спуск (batch gradient descent).

Квадратична помилка цільової функції (squared-error cost function) для одного прикладу буде обчислена за формулою 3.9

$$J(W, b; x, y) = \frac{1}{2} \|h_{W,b}(x) - y\|^2 \quad (3.9)$$

Тоді цільова функція, для  $m$  прикладів, буде виглядати так за формулою 3.10:

$$\begin{aligned} J(W, b) &= \left[ \frac{1}{m} \sum_{i=1}^m J(W, b; x^{(i)}, y^{(i)}) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \\ &= \left[ \frac{1}{m} \sum_{i=1}^m \left( \frac{1}{2} \|h_{W,b}(x^{(i)}) - y^{(i)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \end{aligned} \quad (3.10)$$

Перший член у виразі,  $J(W, b)$ , являє собою суму квадратів помилок, другий – член регуляризації (L2 – зменшення ваги), який дозволяє зменшити вагу і запобігти переоснащення. Параметр регуляризації ваг  $\lambda$  використовується для перевірки відносної значущості частин цього виразу. У завданнях бінарної класифікації у представлений нулем або одиницею

(оскільки сигмоїдальна функція дає значення в діапазоні від 0 до 1, однак при використанні функції активації на основі гіперболічного тангенса інтервал буде від -1 до 1). Основним завданням у навчанні є мінімізація  $J(W, b)$ . Для навчання нейронної мережі необхідно ініціалізувати кожен параметр випадковими значеннями, близькими до нуля, а потім застосувати алгоритм оптимізації [22].

Оскільки  $J(W, b)$  не є опуклою функцією, то градієнтний спуск сприйнятливий до локальних оптимумів. Кожна ітерація градієнтного спуску оновлює параметри таким чином за формулою 3.11, 3.12, 3.13:

$$\begin{aligned} W_{ij}^{(l)} &= W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) \\ b_i^{(l)} &= b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W, b), \end{aligned} \quad (3.11)$$

Тут  $\alpha$  – швидкість навчання.

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) = \left[ \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x^{(i)}, y^{(i)}) \right] + \lambda W_{ij}^{(l)} \quad (3.12)$$

$$\frac{\partial}{\partial b_i^{(l)}} J(W, b) = \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial b_i^{(l)}} J(W, b; x^{(i)}, y^{(i)}) \quad (3.13)$$

Далі ми опишемо кроки алгоритму зворотного розповсюдження:

- Здійснюється прямий прохід по мережі, обчислюються активації шарів  $L_2, L_3$  і так далі, аж до вихідного шару  $L_{nl}$ ;
- Для кожного вихідного елементу  $i$  в вихідному шарі  $n_l$  (the output layer) розраховується помилка за формулою 3.14:

$$\delta_i^{(n_l)} = \frac{\partial}{\partial z_i^{(n_l)}} \frac{1}{2} \|y - h_{W,b}(x)\|^2 = -(y_i - a_i^{(n_l)}) \cdot f'(z_i^{(n_l)}) \quad (3.14)$$

- Для кожного елемента в шарі  $l$ , розраховується за формулою 3.15:

$$\delta_i^{(l)} = \left( \sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) f'(z_i^{(l)}) \quad (3.15)$$

- Вираховуються часткові похідні за формулою 3.16, 3.17 :

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x, y) = a_j^{(l)} \delta_i^{(l+1)} \quad (3.16)$$

$$\frac{\partial}{\partial b_i^{(l)}} J(W, b; x, y) = \delta_i^{(l+1)} \quad (3.17)$$

Ось кілька основних варіантів, які слід враховувати під час навчання мереж. Один з головних – темп навчання. Для його практичного застосування спочатку необхідно оцінити граничне значення для  $\eta$ , при якому значення цільової функції починає негайно зменшуватися без коливань. Спочатку значення оцінки встановлюється на  $\eta = 0,01$ . Якщо значення цільової функції зменшується протягом перших епох, то рівень навчання необхідно збільшувати, поки не буде знайдено значення коливання цільової функції. Якщо при початковому темпі тренінгу значення цільової функції коливаються, то її треба зменшити. Темп тренувань регулює розмір кроку при спуску градієнта, залежно від значення цільової функції, визначаючи, чи занадто великий розмір кроку спуску градієнта.

Застосування ранньої зупинки використовується для визначення розміру епох навчання. Рання зупинка означає, що наприкінці кожної епохи необхідно обчислити валідаційний набір. Коли підвищення точності припиняється, процес навчання повинен бути припинений. Така зупинка також запобігає перекаліфікації [23].

Графік тренувань є ідеєю тримати темп тренування постійним, поки точність даних тестів не почне погіршуватися. Тоді потрібно зменшити темп тренувань (наприклад, у 10 разів).

Розмір пакету також є надзвичайно важливим параметром при навчанні мережі. Якщо розмір пакету занадто малий, не можна повною мірою скористатися хорошими бібліотеками матриць, оптимізованими для швидкого обладнання. Якщо розмір пакету занадто великий, то ваги мережі не оновлюватимуться досить часто. Необхідно вибрати компромісне значення, яке максимально збільшує швидкість навчання.

### 3.5 Глибоке навчання при розробці чат-бота

Що таке чат-глибоке навчання?

Чат-бот з глибоким навчанням вчиться з нуля через процес, який називається «Глибоке навчання». У цьому процесі чат-бот створюється за допомогою алгоритмів машинного навчання. Чат-бот з глибоким навчанням вивчає все на основі своїх даних та діалогу між людьми.

Чат-бот навчається розвивати власну свідомість щодо тексту, і ми можемо навчити його спілкуванню з людьми. Крім того, ми можемо навчати чат-бота за допомогою діалогу з фільмами або відтворення сценаріїв. Однак розмова між людьми є кращим засобом створення найкращого можливого чат-глибокого навчання. Пам'ятайте, чим більше у вас даних, тим кращою буде ефективність машинного навчання [24].

Тепер, коли ми знаєте, що таке глибокий навчальний чат, давайте спробуємо зрозуміти, як можна створити його з нуля.

Перший крок будь-якого процесу машинного навчання - підготовка даних. Вам потрібно мати тисячі існуючих взаємодій між клієнтами та вашим обслуговуючим персоналом, щоб навчати свого чат-бота.

Вони повинні бути максимально детальними та різноманітними, щоб було достатньо точок даних для вашого чату з глибоким навчанням. Цей конкретний процес називається створенням онтології. Вашою єдиною метою на цьому етапі має бути зібрати якомога більше взаємодій.

Цей крок може залежати від вашого джерела даних, залежно від вашого джерела даних. Якщо ваші дані недостатньо відокремлені, вам потрібно буде переробити свої дані в окремі рядки спостережень.

Ці спостереження можна назвати парами повідомлення-відповідь, які будуть додані до класифікатора.

Мета цього кроку - поставити одного диктора як відповідь у розмові. Весь вхідний діалог буде використовуватися як текстові індикатори, які допоможуть передбачити відповідь.

Можливо, знадобиться встановити деякі обмеження під час створення пар відповіді на повідомлення, наприклад:

Розмова повинна бути лише між двома людьми. Це дає зрозуміти, на кого спрямоване повідомлення. Окремі повідомлення, надіслані протягом хвилини, можна об'єднати в одне повідомлення. Щоб з'єднати повідомлення з відповіддю, відповідь на повідомлення повинна надійти протягом 5 хвилин.

Після накопичення цих даних потрібно очистити їх. Вам потрібно видалити URL-адреси, посилання на зображення, стоп-слова тощо.

### 3.6 Дослідження моделі Послідовність до послідовності (Seq2Seq)

Моделі Seq2Seq – це найбільш часто використовувана архітектура в машинному перекладі та системах зіставлення питань нейронної мережі. Найбільша кількість пам'яті в таких моделях витрачається на зберігання матриці уявлень, що містить представлення кожного слова зі словника. Для прислівного породження послідовної відповіді необхідно мати поряд із стандартною формою слова всі його словоформи. У деяких мовах слова мають лише невелику кількість словоформ (наприклад, однини та множини). Однак у мовах, таких як українська, у багатьох словах присутня велика кількість словоформ, які утворюються зміною статі, числа, відмінка та часу.

Алгоритм перекладу складається з таких основних етапів [25]:

1) Енкодер (enc) послідовно зчитує вхідний пропозицію і видає вектор контексту «C». На малюнку передачі вектора контексту від енкодера до декодера відповідає стрілка  $h_4 \rightarrow dec_1$ .

2) Отримавши опис вхідної пропозиції в вигляді «C», декодер (dec) генерує слова на іншій мові послідовно. Окремим «словом» в словнику мови Б позначимо початок пропозиції «BOS» і кінець пропозиції «EOS».

Схематично цей процес представлений на рисунку 3.4.

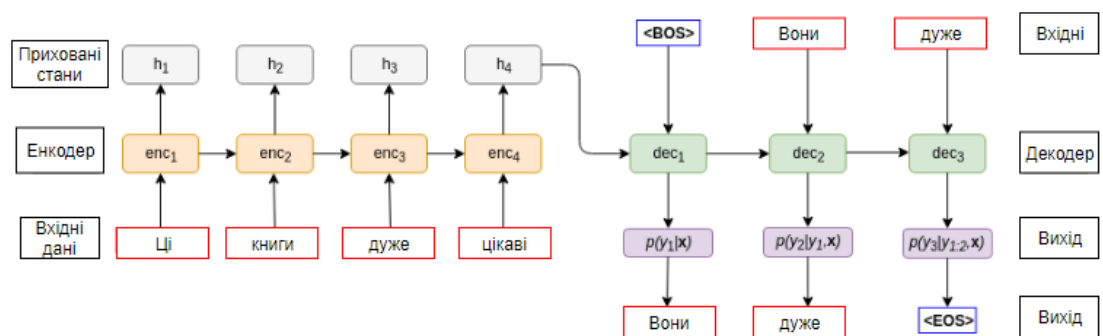


Рисунок 3.4 – Схема моделі перекладу

Декодер можна представити у вигляді функції:

$$h_t, p_t = dec(h_{t-1}, w_{t-1}) \quad (3.18)$$

Тут  $h_t$  - вектор прихованого стану декодера на кроці  $t$ ,  $p_t$  - розподіл за словами на етапі  $t$ ,  $w_{(t-1)}$  це слово, яке вибрано в якості основної гіпотези виходячи з розподілу  $p_{(t-1)}$  на етапі « $t-1$ ».

Для енкодера аналогічне рівняння виглядає наступним чином:

$$h_t = enc(h_{t-1}, w_t) \quad (3.19)$$



Зверніть увагу, що в якості фінального вектора контексту  $C$  вибирається останнім приховане стан енодера, в даному випадку  $h_4$ . У разі більш складних підходів схема трохи ускладниться.

Перевага даної моделі полягає в тому, що кількість параметрів моделі не залежить від довжини пропозиції: одні й ті ж моделі (enc і dec) працюють на всіх етапах за часом.

На завершення, повернемося до питання вибору правильної послідовності слів  $w_1, w_2, \dots, w_n$ , враховуючи що декодер видає тільки ймовірності слів  $p_1, p_2, \dots, p_n$ .

Ми можемо вибрати будь-яке слово  $w(t-1)$  з категоріального розподілу  $p(t-1)$  над словником, отримати нове приховане стан  $h_t$  і новий розподіл  $p_t$ . Вибору цього слова буде відповідати певна ймовірність, яку будемо позначати  $p(t-1)(w(t-1))$ .

Виконуючи на кожному кроці таку процедуру, отримуємо сумарну ймовірність генерації послідовності слів:

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n p_i(w_i) \quad (3.20)$$

Найпростіший спосіб генерувати послідовність - вибрати на кожному кроці найімовірніше слово  $w_t$  з розподілу  $p_t$  («greedy decoding»). Це, однак, не гарантує нам, що ми знайдемо саме ту послідовність слів, яка максимізує  $P(w_1, w_2, \dots, w_n)$  для заданого енодера і декодера. Головними недоліками даної моделі є швидкість навчання та вузька направленість. Друга основна проблема моделей seq2seq полягає в тому, що вони можуть генерувати невідповідні відповіді для перефразованих контекстів, але з тим же сенсом

### 3.7 Дослідження моделі Слово у вектор (Word2Vec)

У міру розвитку обробки природної мови безліч завдань вирішувалося класичними статистичними методами і безліччю правил, однак проблему

нечіткості і неоднозначності в мові це не вирішувало. Якщо ми скажемо «замок» без будь-якого контексту, то навіть живий співрозмовник навряд чи зрозуміє, про що йде мова. Семантику слова в тексті визначають слова-сусіди. Але як пояснити це машині, якщо вона розуміє лише числове уявлення? Так народився статистичний метод аналізу тексту word2vec (Word to vector).

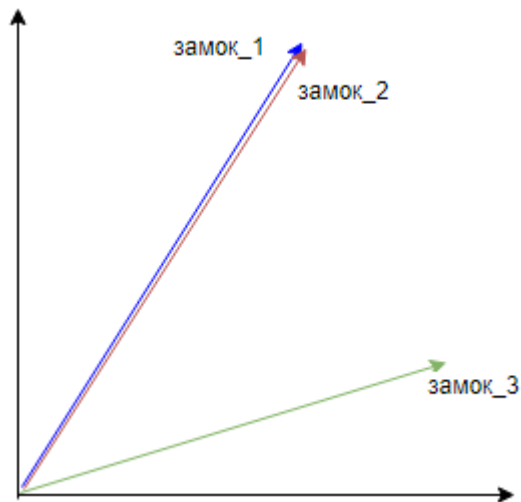


Рисунок 3.4 – Вектори схожих слів «замок\_1», «замок\_2» та омоніма «замок\_3»

Ідея цілком очевидна з назви: уявити слово у вигляді вектора з координатами  $(x_1, x_2, \dots, x_n)$  (рисунок 3.4). Для боротьби з омонімією однаковим словами приєднується тег: «замок\_1», «замок\_2» і так далі. Якщо вектори «замок\_n» і «замок\_m» паралельні, то їх можна вважати одним словом. В іншому випадку ці слова – омоніми. На виході у кожного слова з'являється своє векторне подання в багатовимірному просторі (розмірність векторного простору може варіюватися від 50 до 1000).

Мета та корисність Word2vec – згрупувати вектори подібних слів разом у векторі простору. Тобто виявляє схожість математично. Word2vec створює вектори, які розподіляють числові уявлення про ознаки слова, такі функції, як контекст окремих слів. Це робиться без втручання людини.

Враховуючи достатньо даних, використання та контекстів, Word2vec може робити дуже точні здогадки про значення слова на основі минулих

виступів. Ці здогадки можна використовувати для встановлення асоціації слова з іншими словами (наприклад, «чоловік» – «хлопчик», що «жінка», «дівчина»), або кластеризувати документи та класифікувати їх за темою. Ці кластери можуть стати основою пошуку, аналізу настроїв та рекомендацій у таких різноманітних галузях, як наукові дослідження, юридичні відкриття, електронна комерція та управління відносинами з клієнтами [26].

Вихід нейронної мережі Word2vec – це словниковий запас, в якому кожен елемент має приєднаний до нього вектор, який можна подати в мережу глибокого вивчення або просто запитати для виявлення зв'язків між словами.

Недоліком word2vec є те, що з його допомогою слова, які не зустрічаються у навчальному наборі, не можуть бути представлені.

### 3.8 Рекомендації по використанню методу машинного навчання для вдосконалення роботи чат-бота

Залишається питанням, який тип нейронної мережі використовувати для тренування умовного чату-бота. У людській мові важлива послідовність: ми робимо деякі висновки та приймаємо рішення, ґрунтуючись на сказаному в попередньому реченні або навіть абзаці. Поточна нейронна мережа (RNN) добре підходить для цих критеріїв, але в міру збільшення відстані між спорідненими частинами тексту розмір RNN повинен збільшуватися, що знизить якість обробки інформації. LSTM (Довга короткострокова пам'ять) вирішує цю проблему. Мається на увазі одну важливу особливість - стан клітини, який може залишатися постійним, або змінюватися при необхідності. Таким чином, інформація в ланцюзі не втрачається, що є критичним для природної обробки мови.

Сьогодні існує багато бібліотек для обробки природних мов. Якщо говорити про мову Python, яка часто використовується для аналізу даних, це NLTK та Spacy. Великі компанії також беруть участь у розробці бібліотек NLP,

таких як NLP архітектор Intel або дослідники Facebook та Uber в PyTorch. Незважаючи на такий великий інтерес до нейронних мережевих методів обробки мови великими компаніями, взаємопов'язані діалоги будуються переважно на основі класичних методів, а мережа відіграє допоміжну роль у вирішенні проблеми попередньої обробки та класифікації мови.

Вибір правильного алгоритму машинного навчання вважається важливим моментом у розробці будь-якої системи прогнозування. У більшості випадків цей вибір залежить від специфіки даних, що аналізуються, та умов аналізу. Усі алгоритми машинного навчання поділяються на три типи: контрольоване, неконтрольоване та підкріплене навчання.

Контрольоване навчання застосовується, коли є масив даних, елементи якого мають певні явні властивості, але для іншого масиву ці властивості неявні і їх потрібно передбачити під час роботи. Безконтрольне навчання підходить для виявлення неявних зв'язків у заданому без маркінгу наборі даних. Підтримка навчання - це симбіоз вищезазначених категорій, а його суть полягає у наявності певного зворотного зв'язку, який доступний для кожного кроку чи дії, але не має будь-якого повідомлення про розмітку та помилку.

У разі аналізу запитань існує база питань та відповідей, а також чіткі ознаки у вигляді наявності певних слів у питанні. У зв'язку з цим було прийнято рішення використовувати один із керованих алгоритмів. Далі наведені основні завдання, які можна вирішити за допомогою таких алгоритмів

**Класифікація.** Це завдання полягає у визначенні належності досліджуваного об'єкта до певного класу, підмножини. Якщо визначення відбувається лише за наявності двох варіантів, то таку класифікацію називають двочленною або двокласовою. Якщо існує багато категорій, наприклад, при пошуку серед багатьох подібних питань, то класифікація називається багатоскладною.

Регресія. Це завдання ґрунтується на прогнозуванні ступеня тієї чи іншої характеристики в даному наборі даних.

Виявлення аномалії. У таких випадках завдання полягає у визначенні точок даних, які вибиваються із загального набору деякими атрибутами. Наприклад, виявляючи шахрайство з банківською карткою, підозрюваними є будь-які платіжні операції, які мають занадто різні функції, будь то місце виводу, час чи сума трансакції. У цій роботі вибір ставився між завданням класифікації, а саме задавати кожне запитання з бази даних як окремий клас, і між проблемою регресії, зокрема, можна розділити питання за темами, щоб визначити питання наскільки як участь у кожній темі. Очевидно, що класифікація є оптимальним рішенням, оскільки вирішення проблеми регресії в поточному проекті вимагає занадто великої обробки даних, а також дає далеко не найточніші результати, оскільки багато питань можуть мати однакову участь у заданих темах.

Наступним кроком є вибір відповідного алгоритму класифікації класів, а головну роль відіграє той факт, що наявність певних слів є ознакою питання. Важливо, щоб більш рідкісні та значущі слова відігравали ключову роль у класифікації, а менш відповідні слова будуть використані пізніше. Ця модель майже повністю відповідає алгоритму вирішального дерева, який базується на знаках з найбільшою вагою, а ближче до кінців гілок знаків з найменшими вагами. В одній з перших версій проекту цей алгоритм був взятий за основу, однак відразу було виявлено, що якість прогнозів дуже низька. Це пов'язано, по-перше, з тим, що набір незначущих слів у питанні створює велику відволікання у вибірці, по-друге, невеликий масштаб навчальної вибірки призводить до явища, званого перепідготовкою.

По-перше, відбувається так звана обробка бази діалогу та їх можливе продовження, після чого залишаються лише ті набори, які відповідають поточному етапу діалогу.

По-друге, навчальний пакет складається з підготовленого набору питань. Перед цим текст кожного запитання зводиться до єдиної форми, а саме символи видаляються, а слова, написані англійською мовою, замінюються їх транслітерацією на російську мову. Це робиться для обережного поводження з деякими англійськими термінами, які користувачі можуть писати російською мовою. Наприклад, "CRM" часто використовується замість "CRM". Також усі слова нормалізуються за допомогою морфологічного аналізатора `rumorphy2`, таким чином, кожне слово зводиться до початкової форми, причастя перетворюється на інфінітив дієслова. Це дозволяє обробляти одні й ті самі слова в різних контекстах однаково.

**купити історичну книгу**

word 1                      word 2   word 3

	word 1	word 2	word 3
<b>query</b>	1	1	1
<b>doc 1</b>	1	1	
<b>doc 2</b>	2	2	1
<b>doc 3</b>		5	5
<b>doc 4</b>		1	

Рисунок 3.5 – Обчислення відносної ваги слів (вектор TF-IDF)

По-третє, із складеної та нормалізованої вибірки питань створюється вектор TF-IDF (Термін Частота / Зворотна частота документа). Цей процес включає обчислення відносної ваги кожного слова протягом усієї вибірки приклад на рисунку 3.5. Вага слова обчислюється діленням частоти зустрічі слова на частоту його зустрічі в інших питаннях. Таким чином, слова, що означають послуги, технології та галузі, що цікавлять користувачів, мають масштаби, які набагато більше, ніж слова, що складають решту питань. Потім на основі підготовленого векторизатора обробляється вся база запитань та навчання бота.

Наприклад розглянемо текст, що містить сто слів, у якому слово «поповнити» з'являється п'ять разів. Параметр TF для слова «поповнити» становить  $(5/100) = 0,05$ . Припустимо, у нас є десять мільйонів документів, а слово «поповнити» з'являється у тисячах. Коефіцієнт обчислюється  $1 + \log(10\,000\,000/1000) = 4$ . То TD-IDF дорівнює  $0,05 * 4 = 0,20$  [27].

Недоліки: обмежений словник, можна отримати вагу тільки для тих слів, які присутні в тренувальній вибірці, не враховується перефразування, не враховується порядок, в якому слова зустрічалися в тексті.

На таблиці 3.1 зображена точність даного методу.

Таблиця 3.1– Точність методу TF-IDF

	~ 10 ключових слів	~ 30 ключових слів
ML + tf-idf	0,92	0,85

## 4 РОЗРОБКА ПРИКЛАДНОЇ ПРОГРАМИ COFFEE BAR ЧАТ-БОТА НА ОСНОВІ РОЗПІЗНАВАННЯ КЛЮЧОВИХ СЛІВ, МЕНЮ / КНОПОК ТА КОНТЕКСТУ

### 4.1 Розпізнавання ключових слів, меню / кнопок та контексту

Чати на основі меню / кнопок – це найпоширеніший тип чат-ботів на ринку сьогодні. У більшості випадків ці чати – це прославлені ієрархії дерева рішень, представлені користувачеві у вигляді кнопок. Подібно до автоматизованих меню телефонів, з якими ми всі взаємодіємо майже щодня, ці чати вимагають від користувача зробити кілька виборів, щоб заглибитись до остаточної відповіді.

Хоча цих чатів достатньо, щоб відповісти на ці загальні питання, що складають 80% запитів підтримки; вони недоотримуються в більш просунутих сценаріях, в яких є занадто багато змінних або занадто багато знань, щоб передбачити, як користувачі повинні впевнено дістатись до конкретних відповідей. Варто також зазначити, що чат-боти на основі меню / кнопок є найповільнішими з точки зору досягнення користувачем потрібного значення.

На відміну від чатів на основі меню, чат-боти на основі розпізнавання ключових слів можуть слухати, що користувачі вводять та відповідати належним чином, або принаймні намагатися. Ці чати використовують налаштовані ключові слова та AI, щоб визначити, як подати відповідну відповідь користувачеві. Система відповідності ключових слів не аналізує весь вхід користувача, але фокусується на пошуку слів за фразами, визначеними користувачем. Межа похибки оцінки достовірності не застосовується до цієї системи, тому відповідають лише ті самі слова. Система приймає розділові знаки в кінці слова і максимум одне слово між ключовими словами. Ключові слова краще використовувати, коли ми очікуємо, що користувач скаже певне слово чи фразу, або коли нас цікавить конкретне слово.



Наприклад, якщо користувач задав питання «Яка кава є в COFFEE BAR?», бот, швидше за все, використовує ключові слова «кава» та «COFFEE BAR», щоб найкраще визначити, які відповіді підходять.

Контекстуальні чати – це самий передовий з трьох ботів. Ці чати використовують машинне навчання (ML) та штучний інтелект (AI), щоб запам'ятати розмови з конкретними користувачами, щоб навчитися та зростати з часом. На відміну від чатів на основі розпізнавання ключових слів, контекстні чати досить розумні, щоб самовдосконалитись на основі того, що користувачі просять і що вони запитують.

#### 4.2 Інструменти та додаткові засоби для розробки моделі

Для відображення підходів розробки чат-бота було запропоновано розробити COFFEE BAR бота, для розробки якого потрібні певні засоби та інструменти, в даній роботі я буду використовувати:

- JSON для обміну інформацією (так як бот повертає дані в цьому вигляді);
- Node.js для серверної частини;
- API APP ChatBot.

JSON – це стандартний текстовий формат для подання структурованих даних на основі синтаксису об'єкта JavaScript. Він зазвичай використовується для передачі даних в веб-додатках (наприклад, відправка деяких даних з сервера на клієнт, тому їх можна відображати на веб-сторінці або навпаки). Ви зіткнетеся з цим досить часто, тому в цій статті ми даємо вам все, що вам потрібно для роботи з JSON, використовуючи JavaScript, включаючи розбір JSON, щоб ви могли отримати доступ до даних усередині нього і створювати JSON.

Node.js – це платформа, побудована на JavaScript під час виконання JavaScript для зручного створення швидких та масштабованих мережеских додатків. Node.js використовує керовану подіями модель, що не блокує

введення / виведення, що робить її легкою та ефективною, ідеальною для використання в реальному часі додатків, що працюють на реальному часі, які працюють на розподілених пристроях.

APP ChatBot – це система розуміння мови, яка дозволяє створювати інтелектуальні чати для будь-якої служби. Ви можете легко інтегрувати своїх ботів із улюбленими програмами для обміну повідомленнями та дозволяти їм постійно обслуговувати своїх клієнтів.

Кожен запит API вимагає автентифікації, щоб визначити ліцензію, яка відповідає за подання запиту. Автентифікацію надають маркери доступу (рисунок 4.1).

```
var http = require("https");

var options = {
  "method": "GET",
  "hostname": "api.chatbot.com",
  "port": null,
  "path": "/stories",
  "headers": {
    "authorization": "Bearer ${DEVELOPER_ACCESS_TOKEN}"
  }
};

var req = http.request(options, function (res) {
  var chunks = [];

  res.on("data", function (chunk) {
    chunks.push(chunk);
  });

  res.on("end", function () {
    var body = Buffer.concat(chunks);
    console.log(body.toString());
  });
});

req.end();
```

Рисунок 4.1 – Приклад запиту аутентифікації (код Node js)

### 4.3 Побудова прикладної програми COFFEE BAR BOT

Для початку побудови потрібно створити план роботи бота, який приймає замовлення на каву та показує меню. Розглянемо план бота. Кожне коло візуалізує потенційну взаємодію з користувачами (рисунок 4.2). Як бачимо, бот зможе приймати та виконувати замовлення, підводити підсумки та показувати меню.

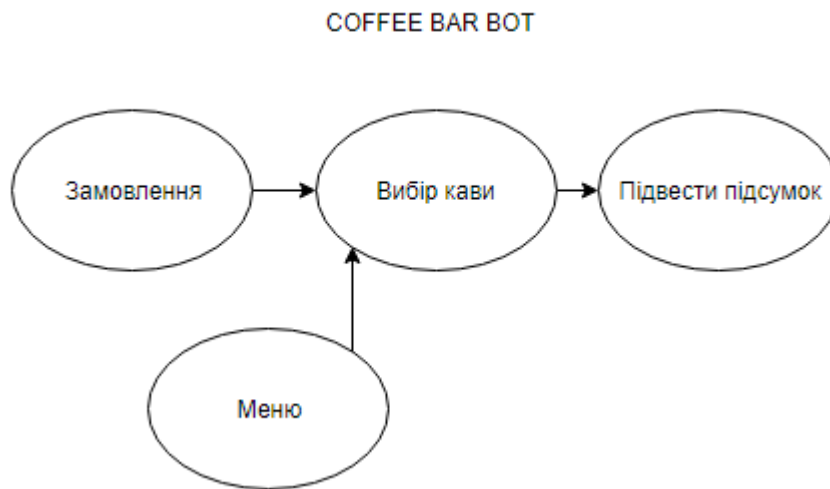


Рисунок 4.2 – План чат-бота COFFEE BAR

Для побудови моделі потрібно використати API ChatBot, Спочатку потрібно перейти та увійти у свій обліковий запис. Якщо ви ще не створили свій обліковий запис, виконайте ці дії. Після успішного входу ви повинні побачити свою інформаційну панель (рисунок 4.3).

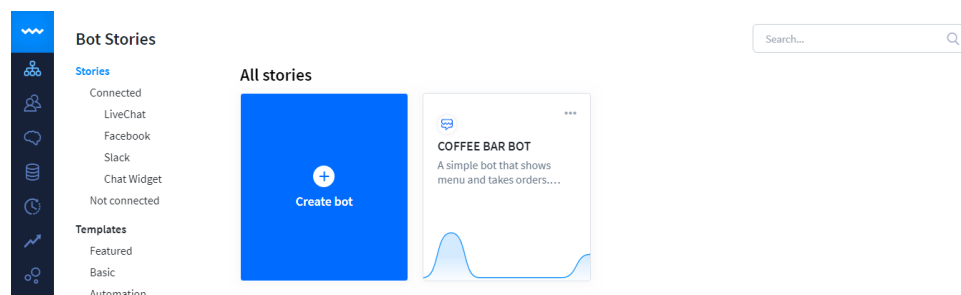


Рисунок 4.3 – Інформаційна панель API ChatBot

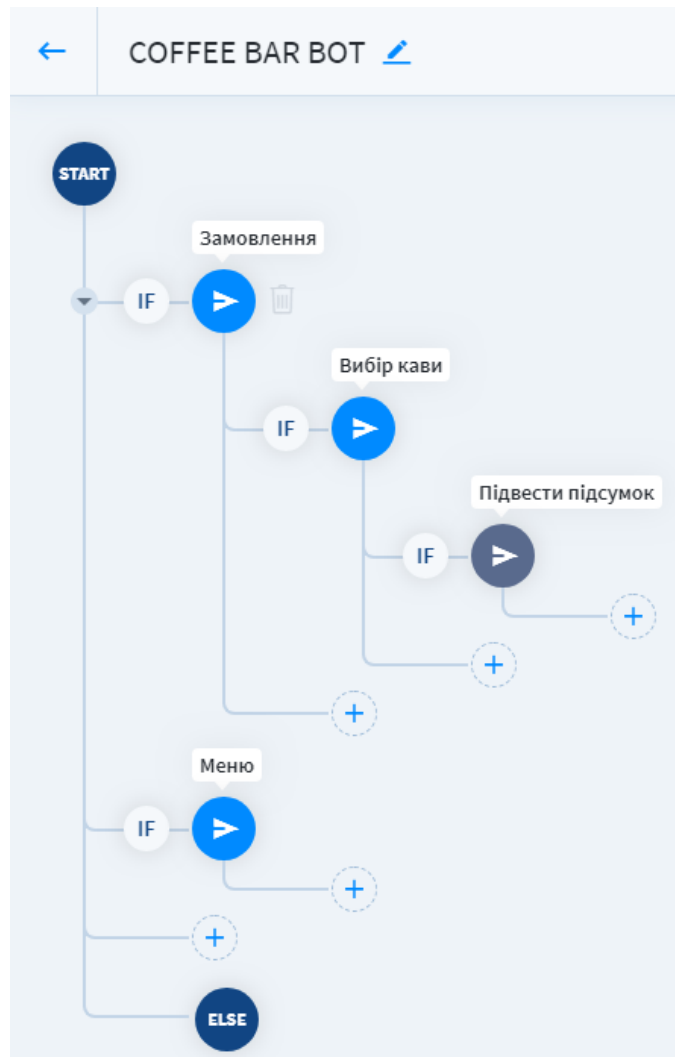


Рисунок 4.4 – Історія умов та відповідей прикладної програми COFFEE BAR BOT

Далі створюємо «Історію» на рисунку 4.4.

Додавши кнопки та картинки для боту отримуємо досить цікавого бота, який здатний приймати замовлення, розпізнавати важливі для нашого бізнесу слова і підтверджувати це. Тепер додаю кнопку дії, яка використовуватиметься для замовлення кави. Ви можете бачити, що кнопка може здійснювати дзвінок, відкривати веб-сторінку, переходити до іншої взаємодії або безпосередньо передавати повідомлення чатові.

Також можливо ввести тип кави в поле після зворотного зв'язку, щоб бот знав, яку каву замовили.

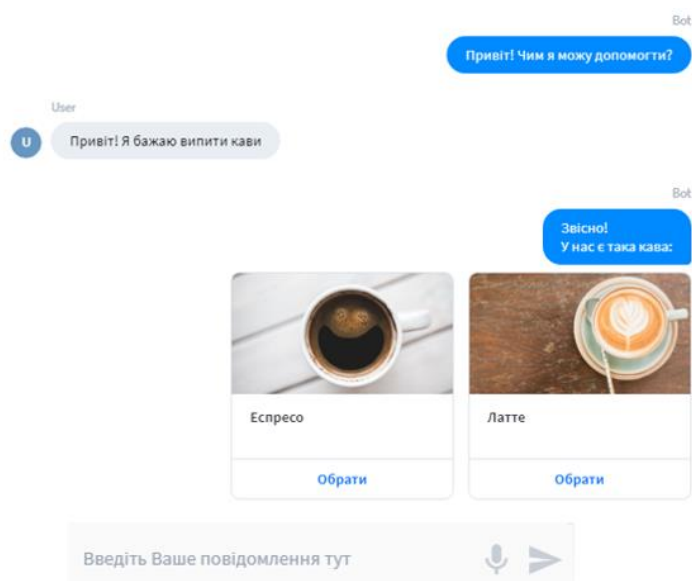


Рисунок 4.5 – Прикладна програма COFFEE BAR BOT в API ChatBot

Користувач може натиснути кнопки або просто ввести тип кави у вікно чату (рисунок 4.6). Йому залишиться лише підтвердження замовлення.

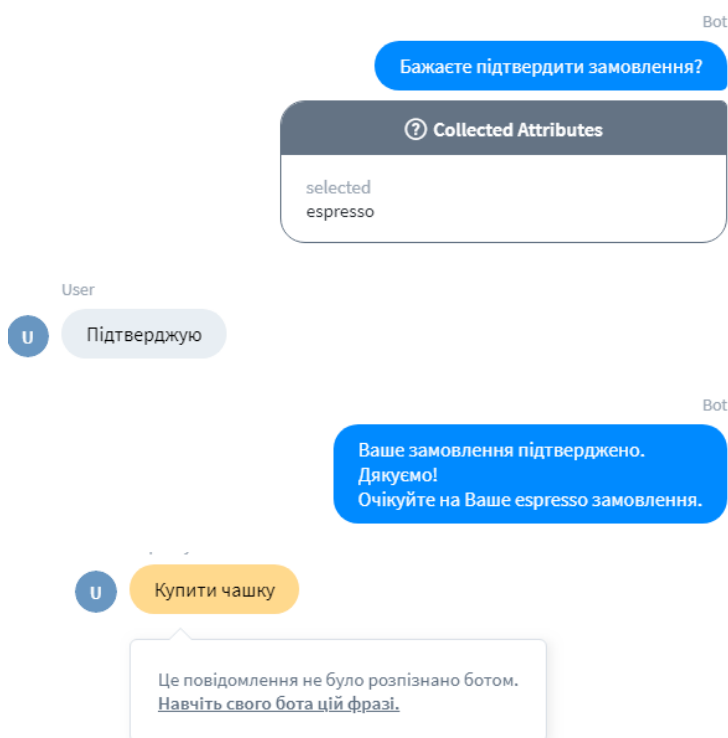


Рисунок 4.6 – Підтвердження замовлення та різка зміна теми

Якщо користувач змінить тему спілкування, то він отримає повідомлення про те що бот не зрозумів, але користувач може навчити бота, обравши до якої категорії віднести фразу, введену користувачем.

Для того, щоб інтегрувати нашого бота потрібно отримати API месендера та додати його до коду.

Власне Bot API – це те, що дозволяє легко та зручно створювати чат-ботів. Цей API дозволяє підключати ботів до системи будь-якого месенджера. З точки зору системи месенджерів, боти – це спеціальні облікові записи, для яких не потрібно встановлювати додатковий номер телефону. Ці облікові записи служать інтерфейсом для коду, що працюватиме на нашому сервері. Ми можемо розглянути розмову в форматі JSON (рисунок 4.7).

```

{
  "date": "2019-12-15T13:38:40.493Z",
  "requestId": "ba49d4e3-443c-4c85-a16b-5e44e0a376fc",
  "source": "app",
  "resolvedQuery": "Привіт! Я бажую випити кави",
  "confidence": 0.5,
  "score": 0.76,
  "lifespan": 2,
  "incomplete": false,
  "storyId": "5deef5e311addd000788535c",
  "interaction": {
    "id": "5deef5e311addd000788535e",
    "name": "Замовлення",
    "action": ""
  },
  "parameters": {},
  "fulfillment": [
    {
      "type": "text",
      "delay": 2000,
      "message": "Звісно!\nУ нас є така кава:"
    },
    {
      "type": "cards",
      "delay": 3000,
      "elements": [
        {
          "title": "Еспресо",
          "subtitle": "",
          "imageUrl": "https://images.pexels.com/photos/6347/coffee-cup-workin",
          "buttons": [
            {
              "type": "postback",
              "title": "Обрати",
              "value": "espresso",
              "id": "92a62210-c9be-43e4-b63b-ac95bd756531-ba49d4e3"
            }
          ]
        },
        {
          "title": "Латте",
          "subtitle": "",
          "imageUrl": "https://images.pexels.com/photos/347144/pexels-photo-34",
          "buttons": [
            {
              "type": "postback",
              "title": "Обрати",
              "value": "latte",
              "id": "3434638c-023a-4dc0-9c3b-8e519e60a08f-ba49d4e3"
            }
          ]
        }
      ]
    }
  ]
}

```

Рисунок 4.7 – Розмова в форматі JSON

В API підтримуються GET і POST HTTP запити. GET-запит використовується за необхідності отримання певної інформації від API, а GET-запит – при необхідності передати якусь інформацію засобами Bot API

#### 4.4 Висновки до розділу

Вирішуючи, який чат-бот підходить саме вам, в першу чергу уявіть себе користувачем і подумайте про значення, яке вони намагаються отримати від вашого бота. Чи буде контекст бесіди суттєво впливати на це значення? Якщо ні, то, вільно використовуйте будь-яку структуру бота [28].

Інша річ, яку слід врахувати – це ваша цільова база користувачів та їх переваги для UX. Деякі користувачі можуть віддати перевагу чат-боту з візуальними кнопками меню, а не відкритим діалогам, іншим навпаки потрібно прямо задавати питання боту.

Правильний чат – це той, який найкраще відповідає цінній пропозиції, яку ви намагаєтеся передати своїм користувачам. Що до поєднання підходів чат-ботів, то можна сказати – система підвищила точність та швидкість відповідей на питання користувачів.



## ВИСНОВКИ

Бот є одним з найбільш перспективних нововведень для електронної комерції. Багато компаній уже використовують цю технологію для збору інформації, продажів та зворотного зв'язку з користувачами. Однак отримати вигоду від таких впроваджень може і малий бізнес. Брак часу, мала кількість співробітників, слабо розвинені технічні потужності – все це позначається на якості і ефективності взаємодії з клієнтами. Боти дозволяють уникнути цих проблем. При розробці ботів доцільно використовувати перевірені методи проектування, уважно вивчати рекомендації месенджерів і проводити тестування на невеликій аудиторії.

В мої магістерській дипломній роботі було проведено аналіз предметної області, а саме історія розробки перших ботів – програми, які здатні вести бесіду з людьми, їхня класифікація на сьогодні. Був проведений аналіз сучасних бот-платформ, таких як WhatsApp, Facebook Messenger, Viber, Telegram, який базувався на статистиці активних користувачів за Липень 2019 року.

Проаналізовано вже існуючих ботів та наведено приклади їх використання. В результаті, було виявлено необхідність популяризації використання розпізнавання команд із повідомлень користувачів, при використанні ботів як засобів роботи із клієнтами. Було формалізовано алгоритм вибору інструментів розробки.

Розглянуто декілька способів використання нейронних мереж при роботі із користувачем, до основних з яких належить застосування їх для аналізу запиту користувача та для слідкування за змістом діалогу. При цьому описано можливості використання як самописних нейронних мереж, так і уже готових бібліотек, заточених під специфічні потреби розробників.

Однією із головних задач було дослідження методів машинного навчання при створенні чат-бота та розробка моделі COFFEE BAR чат-бота на основі розпізнавання ключових слів, меню / кнопок та контексту.

На базі описаних принципів та із використанням описаних технологій було створено прикладну програму для спілкування із користувачем, який дозволяє виділяти ключові слова із його повідомлень та надавати по ним розширену довідку. Ця розробка є доказом концепції такого підходу до створення ботів для месенджерів та ілюструє можливу швидкість та гнучкість розробки, точність та швидкість відповідей на питання.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Державний стандарт України. ДСТУ 3008-2016. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – К.: Держстандарт України, 1995. – 36 с.
2. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 Комп'ютерні науки (освітня програма «Інформаційні управляючі системи та технології» освітньо-кваліфікаційного рівня «магістр» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2019. – 24 с.
3. А.М. Turing. Computing Machinery and Intelligence [Електронний ресурс]/ Mind. – 1959. Vol. 59, N. 236. – P. 433-460. URL: [www.jstor.org/stable/2251299](http://www.jstor.org/stable/2251299) (дата звернення: 20.10.2019).
4. M. Murgia. Can Facebook Messenger Kill Off Apps [Електронний ресурс]/ The Telegraph, 15 Nov. 2015. – Режим доступу: [www.telegraph.co.uk/technology/facebook/11996896/Can-Facebook-Messenger-kill-off-apps.html](http://www.telegraph.co.uk/technology/facebook/11996896/Can-Facebook-Messenger-kill-off-apps.html) (дата звернення: 20.10.2019).
5. Chatbotsmagazine.com [Електронний ресурс]/ Chatbot Report 2018: Global Trends and Analysis – Режим доступу: <https://chatbotsmagazine.com/chatbot-report-2018-global-trends-and-analysis-4d8bbe4d924b> (дата звернення 22.10.2019 р.).
6. Рейтинг меседжерів 2018 [Електронний ресурс]/ Сервис сравнения и выбора приложений и сервисов в сфере IT. – 2018. – Режим доступу: <https://coba.tools/compilation/reiting-messendzherov-2018> (дата звернення 22.10.2019 р.).
7. Statista.com [Електронний ресурс]/ Most popular global mobile-messenger apps/ – Режим доступу:

<https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/> – Загол. з екрану (дата звернення 22.10.2019 р.).

8. A.L.I.C.E. Artificial Intelligence Foundation [Електронний ресурс] – Режим доступу: <http://www.alicebot.org> (дата звернення 22.10.2019 р.).

9. OSP.RU [Електронний ресурс]/ OSP.RU – Режим доступу: <https://www.osp.ru/os/2018/01/13053935/> – Загол. з екрану (дата звернення 22.10.2019 р.).

10. PROMODO [Електронний ресурс]/ PROMODO – Режим доступу: <https://promodo.ua/ua/blog/kak-ispolzovat-chat-botov-v-e-commerce.html#gref> – Загол. з екрану (дата звернення 22.10.2019 р.).

11. Официальный сайт Telegtam [Електронний ресурс]/ API – Режим доступа: <https://core.telegram.org/api> – Загол. з екрану (дата звернення 22.10.2019 р.).

12. Введение в JSON [Електронний ресурс] / JSON – Режим доступу: <https://www.json.org/json-ru.html> (дата звернення 01.11.2019).

13. How to make a responsive telegram bot [Електронний ресурс] – Режим доступу: <https://www.sohamkamani.com/blog/2016/09/21/making-a-telegram-bot/> (дата звернення 22.10.2019 р.).

14. Как заработать на создании чат-ботов: опыт белорусского стартапа BotCube [Електронний ресурс]/ DEVby – Режим доступу: <https://dev.by/lenta/main/kak-zarabotat-na-sozdaniichat-botov-opyt-belorusskogo-startapa> (дата звернення 01.11.2019).

15. Википедия [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/> (дата обращения 01.11.2019).

16. Технологии создания и применения чат ботов [Електронний ресурс]/ Новые технологии – Режим доступу: <https://cyberleninka.ru/article/v/tehnologii-sozdaniya-i-primeneniya-chatbotov> (дата звернення 01.11.2019).

17. Mindhacking [Електронний ресурс] – Режим доступу: <https://github.com/owocki/mindhacking/blob/master/README.md> (дата звернення 01.11.2019)..

18. Celebi Tutorial: Neural Networks and Pattern Recognition Using MATLAB [Електронний ресурс] – Режим доступу: [https://www.byclb.com/TR/Tutorials/neural\\_networks/ch1\\_1.htm](https://www.byclb.com/TR/Tutorials/neural_networks/ch1_1.htm) (дата звернення 01.11.2019).

19. Типові архітектури нейронних мереж [Електронний ресурс] – Режим доступу: <https://studfiles.net/preview/5740125/page:4/> (дата звернення 01.11.2019).

20. Neural-Networks-part-2 [Електронний ресурс] – Режим доступу: <http://www.marekrei.com/blog/neural-networks-part-2-the-neuron/> (дата звернення 01.11.2019).

21. Neural-Networks-part-3 [Електронний ресурс] – Режим доступу: <http://www.marekrei.com/blog/neural-networks-part-3-network/> (дата звернення 01.11.2019).

22. Фанифатьева А. Д. АВТОМАТИЧЕСКИЙ АНАЛИЗ ТОНАЛЬНОСТИ РЕЦЕНЗИЙ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ TENSORFLOW [Електронний ресурс] / Фанифатьева А. Д. – Режим доступу: <http://library.eltech.ru/files/vkr/2017/bakalavri/3382/2017%D0%92%D0%9A%D0%A0338212%D0%A4%D0%90%D0%9D%D0%98%D0%A4%D0%90%D0%A2%D0%AC%D0%95%D0%92%D0%90.pdf> (дата звернення 01.11.2019).

23. Hackernoon.com [Електронний ресурс] – Режим доступу: <https://hackernoon.com/deep-learning-chatbot-everything-you-need-to-know-r11jm30bc> (дата звернення 01.11.2019)

24. Обробка природної мови [Електронний ресурс] – Режим доступу: [https://uk.wikipedia.org/wiki/Обробка\\_природної\\_мови](https://uk.wikipedia.org/wiki/Обробка_природної_мови) (дата звернення 01.11.2019).

25. Otus.ru [Електронний ресурс]/ Модель seq2seq в машинному перекладі – Режим доступу: <https://otus.ru/nest/post/285/> (дата звернення 01.11.2019).

26. Skymind [Електронний ресурс] – Режим доступу: <https://skymind.ai/wiki/word2vec> (дата звернення 01.11.2019).

27. Как создать чат-бота с нуля на Python: подробная инструкция [Електронний ресурс] – Режим доступу: <https://www.dcstyling.ru/robots/kak-sozdat-chat-bota-s-nulia-na-python-podrobnaia-instruktsiia/> (дата звернення 01.11.2019).

28. Левикин В.М., Кудрявцева М.С. Візуальні мови та середовища розробки застосувань: Навч. посібник. – Харків: ХНУРЕ, 2011. – 168 с.