

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)

Кафедра _____ Штучного інтелекту _____
(повна назва)

АТЕСТАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

_____ Розробка інтелектуального персонального помічника в управлінні _____
_____ витратами часу на основі методів машинного навчання _____
(тема)

Виконав:
студент 2 курсу, групи _____ СШМ-19-1 _____
_____ Рогач М.А. _____
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки _____
(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту _____
(повна назва спеціалізації)

Керівник _____ к.т.н., доц. Золотухін О.В. _____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

_____ В.О. Філатов _____
(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)
Кафедра _____ Штучного інтелекту _____
(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 _____ Системи штучного інтелекту (СШІ) _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____ Рогач Марії Андріївни _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Розробка інтелектуального персонального помічника в управлінні витратами часу на основі методів машинного навчання _____

затверджена наказом університету від 30 жовтня 2020 р. № 1497 ст _____

2. Термін подання студентом роботи до екзаменаційної комісії 16 грудня 2020 р.

3. Вихідні дані до роботи _____ Науково-технічні публікації, статті, дані Інтернет-джерел та відомих наукових проектів, Python documentation _____

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі _____

2) Дослідження існуючих ІПА, що використовують техніки тайм-менеджменту _____

3) Дослідження застосування технологій для управління витратами часу _____

4) Реалізація інтелектуальної системи для управління витратами часу _____

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

Рисунок 1 – Потік даних у запропонованій моделі _____

Рисунок 2 – Зображення процесу вилучення ключових слів _____

Рисунок 3 – Схема попередньої обробки тексту _____

Рисунок 4 – Схема реалізації запропонованого підходу _____

Рисунок 5 – Графік двадцяти ключових слів за частотою зустрічей _____

Рисунок 6 – Графік двадцяти біграм за частотою зустрічання _____

Рисунок 7 – Графік двадцяти триграм за частотою зустрічання _____

Рисунок 8 – Список ключових слів _____

Рисунок 9 – Скріншот використаної виборки _____

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	к.т.н., доц. Золотухін О.В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	11.11.2020	виконано
2	Аналіз предметної галузі	12.11.2020 – 15.11.2019	виконано
3	Постановка задачі та узгодження з керівником	16.11.2020 – 18.11.2020	виконано
4	Дослідження існуючих ПІА що використовують	19.11.2020 – 22.11.2020	виконано
5	Дослідження застосування технологій, методів і	23.11.2020 – 24.11.2020	виконано
6	Практична реалізація	25.11.2020 – 26.11.2020	виконано
7	Підготовка пояснювальної записки та графічного	27.11.2020 – 01.12.2020	виконано
8	Надання пояснювальної записки на перевірку	02.12.2020	виконано
9	Захист перед ЕК	16.12.2020	

Дата видачі завдання 11 листопада 20 20 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис) _____
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 112 с., 1 табл., 26 рис., 2 дод., 41 джерело.

АНСАМБЛЬ КЛАСИФІКАТОРІВ, ВИЯВЛЕННЯ КЛЮЧОВИХ СЛІВ, ІНТЕЛЕКТУАЛЬНИЙ ПЕРСОНАЛЬНИЙ АСИСТЕНТ, МАШИННЕ НАВЧАННЯ, ПРИРОДНА ОБРОКА МОВЛЕННЯ, ТАЙМ-МЕНЕДЖМЕНТ, ШТУЧНИЙ ІНТЕЛЕКТ

Метою атестаційної роботи є розробка інтелектуального персонального асистента (ПА) в управлінні витратами часу, дослідження технологій, методів і засобів машинного навчання, що використовуються в ПА.

Об'єктом дослідження є ПА.

Предметом дослідження є застосування технологій, методів і засобів машинного навчання в ПА, засоби оброки природного мовлення, методи вилучення ключових слів, алгоритми класифікації, методи побудови ансамблів класифікаторів.

Методи дослідження – методи системного та об'єктно-орієнтованого аналізу.

У рамках наукових результатів дослідження представлено метод розробки інтелектуальної системи з використанням технік тайм-менеджменту, представлено опис проекту CALO, представлено рішення, що усуває недоліки розглянутого проекту.

У рамках практичних результатів наведена аналітична реалізація інтелектуальної системи, реалізація задачі виявлення ключових слів, реалізація задачі прогнозування часових витрат, що необхідні для виконання задачі. Також наведено результати практичних експериментів та виконано їх аналіз.

РЕФЕРАТ

Пояснительная записка: 112 с., 1 табл., 26 рис., 2 пр., 41 источник.

АНСАМБЛЬ КЛАССИФИКАТОРОВ, ИЗВЛЕЧЕНИЕ КЛЮЧЕВЫХ СЛОВ, ИНТЕЛЛЕКТУАЛЬНЫЙ ПЕРСОНАЛЬНЫЙ АССИСТЕНТ, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, МАШИННОЕ ОБУЧЕНИЕ, ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА, ТАЙМ-МЕНЕДЖМЕНТ

Целью аттестационной работы является разработка интеллектуального персонального ассистента (ИПА) в управлении затратами, исследования технологий, методов и средств машинного обучения, используемых в ИПА.

Объектом исследования является ИПА.

Предметом исследования является применение технологий, методов и средств машинного обучения в ИПА, обработки естественного языка, извлечение ключевых слов, алгоритмы классификации и методы построения ансамблей классификаторов.

Методы исследования – методы системного и объектно-ориентированного анализа.

В рамках научных результатов исследования представлены метод разработки интеллектуальной системы с использованием техник тайм-менеджмента, представлено описание проекта CALO, представлены решения, устраняют недостатки рассматриваемого проекта.

В рамках практических результатов приведена аналитическая реализация интеллектуальной системы, реализация задачи выявления ключевых слов, реализация задачи прогнозирования временных затрат, необходимых для выполнения задачи. Также приведены результаты практических экспериментов и выполнено их анализ.

ABSTRACT

The certification work: 112 p., 1 tab., 26 fig., 2 add., 41 references.

ARTIFICIAL INTELLIGENCE, ENSEMBLE OF CLASSIFIERS, INTELLECTUAL PERSONAL ASSISTANT, KEYWORDS EXTRACTION, MACHINE LEARNING, NATURAL LANGUAGE PROCESSING, TIME MANAGEMENT

The purpose of the certification work is to develop an intelligent personal assistant (IPA) in time management, research of technologies, methods and tools of machine learning used in IPA.

The object of the study is IPA.

The subject of research is the application of technologies, methods and means of machine learning in IPA, means of arrears of natural speech, methods of extracting keywords, classification algorithms, methods of building ensembles of classifiers.

Research methods – methods of systematic and object-oriented analysis.

As part of the scientific results of the study, a method of developing an intelligent system using time management techniques, a description of the CALO project, a solution that eliminates the shortcomings of the project.

The practical results include the analytical implementation of the intelligent system, the implementation of the task of keyword detection, the implementation of the task of forecasting the time required to perform the task. The results of practical experiments are also presented and their analysis is performed.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	9
Вступ.....	10
1 Аналіз предметної області та постановка задачі	12
1.1 Загальний огляд інтелектуальних особистих асистентів	12
1.2 Еволюція ІПА	14
1.3 Аналіз ролі штучного інтелекту в розвитку ІПА.....	17
1.4 Аналіз типових характеристик ІПА	21
1.5 Типи ІПА.....	24
1.6 Аналіз управління витратами часу	28
1.6.1 Аналіз існуючих технік управління витратам часу	31
1.7 Постановка задачі.....	37
2 Дослідження існуючих ІПА, що використовують техніки тайм-менеджменту	39
2.1 Дослідження існуючих рішень	39
2.1.1 Дослідження PASSAT	39
2.1.2 Дослідження MARGEN	43
2.1.3 Дослідження LinTO.....	47
2.2 Висновки дослідження існуючих ІПА	50
2.3 Критика ІПА	51
2.4 Огляд проекту CALO	52
3 Дослідження застосування технологій для управління витратами часу	57
3.1 Задача розпізнавання мовлення.....	58
3.2 Задача природної обробки мовлення (ПОМ).....	64
3.2.1 Задача виявлення ключових слів.....	68
3.3 Задача прогнозування часових витрат, необхідних на виконання задачі	70

4 Реалізація інтелектуальної системи для управління витратами часу	74
4.1 Вирішення задачі виявлення ключових слів.....	76
4.2 Вирішення задачі прогнозування часових витрат, необхідних на виконання задачі	81
4.3 Результати експериментів.....	85
4.4 Практична реалізація на мові Python	87
Висновки.....	96
Перелік джерел посилання	98
Додаток А. Вихідний код.....	102
Додаток Б. Відомість атестаційної роботи.....	112

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

ІС – інформаційна система;

НБА – наївний алгоритм Байєса;

ПОМ – природна обробка мовлення;

РНН – рекурентна нейронна мережа;

ПА – інтелектуальний персональний асистент;

ШІ – штучний інтелект.

ВСТУП

Час – це один з найцінніших ресурсів, що є у кожної людини. Незалежно від віку, статі, середньомісячного доходу, релігії або професії людини, у кожного є одна й та сама кількість часу. Але цю кількість часу можна використовувати дуже по-різному. Саме тому дуже важливим є те, як саме можна використовувати час ефективно, з максимальною користю, щоб уникнути відчуття нестачі часу та отримати професійний або особистісний розвиток.

Типовий працівник знань має широкий спектр відповідальності та повинен жонглювати широким набором завдань. Роблячи це, він повинен підтримувати обізнаність про терміни та ресурси, а також відстежувати поточну діяльність та нову інформацію, яка може вплинути на його цілі та продуктивність. Значна частина його роботи потребує координації та співпраці з широким колом людей, як всередині, так і поза організацією. Оскільки організації прагнуть підвищити економічну ефективність та ефективність, навантаження для багатьох працівників знань зростає. Кінцевим результатом є високий рівень когнітивних перевантажень на робочому місці.

В цій роботі описано інтелектуального особистого асистента, який був розроблений, щоб допомогти зайнятому працівникові знань в управлінні часовими витратами та виконанні завдань. Архітектура системи мотивована додатковою метою: звільнення користувача від рутинних завдань, що дозволить зосередитись на завданнях, які критично потребують навичок вирішення проблем користувача.

Для виконання атестаційної роботи необхідно:

- проаналізувати поставлену задачу розробки інтелектуального персонального помічника в управлінні витратами часу на основі методів машинного навчання;

- проаналізувати існуючих асистентів, що використовують техніки тайм-менеджменту;
- проаналізувати основні завдання машинного навчання, необхідні для розробки;
- сформулювати необхідні вимоги до системи;
- створити систему у відповідності з заданими вимогами.

Актуальність обраної теми полягає в тому, що в сучасних умовах тайм-менеджмент є невід'ємною частиною діяльності будь-якої успішної людини або компанії. Під тайм-менеджментом розуміється технологія організації часу і підвищення ефективності його використання.

Результати роботи оформлені у вигляді пояснювальної записки до атестаційної роботи. У першому розділі виконано аналіз предметної галузі та постановка задачі, другий розділ присвячений дослідженню існуючих рішень, що використовують техніки тайм-менеджменту. В третьому розділі виконано дослідження застосування технологій, методів і засобів машинного навчання для вирішення поставленої задачі. Четвертий розділ присвячено реалізації системи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальний огляд інтелектуальних особистих асистентів

Інтелектуальні особисті асистенти (ІПА) – також відомі як розумні особисті асистенти або персональні цифрові асистенти – це системи, які збирають інформацію від користувача, оцінюють її і дають належну відповідь [1]. ІПА – це програмне забезпечення, яке може виконувати завдання або послуги від імені користувача на основі поєднання введених ним даних, обізнаності про місцезнаходження та, при можливості доступу, інформації з різних Інтернет-джерел (наприклад, погодні умови, затори на дорогах, новини, ціни на акції, роздрібні ціни на конкретні товари тощо). ІПА можуть виконувати різні завдання, запам'ятовувати інформацію, здійснювати пошук в Інтернеті, керувати пристроями тощо. Використовуючи ці здібності ІПА може пропонувати допомогу в наступних основних видах діяльності: управління будинком, розваги, планування, управління контактами, покупки, життя.

Незважаючи на те, що ІПА відносно нещодавно досягли успіху на споживчому ринку, особиста допомога, яку можуть надавати інформаційні системи (ІС), зовсім не нова тема досліджень. У минулому дослідження в галузі штучного інтелекту (ШІ) були зосереджені на експертних системах. Однак технічна еволюція, та створення таких інструментів як інфраструктура хмарних сервісів, обробка природних мов, семантичні міркування, розпізнавання голосу та синтез голосу, відкрили шлях для сучасних ІПА, таких як Siri від Apple, Cortana від Microsoft, Samsung від Vixby, Alexa від Amazon, Google Асистент від Google а також різноманітні чат-боти. Ці інтелектуальні сервісні системи взаємодіють з користувачем за допомогою природної мови та пропонують безліч можливостей надання послуг та інформації для зменшення зусиль та складності повсякденних завдань користувачів.

Також ІПА можна описати як спеціалізований інтелектуальний штучний агент, який допомагає користувачам здійснювати свою діяльність, як посередника між людьми та іншими агентами в мультиагентному середовищі. Термін «агент» має на меті зазначити, що ІПА як автономний суб'єкт здатний сприймати та вживати дії у своєму середовищі для досягнення певної мети [2], а саме допомагати користувачеві у виконанні конкретних завдань. Крім того, ІПА як агент (наприклад, Alexa) може взаємодіяти з іншими агентами, такими як технічні агенти (наприклад, розумний холодильник). У концепції багатоагентності ІПА складається з різних рівнів, кожен з яких виконує певне завдання (наприклад, агент інтерфейсу, агент взаємодії, агент транзакцій). Так користувач взаємодіє з агентом інтерфейсу, який делегує більш конкретні завдання іншим типам агентів [3]. У цьому контексті ІПА служить єдиною, повсюдною та простою точкою доступу до інтелектуальної інфраструктури послуг.

ІПА діє від імені користувача під час збору інформації, автоматизації складних завдань та співпраці з іншими ІПА, що призводить до соціальної взаємодії між користувачем та ІПА. Основною метою ІПА є покращення сприйняття, пізнання та / або здібностей користувача [4]. З соціотехнічної точки зору та порівняно з іншими класами інформаційних систем, новизна ІПА полягає у двох основних аспектах: способі взаємодії користувачів із пристроєм, а також знаннями про людську поведінку.

У деяких випадках ІПА призначені виключно для розважальних цілей. Деякі інтелектуальні асистенти здатні інтерпретувати людську мову та реагувати синтезованими голосами. Користувачі можуть задавати асистентам запитання, керувати домашніми автоматизованими пристроями та відтворювати мультимедіа за допомогою голосу, а також керувати іншими основними пристроями, такими як електронна пошта, списки справ та календарі за допомогою голосових команд [5].

Для того, щоб задовольняти інтереси користувача, ідеальний ІПА:

- використовує різноманітні технології;
- покращує щоденну взаємодію, пристосовуючись як до середовища, так і до діяльності користувача;
- використовує характерні інтерфейси та відображає емоційні стани (наприклад, гумор), щоб зробити можливим спілкування схожим на спілкування з людиною;
- використовує датчики та зібрані дані;
- має різні входи та типи виведення, підвищуючи комфорт взаємодії користувача.

1.2 Еволюція ІПА

«Radio Rex» був першим пристроєм, що реагував на голос. Він був випущен в 1911 році [6]. Це була іграшкова собака, яка виходила зі свого будинку, коли називають її ім'я.

У 1952 році компанія Bell Labs представила «Одрі», машину з автоматичним розпізнаванням цифр. Машина займала релейну стійку заввишки шість футів, споживала значну потужність, мала потоки кабелів і демонструвала безліч проблем з технічним обслуговуванням, пов'язаних зі складною схемою вакуумних труб. Вона могла розпізнавати основні одиниці мови, фонемі. Машина вміла розпізнавати цифри, вимовлені призначеними спікерами. «Одрі» можна було використовувати для голосового набору, але в більшості випадків кнопковий набір був дешевшим та швидшим.

Ще одним раннім інструментом, який отримав можливість цифрового розпізнавання мови, був голосово-активований калькулятор IBM Shoebox, представлений широкому загалу під час Всесвітньої виставки в Сіетлі 1962

року після первинного запуску на ринок у 1961 р. Цей ранній комп'ютер, розроблений майже за 20 років до впровадження першого персонального комп'ютера ІВМ у 1981 році, міг розпізнати 16 вимовлених слів та цифри від 0 до 9.

Перша комп'ютерна програма для обробки природної мови або чат-бот «Eliza» була розроблена професором Массачусетського технологічного інституту Джоозефом Вайценбаумом у 1960-х. Вона була створена для того, щоб продемонструвати, що спілкування між людиною та машиною може існувати. «Eliza» використовувала методологію узгодження та заміни шаблонів у сценарії відповідей для імітації розмови, що створювало ілюзію розуміння з боку програми [7].

Наступна віха у розвитку технології розпізнавання голосу була досягнута в 1970-х роках в Університеті Карнегі-Меллона в Пітсбурзі, штат Пенсільванія, за суттєвої підтримки Міністерства оборони США та агенції оборонних дослідницьких проектів DARPA, що фінансувала п'ять років програми досліджень розуміння мови, спрямованих на те, щоб досягти мінімального словника з 1000 слів. Результатом стала «Harpy», що засвоїла близько 1000 слів, словниковий запас трирічного віку, також вона могла розуміти речення. Машина могла обробляти мовлення, яке слідує за попередньо запрограмованим словниковим запасом, вимовою та граматичними структурами, щоб визначити, які послідовності слів мають сенс разом.

У 1986 році було розроблено «Tangora» – оновлення Shoebox. Tangora представляла собою друкарську машинку, що розпізнавала голос. Даний пристрій було названо на честь найшвидшої на той час машиністки у світі, він мав словниковий запас у 20000 слів і використовував передбачення для визначення найбільш вірогідного результату на основі сказаного в минулому. Підхід ІВМ базувався на моделі Прихованого Маркова, яка додає статистику до методів цифрової обробки сигналів. Метод дозволяє передбачити найбільш

імовірні фонемі, які слідуватимуть за даною фонемою. Фонема – це одиниця звукового ладу мови, що служить для впізнання і розрізнення значущих одиниць – морфем, до складу яких вона входить в якості мінімального сегментного компонента, а через них – і для впізнання і розрізнення слів [8]. Тим не менше, кожен оратор повинен був окремо навчити друкарську машинку розпізнавати свій голос і робити паузи між кожним словом [9].

Після вдосконалення технології, пристрої у 1990-х роках могли виконувати планування виконання задач, записувати адреси та номери, встановлювати зустрічі та складати списки справ. Однак це були лише дуже прості форми того, що сьогодні називають інтелектуальним особистим помічником. Їх можна оцінити як пристрої-предки сучасних інноваційних ША [10].

В 1990-х технологія цифрового розпізнавання мовлення стала використовуватися в персональному комп'ютері. Значно пізніше випуск на ринок першого смартфона IBM Simon в 1994 році заклав основу для розумних віртуальних помічників, якими користувачі знають їх сьогодні.

У 1997 році програма Dragon's Naturally Speaking могла розпізнавати та транскрибувати природну людську мову без пауз між кожним словом у документ зі швидкістю 100 слів на хвилину. Версія програми Naturally Speaking все ще доступна для завантаження, і вона використовується і сьогодні, наприклад, багатьма лікарями в США та Великобританії для документування своїх медичних записів.

У 2001 році Colloquis публічно запустив SmarterChild на таких платформах, як AIM та MSN Messenger. Хоча повністю текстовий SmarterChild міг грати в ігри, перевіряти погоду, шукати факти та до певної міри спілкуватися з користувачами.

Після того, як ШІ почав швидко розвиватися, еволюція віртуальних помічників набрала обертів, роблячи їх більш інтелектуальними. Досягнення

технік машинного навчання підвищило здатність ІПА до навчання. Використання датчиків, хмарного підключення та технології Інтернету речей значно підвищило ефективність пристроїв, що дозволило зв'язати їх з іншими пристроями. Завдяки всім цим інноваційним технологіям ІПА почали більш широко використовуватися у повсякденному житті. Однак розвиток всіляких технологій не має обмежень, і в певних сферах все ще існують певні проблеми.

Першим сучасним цифровим віртуальним помічником, встановленим на смартфоні, була Siri, яка була представлена як функція iPhone 4S 4 жовтня 2011 року. Apple Inc. розробила Siri після придбання в 2010 році Sri Inc., відділення SRI International, яка є науково-дослідним інститутом, що фінансується DARPA та Міністерством оборони США. Його метою було допомогти у виконанні таких завдань, як надсилання текстових повідомлень, здійснення телефонних дзвінків, перевірка погоди або встановлення будильника. З часом було розроблено систему рекомендацій щодо ресторанів та вказівок щодо руху.

У листопаді 2014 року Amazon анонсувала Alexa.

У квітні 2017 року Amazon випустила послугу для побудови розмовних інтерфейсів для будь-якого типу віртуального помічника або інтерфейсу.

1.3 Аналіз ролі штучного інтелекту в розвитку ІПА

Нещодавній активний розвиток ІПА безпосередньо пов'язаний із прогресом у технологіях штучного інтелекту. Штучний інтелект можна визначити кількома способами. Грубо кажучи, це здатність машини імітувати когнітивну поведінку людини.

Штучний інтелект – це широка галузь комп'ютерних наук, яка займається побудовою розумних машин, здатних виконувати завдання, які зазвичай потребують людського інтелекту. Штучний інтелект – це одна з

об'ємних областей науки. В даний час тематика штучного інтелекту охоплює величезний перелік наукових напрямків, починаючи з таких завдань загального характеру, як навчання і сприйняття, і закінчуючи такими спеціальними задачами, як гра в шахи, доказ математичних теорем, твір поетичних творів і діагностика захворювань. У штучному інтелекті систематизуються і автоматизуються інтелектуальні завдання і тому ця область стосується будь-якої сфери інтелектуальної діяльності людини. У цьому сенсі штучний інтелект є універсальною науковою областю [2]

Штучний інтелект може бути розкладено на підмножину характеристик:

- автономність – здатність машини діяти самостійно, сприймаючи навколишнє середовище та вживаючи заходів, щоб максимізувати свої шанси на успіх у визначеному завданні. Розвиток цих можливостей породив сферу автоматизації та таких додатків, як робототехніка або Інтернет речей;

- вирішення проблем – здатність машини за заданих вхідних даних отримувати результати, які максимізують заздалегідь визначені критерії успіху. Розвиток цих можливостей зазвичай вважається полем алгоритміки;

- машинне навчання – здатність покращуватися з часом при виконанні певного завдання. Або інакше кажучи, здатність машини самостійно прогресувати при вирішенні проблем. Це сфера, яку сьогодні зазвичай вважають провідною в штучному інтелекті. Активний розвиток машинного навчання зробив можливим розвиток ІПА, що походить від специфічного підходу до машинного навчання: нейронні мережі та, більш конкретно, глибоке навчання нейронних мереж. Дійсно, цей новий підхід приносить користь багатьом різним задачам, які лежать в основі обробки природної мови.

Хоча штучний інтелект – це широка наука, що імітує людські здібності, машинне навчання – це підгрупа ШІ, яка тренує машину, як вчитися.

Машинне навчання – це наука програмування комп'ютерів таким чином, щоб вони могли вчитися на даних. Машинне навчання – галузь дослідження, яка надає комп'ютерам можливість вчитися без явного програмування [11].

Через нові обчислювальні технології машинне навчання сьогодні не схоже на машинне навчання минулого. Воно народилося завдяки розпізнаванню образів та теорії, згідно з якою комп'ютери можуть вчитися, не будучи запрограмованими на виконання конкретних завдань. Дослідники, які цікавляться штучним інтелектом, хотіли перевірити, чи можуть комп'ютери вчитися на даних. Ітераційний аспект машинного навчання важливий, оскільки, коли моделі отримують нові дані, вони можуть самостійно адаптуватися. Вони вчаться на попередніх обчисленнях для отримання надійних, повторюваних рішень та результатів. Це наука не нова, але така, що набрала нових обертів. Відновлення інтересу до машинного навчання зумовлене тими ж факторами, що зробили обробку даних та байєсівський аналіз більш популярними, ніж будь-коли. Такі речі, як збільшення обсягів та різновидів доступних даних, обчислювальна обробка, яка є більш дешевою та потужною, та доступне зберігання даних [2].

Усе це означає, що можна швидко та автоматично створювати моделі, які можуть аналізувати більші, складніші дані та забезпечувати швидші та точніші результати – навіть у дуже великих масштабах. А будуючи точні моделі, організація має більше шансів виявити вигідні можливості або уникнути невідомих ризиків.

За останнє десятиліття машинне навчання дало людям можливість самостійного керування автомобілями, практичного розпізнавання мовлення, ефективного пошуку в Інтернеті та значно покращило розуміння геному людини. Машинне навчання настільки поширене сьогодні, що люди використовують його десятки разів на день, не знаючи про це.

Більшість галузей, що працюють з великими обсягами даних, визнали цінність технології машинного навчання. Отримуючи інформацію з цих даних – часто в реальному часі – організації можуть працювати ефективніше або отримати перевагу над конкурентами. Розглянемо приклади галузей, в яких активно використовується технологій машинного навчання.

Фінансові послуги. Банки та інші підприємства у фінансовій галузі використовують технологію машинного навчання для двох ключових цілей: виявлення корисної інформації з даних та запобігання шахрайству. Аналіз може визначити інвестиційні можливості або допомогти інвесторам надавати інформацію про вигідність торгівлі в конкретний момент часу. Видобуток даних також може ідентифікувати клієнтів з профілями високого ризику або використовувати кібернагляд, щоб визначити попереджувальні ознаки шахрайства.

Уряд. Державні установи, такі як ті, що відповідають за громадську безпеку або комунальні послуги, мають особливу потребу в машинному навчанні, оскільки вони мають безліч джерел з даними, з якими можна працювати для отримання знань. Наприклад, аналізуючи дані з різноманітних датчиків, визначаються шляхи підвищення ефективності та економії грошей. Машинне навчання також може допомогти виявити шахрайство та мінімізувати викрадення особистих даних.

Охорона здоров'я. Машинне навчання – це швидкозростаюча тенденція у галузі охорони здоров'я завдяки появі переносимих пристроїв та датчиків, які можуть використовувати дані для оцінки стану здоров'я пацієнта в режимі реального часу. Ця технологія також може допомогти медичним експертам проаналізувати дані для виявлення тенденцій, що може призвести до поліпшення діагнозів та лікування.

Роздрібна торгівля. Веб-сайти, що рекомендують товари, які можуть сподобатися конкретному користувачу на основі попередніх покупок,

використовують машинне навчання для аналізу історії покупок. Роздрібні продавці покладаються на машинне навчання для збору даних, їх аналізу та використання для персоналізації досвіду покупок, реалізації маркетингової компанії, оптимізації цін, планування постачання товарів та отримання інформації про клієнтів.

Нафта і газ. Пошук нових джерел енергії. Аналіз мінералів у землі. Прогнозування несправності датчиків. Впорядкування розподілу нафти, щоб зробити її більш ефективною та економічно вигідною. Кількість випадків використання машинного навчання для цієї галузі величезна – і все ще зростає.

Транспортування. Аналіз даних для виявлення закономірностей та тенденцій є ключовим для транспортної галузі, яка покладається на підвищення ефективності маршрутів та прогнозування потенційних проблем для збільшення прибутковості. Аналіз даних та моделювання аспектів машинного навчання є важливими інструментами для компаній-постачальників, громадського транспорту та інших транспортних організацій.

1.4 Аналіз типових характеристик ША

На основі інтелекту та взаємодії як важливих факторів проектування ША, можна виділити конкретні атрибути проектування. Тим самим можна приділили особливу увагу формулюванню характеристик проектування, щоб вони були взаємовиключними.

Характеристиками, що визначають ступінь взаємодії, є:

– режим спілкування: основний спосіб спілкування користувача зі ША та навпаки. Зв'язок базується на введеному користувачем та / або створеному ША тексті, користувацькому та / або синтезованому голосі, датчиках зору, камерах та генерованій анімації, поєднанні голосу та зору;

– тип взаємодії: включає взаємодію користувача з системою, взаємодію системи з користувачем та двонаправлену взаємодію. Взаємодія від користувача до системи означає, що користувач надає дані, які навмисно та свідомо спрямовані до ІІА. Реакція системи може бути несвідомою для користувача. Взаємодія від системи до користувача означає, що ІІА звертається до користувача, щоб створити зміни в середовищі, яких користувач не може свідомо уникнути. У цьому випадку користувач не ставить явний запит заздалегідь, а отримує результат здатності ІІА до пасивного спостереження та осмислення контекстної інформації. Двонаправлена взаємодія означає, що вона надає послуги з комунікаційного обміну [12];

– введення запиту: спосіб, яким користувачі формулюють запити до ІІА. Запити можуть бути заздалегідь визначеними створеними підказками, які користувач повинен знати, щоб викликати бажану дію, запити на природній мові або накопичення даних з датчиків;

– отримання результату: спосіб, яким ІІА формулює відповіді на запити користувачів. ІІА надає візуальну відповідь, якщо реагує за допомогою тексту, зображень, відео, аватарів або будь-якої їх комбінації. Голосовий вихід відноситься до відповідей за допомогою синтезованої мови, оскільки це загальноприйнято для більшості комерційних ІІА, доступних на даний момент. ІІА, що поєднують візуальні та словесні відповіді, такі як розумні колонки з вбудованим екраном, класифікуються як «голос та зір» [13];

– дія: можливість ІІА виконувати послуги на основі введення запиту. Можна широко розрізнити загальну здатність, наприклад, відтворювати музику, встановлювати будильники або керувати розумними побутовими об'єктами як частину більшої системи інтелектуального обслуговування (наявність послуги виконання) та «просту» функціональність,

таку як відповідь на запитання та пошук інформації (відсутність послуги виконання).

Характеристиками, що визначають ступень інтелекту ШІ є:

– домен допомоги: визначає як функціональні можливості, так і моделі знань (тобто семантичні моделі, такі як онтології), які повинні бути реалізовані для надання відповідної допомоги для даного контексту. ШІ може надавати загальну допомогу, наприклад, отримання інформації, пошук в Інтернеті або відтворення музики, або спеціальну допомогу для певних складних завдань користувача або спеціальної групи користувачів [13];

– прийняті команди: забезпечують контроль над поведінкою ШІ. Найпростіша форма – це ручний ввід даних, за яким слідує прості команди, такі як відправлення електронного листа, та складені команди, наприклад, отримувати кожен день в заданий час останню погоду та надсилати її електронною поштою заданому користувачу;

– адаптивність: здатність системи вчитися, інтерпретуючи, як правило, великий обсяг даних, і відповідно адаптувати послуги виконання. Прикладами є вдосконалення функції розпізнавання мови для різних користувачів з часом. ШІ характеризується або статичною поведінкою, якщо надання послуг не переглядається з урахуванням нових даних, або адаптивною поведінкою, якщо надання послуг є функцією, що залежить від контексту або попередньої послуги [14];

– колективний інтелект: здатність вчитися, розуміти та адаптуватися до середовища, використовуючи знання групи користувачів. ШІ можуть використовувати потенціал колективного інтелекту для вдосконалення алгоритмів машинного навчання і, отже, підвищення якості обслуговування. Наприклад, аналіз висловлювань багатьох користувачів на природній мові може призвести до більш крутої кривої навчання для алгоритмів розпізнавання

мови, оскільки адаптивність базується на великому та неоднорідному наборі даних. Отже, окремі користувачі ІПА можуть отримати вигоду від залучення групи [15].

1.5 Типи ІПА

Існує два типи автоматизованих персональних помічників: інтелектуальні автоматизовані помічники та розумні персональні агенти. Інтелектуальні автоматизовані помічники, наприклад, Apple Siri та Tronton's Cluzee, можуть виконувати завдання, що зазвичай виконують люди, що працюють персональними помічниками (наприклад, оформлення бронювання вечері, придбання квитків на події, організація подорожей) або надання інформації на основі голосового, письмового введення або команд. Розумні особисті агенти автоматично виконують завдання з управління або обробки даних на основі інформації та подій в Інтернеті, часто без ініціації користувача або взаємодії з користувачем. У міру того, як автоматизовані особисті помічники стають більш популярними, виникають зростаючі юридичні ризики.

Приклади завдань, які може виконувати розумний персональний агент [4], включають управління розкладом (наприклад, надсилання попередження на дату вечері, надсилання повідомлення про те, що користувач запізнюється через умови дорожнього руху, оновлює графіки для обох сторін та змінює час бронювання ресторану) та особисте управління здоров'ям (наприклад, контроль споживання калорій, частоти серцевих скорочень та режиму фізичних вправ, а потім дає рекомендації щодо здорового вибору).

Обидва типи автоматизованої технології персонального асистента забезпечуються поєднанням мобільних обчислювальних пристроїв та інтерфейсів прикладного програмування (API). Інтелектуальні автоматизовані

помічники призначені для виконання конкретних, одноразових завдань, визначених голосовими або письмовими інструкціями користувача, тоді як розумні персональні агенти виконують поточні завдання (наприклад, управління розкладом) автономно.

Також ІПА можна розділити на адаптивні голосові або зорові помічники, чат-боти, втілені віртуальні помічники, пасивні помічники та розмовні помічники.

Адаптивні голосові або зорові помічники. Дана група містить ІПА, які допомагають користувачам головним чином за допомогою мови, або за допомогою оптичних датчиків та візуального виводу на екран. Хоча більшість об'єктів цієї групи поєднують управління мовленням із візуальною взаємодією, наприклад управління жестами над вбудованими камерами або додаткову інформацію на екрані, мова в даний час залишається переважним режимом взаємодії. Ці системи здатні як розуміти, так і реагувати природною мовою та виконувати різні послуги на запити користувачів. Переважна більшість ІПА даного типу отримує моделі знань загального призначення, такі як керування розумними побутовими гаджетами, отримання пошти або додавання записів до календаря. Ці моделі знань є адаптивними, оскільки вони розвиваються протягом довших періодів використання і, таким чином, забезпечують вищу якість обслуговування при регулярному та тривалому використанні. Адаптивні помічники зазвичай використовують колективний інтелект. Дані про мовлення та використання широкого кола користувачів реєструються та зберігаються у великих центрах обробки даних (або виділених хмарних середовищах) та обробляються для поліпшення якості обслуговування ІПА.

За рахунок колективного інтелекту окремі користувачі отримують певні переваги від досвіду та взаємодії інших членів групи користувачів. Крім того, оскільки мова є переважним режимом взаємодії, більшість ІПА даного типу реагують за допомогою комп'ютерного голосу(як правило, подібного до

людини). Яскравими прикладами цього класу SPA є пристрої Amazon з операційною системою Alexa, Siri від Apple, Google Assistant, Cortana від Microsoft та Vixby від Samsung.

Асистенти – чат-боти. Другий тип містить ІПА, які в основному покладаються на взаємодію в текстовому чаті для надання послуг допомоги. Ця категорія включає в себе чат-боти, розмовні агенти на основі тексту, які здатні реагувати на введенні користувачем дані на основі семантичного аналізу тексту. Такі системи дедалі частіше застосовуються у службах підтримки першого рівня, оскільки вони здатні відповідати на поширені запитання та направляти користувачів через процеси підтримки. Для спрощення роботи з ними більшість помічників чат-ботів здатні інтерпретувати природну мову та відповідати користувачу. Оскільки обмін здійснюється на основі тексту, взаємодія здійснюється на екранах, які, однак, можуть також відображати додаткову інформацію, таку як зображення чи відео, відповідно до запиту користувача. Асистенти чат-боти зазвичай охоплюють досить специфічні доменні знання і використовуються для подання інформації, а не для виконання послуг. Репрезентативним прикладом для цього типу ІПА є MentorChat – настроюваний текстовий агент для спільного навчання [16].

Втілені віртуальні помічники. Системи створені, щоб представити віртуального персонажа (або аватара) з мовою, імітацією та жестами на природній мові, щоб забезпечити звичну взаємодію. Це досягається як мовленням, так і зоровим ефектом.

Часто ці асистенти призначені для спеціальних цілей, таких як електронне навчання, яке є найбільшим доменом для ІПА даного типу. У порівняно рідкісних випадках користувачі мають який-небудь контроль над поведінкою системи, ІПА даного типу переважно приймають ручне введення вхідних даних або прості команди (наприклад, для регулювання рівня серйозності в електронному навчанні). Однак приблизно третина цих систем

може самостійно пристосовуватися до уподобань або поведінки користувача. Метою ІІА третього типу є покращення взаємодії з користувачем шляхом плавного перенесення дій від людини до людини, таких як навчання, у віртуальний світ, зберігаючи при цьому переваги людської взаємодії, такі як емпатія, гумор та контекст для учнів. Типовими прикладами є AutoTutor та Віктор, віртуальний репетитор [17].

Пасивні помічники. Хоча всі попередні типи ІІА зосереджені на двонаправленому та явному обміні, ІІА четвертого типу розроблені, щоб бути максимально непомітними. Це означає, що користувачі мало взаємодіють із самою системою, коли вона збирає дані із, як правило, декількох різних джерел та рекомендує відповідні дії. Іншими словами, для надання відповідної інформації та порад ІІА не вимагає введення даних користувачем вручну. Отже, явна взаємодія зазвичай ініціюється системою, яка пасивно спостерігає за задачами та контекстом користувача. Таким чином, допомога в основному надається за допомогою екранного виводу. Окрім того, ІІА можуть самостійно виконувати дії як реакцію на сприйняті події через тригер. Більшість ІІА даного типу пропонують допомогу в спеціалізованих цілях, таких як приготування їжі чи огляд визначних пам'яток. Моделі знань, що лежать в основі, рідко адаптуються до спостережуваного контексту або шаблонів завдань, і жодна з досліджуваних систем не використовує генеровані натовпом дані для покращення якості послуг. Одним з репрезентативних прикладів є MimiCook, щоденний помічник у приготуванні їжі, який інтегрований у фізичне середовище кухні.

Розмовні помічники. Цей тип ІІА можна вважати помічником для наступного покоління службових зустрічей. Зосереджуючись на мовленнєвій взаємодії, ІІА п'ятого типу мають на меті збільшити схожість із взаємодією природної мови від людини до людини. Вони охоплюють більш складні можливості розпізнавання мови та розуміння мови, ніж будь-який інший клас.

Основна мета проекту – імітувати людську природну мову, щоб забезпечити найбільш природний та звичний досвід взаємодії. Це вимагає, щоб основна лінгвістична модель не тільки правильно реагувала на людські висловлювання, а й працювала з такими наповнювачами, як «ах», «гм» або паузи. Як і ППА другого типу, природні помічники для розмов можуть бути використані як агенти підтримки першого рівня, наприклад, як одна точка контакту в кол-центрі. Одним з яскравих прикладів для цього класу є розмовний агент Google Duplex.

1.6 Аналіз управління витратами часу

Управління часом визначається як процес планування та організації вашого часу з метою найбільш продуктивного його використання. Управління часом може допомогти отримати ряд навичок, інструментів і методів, що можна використовувати при виконанні конкретних завдань, проектів і цілей. Цей набір включає в себе широкий спектр діяльності, а саме: планування, розподіл, постановку цілей, делегування, аналіз тимчасових витрат, моніторинг, організацію, складання списків і розстановку пріоритетів.

Потреба в тайм-менеджменті в даний час обумовлена такими факторами, як швидкі темпи змін зовнішнього середовища, нестабільність світової економіки і кон'юнктури ринку, постійна потреба в змінах і інноваціях, які вже стають нормою, а не рідкісним винятком, зростання питомої ваги нематеріальних активів у вартості організації та інші складові, безпосередньо пов'язані зі сферою функціонування організації [37]. Всі вони призводять до зниження ефективності роботи.

Тайм-менеджмент – це система, що характеризується наступними ознаками [37]:

- спрямована на підвищення продуктивності праці;

- заснована на застосуванні існуючих і вироблення нових унікальних методів та інструментів гармонізації тимчасових процесів з досягненням їх логічною організацією, яка відповідає всім заданим умовам і обмеженням;

- передбачає вибудовування ієрархії цілей і завдань окремих співробітників і організації, реалізація яких пов'язана з тимчасовим розгортанням виробничих операцій;

- здійснюється у вигляді планування, а також безпосереднього виконання та коригування складених планів.

В управлінні часом можна виділити наступні процеси:

- постановка мети: визначення та формулювання однієї мети або декількох;

- планування і розстановка пріоритетів: розробка плану досягнення поставлених цілей і виділення пріоритетних та найважливіших завдань для виконання;

- реалізація: конкретні кроки і дії у відповідності зі створеним планом і порядком досягнення мети.

Для зменшення витрат часу можна використовувати різноманітні техніки тайм-менеджменту та максимально автоматизувати різноманітні процеси. Завдяки таким функціям, як розпізнавання обличчя та мовлення, використання чат-ботів на основі штучного інтелекту та інтелектуальне управління бізнес-процесами, сучасні рішення для управління часом можуть бути автоматизованими та ненав'язливими.

Технологія розпізнавання мовлення – це програма, яка використовує підключений до пристрою мікрофон, щоб перетворити мовлення на текст, який найчастіше використовується разом із програмним забезпеченням для обробки текстів. Сьогодні дана технологія дозволяє програмному забезпеченню не

лише перетворити свою мову в текст, але й фактично керувати пристроєм за допомогою голосових команд, усуваючи при цьому необхідність використання миші або інших пристроїв вводу. За дослідженнями людина набирає слова втричі повільніше ніж говорить. Це говорить про те, що існує потенціал для підвищення ефективності роботи для будь-якої професії чи хобі, що вимагає великої кількості обробки текстів.

Розпізнавання мовлення – це ефективний інструмент управління витратами часу, що має другорядну перевагу, а саме: при використанні програмного забезпечення не потрібно знаходитися безпосередньо біля комп'ютера. Це є вагомою перевагою для людей з обмеженими можливостями або для людей з травмами, оскільки вони можуть відпочивати руками, зап'ястями, спиною та шиєю, бо немає необхідності сидіти в одному положенні для використання клавіатури та миші.

Складання розкладу робочого дня вручну є процесом, що займає багато часу. Крім того, необхідно шукати винятки, помилки та невідповідності. Окрім додавання адміністративних накладних витрат, реальна небезпека полягає у помилках та затримках, які неминуче виникають внаслідок ручного процесу. Інтелектуальні робочі процеси спрощують бізнес-процеси, перевіряючи дані в режимі реального часу. Автоматичне складання розкладу діє як особистий асистент, автоматично плануючи завдання навколо зустрічей, нарад та інших подій. В результаті користувач просто можете зосередитись на тому, щоб виконувати свою роботу.

Таким чином, одними з найефективніших інструментів для зменшення витрат часу є складання розкладу автоматично і використання голосу та його розпізнавання.

1.6.1 Аналіз існуючих технік управління витратам часу

На сьогоднішній день існує безліч технік управління витратам часу. Їх можна розділити на категорії за визначенням пріоритетів, плануванням і організацією процесу роботи і відпочинку.

Найпопулярніші техніки представлені нижче [37].

Матриця Ейзенхауера (Принцип Ейзенхауера, Метод Ейзенхауера) – це техніка тайм-менеджменту для розстановки пріоритетів. Її використання дозволяє виділити важливі і суттєві задачі і вирішити, що робити з ними та рештою. В основі матриці Ейзенхауера лежить принцип Парето, згідно якого 20% витрачених зусиль дають 80% результату. Ейзенхауер удосконалив цей принцип.

Для використання даної техніки необхідно створити матрицю, що виглядає як таблиця з чотирьох клітинок. Кожна клітинка повинна мати певні назви. По горизонталі: «термінові», «не термінові», по вертикалі: «важливі», «не важливі». Таким чином в результаті вийде чотири клітинки, в які можна вписувати задачі:

- «важливі і термінові»: наприклад, критичні ситуації, проекти з швидким дедлайном, задачі такого типу необхідно виконувати без зволікання;
- «важливі і нетермінові»: наприклад, стратегічне планування в бізнесі, особистісний розвиток, для завдань даного типу необхідно встановлювати тимчасові рамки;
- «неважливі і термінові»: наприклад, спонтанні прохання або доручення, що не належать до безпосередніх обов'язки, тощо, по можливості від таких прохань слід відмовлятися або передоручати іншим;

– «неважливі і нетермінові»: це дрібниці, що віднімають час (перекури, перегляд серіалів, та інші поглиначі часу), необхідно прагнути усунути такі справи.

Мета використання створеної матриці – це пріоретизувати задачі по даним чотирьом категоріям і зосередити максимум уваги на виконанні «нетермінові і важливих». Тобто, необхідно намагатися, щоб завдання з цієї клітинки ніколи не переходили в категорію термінових.

Метод «Бридких жаб» – цей спосіб використовується в тому випадку, якщо в своїй діяльності людині доводиться стикатися з завданнями не надто складними і трудомісткими, але для неї неприємними, що не викликають позитивних емоцій, схожими на «бридких жаб».

Суть техніки полягає в тому, щоб розділити завдання за складністю і в першу чергу виконувати найскладніші, щоб не відкладати їх і тим самим не затягувати роботу. Тобто, необхідно скласти список задач, а потім кожній задачі розставити пріоритет від А до Е (А, В, С, D, Е) в порядку зменшення важливості завдання. Тобто А – це найскладніші і важливі завдання, які обов'язково необхідно виконати, а Е – малозначущі справи, які легко можна не робити. За технікою, необхідно одразу братися за задачі, позначені А. До В можна переходити тільки завершивши всі задачі А. До задач, позначених С, можна приступати тільки завершивши всі задачі В, тощо.

Крім очікуваного підвищення ефективності, перевага методу «Бридких жаб» ще й у тому, що такий поетапний перехід від необхідних справ до бажаних буде приносити все більше психологічного задоволення і емоційної віддачі протягом дня.

Метод «Поїдання слона» заснований на тому, що «слоном» в тайм-менеджменті називають великі задачі з великим обсягом робіт, наприклад, написання дисертації, будівництво будинку, вивчення іноземної мови та інші глобальні проекти. Такі справи є важливими, але кожен крок не несе великого

результату, тому втрачається мотивація. Чим більше завдання і чим менш жорсткий термін виконання, тим важче змусити її виконати. У тайм-менеджменті подібні великі справи називають слонами, яких необхідно з'їдати по частинах (нарізати слона на біфштекси). Перекладаючи з мови тайм-менеджменту, великі справи потрібно розбити на безліч дрібних завдань, і вирішувати ці маленькі завдання щодня, приділяючи їм короткий проміжок часу. І ось такими невеликими кроками можна просунути поставленої мети.

Результатом використання даного методу є зменшення нервового напруження та підвищення ефективності роботи. Адже, коли розділити задачу на підзадачі, вона не здається такою масштабною.

Характерно те, що люди, як правило, самі будують «слонів» з своїх справ, виконання яких довгий час відкладають на потім. Кожен раз, коли людина думає про яку-небудь задачу, але не приступає до її виконання, «слон» зростає. Часто, коли людина все-таки починає справу і виконує її, з'ясовується, що робота не така вже складна і неприємна. Але дуже великий потенціал був розтрачений на тривожні думки і негативні емоції. Виконані справи ж, навпаки, додають в життя енергії і вносять позитив [37].

Ще одним популярним методом тайм-менеджменту є техніка Помодоро. Назва техніки пішла від того, що її засновник – Франческо Чирилло, для визначення часу використовував звичайний кухонний таймер у вигляді помідора. Його ідея була заснована на тому, що для багатьох час є ворогом. А він прагнув вигідно співпрацювати з часом, а не боротися з ним.

Створена техніка – це революційна система управління часом, дуже проста в освоєнні і в той же момент ефективна в організації роботи і відпочинку.

Техніка Помодоро дозволяє користувачу бути на максимумі своєї ефективності, прі цьому не перевантажувати себе. Також вона дозволяє чітко сконцентруватися над певним завданням та дозволяє розв'язати поставлену

проблему в більш короткий проміжок часу ніж, якщо б вирішувалося кілька завдань одночасно. Більше того, техніка Помодоро надає можливості грамотного планування, встановлення пріоритетності завдань, ведення хронометражу, тощо [38].

В оригінальній техніці Помодоро всього існує шість кроків обов'язкових до виконання:

- необхідно сформулювати завдання, яке треба виконати;
- виставити помодоро-таймер (зазвичай на 25 хвилин) ;
- працювати над завданням, ні на що не відволікаючись;
- після дзвінка таймера відзначити помодоро як виконане;
- зробити коротку перерву (зазвичай 3-5 хвилин) і повернутися до другого кроку;
- після кожного 4-го помодоро необхідно зробити довгу перерву (15-30 хвилин).

В рамках даної техніки, помодоро – це інтервал часу, який витрачається на виконання завдання. Помодоро відокремлюються один від одного короткими перервами (3-5 хвилин) [38]. Чотири помодоро утворюють підхід. Між підходами є тривала перерва (15-30 хвилин).

Для даної техніки основоположними є наступні етапи: планування, відстеження, запис, обробка та візуалізація. На етапі планування завданням присвоюються пріоритети, що відображається в списку «To Do». Цей список містить всі створені завдання, які необхідно виконати. Етап планування дозволяє оцінити зусилля, які необхідно застосувати до тієї чи іншої задачі. Коли помодоро завершено, користувач це записує, що збільшує почуття виконаного обов'язку і надає необроблені дані для подальшого самопостереження і поліпшення.

Метою даної техніки є зниження впливу внутрішніх та зовнішніх перешкод на фокус на завданні. Помодоро – неподільне. Якщо помодоро переривається, то воно відмічається як не виконане [38].

Задля ефективного використання техніки необхідно дотримуватися наступних правил:

- після кожних чотирьох помодоро треба робити 15-20-хвилинну перерву;
- протягом 25 хвилин роботи не можна відволікатися і перериватися, інакше доведеться ставити таймер та розпочинати помодоро заново;
- завдання довше семи помодоро треба розділяти на кілька підзадачі;
- якщо завдання вимагає менше одного помодоро, то його слід згрупувати з іншим.

Використовуючи техніку Помодоро кілька днів, користувач зможе оцінити, скільки помодоро у нього є на кожен день. Наприклад, стандартний восьмигодинний робочий день складається з чотирнадцяти помодоро. Складаючи список завдань на день, користувач зможе заздалегідь оцінити, на які завдання треба виділити більше часу, на які – менше, а які варто перенести на завтра.

Тобто, використовуючи техніку можна досягти наступних результатів:

- збільшити продуктивність праці;
- мінімізувати кількість безглузвих відволікань;
- навчитися укладатися в певні часові відрізки при виконанні роботи;
- досягти стійкою самодисципліни;
- розвинути швидкість і гнучкість мислення в умовах стресу.

Згодом можна розділяти робочі відрізки на кілька частин, залишаючи 3-5 хвилин на початку відрізка для вивчення зробленої до цього роботи і 3-5 хвилин в кінці для перегляду того, що ви робили зараз. Цей аналіз не вимагає змін в тривалості помодоро.

Як і будь-яку іншу техніку, Помодоро допускається адаптувати під власні потреби. Час роботи і відпочинку можна зменшувати або збільшувати, орієнтуючись на особисті переваги і, звичайно ж, на рід діяльності. Якщо потрібно непередбачено перерватися від роботи, то раціональним і ефективним буде зафіксувати подібну несподіванку: її рівень важливості по 10 бальною шкалою, витрачений час і привід.

Рано чи пізно настане такий момент, коли завдання буде повністю виконане, а на таймері залишиться ще чимало часу. В такому випадку можна зайнятися плануванням майбутніх завдань, підбиттям поточних підсумків, аналізом виконаних задач, читанням якісної літератури або виконанням позапланових завдань, що не входять в поточний список справ.

Таким чином, можна сформулювати основні переваги методу Помодоро:

- грамотне планування: користувач сам обирає найбільш пріоритетні завдання, користувач зможе прогнозувати, скільки часу займуть задачі;
- користувач тримає під контролем поточний стан справ, він відстежує, скільки часу витрачається на його задачі;
- підвищується увага, менше витрачається часу на відволікання;
- техніку можна використовувати як своєрідний хронометраж – в кінці дня у користувача є інформація для аналізу, це повноцінний зворотний зв'язок;
- до кінця дня у користувача є візуалізована інформація, яка допомагає поліпшити ефективність роботи у наступні дні.

Проте, за даною технікою не всім може бути зручно працювати. Жорсткі часові рамки здатні викликати певний дискомфорт у людей, що займаються креативними професіями. Сюди можна віднести і музикантів, і письменників, і художників. Техніка Помодоро буде агресивно впливати на їх натхнення і руйнувати атмосферу творчості [38].

Зазначені техніки тайм-менеджменту є найпопулярнішими та широко використовуваними. Аналізуючи дані техніки, можна зробити висновок, що методи «Бридик жаб» та «Поїдання слона» дозволяють ефективно виконувати складні та неприємні задачі та не концентруються на інших.

А техніки «Матриця Ейзенхауера» та Помодоро концентруються на всіх задачах та можуть бути автоматизовані. Вони допомагають концентруватися на роботі впродовж робочого дня. «Матриця Ейзенхауера» є ефективною для виконання великої кількості проектів для ефективного використання часу та не накопичувати важливі та термінові задачі. Метод Помодоро дозволяє ефективніше працювати за рахунок чітко обмежених переривів.

1.7 Постановка задачі

Пропонується створити інтелектуальний персональний помічник на основі методів машинного навчання, який зможе допомагати підвищувати рівень продуктивності та ефективності працівника, що займається знаннями, допомагаючи йому в організації та виконанні завдань. З функціональної точки зору інтелектуальний персональний помічник повинен зосереджуватися сфері управління завданнями. Управління завданнями передбачає планування, виконання та контроль за виконанням завдань. Такі завдання можуть бути особистими, оскільки вони походять від користувача, або можуть виникати внаслідок відповідальності, пов'язаної з проектом.

Передбачається, що даний помічник зможе підвищити економічну ефективність організації та ефективність, навантаження для багатьох працівників та зменшити рівень когнітивних перевантажень на робочому місці.

Пропонується використовувати готовий проект CALO, де центральним компонентом є PExA, що вирішує схожу поставлену задачу, але має ряд недоліків. Таким чином, пропонується створити програму, що буде використовувати готові функції CALO, буде сфокусована на сфері тайм-менеджменту та усуває основний недолік PExA, при цьому є не менш ефективною.

2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ІПА, ЩО ВИКОРИСТОВУЮТЬ ТЕХНІКИ ТАЙМ-МЕНЕДЖМЕНТУ

2.1 Дослідження існуючих рішень

Коли когнітивні навички людини є невід'ємною частиною вирішення проблеми, більш доцільним є створення помічника, який би полегшував людині рутинні завдання, щоб вона могла зосередитися на прийнятті стратегічних рішень. Змішані технології планування, такі як MARGEN [20], PASSAT [21] та LinTo, надають приклади такого типу помічників таким чином, що вони підтримують людину у прийнятті рішень щодо стратегічного планування, допомагаючи в управлінні заданими обмеженнями та вимогами.

2.1.1 Дослідження PASSAT

PASSAT – Plan-Authoring System based on Sketches, Advice, and Templates – це система складання планів на основі ескізів, порад та шаблонів, що поєднує інтерактивні інструменти для побудови планів із набором автоматизованих та змішаних можливостей, призначених для доповнення навичок планування людини.

У своїй основі PASSAT – це система складання планів, в якій користувач має можливість інтерактивно створювати та модифікувати плани. Користувач може спиратися на бібліотеку шаблонів, яка створена для допомоги у розробці плану. Шаблони відповідають формі мережі ієрархічних завдань і можуть мати за собою як параметризовані стандартні операційні процедури, так і випадки, що відповідають фактичним або умовним планам, розробленим для відповідних завдань.

PASSAT включає ряд можливостей автоматизованого та змішаного планування. Користувачі можуть застосувати автоматизований режим планування, щоб змінити будь-яке відкрите завдання в рамках плану. Ескіз плану дозволяє користувачам створювати попередні плани, які потім заповнюються за допомогою шаблонів, розроблених для подібних завдань. Поради в рамках PASSAT дозволяють користувачам визначати політику високого рівня, яка відповідає як планам, так і процесам планування.

PASSAT також включає механізми сприяння процесам, призначені допомогти користувачеві в управлінні розробкою плану. Ці механізми допомагають користувачеві відстежувати відкриті завдання та вимоги до інформації для поточного плану. Така допомога є критично важливою у складних програмах, оскільки допомагає користувачеві залишатися зосередженим, не пропускаючи важливих деталей.

Завдяки поєднанню інтерактивних та автоматизованих можливостей, PASSAT дозволяє користувачеві швидко розробляти плани, які спираються на минулий досвід, закодований у шаблонах, але з урахуванням його індивідуальних уподобань та вимог поточної ситуації.

Розробка в рамках PASSAT керувалась двома ключовими принципами:

– гнучке «нестандартне» планування: традиційні системи планування обмежують користувачів у певному наборі рішень, а саме тих, що передбачаються заздалегідь визначеними моделями дій, що лежать в основі розробки плану. У рамках PASSAT шаблони розглядаються як керівні принципи для виконання завдань; користувач може вільно розширити набір рішень, визначених шаблонами. Зокрема, користувач може замінити обмеження, пропустити завдання або вставити додаткові завдання відповідно до своїх особистих уподобань або вимог поточної ситуації. Така гнучкість є критично важливою для доменів, у яких неможливо забезпечити правильну та вичерпну колекцію шаблонів;

– керована орієнтована на користувача автоматизація: автоматизовані можливості PASSAT створені як для доповнення навичок планування людини, так і для того, щоб люди могли легко керувати ними. Автоматизація може використовуватися під контролем користувача лише в тих випадках, коли він вважає, що це буде корисно.

PASSAT – це загальна, незалежна від домену технологія, але розроблена для додатків із наступними характеристиками:

– складність домену виключає повне охоплення всіх відповідних знань з планування. Однак можуть бути розроблені моделі часткового планування;

– людський внесок є критично важливим, але певна кількість засобів автоматизації як покращить якість плану, так і зменшить загальний час планування.

Оскільки багато типів операцій можна широко охарактеризувати заздалегідь визначеними шаблонами, можна створити бази знань, які охоплюють певні частини процесу планування. Однак окремі операції, як правило, є надзвичайно відмінними, що робить важливим наявність інструментів, що дозволяють користувачам змінювати та налаштовувати плани відповідно до потреб конкретної ситуації.

Багато потенційних прикладних областей для технологій планування поділяють ці характеристики, оскільки вони мають частково формалізовані знання про домен і потребують значного вкладу користувача для створення високоякісних планів, специфічних для конкретної ситуації. З військової сторони приклади включають повітряні операції, планування ліквідації наслідків стихійних лих та операції евакуації, що не ведуть бойові дії. Космічні програми включають планування наукових місій та планування наземних операцій.

2.1.2 Дослідження MAPGEN

Система MAPGEN – Mixed-Initiative Activity Plan Generator – представляє собою систему з використанням технологій планування. Місія Mars Exploration Rovers (MER) Mars – була однією з найамбітніших наукових місій NASA на той час. Два марсоходи здійснили м'яку посадку взимку 2004 року, і кожен марсохід мав багатий набір приладів для проведення віддалених спостережень на місці для з'ясування клімату, активності води та середовища проживання на планеті. Серед наукових цілей місії MER було:

- визначення водної, кліматичної та геологічної історії ділянок поверхні на Марсі, де умови могли бути сприятливими для збереження доказів добіотичних або біотичних процесів;
- виявлення гідрологічних, гідротермальні та інших процесів, що діяли на місці висадки;
- виявлення та дослідження марсіанських порід та ґрунтів, які мали найбільші можливі шанси зберегти дані про стародавні умови навколишнього середовища та можливу добіотичну чи біотичну активність;
- реагування на інші відкриття, пов'язані з розвідкою на роверах.

Планування наукової діяльності з використанням набору приладів на борту марсоходів забезпечуючи при цьому безпеку марсохода та дотримуючись обмежень ресурсів марсохода (наприклад, потужності) було вирішальним.

Щоб вирішити цю критичність, проект MER обрав MAPGEN як інструмент планування діяльності. MAPGEN має наступні можливості:

- автоматично формує плани та графіки наукової та інженерної діяльності;

- може використовуватися для перевірки гіпотез (із використанням аналізу «що-якщо» за різними сценаріями);
- виконує редагування плану;
- виконує обчислення та аналіз ресурсів;
- виконує обмеження та технічне обслуговування.

MARGEN поєднує в собі дві існуючі системи: APGEN – інструмент планування діяльності від Лабораторії реактивного руху та система планування від Дослідницького центру NASA Ames (надалі – Планувальник). APGEN використовувався як багатоцільовий інструмент для ряду польотних проектів, тоді як Планувальник літав на борту Deep Space 1 NASA як частина першої бортової системи штучного інтелекту. MARGEN використовує сильні сторони обох цих інструментів, щоб надати користувачеві комплексний інструмент для планування наукової діяльності [21].

MARGEN – це інструмент планування наукової діяльності. Основними користувачами цього інструменту були тактичні планувальники місії MER та науковці, що маніпулювали науковими цілями з розрахунком на конкретні інженерні обмеження.

MARGEN надає ключові можливості, що перевищують можливості APGEN. Сюди входить складання плану, активне дотримання правил польотів та місій та вирішення конфліктів, таких як ті, що пов'язані із перекриттям забороненої діяльності або порушеннями використання ресурсів. Заборонені перекриття та інші обмеження вказані в моделі домену Планувальника, яка, в свою чергу, походить від Словника діяльності, який описує повний набір абстрактних дій, які, використовує користувач, а також правила польоту та місії, засновані на Правилах польоту з Словника діяльності. Основний механізм обмежень Планувальника забезпечує застосування правил у моделі домену.

На рисунку 2.2 представлений функціональний вигляд системи. Під час фази формування плану науковці складають список необхідних спостережень для марсіанського дня. Ці спостереження розширюються до відповідних видів діяльності на основі визначень у Словнику діяльності та надаються MARGEN разом з тимчасовими обмеженнями та обмеженнями підтримки (наприклад, «діяльність X може відбуватися лише при наявності діяльності Y» або «діяльність Z і діяльність W повинні відбуватися разом або не відбуватися»). Ці заходи, разом із запропонованою інженерною діяльністю (наприклад, необхідність документувати, за допомогою камер об'єкта гірських порід) та початковими умовами складають основу для початку етапу планування.

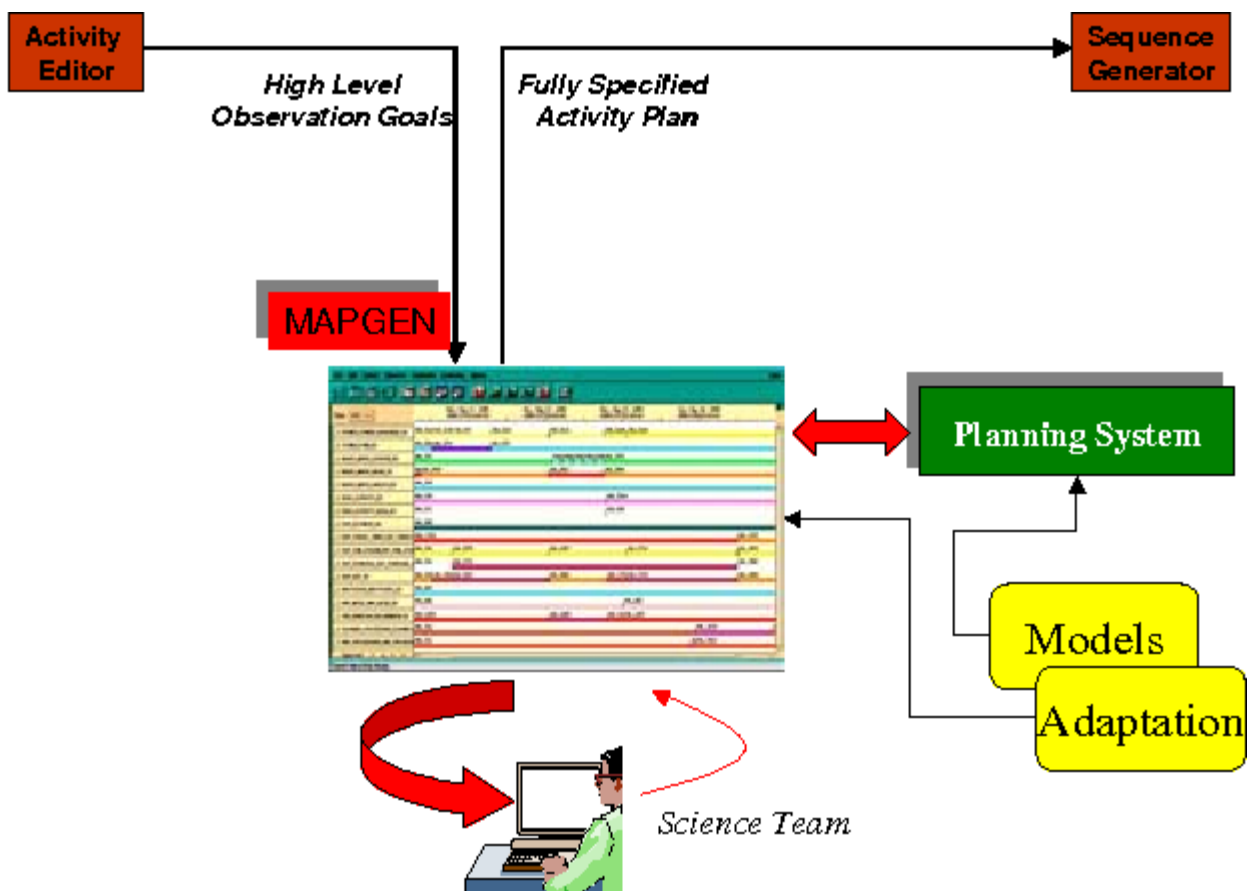


Рисунок 2.2 – Функціональний вигляд MARGEN

Планувальник використовує модель домену та генерує можливий план, який відображається в графічному інтерфейсі користувача APGEN як рішення, яке користувач може змінити. MAPGEN також забезпечує альтернативний спосіб використання можливостей, дозволяючи вибіркоче поступове планування цілей спостереження високого рівня. Якщо використовується цей метод, користувач сам визначає порядок вирішення цілей спостереження, вибираючи їх у графічному інтерфейсі. Користувач може в інтерактивному режимі експериментувати з альтернативними можливостями створення сценаріїв «якщо-то». Результат подається до наступного циклу ітерацій для створення плану, який користувач вважає відповідним. Після завершення цього ітераційного процесу план виводиться у файл.

Таким чином MAPGEN забезпечує наступні переваги:

- залишає більше часу для наукового аналізу;
- забезпечує кращу відповідність обмеженням, пов'язаних з плануванням діяльності, як наслідок, до створення більш надійних наукових планів і, отже, меншої кількості переглядів планів;
- зменшення навантаження на персонал місії.

На рисунку 2.3 схематично зображено систему. В системі APGEN і Планувальник з'єднані через чітко визначений інтерфейс, який синхронізує базу даних Планувальника з наборами дій, якими користувач маніпулює. Це дозволяє отримати доступ до функціональності Планувальника за допомогою звичайного використання графічного інтерфейсу APGEN. Як результат, досвідчений користувач APGEN, який використовує MAPGEN, бачить звичайний графічний інтерфейс і може взаємодіяти з ним звичними способами [22].

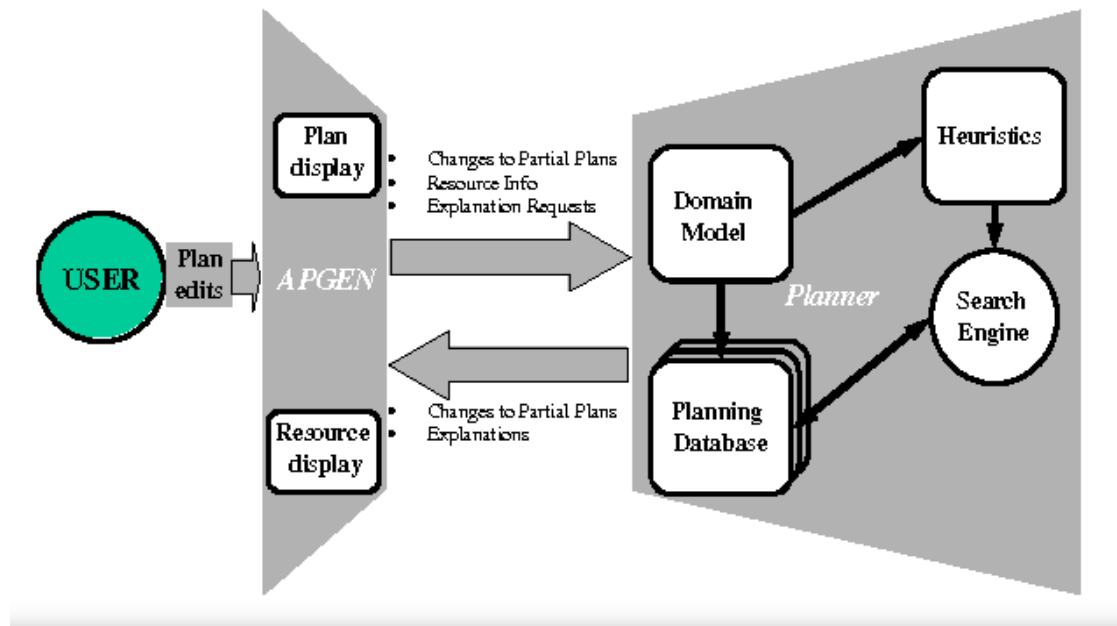


Рисунок 2.3 – Схематичне зображення MAPGEN

2.1.3 Дослідження LinTO

LinTO – це інтелектуальна голосова платформа та розумний помічник для покращення ефективності та продуктивності в бізнесі. Мета створення LinTO – це створення системи розуміння мови, яка підтримує високу ефективність як автоматичного розпізнавання мови, так і обробку природних мов, при цьому є портативною та масштабованою. LinTO має відкритий код з можливістю комерційного та некомерційного використання.

LinTO – це клієнт-серверна система, яка дозволяє розробляти, розгортати та підтримувати програмне та апаратне забезпечення клієнтів за допомогою голосового інтерфейсу користувача. Система підвищує простоту використання та продуктивність як в адміністративному управлінні, так і в контексті клієнтських додатків, пропонуючи процеси керування голосом «вільні руки»,

розпізнавання мови з кількома динаміками та голосовий доступ для програм клієнтів.

На платформі LinTO є користувацька консоль, адміністратор LinTO, яка використовується для проектування, побудови та управління певними голосовими помічниками. Кожен асистент складається з моделей та ресурсів, необхідних для виконання його бажаних функцій, і може включати будь-яку кількість навичок або пар намірів і дій, визначених для конкретного процесу, наприклад надання усного звіту про погоду. Вони представлені в адміністраторі LinTO як робочий процес, яким легко керувати.

На рисунку 2.4 зображено процес розуміння розмовної мови, який складається з механізмів автоматичного розпізнавання мови та обробки природної мови.

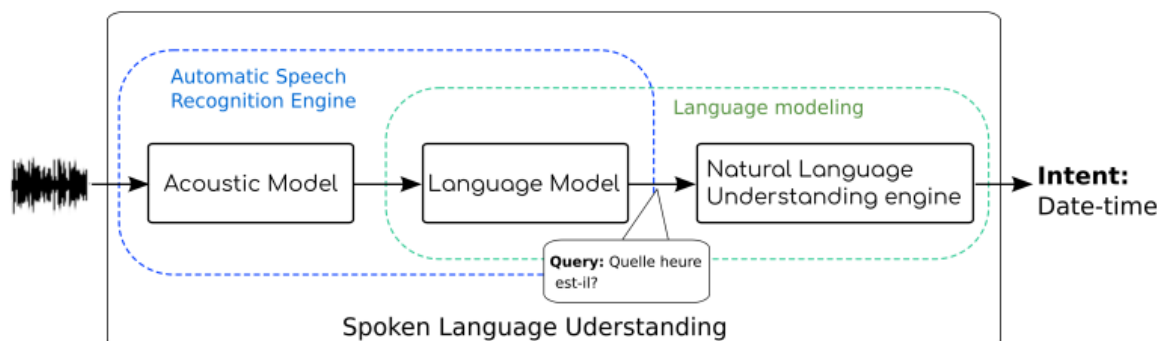


Рисунок 2.4 – Архітектура розуміння розмовної мови LinTO

Механізм розпізнавання мовлення розпізнає розмовну мову і транскрибує її в текст, спочатку зіставляючи звуковий потік у послідовність ймовірностей над акустичними одиницями за допомогою акустичної моделі, а потім перетворюючи виходи моделі в текст за допомогою мовленнєвої моделі. Механізм обробки мовлення інтерпретує мову. Після інтерпретації запиту

користувача завершальним кроком є обчислення відповіді за допомогою менеджера діалогів. Відповідь може приймати різні форми залежно від запиту, наприклад, звукова відповідь, якщо запитується про погоду, або дія на підключеному пристрої, якщо попросять відтворити пісню.

Відповідно до цілей щодо продуктивності, портативності та масштабованості, було розроблено систему розуміння розмовної мови із використанням хмарних технологій. Система відповідає трьом основним вимогам:

- може бути розгорнута в будь-якій системі;
- може обробляти велику кількість запитів;
- відповідає в режимі реального часу з високою точністю.

Для задоволення першої вимоги використовується технологія Docker, щоб створити «контейнерні» служби, які можна використовувати в будь-якому середовищі. Для вимоги до масштабованості використовується Docker Swarm, інструмент, який допомагає управляти кількома контейнерами, розгорнутими на декількох хост-машинах, таким чином, що послугу можна масштабувати вгору або вниз залежно від кількості запитів. На рисунку 2.5 зображено менеджер служби, що використовує Docker Swarm для організації контейнерів. Для того, щоб забезпечити точну реакцію в реальному часі, компоненти системи розроблені відповідно до оптимізації компромісу між точністю та ефективністю обчислень. Моделі навчаються зменшувати розмір навчальної вибірки та підвищувати точність внутрішнього домену, обмежуючи словниковий запас, а також різноманітність запитів, які вони повинні моделювати.

Платформа LinTO та всі її послуги та компоненти публікуються публічно, безкоштовно для комерційного та некомерційного використання. Мовою, яку зараз підтримує платформа, є французька [23].

На рисунку 2.5 схематично зображено менеджер служби, що використовує Docker Swarm для організації контейнерів

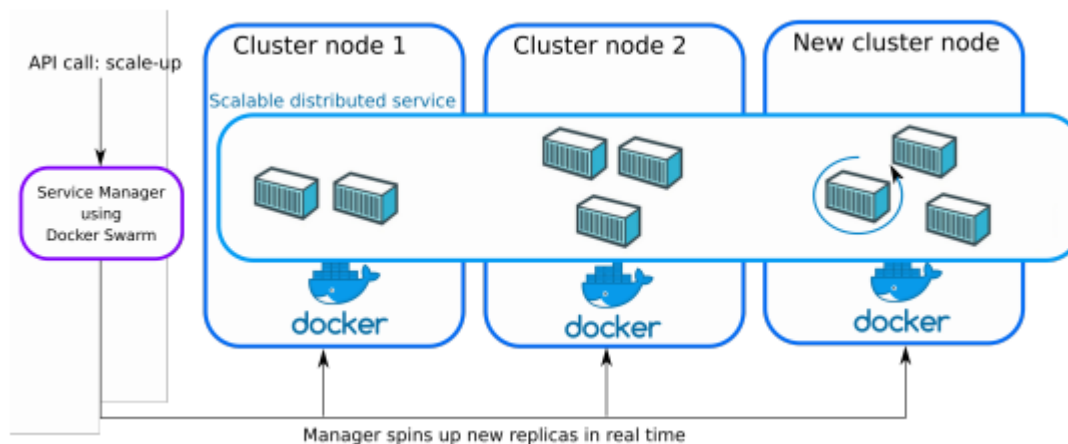


Рисунок 2.5 – Менеджер служби, що використовує Docker Swarm для організації контейнерів

2.2 Висновки дослідження існуючих ІПА

Проаналізувавши існуючі рішення можна виділити, що вони мають як загальні так відмінні друг від друга ознаки. Пошук ІПА здійснювався за наступними критеріями: щоб ІПА мали відкритий код, або можна було знайти детальний опис в научних виданнях та статтях, щоб вони підтримували користувача у прийнятті рішень щодо стратегічного планування, допомагаючи в управлінні заданими обмеженнями та вимогами.

Різні стилі допоміжних технологій підходять для різних застосувань, залежно від типів проблеми, яку потрібно вирішити, та балансу знань та знань між користувачем та системою. Деякі асистенти є частиною цілою екосистеми

пристроїв, що покращує ефективність використання та надає додатковий спектр можливостей, деякі є самостійними системами.

Розглянуті системи є досить популярними та непогано виконують поставлені задачі, але у існуючих помічників досить різні інтерфейси, підтримка функцій сильно варіюється, і споживачі не зовсім точно знають, чого очікувати від кожної платформи. Для того, щоб навчитися ефективно використовувати системи, потрібно багато досліджень на стороні споживача.

2.3 Критика ІПА

Хоча розумні помічники можуть зробити багато користі для користувачів, виникає логічне питання: що роблять користувачі, для того щоб користуватися цими послугами, адже багато помічників є безкоштовними. Користувач надає безкоштовні дані для навчання та вдосконалення віртуального помічника.

Критику щодо конфіденційності викликає той факт, що голосові команди доступні постачальникам віртуальних помічників у незашифрованому вигляді, і, таким чином, можуть передаватися третім особам і оброблятися в несанкціонованому або несподіваному порядку. На додаток до мовного змісту, манера вираження та характеристики голосу можуть неявно містити інформацію про його біометричну ідентичність, риси особистості, форму тіла, стан фізичного та психічного здоров'я, стать, настрій та емоції, соціально-економічний статус та географічне походження [24]. Більше того, випадкові включання помічників означають, що голосовий асистент може записувати приватну інформацію, яку не повинен був чути.

2.4 Огляд проекту CALO

Ідея комп'ютера, який використовує ШІ для обслуговування людей у їх повсякденному житті, відкрила шлях для розвитку «цифрового помічника». Один з проектів, який реалізовує цю ідею, це проект Cognitive Agent that Learns and Organizes (CALO), або Когнітивний помічник, який навчається та організовує.

SRI International очолив розробку програмного забезпечення, яке за планом повинно було зробити революцію в тому, як комп'ютери підтримують користувачів, що приймають рішення.

DARPA, за програмою Perceptive Assistant that Learns (PAL), призначило SRI International на виконання перших двох етапів п'ятирічного контракту на розробку стійкого персоналізованого когнітивного асистента. DARPA очікували, що програма PAL буде генерувати нові інноваційні ідеї, що призведуть до нової науки, нових та фундаментальних підходів до поточних проблем, нових алгоритмів та інструментів, а також дасть нові технології, що матимуть значну цінність для військових.

SRI International охрестила свій новий проект CALO – когнітивний помічник, який навчається та організовує. Назву надихнуло латинське слово «calonis», що означає «солдатський слуга». Мета проекту – створити когнітивні програмні системи, тобто системи, які можуть міркувати, вчитися на досвіді, говорити, що робити, пояснювати, що вони роблять, міркувати про свій досвід і реагувати на події.

Програма PAL тісно співпрацювала з військовими партнерами. PAL використовувався для забезпечення голосової взаємодії з користувачами для підтримки військового планування [25].

Програмне забезпечення CALO, яке навчається, взаємодіючи та отримуючи поради своїх користувачів, обробляє широкий спектр

взаємопов'язаних завдань прийняття рішень, які в минулому були стійкими до автоматизації. Він має можливість брати участь і виконувати рутинні завдання, а також допомагати, коли трапиться щось несподіване. Щоб зосередити дослідження на реальних проблемах та забезпечити відповідність програмного забезпечення таким вимогам, як конфіденційність, безпека та довіра, дослідники проекту CALO самі використовуватимуть цю технологію під час її розробки [26].

Проект CALO ліг в основу багатьох існуючих інтелектуальних асистентів. Найвідомішим нащадком CALO є телефонний цифровий асистент Siri, який зараз є частиною Apple iOS, але напочатку був розробкою SRI International. Також CALO використовується в Social Kinetics, соціальному додатку, який вивчає персоналізовані стратегії лікування хворих на хронічні захворювання, проекті Trapit, який є веб-скрепером та агрегатором новин, що робить розумний вибір веб-контенту на основі уподобань користувачів, Tempo AI, розумному календарі, Kuato Studios, стартапу з розробки ігор.

Розглянемо функції, що надає CALO. CALO допомагає своєму користувачеві в шести функціях високого рівня.

Впорядкування та встановлення пріоритетів інформації. Оскільки користувач працює з електронною поштою, зустрічами, веб-сторінками, файлами тощо, CALO використовує алгоритми машинного навчання, щоб побудувати модель запиту того, хто працює над якими проектами, яку роль вони відіграють, наскільки вони важливі, як пов'язані з цим конкретні документи та результати тощо.

Підготовка інформаційних артефактів. CALO може допомогти своєму користувачу скласти нові документи, такі як презентації PowerPoint, використовуючи вивчення структури та вмісту з попередніх документів, доступних раніше.

Посередництво у спілкуванні з людьми. CALO надає допомогу в спілкуванні як на електронних форумах, так і на фізичних зустрічах. Якщо надається доступ для участі у події, CALO автоматично формує стенограму зустрічі, відстежує призначення задач, виявляє ролі учасників тощо. CALO також може скласти попередній план для зустрічі, що містить матеріали, яку потрібно підготувати заздалегідь або мати під рукою під час зустрічі.

Управління завданнями. CALO може автоматизувати звичайні завдання, а також може навчати нових як необхідно виконувати нові завдання, спостерігаючи та взаємодіючи з користувачем.

Планування та міркування в часі. CALO може вивчити уподобання користувача щодо того, коли йому потрібно щось зробити, та допомагає керувати графіком.

Розподіл ресурсів. В рамках управлінні завданнями CALO може навчитися залучати нові ресурси, щоб допомогти у виконанні завдань.

На рисунку 2.6 схематично зображено структуру CALO.

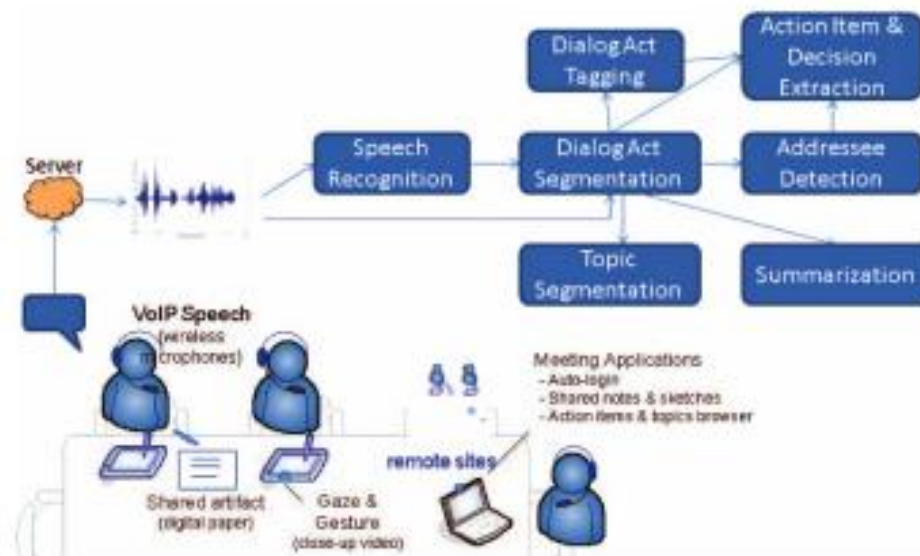


Рисунок 2.6 – Структура CALO

На сьогоднішній день CALO являє собою набір програм, що об'єднує три центри:

- PExA, the Project Execution Assistant – це помічник, що вирішує такі завдання, як планування і розподіл часу, делегування завдань і управління ресурсами, аналіз витрат часу, фіксація часу, призначення пріоритетів, складання списків;

- CALO-MA, the Meeting Assistant – це помічник призначений для розширення участі користувача у зустрічі за допомогою механізмів, що відстежують теми, які обговорюються, позиції учасників та прийняті рішення;

- IRIS, the Information Assistant – це прикладна програма, що дозволяє користувачам створювати «персональну карту» для інформаційних об'єктів, пов'язаних з офісом.

Система PExA є центральним компонентом та містить модулі управління часом і управління завданнями.

На рисунку 2.7 зображена архітектура PExA. У одній системі може бути тільки один користувач, але кожна система пов'язана з іншими користувачами для обміну інформацією та координації завдань. Модуль управління часом орієнтований на планування зустрічей, створення системи нагадувань і балансування робочого навантаження. Модуль управління завданнями здійснює планування, виконання та контроль завдань. Обидва модулі використовують загальну структуру агента SPARK, що реалізується на базі технології BDI, і загальної онтології. Додаткові компоненти в рамках системи PExA забезпечують можливість моніторингу виконання завдань прогнозування, навчання і координації команд.

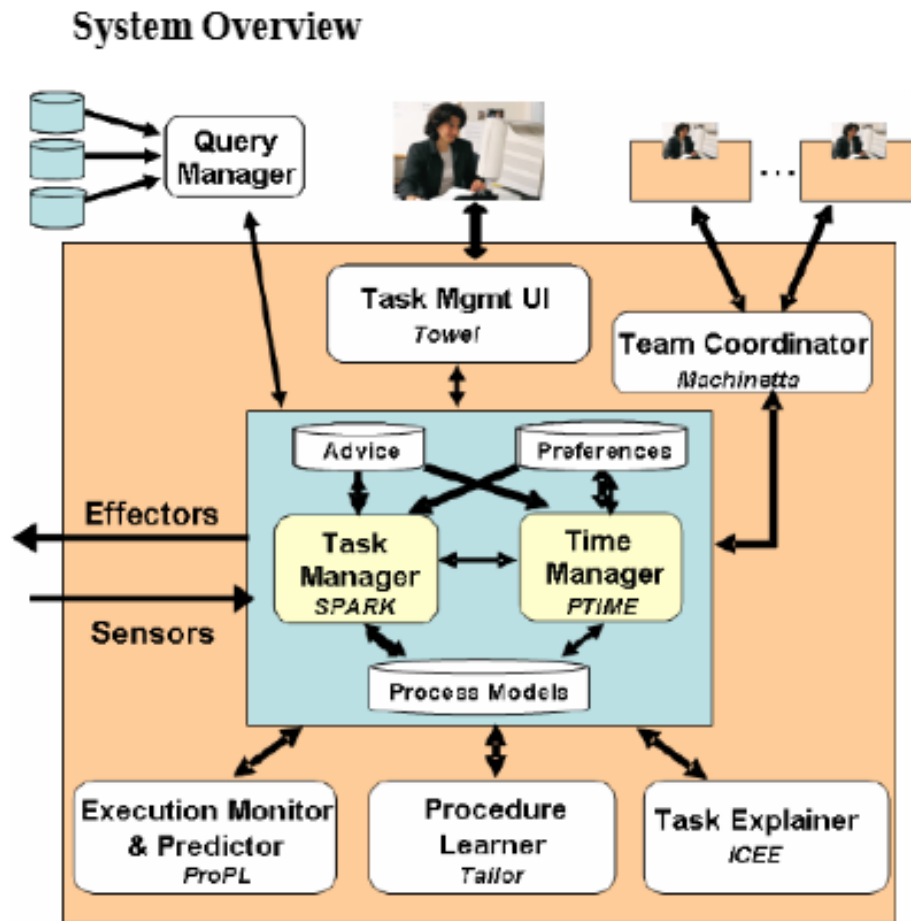


Рисунок 2.7 – Архітектура PEXA

Знання PEXA поширюються по всій системі. Ключові джерела знань включають особистий календар користувача, поштовий сервер, модель уподобань користувача та модель поточної ситуації. Система CALO забезпечує уніфікований інтерфейс Tower.

3 ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ДЛЯ УПРАВЛІННЯ ВИТРАТАМИ ЧАСУ

В ША використовуються різні технології, методи, моделі та засоби. Проведемо їх аналіз.

Віртуальний помічник – це технологія, заснована на штучному інтелекті. Програмне забезпечення використовує мікрофон пристрою для отримання голосових запитів, поки голосовий вихід здійснюється на динаміку. Найцікавіші процеси відбуваються між цими двома діями. В ША, що вирішують задачу управління витратами часу можна виділити такі основні модулі, як:

- перетворення мовлення в текст;
- обробка тексту;
- прогнозування часових витрат на виконання завдання;
- додавання задачі в календар [4].

Коли користувач просить персонального асистента виконати завдання, звуковий сигнал на природній мові перетворюється в цифрові дані, які можуть бути проаналізовані програмним забезпеченням. Потім ці дані порівнюються з базою даних програмного забезпечення за допомогою алгоритмів, щоб знайти відповідну відповідь. Ця база даних розташована на розподілених серверах у хмарних мережах. З цієї причини більшість персональних помічників не можуть працювати без надійного з'єднання з Інтернетом.

Зі збільшенням кількості запитів база даних програмного забезпечення розширюється та оптимізується, що покращує розпізнавання голосу та збільшує час відгуку системи.

3.1 Задача розпізнавання мовлення

Розпізнавання мовлення – автоматичний процес перетворення мовного сигналу в цифрову інформацію (наприклад, текстові дані). Зворотною завданням є синтез мови. Розпізнавання мовлення включає знання та дослідження в галузі комп'ютерних наук, лінгвістики та обчислювальної техніки.

Деякі системи розпізнавання мовлення вимагають окремого навчання, коли мовець читає текст або ізольовані слова. Система аналізує конкретний голос людини та використовує його для точної настройки розпізнавання мови цієї людини, що призводить до підвищення точності. Системи, що не використовують навчання з користувачем, називаються «незалежними від мовця» системами. Системи, що використовують навчання, називаються «залежними від мовця».

З технологічної точки зору, розпізнавання мови має давню історію з кількома хвилями основних нововведень. Зовсім недавно галузь отримала вигоду від успіхів у глибокому навчанні та обробці великих даних. Про досягнення свідчить не лише сплеск академічних робіт, опублікованих у цій галузі, але прийняття у галузі різноманітних методів глибокого навчання при розробці та впровадженні систем розпізнавання мови. Глибоке навчання нарешті зробило розпізнавання мови досить точним, щоб воно стало корисно поза ретельно контрольованого середовища [27].

Перші пристрої, що реалізовували функцію розпізнавання мовлення були розроблені в 50-х роках минулого сторіччя та могли розпізнавати лише декілька визначених слів. Ближче до 2000-х було створено перший комерційний продукт, здатний розпізнавати безперервну мову. За останні кілька років програмне забезпечення для розпізнавання голосу збагатилось потужністю машинного навчання. Це ознаменувало зміну для мобільних

технологічних компаній, оскільки розпізнавання голосу дозволило користувачам керувати своїми пристроями ефективніше, ніж будь-коли раніше.

Можливості ШІ та технологій машинного навчання свідчать про те, що голосова інтеграція має попереду світле майбутнє, і що технологія голосової інтеграції найближчим часом може стати ближче до суті сучасного робочого місця.

Перший крок в розпізнаванні мови очевидний – потрібно передати звуки на комп'ютер. Звукові хвилі одномірні. У кожен момент часу у них є одне значення, залежне від амплітуди хвилі. На рисунку 3.1 зображено слово «Привіт».

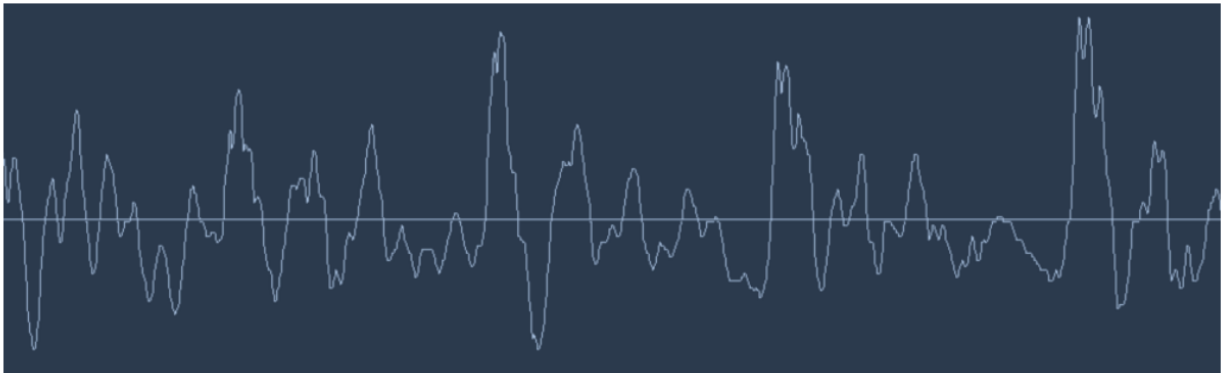


Рисунок 3.1 – Аудіофайл слова «Привіт»

Для того щоб розпізнати текст необхідно підключитися до нейронної мережі та представити аудіофайл у вигляді чисел. Щоб перетворити звукову хвилю в числа, необхідно знайти значення амплітуди хвилі в рівновіддалених точках. Даний процес називається дискретизацією. Дискретизація сигналу – заміна безперервного сигналу послідовністю чисел, які є поданням цього

сигналу з будь-якого базису [24]. Процес дискретизації представлено на рисунку 3.2.

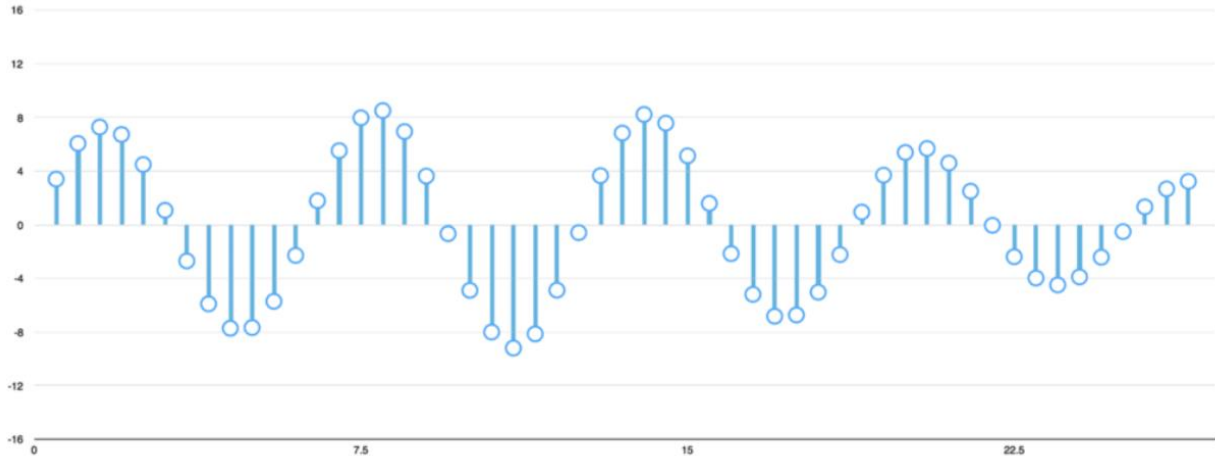


Рисунок 3.2 – Процес дискретизації звукової хвилі

Можна зчитувати дані тисячі разів в секунду і отримати числа, відповідні амплітуді звукової хвилі в цей момент часу.

Звук, що записується на CD-диски, дискретизується з частотою 44,1 кГц (44 100 відліків в секунду). Але для розпізнавання мови досить частоти дискретизації 16 кГц (16000 відліків в секунду), так як діапазон частот людської мови не настільки великий. Результат дискретизації представлено на рисунку 3.3.

```
[-1274, -1252, -1160, -986, -792, -692, -614, -429, -286, -134, -57, -41, -169, -456, -450, -541, -761, -1067, -1231, -1047, -952, -645, -489, -448, -397, -212, 193, 114, -17, -110, 128, 261, 198, 390, 461, 772, 948, 1451, 1974, 2624, 3793, 4968, 5939, 6057, 6581, 7302, 7640, 7223, 6119, 5461, 4820, 4353, 3611, 2740, 2004, 1349, 1178, 1085, 901, 301, -262, -499, -488, -707, -1406, -1997, -2377, -2494, -2605, -2675, -2627, -2500, -2148, -1648, -970, -364, 13, 260, 494, 788, 1011, 938, 717, 507, 323, 324, 325, 350, 103, -113, 64, 176, 93, -249, -461, -606, -909, -1159, -1307, -1544]
```

Рисунок 3.3 – Результат дискретизації слова «Привіт»

Завдяки теоремі Котельникова відомо, що для ідеального відтворення вихідної звукової хвилі досить використовувати частоту дискретизації, що вдвічі перевищує найвищу частоту записуваного звуку.

Теорема Котельникова – це фундаментальне твердження в області цифрової обробки сигналів, що зв'язує безперервні і дискретні сигнали і з якого випливає, що «будь-яку функцію $F(t)$, що складається з частот від 0 до f_1 , можна безперервно передавати з будь-якою точністю за допомогою чисел, що слідуєть один за одним через $1/2f_1$ секунд» [28].

Таким чином, використання більш високих частот дискретизації не призведе до кращої якості звуку. Потрібно використовувати частоту, що вдвічі більше частоти звуку.

Наступним кроком є розкладання складної звукової хвилі на прості складові звукові хвилі. Це робиться за допомогою математичної операції, званої перетворенням Фур'є. Маючи окремі звукові хвилі, необхідно скласти потужності звуку в кожній з них. Кінцевим результатом є оцінка важливості кожного частотного діапазону, від низьких частот до високих. Перетворення Фур'є – інтегральне перетворення однієї комплекснозначної функції дійсної змінної на іншу. Це перетворення розкладає дану функцію на осциляторні функції. Використовується для того, щоб розрахувати спектр частот для сигналів змінних у часі. Перетворення названо на честь французького математика Жана Батиста Жозефа Фур'є, який ввів поняття в 1822 році.

Якщо повторити цей процес на кожному 20-мілісекунди фрагменті аудіо, отримаємо спектрограму (кожен стовпець зліва направо є один 20-мілісекундний фрагмент). Після того, як є аудіо в форматі, з яким можна працювати далі, необхідно виконати навчання нейронної мережі на цих даних. Нейронна мережа буде отримувати аудіо-фрагменти довжиною 20 мс. Для кожного невеликого фрагмента мережу спробує визначити, яка буква була

виголошена. На рисунку схематично 3.4 зображено процес роботи нейронної мережі.

Для задачі розпізнавання мовлення, зазвичай використовують рекурентну нейронну мережу (РНН). РНН – це нейронна мережа, яка має пам'ять, що впливає на майбутні передбачення. Це пов'язано з тим, що, передбачення однієї літери впливає на наступну, яку теж необхідно передбачити. Процес роботи нейронної мережі схематично зображено на рисунку 3.4.

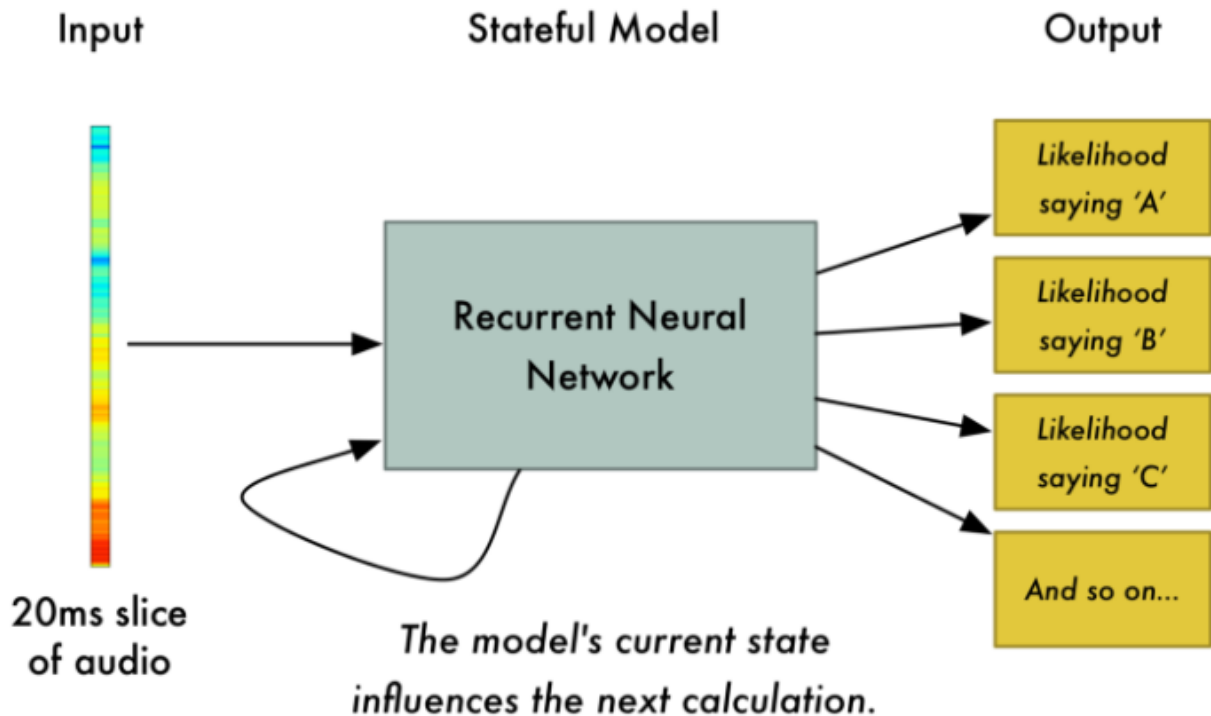


Рисунок 3.4 – Процес роботи нейронної мережі

Для задачі розпізнавання мовлення, зазвичай використовують рекурентну нейронну мережу (РНН). РНН – це нейронна мережа, яка має пам'ять, що впливає на майбутні передбачення. Це пов'язано з тим, що,

передбачення однієї літери впливає на наступну, яку теж необхідно передбачити.

РНН використовує ідею послідовної інформації. Послідовна інформація, яка зберігається в пам'яті РНН, використовується для передбачень. Ідея використовувати РНН замість традиційної нейронної мережі полягає в тому, що в традиційній нейромережі передбачається, що кожен вхід і кожен вихід не залежать один від одного. Отже, використання традиційної нейронної мережі є поганою ідеєю при обробці мовлення [41]. Для передбачення будь-яких слів у реченні потрібна інформація про слово, яке використано раніше, тобто слово, яке обробляється. Наявність пам'яті є однією з особливостей РНН, що відрізняє її від інших мереж та є більш ефективною, ніж інші для розпізнавання мовлення.

Розпізнавання мовлення – це складне завдання. Потрібно вирішити багато різних проблем: погана якість мікрофонів, фонові шуми, реверберація і луна, варіації вимови і акценту і багато іншого. Всі ці проблеми повинні бути враховані в навчальних даних, щоб переконатися, що нейронна мережа справляється з ними.

Розглянемо деякі основні проблеми, з якими стикаються дослідники при декодуванні мови в текст.

Шум. Машини звукозапису виявляють звукові хвилі, що генеруються через мову. Фонові шуми в приміщеннях ускладнюють системам розуміння та розрізнення конкретних звукових хвиль від голосу господаря. Це розмиває підхоплений пристроями звук, заплутуючи та обмежуючи його обробні можливості.

Відлуння. Відлуння – це в основному звукові хвилі, що відбиваються на різних поверхнях, таких як стіни, столи або інші меблі. Це призводить до неорганізованого повернення звукових хвиль назад до рецепторів, тим самим знижує чіткість.

Наголоси. Широкий спектр акцентів у кожній мові – ще один фактор, що призводить до труднощів у розпізнаванні мови. Якщо одне і те ж слово можна вимовляти різними способами, склади та фонетика одного і того ж слова, як правило, різняться, що ускладнює обробку машини.

Подібні звуки. Подібне звучання слів і фраз може перешкодити правильному кодуванню та декодуванню голосового повідомлення.

Помилка машини. Рівні точності виявлення голосу мають високий рівень помилок. Машини все ще стикаються з приблизно 8%–12% помилок, що більш ніж удвічі більше, ніж люди роблять у своїй щоденній промові. Помилки в кодуванні зібраних даних мають вирішальне значення для продуктивності, оскільки це перший крок для дії пристроїв звукозапису.

Дезорганізоване мовлення. Об'єднання слів у наших щоденних розмовах означає, що багато слів і фраз зливаються разом. Це не підходить для розпізнавання машини та голосу в текст, оскільки ускладнює розпізнавання конкретних слів або фраз, які впливатимуть на відповідні реакції та дії пристрою [24].

Вищезазначені фактори є і надалі будуть перешкодою для розвитку помічників ШІ. Однак швидкість, з якою розвиваються наука і техніка, всі великі компанії зосереджуються на створенні оптимальних пристроїв розпізнавання голосу, і рано чи пізно їх складки будуть виправлені.

3.2 Задача природної обробки мовлення (ПОМ)

Комп'ютери чудово працюють зі структурованою інформацією, наприклад таблицями в базах даних. Але люди спілкуються один з одним не таблицями, а словами. Для комп'ютерів не проста задача.

Велика частина інформації в світі не структурована – це просто тексти англійською або будь-якому іншому мовою. Проблемою отримання корисної

інформації з неструктурованих текстів займається особливий напрямок штучного інтелекту: обробка природної мови.

ПОМ – це аналіз лінгвістичних даних, найчастіше у формі текстових даних, таких як документи чи публікації, з використанням обчислювальних методів. Для дослідників основною метою процесу є побудова репрезентації тексту, включаючи структурні доповнення та розуміння з лінгвістики. Обробка природних мов, є одним з найважливіших елементів для створення комп'ютерного програмного забезпечення, яке забезпечує взаємодію людина-комп'ютер для зберігання початкової інформації, вирішення конкретних проблем та виконання повторюваних завдань, що вимагаються користувачем [29].

Процес читання і розуміння тексту сам по собі дуже складний. Крім того, люди часто не дотримуються логіки і послідовність розповіді. Для реалізації рішення будь-якої складної комплексної задачі в машинному навчанні зазвичай необхідно розбити проблему на дуже маленькі частини і вирішувати їх окремо. Поєднавши кілька таких моделей, що поставляють один одному дані, можна отримувати задовільний результати.

Хоча обробка природної мови не є новою наукою, технологія швидко прогресує завдяки підвищеному інтересу до комунікацій «людина-машина», а також наявності великих даних, потужних обчислень та вдосконалених алгоритмів.

Спершу багато систем обробки мови за допомогою ручного кодування набору правил у поєднанні зі словниковим пошуком, наприклад, шляхом написання граматики. Більш нові системи, засновані на алгоритмах машинного навчання, мають багато переваг.

Процедури автоматичного навчання можуть використовувати алгоритми статистичного висновку для створення моделей, стійких до незнайомого введення (наприклад, містять слова або структури, яких раніше не бачили), і

до помилкового введення (наприклад, з неправильно написаними словами або випадково пропущеними словами).

Системи, засновані на автоматичному засвоєнні правил, можуть бути більш точними, просто за рахунок того, що отримують більше вхідних даних. Однак системи, засновані на рукописних правилах, можуть бути більш точними лише за рахунок збільшення складності правил, що є набагато складнішим завданням. Зокрема, існує обмеження складності систем, заснованих на рукописних правилах, поза якими системи стають дедалі більше некерованими.

Незважаючи на популярність машинного навчання в дослідженнях ПОМ, ручні методи все ще використовуються коли обсяг навчальних даних є недостатнім для успішного застосування методів машинного навчання.

Розглянемо процес розпізнавання мовлення.

Перший етап – розбити текст на окремі речення. Можна припустити, що кожне речення – це самостійна думка або ідея. Простіше навчити програму розуміти єдина пропозиція, а не цілий параграф. Можна було б просто розділяти текст за певними знаками пунктуації. Але сучасні алгоритми мають в запасі складніші методи, які підходять навіть для роботи з невідформатованими фрагментами.

Наступний етап – виділення окремих слів або токенів – токенізація. В англійській або українській мові необхідно відокремлювати фрагмент тексту кожен раз, коли зустрічаємо пробіл. Розділові знаки теж є токенами, оскільки можуть мати важливе значення.

Третій етап – на основі заздалегідь підготовленої класифікаційної моделі відбувається аналіз кожного токена на предмет, якою частиною мови він є. Знаючи роль кожного слова в реченні, можна зрозуміти його загальний зміст. Класифікаційна модель заздалегідь навчається на великій кількості речень з зазначеними частинами мови.

Четвертий етап – лематизація. Якщо слова можуть мати різні форми, то необхідно знайти основну (лему). Лематизація зазвичай виконується простим пошуком форм в таблиці. Крім того, можна додати деякі призначені для користувача правила для аналізу слів.

На п'ятому етапі необхідно встановити взаємозв'язок між словами в реченні. Це називається парсинг залежностей. Кінцева мета цього кроку – побудова дерева, в якому кожен токен має єдиного батька.

Шостим етапом є розпізнавання іменованих сутностей. В реченні зазвичай присутні іменники. Частина іменників може позначати реальні речі. Наприклад, географічні сутності, відомі людини, назви літературних творів. На цьому кроці відбувається автоматичне отримання списку реальних об'єктів, згаданих в тексті. Мета розпізнавання іменованих сутностей – виявити такі іменники і зв'язати їх з реальними концепціями. Системи, що відповідають за розпізнавання іменованих сутностей, не просто переглядають словники. Вони аналізують контекст токена в реченні і використовують статистичні моделі, щоб вгадати який об'єкт він представляє. Хороші такі системи здатні відрізнити актрису Brooklyn Decker від міста Brooklyn. Більшість систем, що розпізнають іменовані сутності виділяють такі типи об'єктів, як імена людей, назви компаній, географічні позначення (і фізичні, і політичні), продукти, дату і час, грошові суми, події.

Сьомим етапом є вирішення кореференції. В англійській мові є багато займенників. Це скорочення, якими замінюються в тексті справжні імена і назви. Людина може простежити взаємозв'язок цих слів, ґрунтуючись на контексті. Але ПОМ-модель не знає про те, що означають займенники, адже вона розглядає лише одне речення за раз. Вирішення кореференції називається відстеження займенників в реченнях з метою вибрати всі слова, що відносяться до однієї сутності. Вирішення кореференції – один з найважчих кроків, він навіть складніше парсинга пропозицій. В області глибокого навчання вже

з'явилися способи його реалізації, вони досить точні, але все ще не досконалі [26].

3.2.1 Задача виявлення ключових слів

Виявлення ключових слів – це одна з головних задач в обробці природних мов. Це процес автоматичної ідентифікації термінів, які найкраще описують тему документа, що є корисним для віднесення документів до певних категорій, позначення або упорядкування. Ключові фрази та ключові слова – це термінологія, яка використовується для визначення термінів, що представляють найбільш релевантну інформацію, що міститься в документі.

Більшість даних є неструктурованими, та кількість ресурсів робить процес індексації та пошуку досить складним завданням. Переважна більшість документів також не пов'язані з жодними ключовими словами, що змушує користувачів читати весь документ, щоб отримати загальне уявлення про його зміст. Процес створення набору ключових слів для документів вручну – це клопітке та трудомістке завдання, що легко піддається помилкам. Більше того, загальна кількість документів робить ручне вилучення ключових слів неможливим [39].

На сьогоднішній день найпопулярніші алгоритми виявлення ключових слів – це RAKE та TF-IDF. Ці підходи працюють зовсім по-різному і дають різні результати для різних задач.

RAKE – швидкий автоматичний алгоритм вилучення ключових слів. Алгоритм використовується для вилучення та ранжування ключових слів або фраз з документа без будь-якого іншого контексту, крім самого документа.

Алгоритм працює, спочатку створюючи фрази із слів, які знаходяться між стоп-словами, а потім відкидає самі стоп-слова.

Після відбувається підрахунок кількості випадків, коли кожне слово зустрічається у всіх фразах, щоб знайти оцінку частоти кожного слова.

Потім алгоритм підсумовує кількість випадків, коли кожне слово зустрічається з кожним другим словом у всіх фразах, щоб отримати оцінку ступеня або співіснування.

Щоб отримати остаточний рейтинговий список фраз, він генерує оцінку для кожного слова у фразі шляхом ділення кількості випадків слова на частоту слова в документі, а потім підсумовує ці бали за всіма словами в фразі.

Це означає, що якщо слова багато в документі, але він є розподіленим випадковим чином по фразах, це менш важливе слово і зменшує ранг фраз, в яких воно зустрічається. Навпаки, якщо слово постійно зустрічається в фразах з одними й тими ж словами, то це збільшить рейтинг фрази, в якій він знаходиться.

TF-IDF – алгоритм зворотної частоти документів, що визначає кількість випадків, коли слово зустрічається в документі, а потім порівнює цей підрахунок із кількістю випадків, коли слово зустрічається в інших документах колекції. TF-IDF знаходить числову характеристику, яка має на меті відобразити, наскільки важливим є слово для документів в колекції. Значення TF-IDF збільшується пропорційно кількості випадків, коли слово з'являється в документі, а потім компенсується кількістю документів в колекції, що містять це слово. Кінцевим результатом є список слів, відсортованих за тим, наскільки вони важливі для колекції в цілому.

Цей алгоритм починається з обчислення частоти зустрічання слів або оцінки TF для кожного слова в документі. Це обчислюється як кількість слів у документі, поділена на загальну кількість слів у документі.

Щоб отримати зворотну частоту документів або оцінку IDF для кожного слова, алгоритм обчислює загальну кількість документів, поділену на кількість документів, у яких було знайдено слово, а потім бере логарифм цього рівняння.

Наступним кроком, показники TF та IDF перемножуються, щоб дати слову остаточний рейтинг важливості в документі.

Таким чином, якщо є лише один документ, то RAKE є ефективним рішенням для знаходження найчастіше вживаних словосполучень та присвоєння їм рейтингу важливості. Якщо є корпус документів та необхідно знайти ключові слова, а не фрази чи короткий зміст, тоді TF-IDF є більш ефективним. Отже, для задачі виявлення ключових слів завдання TF-IDF - найбільш прийнятний.

3.3 Задача прогнозування часових витрат, необхідних на виконання задачі

Вхідними даними задачі є опис завдань або подій, які користувачеві необхідно додати в календар, а на виході алгоритм формує графік користувача, спланований найбільш ефективним чином. Задля підвищення ефективності управління часом та поліпшення використання планувальника необхідно використовувати техніки тайм-менеджменту.

В задачі управління витратами часу основною проблемою є проблема розподілу ресурсів у різних науково-технічних аспектах. Де б не була присутня система, яка б приймала у вхідних даних набір завдань, вони завжди включали б концепцію розподілу ресурсів для набору завдань. Данна концепція має в основі концепцію кластерів ресурсів та планування кластерів. Кластерів може бути декілька, наприклад, якщо розглянути проблему мережевого розподілу ресурсів, яка включає такі ресурси, як центральний процесор, блоки вводу-виводу, пам'ять тощо.

В задачі управління витратами часу головним ресурсом є один кластер та один тип ресурсів – час. Отже час, в рамках даної задачі, можна назвати кластером ресурсів, і кожне завдання вимагатиме певної кількості одиниць

ресурсу, що належить цьому кластеру. Схема розподілу ресурсів зображено на рисунку 3.5.

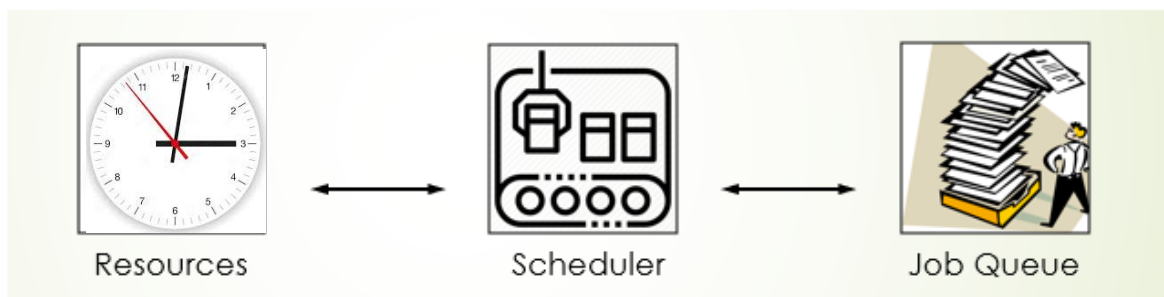


Рисунок 3.5 – Схема розподілу ресурсів

Метою вирішення задачі управління витратами часу є мінімізація витрат будь-яких ресурсів, тобто ці ресурси не повинні простоювати дуже довго, особливо якщо майбутні завдання вимагатимуть їх для свого завершення.

Отже, використання кластера та ефективність приймаються як найважливіші показники для належного прийняття рішень щодо управління ресурсами та планування.

Задля того, щоб визначити, в якому порядку додавати завдання до календаря, можна використовувати матрицю Ейзенхауера. Тобто термінові і важливі завдання будуть призначені на ранкові години, нетермінові і неважливі на вечірні години, що підвищить ефективність виконання завдань.

Для прогнозування часових витрат – необхідно вирішити задачу багатокласової класифікації. Класифікація – один з розділів машинного навчання, присвячений вирішення наступного завдання. Є множина об'єктів (ситуацій), розділених деяким чином на класи. Визначено сукупність об'єктів, для яких відомо, до яких класів вони належать. Ця сукупність називається навчальною вибіркою. Класова приналежність інших об'єктів не відома. Потрібно побудувати алгоритм, здатний класифікувати довільний об'єкт з

початкової множини. У машинному навчанні завдання класифікації відноситься до розділу навчання з учителем.

Задачі класифікації можна розділити за типом вхідних даних:

- признаковий опис – найбільш поширений випадок, кожен об'єкт описується набором характеристик, які називаються ознаками, що можуть бути числовими або нечисловими;

- матриця відстаней між об'єктами – кожен об'єкт описується відстанями до всіх інших об'єктів навчальної вибірки, з цим типом вхідних даних працюють деякі методи, зокрема, метод найближчих сусідів, метод парзенівського вікна, метод потенційних функцій;

- часовий ряд або сигнал являє собою послідовність вимірювань в часі, кожен вимір може представлятися числом, вектором, а в загальному випадку - признаковим описом досліджуваного об'єкта в даний момент часу;

- зображення або відеоряд.

Зустрічаються і більш складні випадки, коли вхідні дані подаються у вигляді графів, текстів, результатів запитів до бази даних. Як правило, вони наводяться до першого або другого випадку шляхом попередньої обробки даних і вилучення ознак.

Також задачі класифікації можна розділити за типом класів:

- двокласова класифікація – найбільш простий в технічному відношенні випадок, який служить основою для вирішення більш складних завдань;

- багатокласова класифікація – коли число класів є більше двох (наприклад, при розпізнаванні ієрогліфів або злитого мовлення), завдання класифікації стає більш важкою;

- непересічні класи;

- пересічні класи – об'єкт може належати одночасно до кількох класів;
- нечіткі класи – потрібно визначати ступінь належності об'єкта кожному з класів, зазвичай це дійсне число від 0 до 1.

Для прогнозування часових витрат – необхідно вирішити задачу багатокласової класифікації з ознаковими описами ознак.

4 РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ДЛЯ УПРАВЛІННЯ ВИТРАТАМИ ЧАСУ

Основним недоліком PExA є те, що вона довго навчається перевагам користувача і робить помилки на ранньому етапі.

Таким чином, пропонується розробити програму, що буде використовувати готові функції CALO, буде сфокусована на сфері тайм-менеджменту та усуватиме основний недолік PExA. Запропонований підхід до створення інтелектуальної системи тайм-менеджменту дозволяє частково усунути цей недолік за рахунок застосування ансамблю класифікаторів, заснованих на використанні такого алгоритму машинного навчання, як наївний класифікатор Баеса (НБА).

На рисунку 4.1 зображено потік даних у запропонованій моделі.

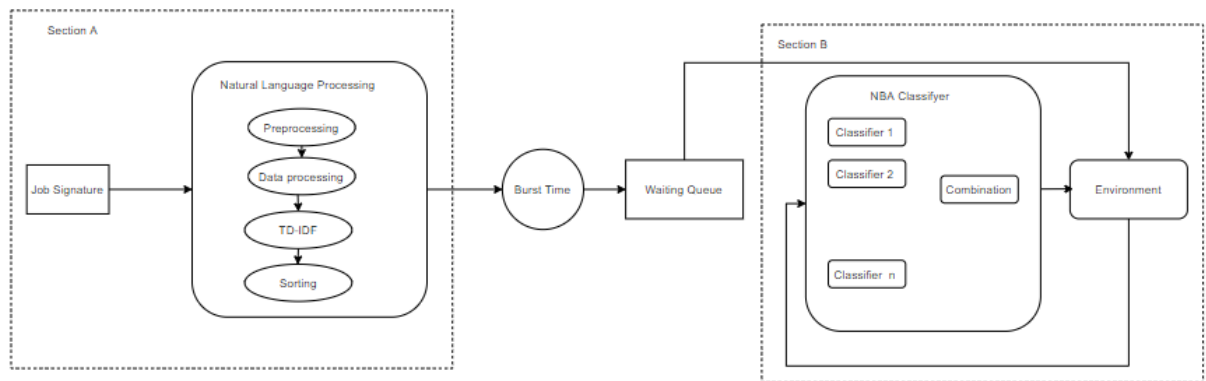


Рисунок 4.1 – Потік даних у запропонованій моделі

В запропонованому підході реалізується алгоритм планування діяльності користувача. Вхідними даними цього алгоритму є опис завдань або подій, які

користувачеві необхідно додати в календар, а на виході алгоритм формує графік користувача, спланований найбільш ефективним чином, використовуючи таку техніку тайм-менеджменту, як матриця Ейзенхауера.

Як показано на рисунку, пропонується розділити модель на 2 розділи: розділ А та розділ В. Розділ А приймає опис завдання або події, яку необхідно додати до календаря як вхідні данні та виділяє ключові слова за допомогою засобів ПОМ. Ключові слова надсилається як вхідні дані до черги очікування. Всі завдання з черги очікування надходять у середовище як початковий стан.

Виявлення ключових слів – це техніка аналізу тексту, яка автоматично витягує з тексту найуживаніші та найважливіші слова та вирази. Це допомагає узагальнити зміст текстів та визначити основні теми. Аналіз тексту використовує машинне навчання з ПОМ, щоб розбити людську мову таким чином, щоб її можна було зрозуміти та проаналізувати комп'ютером.

Розділ В приймає ключові слова завдання в якості вхідних даних, а на виході віддає прогнозований час, що необхідно для виконання завдання. За ключовими словами кожного завдання необхідно навчитися передбачати, скільки часу буде потрібно на виконання завдання, використовуючи передісторію виконання користувачем подібних завдань. Використовуючи надалі оцінку часових витрат, запропоновану системою, можна домогтися більшої точності в складанні розкладів, так як самостійно користувач не завжди здатний правильно оцінити необхідний для виконання завдань час.

Час виконання кожної задачі j можна представити вектором

$$r_j = (r_1, r_2, \dots, r_k), \quad (4.1)$$

де r_j – число дискретних одиниць часу i , що необхідно для виконання завдання j та $i \in [0, k]$. T_j визначимо як час, що потрібен для виконання j .

Розбиття завдань за матрицею Ейзенхауера включає розбиття чотири категорії. наступні категорії:

- P1 – «важливі і термінові»: наприклад, критичні ситуації, проекти з швидким дедлайном, задачі такого типу необхідно виконувати без зволікання;
- P2 – «важливі і нетермінові»: наприклад, стратегічне планування в бізнесі, особистісний розвиток, для завдань даного типу необхідно встановлювати тимчасові рамки;
- P3 – «неважливі і термінові»: наприклад, спонтанні прохання або доручення, що не належать до безпосередніх обов'язки, тощо, по можливості від таких прохань слід відмовлятися або передоручати іншим;
- P4 – «неважливі і нетермінові»: це завдання, виконання яких можна відкласти.

Таким чином, необхідно відсортувати завдання протягом дня відповідно до номеру пріоритету (термінові і важливі завдання будуть призначені на ранкові години, нетермінові і неважливі на вечірні години).

В системі прийняті такі правила: користувач не має переваги щодо часу, коли завдання має бути виконано, і алгоритм може запропонувати будь-який вільний часовий інтервал протягом робочого дня; всі завдання в межах робочого дня упорядковано відповідно до номеру пріоритету; мінімальна тривалість тимчасового інтервалу становить 60 хвилин; кількість годин в робочому дні, що користувач може присвятити виконанню завдань – 6.

4.1 Вирішення задачі виявлення ключових слів

Розглянемо високорівневий підхід до вилучення ключових слів. Зображення процесу вилучення ключових представлено на рисунку 4.2.

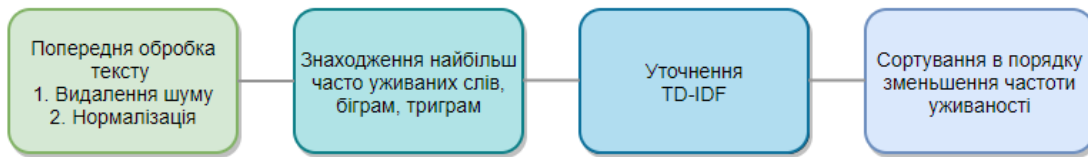


Рисунок 4.2 – Зображення процесу вилучення ключових слів

Першим кроком необхідно зробити попередню обробку тексту. Попередня обробка тексту – це приведення його у форму, яка передбачувана та може аналізуватися в рамках поставленої задачі. Попередню обробку тексту можна розділити на дві великі категорії – видалення шуму та нормалізація. Компоненти даних, зайві для аналізу основного тексту, можна розглядати як шум. Схема попередньої обробки тексту зображена на рисунку 4.3.



Рисунок 4.3 – Схема попередньої обробки тексту

Видалення шуму – це видалення цифр, символів та інших фрагментів тексту, які можуть заважати аналізу тексту. Це один із найважливіших етапів попередньої обробки тексту. Процес сильно залежить від домену. Наприклад, у твітах шумом можуть бути всі спеціальні символи, за винятком хеш-тегів, оскільки це означає поняття, які можуть характеризувати твіт. Проблема шуму полягає в тому, що він може дати результати, які не відповідають подальшим завданням користувача.

Приклад видалення шуму зображено на рисунку 4.4.

	raw_word	cleaned_word
0	..trouble..	trouble
1	trouble<	trouble
2	trouble!	trouble
3	<a>trouble	trouble
4	1.trouble	trouble

Рисунок 4.4 – Приклад видалення шуму

Також необхідно призвести всі слова до нижнього регістру – це одна з найпростіших та найефективніших форм попередньої обробки тексту. На рисунку 4.5 наведено приклад приведення до нижнього регістру.

Обробка декількох форм одного і того ж слова називається нормалізацією. Лематизація, що відноситься до нормалізації виконується на основі словника та морфологічного аналізу слів, вона націлена на видалення флексивних закінчень і повернення до словникової початкової форми слова. На рисунку 4.6 наведено приклад лематизації.

Raw	Lowercased
Canada CanadA CANADA	canada
TOMCAT Tomcat toMcat	tomcat

Рисунок 4.5 – Приклад приведення до нижнього регістру

	original_word	stemmed_words
0	connect	connect
1	connected	connect
2	connection	connect
3	connections	connect
4	connects	connect

Рисунок 4.6 – Приклад лематизації

Також на цьому кроці необхідно видалити з даних стоп-слова. Стоп-слова включають велику кількість прийменників, займенників, сполучників тощо у реченнях. Ці слова потрібно видалити перед аналізом тексту, щоб часто вживаними словами були в основному ті, що мають відношення до контексту, а не загальноживані слова, що використовуються в тексті.

Наступним кроком є токенізація. Текст повинен бути перетворений у формат, який можна інтерпретувати за допомогою алгоритмів машинного навчання. Токенізація – це процес перетворення безперервного тексту в вектор слів. На цьому етапі необхідно знайти вектор найбільш часто уживаних слів, біграм та триграм.

Наступним кроком є уточнення вектора найбільш часто уживаних слів. Проблема простого підрахунку слів полягає в тому, що слова, що зустрічаються у всіх документах корпусу можуть послабляти вплив слів, що часто зустрічаються лише в одному документі. Дану проблему вирішує TF-IDF.

TF-IDF складається з 2 компонентів:

- TF – частота слова;
- IDF – обернена частота документа.

TF – це відношення числа входжень деякого слова до загальної кількості слів документа. Таким чином, оцінюється важливість слова t в межах окремого документа.

$$tf(t, d) = \frac{n_t}{\sum_k n_k}, \quad (4.2)$$

де n_t – число входжень слова t в документ, а в знаменнику – загальне число слів в даному документі.

IDF – інверсія частоти, з якою слово зустрічається в документах колекції. Використання IDF зменшує вагу широкоживаних слів.

$$idf(t, D) = \log \frac{|D|}{|\{d_i \in D \mid t \in d_i\}|}, \quad (4.3)$$

де $|D|$ – кількість документів колекції;

$|\{d_i \in D \mid t \in d_i\}|$ – кількість документів, в яких зустрічається слово t (коли $n_i \neq 0$).

Вибір основи логарифму у формулі не має значення, адже зміна основи призведе до зміни ваги кожного слова на постійний множник, тобто вагове співвідношення залишиться незмінним. Більшу вагу TF-IDF отримують слова з високою частотою появи в межах документа та низькою частотою вживання в інших документах колекції.

Міра TF-IDF є твором добутком співмножників.

$$tf - idf(t, d, D) = tf(t, d) \times idf(t, D), \quad (4.4)$$

Більшу вагу в TF-IDF отримують слова з високою частотою в межах конкретного документа і з низькою частотою вживань в інших документах.

Для задачі, що розглядається, документ – це одне завдання, корпус документів – набір завдань.

4.2 Вирішення задачі прогнозування часових витрат, необхідних на виконання задачі

В системі реалізована функція прогнозування часових витрат, що необхідні для виконання певної задачі на основі даних про задачі, що були виконання раніше.

Як було зазначено, передбачається, що часовий інтервал складає 60 хвилин, а кількість годин, що працівник може присвятити на виконання завдань – 6. Таким чином необхідно реалізувати задачу класифікації. Пропонується використовувати ансамбль класифікаторів з використанням НБА. Теоретичні та практичні результати показують, що результат комбінації

класифікаторів є найбільш ефективний, коли класифікатори є незалежними. Але в даного правила є виключення, а саме, коли похибка окремих класифікаторів ансамблю перевищує 50%. Для побудови незалежних класифікаторів найбільш ефективним методом є навчання окремих членів ансамблю на окремих підмножинах ознак. Таким чином, побудова ансамблю класифікаторів на основі декомпозиції вихідного набору ознак, що описують об'єкти даних, в більшості випадків має переваги [36].

Загальна схема реалізації запропонованого підходу до навчання модуля класифікації нових завдань, що надходять в систему представлена на рисунку 4.7. Відповідно до цієї схеми випадковим чином формуються набори множин ознак навчальної вибірки на n підмножин.

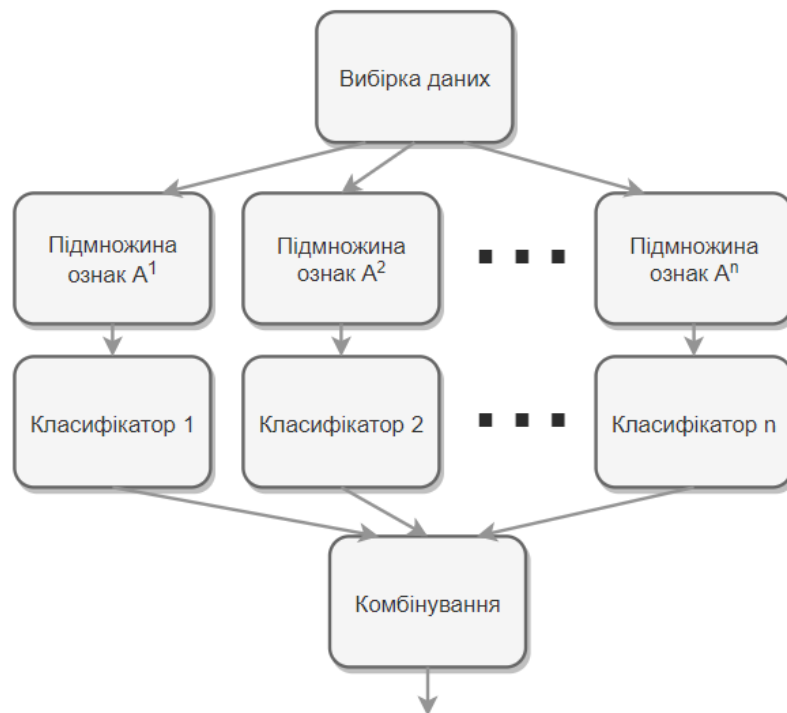


Рисунок 4.7 – Схема реалізації запропонованого підходу

В якості алгоритму класифікації було обрано НБА з допущенням про незалежність ознак. В його основі лежить теорема Байєса. Теорема Байєса – це одна з основних теорем елементарної теорії ймовірностей, яка дозволяє визначити ймовірність певної ознаки за умови, що наявність цієї ознаки в класі не пов'язано з наявністю будь-якої іншої ознаки. Іншими словами, за формулою Байєса можна більш точно перерахувати ймовірність, взявши до уваги як раніше відому інформацію, так і дані нових спостережень.

Теорема Байєса дозволяє розрахувати апостеріорну ймовірність $P(c|x)$ на основі $P(c)$, $P(x)$ і $P(x|c)$.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}, \quad (4.5)$$

де $P(c|x)$ – апостеріорна ймовірність даного класу c (тобто даного значення цільової змінної) при даному значенні ознаки x ;

$P(x|c)$ – ймовірність даного значення ознаки x при даному класі c ;

$P(c)$ – апріорна ймовірність даного класу c ;

$P(x)$ – апріорна ймовірність даного значення ознаки x .

Моделі на основі НБА досить прості і вкрай корисні при роботі з дуже великими наборами даних. При своїй простоті НБА здатний перевершити велику кількість алгоритмів класифікації.

До переваг НБА належить:

- класифікація, в тому числі багатокласова, виконується легко і швидко, що є суттєвою перевагою для використання в ансамблі;
- коли допущення про незалежність виконується, НБА перевершує інші алгоритми, такі як логістична регресія, і при цьому вимагає менший обсяг навчальних даних;

- класифікація в режимі реального часу: НБА дуже швидко навчається, тому його можна використовувати для обробки даних в режимі реального часу;
- багатокласова класифікація: НБА забезпечує можливість багатокласової класифікації, це дозволяє прогнозувати ймовірність для безлічі значень цільової змінної.

Під час експериментів було створено дві схеми розбиття ознак.

В першій схемі окремі класифікатори ансамблю навчаються на окремій підмножині ознак. Тобто, множина вхідних ознак розбивається на декілька непересічних підмножин. Приклад схеми розподілу вхідних ознак з трьома класифікаторами та шести ознаками представлена на рисунку 4.8.

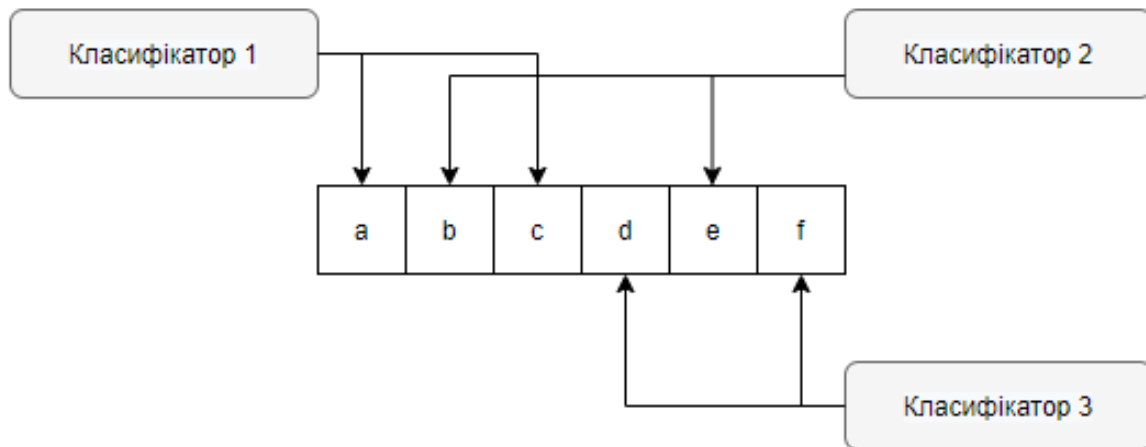


Рисунок 4.8 – Перша схема розподілу вхідних ознак

В другій схемі існує можливість визначення пересічних підмножин ознак. Приклад схеми розподілу вхідних ознак з трьома класифікаторами та шести ознаками представлена на рисунку 4.9.

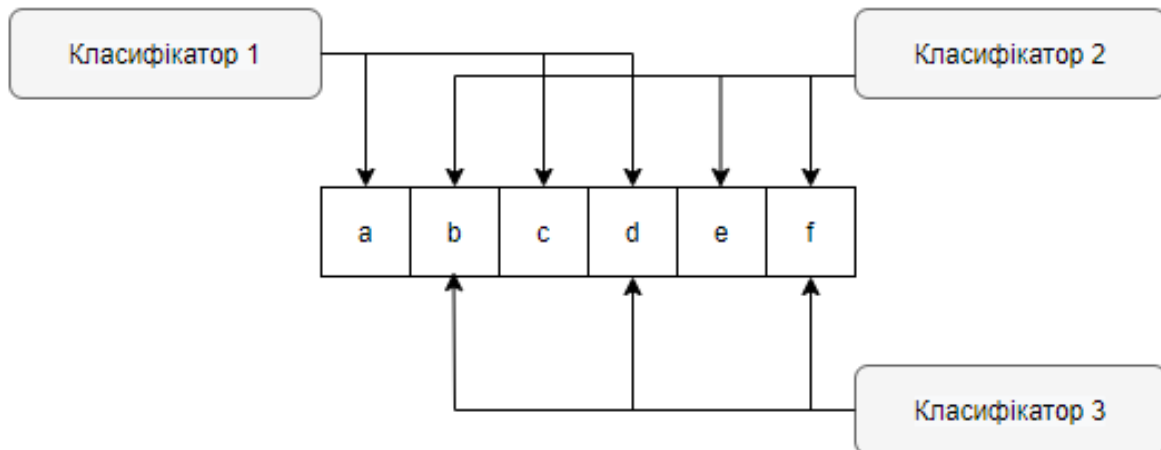


Рисунок 4.9 – Друга схема розподілу вхідних ознак

Після того, як для поточного завдання встановлюється час, що є необхідним для його виконання, завдання відсортовуються за пріоритетами. Перед цим система розбиває загальний часовий інтервал між початком і закінченням процесу рішення вихідної сукупності завдань на кілька інтервалів, кількість і тривалість яких є найбільш прийнятними з точки зору заданих пріоритетів. Пріоритети виконання завдань (P1, P2, P3 або P4) задаються за допомогою матриці Ейзенхауера, при цьому кожному пріоритету відповідає своя рекомендована тривалість інтервалів. Таким чином, завдання, що мають пріоритет P1 будуть в календарі користувача раніше, ніж завдання з пріоритетом P4.

4.3 Результати експериментів

Розроблений підхід до побудови ансамбля класифікаторів з використанням НБА було протестовано на виборці «Запит інформації» від коледжу Шорлайн.

Для оцінки точності класифікації ансамблю класифікаторів було виконано розбиття вихідні набори даних на дві частини, один з яких використовувався для навчання ансамблю (навчальна вибірка), а другий – для тестування результатів (тестова вибірка).

Вибірка включає в себе 457 наборів даних, 20 ознак та 6 класів.

Було поставлено п'ять експерименти з побудови ансамблю класифікаторів:

- побудова ансамблю з трьох класифікаторів на основі непересічних підмножин ознак;
- побудова ансамблю з шести класифікаторів на основі непересічних підмножин ознак;
- побудова ансамблю з дев'яти класифікаторів на основі непересічних підмножин ознак;
- побудова ансамблю з трьох класифікаторів на основі пересічних підмножин ознак;
- побудова ансамблю з шести класифікаторів на основі пересічних підмножин ознак.

Також було проведено експеримент без використання ансамблів та реалізовано дві схеми розбиття ознак на підмножини.

Результати експериментів наведено в таблиці 4.1.

Як видно з результатів обчислювальних експериментів, запропонований в даній роботі підхід до побудови ансамблю класифікаторів забезпечує отримання більш високої точності класифікації об'єктів, що характеризуються великою кількістю ознак. Було підтверджено, що тренування на незалежних множинах ознак є більш ефективним. Використання кількості ансамблів наближеної до кількості класів забезпечує більш високу точність класифікації.

Таблиця 4.1 – Результати експериментів

	Кількість класифікаторів	Точність класифікації
Класифікатор (метод НБА)	1	0,829
Ансамбль класифікаторів (схема 1)	3	0,886
	6	0,912
	9	0,901
Ансамбль класифікаторів (схема 2)	3	0,879
	6	0,898

4.4 Практична реалізація на мові Python

Розглянемо приклад реалізації виділення ключових слів на мові Python.

Мова програмування Python має ряд переваг [40]:

- популярна високорівнева мова програмування з динамічною семантикою;
- досить проста для роботи, саме тому одна з найпоширеніших: використання Python знижує вартість розробки та обслуговування програм;
- Python досить легко взаємодіє з іншими мовами, особливо з мовами групи C;
- Python активно використовується для вирішення завдань інтелектуального аналізу даних і машинного навчання, тобто для розробки додатків, які можуть навчатися і видавати результати автоматично, без людського втручання;
- Python має відмінну продуктивність при обробці даних.

Під час реалізації було використано наступні модулі:

- pandas: інструменту для аналізу та маніпулювання даними;

- nltk: набір бібліотек та програм для символічної та статистичної обробки природничої англійської мови;
- matplotlib: створення статичної, анімованої та інтерактивної візуалізації в Python;
- pillow: бібліотека зображень;
- sklearn: бібліотека з реалізованими інструментами машинного навчання.

На першому кроці необхідно сформувати список стоп-слів. Модуль nltk пропонує список найпоширеніших стоп-слів англійською мовою, які на першому кроці необхідно імпортувати. На цьому кроці також можна додати власні стоп-слова. Також можна завантажити набір власних стоп-слів із текстового файлу. Програмний код формування стоп-слів подано у лістингу 4.1.

Лістинг 4.1 – Програмний код формування стоп-слів

```
import re
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.tokenize import RegexpTokenizer
from nltk.stem.wordnet import WordNetLemmatizer
stop_words = set(stopwords.words("english"))
csw = set(line.strip() for line in open('custom-stopwords.txt'))
csw = [sw.lower() for sw in csw]
stop_words = sorted(stop_words.union(csw))
```

Попередньо обробимо набір даних, щоб отримати очищений і нормалізований текст. Попередня обробка включає видалення знаків пунктуації, міток та спеціальних символів із тексту, а потім нормалізацію того, що залишилось, у впізнавані слова. Програмний код обробки даних подано у лістингу 4.2.

Лістинг 4.2 – Програмний код обробки даних

```

corpus = []
dataset['word_count'] = dataset[dataacol].apply(lambda x:
len(str(x).split(" ")))
ds_count = len(dataset.word_count)
for i in range(0, ds_count):
# Remove punctuation
text = re.sub('[^a-zA-Z]', ' ', str(dataset[dataacol][i]))
# Convert to lowercase
text = text.lower()
# Remove tags
text=re.sub("</?.*?>"," <> ",text)
# Remove special characters and digits
text=re.sub("(\\d|\\W)+"," ",text)
# Convert to list from string
text = text.split()
# Stemming
ps=PorterStemmer()
# Lemmatisation
lem = WordNetLemmatizer()
text = [lem.lemmatize(word) for word in text if not word in
stop_words]
text = " ".join(text)
corpus.append(text)

```

Токенізація тексту. На цьому кроці вже можна створити набір ключових слів та n -грам, а саме фраз із двома та трьома словами (біграми та триграми). Програмний код токенизації подано у лістингу 4.3.

Лістинг 4.3 – Програмний код обробки даних

```
from sklearn.feature_extraction.text import CountVectorizer
import re
cv=CountVectorizer(max_df=0.8,stop_words=stop_words,
max_features=10000, ngram_range=(1,3))
X=cv.fit_transform(corpus)
```

Метод, що знаходить n найчастіше використовуваних слів. Програмний код подано у лістингу 4.4.

Лістинг 4.4 – Програмний код знаходження n найчастіше використовуваних слів

```
def get_top_n_words(corpus, n=None):
    vec = CountVectorizer().fit(corpus)
    bag_of_words = vec.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in
vec.vocabulary_.items()]
    words_freq =sorted(words_freq, key = lambda x: x[1],reverse=True)
    return words_freq[:n]
```

Побудуємо графік найчастіше використовуваних слів. Програмний код побудови графіку подано у лістингу 4.5. Результат роботи коду представлено на рисунку 4.10.

Лістинг 4.5 – Програмний код побудови графіку

```
top_words = get_top_n_words(corpus, n=20)
top_df = pandas.DataFrame(top_words)
top_df.columns=["Keyword", "Frequency"]
print(top_df)
top_df.to_csv(file_prefix + '_top_words.csv')

import seaborn as sns
sns.set(rc={'figure.figsize':(13,8)})
g = sns.barplot(x="Keyword", y="Frequency", data=top_df,
palette="Blues_d")
g.set_xticklabels(g.get_xticklabels(), rotation=45)
g.figure.savefig(file_prefix + "_keyword.png", bbox_inches =
"tight")
```

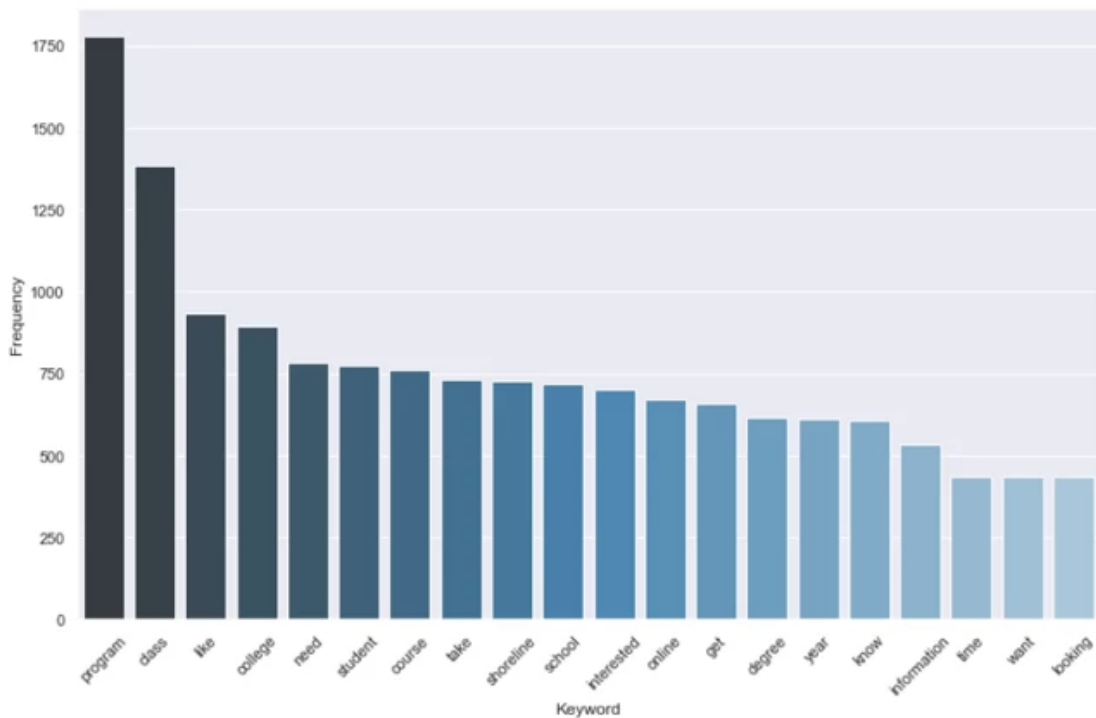


Рисунок 4.10 – Графік двадцяти ключових слів за частотою зустрічей

Метод, що знаходить n найчастіше використовуваних біграм. Програмний код подано у лістингу 4.6.

Лістинг 4.6 – Програмний код знаходження n найчастіше використовуваних біграм

```
def get_top_n2_words(corpus, n=None):
    vec1 = CountVectorizer(ngram_range=(2,2),
        max_features=2000).fit(corpus)
    bag_of_words = vec1.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in
        vec1.vocabulary_.items()]
    words_freq =sorted(words_freq, key = lambda x: x[1],
        reverse=True)
    return words_freq[:n]
```

Подібно було створено метод, що знаходить n найчастіше використовуваних триграм. Та побудовані обидва графіки.

Графіки двадцяти біграм за частотою зустрічей та двадцяти триграм за частотою зустрічей представлено на рисунках 4.11 та 4.12 відповідно.

Наступним кроком знайдемо список найчастіше зустрічаємих слів за TF-IDF. Програмний код пошуку найчастіше зустрічаємих слів за TF-IDF подано у лістингу 4.7.

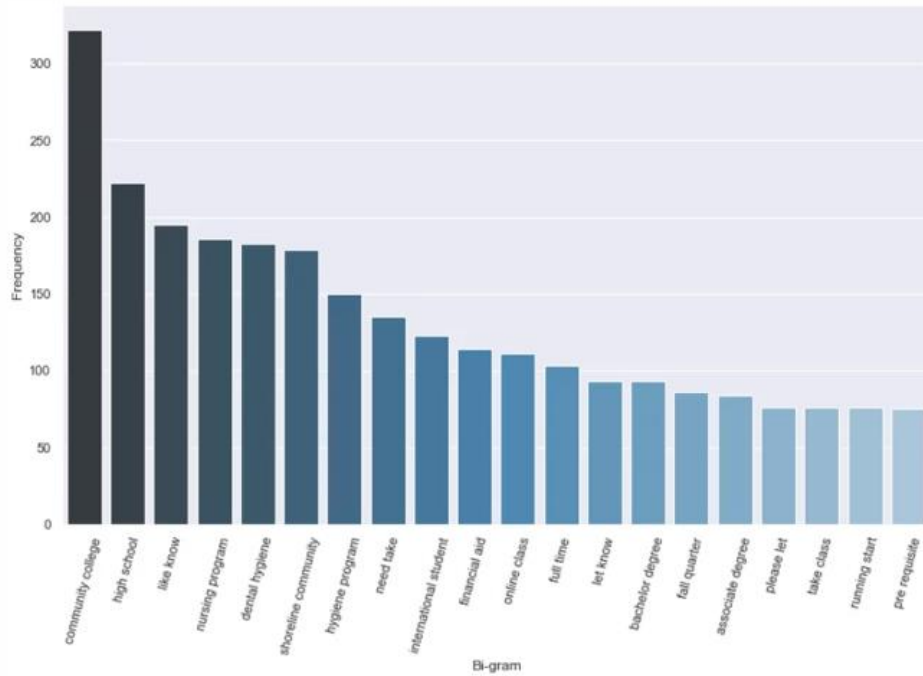


Рисунок 4.11– Графік двадцяти біграм за частотою зустрічей

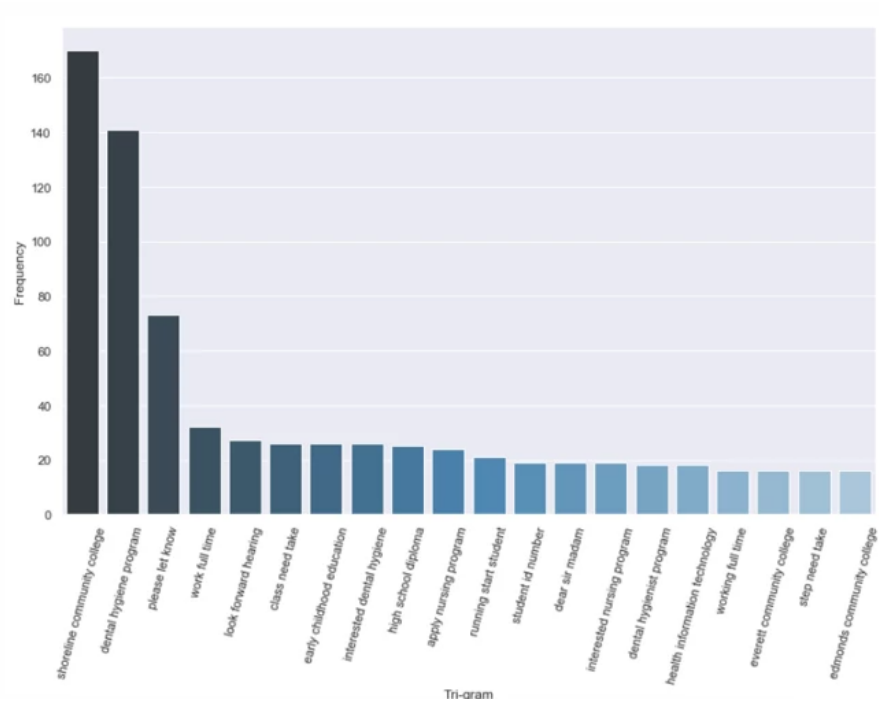


Рисунок 4.12 – Графік двадцяти триграм за частотою зустрічей

Лістинг 4.7 – Програмний код знаходження n найчастіше використуваних біграм

```

from sklearn.feature_extraction.text import TfidfTransformer
tfidf_transformer=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer.fit(X)
feature_names=cv.get_feature_names()
doc=corpus[ds_count-1]
tf_idf_vector=tfidf_transformer.transform(cv.transform([doc]))
from scipy.sparse import coo_matrix
def sort_coo(coo_matrix):
    tuples = zip(coo_matrix.col, coo_matrix.data)
    return sorted(tuples, key=lambda x: (x[1], x[0]), reverse=True)
    def extract_topn_from_vector(feature_names, sorted_items,
topn=25):
    sorted_items = sorted_items[:topn]
    score_vals = []
    feature_vals = []
    for idx, score in sorted_items:
    vals.append(round(score, 3))
    feature_vals.append(feature_names[idx])
    results= {}
    for idx in range(len(feature_vals)):
    results[feature_vals[idx]]=score_vals[idx]
    return results
    sorted_items=sort_coo(tf_idf_vector.tocoo())
    keywords=extract_topn_from_vector(feature_names,sorted_items,25)

```

Результат пошуку представлено на рисунку 4.13.

Abstract:
 online net price calculator cost attendance shoreline show figure year old number find
 current overall cost attendance

Keywords:
 cost attendance 0.488
 attendance 0.45
 overall cost 0.26
 calculator 0.26
 cost 0.257
 number find 0.251
 overall 0.216
 price 0.2
 show 0.189
 figure 0.183
 year old 0.162
 old 0.152
 current 0.152
 number 0.149
 find 0.135
 year 0.096
 online 0.093
 shoreline 0.09

Рисунок 4.13 – Список ключових слів

Виділення ключових слів проводилося на виборці «Запит інформації» від коледжу Шорлайн. Скріншот використаної виборки представлено на рисунку 4.14.

id	date	prioritet	question
495996901	2019-04-21 8:38:01	P1	Hello there I'm from Nepal looking for admission in Regional college to get degree. I would like to know what are the require
519163216	2019-07-10 12:58:12	P4	Hello, My name is. Currently, I am a second-year student, Bachelor of Science in nursing at the University of Ottawa. I am pl
447835205	2018-10-19 11:57:30	P3	Could you give me more information on agronomy or agriculture, thank you very much.
428788128	2018-08-01 23:31:43	P4	Dear, I am a Brazilian student and would like to know if the course of English as a second language could only be done in one y
496056002	2019-04-21 19:22:45	P2	Hello my name is and I've recently enrolled to re-pursue a nursing career but it's been a while since I've taken the classes. I
482804389	2019-03-04 10:02:05	P1	I am interested in the post-bac program for academic enhancement. Is this postbac program the same type of program I would rec
487824965	2019-03-22 9:08:32	P2	I'm looking for an ESL program. Can you help me with tuition, start dates, fees, etc. Also, I wanna know I can apply to the colle
452712356	2018-11-07 23:29:07	P4	what is the schedule of ESL classes?
389845058	2018-03-06 17:04:37	P3	1: what is the schedule, how many hours a day, and how many days per week? How long the program will be? 2. How much does the
450268039	2018-10-29 21:35:04	P1	1:how much cost the tuition?
437206900	2018-09-07 11:27:04	P4	1. How do I obtain my SID? 2. How do I sign up for an orientation session?
500316503	2019-04-28 2:03:04	P4	1. How many years do I need apply for the nursing major in Regional community college? 2. How long will it take from application
486432133	2019-03-16 23:56:20	P4	1. I live in North Carolina, is the HIT AAAS program available in my state? 2. What is the average cost of the total program go:
389894993	2018-03-06 21:07:00	P1	1. Is there a list of the prereqs for the medical lab tech program, and would I be able to take some of those classes elsewhere?
462376647	2018-12-13 4:39:33	P2	1. is there are chances that i can work part time? if i can only work in-campus, do you give a chance for me to work off-campus c
395859935	2018-03-26 23:36:00	P3	1. What is the benefit of having this certificate for someone with a Bachelor's degree in Molecular Biology? 2. What is the dif
434328695	2018-08-26 15:15:35	P4	1. What is the schedule for the Adviser offices so I can come in to help sign up for classes? 2. When is the deadline for fall req
545884141	2019-10-14 15:12:39	P2	1.) I would love yo find out any information about transferring credits I already have at another community college to this schoo
436421976	2018-09-04 17:19:48	P2	1.Application process. Specifically into Nursing program. 2. Types of Financial Aid.
436018122	2018-09-03 1:03:09	P1	1.As a senior, will I get credit for the classes I take at Regional Community College, if I complete the coursework? 2.I have prv
562948035	2019-12-10 6:15:29	P1	1.How many years is the 18 hours cert for digital animation in your college. 2.what is the total break down of how much am goin
406293160	2018-05-08 8:48:47	P1	50 + program video editing
452889246	2018-11-08 14:19:43	P3	A couple questions, I am currently accepted (just recently), and I would like to take three courses but am unsure of the process
518274009	2019-07-07 11:23:26	P4	A couple weeks ago I completed 3 certificates but I have yet to receive the certifications. When will they be sent out?
510796596	2019-06-07 12:38:13	P2	A handful of my nursing prerequisites were taken at Washington State University. Although I have my transcripts and grades avail
445100085	2018-10-08 18:19:53	P3	A Renewable Energy Technician based in Lagos, Nigeria. My specialty is Solar Integrations and Installations. I am interested in
389746553	2018-03-06 11:27:17	P3	about applications fees tuition fees also
555882583	2019-11-14 15:57:19	P2	About LPN to RN program
398155352	2018-04-05 9:30:06	P3	Addition for laborties
466818884	2019-01-03 17:10:02	P4	Admin/Exempt - Dean of Student Support & Success Is this or can this be a live-on position?
427720943	2018-07-28 15:49:10	P4	Admission officer, Im planning to apply for my brother but have a few questions regarding English testing. First, my brother h
411042352	2018-05-26 22:56:41	P3	Admission process in MLT after bsc microbiology
470951449	2019-01-19 17:26:53	P1	After my application was accepted, I will be attending this school in the spring. I haven't thought about to pay for this or what
449417206	2018-10-25 19:46:08	P2	Afternoon, I am a potential student who has taken previous courses at Olympic College and am trying to determine what credits Req
556411083	2019-11-16 14:48:29	P2	All three of my kids have taken coursework at SCC. My son tells me that he took some Tai Chi classes that would be great for he
488156462	2019-03-24 4:57:58	P1	Alpha My name is and I am a nonactive student. I am an alumni graduate of the University of Hawaii at Manoa and I am in the n

Рисунок 4.14 – Скріншот використаної виборки

ВИСНОВКИ

Атестаційна робота виконано згідно з методичними вказівками і містить основні розділи :

- аналіз предметної області та постановка задачі;
- дослідження існуючих ІПА що використовують техніки тайм-менеджменту;
- дослідження застосування технологій, методів і засобів машинного навчання в ІПА;
- реалізація інтелектуальної системи тайм-менеджменту.

На сьогоднішній день застосування інтелектуальних персональних помічників для вирішення задач тайм-менеджменту для ефективного вирішення завдань є перспективним напрямком. Головними причинами впровадження ІПА в життєдіяльності з точки зору тайм-менеджменту є: можливість автоматизації завдань та управління робочими процесами, висока оперативність, зменшення часових витрат користувача на створення списку завдань, надійність, оскільки відсутній людський фактор тощо.

Метою атестаційної роботи є розробка інтелектуального персонального помічника в управлінні витратами часу на основі методів машинного навчання.

Для досягнення поставленої мети в роботі:

- наведено загальний огляд та еволюцію ІПА; проведено аналіз ролі штучного інтелекту в розвитку ІПА; проведено аналіз типових характеристик ІПА; розглянуто типи автоматизованих персональних помічників; розглянуто основні необхідні для розробки завдання машинного навчання;
- в рамках дослідження існуючих ІПА розглянуто MAPGEN, PASSAT та LinTo; розглянуто проект CALO, що використовувався в розробці;

- виконано дослідження застосування технологій, методів і засобів машинного навчання в ІПА, а саме: задача розпізнавання мовлення, задача природної обробки мовлення, задача перетворення тексту в мову та дослідження взаємодії програмних агентів ІПА;
- також представлено реалізацію системи з результатами експериментальних досліджень та практичною реалізацією.

Надалі планується покращення системи для реалізації більш складної політики планування виконання заданого набору завдань.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Baber C. Developing interactive speech technology, United States: Taylor & Francis, Inc, 1993. 212p.
2. Рассел С., Норвиг П. Искусственный интеллект: современный подход. Изд 2-е; пер. с англ. Москва: Вильямс, 2007. 1410 с.
3. Fuckner M., Barthes J., Scalabrin E. Using a personal assistant for exploiting service interfaces, in Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design, Hsinchu, Taiwan. 2014. 94p.
4. Jalaliniya S., Pederson T. Designing Wearable Personal Assistants for Surgeons: An Egocentric Approach, IEEE Pervasive Computing, IT University of Copenhagen, 2015. 31p.
5. Hoy, Matthew B. Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. Medical Reference Services Quarterly, 2018. 81–88
6. Krazit, Tom Google finding its voice. URL: <https://www.cnet.com/news/google-finding-its-voice/> (дата звернення 27.10.2020)
7. Epstein, J., Klinkenberg W. From Eliza to Internet: a brief history of computerized assessment. Computers in Human Behavior. 2001. 314p.
8. Апресян Ю. Идеи и методы современной структурной лингвистики, Москва: Просвещение. 1966.301с.
9. Weizenbaum, Joseph Computer power and human reason : from judgment to calculation. San Francisco : W. H. Freeman. 1976. 1853p.
10. Goksel-Canbek N., Mutlu M. On the track of Artificial Intelligence: Learning with Intelligent Personal Assistants. Instambul: International Journal of Human Sciences. 2006. 601p.
11. Geron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow. Sebastopol: O'Reilly Media, Inc. 2017. 718p.

12. Tsujino K., Iizuka S., Nakashima Y., Isoda Y. Speech Recognition and Spoken Language Understanding for Mobile Personal Assistants: A Case Study of "Shabette Concier, in IEEE 14th International Conference on Mobile Data Management (MDM), Milano. 2013.
13. Kincaid R., Pollock G. Nicky: Toward a Virtual Assistant for Test and Measurement Instrument Recommendations, in IEEE 11th International Conference on Semantic Computing (ICSC). 2017.
14. Grujic Z., Kovacic B., Pandzic I. Building Victor-A virtual affective tutor, in 10th International Conference on Telecommunications (ConTEL) 2009.
15. Leimeister J., Collective Intelligence, Business & Information Systems Engineering, 2(4), 2010, pp. 245–248.
16. Tegos S., Demetriadis S., Karakostas A. MentorChat: Introducing a Configurable Conversational Agent as a Tool for Adaptive Online Collaboration Support, in Proceedings of the 2011 Panhellenic Conference on Informatics (PCI), Kastoria, Greece. 2011.
17. Graesser A., Chipman P., Haynes B., Olney A. AutoTutor: An intelligent tutoring system with mixedinitiative dialogue, IEEE Transactions on Education, 48(4), 2005, pp. 612–618.
18. Blokdyk G. Intelligent Personal Assistant A Complete Guide. 5STARCOoks. 2020. 308p.
19. Norris D. Home Automation with Raspberry Pi: Projects Using Google Home, Amazon Echo, and Other Intelligent Personal Assistants, New York: McGraw-Hill Education. 2019. 240p.
20. Myers K. , Tyson. W., Wolverson M. PASSAT: A User-centric Planning Framework. In: Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space. 2002. 10p.

21. Maldague P., Ko A., Page D., Starbird T. APGEN: A Multi-Mission Semi-Automated Planning Tool. In: Proceedings of the 1st International Workshop on Planning and Scheduling for Space, California, 1997.
22. Muscettola N., Nayak P., Williams B., Pell B., Remote Agent: To Boldly Go Where No AI System has Gone Before, Elsevier: Amsterdam, 1998. 43p.
23. Rebai I., Thompson K., Benhamiche S. LinTO Platform: A Smart Open Voice Assistant for Business Environments. In: Proceedings of the 1st International Workshop on Language Technology Platforms, France. 11-16 May 2020. P. 89–95.
24. Kröger J., Lutz O., Raschke P. Privacy and Identity Management. Data for Better Living: AI and Privacy. Switzerland: Springer International Publishing. 2019. 419p.
25. Документація PAL, SRI International. URL: <https://pal.sri.com/> (дата звернення 19.11.2020)ю
26. Документація CALO Project, SRI International. URL: <http://www.ai.sri.com/project/CALO> (дата звернення 19.11.2020).
27. Kamath U., Liu J., Whitaker J. Deep Learning for NLP and Speech Recognition. Switzerland: Springer International Publishing, 2019. 649p.
28. Котельников В. О пропускной способности “эфира” и проволоки в электросвязи. Москва: УФН, 2006, том 176, С. 762–770.
29. Loper E., Bird S., Klein E. Natural Language Processing with Python. Massachusetts: O'Reilly Media, Inc. 2009. 504p.
30. Hunt A., Black A. Unit selection in a concatenative speech synthesis system using a large speech database. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, USA. 15-20 April 2007.
31. H. Zen, K. Tokuda, A. W. Black. Statistical parametric speech synthesis Speech Communication, In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, USA. 2009. P. 1039-1064.

32. Halliday M. On Language and Linguistics. Edinburgh: A & C Black. 2006. 476p.
33. Bratman M., Intention, Plans, and Practical Reason. Cambridge: Harvard University Press, 1987. 208 p.
34. Phung T., Winikoff M., Padgham L. Learning Within the BDI Framework: An Empirical Analysis. In: Proceedings of the 9th Knowledge-Based Intelligent Information and Engineering Systems Conference. Melbourne. 14-16 September 2005.
36. Новоселова Н., Том И. Подход к построению ансамбля классификаторов с использованием генетического алгоритма. Национальной академии наук Беларуси. Минск. 2009.
37. Міщенко Е.С., Дубровська Н.В. Тайм-менеджмент і його роль в ефективному управлінні персоналом організації. Київ: Університет ім. В.І. Вернадського, 2012. 8 с.
38. Cirillo, Francesco. The Pomodoro Technique. Berlin: FC Garage GmbH, 2011. 128 p.
39. Mihalcea R., Tarau P. Bringing Order into Texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Barcelona. 2004.
40. Васильєв О Python на прикладах. Практичний курс. Санкт-Петербург: Наука і техніка, 2016. 432 с.
41. Yashwanth H., Mahendrakar H., Davia S. Automatic Speech recognition Using Audio Visual Cues. In: Proceedings of the India Annual conference. India. 2004. P. 166-169