

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 _____ Науки про дані (Data Science) _____
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Павленко Оксані Станіславівні _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Мультимодальна система підтримки HR-рішень для оцінки кандидатів на основі аналізу особистісних характеристик _____

затверджена наказом університету від 24 листопада 2025 р. № 1057Ст

2. Термін подання студентом роботи до екзаменаційної комісії 16 грудня 2025 р.

3. Вихідні дані до роботи _____ Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів, документації моделей від OpenAI та HuggingFace, документації з мов програмування Python та JavaScript. _____

4. Перелік питань, що потрібно опрацювати в роботі _____

- 1) Аналіз предметної галузі _____
 - 2) Огляд та аналіз існуючих технологій та методів _____
 - 3) Опис програмної реалізації _____
-
- _____
-
- _____
-
- _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	24.11.2025	виконано
2	Аналіз предметної галузі	25.11.2025	виконано
3	Огляд існуючих методів та технологій	26.11.2025	виконано
4	Проектування системи	27.11.2025	виконано
5	Програмна реалізація застосунку	28.11.2025	виконано
6	Написання пояснювальної записки	01.12.2025	виконано
7	Перевірка на академічний плагіат	05.12.2025	виконано
8	Нормоконтроль	06.12.2025	виконано
9	Підготовка презентації та доповіді	10.12.2025	виконано
10	Попередній захист	11.12.2025	виконано
11	Рецензування	12.12.2025	виконано
12	Захист перед ЕК	16.12.2025	

Дата видачі завдання 24 листопада 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Чала Л.Е.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 121 с., 34 рис., 2 табл., 1 дод., 43 джерела.

АНАЛІЗ ОСОБИСТОСТІ, АНАЛІЗ ТЕКСТУ, ГЕНЕРАЦІЯ ТЕКСТУ,
ОБРОБКА ПРИРОДНОЇ МОВИ, РОЗПІЗНАВАННЯ МОВЛЕННЯ,
ТРАНСФОРМЕРИ.

Об'єкт дослідження – проблема підвищення ефективності процесів обробки та систематизації персональних даних (особистісних характеристик) кандидатів на вакантні посади, які здійснюються HR-менеджером, в умовах значного обсягу вхідної інформації.

Предмет дослідження – методи та засоби автоматизованого аналізу текстової інформації в контексті формування комплексного профілю кандидата та надання обґрунтованої підтримки прийняття рішень щодо найму персоналу.

Мета роботи – дослідження методів та алгоритмів автоматичного аналізу тексту, сфокусованих на ідентифікації психологічних та особистісних характеристик, розробка веб-орієнтованого застосунку, призначеного для оптимізації процесів обліку та аналізу даних про кандидатів у сфері управління персоналом.

Методи дослідження – теоретичний аналіз, опрацювання доступної літератури та досліджень на тему аналізу текстів, особистості та розпізнавання мовлення, практичне дослідження методів та технологій для реалізації застосунку.

В ході роботи було проведено теоретичне дослідження предметної галузі, проведено порівняльний аналіз існуючих методів та засобів розпізнавання мовлення, аналізу текстів та особистості, в результаті чого обрано найбільш ефективні моделі. Створено застосунок для аналізу особистісних характеристик кандидатів.

ABSTRACT

Master's thesis contains: 121 pp., 34 fig., 2 tabl., 1 ann., 43 references.

NATURAL LANGUAGE PROCESSING, PERSONALITY ANALYSIS, SPEECH RECOGNITION, TEXT ANALYSIS, TEXT GENERATION, TRANSFORMERS.

The object of the study is the problem of increasing the efficiency of the processes of processing and systematizing personal data (personal characteristics) of candidates for vacant positions, carried out by an HR manager, in conditions of a significant amount of incoming information.

The subject of the study is methods and means of automated analysis of text information in the context of forming a comprehensive candidate profile and providing substantiated support for making decisions regarding personnel recruitment.

The purpose of the work is to study methods and algorithms of automatic text analysis focused on identifying psychological and personal characteristics, developing a web-based application designed to optimize the processes of accounting and analyzing data about candidates in the field of personnel management.

Research methods are theoretical analysis, processing of available literature and research on the topic of text analysis, personality and speech recognition, practical study of methods and technologies for implementing the application.

During the work, a theoretical study of the subject area was conducted, a comparative analysis of existing methods and means of speech recognition, text analysis and personality was conducted, as a result of which the most effective models were selected. An application has been created to analyze the personal characteristics of candidates.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	10
1 Аналіз предметної галузі	11
1.1 Актуальність задачі автоматичного аналізу особистості для підтримки HR-рішень	11
1.2 Текстовий аналіз.....	14
1.3 Порівняння людського та автоматизованого аналізу текстів.....	19
1.4 Автоматичний аналіз особистості	25
1.5 Розпізнавання мовлення	29
1.6 Постановка задачі.....	32
2 Огляд та аналіз існуючих технологій та методів	34
2.1 Огляд підходів до текстового аналізу	34
2.1.1 Word Spotting	34
2.1.2 Manual Rules	35
2.1.3 Text categorization.....	36
2.1.4 Topic modelling	39
2.1.5 Large Language Models	41
2.2 Огляд моделей текстового аналізу	43
2.2.1 Представлення тексту.....	44
2.2.2 Recurrent Neural Network.....	48
2.2.3 Long Short-Term Memory.....	51
2.2.4 Gated Recurrent Unit.....	54
2.2.5 Transformers	55
2.4 Огляд моделей автоматичного аналізу особистості	68
2.4.1 Підходи до аналізу особистості.....	70
2.4.2 Архітектура моделі BERT та її використання в аналізі особистості.....	72

2.4.3 Архітектура моделі RoBERTa та її використання аналізу настрою та тональності.....	77
2.5 Огляд підходів до задачі розпізнавання мовлення.....	80
3 Опис програмної реалізації.....	84
3.1 Використані технології.....	84
3.1.1 Whisper API.....	85
3.1.2 Hugging Face моделі.....	88
3.1.3 MongoDB Cloud.....	93
3.1.4 Мови програмування.....	94
3.2 Опис проекрованої системи.....	96
3.3 Програмна реалізація.....	99
Висновки.....	115
Перелік джерел посилання.....	116
Додаток А Відомість кваліфікаційної роботи.....	121

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

СУБД – системи управління базами даних;

AI – Artificial Intelligence – штучний інтелект;

API – Application Programming Interface – прикладний програмний інтерфейс;

ASR – Automatic Speech Recognition – автоматичне розпізнавання мовлення;

BERT – Bidirectional Encoder Representations from Transformers – двонапрямні ендодерні репрезентації на основі трансформерів;

BoW – Bag of Words – мішок слів;

CBOW – Continuous Bag of Words – безперервний мішок слів;

CTC – Connectionist Temporal Classification – нейросітна темпоральна класифікація;

DNN – Deep Neural Network – глибинна нейронна мережа;

GMM – Gaussian Mixture Model – модель суміші Гаусових розподілів;

GPT – Generative Pre-Training – генеративне попереднє навчання;

GRU – Gated Recurrent Unit – керований рекурентний елемент;

HMM – Hidden Markov Model – прихована модель Маркова;

HR – Human Resources – людські ресурси;

LDA – Latent Dirichlet Allocation – приховане розподілення Діріхле;

LLM – Large Language Models – великі мовні моделі;

LSA – Latent Semantic Analysis – латентний семантичний аналіз;

LSTM – Long Short-Term Memory – довга короткочасна пам'ять;

MBTI – Myers-Briggs Type Indicator – індикатор типів Маєрс-Брігс;

MLM – Masked Language Model – замаскована мовна модель;

NNMF – Non-Negative Matrix Factorization – не негативна матриця факторизації;

NLP – Natural Language Processing – обробка природної мови;

NSP – Next Sentence Prediction – передбачення наступного речення;

OHE – One-Hot Encoding – one-hot кодування;

RNN – Recurrent Neural Network – рекурентна нейронна мережа;

RoBERTa – Robustly Optimized BERT Approach – стійко оптимізований варіант BERT;

TF-IDF – Term Frequency-Inverse Document Frequency – частота терміна-обернена частота документа.

ВСТУП

В наш час, коли технології стрімко розвиваються, віддалена робота стала звичним випадком, а подати заявку на вакансію можна онлайн та цей процес не займає багато часу, кандидатів на одну посаду можуть бути десятки. Якщо через компанію проходить велика кількість кандидатів, а працівників сфери управління персоналом (HR) мало, процес обліку та аналізу кандидатів може стати складним і заплутаним. В результаті якість та швидкість визначення найкращого кандидата на ту чи іншу вакансію може впасти, тому що людині складно тримати в голові особливу інформацію про кожного з безлічі кандидатів. Крім того, підбір кадрів стає дедалі складнішим завданням, оскільки вимагає не лише оцінки професійних навичок кандидатів, але й аналізу їхніх особистісних якостей, емоційної стабільності та комунікативного стилю.

Створення застосунку, який зможе автоматично аналізувати характеристики кандидатів, робити комплексний аналіз та зберігати повний профіль може вирішити цю проблему, та допомогти HR-спеціалістам в обґрунтуванні свого рішення з найму. Такий застосунок дозволить автоматизувати процес обробки резюме та текстів співбесід, значно зменшуючи навантаження на HR-фахівців і скорочуючи час, необхідний для аналізу великої кількості кандидатів. Застосунок також допоможе забезпечити глибшу аналітичну підтримку у формуванні профілю ідеального працівника для конкретної посади шляхом порівняння особистісних рис, емоційний фону та стилю комунікації кандидата з вимогами вакансії.

У результаті впровадження подібної системи знижується ризик помилкових або упереджених оцінок кандидатів, що сприяє формуванню більш збалансованих команд і забезпечить прозорість у процесі найму персоналу. Тому розробка зазначеного застосунку допоможе покращити якість та пришвидшити процес найму кандидатів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Актуальність задачі автоматичного аналізу особистості для підтримки HR-рішень

У сучасному світі, де темпи розвитку технологій та цифровізації зростають надзвичайно швидко, бізнес-процеси у сфері управління персоналом також зазнають суттєвих змін. Традиційні підходи до підбору кадрів, засновані на ручному аналізі резюме та співбесід, поступово втрачають ефективність, адже обсяг інформації, який необхідно обробити рекрутерам, стрімко зростає. Наразі, в середньому на одну вакансію подається 250 заявок, що робить людську оцінку надзвичайно трудомістким і суб'єктивним процесом [1]. Водночас сучасний ринок праці вимагає швидких і обґрунтованих рішень, адже затримка або помилка у відборі працівників може коштувати компанії значних фінансових та репутаційних втрат.

У цьому контексті особливого значення набуває використання інтелектуальних систем, здатних автоматизувати аналіз особистості кандидатів і надавати HR-фахівцям об'єктивні аналітичні висновки. Застосунки такого типу ґрунтуються на методах штучного інтелекту та машинного навчання, що дозволяють глибше зрозуміти поведінкові та психологічні особливості людини на основі її письмових або усних висловлювань. Аналіз тексту резюме, мотиваційних листів чи транскриптів співбесід може виявити ключові риси особистості, рівень емоційної стабільності, відкритість до нового досвіду, сумлінність, товариськість та інші показники, які безпосередньо впливають на успішність у певних професійних ролях.

Традиційні методи психологічної діагностики, хоча й залишаються корисними, мають низку обмежень. Вони вимагають присутності фахівця, тривалого часу на проведення тестування. Натомість автоматизований

аналіз на основі NLP дає змогу отримати подібну інформацію ненав'язливо, а саме шляхом обробки природних даних, які кандидат уже надає в межах рекрутингового процесу. Це значно підвищує зручність, швидкість та достовірність оцінювання. Крім того, розвиток подібних систем сприяє усуненню суб'єктивних упереджень, які часто виникають під час відбору персоналу. Це робить процес підбору більш прозорим і справедливим, що є важливим аспектом для сучасних організацій, орієнтованих на етичні стандарти та інклюзивність.

Сучасні інструменти автоматичного аналізу особистості охоплюють широкий спектр рішень, існують як класичні психолінгвістичні системи, так і комерційні сервіси та глибинні моделі штучного інтелекту. Одним із перших відомих прикладів такого типу був сервіс IBM Watson Personality Insights, який аналізував мовлення користувача та визначав риси особистості за моделлю «Big Five». Хоча він відіграв важливу роль у становленні автоматизованої психологічної аналітики, сервіс був закритий, а його підхід не отримав подальшого розвитку у вигляді спеціалізованих рішень для HR. Інші інструменти, зокрема LIWC, зосереджені на класифікації слів та обчисленні психолінгвістичних характеристик тексту, однак вони не надають повноцінного типологічного профілю особистості і вимагають значного попереднього аналізу та інтерпретації.

Комерційні системи на кшталт Crystal Knows або Traitify використовуються переважно в рекрутингу, але вони працюють або на основі DISC-моделі, або на базі візуальних стимулів, тобто не здійснюють глибинний аналіз текстів чи усного мовлення кандидата. Інструменти IBM Tone Analyzer та подібні сервіси також не оцінюють особистісні риси, обмежуючись визначенням емоційного тону або загальної стилістики тексту. Таким чином, існуючі комерційні рішення не забезпечують комплексної аналітики, яка б поєднувала автоматичне розпізнавання мовлення, емоційний аналіз, визначення особистісних характеристик та порівняння результатів із профілем конкретної вакансії.

Саме через відсутність цілісного рішення, здатного обробляти відео інтерв'ю, автоматично транскрибувати мовлення за допомогою сучасних моделей, визначати риси особистості, емоційний стан, загальну тональність та співставляти результати із заданим профілем вакансії, постає потреба у створенні нового спеціалізованого застосунку. Розробка такого інструменту є актуальною, оскільки жодна з наявних систем не пропонує комплексного підходу, орієнтованого саме на HR-аналітику, і не дозволяє інтегрувати всі згадані технології у єдиний автоматизований процес оцінювання кандидата.

Таким чином, розробка веб-застосунку для аналізу особистості кандидатів має не лише практичну, а й стратегічну актуальність. Вона поєднує досягнення штучного інтелекту, аналітики даних та когнітивної психології, забезпечуючи новий рівень якості HR-рішень. Завдяки таким інструментам компанії можуть формувати більш ефективні команди, підвищувати продуктивність праці, зменшувати плинність кадрів. У ширшому контексті це сприяє розвитку ринку праці загалом, оскільки робить процес взаємодії між роботодавцем і кандидатом більш якісним, швидким і збалансованим.

Втім, головною ідеєю застосування такої системи є саме підтримка HR-рішень, тобто роль людини у відборі кандидатів все ще залишається головною. Фокус робиться на надання фахівцю інструменту, який зможе пришвидшити процес найму, виокремлювати та зберігати інформацію про особистісні характеристики та навички кандидатів, до того часу, поки вона не стане потрібна для прийняття рішення або для презентації та обґрунтування зробленого вибору.

Отже, створення системи аналізу особистості кандидатів на основі штучного інтелекту є відповіддю на актуальні виклики сучасного ринку праці, де цінуються не лише технічні навички, а й особистісна відповідність, емоційна зрілість і здатність ефективно взаємодіяти в команді. Такий застосунок стає не просто інструментом автоматизації HR-процесів, а потужним аналітичним рішенням, що покращить процес найму.

1.2 Текстовий аналіз

Як зазначалося раніше, текстова аналітика включає в себе процес аналізу неструктурованих та напівструктурованих текстових даних для отримання цінних ідей, тенденцій та закономірностей. Зазвичай його використовують у випадках, коли потрібно обробляти великі обсяги текстових даних для отримання аналітичних висновків, але в іншому випадку це було б занадто ресурсомістким та трудомістким для ручного аналізу людьми.

Галузь текстової аналітики має в собі частки обширного обсягу інших галузей інформаційних технологій. На рисунку 1.1 представлено діаграму Венна, яка показує перетин текстового аналізу (або майнінгу) з шістьма пов'язаними галузями.

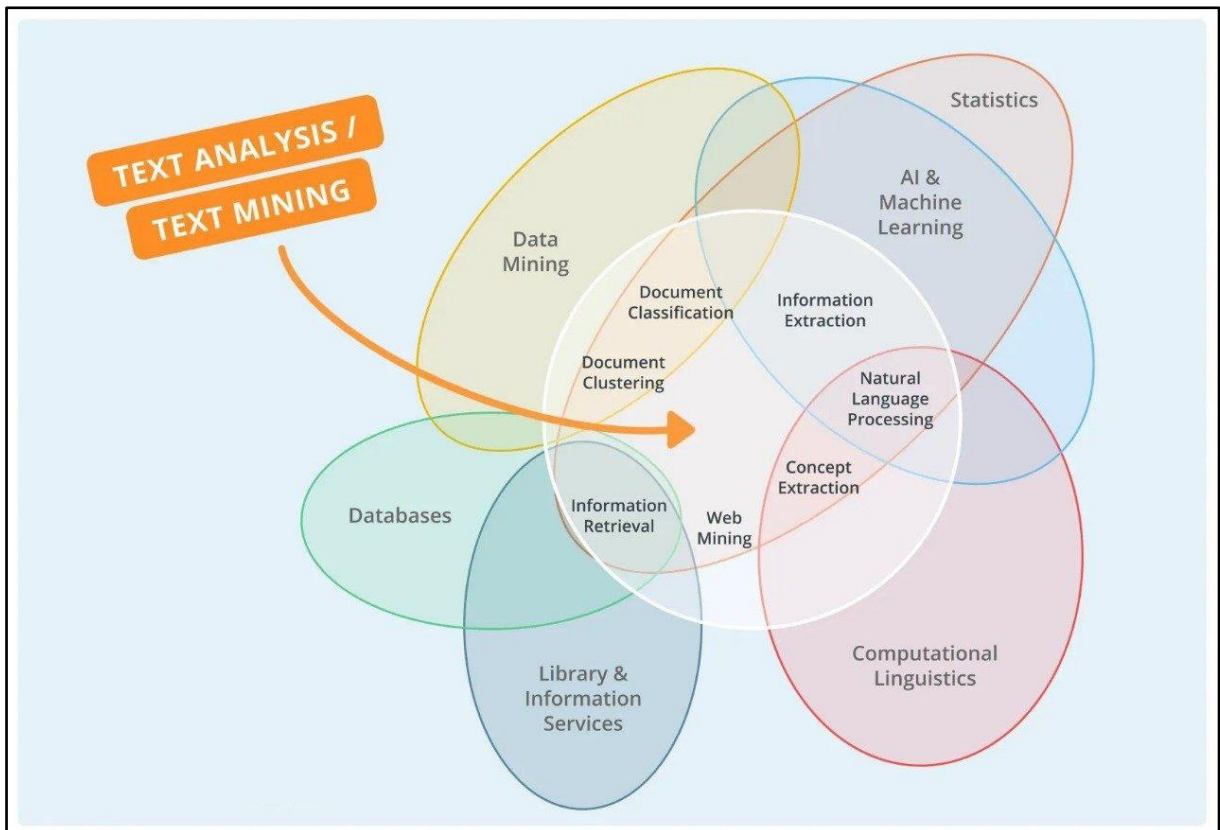


Рисунок 1.1 – Діаграма перетину галузі текстового аналізу з іншими [2]

На діаграмі видно, що текстовий аналіз перетинається з галузями статистики, штучного інтелекту та машинного навчання, комп'ютерної лінгвістики, бібліотечних та інформаційних послуг, баз даних та інтелектуального аналізу даних. У центрі, де всі ці області перетинаються, знаходиться текстовий аналіз або майнінг тексту, які є комплексним процесом, що включають в себе класифікацію та кластеризацію документів, виокремлення інформації, виокремлення понять, веб-майнінг і пошук інформації. Таким чином, діаграма демонструє, що текстовий аналіз є міждисциплінарною сферою, яка включає в себе методи різних галузей для автоматичного отримання знань із неструктурованих текстових даних.

Перед тим як розглядати цю галузь більш детально, важливо визначити відмінності між близькими по назві термінами:

- текстовий аналіз є процесом вивчення та інтерпретації тексту з метою вилучення змістовних висновків. Він може включати як ручні, так і обчислювальні методи;

- майнінгом тексту називають підгалузь текстового аналізу, яка зокрема включає обчислювальні алгоритми для виокремлення та аналізу даних;

- обробка природної мови (NLP) є підгалуззю штучного інтелекту, яка включає розробку алгоритмів для розуміння, інтерпретації та генерації людської мови. Алгоритми NLP використовуються для аналізу тексту, але також мають застосування в перекладі мови та розпізнаванні голосу.

Текстовий аналіз та майнінг тексту часто використовуються як взаємозамінні поняття.

Наразі текстовий аналіз стрімко розвивається, та з кожним роком росте кількість його можливих застосувань у різних галузях, наприклад:

- аналіз настроїв, де текстовий аналіз може допомогти визначити ставлення клієнтів до продуктів або послуг. Це може допомогти визначити сильні та слабкі сторони, а також дозволить розробити стратегії для підвищення задоволеності клієнтів;

– дослідження ринку, де онлайн-відгуки, публікації в соціальних мережах та інші повідомлення клієнтів можна аналізувати за допомогою текстового аналізу. Бізнес може це використовувати, щоб краще зрозуміти, чого бажають клієнти та як вони ставляться до різних товарів і послуг;

– сегментація користувачів, де групи клієнтів зі схожими вимогами чи вподобаннями можна визначити за допомогою текстового аналізу. Це дозволяє компаніям орієнтуватися на певні сегменти клієнтів, завдяки тому, що текстові дані можна групувати за спільними ознаками за допомогою таких методів, як кластеризація. Після отримання потрібної інформації про клієнтів, компанія може налаштувати свої маркетингові або допоміжні ініціативи на основі характеристик кожної групи;

– визначення мови та місцезнаходження, де розташування користувача та мову тексту можна визначити за допомогою методів текстового аналізу. Такі застосування, як переклад, геотаргетинг та рекомендації контенту, можуть покращитися від підвищення точності та релевантності контенту;

– класифікація контенту, де текстовий аналіз можна використовувати для класифікації та категоризації великих обсягів текстових даних. Це може бути корисним для пошуку відповідної інформації, а також для керування базами даних та їх упорядкування;

– виокремлення головного сенсу з текстів, де великі набори даних можна узагальнити за допомогою алгоритмів, що спрощує швидке вилучення важливих концепцій та написання висновків;

– виявлення трендів, де аналізуючи великі обсяги текстових даних, можна знайти закономірності та тенденції в часі. Це може бути корисним для виявлення сезонних тенденцій, нових тем та змін у громадській думці;

– виявлення намірів, де аналіз розмов допомагає визначити мотивацію та мету взаємодії споживача з компанією, таким чином можна покращити обслуговування та досвіду клієнтів, а також забезпечити пошук нових потенційних клієнтів у сфері продуктів чи послуг;

– виявлення шахрайства, де досліджуючи мову та зміст цифрових комунікацій, моделі текстового аналізу можна навчити виявляти шахрайську поведінку;

– управління персоналом, де щоб знайти найкращих кандидатів, текстовий аналіз можна використовувати для вивчення резюме, описів вакансій та інших текстових даних. За його допомогою також можна дослідити відгуки про ефективність роботи та коментарі співробітників.

Таким чином, визначено, що текстовий аналіз та методи обробки природної мови стали невід’ємною частиною сучасного інформаційного простору, та застосовується у багатьох сферах, але разом з цим, присутній простір для створення нових систем для вирішення проблем, які все ще не були досліджені.

Якщо розглядати процес текстового аналізу, то його можна розбити на сім етапів. Діаграму з назвами та порядком етапів представлено на рисунку 1.2.

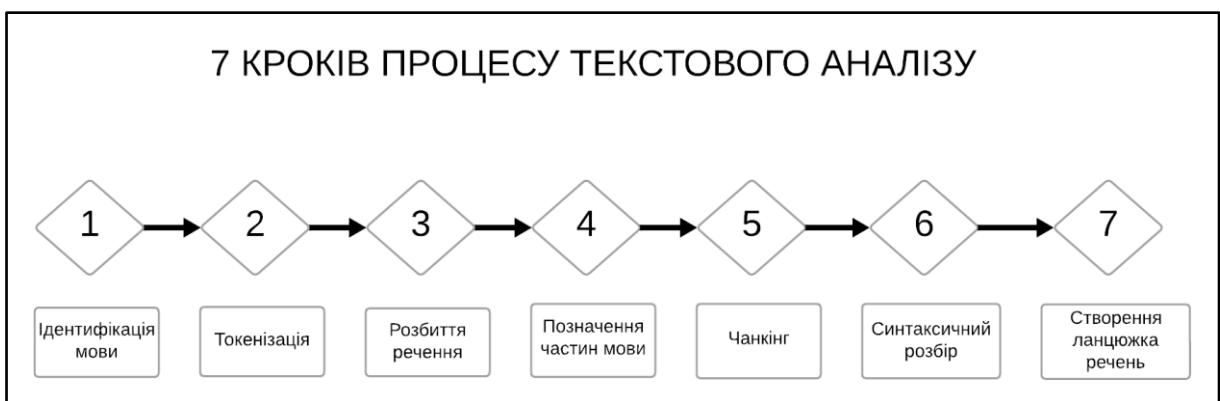


Рисунок 1.2 – Процес текстового аналізу

Як видно на діаграмі, процес починається з ідентифікації мови, в процесі виконання цього кроку, задачею є визначення мови, якою написаний текст, для цього алгоритми аналізують шаблони в тексті, щоб

визначити мову. Цей крок є важливим для подальших етапів обробки, оскільки різні мови можуть мати різні правила та структури.

Наступною виконується токенізація, що представляє з себе розділення тексту на окремі одиниці, часто слова або лексеми. В процесі токенізації текст розбивається на змістовні одиниці, що полегшує його аналіз та обробку, вона включає визначення меж слів та обробку пунктуації.

Наступним етапом є розбиття речення, метою якого є визначення та розділення окремих речень в тексті. В ході цього етапу алгоритми аналізують текст, щоб визначити, де закінчується одне речення і починається інше, так як це є надзвичайно важливим для завдань, які потребують розуміння контексту речень.

Після цього виконується позначення частин мови, головною ціллю є призначення граматичної категорії кожній лексемі в реченні. Під час проходження цього етапу, моделі машинного навчання аналізують контекст і зв'язки між словами, щоб призначити відповідні теги частин мови кожній лексемі.

Наступним етапом є чанкінг, тобто визначити та групування пов'язаних слів разом, часто на основі тегів частин мови. В ході виконання цього етапу виконується розбиття на фрагменти, що допомагає визначити фрази або значущі фрагменти в реченні та є корисним для вилучення інформації про конкретні сутності або зв'язки між словами.

Передостаннім кроком є синтаксичний розбір, метою якого є аналіз граматичної структури речень, щоб зрозуміти зв'язки між словами. В ході етапу синтаксичного розбору створюється синтаксичне дерево, яке представляє граматичну структуру речення та допомагає зрозуміти синтаксичні зв'язки та залежності між словами.

Нарешті останнім кроком йде створення ланцюжка речень, ціллю якого є з'єднання та розуміння зв'язків між кількома реченнями. В ході цього етапу моделі аналізують зміст і контекст різних речень, щоб встановити зв'язки або залежності між ними. Даний крок є вирішальним для

завдань, які потребують ширшого розуміння тексту, таких як підсумовування або аналіз настроїв на рівні документа [3].

Таким чином, для вилучення корисної інформації з текстових даних, текстовий аналіз зазвичай поєднує статистичні методи, моделі машинного навчання та мовні правила. Залежно від застосування та складності завдання текстового аналізу, може бути використано багато методів та інструментів.

1.3 Порівняння людського та автоматизованого аналізу текстів

Аналіз текстів є важливою складовою сучасних досліджень у галузях лінгвістики, психології, соціології, маркетингу та штучного інтелекту. Його основною метою є виокремлення сенсових, емоційних та структурних особливостей тексту з метою отримання нових знань або для прийняття рішень. У сучасній практиці загалом аналіз тексту проводиться або повністю людиною, або комп'ютером, іноді застосовуються гібридні методи.

Людський аналіз тексту – це процес інтерпретації змісту, контексту та емоційного забарвлення тексту, який виконується людиною. Аналітик спирається на власний досвід, культурні знання, інтуїцію та контекст ситуації. Такий підхід характерний для гуманітарних наук і базується на семантичному, синтаксичному, прагматичному та контекстуальному аналізі.

В свою чергу автоматизований аналіз тексту базується на алгоритмах машинного навчання, лінгвістичних правилах і статистичних методах. Метою є обробка великих обсягів текстової інформації для виявлення закономірностей, тем, емоцій або характеристик автора.

Однією з ключових відмінностей між людським і автоматизованим аналізом текстів є масштаб та швидкість обробки інформації. Ці параметри визначають ефективність системи аналізу в умовах сучасного

інформаційного середовища, де обсяги текстових даних зростають експоненційно.

Якщо розглядати цей аспект, то людський аналіз тексту традиційно передбачає послідовну, детальну роботу з обмеженою кількістю текстів. Людина може уважно прочитати, осмислити й інтерпретувати зміст тексту, але цей процес є повільним і трудомістким. Навіть досвідчений аналітик здатний опрацювати лише незначний обсяг інформації за певний проміжок часу. Зазвичай, глибокий та якісний аналіз одного документа, залежно від його складності, контексту та мети дослідження, може тривати від кількох годин до кількох днів. Людина не може ефективно аналізувати великі потоки текстових даних, такі як публікації в соціальних мережах, відгуки користувачів або новинні стрічки, що постійно оновлюються. Через це людський підхід до аналізу є немасштабованим, тобто його складно або практично неможливо адаптувати до великих обсягів інформації без залучення додаткових ресурсів, таких як групи експертів чи спеціалізовані аналітичні центри.

Щодо автоматизованого аналізу текстів, то він навпаки, створений саме для роботи з великими обсягами даних. Сучасні алгоритми обробки природної мови можуть аналізувати мільйони текстових записів за лічені хвилини, виявляючи закономірності, структури, ключові слова, теми, емоції або риси особистості автора. Такі системи дозволяють працювати з потоками даних у реальному часі, що є особливо актуальним для бізнес-аналітики, моніторингу соціальних мереж, маркетингових досліджень чи HR-аналітики. Наприклад, автоматизовані інструменти можуть миттєво визначити загальний емоційний фон тисяч відгуків про компанію або проаналізувати зміст новин для виявлення репутаційних ризиків. Висока швидкість обробки даних досягається завдяки використанню потужних обчислювальних ресурсів, оптимізованих алгоритмів і розподілених обчислень. Ще однією важливою характеристикою автоматизованого аналізу є стабільність швидкодії, так як

система не втомлюється, не знижує продуктивність із часом і може працювати безперервно. Водночас якість результатів залежить від якості вхідних даних і коректності навченої моделі. Якщо модель натренована на неповних або упереджених даних, швидкість обробки не гарантує точності чи коректності висновків. Таким чином, людський аналіз текстів забезпечує повільне, але глибоке розуміння змісту, тоді як автоматизований текстовий аналіз дозволяє опрацювати великі обсяги інформації швидко, масштабовано та систематично.

Одним із найважливіших критеріїв ефективності аналізу текстів є точність виявлення сенсів і глибина розуміння змісту. Ці параметри визначають, наскільки коректно людина або комп'ютер можуть інтерпретувати зміст тексту, виявити його емоційне забарвлення, задум автора, підтекст, стилістичні особливості та взаємозв'язки між частинами повідомлення.

Людський аналіз тексту відзначається високим рівнем семантичного та контекстуального розуміння. Людина сприймає мову не лише як набір слів і граматичних конструкцій, а як складну систему сенсів, пов'язану з культурними, психологічними та соціальними чинниками. Людський мозок здатен інтерпретувати метафори, іронію, сарказм, емоційні натяки та приховані асоціації, що часто недоступні для автоматизованих систем.

Крім того, людина володіє емпатією та культурним розумінням, що дозволяє враховувати соціальні норми, ментальні особливості та комунікативні інтенції автора. Саме тому людський аналіз залишається незамінним у сферах, де важливе глибоке осмислення, таких як літературознавство, психологія, лінгвістичний аналіз та журналістика.

Водночас людський підхід має суттєвий недолік яким є суб'єктивність. Оцінка тексту залежить від особистого досвіду, знань, уподобань і навіть емоційного стану аналітика. Різні експерти можуть інтерпретувати один і той самий текст по-різному, що ускладнює стандартизацію результатів і порівняння висновків.

Автоматизований аналіз текстів, навпаки, прагне до об'єктивності та відтворюваності результатів. Алгоритми працюють за чітко визначеними правилами або навченими моделями, що дозволяє забезпечити стабільність висновків за однакових вхідних даних. Завдяки розвитку технологій NLP, машинного навчання та трансформерних моделей, комп'ютерні системи значно підвищили рівень точності в інтерпретації змісту. Такі моделі здатні враховувати контекст слова в реченні, зв'язки між фразами, синтаксичну структуру та навіть загальний емоційний тон тексту.

Однак навіть найсучасніші моделі мають обмеження. Комп'ютери не володіють глибинним когнітивним розумінням, а отже, не здатні усвідомити підтекст або культурний контекст у повному сенсі. Вони аналізують текст на основі статистичних закономірностей і подібностей у даних, а не на основі справжнього осмислення. Тому у випадках, де важлива інтерпретація сенсів, метафор, мотивів або психологічного стану автора, результати автоматизованого текстового аналізу можуть бути поверховими або хибними. Крім того, точність автоматичних систем безпосередньо залежить від якості навчальних даних. Якщо модель навчалася на упереджених або незбалансованих текстах, вона може відтворювати так само неправильні трактування емоцій або мови, що відрізняється від стандартної.

Отже, якщо підсумовувати, то людський текстовий аналіз має низку переваг. Він забезпечує глибоке розуміння контексту, адже людина здатна враховувати ситуаційні, культурні та емоційні аспекти тексту. Це особливо важливо під час аналізу художніх, публіцистичних або психологічних матеріалів, де значення часто залежить від підтексту, тону чи сарказму. Людина також має здатність інтерпретувати емоції та іронію, адже експерт-лінгвіст або психолог може помічати нюанси, які залишаються недоступними для алгоритмів, зокрема подвійні сенси або натяки. Важливою перевагою є гнучкість мислення, адже людина може адаптувати підхід до різних типів текстів, змінювати критерії оцінювання залежно від цілей дослідження. Крім того, людський аналіз вирізняється етичністю та

усвідомленістю, оскільки дозволяє враховувати моральні, культурні й соціальні норми, які не завжди відображені в алгоритмах. Окремо варто відзначити креативність, так як людина здатна робити висновки, що виходять за межі наявних даних, пропонуючи нові гіпотези або інтерпретації.

Водночас людський підхід має і певні недоліки. Основним обмеженням є невеликий обсяг обробки, тому що людина не може швидко проаналізувати великі масиви текстової інформації, особливо в умовах потокових даних або Big Data. Також важливим чинником є суб'єктивність, адже результати аналізу часто залежать від особистого досвіду, упереджень або настрою аналітика. Недоліком також можна вважати й високу вартість та тривалість процесу, оскільки проведення якісного ручного аналізу потребує значних людських ресурсів, особливо при роботі з великими корпусами текстів. Крім того, існує складність стандартизації, адже різні фахівці можуть по-різному інтерпретувати один і той самий текст, що знижує відтворюваність результатів. Ще одним недоліком є схильність до помилок, оскільки людська втома, неуважність або надмірна складність матеріалу можуть призвести до неточностей під час аналізу.

Водночас автоматизований текстовий аналіз також має свої переваги та недоліки. До основних переваг належить висока швидкість обробки, адже сучасні NLP-системи здатні аналізувати мільйони документів за лічені хвилини, що робить їх незамінними у великих проектах. Важливою перевагою є масштабованість, оскільки алгоритми можуть працювати з необмеженою кількістю даних незалежно від їхнього джерела або формату. Крім того, автоматизований аналіз відзначається об'єктивністю, адже комп'ютери не піддаються емоціям і не мають упереджень, тому результати є більш стабільними та відтворюваними. Системи машинного навчання здатні виявляти приховані закономірності, такі як статистичні зв'язки, патерни або теми, які не завжди помітні людині. Важливою перевагою є також можливість самонавчання, таким способом сучасні моделі з часом

покращують свою ефективність завдяки новим даним і контекстам. Крім того, автоматизований аналіз легко інтегрується з іншими системами, такими як бізнес-процеси, аналітичні панелі чи HR-платформами, що дозволяє швидко оцінювати великі масиви текстових матеріалів.

Однак автоматизований підхід має і певні недоліки. Насамперед це обмежене розуміння контексту, оскільки алгоритми часто не здатні коректно інтерпретувати сарказм, метафори чи культурні натяки. Значною проблемою є залежність від навчальних даних, адже якщо модель тренувалася на упереджених або неповних даних, це безпосередньо впливає на точність результатів. Машини позбавлені інтуїції, тому аналізують тексти формально, без справжнього розуміння їхньої смислової глибини або авторських намірів. Крім того, можливі помилки класифікації такі як хибнопозитивні або хибнонегативні результати, особливо у випадках неоднозначних чи нестандартних текстів. Окремою проблемою залишаються труднощі з обробкою змішаних або неструктурованих текстів, адже алгоритми можуть неправильно аналізувати матеріали, що містять сленг, орфографічні помилки або багатомовні елементи.

Отже кожен з підходів має свої переваги та недоліки. У сучасних умовах ці підходи дедалі частіше поєднуються, в ході чого автоматизовані системи забезпечують масштабність, швидкість і стабільність обробки, дозволяючи аналізувати великі обсяги даних і виявляти приховані закономірності. Водночас людина додає глибину інтерпретації, розуміння контексту, емоцій та культурних особливостей, що виходять за межі можливостей алгоритмів. Такий гібридний підхід поєднує об'єктивність і потужність машин із критичним мисленням і творчістю людини, забезпечуючи найбільш збалансований і точний результат у задачах текстового аналізу.

1.4 Автоматичний аналіз особистості

Особистість людини складається з різноманітних концепцій, такі як ідеї, емоції та переконання, які лежать в основі індивідуальних відмінностей та прогнозують поведінку. Автоматичний аналіз особистості вимагає всебічного розуміння цієї конструкції через її складну та багатовимірну структуру, що є складним навіть за сучасних технологічних проривів [4].

Найбільш помітним і доступним компонентом особистості людини є її стиль мовлення. Тим часом однією з найпоширеніших форм вираження є письмовий текст. Вираження себе через текст зараз стало доступнішим для людей завдяки розвитку інтернет-інфраструктури, такої як соціальні мережі, електронна пошта та інші текстові засоби. Таким чином, з'явилася потреба у визначенні особистості людини використовуючи тільки текст. Як наслідок, використання комп'ютерів для оцінки особистостей людей є важливим і стало предметом досліджень в галузі інформаційних технологій.

Автоматичний аналіз особистості людини, також відомий як автоматичне прогнозування (або сприйняття) особистості, зазвичай здійснюється комп'ютерами. Кількість методів такого аналізу особистості зростає разом з різноманітністю типів даних, доступних для аналізу. Мова, зображення, відео, текст, активність у соціальних мережах, взаємодія з сенсорним екраном тощо є прикладами типів даних, що впливають на різноманітність аналізу особистості. Крім того, кожен з цих типів має підрозділи та підмножини текстових застосувань, таких як публікації та твіти в соціальних мережах, SMS та електронна пошта. Таким чином, використання текстового типу даних та NLP для аналізу підходів до автоматичного аналізу особистості є основним предметом для вивчення в ході цієї роботи.

Для розробки системи автоматичного аналізу особистості краще обирати категоріальну модель, де відповідність кожній характеристиці можна визначити числовим значенням, щоб потім виражати відповідність

за відсотками. Наразі найпопулярнішими моделями, які використовуються для автоматичного аналізу особистості є модель «Big Five» та модель «МВТІ», Обидві психологічні моделі є давно встановленими та широко використовуються у галузях психологічного аналізу.

Щоб визначити основні фактори особистості, кілька незалежних груп дослідників вивчали та факторно аналізували сотні показників рис особистості, що призвело до відкриття та визначення лексичних методів, які породили модель «Big Five». Факторний аналіз списків прикметників, оцінених учасниками, був основним фокусом лексичних підходів. Оскільки більшість цих досліджень визначили п'ять відмінних характеристик, п'ятифакторна модель особистості в результаті отримала відповідну назву [5].

Модель «Big Five» складається з таких п'яти основних рис особистості:

- відкритість (з англ. – Openness) характеризує інтелектуальну допитливість, креативність, уяву та готовність до нових ідей;
- сумлінність (з англ. – Conscientiousness) відображає ступінь організованості, самодисципліни, відповідальності та цілеспрямованості;
- екстраверсія (з англ. – Extraversion) визначає рівень соціальної активності, енергійності та потреби у взаємодії з іншими;
- згодливість (з англ. – Agreeableness) відображає здатність до співпраці, емпатії, довіри та альтруїзму;
- невротизм (з англ. – Neuroticism) характеризує емоційну стабільність і схильність до стресу.

Було визначено, що з психолінгвістичної точки зору кожен із п'яти вимірів вирізняється різними стилями лінгвістичного використання. Наприклад, порівняно з інтровертами, екстраверти говорять більше, голосніше та більше повторюються, менше роблять паузи та вагаються, мають нижче співвідношення типів та лексем, використовують більше слів для позначення позитивних емоцій та використовують менш формальну

мову. Невротики використовують більше конкретних та частотних слів, більше займенників першої особи, більше термінів для позначення негативних емоцій та менше слів для позначення позитивних емоцій. Заперечень та тверджень, що передають негативні емоції, зазвичай уникають люди з високим рівнем сумлінності. Було продемонстровано, що деякі інші мовні показники сумлінності, такі як використання слів-заповнювачів та займенників другої особи, відрізняються залежно від статі людини, що ускладнює автоматичну оцінку особистості. Було показано, що використання ненормативної лексики вказує на неприємність, а уникнення використання минулого часу свідчить про відкритість до нового досвіду.

У сукупності ці п'ять вимірів формують повну картину особистості, дозволяючи зрозуміти, як людина поводить себе, приймає рішення, реагує на виклики та взаємодіє з оточенням. Саме тому модель «Big Five» широко використовується у психологічних дослідженнях, освіті та системах штучного інтелекту для оцінки поведінкових і когнітивних особливостей людини.

МВТІ (Myers-Briggs Type Indicator) було засновано на комплексній теоретичній роботі Карла Юнга. Три характеристики для вимірювання особистості вперше запропонував Юнг, який описав їх як психологічні процеси, які люди люблять використовувати для сприйняття та судження. Четверту пізніше додали Майєрс і Бріггс, і зараз МВТІ є одним з найпопулярніших неклінічних психометричних тестів. Така модель класифікує людей за чотирма дихотоміями, що утворюють 16 можливих типів особистості. Основна ідея полягає в тому, що люди сприймають світ і приймають рішення по-різному, залежно від своїх природних когнітивних характеристик [6].

Модель МВТІ включає в себе такі характеристики:

– екстраверсія (E) та інтроверсія (I), де екстраверти орієнтовані на зовнішній світ, є активними, комунікабельними, та отримують енергію від

взаємодії з іншими. В свою чергу інтроверти, навпаки, зосереджені на внутрішньому світі думок і почуттів, віддають перевагу спокійному середовищу та саморефлексії;

– сенсорика (S) та інтуїція (N), що відображають спосіб сприйняття інформації, де сенсорні типи орієнтуються на факти, деталі, конкретні спостереження та практичний досвід, а інтуїтивні типи звертають увагу на абстрактні ідеї, закономірності, можливості та контексти, дивляться у майбутнє;

– мислення (T) та відчуття (F), що показують, як люди приймають рішення, де мислителі керуються логікою, об'єктивними критеріями, аналізом фактів, а типи, що базуються на відчутті зважають на емоції, гармонію у відносинах, моральні цінності та вплив рішень на інших;

– судження (J) та сприйняття (P), що описують ставлення людини до зовнішнього світу, де люди з перевагою судження структуровані, організовані, планують заздалегідь і прагнуть до визначеності, а сприймаючі типи більш гнучкі, відкриті до змін, діють спонтанно і легко адаптуються до нових обставин.

Якщо порівнювати ці дві моделі, то модель MBTI є зручною для опису індивідуальних особливостей, командної взаємодії чи кар'єрного консультування, адже дозволяє швидко створити узагальнений психологічний профіль людини. Проте, попри свою популярність, MBTI часто критикують за відсутність наукової валідності, адже результати можуть змінюватися з часом, а типологічний підхід занадто спрощує складність людської особистості, крім того, модель не враховує інтенсивність рис, що обмежує її точність у кількісному аналізі.

Натомість модель «Big Five» має міцне емпіричне підґрунтя і є загально визнаним стандартом у сучасній психології. Кожна риса вимірюється за ступенем її вираженості, що дозволяє отримувати більш точні, гнучкі та статистично значущі результати. Цей підхід краще підходить для машинного аналізу текстів, оскільки моделі можуть виявляти

ступінь вираженості кожної риси, а не просто відносити людину до певного типу.

У контексті даного проекту, який спрямований на автоматичний аналіз особистісних характеристик кандидатів за текстовими даними, модель «Big Five» є більш придатною. Вона забезпечує кількісну оцінку, дозволяє глибший психологічний аналіз, краще інтегрується з машинним навчанням і має високу наукову достовірність. Тому саме Big Five виступає оптимальним вибором для реалізації системи аналітики особистості на основі текстових даних.

1.5 Розпізнавання мовлення

Системи розпізнавання мовлення використовують алгоритми для обробки та інтерпретації усної мови з метою перетворення її у текст. Програмне забезпечення аналізує звуковий сигнал, записаний мікрофоном, і переводить його у зрозумілу для людини та комп'ютера текстову форму. Цей процес зазвичай складається з чотирьох основних етапів:

- аналіз аудіосигналу;
- поділ його на окремі сегменти;
- перетворення звуку у цифровий формат, придатний для обробки комп'ютером;
- застосування алгоритмів, які зіставляють отримані дані з найбільш відповідним текстовим варіантом.

Оскільки людська мова дуже різноманітна та залежить від контексту, системи розпізнавання повинні адаптуватися до різних умов. Алгоритми навчаються на великій кількості прикладів мовлення, що охоплюють різні акценти, діалекти, інтонації та стилі вимови. Крім того, програмне забезпечення має вміти відокремлювати голос від фонових шумів, які часто супроводжують аудіосигнал.

Щоб ефективно виконувати свої завдання, системи розпізнавання мовлення використовують два основні типи моделей. Акустичні моделі відображають взаємозв'язок між звуковими сигналами та мовними одиницями, тобто допомагають системі зрозуміти, які звуки відповідають певним фонемам або словам. Мовні моделі, своєю чергою, аналізують послідовність слів і контекст, щоб розрізнити схожі за звучанням слова та формувати найбільш ймовірний текстовий варіант висловлювання. Разом ці моделі забезпечують точніше та природніше перетворення усного мовлення у текст.

На рисунку 1.3 представлено алгоритм роботи процесу розпізнавання мовлення.

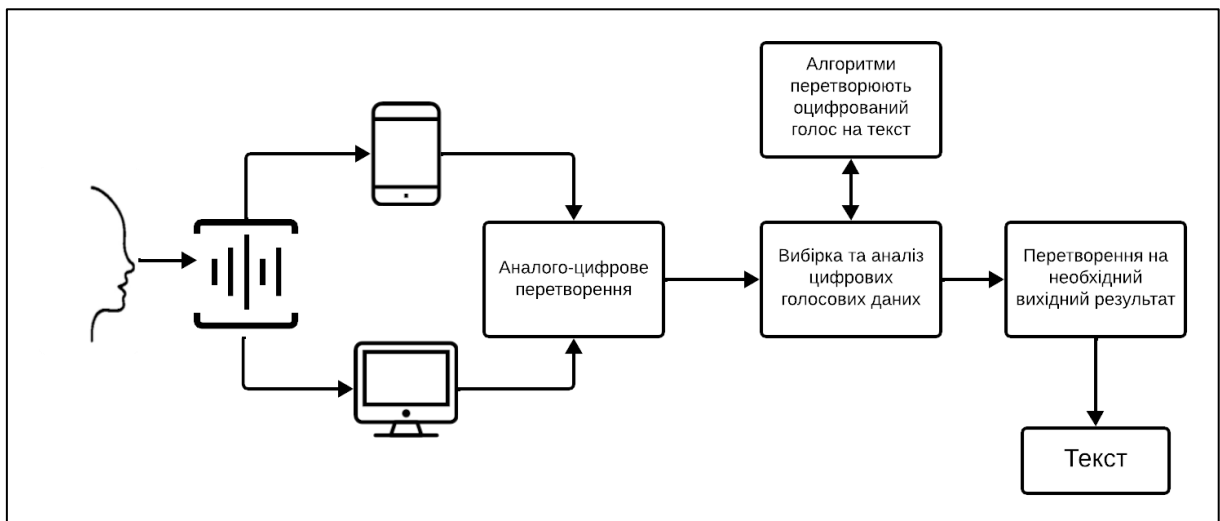


Рисунок 1.3 – Процес розпізнавання мовлення [7]

Програмне забезпечення для розпізнавання мовлення може бути залежним або незалежним від мовця. У системах, що залежать від мовця, досягається вища точність, проте зменшується універсальність. Такі платформи спершу проходять етап навчання, під час якого користувач вимовляє певний набір фраз, щоб система змогла адаптуватися до його індивідуальних особливостей мовлення, таких як тембр, інтонації чи швидкість. Завдяки цьому система з часом покращує точність

розпізнавання, що робить її зручною для таких завдань як, наприклад, диктування чи транскрибування мовлення. В той час системи, що є незалежними від мовця, можуть використовувати будь-які користувачі. Такі системи зіставляють почуті команди або запити з базою універсальних голосових шаблонів, що робить їх гнучкішими та придатними для масового використання. Проте через узагальненість таких моделей точність їх роботи зазвичай нижча.

Дані, з якими працюють системи розпізнавання мовлення, поділяються на три типи залежно від способу введення. Керовані дані представляють собою заздалегідь підготовлені, сценарні фрази або команди, які система розпізнає незалежно від акценту чи вимови. Напівкеровані дані включають запити або команди, сформульовані по-різному, але з однаковим змістом. Такі випадки вимагають складнішого аналізу контенту. Натуральні дані є спонтанним, невимушене мовленням, що є типовим для повсякденних розмов, наприклад під час телефонної бесіди. Їхнє коректне розпізнавання потребує найпотужніших алгоритмів і значних обчислювальних ресурсів [7].

Системи розпізнавання мовлення мають широкий спектр застосувань у різних сферах життя та діяльності людини. У мобільних пристроях вони використовуються для виконання голосових команд, таких як маршрутизація дзвінків, голосовий набір, перетворення мовлення у текст та голосовий пошук. Завдяки цьому користувач може, наприклад, відповісти на повідомлення, не дивлячись на екран. В освіті такі системи активно використовуються для вивчення іноземних мов, де програма аналізує вимову користувача та допомагає її вдосконалити. Вони також мають значення для студентів із порушеннями слуху або нейровідмінністю, адже дають змогу отримувати навчальну інформацію в зручній формі. У сфері обслуговування клієнтів розпізнавання мовлення використовується в автоматизованих голосових асистентах і чат-ботах. Вони здатні приймати запити клієнтів і спрямовувати їх до потрібних ресурсів. Крім того,

технологія застосовується для транскрипції діалогів між клієнтами та операторами з подальшим аналізом їхнього змісту, настрою та типових проблем. Для людей з інвалідністю розпізнавання мовлення відкриває нові можливості комунікації. Наприклад, система може створювати субтитри для осіб із порушеннями слуху або дозволяти керувати комп'ютером голосом тим, хто має обмежені рухові можливості. Також, такі системи відкривають можливості диктування, тобто дають змогу створювати текстові матеріали у реальному часі. Це дозволяє швидко створювати електронне листування чи тексти без ручного введення. Для задачі розпізнавання емоцій технологія аналізує характеристики голосу, щоб визначити емоційний стан мовця. У поєднанні з аналізом тональності це дає змогу оцінити, як ставлення людини.

В даному проекті, модель для розпізнавання мовлення використовувалася для транскрибування аудіо з запису співбесіди, для подальшої відправки отриманого тексту до моделей для особистісного аналізу.

1.6 Постановка задачі

Метою даної кваліфікаційної роботи є дослідження сучасних технологій у сфері аналізу текстів та розробка інтелектуального веб-застосунку для аналізу особистісних характеристик кандидатів з метою підтримки прийняття HR-рішень.

Для досягнення цієї мети в роботі виконано наступні завдання:

- аналіз наукових джерел стосовно теми автоматичного аналізу текстів та особистості;
- аналіз існуючих систем та застосунків, які надають послуги аналізу особистісних характеристик, для визначення запитів користувачів стосовно даної галузі;

– теоретичне дослідження предметної галузі, аналіз існуючих рішень та підходів до автоматизованого аналізу особистості на основі текстових даних, а також визначення оптимальних технологій для реалізації системи;

– аналіз підходів та методів вирішення задачі розпізнавання мовлення;

– розробка серверної частини, яка забезпечує обробку текстових файлів, інтеграцію з моделями для визначення рис особистості та емоцій, а також взаємодію з моделлю для створення узагальненого профілю кандидата;

– реалізація клієнтської частини веб-застосунку з інтуїтивно зрозумілим інтерфейсом, що дозволить користувачу завантажувати резюме, відео співбесіди та профілі вакансій, переглядати результати аналізу й отримувати рекомендації;

– тестування системи на прикладах різних кандидатів і вакансій для оцінки точності, швидкодії та зручності використання застосунку.

На основі проведеного теоретичного аналізу створено застосунок, який дозволить автоматично обробляти резюме та транскрибовані у текст відео співбесід кандидатів, здійснювати аналіз їхніх особистісних рис і емоційного стану за допомогою моделей штучного інтелекту, а також порівнювати отримані результати з профілем ідеального працівника для конкретної вакансії.

У результаті виконання роботи створено функціональний веб-застосунок, що поєднує технології обробки природної мови, машинного навчання та психологічного аналізу, який здатний суттєво підвищити ефективність рекрутингових процесів. Таким чином, застосунок дозволяє автоматично обробляти резюме та транскрибовані у текст відео співбесід кандидатів, здійснює аналіз їхніх особистісних рис і емоційного стану за допомогою моделей штучного інтелекту, а також порівнює отримані результати з профілем ідеального працівника для конкретної вакансії.

2 ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ ТЕХНОЛОГІЙ ТА МЕТОДІВ

2.1 Огляд підходів до текстового аналізу

Текстовий аналіз є однією з ключових галузей обробки природної мови і спрямований на автоматичне отримання сенсової інформації з неструктурованих текстових даних. Існує широкий спектр методів та підходів, що відрізняються рівнем автоматизації, глибиною розуміння тексту та складністю реалізації. Огляд та аналіз існуючих підходів дозволить простежити еволюцію методів текстового аналізу та визначити їхні переваги та обмеження у практичних застосуваннях.

2.1.1 Word Spotting

Підхід за допомогою пошуку слів (з англ. – Word Spotting) часто згадується в неакадемічних джерелах, однак у науковій спільноті обробки природної мови він не є поширеним підходом для аналізу тексту. В академічному контексті термін «word spotting» зазвичай використовується у задачах розпізнавання рукописного тексту, де необхідно визначити, яке слово написано, або в обробці мовлення для виділення ключових слів у мовленні [8]. Попри це, підхід на основі пошуку слів здобув популярність серед практиків, які працюють із невеликими наборами даних, а також у середовищі аналітиків Excel та фахівців із споживчих досліджень. Сутність підходу полягає у припущенні, що наявність певного слова у тексті свідчить про те, що текст стосується теми, пов'язаної з цим словом.

Такий підхід може бути ефективним у випадку невеликих наборів даних, які аналізуються одноразово або кілька разів. Він дозволяє швидко перевіряти результати і забезпечувати високу точність ручної обробки. Проте, у разі обробки великих обсягів даних підхід виявляє суттєві обмеження. Він не підходить для наборів даних з великою кількістю записів,

для автоматизованої обробки без ручного контролю, для візуалізації результатів або для тривалого підтримування даних у узгодженому форматі. Крім того, існує низка інших недоліків цього підходу, які обмежують його застосування у серйозних наукових та промислових проектах.

2.1.2 Manual Rules

Підхід на основі вручну написаних правил (з англ. – Manual Rules) тісно пов'язаний із методом пошуку слів, оскільки обидва базуються на принципі пошуку певних шаблонів у тексті. Водночас шаблони у ручних правилах можуть бути значно складнішими та гнучкішими. Наприклад, у правилах часто використовують регулярні вирази, що дозволяє визначати складні лексичні конструкції, які неможливо реалізувати простими засобами. Багато провайдерів текстової аналітики надають користувачам інтерфейс для створення та управління такими правилами, а деякі компанії пропонують професійні послуги для допомоги у їх розробці. Основною перевагою цього підходу є зрозумілість та простота пояснення правил, таким чином людина може аналізувати, редагувати та налаштовувати правила відповідно до потреб.

Однак створення та підтримка таких правил вимагає значних зусиль. Для спрощення процесу деякі компанії пропонують попередньо підготовлені таксономії, де категорії та підкатегорії вже визначені. Перевагою такого підходу є можливість швидко обробляти великі масиви відгуків та отримувати узагальнений огляд основних категорій. Незважаючи на це, цей підхід має низку недоліків, основними з яких є його обмеженість у гнучкості та масштабованості. По-перше, проблема полісемії ускладнює створення правил, оскільки одне й те саме слово може мати кілька значень залежно від контексту. По-друге, згадування певного слова не завжди означає, що текст стосується саме цієї теми. Крім того, цей підхід не враховує тональність висловлювань, тому не здатний розрізняти

позитивні та негативні оцінки користувачів. Ще одним недоліком є відсутність універсальних таксономій, адже для унікальних продуктів або послуг, зокрема у сфері програмного забезпечення, готові набори правил зазвичай не існують. Нарешті, висока вартість підтримки робить цей метод ресурсномістким, оскільки правила потрібно постійно оновлювати та вдосконалювати фахівцями з лінгвістики, щоб забезпечити точність класифікації текстів.

Таким чином, для великих та динамічних наборів даних підхід на основі вручну написаних правил часто є недостатньо ефективним через високу трудомісткість та обмежену масштабованість.

2.1.3 Text categorization

Категоризація (або класифікація) тексту – це процес автоматичного віднесення текстових документів до попередньо визначених категорій або міток на основі їхнього змісту. Цей підхід є одним із базових завдань у галузях NLP та інформаційного пошуку [9]. Основна мета текстової класифікації полягає в тому, щоб забезпечити ефективне упорядкування, структурування та розуміння текстових даних комп'ютерними системами, що спрощує їх подальше використання, пошук та вилучення релевантної інформації.

Значення цього підходу у сфері інформаційних технологій полягає у здатності систем автоматично керувати, фільтрувати та аналізувати великі обсяги неструктурованих текстових даних [10]. Типовими прикладами застосування текстової класифікації є виявлення спаму, аналіз тональності, тематичне маркування документів, організація текстових баз, автоматичне сортування електронної пошти та виявлення шахрайства. У сучасних дослідженнях також активно розвиваються нові напрями використання, зокрема виявлення мови ворожнечі або класифікація текстів за чутливими темами.

Сьогодні автоматизована класифікація текстів застосовується в широкому спектрі завдань: від індексації веб-сторінок і створення метаданих до розпізнавання жанрів, рекомендацій документів і фільтрації спаму. Завдяки поєднанню класичних алгоритмів і сучасних нейронних мереж цей напрям став ключовим елементом сучасної аналітики текстових даних [11].

На рисунку 2.1 представлено діаграму концепту роботи категоризації текстів [12].

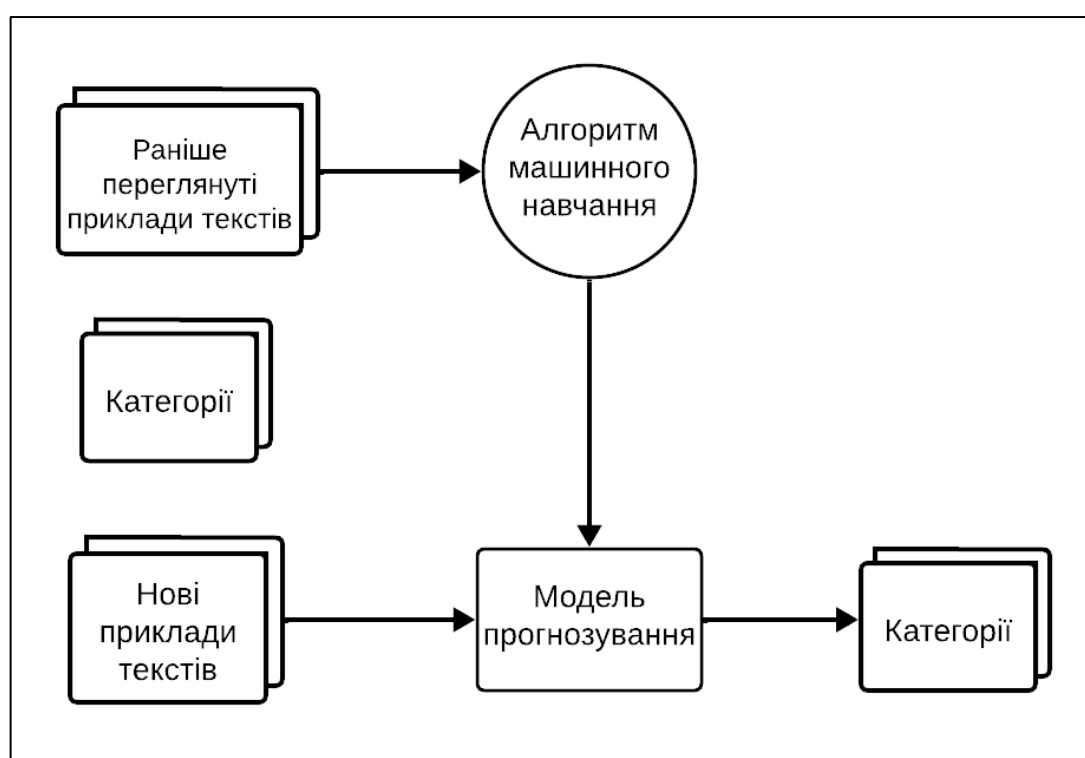


Рисунок 2.1 – Робота категоризації текстів

Текстова класифікація передбачає автоматичне присвоєння текстовим документам попередньо визначених категорій на основі їх змісту та є одним із ключових підходів до обробки великих обсягів неструктурованої текстової інформації у галузях NLP і текстового майнінгу. Процес класифікації зазвичай починається з етапів попередньої обробки тексту, що включають токенізацію, тобто розбиття тексту на окремі слова та розділові

знаки, видалення стоп-слів, що є частовживаними, але малозначущими для аналізу, стемінг або лематизацію, тобто ведення слів до базової форми та перетворення всіх символів у нижній регістр [12].

На етапі представлення тексту у вигляді ознак застосовуються різні підходи. Найпоширенішими є Bag-of-Words, який представляє документ у вигляді вектора частот термів, та TF-IDF, що підвищує вагу слів, часто вживаних у конкретному документі, але рідкісних у решті корпусу. Недоліком Bag-of-Words є втрата контексту та граматичних зв'язків, що частково долається за допомогою векторних представлень слів, таких як Word2Vec або GloVe, які відображають слова у неперервному векторному просторі, зберігаючи семантичну схожість між ними [13].

Серед класичних та найпростіших алгоритмів машинного навчання, що застосовуються для текстової класифікації, найпоширенішими є Naïve Bayes, метод опорних векторів, дерева рішень, метод k-ближчих сусідів та логістична регресія. Для підвищення ефективності часто використовуються методи відбору ознак, зокрема інформаційна вигода, взаємна інформація та підходи на основі кластеризації, що дозволяють обрати найбільш репрезентативні терміни. Завдання текстової класифікації зазвичай реалізується як навчання з учителем, коли модель тренується на розмічених даних для подальшого віднесення нових текстів до певних категорій. У свою чергу, кластеризація є некерованою, що групує тексти за схожістю без використання заздалегідь визначених міток.

Категоризація текстів зазнала значної еволюції, під час якої цей підхід пройшов шлях від традиційних до сучасних архітектур глибинного навчання, що дозволяють автоматично призначати попередньо визначені категорії текстовим документам на основі їх змісту. Згорткові нейронні мережі успішно застосовуються для завдань, таких як виявлення мови ненависті та класифікація тональності, використовуючи векторні представлення слів для вилучення ознак. Такі моделі демонструють кращі результати порівняно з класичними алгоритмами машинного навчання.

Рекурентні нейронні мережі обробляють послідовності тексту елемент за елементом, зберігаючи прихований стан, що містить інформацію про попередні входи, і використовують її для класифікації нових даних з урахуванням контексту. Проте класичні рекурентні мережі стикаються з проблемою затухання градієнтів, що спричинило появу вдосконалених рекурентних архітектур таких як Long Short-Term Memory та Gated Recurrent Unit, які ефективно долають це обмеження. Гібридні моделі, що поєднують згорткові та рекурентні нейронні мережі, застосовуються для крос-доменної категоризації текстів. Моделі на основі трансформерної архітектури, зокрема BERT, використовують контекстні векторні представлення для ефективного вилучення ознак і досягли високих результатів у завданнях класифікації текстів.

Таким чином, підхід категоризації текстів є одним із ключових напрямів сучасної обробки природної мови, оскільки дозволяє ефективно структурувати великі обсяги неструктурованої текстової інформації. Завдяки розвитку методів машинного та глибинного навчання, цей підхід став високоточним інструментом для автоматичного визначення тем, емоцій чи змістових категорій текстів. У межах даної роботи використання категоризації текстів є цілком доречним, адже вона забезпечує автоматичну обробку текстових даних кандидатів, та дає змогу визначити емоційні й поведінкові характеристики особистості.

2.1.4 Topic modelling

Моделювання тем (з англ. – Topic Modelling) – це підхід до автоматичного виявлення прихованих структур сенсів у великих обсягах текстів. Його мета полягає у виявленні основних тем, що описують зміст колекції документів, без необхідності попереднього маркування даних. На відміну від категоризації текстів, де класи заздалегідь визначені, моделювання тем є некерованим методом аналізу [14].

Найпоширенішими алгоритмами моделювання тем, що зробили значний внесок у різні напрями текстового аналізу, є Latent Semantic Analysis, Non-Negative Matrix Factorization, та Latent Dirichlet Allocation. Ці методи широко застосовуються в аналізі текстових даних у різних галузях таких як інформаційний пошук та інтелектуальні освітні системи.

Latent Semantic Analysis (LSA) є алгебраїчним методом, що ґрунтується на сингулярному розкладанні. Його суть полягає у відображенні документів у багатовимірний семантичний простір, де слова та тексти з подібним змістом розташовуються ближче одне до одного. Теоретичною основою LSA є дистрибутивна гіпотеза, яка стверджує, що слова зі схожим значенням мають подібний контекст використання. Усі семантичні зв'язки між текстами виводяться безпосередньо з аналізованого корпусу. Метод LSA знаходить застосування в багатьох завданнях текстового майнінгу, зокрема в автоматичному оцінюванні есе, інтелектуальних навчальних системах, інформаційному пошуку, аналізі соціальних мереж та автоматичному узагальненні текстів.

Non-Negative Matrix Factorization (NNMF) є сучасною технікою зменшення розмірності, що розв'язує проблему появи від'ємних значень у даних шляхом накладання обмежень невід'ємності на модель. Основна ідея NNMF полягає в розкладанні невід'ємної матриці на два невід'ємні фактори. Такий розклад дозволяє представити дані у вигляді сукупності невід'ємних векторів, що є більш інтерпретованим у практичних задачах, де від'ємні значення не мають фізичного сенсу. Метод також підтримує обмеження розрідженості, що робить його ефективним для виявлення найбільш суттєвих ознак у даних. NNMF знайшов широке застосування у текстовому майнінгу для автоматичного узагальнення текстів, де він демонструє кращі результати порівняно з традиційними методами кластеризації. Завдяки поєднанню математичної ефективності та високої інтерпретованості результатів NNMF є потужним інструментом сучасного

аналізу даних і придатним для вирішення складних завдань, пов'язаних з аналізом текстової інформації.

Latent Dirichlet Allocation (LDA) є одним із найпопулярніших методів моделювання тем у текстовому майнінгу та NLP. LDA є ймовірнісною моделлю, яка дозволяє представити кожен документ як комбінацію декількох прихованих тем, в той час як кожен тему представлено як розподіл слів, що її характеризують. Ідея моделі полягає в тому, що документи складаються з набору тем у певних пропорціях, а слова документа генеруються на основі цих тем. Теоретично LDA спирається на Діріхле розподіли для моделювання ймовірностей тем у документах та ймовірностей слів у темах, що дозволяє отримати інтерпретоване представлення тематичної структури текстового корпусу. На відміну від PLSA, LDA забезпечує повну ймовірнісну генеративну модель, що дозволяє краще узагальнювати інформацію на нові документи та уникати перенавчання [15].

Отже, підхід моделювання тем є потужним інструментом для автоматичного виявлення прихованих тематичних структур у великих текстових корпусах, що дозволяє групувати документи за змістом без попередньо визначених категорій. Проте для даного проекту моделювання тем не підходило, оскільки основна мета полягала у класифікації текстів кандидатів за емоційними та поведінковими характеристиками та зіставленні їх з профілем вакансії, для чого більш доречними є підходи на основі текстової категоризації та трансформерів.

2.1.5 Large Language Models

Великі мовні моделі (LLM) відносяться до категорії моделей глибинного навчання. Вони навчені на величезних обсягах даних, що дозволяє їм розуміти та генерувати природну мову та інші види матеріалів для виконання різноманітних завдань. LLM базуються на своєрідній

архітектурі нейронної мережі, відомій як трансформер, яка особливо добре справляється з керуванням послідовностями слів та ідентифікацією текстових шаблонів. LLM суттєво змінили підхід до взаємодії з цифровими системами, адже вперше дозволили штучному інтелекту працювати з людською мовою у тому вигляді, у якому вона природно використовується. Це означає, що користувач може спілкуватися з машиною звичними словами й реченнями, а не через чітко задані команди чи ключові терміни. Тоді як традиційні алгоритми працювали переважно за принципом пошуку збігів між ключовими словами, LLM здатні вловлювати зміст висловлювання, приховані сенсові зв'язки, контекст і логіку міркувань. Завдяки такій гнучкості вони можуть застосовуватися у найрізноманітніших сценаріях, що потребують глибокого розуміння тексту. Крім того, так як вони наділені агентними можливостями, ці моделі здатні певною мірою самостійно виконувати завдання, які раніше вимагали участі людини, що розширює потенціал автоматизації у багатьох галузях.

Великі мовні моделі на початковому етапі проходять навчання з використанням самоуваги. У цьому підході для побудови навчальних сигналів використовуються нерозмічені дані, але процес оптимізації все ж орієнтується на певне приховане еталонне значення. На відміну від класичного навчання з учителем, де для кожного прикладу задано правильну відповідь, у даному випадку система сама формує завдання так, щоб потрібний правильний результат можна було відновити без ручного маркування даних. Таким чином модель самостійно виявляє закономірності, структуру та взаємозв'язки в текстах, навчаючись прогнозувати приховані частини даних чи відновлювати пропущену інформацію, що створює основу для подальшого спеціалізованого донавчання [16].

Великі мовні моделі суттєво змінюють процеси виконання задач та демонструють високу універсальність у широкому спектрі задач різних галузей. Вони здатні виконувати генерацію текстів у таких задачах як написання листів, публікацій, та інших матеріалів у відповідно запитам.

Дані моделі також ефективні у стисненні інформації, перетворюючи об'ємні статті, новини, звіти чи документацію у короткі, структуровані та стилістично узгоджені тексти. У ролі AI-асистентів такі системи здатні відповідати на запитання та надавати детальну інформацію, працюючи як інтерактивні чат-боти в рамках клієнтських сервісів у реальному часі. У сфері розробки програмного забезпечення LLM підтримують генерацію коду, допомагають у створенні застосунків, виявленні помилок, пошуку вразливостей та навіть можуть перекладати код між різними мовами програмування. Крім цього, LLM використовуються для аналізу тональності, забезпечують якісний автоматичний переклад та демонструють здатність до міркування, зокрема розв'язання математичних задач, планування багатокрокових процесів та пояснення складних понять у доступній формі, що робить дані моделі потужними інструментами для підтримки прийняття рішень та інших застосунків.

Таким чином, після огляду всіх основних підходів до текстового аналізу було визначено, що у межах цього проекту оптимальним підходом буде поєднання можливостей LLM та методів категоризації. Першочергово виконується задача класифікації характеристик кандидата, де LLM забезпечують глибоке контекстне розуміння тексту, інтерпретацію емоцій та поведінкових характеристик. Такий симбіоз гарантує високу точність, адаптивність і прозорість аналізу та є найбільш ефективним рішенням для задачі оцінювання кандидатів на основі текстів.

2.2 Огляд моделей текстового аналізу

Упродовж еволюції галузі обробки природної мови послідовні методи та архітектури трансформерного типу сформувалися як два ключові напрями побудови моделей.

Послідовні моделі оперують текстом як впорядкованою послідовністю токенів. Вони поетапно обробляють слова у часовому

порядку та зберігають інформацію про попередні стани, що дозволяє їм моделювати залежності між елементами тексту. Такі методи мають свої переваги, є ефективними, та широко використовуються для різних задач.

Проте, сучасний етап розвитку NLP визначається появою трансформерів, які є моделями, що базуються на механізмі самоуваги. На відміну від послідовних методів, трансформери дозволяють паралельно опрацювати весь текст і формують контекстні представлення токенів, враховуючи зв'язки між словами незалежно від відстані між ними. Саме ця властивість зробила такі моделі основою найпотужніших мовних моделей сьогодення.

Окреме місце у текстовому аналізі посідають методи представлення тексту. Починаючи від класичних підходів і завершуючи щільними векторними ембеддингами та контекстними представленнями трансформерів, способи кодування тексту визначають здатність моделі коректно відображати семантику, синтаксис та контекст.

2.2.1 Представлення тексту

Варіанти представлення текстів постійно розвивалися, і початкові етапи цього розвитку були зосереджені на створенні простих числових векторів, які могли б бути оброблені традиційними алгоритмами машинного навчання. До таких ранніх підходів належать One-Hot Encoding (ONE), Bag-of-Words (BoW) та TF-IDF. Ці методи відіграли ключову роль у формуванні перших систем автоматичної обробки текстів, проте мають суттєві обмеження, які не дозволяють використовувати їх у складних завданнях, таких як аналіз особистісних характеристик або виявлення психологічних патернів у мовленні.

One-Hot Encoding є найпростішою формою подання тексту, за якої кожне слово перетворюється на вектор, довжина якого дорівнює розміру словника. У цьому векторі всі значення дорівнюють нулю, окрім одного

елемента зі значенням 1, що відповідає позиції слова у словнику. Це дає можливість моделі розрізняти слова як окремі символи, однак не забезпечує жодного відображення значення, контексту чи подібності між словами. Саме тому ONE застосовується лише у найпростіших випадках, такі як, базові класифікаційні задачі, де достатньо фіксації факту появи певного терміна. Через повну відсутність семантичної інформації цей метод не підходить для аналізу особистості, оскільки в таких завданнях модель повинна вміти інтерпретувати зміст, стиль, структуру висловлювання та латентні поведінкові ознаки тексту.

Пізніше було розроблено представлення Bag-of-Words, яке відмовляється від позиційного кодування та формує вектор, що описує частоти появи слів у документі. Такий підхід став основою для багатьох класичних моделей класифікації, тематичного аналізу та пошукових систем. Однак BoW ігнорує порядок слів та усі синтаксичні зв'язки. Текст у такому представленні зводиться до набору незалежних термінів, що позбавляє модель можливості розуміти семантичну структуру речення, стилістичні особливості та контекст. З огляду на те, що поведінкові індикатори часто проявляються у способі формулювання думок, а не лише у наборі використаних слів, BoW також не відповідає вимогам дослідження.

TF-IDF став еволюційним розвитком і значно підвищив здатність моделі розпізнавати важливі слова у документі. Цей метод враховує частоту появи слова в конкретному тексті та його рідкісність у корпусі документів, що дає змогу відкидати стоп-слова та виділяти ключові терміни. TF-IDF широко застосовується в інформаційному пошуку, пошукових системах та базових підходах до моделювання тем. Проте, як і BoW, він лишається статистичним методом і не враховує контекст або багатозначність слів. Важливими залишаються лише частотні характеристики, а не сенсові залежності між словами чи структура висловлювань. Така поверхнева інтерпретація тексту є недостатньою для завдань, у яких потрібно визначати

особистісні риси чи емоційні патерни, адже TF-IDF не дозволяє моделі вловлювати глибші семантичні зв'язки та когнітивні особливості мовлення.

Загалом, ONE, BoW та TF-IDF заклали фундамент для розвитку сучасних методів обробки текстів, але їхня ключова спільна проблема полягає у відсутності контекстуального розуміння. Усі три підходи оперують або ізольованими словами, або статистичними властивостями їхньої появи, що робить їх придатними для технічних задач, проте повністю недостатніми для аналізу особистості.

Словникові вбудовування (з англ. – word embeddings) стали ключовим елементом сучасних методів опрацювання природної мови та широко застосовуються у різних задачах NLP. Це щільні вектори, які відображають слова у вигляді числових значень у багатовимірному просторі та дозволяють моделі фіксувати семантичну та реляційну інформацію. Інакше кажучи, вбудовування перетворюють слова на компактні вектори дійсних чисел, які можуть бути ефективно опрацьовані алгоритмами машинного навчання [17].

Раніше розглянуті методи представлення текстів були здебільшого засновані на правилах або статистичних характеристиках. Натомість нейронні підходи дозволяють автоматично виділяти ознаки, що описують структуру та значення мови, без необхідності ручної розробки. Одними з найвідоміших підходів, які можуть бути корисними в цій роботі є Word2Vec, GloVe, Generative Pre-Training (GPT) та Bidirectional Encoder Representations from Transformers (BERT).

Word2Vec став одним із перших проривів у побудові семантичних представлень. Цей підхід базується на двох простих нейронних архітектурах, таких як Continuous Bag of Words (CBOW) та Skip-gram. У CBOW модель передбачає слово за його контекстом, тоді як Skip-gram намагається передбачити контекст за цільовим словом [18]. У процесі навчання мережа формує компактні вектори, які відображають подібність значень слів на основі спільної появи в корпусі. Word2Vec широко

використовувався у тематичному моделюванні, класифікації текстів, системах пошуку та рекомендацій. Проте він формує статичні, неконтекстні вектори, таким чином, слово має один і той самий вектор незалежно від того, у якому реченні воно використане. Через це Word2Vec не здатен відобразити багатозначність та стилістичні патерни.

GloVe є ще одним класичним методом векторизації, який поєднує локальну статистику спільної появи слів в тексті з глобальною інформацією про частотність термінів у всьому корпусі [19]. GloVe будує матрицю спільної появи слів та оптимізує вектори так, щоб різниця між ними передавала семантичні відношення. Завдяки цьому GloVe формує більш стабільні та інформативні статичні представлення, ніж Word2Vec. Модель активно застосовували у класифікації документів, тематичному моделюванні та інформаційному пошуку до ери трансформерів.

На противагу цим статичним методам BERT працює за принципово іншою парадигмою. Це трансформерна модель, яка виконує двонаправлене контекстуальне кодування тексту. BERT навчається на задачах маскування слів та визначення взаємозв'язку речень, завдяки чому може враховувати інформацію з обох боків контексту одночасно [20]. Цей підхід дає змогу формувати динамічні вбудовування, де той самий термін може мати кардинально різні векторні представлення залежно від того, у якому реченні він використаний. Завдяки цьому BERT став стандартом у задачах класифікації текстів, аналізу тональності та багатьох інших. У контексті аналізу особистості BERT є особливо ефективним, оскільки забезпечує глибоке розуміння структури висловлювань, емоційного забарвлення та стилістичних патернів.

GPT представляє інший клас трансформерних моделей, побудованих на підході автоматичної генерації. Модель навчається передбачати наступне слово в послідовності, використовуючи однобічний контекст. Завдяки масовому масштабуванню та величезних тренувальних корпусів сімейство GPT сформувало універсальні мовні представлення, здатні виконувати

генерацію тексту, класифікацію, узагальнення, аналіз емоцій та багато інших задач без спеціального донавчання [21]. У задачах аналізу особистості GPT корисний тим, що вмiє узагальнювати поведінкові індикатори, формувати інтерпретовані профілі та працювати з довгими текстами з урахуванням контексту на рівні дискурсу, а не лише окремих фраз.

2.2.2 Recurrent Neural Network

Рекурентні нейронні мережі (RNN) є різновидом feed-forward мереж, але з додатковою властивістю, адже вони мають внутрішню пам'ять, що дозволяє враховувати попередні елементи послідовності під час прогнозування наступних. Ця пам'ять виникає завдяки рекурентному механізму, де мережа підтримує прихований стан, який накопичує інформацію про послідовність, що надходить на вхід. На відміну від звичайних feed-forward мереж, які трансформують кожен вхід незалежно, RNN обробляє дані з урахуванням того, що було отримано на попередніх кроках. Тому знання, що було набуто в попередній момент часу, впливає на поточне передбачення моделі. На рисунку 2.2 можна побачити різницю між архітектурами feed-forward мережі та RNN.

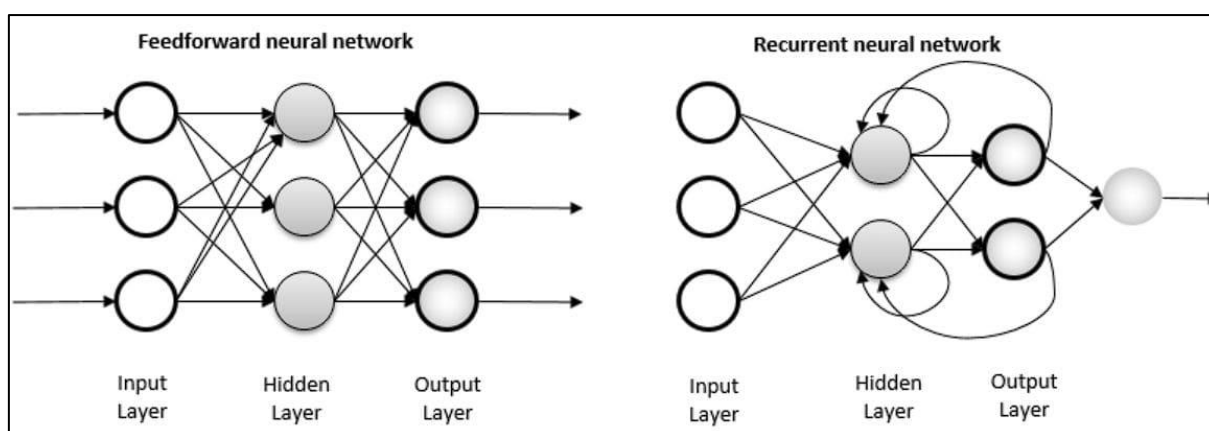


Рисунок 2.2 – Архітектура feed-forward мережі та RNN [22]

Як видно, у стандартній feed-forward мережі або багатошаровому перцептроні дані рухаються лише в одному напрямку, тобто від вхідного шару через приховані шари до вихідного. У такій архітектурі немає зворотних зв'язків, і результат певного шару не впливає на цей самий шар у наступні моменти часу. Кожен вхідний приклад обробляється окремо й не має впливу на наступний, тобто модель не здатна враховувати довготривалі залежності. На відміну від цього, у RNN інформація може циркулювати по замкненому контуру. Під час передбачення модель використовує поточний вхідний сигнал та знання, що було накопичене під час обробки попередніх елементів послідовності. Завдяки цьому RNN може враховувати контекст, що є критично важливим для роботи з часовими рядами або текстом.

RNN обробляє послідовність по елементам, і на кожному кроці t вона зберігає інформацію про попередні стани у вигляді прихованого стану. Саме цей стан дозволяє мережі враховувати контекст (2.1).

$$h_t = \begin{cases} 0, & t = 0 \\ \varphi(h_{t-1}, x_t), & \text{інакше} \end{cases} \quad (2.1)$$

де h_t – прихований стан;

φ – нелінійна функція;

x_t – поточний вхід;

$\varphi(h_{t-1}, x_t)$ – функція оновлення стану.

Зазвичай, приховані стани оновлюються за наступною формулою (2.2).

$$h_t = g(Wx_t + h_{t-1}), \quad (2.2)$$

де g – нелінійна функція;

x_t – вектор поточного входу;

h_{t-1} – прихований стан з попереднього кроку;

W – матриця ваг [23].

Таким чином, при розгортанні RNN мережа відтворюється у вигляді послідовності копій для кожного кроку часу. Після такого розгортання для кожного кроку часу окремо обчислюється функція втрат. Далі мережа виконує зворотне поширення помилки через час, де обчислює градієнти втрат щодо прихованих станів, ваг і зсувів, послідовно поширюючи похибку назад по всій розгорнутій мережі. Архітектуру розгорнутої мережі представлено на рисунку 2.3.

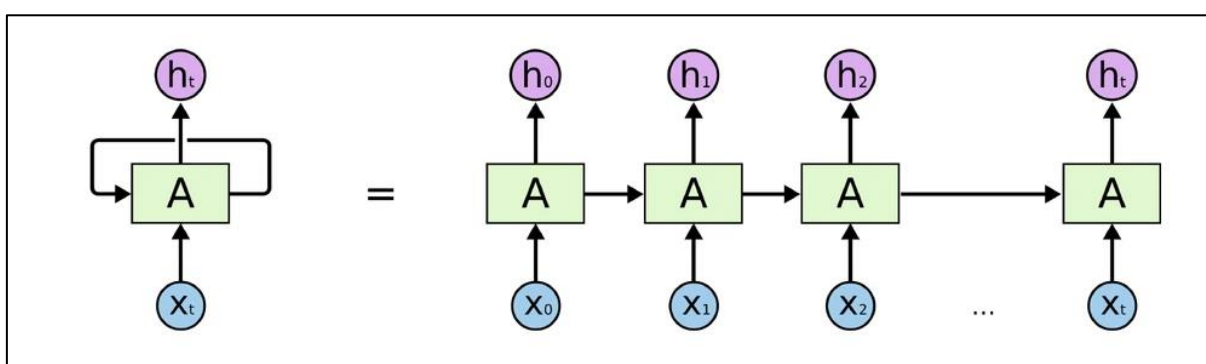


Рисунок 2.3 – Розгорнута архітектура RNN

Однак рекурентні мережі мають низку суттєвих обмежень. Під час навчання часто виникають проблеми вибухаючих або згасаючих градієнтів, через що модель або нестабільно оновлює ваги, або перестає вчитися. Крім того, RNN зберігають контекст лише на короткій дистанції та мають обмежену пам'ять, що ускладнює роботу з довгими послідовностями. Сукупність цих проблем зробила навчання RNN складним і неефективним, що й стало причиною появи архітектури LSTM, яка зберегла базові принципи RNN, але доповнила їх механізмами керованої пам'яті та контролю градієнтів.

2.2.3 Long Short-Term Memory

Мережі Long Short-Term Memory (LSTM) є спеціалізованим різновидом моделей на основі RNN, які було створено для подолання проблем згасання та вибухання градієнтів. Завдяки цьому LSTM значно краще працюють із довгими послідовностями та широко застосовуються в задачах на кшталт аналізу тональності, розпізнавання мовлення та інших форм послідовної обробки тексту. Хоча LSTM зберігають базову архітектуру рекурентних мереж і працюють за схожим принципом, їхня ключова перевага полягає у використанні спеціального механізму керування пам'яттю. На рисунку 2.4 представлено архітектуру LSTM мережі.

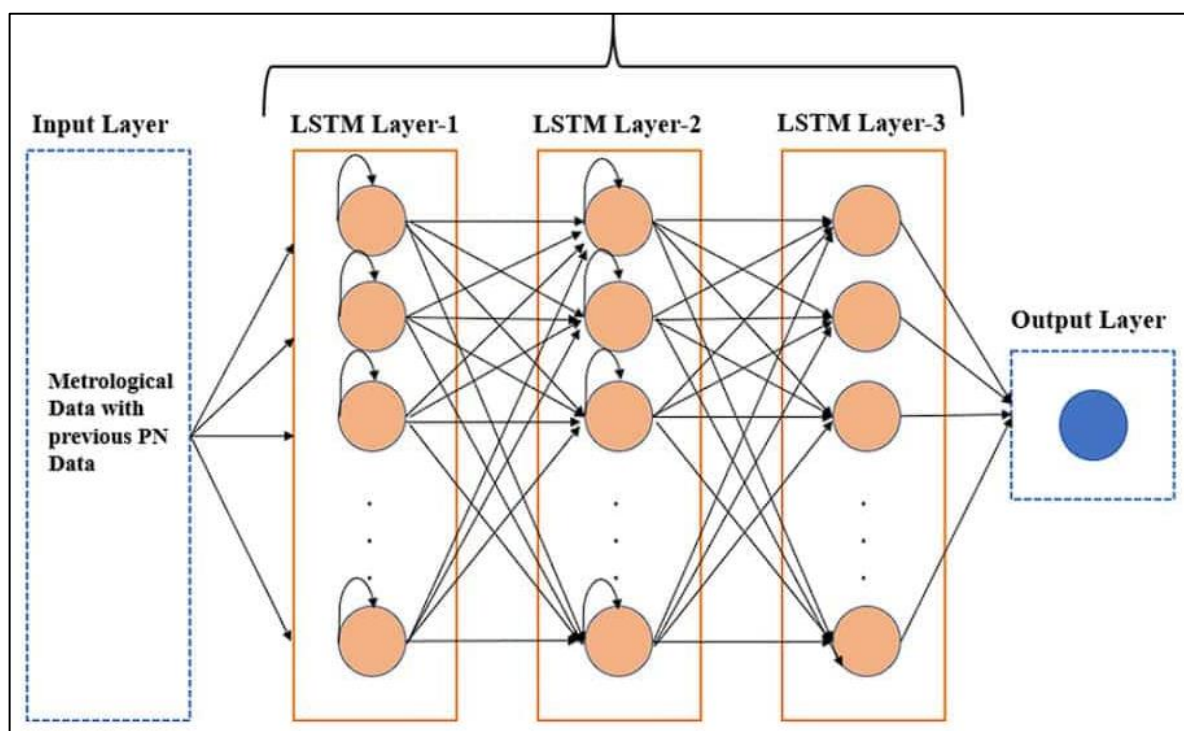


Рисунок 2.4 – Архітектура LSTM [24]

У мережах LSTM ключову роль відіграють два типи внутрішніх стани, а саме стан комірки (cell state) та прихований стан (hidden state), які разом

забезпечують здатність моделі працювати з короткотривалими та довготривалими залежностями. Стан комірки виступає вектором довготривалої пам'яті, який переносить інформацію крізь усю послідовність і змінюється лише під впливом лінійних перетворень, що контролюються гейтами forget, input та output. Завдяки цим механізмам модель може вибірково зберігати важливі дані, оновлювати їх або видаляти непотрібну інформацію, підтримуючи сталість пам'яті протягом тривалого часу. Натомість прихований стан виконує роль короткочасної пам'яті. Він передає інформацію від одного кроку до наступного, аналогічно до рекурентних мереж, і оновлюється на основі попереднього прихованого стану, поточного вхідного значення та оновленого стану комірки.

Діаграма на рисунку 2.5 показує як працює один блок LSTM під час обробки кроку послідовності. Він отримує три основні входи: попередній стан комірки, попередній прихований стан та поточний вхід. У середині блоку послідовно працюють три гейти, кожен із яких керує потоком інформації.

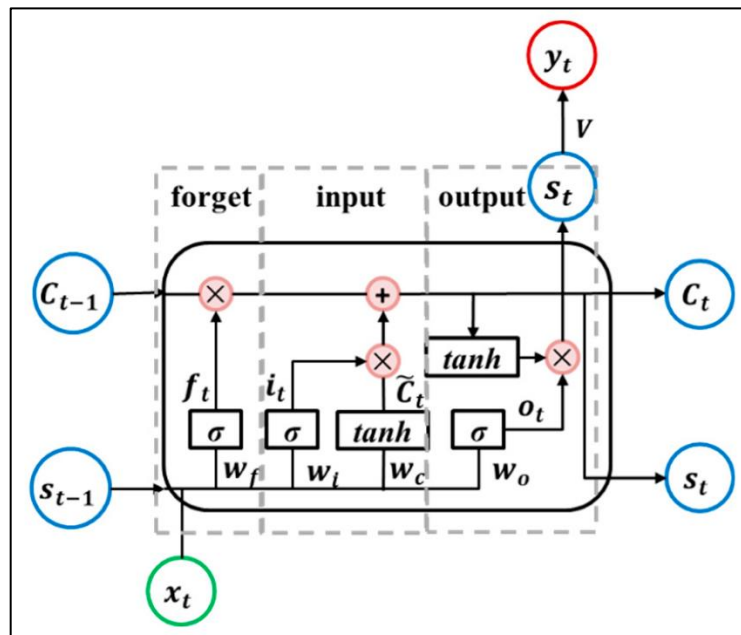


Рисунок 2.5 – Компоненти LSTM [25]

У механізмі LSTM кожен гейт виконує власну функцію, забезпечуючи контроль за тим, яку інформацію зберігати, яку відкидати та яку передавати далі. Робота LSTM починається з forget-гейту, який визначає, які елементи попереднього стану комірки потрібно передати далі, а які вилучити. Для кожного елемента стану комірки гейт формує значення від 0 до 1, де нуль означає повне видалення інформації, а одиниця повне її збереження. Вибір здійснюється за допомогою множення кожного елемента виходу forget-гейту на попередній стан комірки.

Далі активується input-гейт, який відповідає за додавання нових даних до стану комірки. Він складається з самого input-гейту та кандидатного стану. Потім input-гейт, використовуючи сигмоїдну функцію активації, генерує значення від 0 до 1, визначаючи, наскільки важливою є нова інформація. Кандидатний стан, своєю чергою, містить потенційні дані для оновлення комірки. Обидві величини поєднуються, щоб сформувати частину нового стану пам'яті. Значення, що генеруються гейтами, лежать на безперервному проміжку від 0 до 1 завдяки використанню сигмоїдної функції активації, яка стискає будь-яке число до цього діапазону.

Після оновлення стану комірки в роботу вступає output-гейт, який визначає, яка частина стану комірки має бути передана до прихованого стану. Він контролює, яку саме інформацію буде відкрито для подальших обчислень та передано на наступний часовий крок.

Таким чином, якщо узагальнити алгоритм, то спочатку мережа отримує вхідний вектор. На основі цього входу та попередніх станів LSTM обчислює значення трьох гейтів. Потім оновлюється стан комірки шляхом поєднання інформації, що залишилася від попереднього стану, та нових даних, які було вирішено додати. Після цього output-гейт визначає, яка частина цього оновленого стану комірки буде використана для формування прихованого стану. Отриманий прихований стан передається до повнозв'язного шару, який генерує остаточний вихід моделі. Таким чином,

на кожному кроці часу LSTM адаптивно вирішує, що запам'ятати, що змінити та що використати для прогнозування.

2.2.4 Gated Recurrent Unit

Gated Recurrent Unit (GRU) є спрощеним варіантом LSTM, оскільки містить менше гейтів та не використовує окрему комірку пам'яті. Натомість ця мережа лінійно поєднує попередній стан і новий обчислений стан, працюючи подібно до LSTM, але за спрощеною схемою. GRU використовує два основні гейти з назвами *update* та *reset*, що позначені сигмоїдною активацією. Саме вони дають змогу моделі переносити інформацію через багато часових кроків і впливати на майбутні передбачення.

Однак GRU не має окремого механізму, який би контролював, яку частину стану слід приховувати або показувати. На відміну від LSTM, вона завжди відкриває весь стан повністю, без окремого кандидата стану комірки. Вихідний стан GRU так само активується функцією \tanh , але модель не має аналогічного до LSTM компоненту кандидатного стану, що обмежує можливість окремо регулювати приховані стани. На рисунку 2.6 представлено діаграму компонентів GRU.

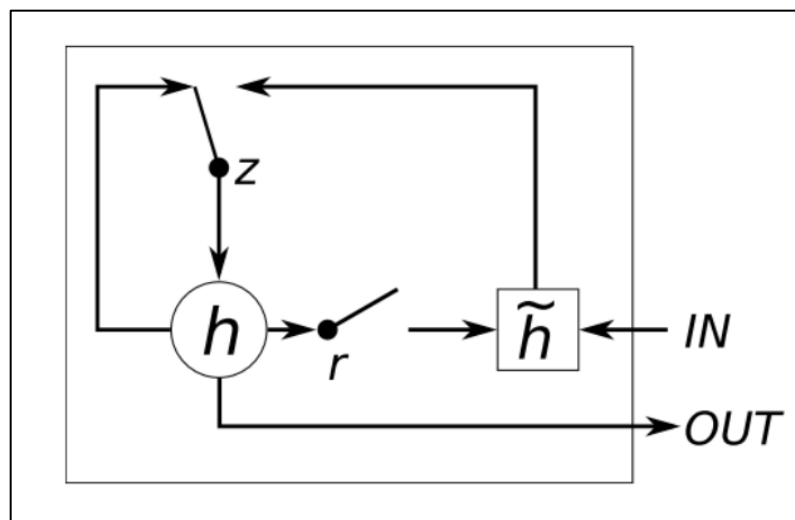


Рисунок 2.6 – Компоненти GRU [26]

На рисунку вище h позначає прихований стан, а \tilde{h} є кандидатним оновленням прихованого стану. Параметри r та z відповідають reset-гейту та update-гейту. На кожному часовому кроці часу GRU отримує попередній прихований стан, а також поточний вхід. Далі мережа модулює вихід за допомогою гіперболічного тангенса, формуючи потенційного кандидата прихованого стану. Потім reset-гейт визначає, наскільки важливо використовувати інформацію попереднього стану при формуванні кандидата.

Фінальне оновлення прихованого стану залежить від значення update-гейту. Якщо значення є близьким до 1, то GRU майже повністю оновлює прихований стан новим кандидатом. Якщо значення близьке до 0, тоді модель зберігає попередній стан, не оновлюючи його. Таким чином, GRU динамічно регулює, що зберігати, що забувати і як оновлювати прихований стан без окремої комірки пам'яті, як у LSTM.

2.2.5 Transformers

Модель трансформеру з'явилася як нове рішення, покликане усунути ключові недоліки рекурентних нейронних мереж. RNN традиційно застосовувалися для роботи з послідовними даними, однак їхня структура робила такі моделі вразливими до проблем зі згасанням або вибуханням градієнтів, а також значно уповільнювала навчання, оскільки обробка виконувалася суто послідовно. При розробці трансформерної моделі підхід було радикально змінено. Таким чином, замість рекурентних з'єднань така модель використовує механізм самоуваги, що дозволяє моделі одночасно враховувати всі елементи послідовності. Завдяки цьому досягається висока ступінь паралелізації, а також можливість ефективно опрацьовувати залежності між віддаленими частинами тексту, що було практично неможливим для класичних RNN.

Модель трансформера складається з кодувальника і декодера, які є двома основними компонентами архітектури. Обидві частини побудовані як набори послідовно розташованих шарів, кожен із яких включає підшари, зокрема механізми самоуваги та передавальної функції нейронної мережі. Повну архітектуру базового трансформера представлено на рисунку 2.7.

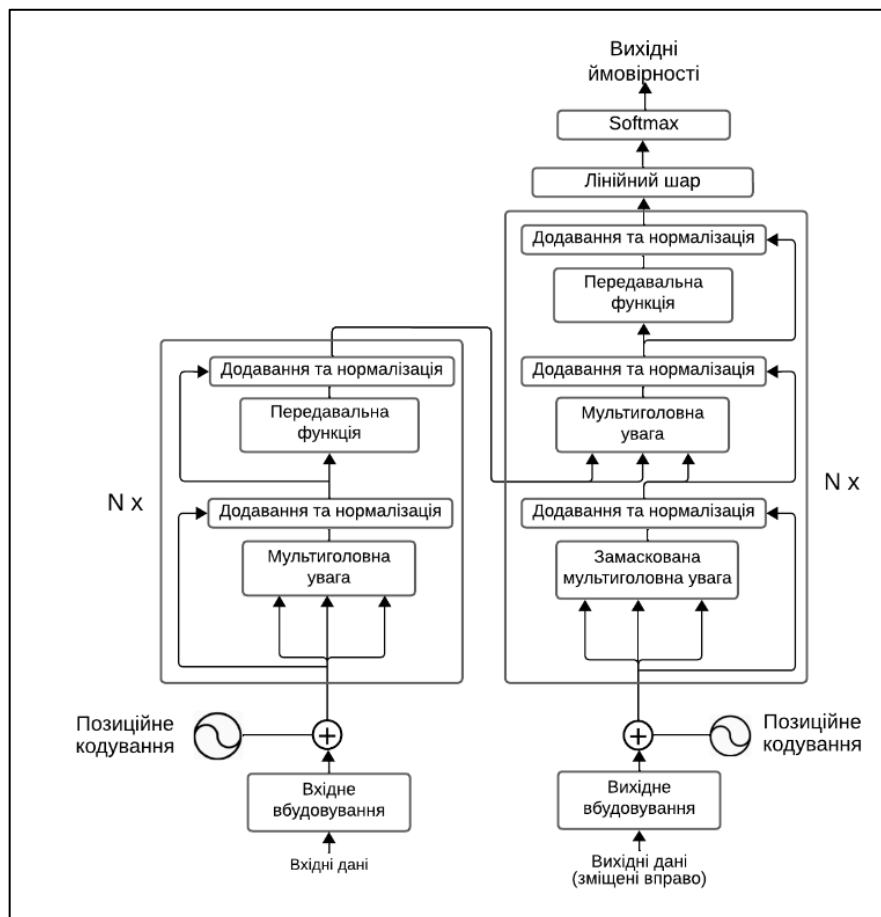


Рисунок 2.7 – Базова модель трансформера [27]

Кодувальник відповідає за обробку вхідної послідовності та формування її внутрішніх представлень. Стандартна реалізація включає шість однакових шарів, кожен із яких складається з двох підшарів, а саме мультиголовний механізм уваги та повнозв'язний передавальний шар, що застосовується незалежно до кожної позиції. Кожен підшар оточується

залишковим з'єднанням та завершується нормалізацією шару. Після виконання підшару вихід обчислюється за формулою 2.3.

$$\text{LayerNorm}(x + \text{Sublayer}(x)), \quad (2.3)$$

де x – вхідний вектор (або матриця), який подається на підшар;

$\text{Sublayer}(x)$ – вихід функції підшара, застосованого до вхідного вектора.

Для підтримки таких залишкових з'єднань усі підшари, а також рівні вбудовування моделі, генерують виходи однакового розміру. У класичному трансформері базова розмірність моделі становить 512. Це означає, що всі вектори, що передаються між шарами, мають фіксовану довжину 512, що забезпечує узгодженість усієї архітектури.

Декодер відповідає за формування вихідної послідовності моделі та використовується в задачах на кшталт генерації тексту. Як і кодувальник, він складається з шести однакових шарів, однак його внутрішня структура трохи складніша.

Кожен шар декодера включає три підшари:

- мультиголовний механізм самоуваги, який обробляє вже згенеровані частини вихідної послідовності;
- позиційно-застосовувану повнозв'язну передавальну мережу, що опрацьовує інформацію незалежно в кожній позиції;
- мультиголовний механізм уваги до виходів кодувальника, який дає змогу моделі враховувати контекст із вхідної послідовності.

Як і в кодувальнику, кожен підшар декодера оточується залишковим з'єднанням і завершується нормалізацією шару, що забезпечує стабільність навчання. Особливість декодера полягає в модифікації підшара самоуваги. Таким чином, він містить маскування, яке забороняє позиції у вихідній послідовності звертатися до наступних позицій. Тобто під час передбачення токена на одній позиції модель може використовувати лише інформацію з позицій менших за неї. Це маскування працює разом із тим, що вихідні

вектори вбудовування зсуваються на одну позицію. Такий механізм гарантує, що модель не бачить майбутнього й генерує послідовність крок за кроком.

Механізм самоуваги дає моделі змогу визначати, які слова в реченні є найбільш значущими для коректного кодування певного токена. Він аналізує взаємозв'язки між усіма словами одночасно, а не послідовно, як це відбувається у рекурентних мережах. Завдяки такій паралельній оцінці контексту модель здатна ефективніше враховувати як близькі, так і віддалені залежності у тексті, що істотно покращує якість роботи з довгими фрагментами та складними мовними структурами. У роботі в якій було представлено трансформерну модель, цю конкретну увагу називають Scaled Dot-Product Attention, діаграму роботи якої представлено на рисунку 2.8.

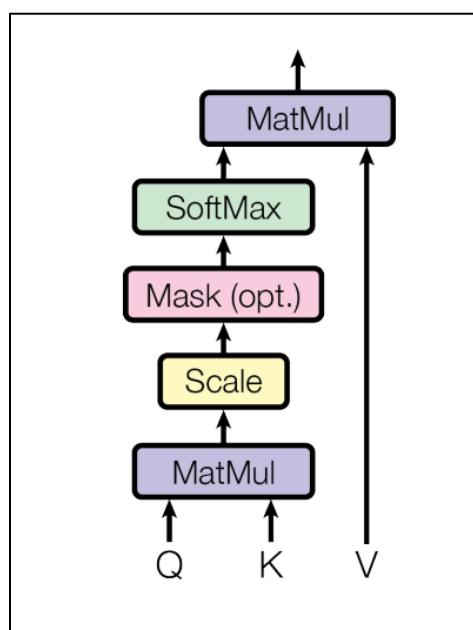


Рисунок 2.8 – Scaled Dot-Product Attention [27]

На першому етапі механізму Scaled Dot-Product Attention кожне слово в послідовності перетворюється на три різні вектори: запит, ключ та значення. Це робиться шляхом множення матриці входу X на відповідні

матриці ваг, що навчаються. Формула 2.4 представляє обчислення матриці запитів.

$$Q = XW_Q, \quad (2.4)$$

де Q – матриця запитів;

X – вхідна матриця, що містить векторні представлення слів;

W_Q – матриця ваг для формування запитів, що навчається.

Формула 2.5 представляє обчислення матриці ключів.

$$K = XW_K, \quad (2.5)$$

де K – матриця ключів;

X – вхідні вектори слів;

W_K – матриця ваг, що використовується для створення ключів.

Формула 2.6 представляє обчислення матриці значень.

$$V = XW_V, \quad (2.6)$$

де V – матриця значень;

X – ті самі вхідні дані;

W_V – матриця ваг, що формує значення.

Ці три представлення дозволяють моделі виконувати механізм уваги, де запит «запитує», ключ визначає релевантність, а значення містить інформацію, що передається на вихід.

Наступним етапом Scaled Dot-Product Attention, модель розраховує оцінки уваги шляхом добутку запитів на ключі, щоб визначити, наскільки кожне слово має увагу до інших слів. Потім ці значення діляться на корінь квадратний з розмірності ключів, що стабілізує градієнти. Отримані оцінки

нормалізуються через функцію softmax. Формула 2.7 представляє механізм обчислення уваги.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2.7)$$

де $\text{Attention}(Q, K, V)$ – фінальна матриця вагомих значень;

QK^T – матричний добуток запитів і транспонованих ключів;

d_k – розмірність векторів ключів;

$\frac{QK^T}{\sqrt{d_k}}$ – масштабовані оцінки уваги;

softmax – функція, яка перетворює оцінки на ймовірнісні ваги;

V – матриця значень, яка зважується відповідно до отриманих ваг.

На останньому етапі механізму вагомі значення V підсумовуються з урахуванням ваг, отриманих після softmax. Результатом є векторне представлення, у якому кожне слово відображає контекст усієї послідовності.

На рисунку 2.9 представлено діаграму роботи мультиголовної уваги.

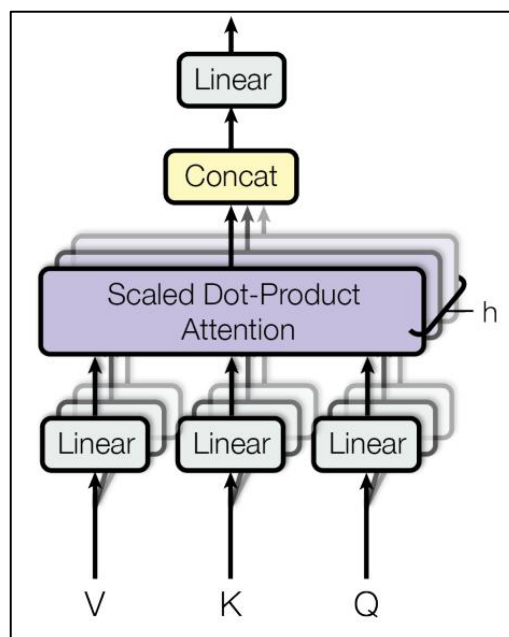


Рисунок 2.9 – Мультиголовна увага [27]

Механізм мультиголовної уваги розширює можливості моделі трансформеру, дозволяючи їй одночасно фокусуватися на різних фрагментах вхідної послідовності. Замість одного механізму уваги модель паралельно запускає декілька незалежних обчислень уваги. Це дає змогу охоплювати різноманітні типи залежностей, виявляти різні патерни та точніше моделювати складні взаємозв'язки в даних.

Після обчислення окремих голів, їх результати об'єднуються та пропускаються через лінійне перетворення. Саме ця операція формує результат мультиголовної уваги (2.8).

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) W_O, \quad (2.8)$$

де $\text{MultiHead}(Q, K, V)$ – фінальний вихід механізму мультиголовної уваги;

$\text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)$ – операція об'єднання результатів кожної голови уваги у спільну матрицю;

head_i – результат роботи i -тої голови attention;

W_O – матриця ваг вихідного лінійного перетворення після конкатенації.

Формула 2.9 представляє обчислення для окремої голови уваги.

$$\text{head}_i = \text{Attention}(QW_{Q_i}, KW_{K_i}, VW_{V_i}), \quad (2.9)$$

де head_i – вихід i -тої голови;

$\text{Attention}(\dots)$ – механізм Scaled Dot-Product Attention;

QW_{Q_i} – проекція матриці запитів за допомогою окремої матриці ваг для голови i ;

KW_{K_i} – проекція матриці ключів для голови i ;

VW_{V_i} – проекція матриці значень для голови i .

2.3 Порівняльний аналіз розглянутих моделей у контексті задачі аналізу особистості

Після того, як було детально розглянуто архітектури моделей, принципи їх роботи та особливості представлення тексту, необхідно визначити, яка з них найбільш придатна для розв'язання поставленої задачі особистісного та емоційного аналізу. Для цього доцільно провести порівняльний аналіз, що дозволить зважити всі переваги та недоліки кожного підходу, оцінити їхню ефективність у контексті аналізу особистості та визначити їх відповідність вимогам проекту. Такий аналіз є ключовим етапом, оскільки саме він дає змогу обґрунтовано обрати модель, здатну забезпечити високу точність, стабільність та інтерпретованість результатів у складних задачах психологічного профілювання.

Спочатку розглянемо рекурентні нейронні мережі. RNN мають низку характеристик, що визначають їхню ефективність у задачах обробки природної мови, проте їхня придатність суттєво відрізняється залежно від типу завдання. До переваг RNN належить те, що вони природно працюють із послідовностями, обробляючи текст крок за кроком і зберігаючи інформацію про попередні стани. Такий підхід дозволяє моделювати локальні часові залежності, що може бути достатньо для простих задач класифікації текстів або базового аналізу настрою. Крім цього, RNN мають відносно низькі вимоги до обчислювальних ресурсів, що робить їх зручними для невеликих корпусів даних, а їхня проста та інтуїтивно зрозуміла архітектура полегшує реалізацію та інтерпретацію основних принципів роботи моделі.

Водночас RNN мають істотні недоліки, які суттєво обмежують їх застосування у складних NLP-задачах. Модель погано зберігає інформацію на довгих послідовностях через проблему згасання градієнта, тому ключові залежності між частинами тексту часто втрачаються. Це особливо критично для аналізу особистості, де психологічні маркери, стилістичні особливості

та поведінкові патерни можуть бути розподілені по всьому документу та потребують узгодженого глобального контексту. RNN також погано уловлюють тон, стиль та інші латентні характеристики автора, оскільки працюють зі статичними векторними представленнями та мають обмежену здатність формувати глибоку семантичну інтерпретацію тексту. Додатковим недоліком є неможливість ефективної паралелізації обчислень, що уповільнює навчання та ускладнює застосування на великих наборах даних.

LSTM є вдосконаленою рекурентною архітектурою, яка була розроблена для подолання ключових обмежень класичних RNN. Однією з головних переваг LSTM є здатність зберігати інформацію на тривалих часових інтервалах завдяки використанню механізмів пам'яті. Це дозволяє ефективно моделювати довгі залежності, що суттєво покращує продуктивність у задачах послідовного аналізу текстів, таких як аналіз настроїв, класифікація або машинний переклад. LSTM також добре працюють на середніх за обсягом корпусах текстів і здатні навчатися навіть тоді, коли у даних присутні нелінійні та складні мовні зв'язки. У контексті аналізу особистості перевагою LSTM є те, що вони краще, ніж прості RNN, уловлюють емоційні патерни та психологічні сигнали, якщо вони розміщені на коротких або середніх текстових інтервалах. Завдяки кращій пам'яті та стабільності градієнтів LSTM можуть відображати певні стилістичні особливості мовлення автора, наприклад схильність до емоційності, використання позитивних або негативних формулювань чи структуру фраз. Це робить їх дещо ефективнішими для завдань попередньої емоційної класифікації або первинного психологічного скринінгу.

Однак, попри свої технічні вдосконалення, LSTM мають низку суттєвих недоліків, які обмежують їх придатність у завданнях глибокого аналізу особистості. По-перше, хоча LSTM краще працюють із довгими залежностями, вони все ж не здатні опрацьовувати глобальний контекст документа так само глибоко, як сучасні трансформери. Психологічні

маркери, що проявляються у тонкій структурі мовлення, приховані семантичні зв'язки або когнітивні характеристики автора, часто втрачаються через лінійність обробки тексту. По-друге, LSTM працюють із фіксованими або тренуваними статичними вбудовуваннями слів, що не дозволяє їм формувати багаті контекстні представлення й точно розрізняти значення одного й того самого слова в різних ситуаціях. Для аналізу особистості, де важливо враховувати дрібні смислові відтінки, це є критичним обмеженням. Хоча LSTM перевершують RNN за стабільністю та точністю, вони все одно істотно поступаються трансформерам у здатності моделювати стиль мовлення, визначати поведінкові патерни та формувати інформативні профілі особистості на основі складних контекстних залежностей.

GRU, як було визначено раніше, є спрощеною та більш ефективною модифікацією LSTM, яка поєднує гнучкість рекурентних мереж із меншою обчислювальною складністю. Завдяки цьому GRU стали популярними в задачах обробки природної мови, де важливими є швидкість навчання та здатність моделювати залежності середньої довжини. Однією з ключових переваг GRU є їхня компактність, адже модель має менше параметрів, ніж LSTM, оскільки використовує лише два типи гейтів. Це дозволяє швидше навчати модель, зменшувати ризик перенавчання й застосовувати GRU в ситуаціях, де ресурси обмежені або необхідно забезпечити роботу в реальному часі. У загальних задачах аналізу текстів GRU демонструють продуктивність, співставну або навіть кращу за LSTM, особливо на невеликих і середніх наборах даних. Вони добре моделюють локальні та середньострокові залежності в тексті, що корисно для таких задач, як класифікація коротких повідомлень, виявлення емоцій або автоматичне завершення фраз. Завдяки стабільності градієнтів, GRU здатні краще, ніж звичайні RNN, запам'ятовувати довготривалі сигнали, що робить їх придатними для невеликих послідовних моделей. Щодо задач аналізу особистості, GRU мають певні переваги. Модель може уловлювати емоційні

патерни, характерні стилістичні структури фраз та базові психологічні ознаки, якщо вони проявляються на локальному рівні. Завдяки ефективному механізму оновлення стану GRU іноді здатні краще адаптуватися до текстів, у яких важлива ритмічність чи емоційна насиченість, що робить їх корисними у первинному аналізі настроїв або у задачах оцінки. Водночас здатність GRU відображати глибинні особистісні характеристики залишається обмеженою.

Суттєвим недоліком таких моделей є те, що попри спрощення архітектури, вони все ще працюють у межах рекурентного підходу та мають ті самі фундаментальні обмеження, що й LSTM. Зокрема, GRU не здатні формувати глобальний контекст документа, оскільки обробка послідовності залишається лінійною. У задачах аналізу особистості, де важливо інтегрувати інформацію з усього тексту, така лінійність призводить до втрати ключових психологічних сигналів. Крім того, GRU працюють зі статичними вбудовуваннями, що обмежує їхню здатність розрізняти сенсові відтінки слів у різних контекстах. Це критично важливо для аналізу особистості, де невеликі зміни у формулюваннях можуть суттєво впливати на висновки про риси характеру. Модель також має обмежену здатність розуміти складні синтаксичні конструкції та поведінкові патерни, що знижує її точність у порівнянні з трансформерними архітектурами.

Трансформери стали фундаментом сучасних NLP-систем завдяки своїй здатності обробляти текст у глобальному контексті та формувати надзвичайно інформативні представлення. Їхні архітектурні особливості, зокрема механізм самоуваги, зробили таку архітектуру найефективнішим підходом для вирішення задач текстового аналізу.

Першою ключовою перевагою трансформерів є здатність працювати з контекстом у всій послідовності одночасно, а не послідовно, як це роблять рекурентні моделі. Механізм самоуваги дозволяє моделі оцінювати важливість кожного слова щодо інших слів у тексті незалежно від їхньої позиції. Це дає змогу точно аналізувати складні синтаксичні та семантичні

зв'язки, що суттєво підвищує якість виконання таких завдань, як класифікація текстів, виявлення емоцій, тематичне моделювання чи пошук інформації. Крім того, трансформери можуть працювати з великими наборами даних і стабільно масштабуються, що робить їх оптимальними для промислових систем NLP. У задачах аналізу особистості трансформери мають ще більш виражені переваги. Контекстні вбудовування, які моделі такі як BERT або GPT формують під час опрацювання тексту, дозволяють вловлювати тонкі нюанси стилю, емоційної виразності, лінгвістичної складності та поведінкових патернів. Завдяки глибокому контекстному аналізу, трансформери здатні враховувати як окремі фрази, так і загальний стиль оповіді, спосіб формулювання думок, риторичні стратегії та когнітивні ознаки автора, тобто все, що має значущість для психологічної оцінки.

Попри очевидні переваги, трансформери мають і кілька важливих недоліків. Перш за все, це висока обчислювальна вартість, адже їхнє навчання потребує значних ресурсів, великих корпусів даних та сучасного апаратного забезпечення. У порівнянні з попередніми моделями вони є набагато важчими як у тренуванні, так і у використанні. Для задач, де доступ до ресурсів обмежений, це може стати суттєвою проблемою. Ще одним аспектом є залежність трансформерів від великих тренувальних корпусів. Модель, що не отримала достатнього різноманіття мовних контекстів на етапі попереднього навчання, може мати упередження або неточності, що можуть впливати на якість аналізу особистості. Незважаючи на це, природа моделей на основі трансформерної архітектури дозволяє доступне використання претренованих готових моделей у своїх проектах.

Після того, як було детально розглянуто основні переваги та недоліки кожної моделі, для більш наочного порівняння було визначено основні критерії, які є важливими для реалізації даного проекту. Ними стали обробка послідовності, що показує принцип роботи моделі з текстом, здатність моделювати довгі залежності, урахування стилю та тону, здатність

уловлювати психологічні індикатори, збереження контексту на великих текстах та придатність для аналізу особистості. Результати такого аналізу представлено у таблиці 2.1.

Таблиця 2.1 – Порівняння розглянутих моделей

Характеристика	RNN	LSTM	GRU	Трансформери
Обробка послідовності	Послідовна, покрокова	Послідовна	Послідовна	Паралельна обробка всього контексту
Здатність моделювати довгі залежності	Низька	Висока	Висока	Дуже висока
Урахування стилю, тону та нюансів викладу	Дуже слабке	Обмежене	Середнє	Високе
Здатність уловлювати психологічні індикатори	Дуже низька	Середня	Середня	Дуже висока
Збереження контексту на великих текстах	Погане	Помірне	Добре	Відмінне
Придатність для аналізу особистості	Непридатна	Обмежено придатна	Частково придатна	Повністю придатна

В результаті можна сказати, що в цілому трансформери є найбільш сучасним і потужним інструментом для аналізу текстів, а особливо для завдань визначення особистісних характеристик. Вони демонструють найвищу здатність до глибокого контекстного розуміння, моделювання поведінкових сигналів та побудови психолінгвістичних профілів. Недоліки такої архітектури пов'язані здебільшого з ресурсами та складністю, а не з якістю аналізу, тому саме вони є оптимальним вибором для розробки системи, яка ґрунтується на задачі аналізу особистості.

2.4 Огляд моделей автоматичного аналізу особистості

Перед початком огляду моделей, які найкраще підходять для задачі аналізу особистості, було визначено психологічно модель, на основі результатів якої буде будуватися повноцінний психологічний профіль. Раніше було розглянуто дві такі моделі, а саме MBTI та «Big Five». Після короткого дослідження, було вирішено використовувати в цьому проєкті саме модель «Big Five». Як було зазначено раніше, в основі даної моделі стоїть п'ять рис, які визначають особистість людини, кожна людина має високий та низький показник кожної риси, який впливає на її загальну характеристику.

Якщо розглядати кожну рису окремо, то людина з низьким показником відкритості зазвичай не полюбляє зміни та не отримує задоволення від нових речей, дотримується звичних, конвенційних підходів і часто опирається новим ідеям, віддаючи перевагу знайомому середовищу. Такій людині властива обмежена увага, а також труднощі у сприйнятті абстрактних чи теоретичних концепцій. Вона схильна до скептицизму та мислить у традиційній, усталеній парадигмі, залишаючись послідовною та обережною у своїх діях. Натомість людина з високим показником відкритості зазвичай дуже креативна, кмітлива та прониклива, із широким колом інтересів, яка відкрито ставиться до нових знайомств. Вона охоче пробує нові речі, відвідує нові місця, демонструє нестандартність у своїх підходах та легко працює з абстрактними поняттями. Крім того, такій особистості властива зосередженість на нових викликах, інтелектуальна допитливість та винахідливість.

Людина з низькою сумлінністю зазвичай невимушена, дещо недбала, не приділяє уваги порядку та деталям і не любить структуру чи фіксований розклад. Вона часто відкладає важливі завдання, рідко завершує їх вчасно, не дотримується графіків, не повертає речі на свої місця та регулярно запізнюється на зустрічі. Людина з високою сумлінністю зазвичай

компетентна, добре організована та орієнтована на цілі та деталі. Вона дисциплінована, завчасно готується, обдуманно діє та доводить важливі завдання до кінця вчасно, не піддається імпульсам, любить чіткий розклад, приходить на зустрічі вчасно й вирізняється надійністю та наполегливістю.

Людина з низькою екстраверсією зазвичай зосереджена на собі, стримана й надає перевагу самотності. Вона не любить уваги, швидко виснажується від активного спілкування, уникає невимушених розмов і важко починає діалоги, а перед тим як заговорити, ретельно обдумує свої слова й відзначається вдумливістю. Людина з високим показником екстраверсії зазвичай відкрита, енергійна й товариська. Вона легко розпочинає розмови, любить бути в центрі уваги та отримує задоволення від взаємодії з іншими. Така людина швидко заводить нові знайомства, має широке соціальне коло, упевнено почувається в соціальних ситуаціях і нерідко говорить спонтанно, не надто замислюючись над словами.

Людина з низькою згодливістю зазвичай більш різка й відсторонена, мало цікавиться іншими та може здаватися образливою чи зневажливою. Вона не надто переймається чужими почуттями чи проблемами, схильна до маніпуляцій, надає перевагу конкурентності й упертості у спілкуванні. Людина з високою згодливістю зазвичай доброзичлива й співчутлива, вирізняється альтруїзмом, терплячістю та лояльністю, щиро цікавиться іншими, прагне допомагати, легко співпереживає та віддає перевагу співпраці замість конфліктів. Така людина ввічлива, надійна, уважна до оточення, скромна та загалом підтримує тепле й позитивне спілкування.

Людина з низьким показником невротизму зазвичай стабільна емоційно, добре справляється зі стресом і рідко відчуває смуток чи пригніченість. Вона мало переймається, залишається розслабленою, упевненою та врівноваженою, а також має оптимістичний погляд на життя. Людина з високим рівнем невротизму часто відчуває тривогу й напруження, легко піддається стресу та стає дратівливою. Їй властиві різкі зміни настрою, імпульсивність, ревності, надмірна чутливість і схильність до самокритики.

Крім того, така людина часто може почуватися невпевненою, нестабільною, сором'язливою та схилитися до песимістичного бачення подій.

Таким чином, після детального дослідження стає більш зрозуміло, як та чи інша риса може впливати на досвід роботи з людиною. Тому важливо обрати моделі, які найкраще зможуть визначити рівень відповідності кандидата тій чи іншій рисі, щоб потім змодельовати найбільш точний психологічний профіль людини.

2.4.1 Підходи до аналізу особистості

У дослідженнях із прогнозування особистісних рис на основі текстів використовуються різні інструменти NLP, що дають змогу оцінювати психологічні характеристики автора. Найпоширенішими підходами є аналіз лінгвістичних індикаторів (LIWC), використання психолінгвістичних словників, а також застосування сучасних методів векторного подання слів та трансформерних моделей.

Одним із найвідоміших підходів є LIWC, що є психологічно орієнтованим лінгвістичним інструментом, створеним для аналізу емоційних, когнітивних і поведінкових патернів у текстах [28]. LIWC містить заздалегідь визначені категорії слів і виразів, які згруповані відповідно до психологічних конструкцій, таких як емоційні стани, соціальні процеси, когнітивні стратегії або маркери стилю письма. Під час аналізу тексту інструмент рахує частоту появи слів у кожній категорії та на основі цього формує статистичний профіль автора. У дослідженнях особистості LIWC часто використовують для прогнозування рис «Big Five», зокрема на основі широко відомого «Essays Dataset». Попри свою популярність, цей підхід має суттєві обмеження, адже він працює зі статичним словником, не враховує контекст слова в реченні та не здатний розпізнавати багатозначність, іронію, прагматичні нюанси чи глибинні смислові структури. Унаслідок чого навіть добре структуровані лінгвістичні

категорії не забезпечують достатньої точності в сучасних сценаріях психологічного аналізу текстів, де контекст відіграє ключову роль.

Іншим традиційним інструментом є психолінгвістична база даних MRC, що містить десятки атрибутів для понад 150 тисяч англійських слів. На відміну від LIWC, MRC не фокусується на психологічних категоріях, а зберігає фундаментальні характеристики слів, що включають в себе семантичні та синтаксичні властивості, фонологічні параметри, орфографічні особливості, асоціативні норми та когнітивні показники, такі як образність або знайомість слова [29]. Такий інструмент створює можливість формувати детальні ознаки, пов'язані з обробкою мови на базовому психолінгвістичному рівні. Проте, як і LIWC, MRC розглядає слова ізольовано, без урахування їхніх зв'язків у конкретному тексті. Ця обмеженість не дозволяє моделі аналізувати структуру висловлювань, враховувати контекст або фіксувати складні комунікативні наміри автора. Статичність ознак і відсутність контекстуального аналізу роблять MRC недостатньо ефективним для точного прогнозування особистісних характеристик, які часто проявляються у стилі, способі формулювання думок та взаємодії між словами в широких фрагментах тексту.

Потреба у поданні, здатному відтворювати значення слів у контексті, призвела до розвитку технік векторного кодування, а згодом і до появи трансформерів. Як було розглянуто раніше, початкові підходи, такі як one-hot encoding, представляли слова у вигляді великих розріджених векторів, що не містили жодної семантичної інформації. Суттєвим кроком уперед стали алгоритми Word2Vec, FastText і GloVe, які створювали щільні вектори, що відображали семантичну подібність між словами. Справжній прорив у моделюванні тексту забезпечили трансформерні моделі, які завдяки механізму самоуваги здатні враховувати взаємозв'язки між словами в реченні та будувати глибоко контекстуальні подання. Однією з найвідоміших моделей цього класу є BERT, попередньо натренована на величезних корпусах і здатна розуміти лексичні, синтаксичні та семантичні

характеристики тексту в обидвох напрямках, тобто ліворуч і праворуч від поточного слова. На основі BERT з'явилися її покращені модифікації, зокрема RoBERTa, яка використовує оптимізовану процедуру навчання, збільшені обсяги даних та усунення деяких обмежень вихідної архітектури. Ці моделі забезпечують рівень розуміння контексту й точності, недосяжний для словникових або статичних методів. З огляду на вимоги задачі побудови профілю особистості, у межах цієї роботи було обрано моделі сімейства BERT та RoBERTa. Обидві моделі здатні враховувати складні мовні патерни, стилістичні особливості та лексичні нюанси, які є ключовими для визначення психологічних характеристик. Крім того, існують спеціалізовані адаптації на основі BERT, що попередньо натреновані на задачах прогнозування рис «Big Five». RoBERTa в свою чергу забезпечує підвищену стабільність та точність завдяки кращим гіперпараметрам і потужнішому навчанню, що добре підходить для задач визначення настрою та тональності. У результаті ці моделі є найбільш придатними для завдання аналізу особистості, оскільки поєднують глибоке розуміння тексту, контексту і високу ефективність у класифікаційних сценаріях.

2.4.2 Архітектура моделі BERT та її використання в аналізі особистості

BERT є попередньо натренованою мовною моделлю, побудованою на архітектурі трансформера. При навчанні моделі виконуються дві ключові задачі, а саме Masked Language Model (MLM), де частину слів у реченні приховують і модель має їх відновити, та Next Sentence Prediction (NSP), яке оцінює зв'язок між двома реченнями. Після цього базове ядро моделі може бути донавчено для конкретних прикладних задач, таких як класифікація текстів, аналізу настроїв чи визначення особистісних рис.

Модель BERT здійснила справжній прорив у сфері обробки природної мови завдяки здатності враховувати контекст одночасно з обох сторін. На

відміну від традиційних моделей, таких як RNN або ранні варіанти трансформерів, які аналізували текст послідовно лише в одному напрямку, BERT опрацьовує речення повністю, одразу формуючи двобічне розуміння контексту. Це дозволяє моделі точніше інтерпретувати значення слів і їхні взаємозв'язки. Архітектурно BERT використовує кодувальну частину трансформера, яка складається з механізму мультиголовної самоуваги та багатошарових передавальних мереж. Така структура дає змогу моделі ефективно відстежувати складні залежності всередині тексту та будувати глибокі контекстуальні подання. Схему процесів навчання моделі BERT представлено на рисунку 2.10.

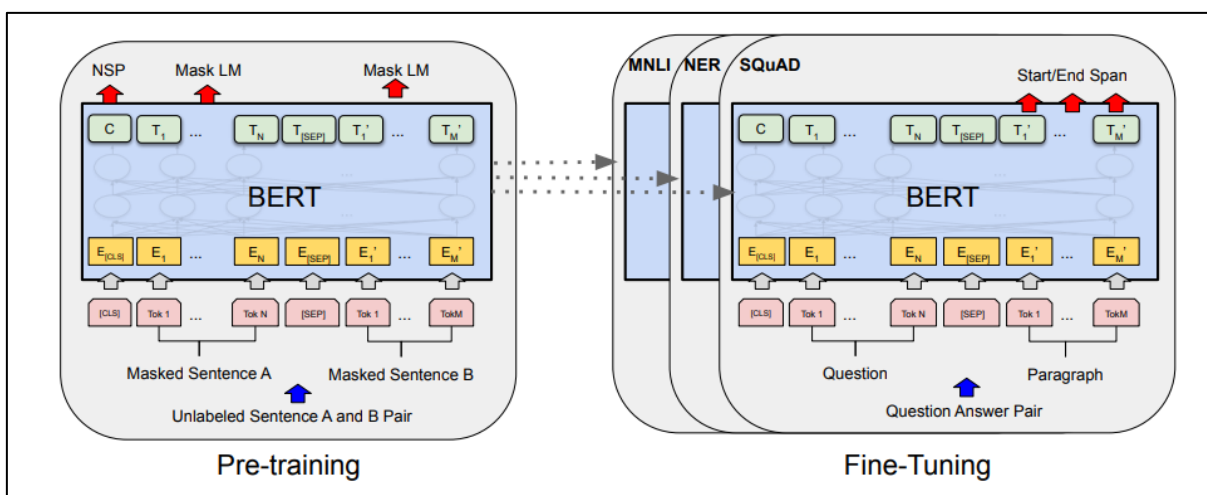


Рисунок 2.10 – Процеси навчання моделі BERT [20]

У завданні Masked Language Model частина слів у вхідній послідовності спеціально приховується, і модель навчається відновлювати ці пропущені токени, спираючись на контекст. Такий підхід дозволяє моделі формувати глибоке контекстуальне розуміння тексту, оскільки вона має враховувати як попередні, так і наступні слова, щоб правильно передбачити замаскований елемент.

Під час навчання у вхідній послідовності обираються випадкові токени які замінюються службовим символом маскування. Для кожної

замаскованої позиції модель прагне максимізувати ймовірність правильного слова, що показано за допомогою формули 2.10.

$$P(w_i | w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_n), \quad (2.10)$$

де w_i – слово, яке було замасковане та повинно бути передбачене моделлю;
 $P(\dots)$ – ймовірність того, що модель правильно відновить слово на позиції i , враховуючи доступний контекст.

Для оцінки різниці між передбаченим та правильним словом використовується крос-ентропія. Функцію втрат для передбачення слова у замаскованій позиції представлено формулою 2.11.

$$\mathcal{L}_{MLM} = - \sum_{i \in mask} \log P(w_i | w_{context}), \quad (2.11)$$

де \mathcal{L}_{MLM} – підсумкова втрата в задачі MLM;

$mask$ – позиції, у яких токени були замінені на замасковані;

$P(w_i | w_{context})$ – ймовірність того, що модель правильно передбачить слово.

У завданні Next Sentence Prediction модель навчається визначати, чи є друге речення логічним продовженням першого в початковому тексті. Такий підхід дає змогу моделі формувати уявлення не лише про контекст усередині речень, а й про зв'язність та структуру більш довгих текстових фрагментів. Навчання здійснюється за допомогою крос-ентропійної функції втрат, яка формально описується формулою 2.12.

$$\mathcal{L} = -[y \log P(\text{IsNext} | w_{[CLS]}) + (1 - y) \log P(\text{NotNext} | w_{[CLS]})], \quad (2.12)$$

де \mathcal{L} – втрата в задачі NSP;

y – бінарний індикатор, що приймає значення 1, якщо друге речення є справжнім продовженням першого, та 0, якщо це випадкове речення;

$P(\text{IsNext} \mid w_{[CLS]})$ – ймовірність, яку модель призначає гіпотезі, що друге речення дійсно є наступним;

$P(\text{NotNext} \mid w_{[CLS]})$ – ймовірність того, що друге речення не є логічним продовженням першого.

Для формування даних створюють пари речень (A, B) , де в половині випадків B є реальним продовженням A , а в іншій половині є випадково вибраним реченням із корпусу. Такий баланс дає моделі змогу навчитися розрізняти справжню текстову зв'язність від випадкової послідовності речень.

У задачі класифікації модель використовує векторне представлення спеціального токена, який містить узагальнену інформацію про всю послідовність. На основі цього вектора модель виконує бінарну класифікацію та намагається визначити, чи є друге речення справжнім продовженням першого. Іншими словами, модель оцінює ймовірність того, що друга фраза логічно слідує за першою.

Для навчання застосовується функція втрат на основі крос-ентропії, яка вимірює різницю між правильними мітками та передбаченими моделлю значеннями. Метою навчання є мінімізація цієї різниці, підвищуючи точність передбачення того, чи є речення наступним у послідовності.

Процес навчання BERT складається з двох етапів, а саме попереднього навчання та подальшого донавчання під конкретну задачу.

На першому етапі модель навчається на великому корпусі текстів у форматі самонавчання, використовуючи Masked Language Model і Next Sentence Prediction. Такий підхід дає можливість BERT сформуванню глибоке та універсальне розуміння структури мови.

Після попереднього навчання BERT уже містить багате лінгвістичне знання, але для ефективного розв'язання конкретних задач його необхідно донавчити на спеціалізованих даних. Це можуть бути завдання класифікації текстів, відповідей на запитання, розпізнавання сутностей чи аналізу тональності. Процес донавчання полягає в додаванні поверх базової моделі

нового вихідного шару, архітектура якого залежить від конкретного завдання. Наприклад, для аналізу емоцій або тональності зазвичай додається повнозв'язний шар із подальшою softmax-функцією, яка повертає ймовірності різних класів. Така гнучкість дозволяє BERT адаптуватися до найрізноманітніших практичних застосувань.

Використання BERT для аналізу особистості ґрунтується на тому, що мовлення людини відображає її когнітивні, емоційні та поведінкові особливості, а тому текст може виступати джерелом даних для визначення рис особистості. BERT, як контекстуальна трансформерна модель, здатна глибоко опрацювати структуру мовлення та вловлювати тонкі лінгвістичні маркери, які класичні словникові чи статистичні методи не помічають.

Однією з найбільш придатних моделей для визначення рис особистості за моделлю «Big Five» є «Minej/bert-base-personality», яка є спеціально адаптованою версією BERT, що попередньо натренована на текстах, що мають відповідні особистісні мітки. Ця модель використовує стандартну архітектуру BERT-base, що складається з 12 шарів енкодера, мультиголовної самоуваги та передавальних блоків. Завдяки механізму самоуваги модель здатна фіксувати як локальні, так і глобальні мовні патерни, що є критично важливими у психологічному аналізі тексту [30]. Модель намагається розпізнати латентні риси, адже мова людини відображає її поведінкові схильності, рівень емоційності, відкритість до нового, соціальність, стабільність чи тривожність.

Модель «Minej/bert-base-personality» була попередньо навчена на корпусах, де кожному фрагменту тексту відповідали значення п'яти основних рис особистості: відкритість, сумлінність, екстраверсія, згодливість та невротизм. Під час процесу BERT навчається асоціювати певні мовні особливості зі специфічними психологічними характеристиками. Наприклад, часте використання абстрактних термінів, метафор чи експресивних конструкцій часто корелює з високою

відкритістю, структуровані та точні висловлювання асоціюються із сумлінністю, активні та енергійні формулювання корелюють з екстраверсією.

Архітектурно модель працює так само, як звичайний BERT, де на вході текст токенізується, потім проходить через усі шари кодувальника, а фінальне представлення токена подається до класифікаційного шару, який повертає п'ять числових значень, де одне для кожної риси Big Five. Значення є неперервними, що дозволяє моделі прогнозувати рівень вираженості кожної риси, а не просто відносити текст до категорії.

Завдяки поєднанню контекстуальної потужності BERT та спеціалізованого донавчання на особистісно маркованих даних, дана модель здатна точно виявляти тонкі мовні індикатори, які класичні словникові методи не здатні зафіксувати. Це робить її ефективним інструментом для HR-аналітики, психологічної діагностики за текстом, а також для побудови профілів кандидатів на основі резюме та транскрибованих інтерв'ю.

2.4.3 Архітектура моделі RoBERTa та її використання аналізі настрою та тональності

RoBERTa є оптимізованою модифікацією моделі BERT і була розроблена для усунення низки обмежень оригіналу. Обидві моделі базуються на трансформерній архітектурі. Саме це дозволяє моделі виявляти контекст та складні зв'язки між словами в межах одного речення. Ключова відмінність RoBERTa полягає в її підході до навчання. З архітектурного погляду RoBERTa зберігає основні елементи багат шарового двонаправленого трансформера BERT, але вводить низку удосконалень, які підвищують ефективність та якість контекстуального аналізу [31].

Одним із ключових рішень стало повне вилучення завдання Next Sentence Prediction. У BERT це завдання мало допомагати моделі

розпізнавати логічний зв'язок між реченнями, однак подальші дослідження показали, що цей підхід мало впливає на ефективність у більшості задач і може вносити зайвий шум у навчання. RoBERTa відмовляється від NSP і зосереджується виключно на завданні Masked Language Modeling, що дозволяє моделі точніше формувати контекстуальні подання на рівні токенів. Другим важливим удосконаленням став перехід від статичного до динамічного маскування. У BERT маски формуються під час попередньої обробки даних і залишаються незмінними для всіх епох навчання, що обмежує різноманітність контекстів, з якими працює модель. RoBERTa натомість використовує динамічне маскування, де на кожному проходженні моделі по даних замасковані позиції вибираються випадково. Це забезпечує більшу варіативність, покращує узагальнення та робить модель стійкішою до різних мовних структур. Третя суттєва різниця стосується параметрів тренування. BERT навчався на відносно невеликих пакетах даних і у фіксовану кількість кроків. RoBERTa значно масштабувала ці параметри, адже використовувалися набагато більші розміри батчу, збільшувалася тривалість тренування та коригувалися графіки зміни швидкості навчання. Завдяки цьому модель отримувала стабільніші градієнти та могла засвоювати складніші мовні закономірності без змін архітектури. Нарешті, одним із найвагоміших покращень стала робота з розширеним та різноманітним корпусом текстів. Якщо BERT тренувався на близько 16 ГБ даних із Wikipedia та BookCorpus, то RoBERTa пройшла навчання на понад 160 ГБ тексту, що включає Common Crawl News, OpenWebText, Stories dataset, а також ті ж корпуси, що й BERT. Це надало моделі значно більш насичений набір мовних конструкцій, стилів і доменів, що прямо вплинуло на її здатність узагальнювати інформацію.

RoBERTa широко використовується для аналізу настрою та тональності завдяки своїй здатності створювати глибокі контекстуальні подання тексту та вловлювати найдрібніші емоційні й стилістичні нюанси. На відміну від простих моделей, які обмежуються поверхневими мовними

ознаками, RoBERTa здатна інтерпретувати приховані емоційні сигнали, визначати тональність висловлювань навіть у випадках, коли емоція виражена непрямо, і працювати з неформальними конструкціями, характерними для реальних текстів соціальних мереж чи контенту користувачів. Завдяки масштабному попередньому навчанню та покращеному тренувальному режиму RoBERTa забезпечує високу точність у визначенні позитивних, нейтральних і негативних оцінок, а також у розпізнаванні складних або змішаних емоцій, що робить її ефективною для побудови емоційних профілів.

Однією з відомих моделей на основі RoBERTa для емоційного аналізу є «SamLowe/roberta-base-go_emotions», створена спеціально для розпізнавання широкого спектра емоцій. В її основі лежить стандартна архітектура RoBERTa-base, яка містить 12 шарів трансформера з мультиголовною самоувагою та передавальними блоками. Це забезпечує здатність враховувати як локальні, так і глобальні зв'язки між словами, що особливо важливо для емоційного аналізу, де сенс формується не лише окремими словами, а й загальною манерою висловлювання. Поверх RoBERTa в цій моделі додано повнозв'язний вихідний шар, що формує ймовірності для 28 емоційних категорій, включно з нейтральною. Оскільки один текст може містити кілька емоцій одночасно, модель використовує активацію sigmoid і навчалася не як багатокласовий, а як мультилейбловий класифікатор.

Навчання «SamLowe/roberta-base-go_emotions» здійснювалося на великому корпусі GoEmotions, що містить приблизно 58 тисяч анотованих фрагментів тексту з 27 емоціями. Для моделі застосовували функцію втрат Binary Cross-Entropy, оптимізатор AdamW та техніку динамічного маскування, завдяки яким RoBERTa отримувала різноманітні контексти та була здатна формувати стійкі та узагальнюючі емоційні представлення [32]. Довжина послідовностей, збалансований розмір батчу та механізми ранньої зупинки забезпечили точне та стабільне навчання. У результаті модель

навчилася вловлювати як сильні емоційні сигнали, так і тонкі психологічні відтінки, що робить її однією з найточніших систем для аналізу емоцій.

Таким чином, для аналізу емоційної складової було вирішено використовувати моделі засновані на архітектурі RoBERTa, через краще налаштування та більшу чутливість до завуальованого емоційного забарвлення.

2.5 Огляд підходів до задачі розпізнавання мовлення

Якщо розглядати сучасні підходи до розпізнавання мовлення, то донедавна одними з найпопулярніших були приховані моделі Маркова (НММ). Цей клас статистичних моделей відіграв ключову роль у класичних системах автоматичного розпізнавання мовлення ще з 1980-х років і став основою перших промислових систем автоматичного розпізнавання мовлення для диктування, діалогових систем та телефонних сервісів.

Приховані моделі Маркова представляють мовлення як послідовність прихованих станів, кожен з яких відповідає певному акустичному фрагменту або фонемі. Архітектурно НММ складаються з множини прихованих станів, ймовірностей переходів між ними та моделей емісії, що описують імовірність появи конкретного акустичного сигналу в кожному стані. Основою роботи НММ є припущення, що мовний сигнал можна розкласти на короткі незалежні інтервали, а перехід між станами підпорядковується властивості Маркова, де поточний стан залежить лише від попереднього.

Функціонування НММ передбачає два ключові етапи, які включають навчання моделі та пошук оптимальної послідовності станів. Для навчання використовують алгоритм Баума–Велша, який визначає найімовірніші параметри переходів та емісій. Для розпізнавання мовлення застосовується алгоритм Вітербі, що знаходить найбільш ймовірний шлях через приховані

стани на основі спостережуваного акустичного сигналу. Такі моделі добре працювали у зв'язці з гаусовими сумішами (GMM), формуючи класичну парадигму GMM-HMM. У архітектурі GMM-HMM роль моделі емісії відіграє суміш гаусових розподілів. Для кожного прихованого стану GMM описує ймовірнісний розподіл значень спектральних ознак, що відповідають цьому стану. Змішання декількох гаусових компонентів дозволяє моделям GMM гнучко наближати складні розподіли акустичних сигналів, що було ключовим для досягнення високої точності у класичних системах. Архітектура GMM-HMM історично була стандартом у галузі розпізнавання мовлення, адже вона добре працювала в умовах контрольованих акустичних середовищ. GMM дозволяли моделювати багатомодальні розподіли ознак, а HMM забезпечували надійне моделювання часової структури мовлення. Система була модульною, де акустична модель, мовна модель та лексичний словник працювали окремо, що давало можливість зручно керувати кожним компонентом [33].

Попри появу глибинних нейронних мереж (DNN), приховані моделі Маркова продовжують широко використовуватись у гібридних архітектурах DNN-HMM. У таких системах глибинні нейронні мережі замінюють гаусові як модель емісії, але сам механізм переходів і структура послідовностей залишаються базованими. Завдяки цьому гібридні моделі поєднують здатність глибинних мереж до складного представлення акустичних ознак з надійною ймовірнісною послідовністю станів HMM, що забезпечує стабільність і керованість.

Однак для даного проекту використання HMM і їх варіацій не є доцільним. Ці моделі погано масштабуються для природного безкомандного мовлення, не вміють ефективно працювати з довгостроковими залежностями, не підтримують контекстуальне моделювання на рівні семантики та суттєво поступаються сучасним end-to-end моделям на основі трансформерів [34].

Після піку епохи GMM-HMM та гібридних DNN-HMM наступним кроком у розвитку систем розпізнавання мовлення стали моделі, які використовують механізм Connectionist Temporal Classification (CTC). З появою CTC-моделей почався перехід до end-to-end підходу, де система навчається зіставляти аудіосигнал безпосередньо з текстовою транскрипцією, без необхідності ручного вирівнювання фонем або побудови складної графової структури станів, які були характерними для HMM. Архітектурно CTC-моделі складаються з рекурентної нейронної мережі, яка генерує послідовність ймовірностей для кожного кроку часу, та CTC-критерію, який визначає спосіб порівняння цих вихідних послідовностей із цільовим текстом. Основна ідея полягає у тому, що довжина аудіосигналу зазвичай набагато більша за довжину текстової послідовності, а відповідність між ними невідома. CTC дозволяє моделі самостійно визначати, які часові фрагменти відповідають яким символам, шляхом введення спеціального пустого символу та правил колапсації повторів [35].

Завдяки природній для рекурентних мереж здатності моделювати часові залежності CTC-системи навчилися з більшою точністю розпізнавати природну мову та демонстрували значно кращу узагальнюючу здатність порівняно з класичними GMM-HMM. Проте CTC має свої обмеження, адже такі моделі припускають умовну незалежність вихідних символів, що ускладнює моделювання мовного контексту. Саме через ці обмеження CTC-моделі стали перехідним етапом між HMM-підходами та сучасними трансформерними end-to-end системами. У рамках даного проекту CTC була б значно кращим варіантом за HMM, але все ж поступається сучасним трансформерам.

Наступним етапом еволюції технологій розпізнавання мовлення стала поява end-to-end системи на основі трансформерів. Цей підхід кардинально змінив парадигму ASR, оскільки дозволив моделювати довгострокові

залежності у звуковому сигналі та контекст текстової послідовності без обмежень, характерних для попередніх моделей.

Архітектурно трансформерні системи для розпізнавання мовлення складаються з кодувальника, який перетворює аудіосигнал у компактні латентні представлення, та декодера, який покроково генерує текстову транскрипцію. Механізм самоуваги дає змогу моделі аналізувати весь аудіосигнал як глобальний контекст, визначаючи важливі фрагменти незалежно від їхньої позиції в часі [36].

Трансформерні підходи мають низку критичних переваг. По-перше, вони значно краще працюють зі спонтанним, нерівномірним, швидким та емоційним мовленням, оскільки здатні враховувати контекст на рівні фрази та навіть абзацу. По-друге, такі моделі можуть навчатися на великих масивах даних, що підвищує їх стійкість до шумів, різних акцентів та варіацій вимови. По-третє, трансформери природно інтегрують мовну модель у процес генерації тексту, що істотно покращує граматичну і семантичну коректність транскрипції. Крім того, можлива багатомовна підтримка, коли одна модель розпізнає десятки мов без потреби у створенні окремих спеціалізованих систем.

У межах даного проекту саме трансформерний підхід був використаний для реалізації системи розпізнавання мовлення. Зокрема, застосовано модель Whisper, яка поєднує багатомовний кодувальник-декодер на основі трансформера, масштабне навчання на великому обсязі аудіо-текстових даних і здатність до високоточного транскрибування навіть за складних умов. Whisper забезпечив стабільну та якісну обробку мовлення кандидатів, що надало якісні транскрибовані текстові дані, з якими працювали моделі аналізу особистості.

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

3.1 Використані технології

Першим кроком для розробки зручного та якісного застосунку є вибір технологій, які можуть найкраще підійти для поставленої задачі та які будуть працювати в тандемі один з одним, утворюючи злагоджену систему. Правильно підібрані інструменти, мови програмування, фреймворки та бібліотеки дозволяють оптимально розподіляти ресурси, зменшувати кількість помилок та забезпечувати стабільну продуктивність. В даному проекті по розробці застосунку для HR, де повинно зберігатися та оброблятися велика кількість кандидатів, дуже важливим є підібрати правильну систему управління базами даних, для управління даними про кандидатів. Також, не менш важливим є вибір моделей, які будуть виконувати особистісний аналіз кандидатів, моделі повинні бути точними та приймати в себе великі обсяги текстових даних. Додатково потрібна хороша модель з розпізнавання мовлення для транскрибування аудіо співбесід для подальшого аналізу. Щоб втілити застосунок у життя та об'єднати допоміжні технології потрібна підходяща та гнучка мова програмування.

В результаті дослідження та порівняння існуючих технологій, для створення застосунку з аналізу особистісних характеристик кандидатів на посаду, було обрано MongoDB Cloud в якості системи управління базами даних (СУБД), Whisper API для транскрибування мовлення, моделі з HuggingFace для визначення особистісних характеристик, мови програмування Python та JavaScript з використанням різних фреймворків для роботи з клієнтською та серверною частинами.

3.1.1 Whisper API

Whisper є системою автоматичного розпізнавання мовлення від OpenAI, яка була навчена на 680 000 годинах різноманітних багатомовних та багатозадачних контрольованих даних, отриманих з Інтернету. Результати тестів системи показують, що використання такого великого та різноманітного набору даних підвищує стійкість системи до акцентів, фонового шуму та спеціалізованої термінології [37]. Крім того, це дозволяє транскрибування різних мов та забезпечує переклад з цих мов на інші. Код даної системи є загальнодоступним, що сприяє зрозумілому та простому застосуванню Whisper у своєму проекті.

Система Whisper заснована на базовій трансформерній моделі, а її архітектура являє собою наскрізний підхід у вигляді кодувальника-декодера. Архітектуру моделі можна побачити на рисунку 3.1.

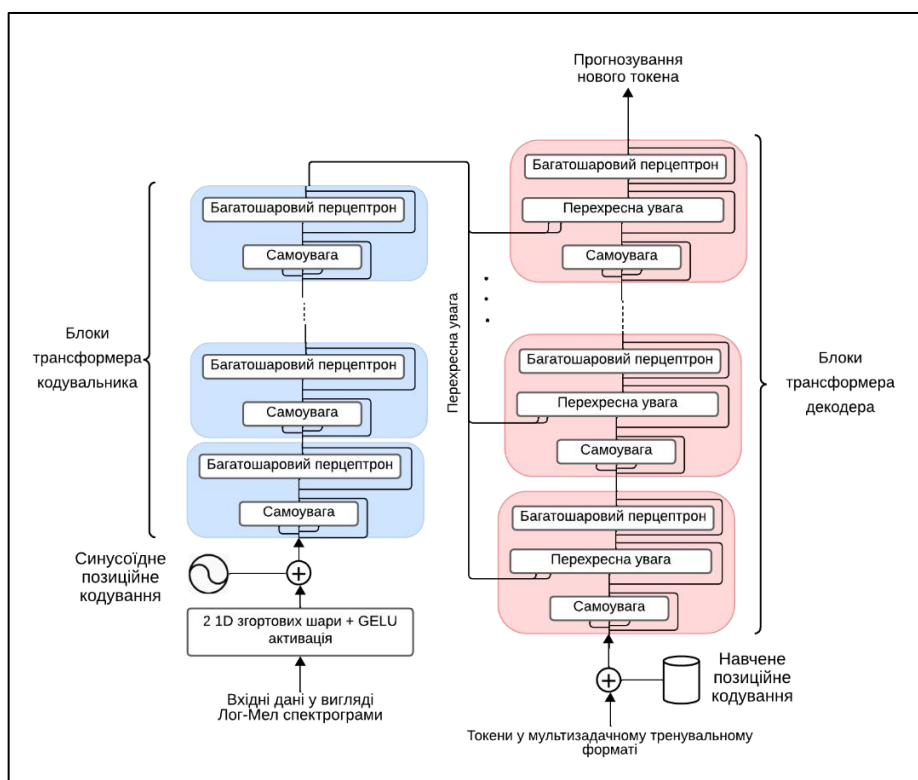


Рисунок 3.1 – Модель трансформера системи Whisper [38]

Система створена у вигляді трансформерного кодувальника-декодера, так як така архітектура є найбільш підходящою для надійного масштабування. На початку роботи моделі, вхідний звук розбивається на 30-секундні фрагменти та представляється як log-mel спектрограма, яка потім масштабується, нормалізується та передається у кодувальник. Це вхідне представлення обробляється кодувальником за допомогою стебла, що складається з двох шарів згортки з шириною фільтра три та функцією активації GELU, де крок другого шару згортки дорівнює двом. Потім застосовуються блоки кодувальника трансформера після додавання синусоїдальних позиційних вбудовувань до виходу стебла. Вихід кодувальника проходить нормалізацію останнього шару після того, як трансформер використовує залишкові блоки перед активацією. Декодер використовує пов'язані представлення токенів входу-виходу та вивчені позиційні вбудовування. На отриманих токенах декодер навчається передбачати відповідний до вхідного аудіо текстовий підпис, разом з цим, спеціальні токени спрямовують єдину модель на виконання таких завдань, як ідентифікація мови, часові позначки на рівні фраз, багатомовне транскрибування мовлення та переклад мовлення на англійську мову. Ширина та кількість блоків трансформера ідентичні для кодувальника та декодера [38].

Хоча передбачення слів, які звучать у аудіофрагменті є головною задачею системи розпізнавання мовлення, одне й те саме аудіо можна використовувати для широкого спектру дій, таких як розпізнавання мовлення, вирівнювання, виявлення голосової активності, транскрибування та переклад. Багато інших елементів, таких як інверсна нормалізація тексту та виявлення голосової активності, можуть бути включені до повноцінної системи розпізнавання мовлення. Ці елементи часто обробляються окремо, створюючи навколо головної задачі розпізнавання мовлення досить складну систему. Розробники даної моделі розширили функціонал та зробили систему мультизадачною, при тому сильно її не ускладнюючи.

Модель, яка зараз розглядається, виконує декілька можливих задач системи розпізнавання мовлення. Повну схему, яка відображає принцип роботи моделі Whisper можна побачити на рисунку 3.2.

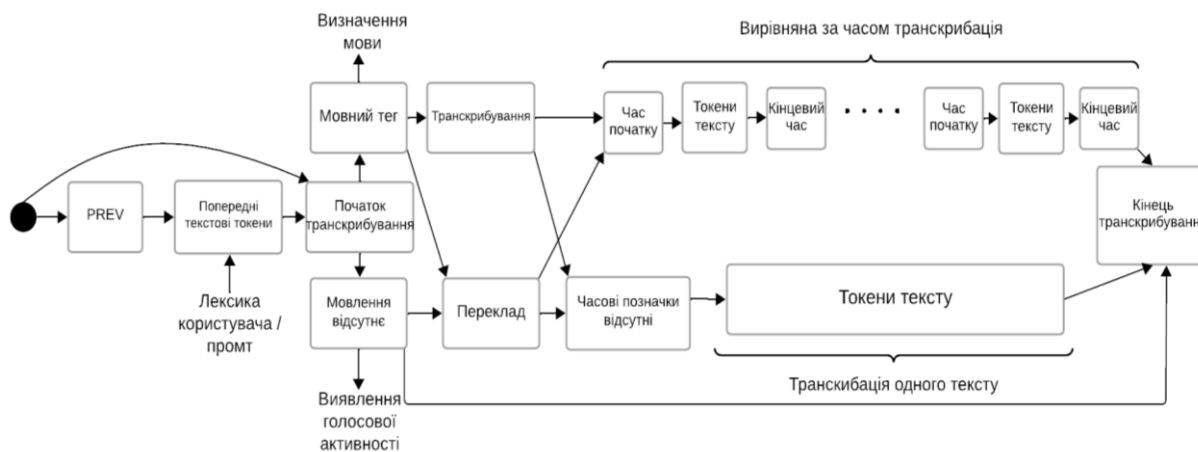


Рисунок 3.2 – Принцип роботи моделі [38]

Початок передбачення визначається токеном початку транскрибування. Спочатку йде визначення мови, якою розмовляють у аудіо, та яка представлена унікальним токеном для кожної мови з навчального набору. Модель навчена передбачати токен, який сигналізує про відсутність мовлення в аудіосегменті. Використовуючи токени транскрибування або перекладу, наступний токен вказує на завдання. Далі, якщо потрібно, додається токен відсутності часових позначок, щоб вказати, чи слід передбачати часові позначки. Вивід починається, коли завдання та бажаний формат повністю визначені. Для визначення часових позначок, час прогнозується відносно поточного аудіосегмента, квантується до найближчих 20 мілісекунд, і додаткові токени додаються до словника для кожного з них. Прогнозування часових позначок чергується з прогнозуванням тексту. Таким чином, токен часу початку прогнозується перед текстом кожної строки, а токен часу завершення визначається після

неї. Після виконання всіх кроків встановлюється токен завершення транскрибування [38].

Таким чином, дана модель трансформера навчається широкому спектру завдань обробки мовлення, таких як виявлення голосової активності, розпізнавання мовлення багатьма мовами, переклад та ідентифікація розмовної мови. Оскільки кожне з цих завдань спільно представлено як послідовність токенів, які декодер повинен передбачити, одна модель може замінити численні кроки у звичайному конвеєрі обробки мовлення.

Після розгляду та аналізу даної моделі, було вирішено використати Whisper API для виконання завдання транскрибування аудіозапису співбесіди. Завдяки своїй якості, мультифункціональності та доступності для простої інтеграції в будь-який проект, дана система відмінно підходить для поставленої задачі.

3.1.2 Hugging Face моделі

Hugging Face – це компанія, яка підтримує величезну спільноту з однойменною назвою та відкритим кодом. Спільнота створює інструменти, моделі машинного навчання та платформи для роботи зі штучним інтелектом, зосереджуючись на науці про дані, машинному навчанні та обробці природної мови. Hugging Face добре відома своєю бібліотекою трансформерів NLP та платформою для обміну наборами даних і моделями.

Hugging Face сформувала одну з найактивніших спільнот у сфері штучного інтелекту, де щодня з'являються нові моделі, набори даних, навчальні матеріали та наукові роботи. Платформа надає потужний API, який дає змогу розробникам легко інтегрувати моделі у будь-які власні застосунки. Крім того, Hugging Face підтримує широкий спектр завдань і застосовується у багатьох галузях. Нижче наведено основні переваги цієї платформи:

– Hugging Face через свій Model Hub надає доступ до тисяч попередньо навчених моделей, призначених для різних завдань. Платформа надає спектр моделей від розпізнавання мовлення й класифікації тексту, генерації та узагальнення тексту до надання відповідей на запитання й створення зображень. Model Hub функціонує як зручна платформа, де користувачі можуть швидко знаходити, завантажувати та налаштовувати моделі всього за кілька рядків коду. Це значно скорочує час і витрати розробників та дослідників порівняно з навчанням моделей з нуля;

– бібліотеки Hugging Face вирізняються простотою у використанні та якісною документацією. Завдяки цьому навіть новачки можуть швидко налаштовувати потужні моделі й виконувати складні завдання. Наявність як базових, так і передових інструментів зробила розробку систем штучного інтелекту доступною для значно ширшого кола користувачів і фахівців;

– крім навчання, Hugging Face значно полегшує розгортання моделей у продакшн. З її допомогою можна інтегрувати моделі в мобільні застосунки, внутрішні корпоративні системи або в Інтернет, не маючи багато досвіду. Завдяки широкій підтримці, платформа особливо зручна та приваблива для бізнесу та стартапів;

– поряд із широким вибором доступних технологій, активна спільнота Hugging Face перетворила платформу на осередок для розробників, аналітиків даних і дослідників. Тут початківці мають змогу навчатися у досвідчених фахівців, обмінюватися досвідом і знаходити відповіді на питання від тих, хто вже розв'язував схожі завдання [39].

Таким чином, після дослідження платформи Hugging Face, було вирішено використовувати моделі аналізу текстів саме звідси. Проста інтеграція моделей, великий вибір моделей, активна спільнота користувачів, та зрозуміла документація ідеально підходять для інтегрування потрібних моделей у даний проект з визначення особистісних характеристик з текстів.

Як зазначалося раніше, В якості моделі для аналізу особистісних характеристик було обрано «bert-base-personality» від Minej, яка працює на

основі системи «Big Five», яка має в собі п'ять характеристик, а саме екстраверсію, невротизм, згодливість, сумлінність, відкритість.

Алгоритм роботи моделі:

- спочатку відбувається подача тексту, можна подавати один або кілька текстів одночасно;
- текст обробляється за допомогою BertTokenizer, всі слова перетворюються на токени;
- токени подаються в BERT Base Uncased, BERT обчислює контекстні представлення кожного токена (вектори), які через класифікаційний шар перетворюються на оцінки для кожної з 5 рис;
- вихідні значення перетворюються у ймовірності або значення від 0 до 1 для кожної риси (де 0 – низька вираженість риси, а 1 – висока);
- результат повертається у вигляді словника з оцінками для кожної риси.

Приклад роботи даної моделі можна побачити на рисунку 3.3.

```
text_input = "I am feeling excited about the upcoming event."  
personality_prediction = personality_detection(text_input)  
  
print(personality_prediction)
```

Output:

```
{  
  "Extroversion": 0.535,  
  "Neuroticism": 0.576,  
  "Agreeableness": 0.399,  
  "Conscientiousness": 0.253,  
  "Openness": 0.563  
}
```

Рисунок 3.3 – Приклад роботи моделі «bert-base-personality»

Після дослідження моделі було обрано саме її для аналізу головних особистісних характеристик за методикою «Big Five». Дана модель видає якісні та точні результати, базування на трансформеній моделі BERT дозволяє максимально точно визначати особистісні характеристики, а розташування на платформі HuggingFace надає просту інтеграцію до цього проекту.

Для визначення головної емоційної складової було використано модель «roberta-base-go_emotions» від SamLowe, архітектуру якої також було розглянуто раніше.

Алгоритм роботи моделі:

- відбувається подача тексту;
- модель токенизує текст через відповідний токенизатор;
- після обробки та аналізу модель виводить 28 ймовірностей, по одній для кожної емоції;
- застосовується поріг (0.5), щоб перетворити ймовірності в бінарні рішення, щоб визначити, чи присутня емоція чи ні.

Приклад роботи моделі можна побачити на рисунку 3.4.

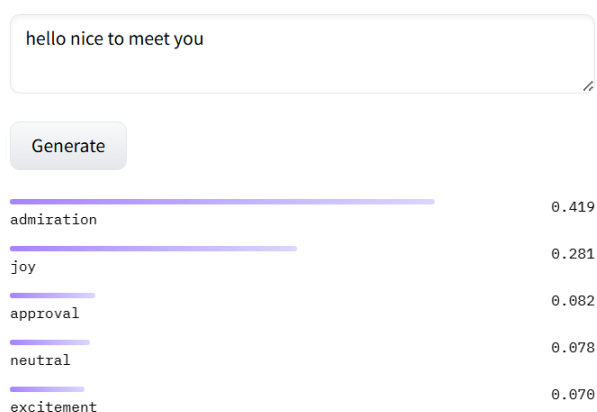


Рисунок 3.4 – Приклад роботи моделі «roberta-base-go_emotions»

В результаті дослідження архітектури моделі та принципу її роботи, було обрано саме її для виконання задачі визначення домінантної емоції.

Для визначання настрою тексту було взято модель «twitter-roberta-base-sentiment». Ця модель є трансформерною та базується на архітектурі RoBERTa-base, розроблена дослідницькою групою з CardiffNLP для аналізу тональності текстів. Її мета полягає у визначенні емоційного забарвлення, чи є текст позитивним, нейтральним чи негативним. Модель була спочатку попередньо навчена на великому корпусі даних, який містить мітки для класифікації тональності, що дозволило моделі навчитися розрізняти емоції у текстах. Завдяки поєднанню великої кількості даних і потужної архітектури RoBERTa, модель демонструє високу точність у визначенні настроїв у публікаціях та може бути використана у різноманітних галузях [40].

Алгоритм роботи моделі:

- відбувається подача тексту;
- модель токенизує текст через відповідний токенизатор;
- модель видає ймовірності для кожного настрою. 0 для негативної, 1 для нейтральної і 2 для позитивної тональності;
- найвища ймовірність визначає, як модель класифікує текст.

Приклад результату роботи моделі можна побачити на рисунку 3.5.

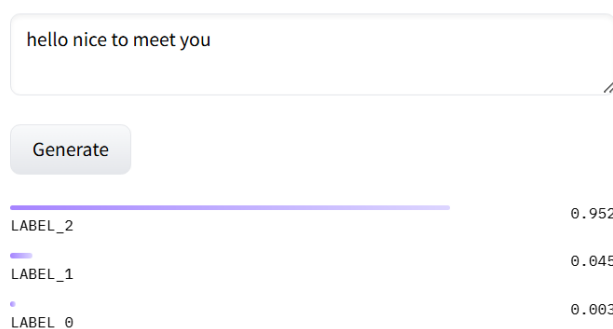


Рисунок 3.5 – Приклад роботи моделі «twitter-roberta-base-sentiment»

В результаті дослідження даної моделі, було обрано використати саме її для інтеграції в проект. Дана модель має хорошу якість та точно класифікує головний настрій текстів.

3.1.3 MongoDB Cloud

MongoDB – це сучасна документно-орієнтована система управління базами даних, що належить до категорії NoSQL. Вона була створена компанією MongoDB Inc. у 2009 році та є однією з найпопулярніших альтернатив традиційним реляційним базам даних, таким як MySQL чи PostgreSQL [41].

На відміну від класичних СУБД, MongoDB зберігає дані у форматі BSON (Binary JSON), що є бінарним представленням JSON-документів. Кожен документ у MongoDB є набором пар «ключ-значення» і може мати гнучку структуру. Наприклад, різні документи в одній колекції можуть містити різні поля. Такий підхід спрощує зберігання напівструктурованих даних і дозволяє швидко змінювати схему без необхідності перебудови всієї бази. MongoDB відома своєю масштабованістю, високою продуктивністю та гнучкістю. Вона підтримує розподіл даних між кількома серверами та реплікацію. Завдяки цьому MongoDB добре підходить для великих проєктів, де потрібно працювати з великими обсягами даних у режимі реального часу. Таким чином, ця СУБД підходить для веб-додатків, аналітичних чи рекомендаційних систем. MongoDB також пропонує хмарний сервіс MongoDB Atlas, який дозволяє створювати, масштабувати й керувати базами даних безпосередньо у хмарі.

MongoDB Cloud – це хмарна екосистема сервісів, створена компанією MongoDB Inc., яка надає можливість розгорнути, керувати та масштабувати бази даних MongoDB у хмарі без необхідності налаштовувати власну інфраструктуру. Центральним елементом цієї платформи є зазначений раніше MongoDB Atlas, який автоматизує всі операції з базою даних, такі як створення, резервне копіювання, оновлення, безпека та масштабування. MongoDB Cloud підтримує розгортання на найпопулярніших хмарних платформах, що дає користувачам гнучкість у виборі середовища. Система автоматично розподіляє навантаження між кількома серверами і створює

репліки даних для забезпечення високої доступності та відмовостійкості. Також, система надає можливість легко керувати кластерами через веб-інтерфейс Atlas або CLI/API, а також інтегрувати їх із застосунками через стандартні драйвери MongoDB [42].

Під час розробки, завдяки MongoDB Cloud, можна повністю зосередитись на логіці застосунку, не переймаючись адмініструванням бази даних. Цей підхід забезпечує високу продуктивність, автоматичне масштабування, безпечне зберігання даних і просту інтеграцію з сучасними хмарними рішеннями, що робить MongoDB Cloud потужним інструментом для створення сучасних, гнучких і масштабованих застосунків. Саме через ці переваги було обрано саме цю СУБД, можливість зберігати великий обсяг даних в хмарі та просте адміністрування ідеально підходить для даного проекту, де потрібно буде зберігати дані про велику кількість кандидатів.

3.1.4 Мови програмування

Вибір мов програмування є критично важливим етапом у розробці будь-якого програмного продукту, оскільки саме він визначає продуктивність, стабільність, масштабованість та зручність подальшої підтримки системи. Кожна мова має власні сильні сторони та обмеження, тому технології слід підбирати відповідно до вимог проекту, його функціональних завдань та специфіки взаємодії з користувачем. У межах даного веб-застосунку для аналізу особистісних характеристик особливо важливо забезпечити узгодженість між серверною та клієнтською частинами. Тому бекенд був реалізований мовою, що пропонує простоту, широкий набір інструментів і підтримку сучасних AI-бібліотек, тоді як інтерфейс потребував технології, здатної забезпечити динамічність і зручність роботи користувача. Успішне поєднання цих мов стало основою ефективною та збалансованою архітектурою всього проекту.

Python – це мова програмування високого рівня загального призначення. Її філософія дизайну наголошує на читабельності коду з використанням значних відступів. Python динамічно перевіряє типізацію та збирає сміття. Він підтримує кілька парадигм програмування, включаючи структурне, об'єктно-орієнтоване та функціональне програмування. Важливою особливістю мови є її велика екосистема, що включає в себе тисячі відкритих бібліотек і фреймворків, які покривають практично всі сфери застосування.

У межах цього проекту Python було використано для реалізації серверної частини системи. Було створено веб-сервер, який відповідає за обробку запитів від клієнтської частини, виконання необхідних обчислень і формування відповідей у форматі JSON. Для цього було застосовано сучасний асинхронний фреймворк FastAPI, який поєднує простоту розробки з високою продуктивністю. FastAPI підтримує автоматичну генерацію документації API, валідацію даних і типізацію, що робить код зрозумілим і надійним. Сервер запускається за допомогою Uvicorn, який є високопродуктивним ASGI-сервером, який забезпечує швидке асинхронне обслуговування HTTP-запитів і підтримує сучасні протоколи взаємодії.

Мова Python у цьому проекті також використовувалася для реалізації логіки аналізу тексту та взаємодії з моделями машинного навчання. З її допомогою оброблялися текстові дані, виконувалася попередня обробка, підключалися попередньо навчені моделі для класифікації чи аналізу емоцій, а також здійснювалася інтеграція результатів у загальний API. Таким чином, Python виступає центральним елементом програмної архітектури, який забезпечує взаємодію між алгоритмічною частиною та користувацьким інтерфейсом.

JavaScript – це мова програмування та основна технологія веб-платформи, поряд з HTML та CSS. Вона була розроблена з метою забезпечення динамічності веб-сторінок, а з часом перетворилася на повноцінну багатопарадигмальну мову програмування, що підтримує

об'єктно-орієнтований, функціональний і подієвий підходи. JavaScript виконується безпосередньо у браузері користувача, що дозволяє створювати інтерактивні елементи, обробляти події та взаємодіяти з сервером у режимі реального часу без необхідності перезавантаження сторінки.

У межах цього проекту JavaScript використовувався для реалізації клієнтської частини системи. Основним інструментом для створення інтерфейсу було обрано фреймворк React, який базується на компонентному підході. Це означає, що інтерфейс поділено на окремі незалежні компоненти, кожен з яких відповідає за свій логічний і візуальний блок. React забезпечує ефективне оновлення інтерфейсу завдяки використанню віртуальної моделі об'єктів документа, що підвищує швидкодію та зменшує навантаження на браузер. Для керування залежностями фронтенду та зручного запуску застосунку було використано Yarn, який є менеджером пакетів, та забезпечує швидке встановлення бібліотек і стабільне керування версіями. Таким чином, JavaScript відповідає за створення зручного, динамічного та інтуїтивно зрозумілого інтерфейсу, який дозволяє користувачу взаємодіяти з аналітичними можливостями системи в реальному часі.

3.2 Опис проектованої системи

Перед тим, як реалізувати проект програмно, потрібно чітко визначити схему проекту та зрозуміти, як зробити процес роботи з веб-застосунком простим та зрозумілим для користувача. Спочатку потрібно визначити цільову аудиторію та її вимоги, це допоможе зробити хороший веб-застосунок, який користувач захоче використовувати.

Передбачається, що цільовою аудиторією застосунку будуть фахівці у сфері управління персоналом. Застосунок з автоматичного аналізу особистості, може значно скоротити час розглядання кандидатів та полегшить зберігання та презентацію даних про них. Наприклад, завдяки

цьому веб-застосунку, HR-фахівці зможуть не витратити час на запам'ятовування та записування всіх особистісних особливостей кожного кандидата, яких можуть бути сотні, всі дані будуть автоматично аналізуватися та зберігатися, а фахівцю достатньо буде згадати співбесіду та презентувати готовий аналіз для обґрунтування свого вибору начальству.

Після визначення цільової аудиторії, потрібно змоделювати вимоги типового користувача веб-застосунку. Визначення всіх необхідних функцій та покрокового алгоритму є невід'ємною частиною процесу розробки хорошого застосунку. Доступ до веб-застосунку можливий із будь-якого пристрою, таких як комп'ютера, ноутбука, планшета чи смартфона. Умовою є лише наявність підключення до інтернету. З цього виходить, що вимогою для користувача є наявність комп'ютеру або мобільного пристрою та володіння базовими навичками роботи з ними. З боку користувачів також є певні вимоги до роботи системи. Насамперед веб-застосунок має бути адаптивним і коректно відображатися на екранах різних розмірів, забезпечуючи комфортну взаємодію незалежно від типу пристрою. Інтерфейс повинен бути зрозумілим, логічно структурованим і простим у використанні, щоб користувач міг легко виконувати необхідні дії без додаткових інструкцій. Крім того, система має бути доступною цілодобово, працювати стабільно та без збоїв, навіть під час високого навантаження. Це буде гарантувати безперервність роботи й надійність сервісу. Головною задачею застосунку є автоматичний аналіз особистісних характеристик, тому робота моделей повинна бути злагодженою і точною. Процес завантаження даних та аналізу повинен бути швидким та ефективним, отримані діаграми та описи мають бути чітко відображеними та зрозумілими. Зв'язок з базою даних та моделями повинен проходити без затримок. Список кандидатів у базі має бути повним, а порівняння кандидата з вимогами до вакансії чітким та точним, рекомендації мають відповідати зазначеним вимогам. Загалом, швидкість, ефективність та точність є головними вимогами до застосунку. Досвід користувача напряму

залежить від якості роботи системи. Після визначення вимог, потрібно визначити покроковий алгоритм дій користувача, який хоче отримати особистісний аналіз кандидата та порівняти його результати з вимогами для вакансії. Для наочності та зрозумілості, було розроблено компонентно-функціональну структуру для користувачів, яка розглядає всі можливі алгоритми дій користувачів для кожного запиту. Повну структуру представлено у вигляді таблиці 3.1.

Таблиця 3.1 – Компонентно-функціональна структура для користувачів

Мета	Задачі	Процедури	Дії користувачів і системи
1	2	3	4
1			Отримати аналіз кандидата
	1.1		Проаналізувати кандидата
		1.1.1	Вести ім'я кандидата
		1.1.2	Завантажити файл з резюме кандидата
		1.1.3	При наявності, завантажити текст співбесіди або перейти на пункт 1.1.4
		1.1.4	Завантажити аудіо або відео файл співбесіди у зазначене для цього поле
		1.1.5	За потреби, завантажити інший файл. У разі вибору завантаження іншого файлу, повернутися в початковий стан.
		1.1.6	Завантажити отриманий особистісний аналіз кандидата у вигляді PDF-файла
2			Отримати порівняння кандидата з вимогами до вакансії
	2.1		Порівняти кандидата з вимогами
		2.1.1	Обрати кандидата зі списку
		2.1.2	Завантажити файл з вимогами до вакансії
		2.1.3	За потреби, завантажити інший файл. У разі вибору завантаження іншого файлу, повернутися в початковий стан.

Після визначення вимог та алгоритмів дій користувача, буде побудовано макет інтерфейсу, який повинен включати в себе виконання всіх потрібних задач. При відвіданні сайту користувач повинен бачити зрозумілий інтерфейс з формою. Користувач буде мати змогу завантажити два файли: резюме кандидата та відео співбесіди яке потім транскрибується у текст. На виході користувач отримує діаграми, які показують результати аналізу особистості за принципом «Big Five», емоційне забарвлення, головний настрій та короткий опис з повним аналізом особистості кандидата. На іншій сторінці застосунку користувач буде мати змогу провести порівняльний аналіз кандидата з вимогами до вакансії. На виході користувач отримає діаграми збіжності та рекомендації на основі результатів.

3.3 Програмна реалізація

Перед початком програмної реалізації було спроектовано, як буде працювати вся система, визначені всі зв'язки між моделями та пропрацьований покроковий алгоритм використання моделей. Діаграму, яка показує алгоритм роботи системи, можна побачити на рисунку 3.6.

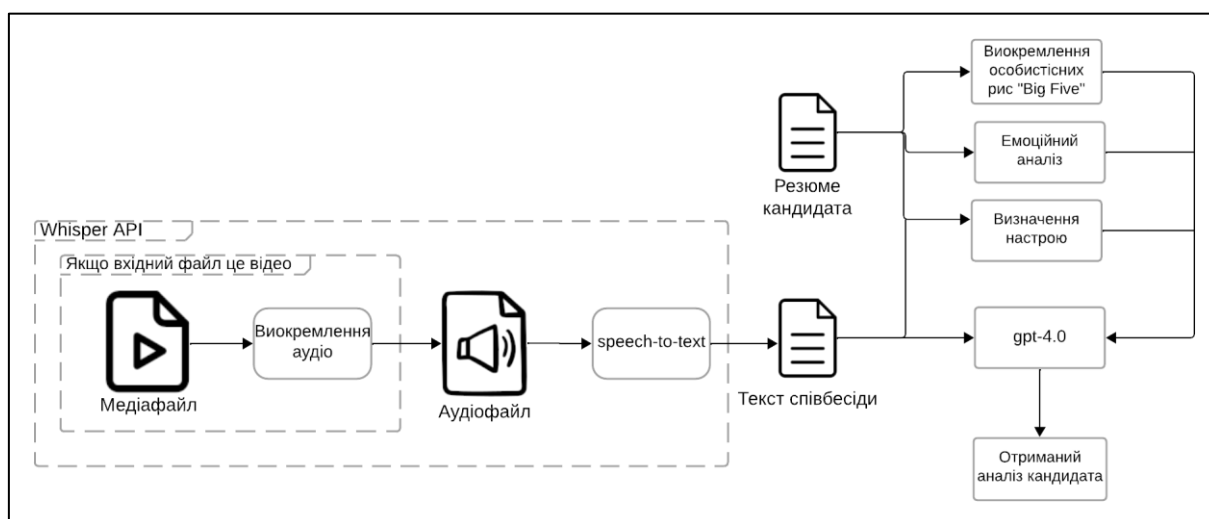


Рисунок 3.6 – Робота системи особистісного аналізу

Таким чином, користувач спочатку завантажує два файли: резюме кандидата у форматі PDF або TXT та відео співбесіди. Відео співбесіди надсилається до моделі Whisper, де з нього виокремлюється аудіо, яке потім транскрибується у файл TXT. Сайт обробляє обидва тексти та надсилає їх на серверну частину для аналізу. Модель HuggingFace «bert-base-personality» використовується для вилучення рис особистості «Big Five». Інші дві моделі «roberta-base-go_emotions» та «twitter-roberta-base-sentiment» використовуються для виявлення основної емоції та настрою. Після цього, отриманий аналіз та вхідні тексти передаються до моделі OpenAI gpt-4.0 для глибшої інтерпретації загального профілю та створення узагальненого опису особистості кандидата.

Для сторінки порівняння кандидатів з вимогами до вакансії також було побудовано діаграму з алгоритмом роботи моделі, яку можна побачити на рисунку 3.7.

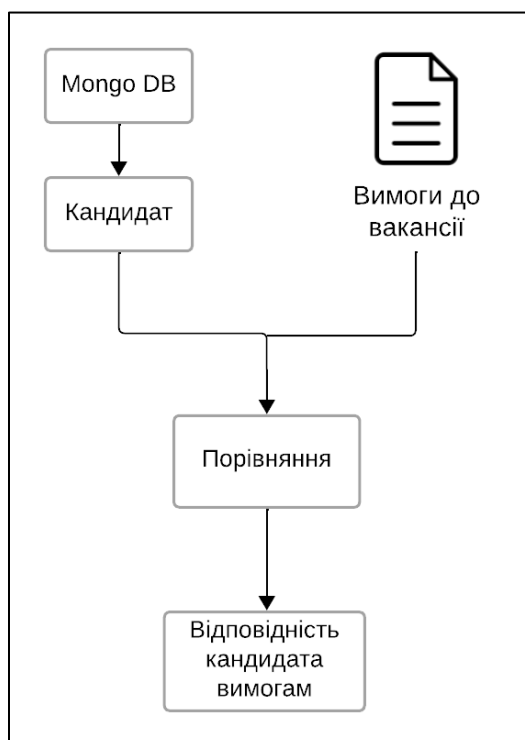


Рисунок 3.7 – Робота системи порівняння кандидата з вимогами

Таким чином, користувач спочатку обирає кандидата, профіль якого зберігається у СУБД MongoDB Cloud. Після цього, користувач має можливість завантажити файл з описом вимог до вакансії у форматі JSON або YAML. Цей профіль включає ідеальні бали за шкалою «Великої п'ятірки», бажані емоційні якості та бажаний стиль спілкування. Система порівнює профіль кандидата з профілем вакансії та показує користувачу, наскільки добре кандидат відповідає вимогам. Результати включають в себе відсоткову відповідність, подібність та відмінності рис «Big Five». Після цього система генерує рекомендації або коментарі, базуючись на отриманих даних. Після визначення алгоритму внутрішньої роботи моделі, постала задача визначитися з середовищем розробки. У межах цього проекту для створення всіх частин застосунку було використано середовище розробки Visual Studio Code (VS Code). Дане середовище ідеально підходить, адже однаково добре підтримує всі мови програмування, які використовувалися. Також VS Code підтримує інтеграцію потрібних фреймворків, розширень та робить роботу з сервером простою та зрозумілою.

На рисунку 3.8 представлено структуру готового застосунку.

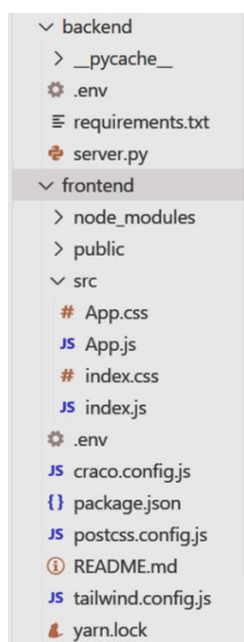


Рисунок 3.8 – Структура проекту

Застосунок складається з двох головних папок які відповідають за серверну та користувацьку частину.

Папка `backend` має таку складову:

- файл `.env` зберігає конфіденційні налаштування, а саме ключі для зв'язку з моделями штучного інтелекту та СУБД;
- файл `server.py` має в собі головну програмну частину, тут відбувається створення серверу, підключення та використання моделей.

Папка `frontend` має таку складову:

- файл `App.js` має в собі головний React-компонент, який збирає інтерфейс застосунку, він імпортує інші компоненти, визначає структуру сторінки та відповідає за візуалізацію результатів аналізу. `App.css` зберігає стилі для компоненту `App.js`;
- `index.js` є точкою входу React-додатка. `index.css` зберігає глобальні стилі застосунку;
- конфігураційний файл CRACO `craco.config.js` дозволяє змінювати налаштування React без "eject";
- файл `package.json` описує застосунок і залежності. Файл `postcss.config.js` має в собі налаштування інструмента для обробки CSS. Файл `tailwind.config.js` зберігає конфігурацію Tailwind CSS. Файл `yarn.lock` фіксує точні версії всіх пакетів, щоб забезпечити стабільність збірки.

Таким чином, застосунок складається з компонентів, які забезпечують стабільну роботу. Завдяки якісному зв'язку, серверна та користувацька частини працюють злагоджено, та видають потрібний результат швидко та точно.

Як було зазначено раніше, файл `server.py` є основною частиною серверної логіки застосунку, створеного на основі фреймворку FastAPI. Перед початком написання логіки застосунку, було підключено низку бібліотек, що забезпечують реалізацію API, обробку файлів, взаємодію з базою даних, інтеграцію зі штучним інтелектом, а також формування звітів і візуалізацій. Основною бібліотекою є FastAPI, яка відповідає за створення

веб-серверу, обробку HTTP-запитів та побудову REST API. Імпортовані класи FastAPI, HTTPException, UploadFile, File і Form використовуються для створення маршрутів, отримання файлів від користувача та передачі даних із форм. Модуль `fastapi.middleware.cors` з класом `CORSMiddleware` забезпечує налаштування політики CORS, що дозволяє фронтенду взаємодіяти з бекендом навіть із різних доменів. Модуль `fastapi.responses` надає класи `JSONResponse` і `FileResponse`, які дозволяють відправляти дані у форматі JSON або повертати файли користувачу, наприклад, готовий звіт у форматі PDF. Для взаємодії з базою даних використовується асинхронний драйвер `Motor`, який реалізовано через модуль `motor.motor_asyncio`. Клас `AsyncIOMotorClient` забезпечує швидке та неблокуюче підключення до бази даних `MongoDB`, що дозволяє серверу обробляти декілька запитів одночасно. Для допоміжних операцій застосовуються стандартні бібліотеки Python, серед яких `os`, `uuid`, `datetime`, `logging`, `io` та `base64`. Вони відповідають за роботу з файловою системою, створення унікальних ідентифікаторів, логування подій, обробку потоків введення-виведення та кодування/декодування даних. Для роботи з моделями штучного інтелекту у файлі імпортуються бібліотеки `OpenAI` та `Requests`. Модуль `openai` забезпечує зв'язок із моделлю `gpt-4.0` яка генерує узагальнений профіль кандидата. Бібліотека `requests` використовується для надсилання HTTP-запитів до моделей `HuggingFace`, що виконують аналіз емоцій та особистісних рис за текстами. Для обробки резюме у форматі PDF застосовується `PyPDF2`, яка дозволяє зчитувати текстові дані без необхідності ручного копіювання. Для обчислювальних і статистичних операцій використовується `NumPy`, що полегшує роботу з числовими масивами, а модуль `tempfile` створює тимчасові файли для проміжних результатів аналізу. Для візуалізації та створення звітів у форматі графіків застосовуються `Matplotlib` та `Seaborn`. Нарешті, бібліотека `ReportLab` забезпечує автоматичне створення PDF-звітів. За допомогою її модулів (`SimpleDocTemplate`, `Paragraph`, `Table`, `TableStyle`) формується

структурований документ із результатами аналізу кандидата, який HR може завантажити безпосередньо із сайту.

Таким чином, набір використаних бібліотек у файлі `server.py` охоплює прийом та аналітичну обробку даних, генерацію графічних результатів та формування готового звіту для користувача, тобто такий набір повністю відповідає вимогам розробки бекенду.

Розглянемо програмну реалізацію частини транскрибування аудіо співбесіди у текст. Частину коду представлено на рисунку 3.9.

```
async def transcribe_audio_with_whisper(audio_content: bytes, filename: str) -> str:
    """Transcribe audio using OpenAI Whisper API"""
    try:
        with tempfile.NamedTemporaryFile(suffix=os.path.splitext(filename)[1], delete=False) as temp_file:
            temp_file.write(audio_content)
            temp_file_path = temp_file.name

        with open(temp_file_path, 'rb') as audio_file:
            transcription = openai_client.audio.transcriptions.create(
                model="whisper-1",
                file=audio_file,
                response_format="text"
            )

        os.unlink(temp_file_path)

    return transcription if isinstance(transcription, str) else transcription.text
```

Рисунок 3.9 – Програмна реалізація транскрибування

Функція `transcribe_audio_with_whisper` є асинхронною і призначена для автоматичного перетворення аудіофайлу у текст за допомогою моделі OpenAI Whisper. Вона приймає байтовий вміст аудіо і назву файлу, щоб визначити його тип. Основна ідея полягає в тому, щоб тимчасово зберегти отримане аудіо, передати його до Whisper API для транскрибування, а потім повернути текстовий результат. На початку функції створюється тимчасовий файл за допомогою бібліотеки `tempfile`, тому що модель Whisper працює не з байтовими потоками, а з реальними файлами. Тимчасовий файл отримує таке саме розширення, як і початковий аудіофайл, а його шлях зберігається у змінній `temp_file_path`. Після цього аудіо записується у цей файл, який потім відкривається у двійковому режимі читання і надсилається

до моделі Whisper через клієнт `openai_client`. В результаті модель повертає транскрибований текст, який зберігається у змінній `transcription`. Після завершення транскрибування тимчасовий файл видаляється, щоб не займати місце у файловій системі.

Після транскрибування, отриманий текст разом з резюме проходить аналіз особистісних рис «Big Five». Фрагмент програмної реалізації даного процесу представлено на рисунку 3.10.

```
async def extract_big_five_traits(text: str) -> BigFiveTraits:
    """Отримати Big Five риси характеру з моделі Hugging Face"""
    try:
        headers = {"Authorization": f"Bearer {HUGGINGFACE_API_KEY}"}
        api_url = "https://api-inference.huggingface.co/models/Minej/bert-base-personality"

        input_text = text[:1700]
        payload = {"inputs": input_text}

        response = requests.post(api_url, headers=headers, json=payload)

        if response.status_code == 200:
            result = response.json()
            logger.info(f"Hugging Face personality analysis result: {result}")

            if isinstance(result, list) and len(result) > 0 and isinstance(result[0], list):
                scores = {item['label']: item['score'] for item in result[0]}

                label_mapping = {
                    'LABEL_0': 'openness',
                    'LABEL_1': 'conscientiousness',
                    'LABEL_2': 'extraversion',
                    'LABEL_3': 'agreeableness',
                    'LABEL_4': 'neuroticism',
                }
    }
```

Рисунок 3.10 – Програмна реалізація аналізу «Big Five»

Метод `extract_big_five_traits` спочатку формує запит до API HuggingFace, а саме до моделі Minej/bert-base-personality. Для цього створюється словник `headers`, у якому міститься токен авторизації API HuggingFace, і визначається змінна `api_url` з адресою моделі. Потім формується тіло запиту, яке передається через HTTP POST-запит за допомогою бібліотеки `requests`. Якщо зв'язок з сервером успішний, результат запиту зчитується у форматі JSON, і з нього вилачуються оцінки

за кожною рисою особистості. Для зручності створюється словник `label_mapping`, який перетворює службові позначення моделі у зрозумілі риси: відкритість (`openness`), сумлінність (`conscientiousness`), екстраверсія (`extraversion`), згодливість (`agreeableness`) та невротизм (`neuroticism`). Потім ці оцінки записуються у словник `traits`. У кінці ці значення передаються до класу `BigFiveTraits`, який формує об'єкт з усіма п'ятьма характеристиками.

Аналіз домінантної емоції та настрою реалізується в одному методі, фрагмент якого представлено на рисунку 3.11.

```

async def analyze_emotions_and_sentiment(text: str) -> EmotionAnalysis:
    """Analyze emotions and sentiment using Hugging Face models"""
    try:
        headers = {"Authorization": f"Bearer {HUGGINGFACE_API_KEY}"}

        emotion_api_url = "https://api-inference.huggingface.co/models/SamLowe/roberta-base-go_emotions"
        emotion_payload = {"inputs": text[:512]}
        emotion_response = requests.post(emotion_api_url, headers=headers, json=emotion_payload)

        sentiment_api_url = "https://api-inference.huggingface.co/models/cardiffnlp/twitter-roberta-base-sentiment"
        sentiment_payload = {"inputs": text[:512]}
        sentiment_response = requests.post(sentiment_api_url, headers=headers, json=sentiment_payload)

        emotion_scores = {}
        dominant_emotion = "neutral"
        sentiment = "neutral"
        sentiment_score = 0.5

```

Рисунок 3.11 – Перша частина програмної реалізації аналізу емоцій та настрою

Метод `analyze_emotions_and_sentiment` виконує аналіз емоцій і настрою за допомогою моделей HuggingFace.

На початку функції створюються заголовки запиту, що містять токен авторизації. Далі відправляються два незалежні запити. Перший запит відправляється до моделі `SamLowe/roberta-base-go_emotions`, яка класифікує текст за різними типами емоцій. Другий відправляється до моделі `cardiffnlp/twitter-roberta-base-sentiment`, що визначає загальний настрій.

Після цього ініціалізуються змінні за замовчуванням:

- `emotion_scores`, що є порожнім словником для збереження оцінок усіх емоцій;

- `dominant_emotion` визначає домінуючу емоцію;
- `sentiment`, визначає загальний настрій тексту;
- `sentiment_score` є числовою оцінкою сентименту.

Наступну частину даного методу представлено на рисунку 3.12.

```

if emotion_response.status_code == 200:
    emotion_data = emotion_response.json()
    logger.info(f"Hugging Face emotion analysis result: {emotion_data}")
    if isinstance(emotion_data, list) and len(emotion_data) > 0:
        for item in emotion_data[0]:
            emotion_scores[item['label']] = item['score']
        dominant_emotion = max(emotion_scores, key=emotion_scores.get)

if sentiment_response.status_code == 200:
    sentiment_data = sentiment_response.json()
    logger.info(f"Hugging Face sentiment analysis result: {sentiment_data}")
    if isinstance(sentiment_data, list) and len(sentiment_data) > 0:
        best_sentiment = max(sentiment_data[0], key=lambda x: x['score'])
        label_map = {
            'LABEL_0': 'negative',
            'LABEL_1': 'neutral',
            'LABEL_2': 'positive'
        }
        sentiment = label_map.get(best_sentiment['label'], best_sentiment['label'])
        sentiment_score = best_sentiment['score']

```

Рисунок 3.12 – Друга частина програмної реалізації аналізу емоцій та настрою

Далі відбувається обробка результатів обох запитів. Якщо запит до емоційної моделі завершився успішно, програма розбирає JSON-відповідь, де кожен елемент містить назву емоції і її ймовірність. Ці пари додаються до словника `emotion_scores`, після чого функція визначає, яка емоція має найбільший бал та стає `dominant_emotion`.

Після цього обробляється відповідь другої моделі, яка аналізує настрої. Якщо запит успішний, із результатів вибирається клас із найбільшою ймовірністю. Оскільки модель повертає службові мітки, створюється словник `label_map`, який перетворює їх у зрозумілі текстові назви: `negative`, `neutral` або `positive`. Після цього `sentiment` отримує текстову назву настрою, а `sentiment_score` отримує його числове значення.

У кінці функція повертає об'єкт EmotionAnalysis, що містить основні результати: домінуючу емоцію, усі емоційні оцінки, тип настрою та його інтенсивність.

Після отримання всього базового аналізу, отримані результати та вхідні дані відправляються на інтерпретацію до моделі gpt-4.0. Фрагмент програмної реалізації представлено на рисунку 3.13.

```

async def generate_comprehensive_analysis(resume_text: str, interview_transcript: str,
                                         big_five: BigFiveTraits, emotions: EmotionAnalysis) -> str:
    """Generate comprehensive personality analysis using OpenAI ChatGPT"""
    try:
        prompt = f"""
        Як експерт-психолог з управління персоналом, проаналізуй профіль цього кандидата та надай комплексну оцінку

        Зміст резюме:
        {resume_text[:1000]}...

        Зміст співбесіди:
        {interview_transcript[:1000]}...

        Особистісні риси Big Five:
        - Відкритість: {big_five.openness:.2f}
        - Сумлінність: {big_five.conscientiousness:.2f}
        - Екстраверсія: {big_five.extraversion:.2f}
        - Згодливість: {big_five.agreeableness:.2f}
        - Невротизм: {big_five.neuroticism:.2f}

        Емоційний аналіз:
        - Домінуюча емоція: {emotions.dominant_emotion}
        - Настрій: {emotions.sentiment} (оцінка: {emotions.sentiment_score:.2f})
        """
    
```

Рисунок 3.13 – Програмна реалізація інтеграції моделі gpt-4.0

Метод `generate_comprehensive_analysis` відповідає за створення комплексного психологічного звіту про кандидата, використовуючи модель OpenAI gpt-4.0. На початку функції формується змінна `prompt`, яка є детальною інструкцією для gpt-4.0. Вона містить опис ролі і перелік усіх даних, які потрібно врахувати при аналізі. У `prompt` включаються тексти резюме та співбесіди. Потім наводяться числові значення п'яти основних рис особистості за «Big Five» та результати емоційного аналізу, що включають в себе домінуючу емоцію, загальний настрій і його оцінку.

У запиті чітко сформульовано, що модель повинна надати шість основних блоків інформації:

- короткий виклад резюме та співбесіди;

- загальний опис особистості;
- сильні сторони та зони розвитку;
- стиль спілкування і соціальні навички;
- уподобання щодо роботи та фактори мотивації;
- потенціал кандидата.

Таким чином, функція створює контекстно багатий запит, який забезпечує змістовну та практичну аналітику, придатну для HR-рішень. Далі викликається метод, який надсилає запит до моделі gpt-4.0. Також задаються параметри генерації, що визначають максимальну кількість токенів, що обмежує довжину відповіді. Також виставлено температуру 0.7, що визначає рівень творчості моделі та забезпечує баланс між точністю та варіативністю. Якщо запит проходить успішно, функція повертає згенерований текстовий звіт.

Загалом, ця функція є ключовою аналітичною частиною системи, адже саме вона перетворює числові результати моделей і текстові дані на цілісний, зрозумілий і професійно оформлений психологічний профіль кандидата.

Наступним було реалізовано метод порівняння результатів аналізу кандидата з вимогами до вакансії. Фрагмент програмної реалізації представлено на рисунку 3.14.

```
def calculate_job_match(candidate: CandidateProfile, job_req: JobRequirements) -> ComparisonResult:
    """Calculate how well candidate matches job requirements"""
    try:
        big_five_comparison = {}
        big_five_scores = []

        candidate_traits = candidate.big_five_traits.dict()
        ideal_traits = job_req.ideal_big_five.dict()

        for trait, candidate_score in candidate_traits.items():
            ideal_score = ideal_traits[trait]
            difference = abs(candidate_score - ideal_score)
            match_score = max(0, 1 - difference)
            big_five_comparison[trait] = {
                'candidate': candidate_score,
                'ideal': ideal_score,
                'match_score': match_score
            }
            big_five_scores.append(match_score)
```

Рисунок 3.14 – Програмна реалізація порівняння

Метод `calculate_job_match` використовує результати аналізу особистості «Big Five» і емоційний профіль, обчислюючи загальний відсоток відповідності та формуючи рекомендації для HR-фахівця. На початку у функції створюються порожні словники `big_five_comparison` і список `big_five_scores`, які зберігатимуть результати порівняння за кожною рисою «Big Five». Із профілю кандидата та з опису вакансії отримуються відповідні п'ять рис.

Далі функція проходить циклом по кожній рисі особистості, порівнюючи оцінку кандидата із ідеальною для вакансії. Обчислюється абсолютна різниця між ними, а потім ця різниця перетворюється у коефіцієнт відповідності, де 1 означає повну відповідність, а 0 означає повну невідповідність. Результати для кожної риси зберігаються у словнику `big_five_comparison`, а значення коефіцієнтів додаються до списку `big_five_scores`. Після завершення циклу розраховується загальна відповідність за моделлю Big Five як середнє значення, що виражається у відсотках. Далі аналізується емоційна відповідність. Після чого формується словник `emotion_match` із детальною інформацією, що включає в себе оцінку, назву емоції кандидата та загальний настрій.

Після цього обчислюється загальний показник відповідності та формується список рекомендацій. Якщо якась риса не відповідає вимогам, система додає рекомендацію щодо розвитку цієї риси, або у разі її високого значення рекомендує що варто використовувати цю сильну сторону на посаді.

Після цього створюються фінальні рекомендації залежно від рівня загальної відповідності:

- якщо збіг 80% і більше система відповідає «Відмінна відповідність, рекомендовано до працевлаштування»;
- якщо збіг 60–79% система відповідає «Придатний кандидат, рекомендовано до розгляду»;

– якщо збіг менше 60% система відповідає «Недостатня відповідність, потрібна додаткова оцінка».

Функція повертає об'єкт `ComparisonResult`, що містить усі розраховані дані, а саме ідентифікатор кандидата, опис вакансії, загальний відсоток відповідності, детальний розбір «Big Five», емоційний аналіз і рекомендації.

Таким чином, ця функція виступає інтелектуальним ядром порівняльного модуля системи, автоматично поєднуючи психологічний та емоційний аналіз для формування об'єктивної оцінки придатності кандидата до конкретної посади.

Для забезпечення комфортної та зрозумілої взаємодії користувача з системою, було створено інтерфейси користувача. Перша сторінка є початковою, на ній користувач може ввести дані про кандидата. Зверху сторінка має кнопки для переходу на сторінки результатів аналізу та проведення порівняльного аналізу з вимогами. У основному блоці знаходяться поля для вводу інформації. Спочатку користувач вводить ім'я кандидата, в поле нижче додається резюме у форматі PDF або TXT. Після цього, користувач завантажує аудіо чи відео співбесіди або, якщо є, можна вставити вже транскрибований текст. Після завантаження всієї потрібної інформації користувач натискає на кнопку, яка відповідає за початок аналізу. Повний дизайн сторінки представлено на рисунку 3.15.

Рисунок 3.15 – Вигляд початкової сторінки

Після успішного виконання аналізу, система автоматично переводить користувача на сторінку з результатами. В правому верхньому куті користувач може одразу завантажити отримані результати у вигляді PDF-файлу. Нижче на сторінці відзначено ім'я кандидата та час аналізу. Після цього представлено графіки, на яких наочно показано результати особистісного аналізу за «Big Five». Ще нижче зазначено результати емоційного аналізу та написано домінуючу емоцію та загальний настрій. Вигляд цієї частини результатів представлено на рисунку 3.16.

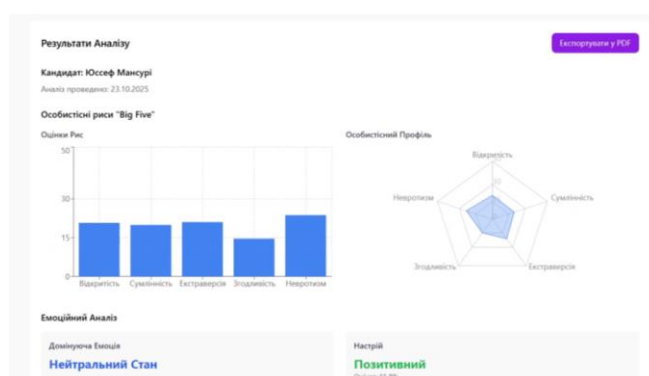


Рисунок 3.16 – Вигляд сторінки з результатами

На рисунку 3.17 представлено процес зв'язку з моделями та передачу числових результатів до gpt-4.0 для інтерпретації.

```
INFO:httpx:HTTP Request: POST https://api.openai.com/v1/audio/transcriptions "HTTP/1.1 200 OK"
INFO:server:Hugging Face personality analysis result: [{"label": 'LABEL_4', 'score': 0.23695947229862213}, {"label": 'LABEL_2', 'score': 0.21036599576473236}, {"label": 'LABEL_0', 'score': 0.20745722949504852}, {"label": 'LABEL_1', 'score': 0.19917692244052887}, {"label": 'LABEL_3', 'score': 0.14604032039642334}]]
INFO:server:Hugging Face emotion analysis result: [{"label": 'neutral', 'score': 0.7962521314620972}, {"label": 'admiration', 'score': 0.13996216654777527}, {"label": 'approval', 'score': 0.13800738751888275}, {"label": 'realization', 'score': 0.010156159289181232}, {"label": 'optimism', 'score': 0.007229850627481937}]]
INFO:server:Hugging Face sentiment analysis result: [{"label": 'LABEL_2', 'score': 0.5579587817192078}, {"label": 'LABEL_1', 'score': 0.4337258040904999}, {"label": 'LABEL_0', 'score': 0.008315440267324448}]]
INFO:httpx:HTTP Request: POST https://api.openai.com/v1/chat/completions "HTTP/1.1 200 OK"
INFO: 127.0.0.1:61316 - "POST /api/analyze-candidate HTTP/1.1" 200 OK
INFO: 127.0.0.1:61316 - "GET /api/candidates HTTP/1.1" 200 OK
INFO: 127.0.0.1:61365 - "POST /api/compare-with-job HTTP/1.1" 200 OK
```

Рисунок 3.17 – Вигляд сторінки з результатами

Нижче користувач побачить комплексний аналіз, який включає в себе аналіз резюме, співбесіди та результати аналізу особистості (рисунок 3.18).

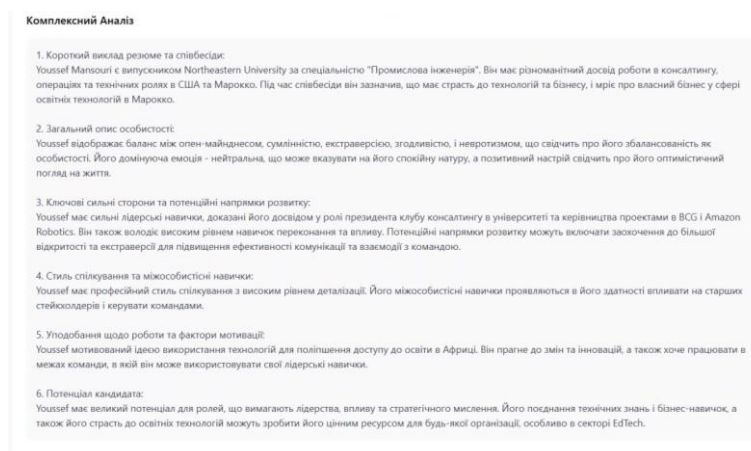


Рисунок 3.18 – Продовження сторінки з результатами

Наступною сторінкою є порівняльний аналіз результатів кандидата та вимог до вакансії. Спершу користувач обирає зі списку ім'я потрібного кандидата. Після цього користувач завантажує файл з вимогами та натискає на кнопку порівняння. Вигляд сторінки представлено на рисунку 3.19.

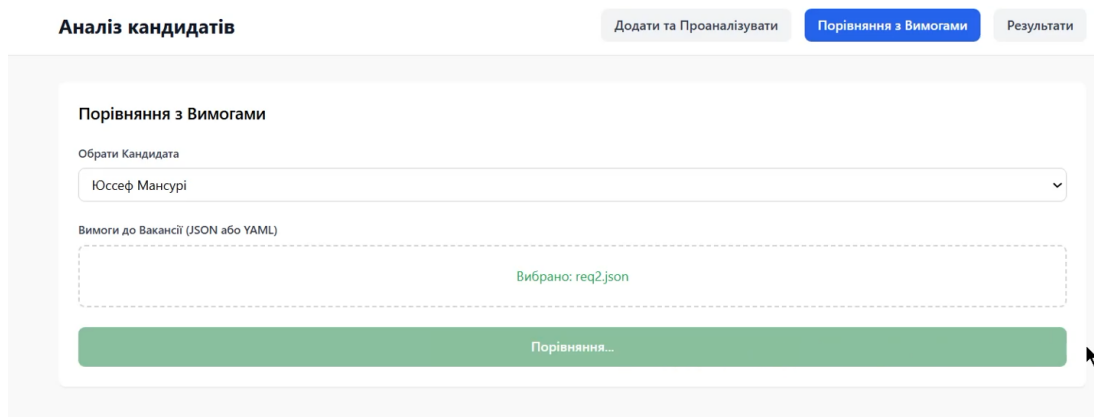


Рисунок 3.19 – Вигляд сторінки порівняльного аналізу

Після успішного порівняння, система видає відсоток загального збігу та гістограму, яка показує бажаний відсоток риси, та відсоток риси

кандидата. Із отриманого порівняння, система пише рекомендації. Вигляд сторінки представлено на рисунку 3.20.

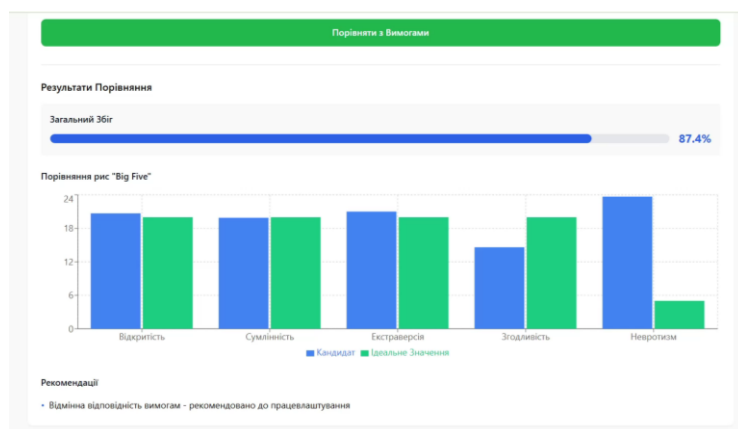


Рисунок 3.20 – Продовження сторінки порівняльного аналізу

У випадку невідповідності кандидата вимогам, система дає рекомендації, над якими рисами потрібно попрацювати кандидату, та рекомендує HR переглянути рішення щодо цього кандидата (рисунок 3.21).

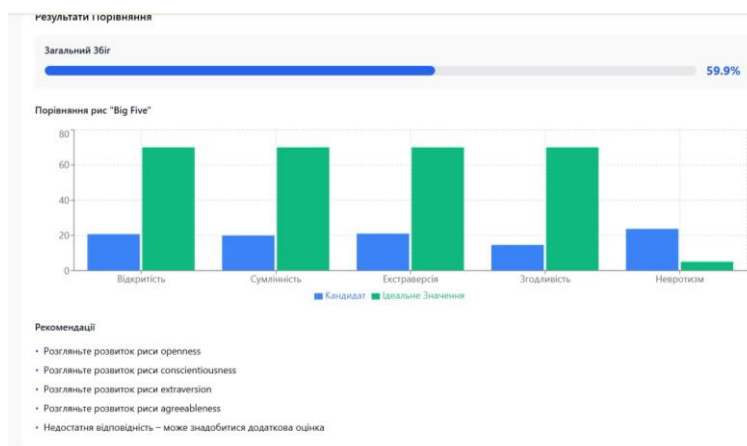


Рисунок 3.21 – Випадок непридатного кандидата

Таким чином, було розроблено повністю робочий та комфортний у використанні веб-застосунок для підтримки HR-рішень засновуючись на автоматичному аналізі особистості кандидата.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи, було розглянуто та досліджено проблему перенасичення ринку праці, в зв'язку з чим фахівцям з управління персоналом потрібно обробляти великі обсяги різних резюме та співбесід. Було запропоновано систему, яка буде автоматично обробляти тексти резюме та транскрибований текст проведених співбесід, після чого система буде проводити аналіз особистості та на виході HR-фахівець отримає повний особистісний профіль кандидата, який потім можна буде порівняти з вимогами до вакансії.

У першому розділі було проведено детальний аналіз предметної галузі. Дослідження включало в себе визначення концепту текстового аналізу, порівняння людського текстового аналізу з автоматичним та сфокусовано на задачі аналізу особистості. Після цього вивчено основні відомості про роботу систем розпізнавання мовлення, які будуть використані в ході розробки фінальної системи. Було досліджено існуючі рішення та визначено, що наразі не існує подібного застосунку, тому його розробка є актуальною.

В другому розділі було детально проаналізовано та порівняно існуючі підходи та методи до задач текстового аналізу, аналізу особистості та розпізнавання мовлення, після чого було обрано оптимальний варіант для подальшої роботи.

В третьому розділі було обрано відповідні технології, що дозволили розробити якісний застосунок, визначено вимоги до спроектовано систему, для якої потім було написано логіку та інтерфейс користувача, який в результаті було протестовано.

В результаті всі поставлені задачі було виконано. В перспективі розвитку буде покращення якості особистісного аналізу, адже галузь машинного навчання постійно розвивається, а також додавання додаткового функціоналу, наприклад у вигляді аналізу міміки [43].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. How many applications does it take to get one interview? URL: <https://www.interviewpal.com/blog/how-many-applications-does-it-take-to-get-one-interview> (date of access: 15.11.2025).
2. Practical text mining and statistical analysis for non-structured text data applications / G. Miner et al. Academic Press, 2012.
3. What is text analytics? URL: <https://www.geeksforgeeks.org/data-science/what-is-text-analytics/> (date of access: 16.10.2025).
4. Roberts B. W., Mroczek D. Personality trait change in adulthood. *Current directions in psychological science*. 2008. Vol. 17, no. 1. P. 31–35.
5. Costa P. T., McCrae R. R. The revised NEO personality inventory (NEO-PI-R). *The SAGE handbook of personality theory and assessment: volume 2 – personality measurement and testing*. 1 Oliver's Yard, 55 City Road, London EC1Y 1SP United Kingdom. P. 179–198.
6. Myers P. B., Myers I. B. Gifts differing: understanding personality type. 2nd ed. Consulting Psychologists Pr, 1993. 224 p.
7. Kirvan P., Lutkevich B., Kiwak K. What is speech recognition? URL: <https://www.techtarget.com/searchcustomerexperience/definition/speech-recognition> (date of access: 15.11.2025).
8. Rath T. M., Manmatha R. Word spotting for historical documents. *International journal of document analysis and recognition (IJ DAR)*. 2006. Vol. 9, no. 2-4. P. 139–152.
9. Text mining methodology. *Practical text mining and statistical analysis for non-structured text data applications*. 2012. P. 73–89. URL: <https://doi.org/10.1016/b978-0-12-386979-1.00005-0> (date of access: 03.12.2025).
10. Sebastiani F. Classification of text, automatic. *Encyclopedia of language & linguistics*. 2006. P. 457–462.

11. Jackson P., Schilder F. Natural language processing: overview. *Encyclopedia of language & linguistics*. 2006. P. 503–518.
12. Meunier J. G., Forest D., Biskri I. Classification and categorization in computer-assisted reading and text analysis. *Handbook of categorization in cognitive science*. 2005. P. 955–978.
13. Capuano N. Transfer learning techniques for cross-domain analysis of posts in massive educational forums. *Intelligent systems and learning data analytics in online education*. 2021. P. 133–152.
14. Blei D. M. Probabilistic topic models. *Communications of the ACM*. 2012. Vol. 55, no. 4. P. 77–84.
15. Kherwa P., Bansal P. Topic modeling: a comprehensive review. *ICST transactions on scalable information systems*. 2018. P. 159623. URL: <https://doi.org/10.4108/eai.13-7-2018.159623> (date of access: 03.12.2025).
16. Stryker C. What are large language models (llms)? | IBM. *IBM*. URL: <https://www.ibm.com/think/topics/large-language-models> (date of access: 03.12.2025).
17. Training and evaluating multimodal word embeddings with large-scale web annotated images / J. Mao et al. *NIPS'16: proceedings of the 30th international conference on neural information processing systems*, Barcelona, 5 December 2016. P. 442–450.
18. Efficient estimation of word representations in vector space / T. Mikolov et al. *International conference on learning representations*, 2013.
19. Pennington J., Socher R., Manning C. Glove: global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar. Stroudsburg, PA, USA, 2014. P. 1532–1543. URL: <https://doi.org/10.3115/v1/d14-1162> (date of access: 03.12.2025).
20. BERT: pre-training of deep bidirectional transformers for language understanding / J. Devlin et al. *Proceedings of the 2019 conference of the north,*

Minneapolis, Minnesota. Stroudsburg, PA, USA, 2019.

URL: <https://doi.org/10.18653/v1/n19-1423> (date of access: 03.12.2025).

21. Improving language understanding by generative pre-training / A. Radford et al. 2018.

22. Pekel E., Soner Kara S. A comprehensive review for artificial neural network application to public transportation. *Sigma journal of engineering and natural sciences*. 2017. Vol. 35. P. 157–179.

23. Empirical evaluation of gated recurrent neural networks on sequence modeling / J. Chung et al. *Deep learning and representation learning*. 2014.

24. Time-Lag selection for time-series forecasting using neural network and heuristic algorithm / O. Surakhi et al. *Electronics*. 2021. Vol. 10, no. 20. P. 2518. URL: <https://doi.org/10.3390/electronics10202518> (date of access: 03.12.2025).

25. Evaluation of three deep learning models for early crop classification using sentinel-1a imagery time series—a case study in zhanjiang, china / H. Zhao et al. *Remote sensing*. 2019. Vol. 11, no. 22. P. 2673. URL: <https://doi.org/10.3390/rs11222673> (date of access: 03.12.2025).

26. Deep sequence models for text classification tasks / S. S. Abdullahi et al. *2021 international conference on electrical, communication, and computer engineering (ICECCE)*, Kuala Lumpur, Malaysia, 12–13 June 2021. 2021. URL: <https://doi.org/10.1109/icecce52056.2021.9514261> (date of access: 03.12.2025).

27. Attention is all you need / A. Vaswani et al. *Proceedings of the 31st conference on neural information processing systems*, Long Beach, CA, 4–9 December 2017. 2017.

28. Pennebaker J. W., Francis M. E. Cognitive, emotional, and language processes in disclosure. *Cognition and emotion*. 1996. Vol. 10, no. 6. P. 601–626. URL: <https://doi.org/10.1080/026999396380079> (date of access: 03.12.2025).

29. Wilson M. MRC psycholinguistic database: machine-usable dictionary, version 2.00. *Behavior research methods, instruments, & computers*. 1988. Vol. 20, no. 1. P. 6–10.

30. Minej/bert-base-personality · hugging face. *Hugging Face – The AI community building the future*. URL: <https://huggingface.co/Minej/bert-base-personality> (date of access: 03.12.2025).

31. RoBERTa: A robustly optimized BERT pretraining approach / Y. Liu et al. 2019.

32. SamLowe/roberta-base-go_emotions · hugging face. *Hugging Face – The AI community building the future*. URL: https://huggingface.co/SamLowe/roberta-base-go_emotions (date of access: 03.12.2025).

33. Gales M., Young S. The application of hidden markov models in speech recognition. *Foundations and trends® in signal processing*. 2007. Vol. 1, no. 3. P. 195–304. URL: <https://doi.org/10.1561/2000000004> (date of access: 03.12.2025).

34. Analysis of automatic speech recognition methods / O. Iosifova et al. *Cybersecurity providing in information and telecommunication systems*, Kyiv, 28 January 2021. P. 252–257.

35. Connectionist temporal classification / A. Graves et al. *The 23rd international conference*, Pittsburgh, Pennsylvania, 25–29 June 2006. New York, New York, USA, 2006. URL: <https://doi.org/10.1145/1143844.1143891> (date of access: 03.12.2025).

36. Papastratis I. Speech Recognition: a review of the different deep learning approaches | AI Summer. *AI Summer*. URL: <https://theaisummer.com/speech-recognition/#fn-20> (date of access: 03.12.2025).

37. Introducing Whisper URL: <https://openai.com/index/whisper/> (date of access: 20.05.2024).

38. Robust speech recognition via large-scale weak supervision / A. Radford et al. *Proceedings of the 40th international conference on machine learning*. 2023. P. 28492–28518.

39. Stryker C. What is hugging face? | IBM. *IBM*. URL: <https://www.ibm.com/think/topics/hugging-face> (date of access: 03.12.2025).

40. Cardiffnlp/twitter-roberta-base-sentiment · hugging face. *Hugging Face – The AI community building the future*. URL: <https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment> (date of access: 03.12.2025).

41. MongoDB: the world's leading modern database. *MongoDB*. URL: <https://www.mongodb.com/> (date of access: 03.12.2025).

42. MongoDB cloud. *MongoDB*. URL: <https://www.mongodb.com/products/platform/cloud> (date of access: 03.12.2025).

43. Павленко О., Чала Л. Технології розробки мультимодальної системи підтримки hr-рішень для оцінки кандидатів на основі аналізу особистісних характеристик. *Інформаційно-комунікаційні технології та кібербезпека*: Матеріали Міжнародної науково-технічної конференції, м. Харків, 4–5 груд. 2025 р. С. 101–105.