

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА

Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

Кафедра системного аналізу та кібербезпеки

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «СИСТЕМНИЙ АНАЛІЗ»

Галузь знань 12 «Інформаційні технології»

Спеціальність 124 «Системний аналіз»

Форма навчання: очна (денна)
очна (денна), заочна, дистанційна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему:

«Розробка алгоритмів для оцінювання навичок кандидатів на посади ІТ-фахівців з використанням SMART-аналізу»

здобувача Жука Дмитра Валентиновича
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник д.ф.-м.н., проф. Джалладова І.А.
(науковий ступінь, учене звання, ПІБ)

(підпис)

Робота допущена до захисту перед екзаменаційною комісією з атестації здобувачів вищої освіти (ЕК)

Завідувач кафедри: д.ф.-м.н., проф. Джалладова І.А.

(підпис)

Київ 20 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА

Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

Кафедра системного аналізу та кібербезпеки

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «СИСТЕМНИЙ АНАЛІЗ»

СПЕЦІАЛЬНІСТЬ 124 «СИСТЕМНИЙ АНАЛІЗ»

ПОГОДЖЕНО

Керівник проєктної групи (гарант)
освітньо-професійної програми

_____ Галіцин В.К.

_____ 2024 р.

ЗАТВЕРДЖУЮ

Завідувач кафедри
системного аналізу та кібербезпеки

_____ Джалладова І.А.

_____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувачу вищої освіти Жуку Дмитру Валентиновичу

очної (денної) форми навчання

на підготовку кваліфікаційної бакалаврської роботи

на тему: «Розробка алгоритмів для оцінювання навичок кандидатів на посади ІТ-фахівців з

використанням SMART-аналізу»

Тему затверджено наказом ректора Університету від «___» _____ 20__ р. № _____

Кваліфікаційна бакалаврська робота виконується на матеріалах «Whalebone s.r.o.»

План кваліфікаційної бакалаврської роботи

Розділ 1	Трансформація процесів управління людськими ресурсами за допомогою SMART-аналізу
Розділ 2	Аналіз алгоритмів для оцінювання навичок кандидатів
Розділ 3	Створення алгоритмів для ефективного оцінювання навичок кандидатів на посади ІТ-фахівців

Об'єкт дослідження:	Процес підбору та оцінювання навичок кандидатів на посади ІТ-фахівців
Предмет дослідження:	Методи та інструменти SMART-аналізу для оцінювання навичок кандидатів на посади ІТ-фахівців
Мета кваліфікаційної бакалаврської роботи:	Обґрунтування теоретичних і прикладних засад алгоритмів для оцінювання навичок кандидатів на посади ІТ-фахівців за допомогою SMART-аналізу для підбору ідеального кандидата на вакантну посаду ІТ-фахівця з урахуванням специфіки ІТ-індустрії

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі 1	Визначити сутність та роль SMART-аналізу як одного з сучасних підходів управління людськими ресурсами для оцінювання навичок кандидатів на посади ІТ-фахівців
У розділі 2	Охарактеризувати можливі підходи для оцінювання навичок кандидатів на посади ІТ-фахівців Скласти порівняльну характеристику методів оцінювання навичок кандидатів на посади ІТ-фахівців
У розділі 3	Розробити алгоритми для ефективного оцінювання навичок кандидатів на посади ІТ-фахівців та підбору ідеального кандидата Оцінити точність алгоритмів з урахуванням специфіки ІТ-індустрії

**Завдання підготував
науковий керівник**

(підпис)

Джалладова І.А.

(ініціали, прізвище)

«____» _____ 2024 р.

**Завдання одержав
здобувач**

(підпис)

Жук Д.В.

(ініціали, прізвище)

«____» _____ 2024 р.

Реферат

Кваліфікаційна бакалаврська робота містить 86 сторінок, 4 таблиці, 36 рисунків, список літератури з 91 найменування, 1 додаток.

«Розробка алгоритмів для оцінювання навичок кандидатів на посади ІТ-фахівців з використанням SMART-аналізу»

Об'єктом дослідження є процес підбору та оцінювання навичок кандидатів на посади ІТ-фахівців.

Предметом дослідження є методи та інструменти SMART-аналізу для оцінювання навичок кандидатів на посади ІТ-фахівців.

Метою дослідження є обґрунтування теоретичних і прикладних засад алгоритмів для оцінювання навичок кандидатів за допомогою SMART-аналізу для підбору ідеального кандидата на вакантну посаду з урахуванням специфіки ІТ-індустрії.

Мета дослідження розкривається переліком таких завдань дослідження:

- Охарактеризувати можливі підходи для оцінювання навичок кандидатів;
- Визначити сутність та роль SMART-аналізу як одного з сучасних підходів управління людськими ресурсами;
- Скласти порівняльну характеристику методів оцінювання навичок кандидатів;
- Розробити алгоритми для підбору ідеального кандидата, враховуючи та забезпечуючи об'єктивність та нейтральність процесу оцінювання, оцінити їх точність.

Практична цінність результатів дослідження полягає у покращенні ефективності та точності процесу відбору кандидатів. Зокрема, за результатами дослідження, використання автоматизованих технологій за допомогою SMART-аналізу поліпшує процес підбору кандидатів, забезпечуючи відповідність їхніх навичок потребам бази практики Whalebone s.r.o. Також практичні та методичні розробки були розглянуті керівництвом бази практики та прийняті до апробації.

Рік виконання кваліфікаційної бакалаврської роботи 2024.

Рік захисту кваліфікаційної бакалаврської роботи 2024.

Ключові слова: SMART-аналіз, автоматизація, штучний інтелект, машинне навчання, кібербезпека, підбір кандидатів, оцінка навичок, ІТ-індустрія.

ВІДГУК

на кваліфікаційну бакалаврську роботу

студента **Жука Дмитра Валентиновича**, групи ІА-401

на тему «Розробка алгоритмів для оцінювання навичок кандидатів на посади ІТ-фахівців з використанням SMART-аналізу»

Актуальність теми: актуальність даного дослідження обумовлена зростанням попиту на використання новітніх технологій та аналітичних інструментів для ефективного управління людськими ресурсами у технологічних стартапах, компаніях середнього та великого розміру, корпораціях, що активно використовують інформаційні технології для своєї діяльності по всьому світу. Зокрема, для України, Європи, Азії тощо, де наукові публікації у сфері оцінки навичок кандидатів відіграють важливу роль у формуванні стратегій підбору персоналу. Дослідження взаємозв'язку між розвитком іноваційного напрямку SMART-аналізу та його використанням в автоматизованих системах для ефективної оцінки навичок кандидатів дозволяє не лише розуміти динаміку росту SMART-аналізу як унікального мультидисциплінарного явища, але й розуміти та вирішувати практичні задачі управління людськими ресурсами. Це має важливе значення для розробки нових та вдосконалення існуючих інтернаціональних систем, підвищення кваліфікації фахівців, а також розробки заходів для ефективного автоматизованого підбору кандидатів на посади ІТ-фахівців, зберігаючи об'єктивність та нейтральність процесу. В умовах сучасного світу, де необхідно мати об'єктивні, точні та ефективні методи оцінювання навичок кандидатів, щоб забезпечити високу якість кадрового забезпечення, такі дослідження набувають особливої важливості.

Повнота розкриття теми: структура роботи та викладення матеріалу характеризуються логічністю та послідовністю. Вимоги до написання кожного розділу дотримані. Позитивним у роботі є те, що студентом було розроблено алгоритм для ефективного оцінювання навичок кандидатів на посади ІТ-фахівців для підбору

ідеального кандидата, враховуючи та забезпечуючи об'єктивність та нейтральність процесу оцінювання. Таким чином, тема дослідження була розкрита повно і всебічно, що сприяє поглибленому розумінню та покращенню управління людськими ресурсами на національному та міжнародному рівнях.

Теоретичний рівень: дослідженням визначено сутність та роль SMART-аналізу як одного з сучасних підходів управління людськими ресурсами, що відображає рівень розвитку, сприйняття та впровадження інновацій в сфері оцінки навичок кандидатів та ІТ-індустрії загалом. Також, було охарактеризовано можливі підходи для оцінювання навичок кандидатів на посади ІТ-фахівців, складено порівняльну характеристику методів та їхню ефективність для підбору персоналу в умовах воєнного стану та цифрової трансформації.

Практична значущість: запропоновано конкретні рекомендації щодо використання автоматизованих технологій, включаючи впровадження алгоритму для ефективного оцінювання навичок кандидатів на посади ІТ-фахівців для підбору ідеального кандидата на основі напряму SMART-аналізу, підвищення кваліфікації фахівців, інвестиції у дослідження та розробку нормативно-правової бази. Результати дослідження орієнтовані на використання в інтернаціональних прийнятих практиках у складі відповідних проектів для їх реалізації.

Самостійність виконання роботи: студентом було оцінено предмет наукової діяльності, визначено загальну мету і конкретні задачі кваліфікаційної бакалаврської роботи, та досягнуто їх результати, зокрема за рахунок максимальної активності самого студента.

Якість оформлення, загальна та спеціальна грамотність: Оформлення роботи відповідає вимогам до написання кваліфікаційних бакалаврських робіт.

Переваги та недоліки: беззаперечною перевагою роботи є її наукова новизна та практична цінність. Окрім помилок стилістичного характеру суттєвих недоліків не виявлено.

Загальна оцінка роботи та висновок щодо рекомендацій до захисту в ЕК:

аналіз роботи свідчить про глибоке бачення автором проблем проведеного дослідження. Це дозволяє зробити висновок про те, що кваліфікаційна бакалаврська робота Жука Дмитра Валентиновича на тему «Розробка алгоритмів для оцінювання навичок кандидатів на посади ІТ-фахівців з використанням SMART-аналізу» відповідає всім вимогам, що висуваються до кваліфікаційних робіт. Робота виконана на високому науковому та теоретичному рівнях і заслуговує найвищої оцінки, а автору роботи може бути присвоєно ступінь бакалавра зі спеціальності 124 «Системний аналіз».

Науковий керівник

завідувач кафедри системного аналізу

та кібербезпеки

Ірада ДЖАЛЛАДОВА

(підпис)

(ініціали, прізвище)

КНЕУ ім. Вадима Гетьмана

д.ф.-м.н., професор

«__» _____ 2024 р.

DIPLOMA THESIS REVIEW

Today, entrepreneurship emphasizes the use of the latest technologies and analytical tools for effective human resource management. It is vital to employ objective, accurate, and efficient methods to assess candidate skills, ensuring high-quality staffing and competitiveness. Implementing a modern approach of SMART analysis can significantly enhance the candidate selection process and human resource management. This ensure that candidates' skills align with the needs of the IT industry, contributing to the further sector's development overall.

In my Bachelor's Diploma Thesis, I investigated the general recruitment process, where pre-planning of the recruitment process to assess candidates' skills is the first stage and one of the key aspects of effective candidate selection. The second stage involves obtaining a large array of data, such as resumes, containing essential information about each candidate's skills, education, work experience, achievements, etc. The third stage is the selection stage itself: this involves drawing up a short list of a few applicants for a vacant position from the initial large pool of candidates.

To solve the problem of selecting the ideal candidate, I proved the need to implement an automated system for assessing skills precisely at the third stage, where the system can provide the most effective evaluation of candidate skills.

I proposed an automated solution consisting of three stages:

- 1) Parsing information from candidates' resumes using parsers, specifically with libraries like PyPDF, re, and NLTK;
- 2) Converting the extracted information into vectors using SMART analysis methods, such as Bag-of-words, TF-IDF, Word2Vec, GloVe, and fasttext. This stage also involves summarizing knowledge about words and considering their contexts in the text using models like ELMo and BERT;
- 3) Comparing vectors of resume corpora and job descriptions to assess the candidate's skills, experience and qualifications against the company's

requirements using similarity measures like Jaccard, Dice, cosine, sqrt-cos, and ISC.

To compare human and algorithmic efficiency and understand their correlation, a manual ranking was conducted. This involved manual evaluation for assessing and ranking candidates' skills based on their resumes. These manual rankings were then compared to the rankings generated by the intelligent system. The comparison aimed to identify discrepancies, validate the effectiveness of the automated system, and determine the extent to which human judgment aligns with algorithmic assessments.

The results obtained show a significant improvement in assessing candidates' skills, achieving a Weighted Tau coefficient of up to 0.857, though the initial data must be taken into account. My recommendations also include the active involvement of a hybrid "Human-in-the-loop" approach to ensure the optimal combination of intelligent technologies and peer review, thereby ensuring objectivity and neutrality in the process of assessing candidate skills.

Date:

22.5.2024



Whalebone, s.r.o., IČO: 05120403
Jezuitská 14/13, 602 00, Brno

Whalebone s.r.o.
Chief Technical Officer
Robert Šefr

JOB CHARACTERISTICS

Whalebone s.r.o. is a fast-growing cybersecurity company developing zero-disruption products for Telcos, ISPs, and enterprises. It provides millions of internet users with seamless DNS security for all the connected devices, including the IoT and other devices, protecting them from malware, phishing schemes, ransomware, and other malicious digital attacks without the need to download anything. Whalebone s.r.o. has more than 350+ customers in telecommunications and corporate sectors around the globe and 90+ team members with representation worldwide. The company's products include Aura for Telco operators, Immunity for enterprises and public institutions, Peacemaker for regional internet providers, and DNS4EU for development of the official DNS resolver for the European Union to protect 100 million EU citizens.

Dmytro Zhuk has been working as a Threat Intelligence Specialist since December, 2022. His responsibilities include but not limited to:

- 1) Performing the full data life cycle, including data mining, processing, analysis, and visualization, ensuring the integrity and accessibility of critical information;
- 2) Conducting research and development to analyze security trends, develop threat identification techniques, investigate root causes behind falsely categorized threats, and use machine learning methodologies to enhance threat detection capabilities;
- 3) Utilizing a combination of independent analysis and Open Source Intelligence (OSINT) to identify and assess threats.

He is capable of working both in team and independently, bringing a high level of expertise and dedication to each task. Also, he is responsible, problem-solving, and innovative. In addition to that, Dmytro continues to improve his communication and technical abilities.



Date:

22.5.2024


Whalebone s.r.o.
Chief Technical Officer
Robert Šefr

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 ТРАНСФОРМАЦІЯ ПРОЦЕСІВ УПРАВЛІННЯ ЛЮДСЬКИМИ РЕСУРСАМИ ЗА ДОПОМОГОЮ SMART-АНАЛІЗУ	5
1.1 Актуальні тенденції розвитку SMART-аналізу як сучасного підходу управління людськими ресурсами.....	5
1.2 Роль SMART-аналізу в ефективному управлінні персоналом підприємства.....	14
1.3 Впровадження інструментів та методів SMART-аналізу для оцінки навичок кандидатів в ІТ компаніях.....	23
РОЗДІЛ 2 АНАЛІЗ АЛГОРИТМІВ ДЛЯ ОЦІНЮВАННЯ НАВИЧОК КАНДИДАТІВ	35
2.1 Основні принципи, методи та характеристика ефективності алгоритмів для оцінювання навичок кандидатів.....	35
2.2 Вплив алгоритмів на об'єктивність та нейтральність процесу оцінювання навичок кандидатів.....	58
РОЗДІЛ 3 СТВОРЕННЯ АЛГОРИТМІВ ДЛЯ ЕФЕКТИВНОГО ОЦІНЮВАННЯ НАВИЧОК КАНДИДАТІВ НА ПОСАДИ ІТ-ФАХІВЦІВ	65
3.1 Розробка алгоритмів для оцінювання навичок кандидатів на посади ІТ-фахівців.....	65
3.2 Адаптація алгоритмів до специфічних вимог та особливостей ІТ-індустрії при оцінюванні навичок кандидатів на посади ІТ-фахівців.....	68
3.3 Тестування, оцінювання та узагальнення алгоритмів для різноманітних категорій навичок та компетенцій кандидатів на посади ІТ-фахівців.....	74
ВИСНОВКИ	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	80
ДОДАТКИ	87

ВСТУП

Актуальність теми. На сьогоднішній день підприємництво виявляє все більший інтерес до використання новітніх технологій та аналітичних інструментів для ефективного управління людськими ресурсами. Життєво необхідно мати об'єктивні, точні та ефективні методи оцінювання навичок кандидатів, щоб забезпечити високу якість кадрового забезпечення та конкурентоспроможність.

Використання сучасного підходу SMART-аналізу може значно покращити процес підбору кандидатів та управління людськими ресурсами, забезпечуючи відповідність їхніх навичок потребам ІТ-індустрії та сприяючи подальшому розвитку цього сектору в цілому.

Метою дослідження є обґрунтування теоретичних і прикладних засад алгоритмів для оцінювання навичок кандидатів на посади ІТ-фахівців за допомогою SMART-аналізу для підбору ідеального кандидата на вакантну посаду ІТ-фахівця з урахуванням специфіки ІТ-індустрії.

Мета дослідження розкривається переліком таких завдань дослідження:

- Охарактеризувати можливі підходи для оцінювання навичок кандидатів на посади ІТ-фахівців;
- Визначити сутність та роль SMART-аналізу як одного з сучасних підходів управління людськими ресурсами для оцінювання навичок кандидатів на посади ІТ-фахівців;
- Скласти порівняльну характеристику методів оцінювання навичок кандидатів на посади ІТ-фахівців;
- Розробити алгоритми для ефективного оцінювання навичок кандидатів на посади ІТ-фахівців та підбору ідеального кандидата, враховуючи та забезпечуючи об'єктивність та нейтральність процесу оцінювання, оцінити їх точність з урахуванням специфіки ІТ-індустрії.

Об'єктом дослідження є процес підбору та оцінювання навичок кандидатів на посади ІТ-фахівців.

Предметом дослідження є методи та інструменти SMART-аналізу для оцінювання навичок кандидатів на посади ІТ-фахівців.

Методи дослідження включають в себе такі теоретичні та емпіричні методи: пояснення, класифікація, аналіз, синтез, моделювання, експеримент, порівняння, розрахунок, узагальнення, інтерпретація, ранжування, експертне оцінювання.

Практична цінність результатів дослідження полягає у покращенні ефективності та точності процесу відбору кандидатів на посади ІТ-фахівців в ІТ-компаніях. Зокрема, за результатами дослідження, використання автоматизованих технологій за допомогою сучасного підходу SMART-аналізу поліпшує процес підбору кандидатів та управління людськими ресурсами, забезпечуючи відповідність їхніх навичок потребам бази практики Whalebone s.r.o. та сприяючи подальшому гармонічному розвитку цього напрямку в компанії. Також практичні та методичні розробки були розглянуті керівництвом бази практики та прийняті до апробації.

Інформаційною базою дослідження слугують наукові дослідження, публікації та періодичні видання, збірники, дослідницькі та оглядові статті, нормативно-правові акти органів державної влади, інтернет-джерела включаючи офіційні сайти підприємств, аналітичні матеріали, відкриті фактологічні дані та власні результати емпіричних досліджень, підручники та навчальні посібники у сфері SMART-аналізу та управління людськими ресурсами.

Структура роботи – кваліфікаційна бакалаврська робота складається зі вступу, трьох розділів, висновків, списку використаних джерел із 91 найменування та 1 додаток. Робота містить 4 таблиці і 36 рисунків. Загальний обсяг роботи складає 86 сторінок.

РОЗДІЛ 1

ТРАНСФОРМАЦІЯ ПРОЦЕСІВ УПРАВЛІННЯ ЛЮДСЬКИМИ РЕСУРСАМИ ЗА ДОПОМОГОЮ SMART-АНАЛІЗУ

1.1 Актуальні тенденції розвитку SMART-аналізу як сучасного підходу управління людськими ресурсами

Такі технології, як штучний інтелект, аналітика даних, хмарні рішення та програмне забезпечення для управління персоналом, забезпечують підприємствам можливість здійснювати більш точне прогнозування потреб у робочій силі, залучати та утримувати талановитих працівників, оптимізувати процеси найму та навчання, а також підвищувати ефективність комунікації та співпраці в колективі.

Загалом, термін “управління людськими ресурсами” має декілька значень. Наприклад, управління людськими ресурсами – це діяльність з управління працівниками шляхом найму нових працівників, їх навчання та подальшого супроводження, підтримки в компанії [1]. Згідно визначенню британського вченого М. Армстронга, термін “управління людськими ресурсами” означає “функцію менеджменту; послідовний та стратегічний підхід до управління цінними активами організації, а саме – працюючою там людиною, яка індивідуально й колективно робить свій внесок для вирішення задач організації” [2]. У підсумку, управління людськими ресурсами є однією з функцій менеджменту; це вид діяльності та послідовний, стратегічний підхід до працівників, які приносять користь організації та роблять внесок як індивідуально, так і в команді.

Управління людськими ресурсами включає в себе такі категорії цілей:

- Соціальні цілі. Це заходи, вжиті для відповіді на етичні та соціальні потреби чи виклики компанії та її працівників – включає в себе правові питання, такі як рівні можливості та рівна оплата за однакову працю;
- Організаційні цілі. Це заходи, які вживаються для забезпечення ефективності організації, включаючи забезпечення відповідного навчання, наймання належної кількості працівників для виконання завдання та підтримку високих кількісних та якісних показників утримання працівників;
- Функціональні цілі. Це вказівки, які використовуються для належного функціонування відділу кадрів в організації. Вони включають забезпечення повного використання всіх кадрових ресурсів;
- Особисті цілі. Це ресурси, які використовуються для підтримки особистих цілей кожного працівника. Вони включають можливості для навчання та розвитку кар'єри, а також збереження задоволеності працівників.

Інші цілі управління людськими ресурсами включають наступне:

- Забезпечення та утримання продуктивних працівників;
- Повідомлення працівникам політику, процедури, правила та положення компанії;
- Дотримання етичної, правової та соціально відповідальної політики та поведінки на робочому місці;
- Керування внутрішніми та зовнішніми змінами, які можуть вплинути на працівників і персонал.

Метою практик управління людськими ресурсами передусім є управління людьми на робочому місці для досягнення місії організації та зміцнення корпоративної культури. Коли управління людськими ресурсами здійснюється ефективно, менеджери з персоналу можуть допомогти найняти нових співробітників, які мають необхідні навички для досягнення цілей компанії. Фахівці з управлінням людськими ресурсами також допомагають у навчанні та професійному розвитку працівників для досягнення цілей організації. Крім того, менеджери з персоналу

відстежують стан ринку праці, щоб допомогти організації залишатися конкурентоспроможною. Це може включати забезпечення конкурентоспроможності винагороди та пільг, планування заходів, щоб уберегти працівників від вигорання, і адаптацію посадових ролей відповідно до ринку.

Впровадження сучасних інструментів та методів для ефективного та якісного управління людськими ресурсами включають широкий спектр технологічних рішень, які оптимізують процеси управління персоналом, покращують комунікацію та покращують загальну роботу відділу кадрів, за допомогою штучного інтелекту, аналітики даних та Big Data, хмарних обчислень, мобільних технологій, віртуальної та доповненої реальності, чат-ботів, віртуальних асистентів тощо. Серед них можна виділити зокрема:

- Інформаційні системи людських ресурсів (HRIS): платформи HRIS автоматизують та централізують процеси управління людськими ресурсами, включаючи управління даними про співробітників, нарахування заробітної плати, адміністрування пільг, відстеження робочого часу та відвідуваності, а також управління продуктивністю. Вони забезпечують єдине джерело даних про кадри, забезпечуючи ефективний аналіз даних і звітність;
- Системи відстеження кандидатів (ATS): програмне забезпечення ATS спрощує процес найму, автоматизуючи оголошення про роботу, перевірку резюме, відстеження кандидатів і планування співбесід. Це допомагає командам відділу кадрів оптимізувати зусилля із залучення талантів і ефективно керувати набором кандидатів;
- Системи управління навчанням (LMS): платформи LMS дозволяють створювати, надавати та відстежувати онлайн-навчання та програми розвитку. Вони забезпечують централізований центр навчання співробітників, пропонуючи різноманітні навчальні ресурси, оцінювання та функції відстеження прогресу;

- Програмне забезпечення для управління продуктивністю: Інструменти управління ефективністю сприяють поточній оцінці продуктивності, постановці цілей і обміну відгуками між керівниками та співробітниками. Вони дозволяють у режимі реального часу відстежувати прогрес співробітників, показники продуктивності та перевіряти продуктивність;
- Портали самообслуговування співробітників: портали самообслуговування надають співробітникам легкий доступ до особистої інформації про кадри, такої як квитанції про оплату праці, відомості про переваги, запити на відпустку та відгуки про продуктивність. Ці портали дають можливість співробітникам самостійно керувати своїми завданнями, пов'язаними з кадрами, зменшуючи адміністративне навантаження на команди відділу кадрів;
- Мобільні HR-додатки: мобільні HR-додатки дозволяють співробітникам отримувати доступ до кадрових послуг і інформації в будь-який час і в будь-якому місці за допомогою смартфонів або планшетів. Ці додатки зазвичай пропонують такі функції, як функції самообслуговування, інструменти зв'язку та доступ до кадрових політик і документів;
- Інструменти для співпраці та комунікації: такі інструменти, як програмне забезпечення для управління проектами, платформи миттєвого обміну повідомленнями та програми для відеоконференцій, сприяють ефективній співпраці та комунікації між командами відділу кадрів, а також між відділом кадрів і співробітниками. Вони сприяють ефективному обміну інформацією та командній роботі;
- Інструменти для аналізу даних і звітності: програмне забезпечення для аналітики кадрів використовує дані, щоб надати інформацію про тенденції в робочій силі, продуктивність співробітників, плинність кадрів та інші показники кадрової роботи. Ці інструменти допомагають HR-фахівцям приймати рішення на основі даних і вимірювати вплив HR-ініціатив;

- Соціальні медіа та онлайн-платформи: платформи соціальних медіа та професійні онлайн-мережі все частіше використовуються для брендингу роботодавця, найму та ініціатив щодо залучення працівників. Вони надають канали для пошуку талантів, брендингу роботодавця та сприяння спільнотам працівників;
- Віртуальні рішення для адаптації та навчання: із зростанням віддаленої роботи та віртуальних команд рішення для віртуальної адаптації та навчання стали вирішальними. Ці інструменти дозволяють організаціям ефективно залучати нових співробітників і дистанційно проводити навчальні програми, забезпечуючи плавний перехід і можливості постійного навчання.

Згідно з дослідженням аналітичної компанії Gartner, найважливіші напрямки у сфері управління людськими ресурсами визначають передові технологічні рішення для управління людськими ресурсами, нові стандарти ефективної організаційної культури, зростання індивідуального та професійного розвитку через управління кар'єрою, а також вдосконалення внутрішньої мобільності персоналу [3]. Дослідження аналітичної компанії Deloitte вказують на найбільш поширені напрями застосування та впровадження технологій наступні 2-4 роки, табл. 1.1:

Табл. 1.1 – Основні напрями впровадження технологій для управління людськими ресурсами [4]

Напрямок впровадження технологій наступні 2-4 роки	Пріоритет в Україні	Пріоритет у світі
Створення мережі команд, як внутрішніх, так і зовнішніх, для спільної роботи та обміну інформацією як всередині, так і поза межами організації.	1	3
Підвищення ефективності роботи працівників на індивідуальному рівні шляхом оптимізації операційних процесів.	2	4
Організація комунікації між членами команд.	3	9
Сприяння розробці та генерації ідей.	4	8

Продовження таблиці 1.1

1	2	3
Розвиток можливостей для ефективного використання професійних навичок та компетенцій.	5	2
Підтримка у прогнозуванні потреб, працівників, спонукання до розвитку команд та працівників.	8	1

Підтримка у прогнозуванні потреб, працівників, спонукання до розвитку команд та працівників, а також розвиток можливостей для ефективного використання професійних компетенцій мають найвищий пріоритет у світі, але становлять одні з найнижчих пріоритетів в Україні. Це може свідчити про необхідність для українських компаній звернути більшу увагу на стратегічне планування та прогнозування потреб у працівниках. В Україні та світі виявляється загальна тенденція до зосередження уваги на побудові ефективних комунікаційних зв'язків та розвитку колективної співпраці. Українські компанії, хоча і визнають важливість цих напрямків, приділяють їм менше пріоритетності порівняно зі світовими стандартами. Проте, в обох випадках виділяється необхідність використання та впровадження технологій для ефективного управління людськими ресурсами.

Згідно консалтингової компанії FDM Group, близько 67% компаній використовують автоматизацію бізнес-процесів для оптимізації рутинних завдань та підвищення загальної продуктивності працівників. Впровадження технологій для управління людськими ресурсами допомагає вивільняти людські ресурси для зосередження на більш складних та стратегічних завданнях [5].

Тим не менш, існують перешкоди у впровадженні сучасних інструментів та методів для управління людськими ресурсами:

- Фінансові обмеження: Впровадження сучасних технологій може потребувати значних інвестицій у програмне забезпечення, інфраструктуру та навчання персоналу. Фінансові обмеження можуть ускладнити процес впровадження, особливо для малих і середніх підприємств;

- Недостатня підготовка персоналу: Успішне впровадження нових технологій вимагає належної підготовки та навчання персоналу. Брак кваліфікованих кадрів або недостатня освіченість з використання технологій може стати перешкодою для їх успішного впровадження;
- Відсутність стратегії впровадження: Брак чіткої стратегії впровадження технологій може призвести до недостатньої координації, низької впливовості та неефективного використання ресурсів;
- Проблеми з безпекою та конфіденційністю даних: Використання сучасних технологій може викликати питання щодо захисту конфіденційності та безпеки персональних даних співробітників.

Зважаючи на стрімкий розвиток технологій та зміни в бізнес-середовищі за останні п'ять років, масштаби роботи в сфері управління людськими ресурсами різко змінилися, з чим згодні близько 91% професіоналів [6]. Також, згідно з дослідженням аналітичної компанії Gartner [3], серед головних викликів для впровадження технологій професіонали у сфері управління людськими ресурсами зазначають:

- Невпевненість щодо впливу технологій – згідні 60% респондентів: Невизначеність щодо впливу передових технологій, таких як генеративний штучний інтелект, на управління кадрами та талантами. Зростаюча автоматизація та автономія роботи можуть вплинути на структуру робочих місць та навички, які необхідно мати для успішного виконання завдань. Професіонали у сфері управління людськими ресурсами відчують потребу в більш глибокому розумінні технологій та їхнього впливу на ринок праці;
- Невідповідність технологічних рішень та стратегій управління людськими ресурсами бізнес-потреbam – згідні 56% респондентів: Інший значущий виклик полягає у забезпеченні того, щоб технологічні рішення у сфері управління людськими ресурсами відповідали поточним та майбутнім потребам бізнесу. Це включає в себе адаптацію до цифрової трансформації, урахування змін у

робочих методах та створення інноваційних стратегій управління персоналом для підтримки стратегічних цілей організації.

На думку респондентів дослідження аналітичної компанії Deloitte, ключовими бар'єрами на шляху впровадження змін зазначають, табл. 1.2:

Табл. 1.2 – Основні виклики впровадження змін для управління людськими ресурсами [4]

Виклики в Україні	Виклики в світі
Швидкість технологічних змін.	Технології та технічна інфраструктура компаній.
Технології та технічна інфраструктура компаній.	Корпоративна культура.
Обмеженість навичок, рівень залученості керівництва.	Способи роботи.

Підсумовуючи, впровадження сучасних інструментів та методів для управління людськими ресурсами неминуче супроводжується рядом перешкод, які варто враховувати при розробці та реалізації стратегій управління. Технології та інфраструктура компаній визначають готовність до змін – швидкість змін може бути значним викликом, особливо в контексті розвитку технологій та нестабільності бізнес-середовища. Саме тому зростає потреба у якісних, вимірюваних, досяжних, актуальних та оптимальних підходах, методик для ефективного управління людськими ресурсами.

Однією з найбільш обіцяючих та швидкозростаючих технологічних методик є SMART-аналіз, який спрямований на збільшення продуктивності, оптимізацію робочих процесів та вдосконалення управлінських рішень у сфері управління людськими ресурсами. SMART-аналіз стає невід'ємною складовою сучасного управління людськими ресурсами, адже він дозволяє підприємствам отримувати корисну інформацію з великих обсягів даних, що стосуються кадрового планування, рекрутингу, оцінки робочої продуктивності, управління талантами та інших аспектів управління людськими ресурсами для прийняття зваженого та об'єктивного рішення.

Використання SMART-аналізу дозволяє систематизувати та враховувати ключові параметри та критерії, що є важливими для успішної та продуктивної роботи

в галузі ІТ. Такі дослідження підкреслюють потребу в подальшому розвитку та впровадженні алгоритмів оцінювання, які базуються на SMART-підході, для оптимізації процесу підбору та забезпечення відповідності навичок кандидатів вимогам ІТ-індустрії. Варто зазначити, що дослідженню проблеми ефективного підбору присвятили багато уваги такі вчені, як Е. Хаббард, М. Армстронг, Д. Доран, А. Міллер, Д. Каннінгем та інші.

Загальні технологічні тренди в управлінні людськими ресурсами, згідно з аналітичною компанією Gartner, наведені на рис 1.1.

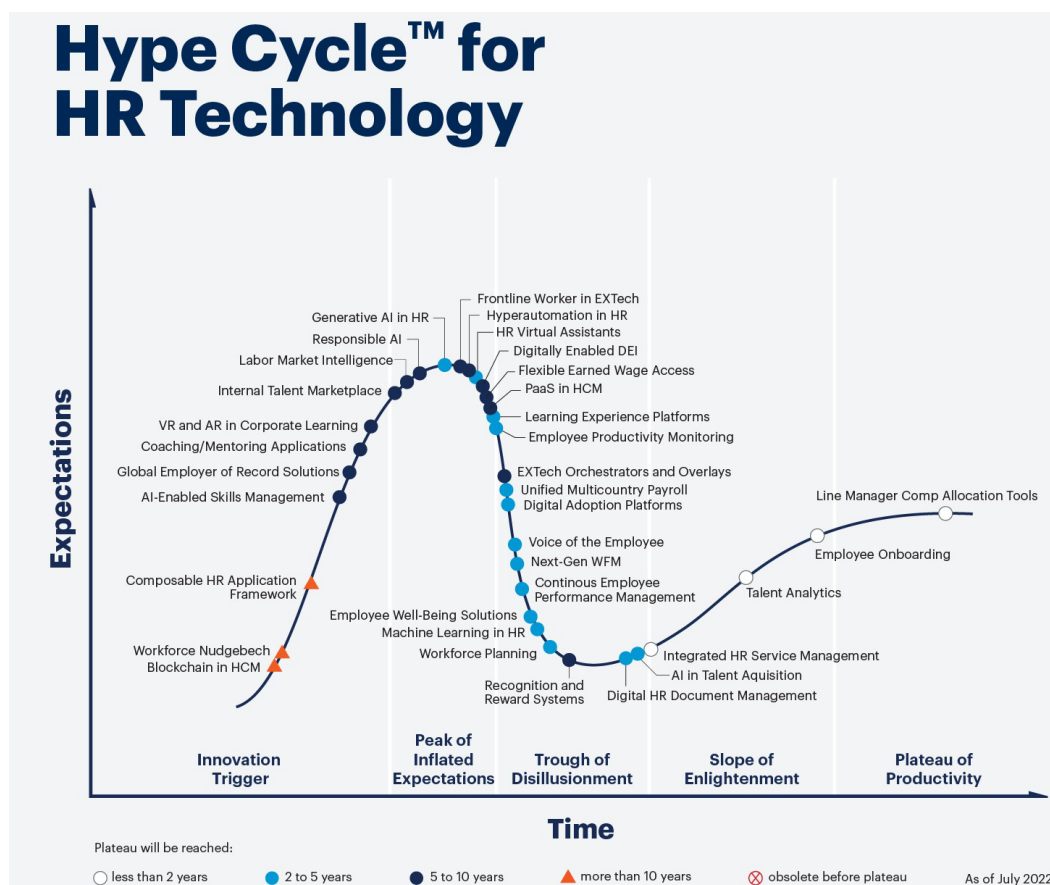


Рис. 1.1 – Основні технологічні тренди в сфері управління персоналом [7]

1.2 Роль SMART-аналізу в ефективному управлінні персоналом підприємства

SMART-аналіз – мультидисциплінарний інноваційний напрям, що базується на використанні штучного інтелекту, зокрема машинного навчання, інтернету речей, хмарних обчислень та кібербезпеки. Цей напрям орієнтований на створення інтелектуальних систем та інструментів для збору, обробки, аналізу та інтерпретації даних з метою прийняття більш обґрунтованих та ефективних рішень:

- Використання штучного інтелекту дозволяє створювати алгоритми та моделі, які можуть аналізувати великі обсяги даних та виявляти складні закономірності, що допомагає в розумінні та прогнозуванні трендів;
- Машинне навчання використовується для створення моделей, які можуть самостійно навчатися на основі даних та вдосконалювати свої результати з часом, що дозволяє покращувати якість аналізу та передбачення;
- Хмарні обчислення використовуються для збереження та обробки великих обсягів даних у хмарних обчислювальних сервісах, що дозволяє підвищити масштабість та швидкодію аналітичних процесів;
- Інтернет речей надає можливість збирати дані з різних джерел, таких як сенсори, пристрої та обладнання, що дозволяє отримувати більше інформації для аналізу та прийняття рішень;
- Кібербезпека для забезпечення захисту конфіденційності, цілісності та доступності даних у процесі їх збору, передачі та аналізу.

Цілі SMART-аналізу базується на методиці SMART. Ще наприкінці 19 століття американський філософ Елберт Хаббард зрозумів, що багато людей зазнавали невдачі у своїх починаннях та цілях – вони зазнавали невдачі не тому, що їм бракувало розуму чи сміливості, а тому, що вони не організовували ефективно свої зусилля навколо мети. Їм потрібен був спосіб організовувати та спрямовувати свої зусилля, щоб

підвищити шанси на успіх – лише наприкінці 20-го століття допомога прийшла у вигляді методики постановки цілей SMART. Методика постановки цілей SMART була розроблена Д. Дораном, А. Міллером і Д. Каннінґемом у 1981 році. Вперше вона була представлена у статті під назвою “There’s a S.M.A.R.T. way to write management goals and objectives” (в перекладі: “Існує SMART-метод написання цілей та задач управління”), і з тих пір методика перетворилася на спосіб, у який керівники та менеджери створюють цілі для бізнесу. Методика SMART надає структуру та організованість при постановці цілей, допомагаючи уникнути загальних, нечітких або недосяжних формулювань цілей. Проте необхідно зазначити, що ціль не мусить підпадати під кожен вид цілі за методикою SMART – наприклад, в деяких випадках може бути важливіше мати конкретну та вимірювану ціль, ніж встановлення строгої часової рамки. SMART-аналіз складається з цілей, які повинні бути, див. табл 1.3.

Табл. 1.3 – Види цілей за допомогою методики постановки цілей SMART [8]

Вид цілі	Опис цілі
S: Specific Конкретна ціль	Ціль повинна бути чітко сформульованою та спрямованою на конкретний результат. Типові питання: Що я хочу досягти? Які дії мені потрібно буде вжити?
M: Measurable Вимірна ціль	Ціль повинна бути вимірюваною, повинна існувати можливість об’єктивно оцінити ступінь досягнення цілі. Типові питання: Як я дізнаюся, що я досяг своєї мети? За якими показниками я буду знати, чи досягну своїх цілей?
A: Achievable Досяжна ціль	Ціль повинна бути реалістичною та досяжною для виконання з наявними ресурсами та обмеженнями. Типові питання:

Продовження таблиці 1.3

1	2
	Чи моя мета досяжна?
R: Relevant Релевантна ціль	Ціль повинна бути зв'язаною з іншими стратегічними цілями та контекстом завдань. Типові питання: Чи матиме досягнення цієї мети важливий вплив? Чи добре ця мета узгоджується з іншими цілями?
T: Time-bound Обмежена часом ціль	Ціль повинна мати чітко визначений термін досягнення. Типові питання: Коли я виконаю це завдання?

Дослідження трендів у сфері SMART-аналізу для управління людськими ресурсами аналітичною компанією Gartner вказують на такі основні тренди, див. табл. 1.4.

Табл. 1.4 – Впровадження технологій SMART-аналізу для управління людськими ресурсами [7]

Очікуваний час масового застосування технологій SMART-аналізу для управління людськими ресурсами			
Менш, ніж 2 роки	Від 2 до 5 років	Від 5 до 10 років	Більше 10 років
SMART-аналітика компетенцій працівників	Генеративний штучний інтелект в управлінні персоналом	Управління навичками за допомогою штучного інтелекту	Блокчейн в управлінні людськими ресурсами
Інструменти для розподілу компенсації та управління	Віртуальні асистенти для управління персоналом	Технології віртуальної та доповненої реальності в корпоративному навчанні	
Інтегроване в екосистеми управління службами для управління персоналом	Платформи для обміну досвідом в компанії	Відповідальний штучний інтелект	

Продовження таблиці 1.4

1	2	3	4
	Системи моніторингу продуктивності працівників	Гіперавтоматизація в управлінні персоналом	
	SMART-управління робочим часом (WFM)	Цифрові системи з акцентом на різноманіття, рівність та інклюзію (DEI)	
	Машинне навчання в управлінні персоналом	Платформа як сервіс (PAAS) для управління людськими ресурсами	

Компанії сьогодні впроваджують штучний інтелект в процесах управління людськими ресурсами, щоб отримати інформацію для прийняття рішень, полегшити взаємодію з відділом кадрів для співробітників, кандидатів, звільнити час для прийняття кращих рішень щодо найму [9]. Інші застосування штучного інтелекту та машинного навчання в управлінні людськими ресурсами включають:

- Професійний розвиток: штучний інтелект може надати персоналізовані рекомендації та навчальні модулі для подальшого кар'єрного росту для кожного співробітника. Аналізуючи дані про кожного працівника, їх навички та уподобання, алгоритми машинного навчання можуть відповідно адаптувати процес навчання;
- Пошук та найм кандидатів: штучний інтелект прискорює цей процес, допомагаючи професіоналам з управління людськими ресурсами автоматично слідкувати за статусом кожного потенційного кандидата в робочих соціальних мережах та надавати сповіщення, коли кандидат подає заявку на відкриту посаду;
- Процес онбордингу: алгоритми машинного навчання можуть зробити процес збору інформації більш плавним та персоналізованим. Чат-боти на базі штучного інтелекту можуть допомагати новим працівникам: відповідати на

питання, надавати необхідну інформацію та надсилати нагадування про ключові робочі моменти;

- Аналіз задоволеності та збереження персоналу: за допомогою алгоритмів машинного навчання можна аналізувати великі обсяги даних щодо задоволеності співробітників, виявляти патерни та фактори, що впливають на збереження персоналу. Це дозволяє розробляти стратегії збереження та підвищення задоволеності співробітників;
- Оптимізація робочих процесів: штучний інтелект може аналізувати робочі процеси та виявляти можливості для їх оптимізації та автоматизації;
- Прогнозування потреб у робочій силі: на основі аналізу даних про динаміку бізнесу, штучний інтелект може допомогти у прогнозуванні потреб у робочій силі компанії. Це дозволяє планувати найм та розвиток персоналу заздалегідь, зменшуючи ризик нестачі або перевищення кадрових ресурсів.

В сфері управління людськими ресурсами хмарні обчислення відносяться до технологічних рішень, які зберігають та керують корпоративними даними, процесами та додатками на віддалених серверах. Хмарні обчислення використовують зовнішні центри обробки даних і пропонують більшу гнучкість та ефективність, ніж локальні рішення, які потребують фізичного обладнання. Наприклад, компанія PricewaterhouseCoopers (PwC) активно використовує хмарні технології для оптимізації управління людськими ресурсами, співпрацюючи з компаніями Oracle, SAP та Workday. Це включає в себе використання хмарних додатків, які охоплюють всі аспекти управління, забезпечуючи співробітників та кандидатів можливістю онлайн-обслуговування: чат-боти для забезпечення ефективної машино-людської взаємодії, рекомендаційні системи [10]. Інші можливості застосування хмарних обчислень в управлінні людськими ресурсами можуть бути:

- Централізація даних: рішення у хмарі зберігають всі дані, включаючи інформацію про співробітників, оцінки результативності, деталі заробітної

плати тощо, в одному централізованому та віддаленому місці для спрощення управління даними та гарантії їх однорідності у всій організації;

- Аналітика в реальному часі: хмарні рішення мають вбудовані інструменти аналітики, що надають цінні знання щодо тенденцій розвитку робочої сили, показників продуктивності, рівня текучості кадрів та інших ключових показників управління людськими ресурсами – це дозволяє приймати рішення на основі даних та допомагає організаціям оптимізувати свої стратегії управління людськими ресурсами;
- Підвищена безпека: провідні постачальники хмарних послуг інвестують значні кошти у заходи безпеки для захисту даних, збережених на їхніх платформах. Системи управління людськими ресурсами у хмарі пропонують потужні засоби безпеки, такі як шифрування, контроль доступу та регулярні резервні копії даних, для захисту чутливої інформації;
- Колаборація у віддаленому середовищі: рішення у хмарі сприяють співпраці між керівниками та співробітниками – вони дозволяють обмінюватися інформацією, документами та оновленнями в реальному часі, покращуючи комунікацію та співпрацю в організації.

Інтернет речей та його складові відіграють роль в ефективному управлінні персоналом підприємства, надаючи спектр можливостей для автоматизації процесів, моніторингу даних та взаємодії з працівниками. Інтеграція SMART-об'єктів Інтернету речей в офісне середовище дозволяє автоматизовувати збір даних та аналізувати їх для забезпечення оптимальних умов для працівників. Наприклад, фірма Spark Innovation Studio, що спеціалізується на Інтернеті речей, в 2023 році почала співпрацю зі стартапом AirSuite, командою спеціалістів з охорони навколишнього середовища, для створення більш екологічного середовища для співробітників всередині приміщення офісних будівель за допомогою моніторингу якості повітря, рівня тиску, вологості, CO₂, температури, шуму та освітлення [11].

Такий підхід до управління середовищем у приміщенні може позитивно вплинути на здоров'я, комфорт та продуктивність працівників, а також сприяти збереженню енергії та зниженню операційних витрат. Інші застосування Інтернету речей в управлінні людськими ресурсами включають:

- Моніторинг співробітників та відстеження відвідуваності: пристрої Інтернету речей, такі як біометричні датчики, мітки RFID або системи розпізнавання облич, можна використовувати для точного відстеження відвідуваності співробітників і робочих годин. Це зменшує ручні зусилля в управлінні відвідуваністю та забезпечує точність;
- Управління зайнятістю та простором: SMART-об'єкти Інтернету речей можна розгорнути для моніторингу використання офісного простору, допомагаючи менеджерам з персоналу оптимізувати макети робочого простору та ефективніше керувати офісними ресурсами. Ці дані також можна використовувати для планування об'єктів і зниження витрат;
- Програми оздоровлення співробітників: такі SMART-об'єкти Інтернету, як фітнес-трекери або розумні ваги, можна інтегрувати в програми оздоровлення співробітників, щоб стимулювати фізичну активність і здорові звички. Дані з цих пристроїв можна використовувати для адаптації ініціатив щодо оздоровлення та надання персоналізованого зворотного зв'язку співробітникам;
- Моніторинг віддаленої роботи: із зростанням віддаленої роботи технології Інтернету речей можна використовувати для моніторингу продуктивності та залученості співробітників поза офісом. Це може включати відстеження робочого часу, використання мережі Інтернет та виконання завдань через підключені пристрої, такі як ноутбуки чи смартфони;
- Навчання та розвиток: пристрої з підтримкою Інтернету речей можуть полегшити програми навчання та розвитку співробітників, забезпечуючи інтерактивне навчання. Наприклад, гарнітури віртуальної реальності можна

використовувати для імерсивного моделювання навчання, тоді як датчики Інтернету речей можуть відстежувати прогрес і показники продуктивності;

- Зворотний зв'язок і залучення: SMART-об'єкти Інтернету можуть збирати дані про настрої та залученість співробітників за допомогою таких методів, як опитування пульсу або датчики відстеження настрою. Ця інформація може допомогти менеджерам з персоналу визначити сфери для вдосконалення та реалізувати стратегії для підвищення задоволеності та утримання співробітників;
- Прогнозна аналітика для відділу кадрів: дані, створені мережею SMART-об'єктів, можна аналізувати за допомогою прогновної аналітики для прогнозування тенденцій у поведінці, продуктивності та утриманні співробітників. Це дозволяє завчасно вирішувати проблеми, і приймати рішення на основі даних для управління талантами.

З погляду кібербезпеки, управління людськими ресурсами вимагає підвищеної уваги до забезпечення безпеки інформації та захисту від потенційних кіберзагроз, що можуть виникнути як ззовні, так і всередині організації, оскільки велика частина даних, що обробляються, є конфіденційною та чутливою інформацією: персональні дані співробітників, документація про найм та звільнення, оцінки працівників тощо.

Наприклад, один із керівників служби безпеки, опитаний для звіту Global Cybersecurity Outlook 2023 спільно з компанією Accenture, зазначив, що команда відділу кадрів організації значно частіше відкриває підозрілі вкладення, ніж інші підрозділи організації. Подальше розслідування виявило, що співробітники команди не мали захищеного порталу для доступу до заявок на вакансію від зовнішніх кандидатів, тому їм доводилося відкривати велику кількість резюме, які надходили як вкладення електронної пошти. Обсяг вкладень, оброблених цією командою, збільшив ймовірність відкриття зараженого шкідливим програмним забезпеченням вкладення – це дозволило керівнику служби безпеки надати організаційну рекомендацію, щоб команда відділу кадрів була забезпечена онлайн-порталом для подання заявок на

вакансію, щоб зменшити ризик відкриття шкідливих файлів, які можуть завдати серйозної шкоди компанії в цілому [12]. Для безпечного управління людськими ресурсами необхідна розробка та впровадження стратегій і політик безпеки інформації, які враховують заходи для захисту конфіденційності, цілісності та доступності даних про співробітників та організацію.

1.3 Впровадження інструментів та методів SMART-аналізу для оцінки навичок кандидатів в ІТ компаніях

Зростаюча конкуренція на ринку праці та постійна потреба висококваліфікованих кадрів підштовхує ІТ компанії до використання передових інструментів та методів оцінки кандидатів. Проведення попереднього відбору має вирішальне значення для знаходження ідеального кандидата на вакантну посаду. Саме перед прийомом на роботу визначається процес попереднього відбору, а також конкретні потреби компанії, які враховуються під час процесу, наприклад:

- Посадова інструкція: функції, завдання та обов'язки за посадою;
- Відділ: робоча група, обов'язки нагляду;
- Режим вакансії: очний, гібридний або віддалений;
- Обов'язки: завдання та очікувані результати;
- Графік роботи: неповний робочий день, повний робочий день, конкретні дати;
- Вид найму: безстроковий або тимчасовий;
- Кількість вакансій: чи шукаються одна чи декілька осіб.

Оцінка навичок кандидатів – це процес визначення та аналізу навичок, знань, досвіду та інших характеристик, необхідних для виконання конкретної ролі або посади. Це ключовий етап у процесі попереднього відбору персоналу, оскільки допомагає роботодавцям прийняти найбільш об'єктивне рішення щодо того, чи відповідає кандидат потребам організації.

SMART-аналіз вважається ефективним підходом для підвищення якості найму та забезпечення відповідності кандидатів згідно вимог позицій, оскільки багато компаній у сфері ІТ-індустрії вже впроваджують інструменти та методи SMART-аналізу для оцінки навичок кандидатів. Згідно дослідження Harvard Business Review в співробітництві з компанією Paradox, що спеціалізується на інтелектуальному програмному забезпеченні для найму персоналу, 97% респондентів опитування, чий

організації застосували автоматизовані технології в процесі найму, сказали, що це допомогло ефективніше наймати людей, призвело до швидшого планування співбесід і скорочення відходу кандидатів. Також переважна частина професіоналів (близько 91% респондентів) з найму вважає, що оптимізація процесів найму за допомогою автоматизації та штучного інтелекту є необхідною для довгострокового успіху в бізнесі [13].

Розглянемо модельні приклади, де використовується світовий досвід впровадження SMART-аналізу, для ефективного управління людськими ресурсами, зокрема оцінюванням навичок кандидатів.

Компанія Oracle – багатонаціональна технологічна корпорація, що спеціалізується на програмному забезпеченні, включаючи програмне забезпечення для планування ресурсів підприємства, управління людським капіталом та людськими ресурсами, взаємовідносинами з клієнтами, базах даних, хмарних обчислень тощо.

Oracle Fusion Cloud Applications – екосистема хмарних програмних додатків, розроблених компанією Oracle, які об'єднують в собі різноманітні функції для автоматизації бізнес-процесів та розповсюджуються як самостійні послуги для управління людськими ресурсами, управління виробництвом, управління постачанням тощо. Вона надає набір готових рішень на одній платформі даних та має близько 14 000 клієнтів по всьому світу.

Oracle Cloud Human Capital Management (Oracle Cloud HCM) – хмарне програмне рішення, що розповсюджується як частина екосистеми Oracle Cloud Applications, для планування ресурсів підприємства, управління ланцюгом поставок, реклами та взаємодії з клієнтами для кращого узгодження бізнес-цілей для ефективного управління людськими ресурсами. Також зазначається, що рішення використовує інструменти та методи SMART-аналізу (близько 5000 практик), такі як штучний інтелект, машинне навчання, блокчейн, інтернет речей та віртуальні асистенти [14].

Важливо відмітити, що функції на основі штучного інтелекту в Fusion Cloud Applications працюють на базі хмарної інфраструктури Oracle Cloud Infrastructure (OCI) – їх швидкодія досягається за допомоги кластерної технології GPU з масштабом понад 16K H100 GPU на кластер [15].

Oracle Recruiting – хмарний програмний додаток, що спеціалізується виключно на процесі рекрутингу кандидатів, є частиною рішення Oracle Cloud Human Capital Management. [16] Особливостями додатку є:

1. SMART-публікація вакансії: Oracle Recruiting дозволяє легко публікувати вакансії на персоналізованих сайтах пошуку роботи в компанії, створювати опис вакансії за допомогою штучного інтелекту та надавати цілодобову підтримку кандидатам за допомогою віртуальних асистентів;
2. Ефективне залучення кандидатів: за допомогою штучного інтелекту додаток може виявляти найбільш підходящих кандидатів, а також працювати як з масовим, так і з традиційним наймом, використовуючи масові email-кампанії та соціальні мережі, щоб посилити охоплення та збільшити кількість заявок на вакансію;
3. Автоматизація рекрутингової діяльності: в Oracle Recruiting існує можливість призначати співбесіди як з боку рекрутера, так і кандидата, враховуючи зручний час для кожної з сторін, створення центру рекрутингової діяльності для швидкого розуміння пріоритетних завдань та дій через персоналізований та узгоджений інтерфейс, див. рис 1.2., а також використання шаблонів та даних з попередніх наймів відповідно до внутрішніх бізнес-процесів та політик компанії;
4. Аналітика та створення звітів у реальному часі: додаток дозволяє візуалізувати KPI та прогнозні показники, що створюються на основі попереднього досвіду за допомогою машинного навчання, найму в реальному часі для прийняття зважених рішень на основі даних, також дізнаватися більше про досвід, навички та таланти працівників в наочній формі;

5. Швидкий онбординг кандидатів: за допомогою Oracle Recruiting можна створювати заходи адаптації та покрокові вказівки для нових працівників в залежності від конкретних відділів, ролей, типів зайнятості тощо.

Oracle Recruiting

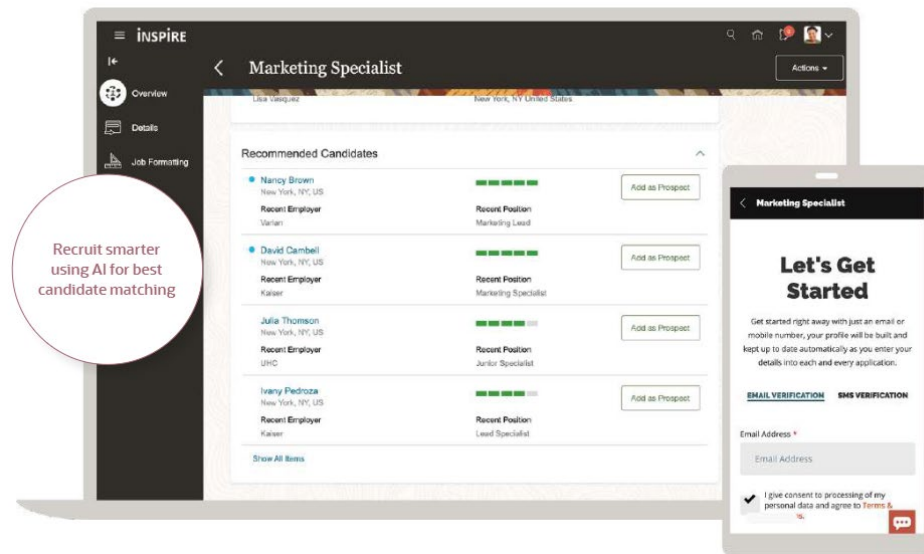


Рис. 1.2 – Інтерфейс програмного додатку Oracle Recruiting, центр рекрутингової діяльності [14]

Окремо необхідно виділити таку особливість додатку як рекрутинг на основі оцінки навичок. Автоматизоване рішення Oracle Recruiting на базі штучного інтелекту допомагає зберігати актуальну інформацію про навички кандидатів, що подали заявку, та допомагає виявляти прогалини та слабкі місця. Тому кандидатам, як зовнішнім, так і внутрішнім в компанії, надаються рекомендації щодо навичок, які вони можуть додати до заявок на вакансії або свого профілю працівника, щоб виділяти релевантні кваліфікації. Також рекрутинговим командам надається підтримка у визначенні навичок, які варто включати в оголошення про роботу та описи посад, базуючись на навичках топ-виконавців у схожих ролях.

В додатку Oracle Recruiting генеративний штучний інтелект використовується для автоматизованої сумаризації навичок та досвіду кандидатів, що допомагає рекрутерам швидко та об'єктивно оцінити сумарний потенціал претендентів на

вакантні посади, див рис. 1.3. Це забезпечує більш швидкий та ефективний процес відбору, дозволяючи рекрутерам сконцентруватися на стратегічних аспектах рекрутингу та вибору кандидатів.

Table 1: Examples of Oracle's new generative AI capabilities in Fusion Applications

	Assisted Authoring	Suggestions	Summarization
Fusion HCM			
	Employee profile	Survey questions	Performance review summary
	Employee recognition	Competency development tips	Candidate qualification
	Goal creation		
	Job description + requisition		
	Knowledge articles		
Fusion CX			
	Service agent responses	Field service recommendations	Customer engagement summaries
	Knowledge articles		
	Search augmentation		
	Administrator guidance		
Fusion ERP			
	Management reporting narrative		
	Financial reporting narrative		
	Journal entry, reconciliation notes		
	Financial data explanations		
Fusion SCM			
	Item descriptions, attributes		
	Supplier suggestions for negotiations		

Рис. 1.3 – Впровадження генеративного штучного інтелекту як елементу SMART-аналізу в програмних додатках Oracle Fusion Cloud Applications [15]

Компанія IBM – багатонаціональна електронна корпорація, є одним зі світових виробників програмного забезпечення та персональних комп'ютерів, провайдером глобальних інформаційних мереж. Спеціалізується на операційних та файлових системах, мікропроцесорах, серверах, суперкомп'ютерах, когнітивних обчисленнях.

IBM Watson – суперкомп'ютер, розроблений компанією IBM, який поєднує можливості штучного інтелекту, автоматизацію та когнітивні обчислення для забезпечення оптимальної продуктивності як машини, що може відповідати на будь-яке запитання. У статті восени 2010 року, опублікованій в журналі AI Magazine, дослідники компанії IBM повідомили про створення комп'ютерної системи штучного інтелекту, яка могла б конкурувати з людьми у правильних відповідях на запитання в режимі реального часу в телевізійній вікторині “Jeopardy!”. IBM Watson кинув виклик двом чемпіонам з найвищим рейтингом гри у вікторині “Jeopardy!”. Відома перемога IBM Watson в 2011 році викликала значний інтерес і стала важливим кроком у

розвитку штучного інтелекту як елементу SMART-аналізу. Саме це продемонструвало потенціал і можливості штучного інтелекту в обробці та розумінні великої кількості неструктурованих даних, що можуть бути корисними для вирішення бізнес-задач підвищеної складності.

IBM Watson працював на кластері з десятків серверів IBM Power 750 – 90, мав 2880 процесорних ядер, 15 терабайт операційної пам'яті, операційну систему SUSE Enterprise Linux Server, Apache Unstructured Information Management Architecture та Apache Hadoop для аналізу неструктурованих даних та обробки великих наборів даних у розподілений спосіб [17].

Далі, приблизно в 2013 році, перехід технології IBM Watson до хмарної платформи відкрив значні можливості для інновацій та розвитку екосистеми програмного забезпечення для підприємств. Стартапи, компанії, що розвиваються, а також такі великі компанії як Autodesk, Chevrolet змогли використовувати можливості IBM Watson для створення нових рішень у таких областях, як медицина, фінанси, маркетинг, логістика та інші. Протягом 10 наступних років до можливостей IBM Watson додається ряд наступних функцій: IBM Watson Discovery Advisor для знаходження паттернів в необроблених даних, бібліотека IBM Watson Natural Language Processing для об'єднання додатків для природної обробки мови в один стек для всіх програмних додатків IBM та бета-версія віртуального асистенту IBM Watson Assistant для класифікації намірів користувачів [18].

IBM Watsonx – хмарна платформа штучного інтелекту та даних, що була анонсована 9 травня 2023 року на конференції “Think Conference” компанії IBM. Загалом складається з 3 окремих продуктів [19], які побудовані на єдиній платформі:

- IBM watsonx.ai – студія для навчання, налаштування, впровадження як традиційних моделей машинного навчання, так і нових великих мовних моделей (генеративного штучного інтелекту) через відкритий та інтуїтивно зрозумілий користувацький інтерфейс;

- IBM watsonx.data – сховище даних, що ґрунтується на відкритій архітектурі lakehouse, спеціально оптимізоване для управління даними та високопродуктивних обчислень для моделей штучного інтелекту;
- IBM watsonx.governance – набір інструментів для управління етичним штучним інтелектом для забезпечення надійних бізнес-процесів.

Приклад інтерфейсу платформи IBM Watsonx можна побачити на рис 1.4.

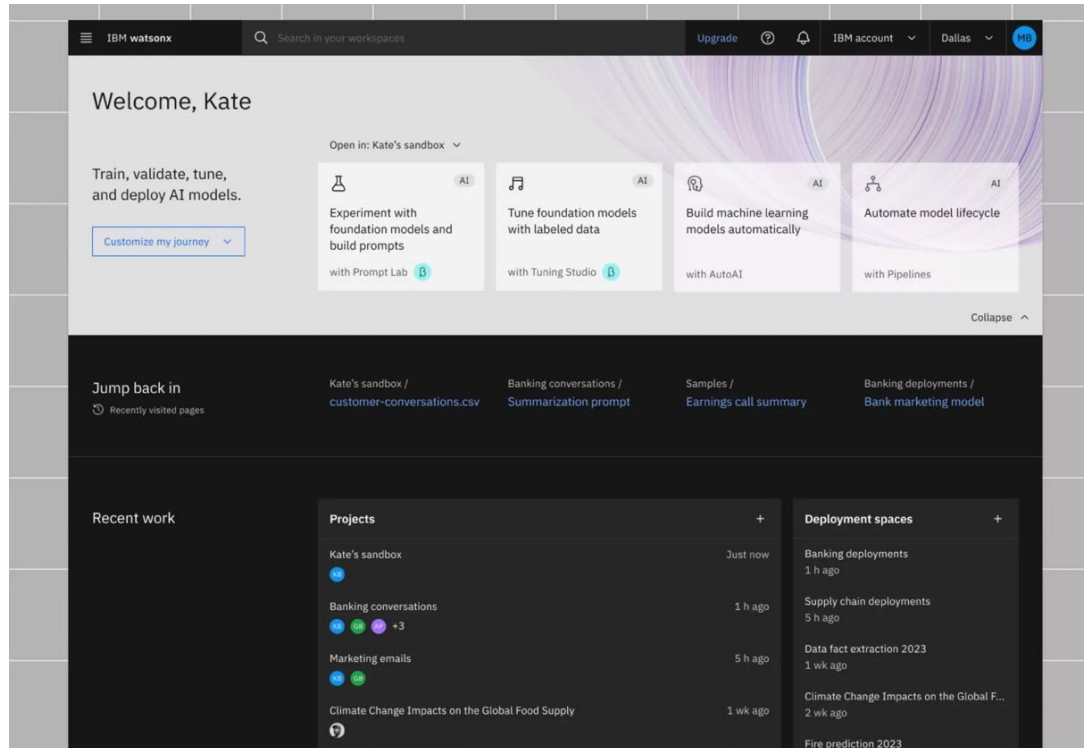


Рис. 1.4 – Інтерфейс платформи штучного інтелекту та даних як елементу SMART-аналізу IBM Watsonx [20]

Варто також вказати, що команда, яка займається дистрибутивом для мови програмування Python Anaconda, почала співпрацювати з компанією IBM 13 лютого 2024 року, щоб існувала можливість вбудовувати кастомні пакети мови програмування Python з відкритим кодом до платформи IBM Watsonx, зберігаючи конфіденційність і безпеку корпоративних даних [21].

IBM Watsonx Orchestrate – це платформа для автоматизації, оркестрації та управління задач та процесів управління людськими ресурсами [22]. Особливостями платформи є:

1. Попередньо створені “навички”: “навички” в контексті IBM Watsonx Orchestrate є дії, які можна автоматизувати (наприклад, надіслати листа електронною поштою, завантажити файл, створити опис для вакансії тощо). IBM Watsonx Orchestrate використовує обробку природної мови, щоб отримати з каталогу бази даних базові та розширені “навички” для виконання запитів для автоматизації без спеціальної підготовки, див. рис. 1.5.;
2. Автоматична побудова “навичок” та тренування платформи: існує можливість імпортувати вже існуючі засоби автоматизації як “навички” та комбінувати їх з раніше створеними або новими засобами автоматизації та за допомогою Unified Automation Builder створювати нові “навички” для унікальних бізнес-завдань в сфері управління людськими ресурсами;
3. Сумісність з великою кількістю додатків та екосистем додатків: IBM Watsonx Orchestrate може підключатися до різноманітних додатків та інструментів для безперервної роботи з продуктами Salesforce, Workday, Outlook, Gmail та інших інструментів для виконання завдань в зручному інтерфейсі без коду;
4. SMART-віртуальний асистент: на платформі IBM Watsonx Orchestrate можливо розгортати віртуальних асистентів для рекомендації релевантного контенту.

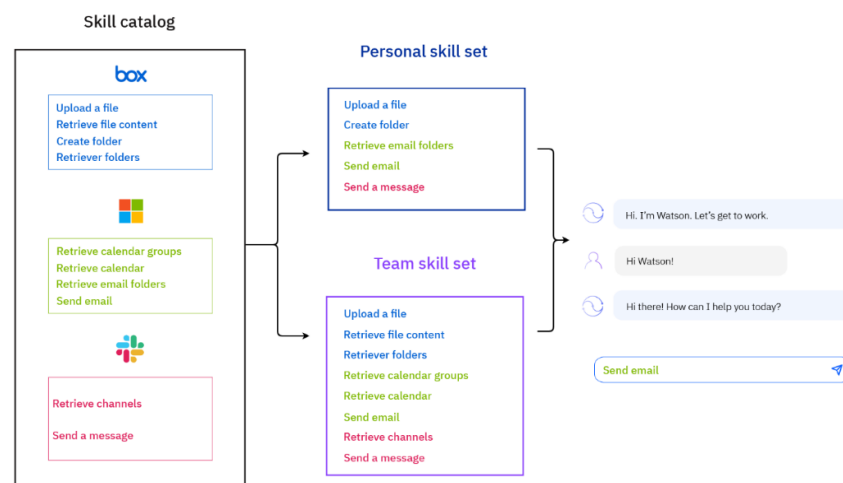


Рис. 1.5 – процес створення “навичок” в IBM Watsonx Orchestrate [23]

За допомогою платформи IBM Watsonx Orchestrate спеціалісти із залучення талантів можуть делегувати повторювані та повсякденні завдання та зосередити свою енергію на більш стратегічних аспектах залучення талантів, що сприяє підвищенню продуктивності, кращому найму та досвіду кандидатів. Платформа може шукати кандидатів в мережах LinkedIn, Glassdoor тощо, використовуючи ключові слова. В контексті оцінювання навичок кандидатів IBM Watsonx Orchestrate надає список найбільш підходящих кандидатів, їх профілів та резюме, потім надсилає повідомлення в реальному часі, коли кваліфіковані кандидати з релевантними навичками подають заявку. Також IBM Watsonx Orchestrate має можливість ділитися профілями кандидатів в команді та відстежувати будь-які оновлення статусів кандидатів в мережах [24].

SAP – німецька корпорація, розробник програмного забезпечення, також надає консалтингові послуги. Є одним з найбільших розробників корпоративного програмного забезпечення в Європі.

SAP SuccessFactors Human Experience Management – хмарне рішення у сфері управління людськими ресурсами, наприклад, система управління людськими ресурсами (HRMS), яка поєднує основні відділи кадрів та заробітну плату, управління талантами, управління ефективністю продажів, аналітику персоналу та планування робочої сили завдяки штучному інтелекту, розробленим для задоволення індивідуальних потреб та сприяння гнучкості організації в масштабі [25]. SAP SuccessFactors Human Experience Management пропонує широкий набір інструментів для автоматизації та оптимізації бізнес-процесів, надає можливості для стратегічного управління талантами, включаючи розробку кар'єрних планів, планування наступництва та оцінку потенціалу працівників, допомагає створювати індивідуальні навчальні плани для працівників, відстежувати їхні досягнення, забезпечувати постійний розвиток персоналу та надає засоби для аналізу даних щодо працівників, продуктивності та задоволення роботою для прийняття обґрунтованих рішень. Також рішення має ефективний та зручний інтерфейс для роботи, див. 1.6.

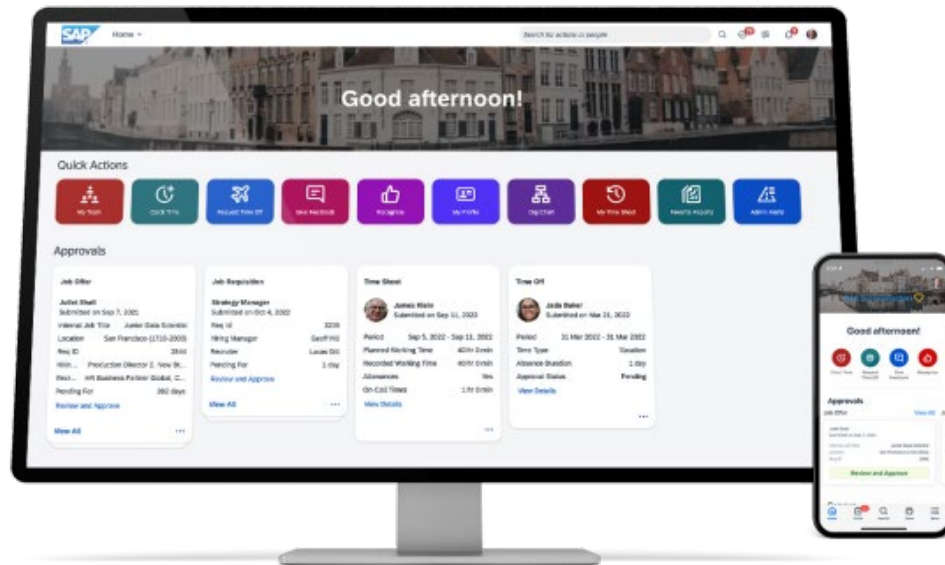


Рис. 1.6 – Інтерфейс рішення SAP SuccessFactors Human Experience Management [26]

Корпорація SAP активно впроваджує штучний інтелект та машинне навчання в свою екосистему додатків, зокрема в SAP SuccessFactors Human Experience Management. Наприклад, генеративний штучний інтелект використовується для створення опису вакансії та запитань для співбесіди. Віртуальний асистент Joule допомагає у вирішенні бізнес-завдань підвищеної складності для управління людськими ресурсами в розмовному контексті, адже він має 3 розмовні патерни: навігаційний (допомагає користувачам перейти до функціональних можливостей, які вони шукають), транзакційний (допомагає користувачам ефективно виконувати їхні завдання) та інформаційний (допомагає користувачам отримати інформацію з існуючих джерел) [27, 28].

SAP SuccessFactors Recruiting – програмний додаток орієнтований виключно на процес рекрутингу кандидатів. Додаток поєднує систему відстеження кандидатів (ATS), розподілу вакансій та встановлення взаємовідносин з кандидатами на основі штучного інтелекту як елементу SMART-аналізу, надійної системи навичок, автоматизації та аналітичної інформації, що базується на даних [29]. Особливостями додатку є:

1. Оптимізація процесу рекрутингу: SAP SuccessFactors Recruiting підтримує організації будь-якого розміру та може залучати таланти на основі навичок, масових розсилок, підтримки брендингу роботодавця за допомогою автоматизованих робочих процесів, інтелектуальних засобів штучного інтелекту та вбудованої аналітики;
2. Всебічне охоплення талантів: використовуючи багатоканальний підхід до глобального охоплення, додаток має доступ до понад 4000 робочих мереж та сайтів (LinkedIn, Indeed, Meetup тощо) у більш ніж 80 країнах, що значно збільшує обсяг та різноманітність кандидатів, забезпечуючи рівність та об'єктивний процес подачі заявок. Також наявна власна пошукова оптимізація, яка автоматично покращує рейтинг сайту в основних пошукових системах, а система з гнучкими правилами автоматизує розміщення зовнішніх і внутрішніх вакансій;
3. Забезпечення позитивного досвіду кандидатів: додаток SAP SuccessFactors Recruiting допомагає з налаштуванням параметрів заявок кандидатів, підтримує зв'язок з потенційними талантами через спільноти талантів, сповіщення про роботу та нові вакансії, соціальні зв'язки, має інтуїтивно зрозумілий спосіб навігації, пошуку та подання заявки на посаду. Також існує можливість надавати кандидатам контактні точки для зворотного зв'язку, щоб оптимізувати цілий процес найму – від заявки до пропозиції;
4. Підвищення ефективності професіоналів з рекрутингу: додаток має можливість створювати консолідований профіль кандидата з інформацією про минулу діяльність кандидата, його комунікації з компанією, відгуки про співбесіду. Запроваджена фільтрація кандидатів та налаштовувані інформаційні панелі для візуалізації статусу кандидатів. Інтелектуальні засоби забезпечують залучення найкращих кандидатів за допомогою цифрових пропозицій, керування схваленнями та ініціювання онбордингу.

В контексті оцінки навичок SAP SuccessFactors Recruiting надає цілісне уявлення про навички всієї організації та дає змогу командам із залучення талантів заповнювати прогалини в талантах шляхом найму. Завдяки рекомендаціям на основі штучного інтелекту професіонали з найму можуть легко створювати переконливі описи посад, які відображатимуть відповідні навички та компетенції, необхідні для роботи. Оцінювання навичок і рейтинг компетенцій дозволяють порівнювати найкращих кандидатів, а вбудовані можливості в Microsoft Teams дозволяють легко планувати співбесіди, автоматизувати нагадування, допомогу штучного інтелекту та відгуки про співбесіди в режимі реального часу – і все це в межах робочого процесу, що робить кожну співбесіду цінною та послідовною.

РОЗДІЛ 2

АНАЛІЗ АЛГОРИТМІВ ДЛЯ ОЦІНЮВАННЯ НАВИЧОК КАНДИДАТІВ

2.1 Основні принципи, методи та характеристика ефективності алгоритмів для оцінювання навичок кандидатів

Попереднє планування процесу найму персоналу для оцінювання навичок кандидатів є першим та одним з ключових аспектів підбору, який має вирішальне значення для створення життєздатної основи для ефективного підбору кандидатів. Ця фаза включає визначення вимог до найму персоналу, визначення стратегій оцінювання навичок, окреслення посадових інструкцій та фінансових переваг. Важливо вказати, що ця фаза встановлює взаємозв'язок між процесом найму, оцінюванням навичок та стратегічними цілями організації; дозволяє визначити прогалини в навичках, проаналізувати плинність кадрів та показники утримання, що дозволяє приймати обґрунтовані управлінські рішення.

Другим етапом оцінювання навичок є отримання великого масиву даних, тобто резюме, що містять всю необхідну інформацію про навичку, освіту, досвід роботи, досягнення кожного кандидата тощо. Збір резюме може бути автоматизованим через використання різноманітних технічних засобів, таких як системи управління кандидатами (наприклад, Rise, Pinpoint, BambooHR), робочі соціальні мережі (наприклад, LinkedIn, Glassdoor, Indeed, Dice), власні онлайн-платформи для збору резюме. Ці інструменти допомагають у впорядкуванні великої кількості даних про кандидатів. Завдяки цьому можна швидше та ефективніше відбирати найбільш підходящі резюме для подальшого розгляду.

Далі, третім етапом оцінювання навичок кандидатів є власне етап відбору: він передбачає складання короткого списку з невеликою кількістю претендентів на

вакантну посаду з великої групи кандидатів. Цей етап вимагає уважного вивчення резюме та інших даних про кандидатів, щоб відібрати найбільш спроможних кандидатів для подальшого співбесіди. На етапі відбору професіонал з управління людськими ресурсами аналізує інформацію, надану кандидатами, щоб визначити, які з них найбільш відповідають вимогам вакансії та мають релевантні навички та досвід.

Традиційні методи етапу відбору включають ручний систематичний відбір та ранжування кандидатів.

Ручний систематичний відбір – традиційний метод етапу відбору, що передбачає ретельне та систематичне переглядання та оцінку резюме та навичок кандидатів на вакантну позицію роботодавцем або професіоналом з управління людськими ресурсами [30]. Він включає визначення ключових навичок та кваліфікації, необхідних для роботи; розбивку резюме за категоріями (освіта, досвід роботи, інтереси тощо), для полегшення порівняння кандидатів і швидкий пошук важливої інформації; аналіз історії роботи кандидата для уявлення про його професійне зростання та стабільність; перевірка рекомендацій та відгуків від колишніх роботодавців і колег для отримання інформації про стиль роботи та здібності кандидата; перегляд граматичних помилок, форматування та загального вигляду резюме для визначення професіоналізму та інтерес до вакантної посади [31]. Також, ручний систематичний відбір може включати створення чек-листа з критеріями та оцінювання кандидатів з точки зору врахування цих критеріїв.

Ранжування кандидатів – традиційний метод етапу відбору за допомогою експертних оцінок за участі одного або декількох експертів, тобто професіоналів з управління людськими ресурсами або роботодавців. Експерти оцінюють кандидатів на основі критеріїв та встановлюють порядок їх пріоритетності. Експертна оцінка може бути проведена шляхом індивідуального розгляду кожного кандидата або шляхом колективного обговорення результатів оцінки. Після цього встановлюється порядок кандидатів залежно від їх оцінок за допомогою методу медіан рангів, середніх арифметичних рангів, Дельфі та оцінки узгодженості за допомогою

коефіцієнта кореляції, конкордації, Кендала тощо.

Проте, традиційні методи етапу відбору можуть призвести до погіршення найму кваліфікованих та потенційно підходящих кандидатів через наявність необ'єктивного ставлення професіоналів з управління людськими ресурсами або роботодавців, неповноти оціночних методів, надмірної втрати часу та зусиль. Крім того, ці процеси є трудомісткими через велику кількість заявок, отриманих на одну вакансію в ІТ-індустрії. Наприклад, згідно даних опитування спільноти програмістів DOU.ua, на одну вакансію на технічну посаду в ІТ-індустрії в листопаді 2023 року може припадати від 5 до 119 заявок, не технічну посаду від 10 до 71 заявок, див рис. 2.1, 2.2.

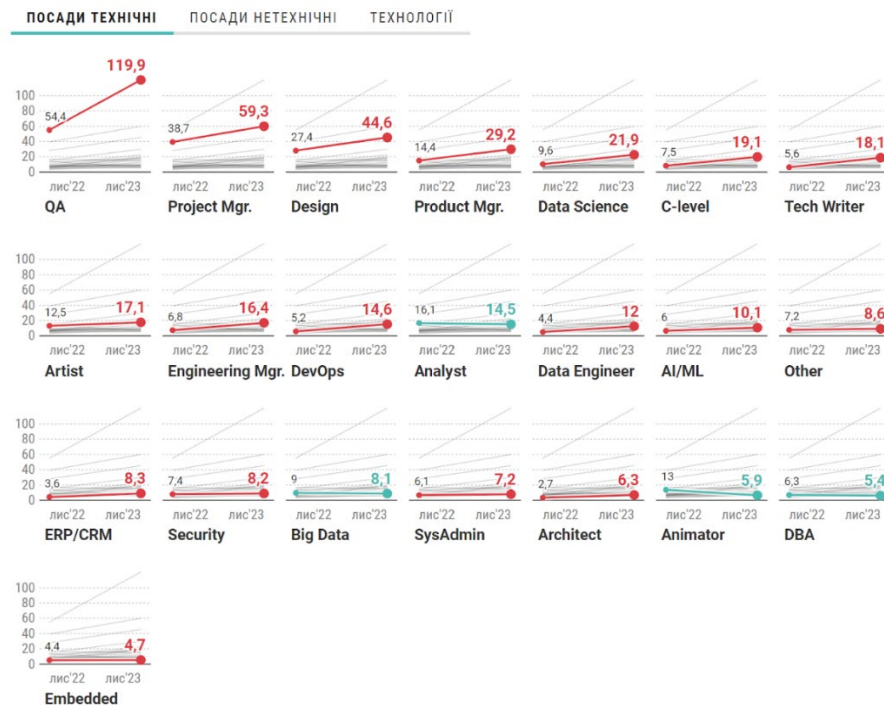


Рис. 2.1 – Дані щодо кількості заявок на технічні посади в ІТ-індустрії, листопад 2023 рік [32]



Рис. 2.2 – Дані щодо кількості заявок на не технічні посади в ІТ-індустрії, листопад 2023 рік [32]

Щоб вирішити ці проблеми, більшість ІТ компаній переходять на використання більш ефективних методів етапу відбору для оцінювання навичок кандидатів.

Автоматизований відбір – інноваційний метод етапу відбору, що передбачає використання технологій та методів SMART-аналізу для автоматизації процесу оцінки резюме та навичок кандидатів на вакантну позицію роботодавцем або професіоналом з управління людськими ресурсами. Головною перевагою автоматизованого відбору є можливість швидко та ефективно обробляти великий обсяг заявок, що дозволяє значно прискорити процес відбору. Такі системи можуть автоматично аналізувати резюме кандидатів на відповідність навичкам – це дозволяє зменшити вплив людського фактору та забезпечити більш об'єктивний процес відбору.

Важливо відмітити, що резюме – це єдиний уніфікований документ, у якому кандидат підсумовує життєво важливі факти та інформацію про всю свою професійну кар'єру: навички, особисті дані, освіту, історію роботи, проекти, завдання, досягнення тощо. Існують багато варіацій в структурі, змісті та стилях резюме різними мовами, країнами походження, функціональними сферами (наприклад, інженерія, фінанси, маркетинг, продажі, кадри тощо), а також промисловими сферами. Резюме є структурований (містять розділи, таблиці, колонтитули та іноді зображення) або

напівструктурований, оформлений (з різними шрифтами, індикаторами виділення, кольорами) та творчо написаний документ [33].

У загальному, автоматизований відбір базується на використанні обробки природної мови для оцінювання навичок кандидатів. Обробка природної мови – наука, міждисциплінарна сфера, яка вивчає та поєднує комп'ютерну лінгвістику – моделювання людської мови – зі статистичними моделями та методами SMART-аналізу для розуміння, аналізу та генерування тексту та мови [34].

Видобування інформації – задача у сфері обробки природної мови, яка включає вилучення структурованих даних із неструктурованих або напівструктурованих текстових даних. Рішення цієї задачі значно спрощує процес аналізу великих обсягів текстових даних, що дозволяє отримувати цінну інформацію. Видобування інформації може включати в себе різноманітні завдання, такі як виявлення іменованих сутностей, фактів, подій або відносин між ними [35].

Першим етапом автоматизованого відбору є парсинг інформації з резюме кандидатів за допомогою парсерів, оскільки різноманітність резюме разом із використанням специфічної термінології, скорочення, використання фрагментів тексту, орфографічні та граматичні помилки тощо ускладнюють автоматизовану обробку резюме [36].

Синтаксичний парсер на основі ключових слів є найбільш поширеним та порівняно простим автоматизованим методом. Цей метод полягає у визначенні ключових слів, фраз або шаблонів у тексті резюме, які вказують на наявність певних навичок, досвіду або кваліфікацій кандидатів. Ключові слова можуть включати назви конкретних навичок, інструментів, технологій, освіти та інших релевантних даних. Хоча цей метод є відносно простим у реалізації, його ефективність залежить від якості і точності набору ключових слів та їхніх поєднань. Наприклад, якщо синтаксичний парсер шукає рік роботи, він може шукати шаблон із чотирма цифрами. Однак це може бути помилкою, оскільки це число також може бути роком народження, поштовим індексом тощо. Те саме стосується слів, які мають декілька застосувань у

резюме, наприклад навичка “економіка” (та її похідні слова) може бути частиною назви посади, досвіду роботи або входити до будь-якої кількості інших категорій [37]. Крім того, синтаксичні парсери на основі ключових слів можуть бути вразливими до різних мовних варіацій, синонімів та контекстуальних викликів, що можуть призводити до помилкових результатів або пропусків важливої та релевантної інформації.

Парсер на основі граматики покладається на використання ряду граматичних правил для інтерпретації різноманітної інформації. Цей підхід зазвичай полягає в створенні формальних граматик або правил, які описують синтаксичну та семантичну структуру резюме. Потім ці правила застосовуються до тексту резюме для виявлення та виділення текстових блоків та сутностей, таких як імена, контактна інформація, досвід роботи, навчання тощо. Наприклад, якщо вимагається, щоб кандидати вказували навичку “відповідальність” як основну характеристику, парсер на основі граматики буде виділяти лише тих кандидатів, які використовують цю навичку у відповідному контексті, замість того, щоб включати такі фрази, як “відповідальний за”, у розділі про досвід роботи [38]. Проте, ці парсери можуть бути обмежені в здатності розпізнавати або адаптуватися до нових шаблонів або форматів резюме – наприклад, коли резюме використовує нестандартні структури або вираження. Також, парсери на основі граматики часто вимагають точні граматичні правила або шаблони, що може призвести до втрати інформації або недостовірних результатів, якщо текст резюме не відповідає цим правилам; вимагається значний час та експертні знання для того, щоб адаптувати парсер до різноманітних видів резюме та мов.

Статистичний парсер є інструментом для виділення релевантної інформації за допомогою статистичних методів та методів машинного навчання. Його функціональність полягає в аналізі структури та змісту резюме з метою парсингу ключових деталей, таких як особиста інформація, досвід роботи, освіта, навички, досягнення тощо. Унікальність статистичних парсерів полягає в їх здатності постійно вдосконалювати точність за рахунок використання методів машинного навчання, які

дозволяють адаптуватися до нових шаблонів та форматів у резюме [38]. Незважаючи на свої переваги, статистичні парсери також мають свої недоліки: статистичні парсери можуть неправильно інтерпретувати чи пропускати деякі деталі у резюме через складнощі в розпізнаванні або обробці певних форматів резюме або мовних виразів; процес адаптації до шаблонів статистичних парсерів може бути досить часоємним, особливо якщо нові шаблони суттєво відрізняються від тих, на яких був навчений парсер; точність роботи статистичного парсера значно залежить від якості та чистоти вхідних даних – некоректно введені або неповні дані можуть призвести до помилкових висновків або втрати важливої інформації.

Другим етапом автоматизованого відбору є перетворення категоризованої інформації в вектори. Для подальшого автоматичного аналізу за допомогою методів SMART-аналізу підхід Word Embedding дозволяє узагальнювати знання про слова, враховуючи їх контексти у тексті. На сьогоднішній день, Word Embedding є одним із основних напрямків дослідження обробки природної мови.

Word Embedding – це числове представлення слів у просторі нижчого виміру, що здатне фіксувати синтаксичну та семантичну інформацію [39]. Word Embedding є особливим типом представлення слова, що дозволяє однорідну репрезентацію слів із пов'язаними значеннями. Згідно з “теорією розподілу”, слова, які мають подібний контекст, як правило, мають пов'язані або подібні представлення [40]. Роботу моделей з перетворення слів та тексту загалом на Word Embeddings з урахуванням їх синтаксичної та семантичної інформації можна представити у вигляді кластерів, див. рис. 2.3.

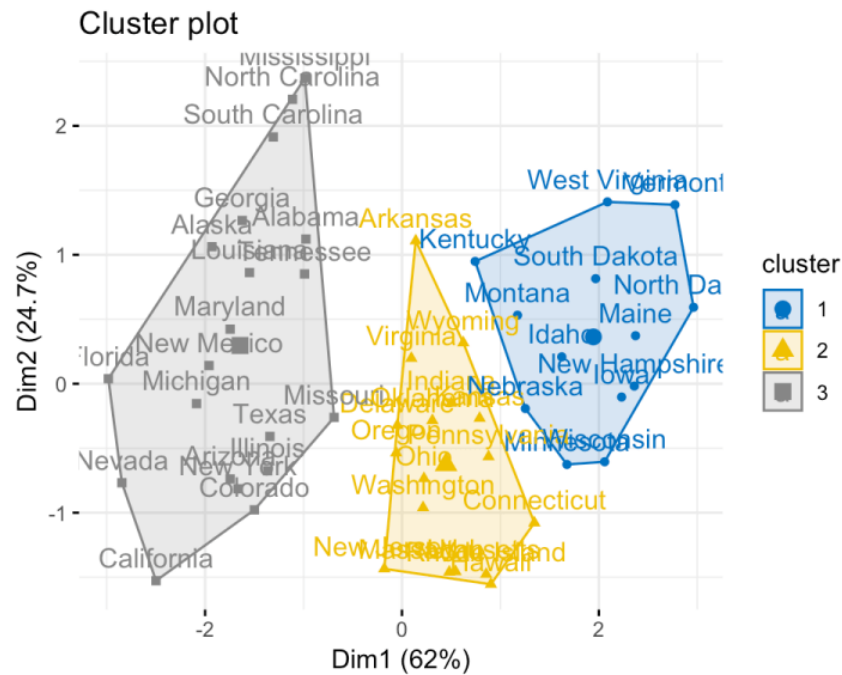


Рис. 2.3 – Кластери Word Embeddings та їх взаємозв'язок [41]

Наприклад, слова “кава”, “чай” та “пиво”: розглянемо два виміри – один для того, скільки напій містить кофеїн, а інший – для того, наскільки напій є алкогольним. Можливе представлення слова “кава” як вектора $[1,0, -1,0]$, слова “чай” як вектора $[0,5, -1,0]$ та слова “пиво” як вектора $[-1,0, 1,0]$ – саме ці вектори фіксують ключові характеристики цих напоїв, див. рис. 2.4.

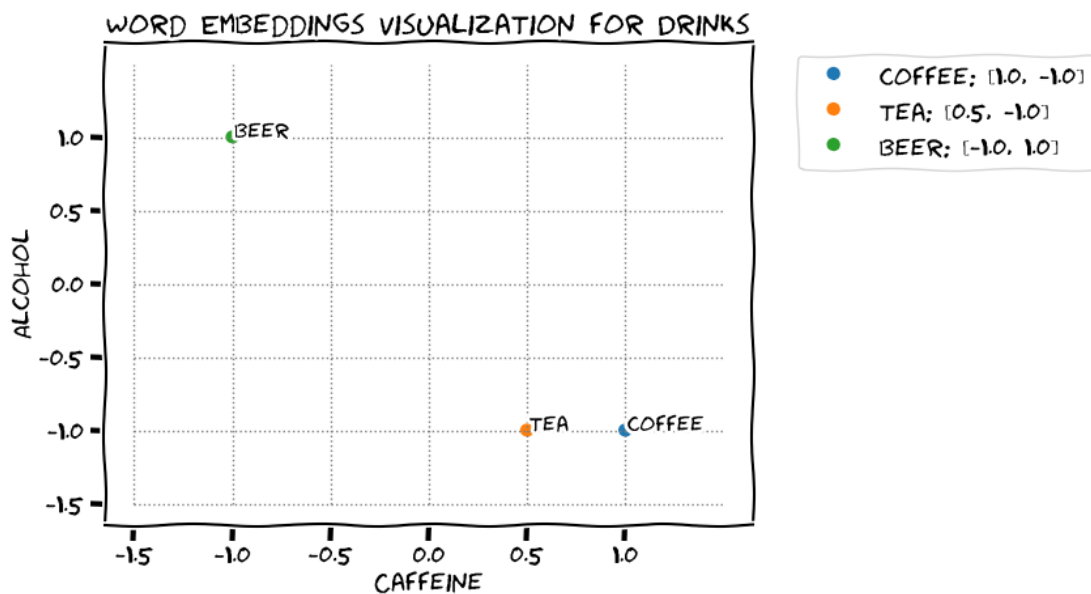


Рис. 2.4 – Word Embeddings у двовимірному векторному просторі [42]

У двовимірному векторному просторі слова “кава” та “чай” будуть ближче одне до одного, ніж “пиво”. Це вказує на те, що кава та чай є безалкогольними напоями з кофеїном, тоді як пиво є алкогольним напоєм без кофеїну.

Перш за все, існують традиційні методи Word Embeddings:

1. Модель Bag-of-words – це один із найбільш простих способів представити текст у векторній формі на основі частоти кожного слова в тексті. Концепція моделі полягає в тому, щоб розглядати весь документ, текст (або фрагмент тексту) як “мішок” (набір) його слів, нехтуючи граматику та порядком слів. Частота кожного слова в тексті використовується для створення Word Embeddings [43]. Першим кроком роботи моделі є токенизація або розбиття тексту на окремі слова або лексеми. Знаки пунктуації та інші символи, що не є алфавітно-цифровими, зазвичай видаляються, а слова стандартизуються (у нижній регістр), щоб гарантувати, що варіанти того самого слова розглядаються як ідентичні. Далі створюється словник шляхом збору всіх унікальних слів із усього тексту документу. Кожному слову в цьому словнику присвоєно унікальний індекс або ідентифікатор. Потім для тексту створюється вектор, де кожен елемент відповідає слову в словнику, а значення елемента представляє частоту цього слова в документі. Це призводить до розрідженого вектора, де більшість елементів дорівнює нулю, оскільки більшість слів у словнику не з’являється в іншій частині тексту, див. рис. 2.5.

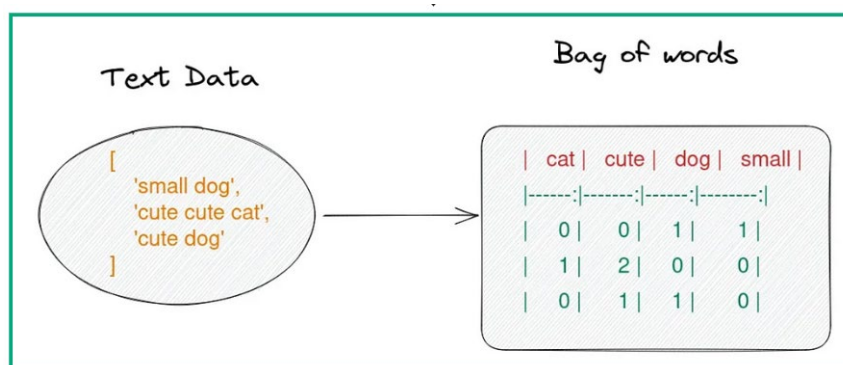


Рис. 2.5 – Створення Word Embeddings за допомогою моделі Bag-of-words

Для виділення ключових навичок та перетворення навичок у дискретні представлення, модифікована модель Bag-of-words досягає приблизно 62.60% точності [45].

2. Модель Term frequency-inverse document frequency (TF-IDF) перетворює текст у вектор ознак, де кожна ознака представляє неперервну послідовність із n слів в документі. Модель поєднує два показники: TF (частота терміну) та IDF (зворотня частота в документі). Ваги, пов'язані з кожною ознакою, обчислюється як частота терміну, помножена на його зворотню частоту документа. Висока вага ознаки досягається високою частотою терміну в документі та низькою частотою документа в тексті [46]. Модель TF-IDF працює на основі статистичної міри пошуку релевантності слова в тексті, який може бути у вигляді тексту одного документа або різних документів, які називаються корпусом [47]. Оцінка частоти терміну (TF) вимірює частоту слів у певному документі, див. формулу (2.1).

Формула для оцінки частоти терміну (TF):

$$TF(t, d) = \frac{count(t, d)}{count(d)}, \quad (2.1)$$

де t – термін;

d – документ.

Зворотна частота в документі (IDF) вимірює рідкість слів у тексті, див. формулу (2.2) – йому надається більше значення, ніж оцінці частоти терміну, тому що, незважаючи на те, що оцінка частоти терміну (TF) надає більшу вагу словам, які часто зустрічаються, зворотня частота в документі (IDF) фокусується на рідко використовуваних словах у корпусі, які можуть містити важливу інформацію [48].

Формула для оцінки зворотної частоти в документі (IDF):

$$IDF(t, d) = \ln \frac{N}{n_t}, \quad (2.2)$$

де N – загальна кількість документів;

$n(t)$ – кількість документів, що містять термін t .

Формула для обчислення показника TF-IDF в документі:

$$TF - IDF(t, d) = TF(t, d) * IDF(t, d) \quad (2.3)$$

В поєднанні з такими методами SMART-аналізу, як k-nearest neighbours (KNN), опорними векторними машинами, логістичною регресією, точність моделі TF-IDF може сягати до 80% у задачі оцінки навичок кандидатів та їх відповідності до вакансії [49].

Незважаючи на свою простоту, традиційні методи мають істотні недоліки: вони не враховують порядок слів у тексті, тобто мають дискретне представлення (sparse representation), що може бути критичним у випадках, коли порядок слова має значення для семантики або інтерпретації тексту; у великих корпусах текстів зазвичай міститься більша кількість унікальних слів, що може призвести до розріджених векторів ознак, що ускладнює обробку та зберігання даних; не враховуються семантичні відносини між словами.

На відміну від традиційних методів та їх відповідних дискретних представлень, існують неперервні (щільні) представлення (dense representation), які зазвичай дозволяють працювати з більш широким спектром текстових даних, забезпечуючи кращу узагальнювальну здатність моделей для оцінювання навичок кандидатів. Щільні представлення є векторами з дійсними числами, на противагу двійковому або категоріальному представленню. Кожен вимір представляє особливість або характеристику, а значення в цьому вимірі є фактичним числом. Щільні представлення враховують семантичні зв'язки між словами. Також, такі представлення зазвичай мають меншу розмірність [50].

Для отримання щільних представлень існують статичні методи Word Embeddings:

1. Word2Vec є алгоритмом для перетворення слів у щільні представлення, який був розроблений компанією Google в 2013 році. Фактично, Word2Vec є алгоритмом навчання без вчителя за допомогою двошарової навченої нейронної мережі для побудови семантичного контексту корпусу тексту у векторній формі. Двошарова

нейронна мережа може швидко розпізнавати семантичну подібність та ідентифікувати слова-синоніми за допомогою методів логістичної регресії, що робить її швидше, ніж глибокі нейронні мережі. Це робить обробку більш оперативною та прозорою [51]. На вхід Word2Vec приймає великий корпус тексту та генерує векторний простір із сотнями вимірів. Кожному унікальному слову в корпусі присвоюється вектор у цьому просторі. Розробка слова до вектора передбачає аналіз вивчених векторів та дослідження того, як ними можна маніпулювати за допомогою векторного аналізу. Наприклад, віднімаючи “чоловічість” від “короля” та додаючи “жіночність”, у результаті отримаємо слово “королева”, яке відображає таку аналогію: “Король для королеви, як чоловік для жінки”, див. рис. 2.6.

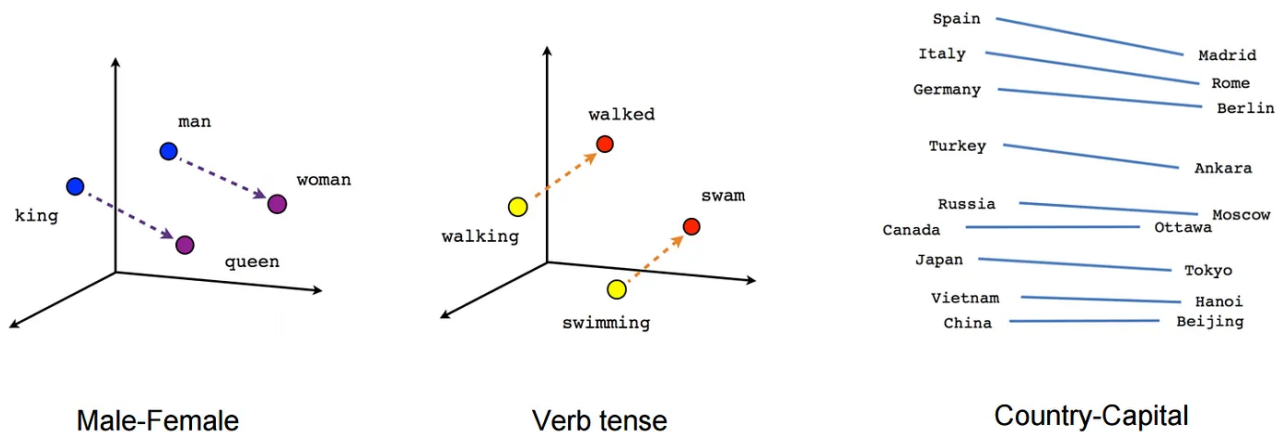


Рис. 2.6 – Співвідношення “Король для королеви, як чоловік для жінки” [52]

Для представлення конкретного слова як вектору в багатовимірному просторі, алгоритм використовує один із двох способів: Continuous bag-of-words або Skip-gram.

Continuous bag-of-words базується на думці, що сусідні слова, які з’являються близько в тексті, мають бути дуже схожими за значенням, а слова, які з’являються далеко одне від одного, часто відрізняються [53]. Тому очікується, що з’явиться центральне слово, враховуючи умови появи сусідніх слова, див. рис. 2.7.

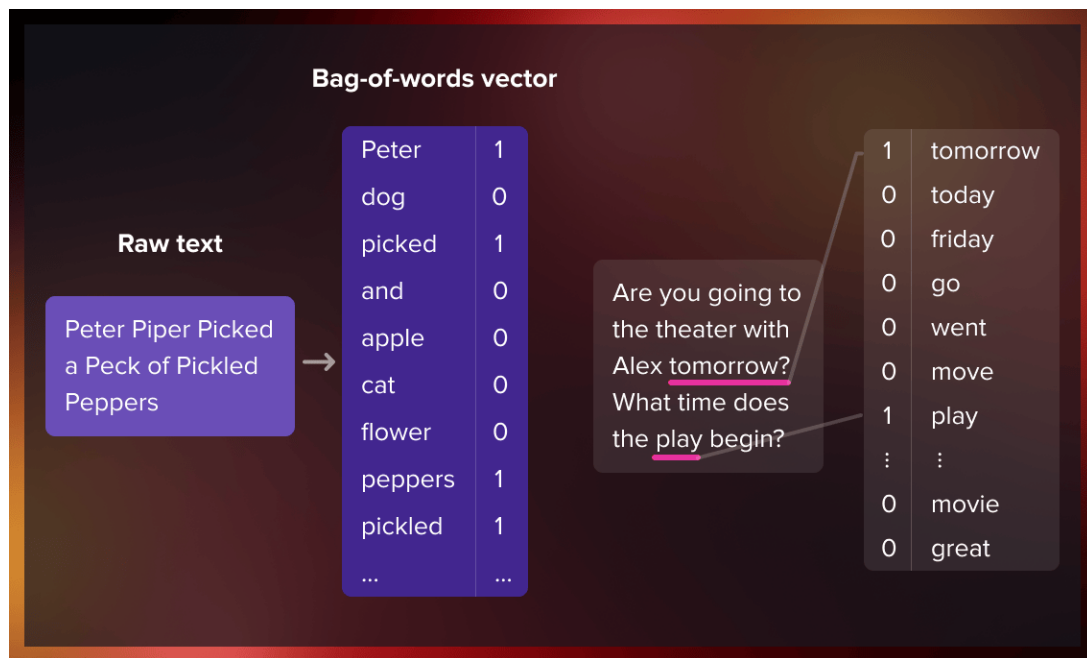


Рис. 2.7 – Робота способу Continuous bag-of-words на невеликому корпусі тексту [51]

Skip-gram навпаки намагається передбачити слова, що оточують деяке центральне слово в тексті, див. рис. 2.8.

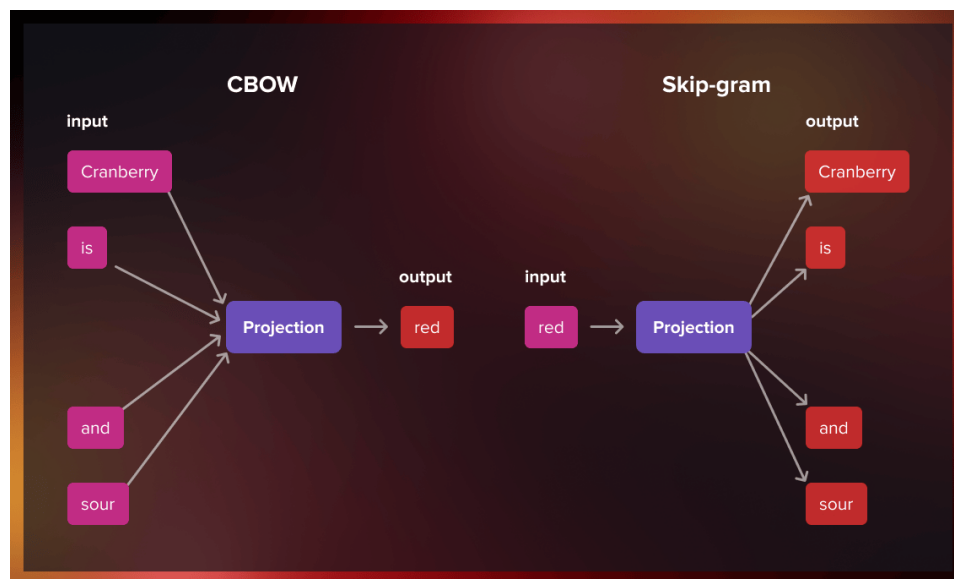


Рис. 2.8 – Порівняння методів навчання Continuous bag-of-words та Skip-gram. [51]

Модель Skip-gram добре працює з обмеженими обсягами даних та особливо ефективна для представлення рідкісних слів. Модель Continuous bag-of-words

забезпечує кращі представлення слів, які частіше зустрічаються. Чим більше епох навчання, тим векторні представлення слів стають все більш семантично збагаченими.

Результат моделі Word2Vec, яка була навчена на 1220 документах, що складаються з резюме та деталей вакансії, згідно авторів дослідження, був задовільним [54].

2. GloVe є алгоритмом для перетворення слів у щільні представлення, який був розроблений Стенфордським університетом в 2014 році як відкрита (open source) розробка. GloVe є алгоритмом навчання без вчителя. GloVe має попередньо визначені щільні представлення приблизно для 6 мільярдів слів англійською мовою разом із іншими символами загального користування, такими як коми, дужки та крапки з комою. Розробники алгоритму часто роблять доступними попередньо навчені представлення GloVe, тому немає необхідності навчати модель з нуля. Користувач може обрати попередньо підготовлені представлення GloVe у необхідному вимірі (наприклад, вектори 50-d, 100-d, 200-d або 300-d), який найкраще відповідає потребам з точки зору обчислювальних ресурсів і специфіки завдання [55].

Основна ідея GloVe полягає в тому, щоб використовувати статистику співвходження слів для навчання векторних представлень. Це означає, що GloVe спирається на глобальні статистичні властивості – частоту співвходження слів у великому корпусі тексту. Поточкова взаємна інформація (PMI) є однією із найпопулярніших метрик співвходження слів [56], див. формулу (2.4).

Формула для обчислення поточної взаємної інформації (PMI):

$$PMI(w, c) = \ln \frac{p(w, c)}{p(w) * p(c)}, \quad (2.4)$$

де w – слово;

c – категорія;

$p(w, c)$ – ймовірність появи слова w в категорії c ;

$p(w)$ – ймовірність появи слова;

$p(c)$ – ймовірність появи категорії.

Будується матриця співвходження слів, яка містить інформацію про те, як часто кожне “слово” (збережене в рядках) спостерігається в певному “контексті” (стовпці), див. рис. 2.9.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k ice)/P(k steam)$	8.9	8.5×10^{-2}	1.36	0.96

Рис. 2.9 – Приклад матриці співвходження слів GloVe [57]

Потім проводиться процес факторизації матриці співвідношення за допомогою Singular Value Decomposition для отримання щільного представлення Word Embeddings. Використання глобальної та локальної інформації зробили GloVe більш придатним алгоритмом для моделювання частих слів.

В поєднанні з такими методами SMART-аналізу, як k-nearest neighbours (KNN), опорними векторними машинами, логістичною регресією, точність моделі GloVe може сягати до 87.5% у задачі оцінки навичок кандидатів та їх відповідності до вакансії [58].

3. fastText – це бібліотека для ефективного виводу щільного представлення слів та класифікації тексту, розроблена компанією Meta (Facebook у минулому) підрозділом Meta AI Research в 2016 році; відкрита (open source) розробка [59]. На відміну від алгоритму Word2Vec, у бібліотеці fastText розглядається не лише цілі слова, але й міститься інформацію про підслова – частини слів, наприклад n-грами. Цей підхід дозволяє працювати з морфологічно багатими мовами та ефективніше фіксувати інформацію про саму структуру слова.

Бібліотека fastText представляє кожне слово як мішок n-грам символів на додаток до самого слова. Це означає, що слово “яблуко» представлено самим словом та його складовими n-грамами, тобто “яб”, “бл”, “лу”, “ук”, “ко” тощо. Цей підхід допомагає охопити значення коротших слів та дає краще розуміння суфіксів і

префіксів. Подібно до алгоритму Word2Vec, у бібліотеці fastText містяться моделі з архітектурою Continuous bag-of-words або Skip-gram. Тому нейронна мережа навчена передбачати слова або контекст не лише на основі цільових слів, але й на основі n-грам [60]. Також значною перевагою бібліотеки fastText є здатність генерувати щільні представлення слів для рідкісних слів або навіть слів, які були відсутні під час навчання [50].

Бібліотека fastText в залежності від домену класифікації тексту може досягати значень міри F1 90-91% [61].

Варто сказати, що статичні методи отримання щільних представлень обмежені у відображенні полісемії, або багатозначності, слів. Наприклад, у реченні англійською мовою: “I left my phone on the left side of the table”, статичні методи, наприклад Word2Vec, GloVe, fastText, відображають слово “left” лише у одному контексті – “залишив”. Проте, “left” також може мати інше значення, а саме “лівий”. Це означає, що для вірного розуміння контексту потрібні два різних багатовимірних вектори, один для кожного значення слова, замість одного.

Щоб вирішити цю проблему, існують контекстуальні методи Word Embeddings. Результати досліджень статичних методів вказують на те, що статичні методи не можуть точно відобразити значення багатозначних слів, але контекстуальні методи Word Embeddings, які вже включають багатозначну інформацію, працюють набагато краще [62].

1. Bidirectional Encoder Representation for transformers (BERT) є контекстуальною моделлю для отримання щільних представлень Word Embeddings, що була представлена корпорацією Google у 2018 році. Модель BERT здатна генерувати щільні представлення слів, які ураховують контекст, у якому ці слова з’являються. Основна ідея полягає в тому, щоб навчити модель передбачати наступне слово в реченні, використовуючи всі попередні слова як послідовність. BERT використовує бідирекціональний підхід, тобто вона розглядає всі слова в реченні як зліва-направо, так і навпаки, для генерації щільного представлення кожного слова –

це дозволяє моделі краще враховувати взаємозв'язки між словами. Модель BERT є натренованою на даних з Вікіпедії (близько 2,5 млрд слів) і Google BooksCorpus (близько 800 млн слів) [63].

Також особливістю моделі BERT є використання архітектури Transformer (а саме її частини Encoder) для ефективного врахування довготривалих залежностей в тексті. Архітектура Transformer дозволяє ефективно розпаралелювати навчання моделі, тому є можливим навчання BERT на великих обсягах даних за короткий період часу [64]. Моделі на основі архітектури Transformer вимагають менше часу на навчання, ніж попередні архітектури рекурентної нейронної мережі, такі як довга короткочасна пам'ять (LSTM).

У моделі на основі архітектури Transformer замість перегляду кожного слова в тому порядку, в якому воно з'являється в реченні, кожному слову присвоюється унікальний номер. Це надає інформацію про позицію кожного токена у послідовності, дозволяючи моделі враховувати послідовну інформацію послідовності.

Attention – механізм, який обчислює ваги для кожного слова в реченні. Оскільки ваги стосуються кожного слова в реченні, тому модель може передбачити слова, які, ймовірно, будуть використані в одній послідовності. Це розуміння засвоюється з часом, коли модель навчається на великій кількості даних. Механізм Self-attention дозволяє кожному слову паралельно звертати увагу на кожне інше слово в послідовності, зважуючи їх важливість для поточного слова. Таким чином, можна сказати, що моделі з механізмом Attention можуть “вивчати” правила граматики на основі статистичних ймовірностей того, як слова зазвичай використовуються в мові [65]. Архітектура Transformer складається з 2 частин: Encoder та Decoder, див. рис. 2.10.

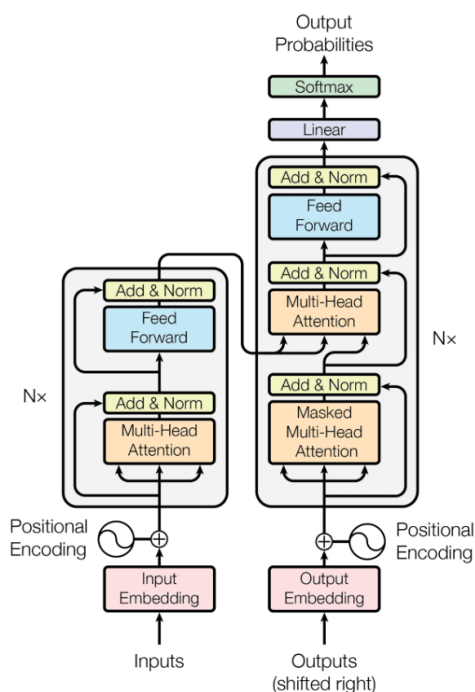


Рис. 2.10 – Побудова архітектури Transformer, що складається з Encoder (ліва частина) та Decoder (права частина) [66]

Основною функцією Encoder є перетворення вхідних слів у щільні представлення.

Спочатку, всі слова у тексті перетворюються у дискретні представлення за допомогою One-hot encoding, який перетворює кожне слово вхідного тексту у вектор, де кожна позиція вектору відповідає словнику, а на позиції, що відповідає конкретному слову, стоїть 1, а всі інші значення – 0. Потім до дискретних представлень застосовується Positional encoding, що додається до первісних дискретних представлень, за допомогою тригонометричних функцій.

Далі вхідна послідовність подається через Encoder layers. Кожен Encoder layer містить два підшари: Self-attention та нейронну мережу. Self-attention дозволяє трансформеру “дивитися” на інші слова у вхідному реченні під час обробки кожного слова, допомагаючи моделі узгоджувати контекст. У подальшому результати Multi-head attention вектори контексту передаються через нейронну мережу, яка працює з кожною позицією незалежно. Результатом Encoder архітектури Transformer є позиційно та контекстуально збагачені щільні представлення [67].

Модель BERT має два різні варіанти: BERTbase та BERTlarge. Основні відмінності між ними полягають у розмірі моделі та кількості параметрів. BERTbase має меншу кількість параметрів порівняно з BERTlarge, але все ще здатна досягати досить високої продуктивності на багатьох завданнях обробки природної мови. BERTbase містить близько 110 мільйонів параметрів. BERTlarge зазвичай досягає кращих результатів на багатьох завданнях, але вимагає більших обчислювальних ресурсів для навчання та використання. BERTlarge містить близько 340 мільйонів параметрів [64].

Модель BERTlarge у задачі класифікації навичок та речень резюме досягає значень міри F1 приблизно 86.67% [68].

2. ELMo є контекстуальною моделлю для отримання щільних представлень Word Embeddings, що була представлена інститутом Allen Institute for Artificial Intelligence та університетом University of Washington у 2018 році. Модель ELMo здатна генерувати щільні представлення слів, які ураховують контекст, у якому ці слова з'являються. На відміну від моделі BERT, що використовує архітектуру Transformer, архітектура ELMo містить двохшарову рекурентну нейронну мережу LSTM (biLM або bi-directional LSTM), див. рис. 2.11. Така архітектура дозволяє моделі ELMo враховувати не лише слова зліва, а й зправа від поточного слова.

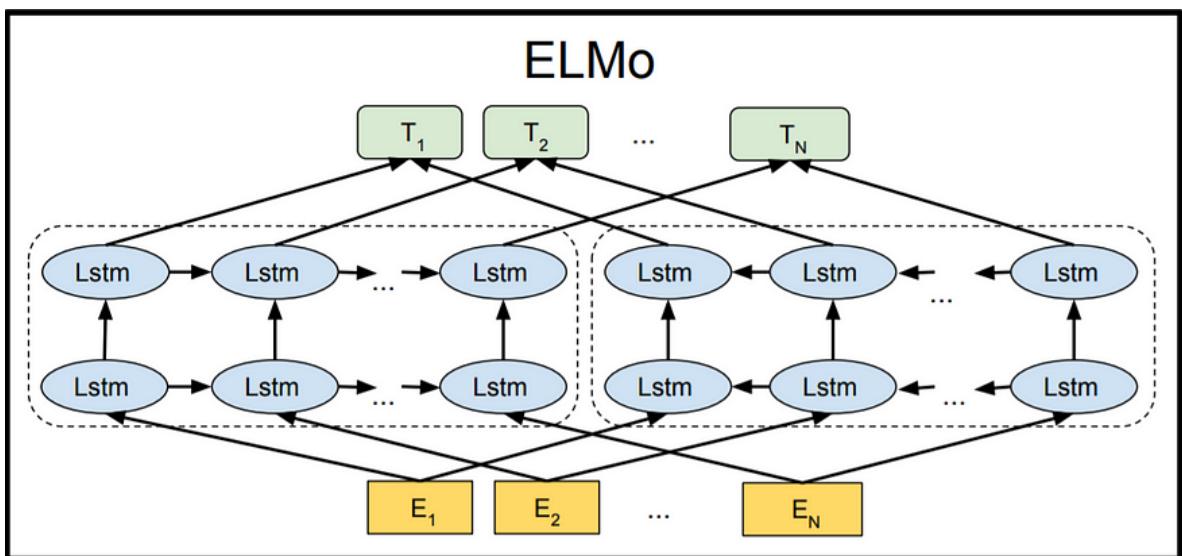


Рис. 2.11. – Побудова архітектури biLM моделі ELMo [69]

Архітектура biLM використовує згорткову нейронну мережу (CNN) на рівні символів для представлення слів тексту у вектори необроблених слів – ці вектори необроблених слів діють як вхідні дані для першого рівня biLM.

Пара інформації з прямого та зворотного проходів утворює проміжні вектори слів, що дозволяє враховувати контекст минулих та наступних слів. Далі проміжні вектори слів подаються на наступний рівень архітектури biLM. Остаточне щільне представлення – це зважена сума векторів необроблених слів і 2 проміжних векторів слів.

Оригінальна модель ELMo є навченою на бенчмарку 1 Billion Word Benchmark та приблизно 800 млн токенів даних новин від WMT 2011. Велика модель ELMo 5,5B є навченою на наборі даних із 5,5 мільярдів токенів, що складається з Wikipedia (1,9 мільярдів токенів) та одномовних даних сканування новин від WMT 2008-2012 (3,6 мільярдів токенів) [70].

Результати моделі ELMo для класифікації короткого тексту (в тому числі, резюме) в поєднанні з такими методами SMART-аналізу, як опорними векторними машинами, k-nearest neighbours (KNN), деревами рішень можуть сильно варіюватися у точності від 66.1% до 96.2% в залежності від вхідних даних [71].

Третім етапом автоматизованого відбору є порівняння щільних представлень корпусів резюме та опису вакансії на подібність. Цей етап дозволяє автоматизованим системам за допомогою методів SMART-аналізу оцінювати, наскільки навички, досвід та кваліфікації кандидата відповідають вимогам.

1. Досить популярною мірою подібності є косинусова подібність для обчислення подібності між будь-якими текстовими документами, незалежно від їх розмірів. Обчислення косинусної подібності між текстовими документами вимагає щільних представлень, а потім нормалізації цих векторів у L2 норму (або Евклідову відстань). Після отримання нормалізованих векторів двох текстових документів, косинусова подібність буде їх скалярним добутком, див. формулу (2.5). Величина міри подібності варіюється від -1 до +1.

Формула для обчислення косинусової подібності (cos):

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \|y\|} \quad (2.5)$$

2. Іншою мірою подібності є sqrt-cos подібність, див. формулу (2.7) [72]. Стверджується, що в якості вимірювання подібності sqrt-cos подібність забезпечується значеннями від 0 до +1, яке краще можна оцінити за допомогою ймовірнісних підходів. Ця міра заснована на відстані Хелінгера, див. формулу (2.6).

Формула для обчислення відстані Хелінгера (HellDist):

$$\text{HellDist}(x, y) = \sqrt{2 - 2 \sum_i \sqrt{x_i y_i}} \quad (2.6)$$

Формула для обчислення sqrt-cos подібності (SqrtCos):

$$\text{SqrtCos}(x, y) = \frac{\sum_{i=1}^m \sqrt{x_i y_i}}{(\sum_{i=1}^m x_i)(\sum_{i=1}^m y_i)} \quad (2.7)$$

3. Не менш популярною мірою подібності є поліпшена sqrt-cos подібність або ISC (improved sqrt-cos) подібність [72]. У цій мірі використовується корінь з L1 норми (або Манхеттенська відстань), див. формулу (2.8). Величина міри варіюється від 0 до +1.

Формула для обчислення ISC подібності (ISC):

$$\text{ISC}(x, y) = \frac{\sum_{i=1}^m \sqrt{x_i y_i}}{\sqrt{(\sum_{i=1}^m x_i)(\sum_{i=1}^m y_i)}} \quad (2.8)$$

Варто зазначити, що інформація, отримана з даних великої розмірності, таких як резюме, що містять багато слів, може стати проблематичною при використанні Евклідової відстані, яка в даному випадку рідко вважається ефективною мірою відстані. Також доведено, що L2 норма часто є неефективною для порівняння даних великої розмірності. Було досліджено поведінку Lk норм у просторі великої розмірності та дійшли до висновку, що рекомендується використовувати нижче значення k при роботі з даними великої розмірності. Тобто, L1 норма є більш зручною, ніж L2 норма у великих даних, таких як документи резюме, які можуть містити забагато слів [72]. Процес експерименту в дослідженні передбачав збір реальних наборів даних від двох компаній, однієї в Європі та іншої в США, на посади

менеджера з розвитку бізнесу та інженера-програміста. Потім набори даних були підготовлені для аналізу шляхом виконання тих самих кроків перетворення та очищення даних, стембування та векторизації. Косинусова, sqrt-cos та ISC подібність порівнювалися з людською продуктивністю як еталоном за двома показниками: час виконання та точність. Було визначено, що до складу 10 найкращих кандидатів, проранжованих за допомогою професіоналу з управління людськими ресурсами, увійшли 11 найкращих кандидатів за ISC подібністю, 17 найкращих кандидатів за sqrt-cos подібністю та 18 найкращих кандидатів за косинусовою подібністю. Подібні результати зафіксовано на обидві посади [72].

4. Також, для визначення подібності існує коефіцієнт Серенсена-Дайса (Dice Coefficient). Він обчислюється шляхом порівняння кількості спільних об'єктів з їх загальною кількістю в двох векторах, див. формулу (2.9) [73]. Чим більше спільних об'єктів, тим вище значення коефіцієнта, що вказує на більшу подібність між наборами. Величина коефіцієнту варіюється від 0 до +1.

Формула для обчислення коефіцієнту Серенсена-Дайса (Dice):

$$Dice(x, y) = \frac{2 * |x \cap y|}{|x| + |y|} \quad (2.9)$$

5. Коефіцієнт Жаккара (Jaccard Coefficient) є коефіцієнтом для визначення подібності, див. формулу (2.10). У коефіцієнті Жаккара вимірюється кількість спільних об'єктів до загальної кількості усіх об'єктів [73]. Коефіцієнт Жаккара зосереджується на наявності або відсутності ознак, тому забезпечує міру подібності між 0 і +1, що особливо корисно для порівняння векторів з різною кількістю просторів.

Формула для обчислення коефіцієнту Жаккара (Jaccard):

$$Jaccard(x, y) = \frac{|x \cap y|}{|x \cup y|} \quad (2.10)$$

Для класифікації резюме разом із такими методами SMART-аналізу, як k-nearest neighbours (KNN), логістичною регресією, точність алгоритму з комбінацією коефіцієнту Жаккарда та косинусовою подібністю може сягати 85.27% [74]. Навпаки, для підбору вакансій для конкретного кандидата, точність коефіцієнту Жаккарда

може сягати 94% [75]. Через подібність коефіцієнтів Жаккарда та Серенсена-Дайса, ймовірна висока кореляція між результатами [76].

Інші концептуальні підходи для автоматизованого відбору включають: рекомендаційні системи, що рекомендують роботу потенційним кандидатам, а відповідні кандидати – професіоналам з управління людськими ресурсами [77]; робота з великими мовними моделями тощо [68].

Варто зазначити, що постановка та методи задач оцінювання навичок кандидатів, вхідні дані можуть суттєво відрізнятися, тому необхідний комплексний та індивідуальний підхід, що дозволяє обирати найбільш ефективне та оптимальне рішення. Завдяки частковій автоматизації процесу попереднього відбору професіонали з управління ресурсами можуть суттєво заощаджувати час і ресурси та гарантувати, що відповідних кандидатів не будуть випадково пропущено серед великої кількості заявок на вакантну посаду.

2.2. Вплив алгоритмів на об'єктивність та нейтральність процесу оцінювання навичок кандидатів

Згідно конвенції C111 Міжнародної організації праці, дискримінація на ринку праці – будь-яке розрізнення, виключення або переваги, зроблені на основі раси, кольору шкіри, статі, релігії, політичних поглядів, національного походження або соціального походження, що має наслідком зведення нанівець або погіршення рівності можливостей або ставлення у професійній сфері [78].

Теорія статистичної дискримінації стосується упередженого ставлення до критеріїв оцінки, які узагальнюють групові характеристики на окремих осіб. Ця теорія передбачає, що прийняття рішень може бути ґрунтоване на стереотипах або упередженостях, які базуються на групових характеристиках, таких як раса, стать, вік, національність тощо [79]. У теорії статистичної дискримінації вважається, що особа або група, яка приймає рішення, може застосовувати статистичні методи, такі як середні, медіани або інші показники, для оцінки індивідуальних осіб на основі групових характеристик. Наприклад, у розрізі ринку праці роботодавець може використовувати середній рівень доходу чи освіти для прийняття рішення про найм працівника. Однак ця практика може призводити до негативних наслідків, таких як дискримінація проти індивідів або груп на основі їхніх особистих характеристик, а не на основі їхніх індивідуальних здібностей або заслуг.

Через широке застосування технологій SMART-аналізу для управління людськими ресурсами, зокрема для процесу оцінювання навичок кандидатів, виникне проблема алгоритмічної дискримінації, зміщуючи центр статистичної теорії дискримінації з традиційних підходів найму на SMART-підходи, див. рис. 2.12.

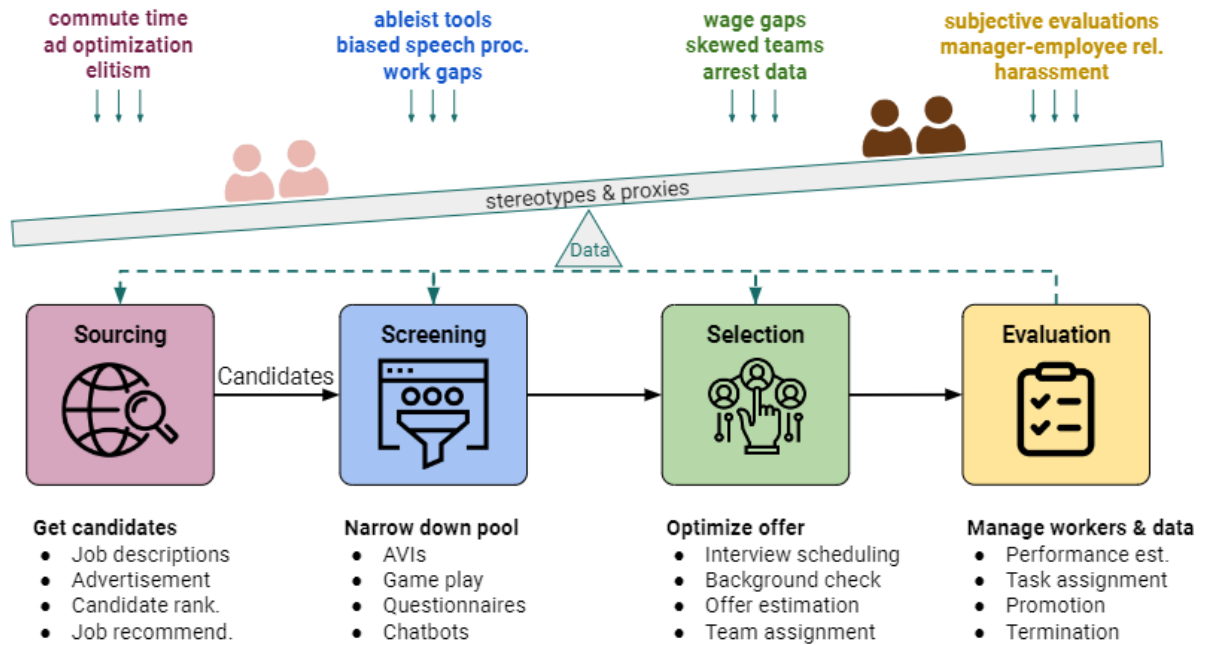


Рис. 2.12 – Виникнення упереджень на етапах управління людськими ресурсами [80]

Механізми, що породжують проблеми дискримінації при оцінюванні навичок, залишаються подібними, оскільки покладаються на історичні дані конкретних репрезентованих груп кандидатів для прогнозування майбутніх результатів найму.

Хоча оцінювання навичок кандидатів за допомогою алгоритмів з підходами SMART-аналізу й пропонує численні переваги, воно також чутливе до алгоритмічних упереджень. Алгоритмічна дискримінація виникає, коли автоматизовані системи сприяють невинуватному відмінному поводженню або негативному впливу на людей на основі їх раси, кольору шкіри, етнічної приналежності, статі (включаючи вагітність, пологи та пов'язані захворювання, гендерну ідентичність, сексуальну орієнтацію), релігії, віку, національного походження, інвалідності чи будь-якої іншої класифікації, захищеної законом [81].

Наприклад, у 2015 році у корпорації Amazon виник великий скандал через роботу алгоритмів оцінювання навичок кандидатів на вакантні посади та їх алгоритмічних упереджень. Програмне забезпечення з використанням SMART-аналізу було створено командою в Единбурзі Amazon у 2014 році як спосіб

автоматичного сортування резюме та відбору найталановитіших кандидатів. Але, за словами членів команди, алгоритми швидко навчилися віддавати перевагу кандидатам-чоловікам, а не жінкам. Розробники зрозуміли, що алгоритми оцінюють резюме кандидатів негативно, які містять слово “жінка” (та похідні слова), наприклад, “капітан жіночого шахового клубу”. Відомо, що система також знизила оцінку випускників двох виключно жіночих коледжів.

Стверджувалося, що причина алгоритмічних упереджень полягала в тому, що програмне забезпечення було навчено на даних, наданих протягом 10-річного періоду, значна частина яких надходила від кандидатів-чоловіків. У зв’язку з тим, що значна частина даних була зібрана від кандидатів-чоловіків, алгоритми навчилися виявляти та оцінювати певні характеристики, що були більш типовими для більш репрезентативної групи. Повідомлялося, що спеціалісти з управління людськими ресурсами корпорації Amazon розглядали рекомендації, створені алгоритмами оцінювання навичок кандидатів, але ніколи не покладалися виключно на їх судження та роботу [82].

Після виявлення упередженості та негативних наслідків використання алгоритмів оцінювання навичок, корпорація Amazon прийняла рішення закрити проект на початку 2017 року. Цей крок був зумовлений підвищенням обговоренням та критикою з боку громадськості, а також усвідомленням ризиків, пов’язаних з продовженням використання алгоритмів, які спричиняли упередженість та непрозорість у процесі прийняття рішень. Закриття проекту підкреслило необхідність критичного оцінювання та управління алгоритмами, які використовуються в корпоративному середовищі, зокрема, забезпечення їхньої справедливості, прозорості та відповідності принципам рівних можливостей.

Частіше за все, методи SMART-аналізу для оцінювання навичок кандидатів мають проблему алгоритмічної дискримінації через їх залежність від даних, які містять когнітивні упередження та досвід про минулі рішення. Навіть найменші упередження в даних можуть призвести до суттєвих наслідків та спричинити

дискримінацію та суттєву втрату продуктивності [83].

Black Box моделі SMART-аналізу для оцінювання навичок кандидатів, які важко інтерпретувати або пояснити, сприяють відсутності прозорості в процесі оцінювання. Без чіткого розуміння того, як приймаються рішення, стає складно виявляти й усувати потенційні упередження або дискримінаційні шаблони [84]. Також методи SMART-аналізу можуть бути схильні до надмірного узагальнення, та посилити алгоритмічну дискримінацію потенційних кандидатів на вакантні позиції, роблячи припущення або узагальнення щодо окремих осіб на основі обмежених або неповних даних. Це може призвести до неточних оцінок і зміцнити стереотипи, ще більше закріпивши нерівність у доступі до роботи.

Саме тому впровадження об'єктивних та нейтральних практик для запобігання дискримінації у процесі оцінювання навичок кандидатів є важливою частиною розробки самих алгоритмів.

По-перше, необхідний технічний підхід для впровадження об'єктивних та нейтральних практик для запобігання дискримінації. Наприклад, пропонується систематичний підхід до зменшення алгоритмічної дискримінації, а саме вичерпну інструкцію, за якою розробники програмного забезпечення могли б слідкувати та дотримуватись при розробці відповідальних систем за допомогою методів SMART-аналізу. Хоча автори підходу спеціально не зосереджуються на алгоритмах під час найму, кроки є універсальними, включно з розробкою алгоритмів для оцінювання навичок кандидатів на вакантні посади. Спочатку визначаються стереотипні знання шляхом ідентифікації захищених властивостей та пов'язаних стереотипних аспектів. Цей початковий крок закладає основу для розуміння потенційних упереджень, притаманних даним та алгоритмам. Далі розробників програмного забезпечення закликають оцінити поведінку моделі за допомогою стереотипних і захищених виразів, по суті перевіряючи, як алгоритм реагує на різні демографічні характеристики. Ця оцінка допомагає точно визначити області, де може бути упередженість та алгоритмічна дискримінація. Потім необхідно проаналізувати

результати оцінювання, щоб визначити категорії, що призводять до більшого упередження. Цей аналіз слугує діагностичним інструментом для розуміння природи та ступеня дискримінації в алгоритмах. Після цього розробники програмного забезпечення мають заново переоцінювати модель ітеративно, доки не досягнуть прийняттого рівня справедливості. Нарешті, про результати оцінки слід повідомляти прозоро, щоб забезпечити підзвітність і полегшити постійний моніторинг і вдосконалення [85].

З іншого боку, зазначається інший підхід, а саме класифікація методів пом'якшення упередженості, що забезпечує комплексну основу для усунення алгоритмічної дискримінації в алгоритмах оцінювання навичок кандидатів. Методи попередньої обробки зосереджені на коригуванні початкового набору даних для відновлення балансу та забезпечення справедливості таких характеристик, як раси, кольору шкіри, етнічної приналежності, статі, релігії, віку тощо. Змінюючи набір даних, методи попередньої обробки спрямовані на пом'якшення упередженості в джерелі моделі, тим самим закладаючи основу для справедливого навчання моделі. Однак попередня обробка може бути неефективною, якщо сам набір даних за своєю суттю упереджений, та може потребувати ресурсів, особливо для великих наборів даних. Методи обробки, з іншого боку, зосереджені на покращенні справедливості моделі під час навчання. Наприклад, *Adversarial Debiasing* є алгоритмом для усунення алгоритмічної дискримінації. Він передбачає одночасне навчання предиктора і дискримінатора. Мета предиктора – точно передбачити цільову змінну. У той же час дискримінатор прагне передбачити захищену змінну (наприклад, стать або расу) на основі прогнозів предиктора. Основна мета полягає в тому, щоб максимізувати здатність предиктора передбачати цільову змінну, одночасно зменшуючи здатність дискримінатора передбачати захищену змінну на основі цих прогнозів. Нарешті, методи пост-обробки маніпулюють прогнозами моделі, використовуючи обмеження на основі незміщених даних, але можуть призвести до втрати продуктивності. Незважаючи на цей недолік, методи постобробки забезпечують ненав'язливий спосіб

усунути зміщення без доступу до параметрів моделі [85].

По-друге, необхідне створення стандартизованого та структурованого фреймворку для впровадження відповідальних систем за допомогою методів SMART-аналізу.

У загальному можна виділити такі загальні принципи регулювання [86]:

1. Встановлення чітких етичних принципів, узгоджених з правами людини. Створення відповідальних систем за допомогою методів SMART-аналізу лише з наміром не завдати шкоди;
2. Надання прозорих та задокументованих принципів функціонування та загальних процесів прийняття рішень. Результати систем за допомогою методів SMART-аналізу необхідно пояснити у зрозумілій людині спосіб;
3. Впровадження заходів для виявлення та пом'якшення упереджень в алгоритмах SMART-аналізу, тобто прагнення до справедливості та рівного ставлення, як-от уникнення будь-якої форми виключення на основі інвалідності, соціально-економічного статусу, статі тощо;
4. Забезпечення високого пріоритету конфіденційності та безпеки даних протягом усього життєвого циклу створення відповідальних систем за допомогою методів SMART-аналізу. Дотримання правил конфіденційності даних та отримання згоди користувача на збір даних;
5. Включення підходу “human-in-the-loop” для моніторингу продуктивності відповідальних систем за допомогою методів SMART-аналізу. Встановлення циклу зворотного зв'язку із залученням людського досвіду для неперервного вдосконалення алгоритмів;
6. Притягнення окремих осіб та організації до відповідальності за дії та результати систем за допомогою методів SMART-аналізу. Проводення регулярних незалежних перевірок моделей, алгоритмів і процесів на об'єктивність та нейтральність.

Наразі розвинені країни та країни, що розвиваються, у всьому світі

перебувають на різних етапах розвитку підходу до регулювання методів SMART-аналізу та дотримуються протилежних поглядів на те, як це найкраще зробити.

Наприклад, Європейський Союз перебуває в процесі прийняття Закону Європейського Союзу про штучний інтелект, що попередньо був схвалений Європейським Парламентом під час переговорів із державами-членами в грудні 2023 року. Закону Європейського Союзу про штучний інтелект спрямований на захист фундаментальних прав, демократії, верховенства права та стійкості навколишнього середовища від штучного інтелекту з високим ризиком, одночасно стимулюючи інновації та утверджуючи Європу як лідера в цій галузі.

Чіткі зобов'язання передбачені для систем з високим ризиком через їх значну потенційну шкоду для здоров'я, безпеки, основних прав, навколишнього середовища, демократії та верховенства права. Приклад використання методів SMART-аналізу, зокрема штучного інтелекту, з високим ризиком включає сферу працевлаштування. Зазначається, що такі системи повинні оцінювати та зменшувати ризики, бути прозорими та точними, забезпечувати підхід “human-in-the-loop” [87].

4 жовтня 2022 року Управління науково-технічної політики Білого дому Сполучених Штатів Америки оприлюднило план під назвою “Blueprint for the Development, Use, and Deployment of Automated Systems”, який часто називають “AI Bill of Rights”. Цей план є не обов'язковим набором правил, на відміну від Закону Європейського Союзу. Натомість у ньому викладено п'ять керівних принципів, спрямованих на зменшення можливої шкоди, заподіяної системами за допомогою методів SMART-аналізу, включаючи алгоритмічну дискримінацію, конфіденційність даних, безпеку використання тощо [88].

РОЗДІЛ 3

СТВОРЕННЯ АЛГОРИТМІВ ДЛЯ ЕФЕКТИВНОГО ОЦІНЮВАННЯ НАВИЧОК КАНДИДАТІВ НА ПОСАДИ ІТ-ФАХІВЦІВ

3.1. Розробка алгоритмів для оцінювання навичок кандидатів на посади ІТ-фахівців

В практичній частині кваліфікаційної бакалаврської роботи була зроблена вибірка розміром 12 резюме випадковим чином із загального переліку, що становить близько 100 резюме. Алгоритми для оцінювання навичок кандидатів імплементовані на основі скриптової мови програмування Python та бібліотек для роботи з текстовими даними NLTK, машинного навчання як елементу SMART-аналізу Tensorflow, Pytorch, Gensim тощо. Типова структура резюме включає в себе особисту інформацію, досвід роботи кандидата, освіту, навички, досягнення тощо; приклад резюме з вибірки зображено на рис. 3.1.

TECHNICAL SKILLS

- **Language:** Java
- **Frameworks:**
 - Spring Framework
 - Spring Boot
 - Spring Security
 - Spring Web
 - Spring Data
 - Hibernate
- **Databases:** MySQL, PostgreSQL
- **Developer Tools:** Git, Apache Maven, Gradle
- **API:** Collection API, Stream API
- **Testing:** Junit, Mockito

EDUCATION

BACHELOR OF SOFTWARE ENGINEER
KHARIV NATIONAL UNIVERSITY OF RADIOELECTRONICS
2021 - NOW

LANGUAGES

- English - Intermediate
- Ukrainian - Native

CERTIFICATE

- Cross-platform programming (Java) (2022)
- Technologies of Computer Aided Design (2023)
- Java Basic Courses at GeeksForLess Inc. (2022)
- Java Advanced Courses at GeeksForLess Inc. (2023) (Traineeship)

REWARD

Java Basic course for GeeksForLess
Studied from October 2022 - 2022 December
One of the best students

EXPERIENCE

JAVA SOFTWARE ENGINEER
FFREELANCE | MAY 2023 - AUG 2023

- Made integration with third-party services through the REST API
- Solve performance issues
- Analysis of different proxy service sources (file, http source)
- Authorization using spring security token

TRAINEE JAVA DEVELOPER
GEEKSFORLESS INC | JAN 2023 - MAY 2023

- Developed Rest API for putting a new script in the execution and transmission queue new scripts to execute.
- ProxyProcessingService services for proxy parsing and validation File / URL lists
- EventPublisherService - publisher of new scripts for execution.
- All services are covered by tests.

JAVA SOFTWARE ENGINEER
FFREELANCE | SEP 2023 - DEC 2023

- Interpretation of commands and scripts in Selenium webdriver operation
- Starting the Selenium web driver and executing the script
- Solve performance issues
- Database merging and optimization using query plan

PROFILE INFO

I have been learning Java for almost two years. In addition, I am currently a third-year student at NURE, where I study majoring in computer science. I am focused on becoming a web developer and in my free time I devote myself to improve their skills and work on personal projects.

Рис. 3.1 – Приклад структури резюме з вибірки (власна розробка)

Оскільки повний опис вакансій резюме на базі практики компанії Whalebone s.r.o. є конфіденційним, розглянемо модельні описи вакансій на прикладі позицій Business Analyst та Senior Software Engineer Manager в корпораціях Amazon та Google відповідно через їх вільний доступ.

Організація Middle Mile Transportation Technology корпорації Amazon розробляє складні програмні рішення, що працюють через постачальників, склади та перевізників для оптимізації як часу, так і витрат на доставку товарів – через зростаючу конкуренцію та стрімку зміну технологій корпорація Amazon потребує фахівця на позицію Business Analyst. Основні відповідальності включають в себе дослідження даних для зниження витрат на виконання замовлень, покращення ефективності та досвіду клієнтів в підтримці прийняття бізнес-рішень, розробка ключових показників ефективності, порівняння короткострокового та довгострокового планування мережі та прогнозування. Навички та кваліфікації, якими повинен володіти ідеальний кандидат, включають в себе релевантний бакалаврський рівень освіти, досвід з табличним процесором Excel, засобами візуалізації даних тощо, див. рис. 3.2.

BASIC QUALIFICATIONS

- Bachelor's degree or equivalent
- Experience in Excel (including VBA, pivot tables, array functions, power pivots, etc.) and data visualization tools such as Tableau
- Experience in business or financial analysis
- Experience defining requirements and using data and metrics to draw business insights
- Experience with Excel
- Experience with SQL

PREFERRED QUALIFICATIONS

- Experience with other languages like Tableau, python or R
- Experience making business recommendations and influencing stakeholders

Рис. 3.2 – Опис необхідних кваліфікацій на позицію Business Analyst [89]

Також, корпорація Google шукає фахівця на посаду Senior Software Engineer Manager на проект, що пов'язаний з Embedded Systems/Firmware, для підтримки загальної технічної інфраструктури корпорації. Основні відповідальності включають

в себе узгодження стратегічних цілей, процесів та прийняття рішень в командах, розробка довгострокового технічного бачення та дорожньої карти для потреби технічної інфраструктури, огляд розвитку продукту або системного коду. Навички та кваліфікації, якими повинен володіти ідеальний кандидат, включають в себе релевантний бакалаврський рівень освіти або еквівалентний рівень досвіду, мінімум 5 років досвіду з мовами програмування (Python, C, C++, Java, JavaScript), технічного лідера та роботи з вбудованими операційними системами, див. рис. 3.3.

Minimum qualifications:

- Bachelor's degree or equivalent practical experience.
- 5 years of experience with software development in one or more programming languages (e.g., Python, C, C++, Java, JavaScript).
- 5 years of experience in a technical leadership role; overseeing projects, with 5 years of experience in a people management, supervision/team leadership role.
- 5 years of experience with embedded operating systems.

Preferred qualifications:

- Master's degree or PhD in Engineering, Computer Science, or a related technical field.
- 5 years of experience working in a complex, matrixed organization.

Рис. 3.3 – Опис необхідних кваліфікацій на позицію Senior Software Engineer Manager [90]

Узагальнена робота вибору найкращого кандидата на дані позиції описана в розділі 2 кваліфікаційної бакалаврської роботи та зображена на рис. 3.4.

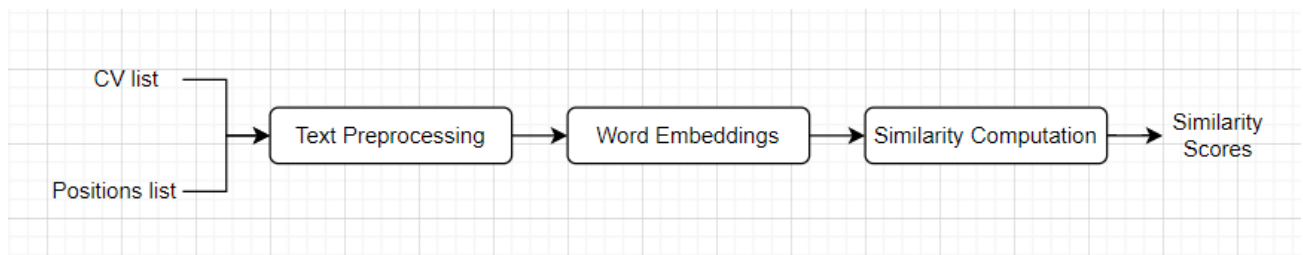


Рис. 3.4 – Етапи автоматизованого відбору оцінювання навичок кандидатів (власна розробка)

3.2. Адаптація алгоритмів до специфічних вимог та особливостей ІТ-індустрії при оцінюванні навичок кандидатів на посади ІТ-фахівців

Варто зазначити, що усі резюме з вибірки є англійською мовою, що свідчить про загальноприйняту практику в галузі ІТ представляти інформацію англійською. Більшість вакансій в ІТ-індустрії вимагають вільного володіння англійською мовою, оскільки це мова комунікації в багатьох технічних та міжнародних командних проектах. Крім того, згідно Загального Регламенту Захисту Даних (The General Data Protection Regulation або GDPR), дані, за допомогою яких можна однозначно ідентифікувати особу, повинні бути псевдоанонімізовані або анонімізовані (видалені) для подальшої обробки та зберігання [91]. Саме тому текстові дані, що містять інформацію для однозначної ідентифікації кандидата, всіх резюме було змінено або видалено для забезпечення об'єктивного та нейтрального процесу оцінювання навичок кандидатів. На модельному прикладі розберемо варіант підбору кандидата на позицію Business Analyst корпорації Amazon.

Для початку необхідно опрацювати текстові дані з резюме потенційного робітника та опису позиції, див. рис. 3.5.

```
SUMMARY
I'd like to suggest to you my expertise, knowledge and skills. My skills
of instant response to changes in the cases, proficiency of
negotiations, mediation, issues solving allow me be sure in my ability
to quickly dive into a new sphere of activity
EDUCA TION
Years Educational institution Qualification
1989-1994 Yaroslav Mudryi National Law University LL.M.
PROFESSIONAL SKILLS
OS Windows, Android, iOS , MacOS, Ubuntu
Technologies WEB, mobile
DBMS MS SQL, MySQL, Oracle, Access (surface immersion )
BA support tools Jira, Confluence, ClickU p, MS project, Google Docs, Enterprise architect ,
MS Office etc. , Postman
Design , diagrams AdobeXD, inVision, Figma, Draw.io, Miro,
Methodologies Agile (Scrum, Kanban), Waterfall
Foreign languages English Upper Intermediate
Other skills IDEF, UML, BPMN , REST API, GraphQL , XML, JSON
SERVICE RECORDS

Years Employer Position and brief job description
01/2021 Telesens Business analyst
06/2020 -12/2020 DiPocket Business analyst
08/2018 -05/2020 Resty application Business analyst, Product manager
```

Рис. 3.5 – Текстові дані з резюме кандидата (власна розробка)

Відбувається токенизація цілого тексту (резюме кандидата або опису вакансії) для розбивки на окремі речення, див. рис. 3.6.

```
'As a Business Analyst, you will operate at the crossroads of multiple complex Amazon systems, getting global visibility of how Amazon moves
inventory across our network and serves our customers.',
'You will work with the core optimization models that drive the Middle Mile planning business for Amazon.',
'You will enable the creation of products that drive ever-greater automation, scalability and optimization of every aspect of transportation,
removing cost and delivering speed of execution for our customers.',
'The impact of your work will be global, material, and remarkable.',
'The successful candidate will be voraciously curious about Amazon's transportation operations and how data consumed and produced by our
systems can be used to improve outcomes and lower costs.',
'Your responsibility is to expose and measure the current performance of our systems, find and quantify opportunities for improvement, and dive
deep into existing algorithms to explain unexpected performance.',
'We are looking for a sophisticated user of data querying tools and an expert at synthesizing / communicating insights & recommendations to
audiences of varying levels of technical sophistication.'
```

Рис. 3.6 – Розбивка текстових даних з резюме на окремі речення (власна розробка)

Далі потрібно провести очистку кожного речення в текстових даних (резюме кандидата або опису вакансії) – видалити пунктуацію, стоп-слова, числа тощо, див. рис. 3.7.

```
['httpswwwamazonjobsenjobsbusinessanalyst description will able determine amazons middle mile automated driver supply planning predicting right
number driver shift deliver million package every day customers',
'as business analyst operate crossroad multiple complex amazon systems getting global visibility amazon move inventory across network serf
customers',
'you work core optimization model drive middle mile planning business amazon',
'you enable creation product drive evergreater automation scalability optimization every aspect transportation removing cost delivering speed
execution customers',
'the impact work global material remarkable',
'the successful candidate voraciously curious amazons transportation operation data consumed produced system used improve outcome lower costs',
'your responsibility expose measure current performance systems find quantify opportunity improvement dive deep existing algorithm explain
unexpected performance',
'we looking sophisticated user data querying tool expert synthesizing communicating insight recommendation audience varying level technical
sophistication',
'the middle mile transportation technology organization build complex software solution work across vendors warehouses carrier optimize time
cost getting package delivered',
```

Рис. 3.7 – Очистка кожного речення опису вакансії (власна розробка)

Проводиться токенизація кожного речення в текстових даних (резюме кандидата або опису вакансії), див. рис. 3.8.

```
['my',
'skill',
'instant',
'response',
'change',
'cases',
'proficiency',
'negotiations',
'mediation',
'issue',
```

Рис. 3.8 – Токени кожного речення з резюме кандидата (власна розробка)

Далі створюється загальний корпус з очищених речень (резюме кандидата або

опису вакансії), див. рис. 3.9.

```
'httpswwwamazonjobsenjobsbusinessanalyst description will able determine amazons middle mile automated driver supply planning predicting right n
umber driver shift deliver million package every day customers as business analyst operate crossroad multiple complex amazon systems getting glo
bal visibility amazon move inventory across network serf customers you work core optimization model drive middle mile planning business amazon y
ou enable creation product drive evergreater automation scalability optimization every aspect transportation removing cost delivering speed exec
ution customers the impact work global material remarkable the successful candidate voraciously curious amazons transportation operation data co
nsumed produced system used improve outcome lower costs your responsibility expose measure current performance systems find quantify opportunity
improvement dive deep existing algorithm explain unexpected performance we looking sophisticated user data querying tool expert synthesi...'
```

Рис. 3.9 – Корпус опису вакансії (власна розробка)

На другому етапі оцінювання навичок кандидатів відбувається створення статистичних моделей, моделей машинного, глибокого навчання та отримання Word Embeddings з оброблених текстових даних резюме та опису вакансії. Word Embeddings моделі Bag-of-words з оброблених текстових даних резюме, див. рис. 3.10.

```
array([ 1,  0,  1,  0,  3,  1,  1,  1,  0,  1,  0,  0,  1,  0,  0,  0,  0,
        1, 12,  5,  0,  1,  1,  1,  1,  1,  1,  0,  0,  0,  0,  1,  0,
        1,  0,  2,  0,  1,  0,  0,  0,  1,  2,  0,  9,  3,  1,  0,  0,
        1,  0,  0,  0,  6,  1,  1,  1,  1,  1,  1,  2,  0,  0,  0,  2,  0,
        1,  1,  0,  0,  1,  0,  7,  1,  0,  0,  0,  0,  1,  2,  0,  0,  0,
        0,  1,  0,  0,  0,  1,  0,  0,  0,  0,  0,  2,  1,  1,  0,  0,  0,
        2,  1,  1,  1,  1,  0,  0,  1,  0,  0,  0,  1,  1,  0,  0,  1,  2,
        1,  1,  0,  0,  1,  0,  1,  0,  0,  0,  1,  0,  0,  0,  0,  1,  0,
        0,  1,  0,  1,  0,  0,  2,  0,  1,  0,  0,  1,  0,  0,  0,  0,  1,
        1,  0,  1,  0,  0,  0,  0,  1,  1,  0,  0,  0,  0,  1,  0,  0,  0,
        3,  1,  1,  1,  1,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0,  1,  1,
        0,  3,  1,  0,  1,  1,  1,  1,  5,  1,  1,  0,  1,  1,  0,  2,  0,
        0,  1,  0,  0,  1,  0,  1,  0,  0,  1,  1,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  1,  0,  0,  0,  4,  3,  0,  1,  0,  0,  3,  0,  1,  0,
        0,  0,  0,  1,  0,  1,  0,  0,  3,  1,  0,  0,  1,  1,  1,  2,  3,
        1,  0,  1,  1,  0,  0,  1,  1,  0,  0,  0,  0,  0,  0,  0,  1,  0,
        1,  1,  0,  0,  1,  1,  0,  0,  0,  0,  0,  2,  1,  1,  2,  0,  1,
        0,  1,  0,  1,  0,  0,  0,  0,  0,  2,  1,  0,  6,  0,  0,  2,  0,
        0,  0,  8,  1,  0,  1,  1,  0,  6,  0,  0,  0,  0,  1,  0,  0,  0,
        1,  1,  0,  1,  0,  1,  0,  0,  0,  0,  0,  1,  0, 10, 13,  0,  1,
        0,  1,  1,  1,  1,  1,  0,  1,  1,  1,  0,  1,  0,  1,  0,  6,  0,
        3,  0,  3,  0,  2,  1,  0,  0,  0,  3,  2,  0,  1,  0,  3,  1,  0,
        0,  0,  1,  7,  0,  1,  1,  0,  0,  0,  0,  0,  2,  0,  1,  1,  0,
        1,  0,  1,  1,  1,  0,  1,  0,  0,  0,  0,  1,  0,  1,  0,  5,  2,
        2,  0,  0,  1,  0,  1,  1,  0,  0,  0,  1,  0,  0,  1,  1,  2,  0,
        1,  1,  1,  4,  0,  0,  1,  0,  1,  6,  0,  1,  0,  0,  2,  3,  0,
        0,  0,  0,  2,  1,  0,  0,  0,  0,  1,  0,  1,  0,  0,  1,  0,
        0,  0,  1,  1,  0,  3,  0,  1,  1], dtype=int64)
```

Рис. 3.10 – Word Embeddings моделі Bag-of-words з оброблених текстових даних резюме (власна розробка)

Подібні Word Embeddings має модель Tf-idf.

Word Embeddings за допомогою моделей Word2Vec, GloVe, fasttext можливо отримати за умови попередньо скачаних моделей локально або в хмарному

середовищі, див. рис. 3.11. Було вибрано моделі word2vec-google-news-300.model (навчена на наборі даних Google News), glove-wiki-gigaword-300.model (навчена на наборі даних Wikipedia) та fasttext-wiki-news-subwords-300.model (навчена на наборі даних Wikipedia та університету Балтимору).

```
array([-4.51545840e+01,  3.76117587e+01,  2.49785934e+01, -6.11714920e+01,
       -1.67972928e+00,  1.08823768e+01,  9.80432490e+00,  3.25507912e+01,
       -1.21903650e+01, -5.73686909e+02,  3.54553552e+01,  1.44845262e+01,
       -7.57690645e+00, -5.77604478e+00, -3.42272895e+00, -1.37822165e+01,
       -1.64216718e+01, -6.12296406e+00, -5.10009174e+00, -2.82685349e+01,
       -2.05029389e+01,  2.67837936e+01,  6.05211656e+01,  5.13454529e+01,
       -4.90435249e+01,  4.29959286e+01,  1.36267328e+01,  4.56900998e+01,
       -1.24349722e+01,  2.82921999e+01,  1.57227921e+01,  5.45165355e+01,
       -2.63877041e+01,  1.27553344e+01, -2.04097870e+02,  6.13181091e+01,
       -9.56150485e-01,  4.04674055e+00, -8.62844493e+00, -4.24093895e+01,
       -4.00650267e+01, -9.79982828e-01, -6.70239295e+00,  2.07779492e+01,
       -1.66755874e+01,  1.81920216e+01,  9.19337657e+00,  5.11151790e+01,
       -1.33100999e+01, -8.02099291e+00,  5.65666001e+01, -3.79553619e+01,
       2.52442072e+01, -3.50140872e+01, -3.99131242e+01,  3.47105583e+01,
       1.66021409e+01,  2.63829537e+01, -3.42668430e+01,  5.08004373e+01,
       3.54273567e+00, -1.48989418e+00,  5.50712979e+01, -1.90406509e+01,
       5.67206815e-02,  3.28312133e+01, -7.24909805e-01,  4.19948089e+01,
       -1.48814532e+01,  5.54941361e+01, -3.07048318e+01,  3.21149803e+01,
       1.70147720e+01,  1.72161996e+01,  2.85830111e+00,  2.40878126e+00,
       -2.54344812e+01,  2.31874043e+01, -1.56510143e+01, -1.76168452e+01,
       -5.26091616e+01, -3.73403795e+01,  2.02266869e+01, -4.55847612e+01,
       5.61024433e+01,  1.89556448e+01, -1.07771314e+01,  1.20441729e+01,
```

Рис. 3.11 – Word Embeddings моделі GloVe з оброблених текстових даних опису вакансії (власна розробка)

Подібну структуру мають Word Embeddings в моделях Word2Vec та fasttext.

Word Embeddings за допомогою моделей BERT та ELMo також можливо отримати за умови попередньо скачаних бібліотек Transformers, tensorflow_hub локально або в хмарному середовищі, див. рис. 3.12. Було вибрано моделі bert-base-uncased та ELMo версії 3.0.

```
<tf.Tensor: shape=(100, 768), dtype=float32, numpy=
array([[ -2.5952500e-01,  2.4334517e-01,  1.7059627e-01, ...,
        -5.4074597e-01, -3.2522529e-04,  3.5067403e-01],
       [ -3.5544056e-01,  5.4763180e-01,  8.3422869e-01, ...,
        -1.7934866e-01,  8.3034456e-01, -4.6914306e-01],
       [ -3.3476093e-01, -4.8363453e-01,  1.3040342e+00, ...,
        -4.2479029e-01, -1.7363569e-01,  5.3642404e-01],
       ...,
       [ -1.1019158e+00,  3.2087073e-02, -1.4818817e-02, ...,
         1.0186414e-01, -7.1492147e-01, -3.2550913e-01],
       [ -3.5939214e-01,  7.3807913e-01,  3.6756143e-01, ...,
         6.6758350e-02, -3.6746690e-01,  3.7395549e-01],
       [  2.0293629e-01, -5.1222569e-01,  1.2524363e+00, ...,
         7.0521080e-01,  2.3861597e-01,  5.6058103e-01]], dtype=float32)>
```

Рис. 3.12 – Word Embeddings моделі BERT з оброблених текстових даних резюме кандидата (власна розробка)

Подібну тензорну структуру мають Word Embeddings в моделях BERT та ElMo.

На третьому етапі Word Embeddings всіх моделей порівнюються за допомогою метрик подібності, що були розібрані в розділі 2 кваліфікаційних бакалаврської роботи, див. рис. 3.13.

```
'Resume': 'Business analyst Male.pdf',
'Position': 'Business Analyst.txt',
'Jaccard similarity': 0.06623931623931624,
'Dice similarity': 0.12424849699398798,
'Bag-of-words cosine similarity': 0.1833751336019456,
'Bag-of-words sqrt-cos similarity': nan,
'Bag-of-words improved sqrt-cos similarity': 0.4282232287043121,
'Tf-idf cosine similarity': 0.1109284558296008,
'Tf-idf sqrt-cos similarity': nan,
'Tf-idf improved sqrt-cos similarity': 0.33305923771845874,
'Word2Vec cosine similarity': 0.8786819886618914,
'Word2Vec sqrt-cos similarity': 0.5402326442262962,
'Word2Vec improved sqrt-cos similarity': 0.9373803863223784,
'GloVe cosine similarity': 0.9281991638160322,
'GloVe sqrt-cos similarity': 0.966447319912066,
'GloVe improved sqrt-cos similarity': 0.9634309335993069,
'FastText cosine similarity': 0.9604441217091204,
'FastText sqrt-cos similarity': 0.9252787595104489,
'FastText improved sqrt-cos similarity': 0.9800225108175427,
'BERT cosine tensor similarity': 0.36940956,
'BERT sqrt-cos tensor similarity': nan,
'BERT improved sqrt-cos tensor similarity': 0.6077907,
'ElMo cosine tensor similarity': 0.4310895,
'ElMo sqrt-cos tensor similarity': nan,
'ElMo improved sqrt-cos tensor similarity': 0.656574},
```

Рис. 3.13 – Порівняння результатів метрик подібності всіх зазначених моделей (власна розробка)

Для даного кандидата на позицію Business Analyst в корпорацію Amazon найбільша подібність досягається за допомогою моделі fasttext з мірою подібності ISC. Також необхідно зазначити про можливість пропущених значень міри подібності sqrt-cos, оскільки відстань Хелінгера не може опрацьовувати негативні значення, на відміну від косинусової або ISC мір подібності.

3.3 Тестування, оцінювання та узагальнення алгоритмів для різноманітних категорій навичок та компетенцій кандидатів на посади ІТ-фахівців

Для автоматизованого опрацювання вибірки резюме та позицій складається список резюме з розширенням pdf та список вакансій, що складається з опису вакансій з розширенням txt. Потім усі результати заносяться в структуру даних DataFrame, див. рис. 3.14.

	Resume	Position	Jaccard similarity	Dice similarity	Bag-of- words cosine similarity	Bag-of- words sqrt-cos similarity	Bag-of- words improved sqrt-cos similarity	Tf-idf cosine similarity	Tf-idf sqrt-cos similarity	Tf-idf improved sqrt-cos similarity	...
0	Business analyst Male.pdf	Business Analyst.txt	0.066239	0.124248	0.183375	NaN	0.428223	0.110928	NaN	0.333059	...
1	Business analyst Male.pdf	Senior Software Developer.txt	0.063181	0.118852	0.128186	NaN	0.358031	0.074764	NaN	0.273430	...
2	Junior Frontend Developer Male.pdf	Business Analyst.txt	0.056604	0.107143	0.116304	NaN	0.341034	0.064117	NaN	0.253213	...
3	Junior Frontend Developer Male.pdf	Senior Software Developer.txt	0.076419	0.141988	0.175285	NaN	0.418670	0.104283	NaN	0.322929	...
4	Junior Full Stack Developer Female.pdf	Business Analyst.txt	0.039583	0.076152	0.109768	NaN	0.331313	0.060767	NaN	0.246509	...

Рис. 3.14 – Структура даних DataFrame для зберігання результатів опрацювання вибірки резюме (власна розробка)

Для виявлення найкращої моделі та метрики необхідне залучення експерта в сфері управління людськими ресурсами для ручної оцінки резюме на відповідні посади. Експертні оцінки виставляються за шкалою від 1 до 12, де оцінка 1 означає найбільш підходящого кандидата, а 12 – найменш підходящого кандидата на думку експерта, див. рис. 3.15.

Business Analyst, Amazon:

- 1 - Business analyst Male.pdf
- 7 - Junior Frontend Developer Male.pdf
- 3 - Junior Full Stack Developer Female.pdf
- 12 - Junior Software Developer Female.pdf
- 10 - Middle .NET Developer Male.pdf
- 9 - Middle Java Engineer Male.pdf
- 5 - Product Marketing Manager Female.pdf
- 6 - QA Specialist Female.pdf
- 4 - Senior .NET Developer Male.pdf
- 8 - Senior Electrical Engineer Male.pdf
- 3 - Senior Software Engineer Male.pdf
- 12 - Trainee UX-UI Designer Female.pdf

Senior Software Engineer, Google:

- 7 - Business analyst Male.pdf
- 8 - Junior Frontend Developer Male.pdf
- 5 - Junior Full Stack Developer Female.pdf
- 10 - Junior Software Developer Female.pdf
- 2 - Middle .NET Developer Male.pdf
- 7 - Middle Java Engineer Male.pdf
- 8 - Product Marketing Manager Female.pdf
- 3 - QA Specialist Female.pdf
- 3 - Senior .NET Developer Male.pdf
- 9 - Senior Electrical Engineer Male.pdf
- 1 - Senior Software Engineer Male.pdf
- 12 - Trainee UX-UI Designer Female.pdf

Рис. 3.15 – Експертні оцінки на позиції Business Analyst та Senior Software Engineer відповідно (власна розробка)

Потім розраховується зважений коефіцієнт Кендала Тау для виявлення залежності між експертними оцінками та ранжуванням результатів всіх моделей з метриками подібності, див. рис. 3.16. та рис. 3.17.

```

Weighted Tau coefficient for BERT cosine tensor similarity equals 0.8572436929811384
Weighted Tau coefficient for BERT improved sqrt-cos tensor similarity equals 0.8572436929811384
Weighted Tau coefficient for Word2Vec cosine similarity equals 0.5801721967564071
Weighted Tau coefficient for Word2Vec improved sqrt-cos similarity equals 0.5801721967564071
Weighted Tau coefficient for GloVe cosine similarity equals 0.5643702849778357
Weighted Tau coefficient for GloVe improved sqrt-cos similarity equals 0.5643702849778357
Weighted Tau coefficient for ElMo cosine tensor similarity equals 0.5592359727945382
Weighted Tau coefficient for ElMo improved sqrt-cos tensor similarity equals 0.5592359727945382
Weighted Tau coefficient for FastText cosine similarity equals 0.38780912975348425
Weighted Tau coefficient for Jaccard similarity equals 0.339690242290892
Weighted Tau coefficient for Dice similarity equals 0.339690242290892
Weighted Tau coefficient for GloVe sqrt-cos similarity equals 0.2326583048121824
Weighted Tau coefficient for FastText improved sqrt-cos similarity equals 0.1644450799144757
Weighted Tau coefficient for Bag-of-words cosine similarity equals 0.06397369357108959
Weighted Tau coefficient for Bag-of-words improved sqrt-cos similarity equals 0.06397369357108959
Weighted Tau coefficient for Tf-idf cosine similarity equals 0.06397369357108959
Weighted Tau coefficient for Tf-idf improved sqrt-cos similarity equals 0.06397369357108959
Weighted Tau coefficient for Word2Vec sqrt-cos similarity equals -0.31141906007369613
Weighted Tau coefficient for FastText sqrt-cos similarity equals -0.4317478471847848
Weighted Tau coefficient for Bag-of-words sqrt-cos similarity equals -0.483619504543505
Weighted Tau coefficient for Tf-idf sqrt-cos similarity equals -0.483619504543505
Weighted Tau coefficient for BERT sqrt-cos tensor similarity equals -0.483619504543505
Weighted Tau coefficient for ElMo sqrt-cos tensor similarity equals -0.483619504543505

```

Рис. 3.16 – Розраховані коефіцієнти кореляції Кендала Тау на позицію Business Analyst (власна розробка)

```

Weighted Tau coefficient for GloVe sqrt-cos similarity equals 0.8276414238016554
Weighted Tau coefficient for GloVe cosine similarity equals 0.31069970138022807
Weighted Tau coefficient for GloVe improved sqrt-cos similarity equals 0.31069970138022807
Weighted Tau coefficient for FastText improved sqrt-cos similarity equals 0.2969169791042248
Weighted Tau coefficient for Word2Vec cosine similarity equals 0.22757849998231766
Weighted Tau coefficient for Word2Vec improved sqrt-cos similarity equals 0.22757849998231766
Weighted Tau coefficient for ElMo cosine tensor similarity equals 0.14686813039501512
Weighted Tau coefficient for ElMo improved sqrt-cos tensor similarity equals 0.14686813039501512
Weighted Tau coefficient for BERT cosine tensor similarity equals 0.13905339040702688
Weighted Tau coefficient for BERT improved sqrt-cos tensor similarity equals 0.13905339040702688
Weighted Tau coefficient for Jaccard similarity equals 0.13644042210955845
Weighted Tau coefficient for Dice similarity equals 0.13644042210955845
Weighted Tau coefficient for FastText cosine similarity equals -0.004999068814471527
Weighted Tau coefficient for Tf-idf cosine similarity equals -0.050225202732833485
Weighted Tau coefficient for Tf-idf improved sqrt-cos similarity equals -0.050225202732833485
Weighted Tau coefficient for Bag-of-words sqrt-cos similarity equals -0.0932795010340347
Weighted Tau coefficient for Tf-idf sqrt-cos similarity equals -0.0932795010340347
Weighted Tau coefficient for BERT sqrt-cos tensor similarity equals -0.0932795010340347
Weighted Tau coefficient for ElMo sqrt-cos tensor similarity equals -0.0932795010340347
Weighted Tau coefficient for Bag-of-words cosine similarity equals -0.09366269901859926
Weighted Tau coefficient for Bag-of-words improved sqrt-cos similarity equals -0.09366269901859926
Weighted Tau coefficient for Word2Vec sqrt-cos similarity equals -0.36186863201773917
Weighted Tau coefficient for FastText sqrt-cos similarity equals -0.4188873196519247

```

Рис. 3.17 – Розраховані коефіцієнти кореляції Кендала Тау на позицію Senior Software Engineer (власна розробка)

На позицію Business Analyst в корпорацію Amazon найбільш ефективною моделлю виявилася BERT з косинусовою метрикою подібності з коефіцієнтом

кореляції Кендала Тау близько 0,857; на позицію Senior Software Engineer в корпорацію Google – модель GloVe з метрикою подібності sqrt-cos з коефіцієнтом кореляції Кендала Тау близько 0,827 відповідно.

На базі практики я провів експеримент на описі вакансії на позицію DevOps Engineer та вибірки з резюме розміром 20 вибраних випадковим чином з переліку кандидатів. Результати алгоритму було апробовано – як вже зазначалося, повний опис вакансії в компанії Whalebone s.r.o. є конфіденційним, тому було проведено анонімне опитування щодо задовільності роботи за шкалою від 1 до 10, де 1 – найнижча оцінка ефективності, 10 – найвища, див. рис. 3.18.



Рис. 3.18 – Анонімне опитування для оцінки роботи алгоритму (власна розробка)

На основі зібраної інформації, особливо результатів роботи алгоритму, я виявив можливості для подальшого вдосконалення у сфері управління людськими ресурсами бази практики, зокрема використання автоматизованих технологій за допомогою підходу SMART-аналізу для покращення процесу підбору кандидатів, забезпечуючи відповідність їхніх навичок потребам компанії Whalebone s.r.o. та сприяючи подальшому розвитку цього напрямку в компанії. Мої рекомендації також включають в себе активне залучення гібридного підходу “Human-in-the-loop” для забезпечення оптимального поєднання інтелектуальних технологій та експертної оцінки для об’єктивності та нейтральності процесу оцінювання навичок кандидатів.

ВИСНОВКИ

Отже, у кваліфікаційній бакалаврській роботі наведено обґрунтування теоретичних та прикладних засад алгоритмів для оцінювання навичок кандидатів на посади ІТ-фахівців за допомогою SMART-аналізу для підбору ідеального кандидата на вакантну посаду ІТ-фахівця з урахуванням специфіки ІТ-індустрії та запропоновано рекомендації щодо вдосконалення автоматизованого відбору кандидатів на базі практики.

За результатами дослідження можна зробити наступні висновки:

1. SMART-аналіз дійсно є новітнім та сучасним підходом для управління людськими ресурсами, зокрема для оцінювання навичок кандидатів на посади ІТ-фахівців, через його мультидисциплінарність, що прямо базується на використанні штучного інтелекту, зокрема машинного навчання, інтернету речей, хмарних обчислень та кібербезпеки. Цей напрям орієнтований перш за все на створення інтелектуальних систем та інструментів для збору, обробки, аналізу та інтерпретації даних з метою прийняття більш обґрунтованих та ефективних управлінських рішень за допомогою чітко визначених SMART-критеріїв;

2. Можливі підходи для оцінювання навичок кандидатів на посади ІТ-фахівців включають використання як традиційного відбору за допомогою ручного ранжування кандидатів, так й автоматизованого відбору за допомогою синтаксичних, граматичних, статистичних парсерів для видобування інформації, отримання Word Embeddings як результатів безконтекстних або контекстних моделей та порівняння векторів за допомогою мір подібності, а також роботу з рекомендаційними системами, що рекомендують роботу потенційним кандидатам, а відповідні кандидати – професіоналам з управління людськими ресурсами, з великими мовними моделями тощо;

3. Була складена порівняльна характеристика методів оцінювання навичок

кандидатів на посади IT-фахівців, де зазначено ефективність кожного з методів SMART-аналізу. Варто зазначити, що постановка та методи задач оцінювання навичок кандидатів, вхідні дані можуть суттєво відрізнятися, тому необхідний комплексний та індивідуальний підхід, що дозволяє обирати найбільш ефективне та оптимальне рішення;

4. Було розроблено алгоритм для ефективного оцінювання навичок кандидатів на посади IT-фахівців для підбору ідеального кандидата, враховуючи та забезпечуючи об'єктивність та нейтральність процесу оцінювання, що складається з 3 частин, а саме: парсингу інформації з резюме кандидатів за допомогою таких бібліотек, як PyPDF, re та NLTK; векторизації отриманої інформації за допомогою таких методів SMART-аналізу, як Bag-of-words, TF-IDF, Word2Vec, GloVe та fasttext, узагальнення знань про слова та врахування їхнього контексту в тексті за допомогою таких моделей, як ELMo та BERT; порівняння векторів корпусів резюме та посадових інструкцій для оцінки навичок, досвіду та кваліфікації кандидата за допомогою таких мір подібності, як косинусова, $\sqrt{\text{cos}}$, ISC, Жаккарда, Серенсена-Дайса. Також відбулася апробація алгоритму на базі практики з точністю близько 0,83 коефіцієнта кореляції Кендала Тау з безпосередньою оцінкою експертів бази практики задовільності роботи алгоритму та з урахуванням специфіки IT-індустрії загалом.

Наостанок варто сказати, що людина – найвища цінність, адже жодна її комплексна, творча діяльність, риси характеру, людяність не можуть бути повністю замінені на інтелектуальні системи. Найбільш перспективним напрямком автоматизації є розумна або SMART-автоматизація, в якій покладено цінності трансгуманізму, де технології використовуються для розширення можливостей людини, а не для її заміни. Таким чином, SMART-автоматизація стає інструментом для реалізації людського потенціалу в повній мірі, дозволяючи людям зосереджуватися на тих завданнях, які вимагають людської інтуїції, креативності та емоційного інтелекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cambridge Dictionary, Cambridge University Press. URL: <https://dictionary.cambridge.org/dictionary/english/human-resource-management> (Accessed: 20.05.2024).
2. Armstrong M., Taylor S. Armstrong's Handbook of Human Resource Management Practice: A Guide to the Theory and Practice of People Management, 16th edition. London; New York, NY: Kogan Page, 2023. 760 p.
3. Top 5 Priorities for HR Leaders in 2024, Gartner. URL: <https://www.gartner.com/en/human-resources/trends/top-priorities-for-hr-leaders> (Accessed: 20.05.2024).
4. Акулов Я., Шевченко М. Епоха навичок, інновацій та сильних лідерів: компанія «Делойт» назвала HR-тренди 2023. URL: <https://www2.deloitte.com/ua/uk/pages/press-room/press-release/2023/human-capital-trends-event.html> (Дата звернення: 20.05.2024).
5. Brown P. 9 workplace trends for 2024. URL: <https://www.fdmgroup.com/news-insights/workplace-trends/> (Accessed: 20.05.2024).
6. The changing face of HR in 2024, Sage. URL: <https://www.sage.com/en-gb/-/media/files/sagedotcom/master/documents/pdf/sage%20people/infographic-the-changing-face-of-hr-sage-uk.pdf> (Accessed: 20.05.2024).
7. Turner J. 8 Learnings from Gartner's Hype Cycle for HR Transformation. URL: <https://www.gartner.com/en/articles/8-learnings-from-gartner-s-hype-cycle-for-hr-transformation> (Accessed: 20.05.2024).
8. Boogaard K. How to write SMART goals. URL: <https://www.atlassian.com/blog/productivity/how-to-write-smart-goals> (Accessed: 20.05.2024).
9. Artificial intelligence and a new era of human resources, IBM Consulting. URL: <https://www.ibm.com/blog/artificial-intelligence-and-a-new-era-of-human-resources/> (Accessed: 20.05.2024).
10. HR Transformation. Modernize HR. Fuel growth with talent., Price waterhouseCoopers. URL: <https://www.pwc.com/us/en/services/consulting/business-transformation/workforce-transformation/hr-transformation.html> (Accessed: 20.05.2024).
11. Walters J. The Internet of Things (IoT) and its applications for HR. URL: <https://www.hcamag.com/nz/specialisation/hr-technology/the-internet-of-things-iot-and-its-applications-for-hr/465230> (Accessed: 20.05.2024).
12. Global Cybersecurity Outlook 2023, World Economic Forum. URL: https://www3.weforum.org/docs/WEF_Global_Security_Outlook_Report_2023.pdf (Accessed: 20.05.2024).
13. Harvard Business Review Research Reveals How AI Is Making the Recruiting

Process More Effective – and Positively Impacting Business Success, Paradox. URL: <https://www.prnewswire.com/news-releases/harvard-business-review-research-reveals-how-ai-is-making-the-recruiting-process-more-effective--and-positively-impacting-business-success-301900660.html> (Accessed: 20.05.2024).

14. Oracle Cloud HCM, SOLUTION OVERVIEW, Oracle. URL: <https://www.oracle.com/a/ocom/docs/oracle-hcm-cloud-overview.pdf> (Accessed: 20.05.2024).

15. Oracle AI for Fusion Applications, Oracle. URL: <https://www.oracle.com/a/ocom/docs/applications/oracle-ai-for-fusion-apps.pdf> (Accessed: 20.05.2024).

16. Oracle Recruiting and Recruiting Booster, Oracle. URL: <https://www.oracle.com/human-capital-management/recruiting/#attraction> (Accessed: 20.05.2024).

17. Awati R., Burns E. IBM Watson supercomputer. URL: <https://www.techtarget.com/searchenterpriseai/definition/IBM-Watson-supercomputer> (Accessed: 20.05.2024).

18. IBM Watson to watsonx, IBM. URL: <https://www.ibm.com/watson> (Accessed: 20.05.2024).

19. IBM Unveils the Watsonx Platform to Power Next-Generation Foundation Models for Business, IBM. URL: <https://newsroom.ibm.com/2023-05-09-IBM-Unveils-the-Watsonx-Platform-to-Power-Next-Generation-Foundation-Models-for-Business> (Accessed: 20.05.2024).

20. IBM watsonx.ai: Open source, pre-trained foundation models make AI and automation easier than ever before, IBM. URL: <https://www.ibm.com/blog/ibm-watsonx-ai-open-source-pre-trained-foundation-models-make-ai-and-automation-easier-than-ever-before/> (Accessed: 20.05.2024).

21. Anaconda Partners with IBM watsonx to Deliver Enterprise Scale AI Solutions, Anaconda. URL: <https://www.datanami.com/this-just-in/anaconda-partners-with-ibm-watsonx-to-deliver-enterprise-scale-ai-solutions/> (Accessed: 20.05.2024).

22. watsonx Orchestrate: Get more time back for things that matter, IBM. URL: <https://www.ibm.com/products/watsonx-orchestrate> (Accessed: 20.05.2024).

23. Overview of IBM watsonx Orchestrate, IBM. URL: <https://www.ibm.com/docs/en/watson-orchestrate?topic=orchestrate-overview> (Accessed: 20.05.2024).

24. Human resources process automation: Want to offload tedious tasks? Just ask. Put the human back in human resources, IBM. URL: <https://www.ibm.com/products/watsonx-orchestrate/candidate-sourcing> (Accessed: 20.05.2024).

25. Human capital management products: Cloud solutions to address all your HR needs – core HR and payroll, talent management, HR analytics, workforce planning, and more, SAP. URL: <https://www.sap.com/products/hcm/solutions.html> (Accessed: 20.05.2024).

26. Human capital management: Take your business to the next level with a truly global AI-enabled suite that fuels the success of every individual and drives org agility at scale, SAP. URL: <https://www.sap.com/products/hcm.html> (Accessed: 20.05.2024).
27. Artificial Intelligence and Machine Learning at SAP: Become an intelligent, sustainable enterprise with built-in artificial intelligence capabilities for your core business processes, SAP. URL: <https://pages.community.sap.com/topics/machine-learning> (Accessed: 20.05.2024).
28. What is Joule? – SAP Help Portal Documentation, SAP. URL: <https://help.sap.com/docs/joule/serviceguide/what-is-joule> (Accessed: 20.05.2024).
29. SAP SuccessFactors Recruiting: Source, engage, and hire the right talent with a recruiting management solution that's agile and intelligent, SAP. URL: <https://www.sap.com/products/hcm/recruiting-software.html> (Accessed: 20.05.2024).
30. Bahr K. Resume Screening. URL: <https://eddy.com/hr-encyclopedia/resume-screening/> (Accessed: 20.05.2024).
31. Guide for recruiters: how to effectively evaluate job applicants' CVs, PinyaHR. URL: <https://en.pinya.hr/blog/pruvodce-pro-personalisty-jak-efektivne-hodnotit-zivotopisy-uchazecu-o-praci> (Accessed: 20.05.2024).
32. Ринок праці 2023: найменше вакансій за 5 років, але +15% нетехнічних за рік, DOU.ua. URL: <https://dou.ua/lenta/articles/jobs-and-trends-2023/> (Дата звернення: 20.05.2024).
33. Bloomenthal A. Clarke C., Munichello K., Resume Definition: Meaning, Purpose, and What Should Not Be on Yours. URL: <https://www.investopedia.com/terms/r/resume.asp> (Accessed: 20.05.2024).
34. What is natural language processing (NLP)?, IBM. URL: <https://www.ibm.com/topics/natural-language-processing> (Accessed: 20.05.2024).
35. Gao C., Zhang W., Lam W., Bing L. Easy-to-Hard Learning for Information Extraction. *Findings of the Association for Computational Linguistics: ACL 2023*. P. 11913–11930.
36. Palshikar G.K., Pawar P., Banerjee A.S., Srivastava R., Ramrakhiani N. RINX: A system for information and knowledge extraction from resumes. *Data & Knowledge Engineering*. 2023. Volume 147. P. 102202.
37. 4 Key Resume Parsing Techniques and Why You Only Need the Last One, Affinda. URL: <https://www.affinda.com/recruitment-ai/resume-parsing-techniques> (Accessed: 20.05.2024).
38. Birt J. What Is Resume Parsing? (With Benefits and Tips). URL: <https://www.indeed.com/career-advice/resumes-cover-letters/resume-parsing> (Accessed: 20.05.2024).
39. Word Embeddings in NLP, geeksforgeeks.org. URL: <https://www.geeksforgeeks.org/word-embeddings-in-nlp/> (Accessed: 20.05.2024).
40. Liu Z., Sun M., Lin Y. Representation Learning for Natural Language Processing, 2nd edition. Singapore; Singapore: Springer, 2023. 491 p.
41. Barla N. The Ultimate Guide to Word Embeddings. URL:

<https://neptune.ai/blog/word-embeddings-guide> (Accessed: 21.05.2024).

42. Demir N. Understanding Text Embeddings: A Simplified Guide. URL: <https://blog.demir.io/understanding-text-embeddings-a-simplified-guide-344238f2e5a4> (Accessed: 21.05.2024).

43. Topper N. Bag of Words Model in NLP Explained. URL: <https://builtin.com/machine-learning/bag-of-words> (Accessed: 21.05.2024).

44. Aydin A. 4 – Bag of Words Model in NLP. URL: <https://ayselaydin.medium.com/4-bag-of-words-model-in-nlp-434cb38cdd1b> (Accessed: 21.05.2024).

45. Sharma M., Choudhary G., Susan S. Resume Classification using Elite Bag-of-Words Approach. *International Conference on Smart Systems and Inventive Technology (ICSSIT)*. 2023. Volume 5. P. 1409-1413.

46. Li P., Ren G.-J., Gentile A., DeLuca C., Tan D., Gopisetty S. Long-Form Information Retrieval for Enterprise Matchmaking. *SIGIR '23: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2023. Volume 46. P. 3260-3264.

47. En Y., Long Z. Research on the Weighting Method Based on Tf-IDF and Apriori Algorithm. *IEEE 6th International Conference on Information Systems and Computer Aided Education (ICISCAE)*. 2023. Volume 6. P. 1003-1005.

48. A Guide on Word Embeddings in NLP, Turing. URL: [https://www.turing.com/kb/guide-on-word-embeddings-in-nlp#bert-\(bidirectional-encoder-representations-from-transformers\)](https://www.turing.com/kb/guide-on-word-embeddings-in-nlp#bert-(bidirectional-encoder-representations-from-transformers)) (Accessed: 21.05.2024).

49. Borgave S., Gavali V., Kudumbale S., Saoji S. Resume Shortlisting and Grading using TF-IDF, Cosine Similarity and KNN. *Journal of Emerging Technologies and Innovative Research (JETIR)*. 2023. Volume 10, Issue 5. P. 423-431.

50. Zaland O., Abulaish M., Fazil M. A Comprehensive Empirical Evaluation of Existing Word Embedding Approaches. *ArXiv abs/2303.07196*. 2023. P. 1-28.

51. Logunova I. Word2Vec: Why Do We Need Word Representations? URL: <https://serokell.io/blog/word2vec> (Accessed: 21.05.2024).

52. Chouhan S. Word2Vec – Analogical Reasoning. URL: <https://sanjayc.medium.com/word2vec-analogical-reasoning-d47d3a66b9fb> (Accessed: 21.05.2024).

53. Xia H. Continuous-bag-of-words and Skip-gram for word vector training and text classification. *The 2nd International Conference on Computing Innovation and Applied Physics (CONF-CIAP)*. 2023. Volume 2634. P. 1-9.

54. Pudasaini S., Shakya S., Lamichhane S., Adhikari S., Tamang A., Adhikari S. Scoring of Resume and Job Description Using Word2vec and Matching Them Using Gale-Shapley Algorithm. *Expert Clouds and Applications, Lecture Notes in Networks and Systems*. 2022. Volume 209. P. 705-713.

55. Pre-trained Word embedding using Glove in NLP models, [geeksforgeeks.org](https://www.geeksforgeeks.org/pre-trained-word-embedding-using-glove-in-nlp-models/). URL: <https://www.geeksforgeeks.org/pre-trained-word-embedding-using-glove-in-nlp-models/> (Accessed: 21.05.2024).

56. Bestvater S., Shah S. Analyzing text for distinctive terms using pointwise mutual information. URL: <https://www.pewresearch.org/decoded/2022/07/13/analyzing-text-for-distinctive-terms-using-pointwise-mutual-information/> (Accessed: 21.05.2024).
57. NLP Tutorials – Part 5: GloVe, Applied Singularity. URL: <https://appliedsingularity.com/2022/01/25/nlp-tutorials-part-5-glove/> (Accessed: 21.05.2024).
58. Raut S., Rathod A., Sharma P., Bhosale P., Zope B. Best-Fit: Best Fit Employee Recommendation. *IEEE Pune Section International Conference (PuneCon)*. 2022. Volume 1. P. 1-6.
59. Han S., Pino J. Welcome fastText to the Hugging Face Hub. URL: <https://huggingface.co/blog/fasttext> (Accessed: 21.05.2024).
60. Durna M.B. Advanced Word Embeddings: Word2Vec, GloVe, and FastText. URL: <https://medium.com/@mervebdurna/advanced-word-embeddings-word2vec-glove-and-fasttext-26e546ffedbd> (Accessed: 21.05.2024).
61. Edward A., Camacho-Collados J., De Ribaupierre H., Preece A. Go Simple and Pre-Train on Domain-Specific Corpora: On the Role of Training Data for Text Classification. *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*. 2020. Volume 28. P. 5522-5529.
62. Fodor J., De Deyne S., Suzuki S. The Importance of Context in the Evaluation of Word Embeddings: The Effects of Antonymy and Polysemy. *Proceedings of the 15th International Conference on Computational Semantics (IWCS)*. 2023. Volume 15. P. 155-172.
63. Schulze J. What Is the BERT Model and How Does It Work? URL: <https://www.coursera.org/articles/bert-model> (Accessed: 21.05.2024).
64. Muller B. BERT 101 – State Of The Art NLP Model Explained. URL: <https://huggingface.co/blog/bert-101> (Accessed: 21.05.2024).
65. What is a transformer model?, IBM. URL: <https://www.ibm.com/topics/transformer-model> (Accessed: 21.05.2024).
66. Brownlee J. The Transformer Model. URL: <https://machinelearningmastery.com/the-transformer-model/> (Accessed: 21.05.2024).
67. Ferrer J. How Transformers Work: A Detailed Exploration of Transformer Architecture. URL: <https://www.datacamp.com/tutorial/how-transformers-work> (Accessed: 21.05.2024).
68. Application of LLM Agents in Recruitment: A Novel Framework for Resume Screening. *arXiv: abs/2401.08315*. 2024. P. 1-14.
69. Helwan A. ELMo Embeddings. URL: <https://abdulkaderhelwan.medium.com/elmo-e-c8e6fc068882> (Accessed: 21.05.2024).
70. ElMo, Allen Institute for AI. URL: <https://allenai.org/allennlp/software/elmo> (Accessed: 21.05.2024).
71. Jayakody J., Vidanagama V., Perera I., Herath H. ELMo Layer Embedding Comparison with Short Text Classification. *3rd Asian Conference on Innovation in Technology (ASIANCON)*. 2023. Volume 3. P.1-6.

72. Alsharef A., Nassour H., Garg S. Exploring the Efficiency of Text-Similarity Measures in Automated Resume Screening for Recruitment. *10th International Conference on Computing for Sustainable Global Development (INDIACom)*. 2023. Volume 10. P. 36-42.
73. Yasser M., Hussain K.F., Ali S.A. Comparative Analysis of Similarity Methods in High-Dimensional Vectors: A Review. *2nd International Conference on Artificial Intelligence Science and Applications in Industry and Society (CAISIS)*. 2023. Volume 2. P. 1-6.
74. Amin M.D., Harkare A.C., Parmar N.S., Wadhwa R.R., Patil R.A. Real Time Data based Automated Resume Classification and Job Matching using SVC, Jaccard Index and Cosine Similarity. *7th International Conference on Computer Applications in Electrical Engineering-Recent Advances (CERA)*. 2023. Volume 7. P. 1-6.
75. De D., Dwivedi R., Allwani N. Combined Application of Various Techniques for Personalized Job Recommendation. *International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF)*. 2023. Volume 1. P. 1-7.
76. Özçevik Y., Yücalar F., Demircioğlu M. Determining a Proper Text Similarity Approach for Resume Parsing Process in a Digitized HR Software. *Celal Bayar University Journal of Science*. 2022. Volume 18. P. 371-378.
77. Mashayekhi Y., Li N., Kang B., Lijffijt J., De Bie T. A challenge-based survey of e-recruitment recommendation systems. *ACM Computing Surveys*. 2024. P. 1-33.
78. C111 – Discrimination (Employment and Occupation) Convention, 1958 (No. 111), International Labour Organization. URL: https://normlex.ilo.org/dyn/normlex/en/f?p=NORMLEXPUB:12100:0::NO::P12100_Ilo_Code:C111 (Accessed: 21.05.2024).
79. Escudé M., Onuchic P., Sinander L., Valenzuela-Stookey Q. Statistical discrimination and statistical informativeness. *ArXiv abs/2205.07128*. 2022. P. 1-11.
80. Fabris A., Baranowska N., Dennis M.J., Graus D., Hacker P., Saldivar J. Fairness and Bias in Algorithmic Hiring: a Multidisciplinary Survey. *ArXiv abs/2309.13933*. 2023. P. 1-53.
81. Algorithmic Discrimination Protections, The White House. URL: [https://www.whitehouse.gov/ostp/ai-bill-of-rights/algorithmic-discrimination-protections-2/#:~:text=Algorithmic%20discrimination%20occurs%20when%20automated,orientation\)%2C%20religion%2C%20age%2C](https://www.whitehouse.gov/ostp/ai-bill-of-rights/algorithmic-discrimination-protections-2/#:~:text=Algorithmic%20discrimination%20occurs%20when%20automated,orientation)%2C%20religion%2C%20age%2C) (Accessed: 21.05.2024).
82. Incident 37: Female Applicants Down-Ranked by Amazon Recruiting Tool, incidentdatabase.ai. URL: <https://incidentdatabase.ai/cite/37/> (Accessed: 21.05.2024).
83. Rieskamp J., Hofeditz L., Mirbabaie M., Stieglitz S. Approaches to Improve Fairness when Deploying AI-based Algorithms in Hiring – Using a Systematic Literature Review to Guide Future Research. *Proceedings of the 7th Annual Hawaii International Conference on System Sciences (HICSS)*. 2023. Volume 7. P. 216-225.
84. Kelan E. Algorithmic inclusion: Shaping the predictive algorithms of artificial intelligence in hiring. *Human Resource Management Journal*. 2023. P. 1-14.

85. Albaroudi E., Mansouri T., Alameer A. A Comprehensive Review of AI Techniques for Addressing Algorithmic Bias in Job Hiring. *AI*. 2024. Volume 5. P. 383-404.
86. Zhuk D. The Ethics of Artificial Intelligence: Challenges and Frameworks. *23rd International Scientific and Technical Conference Artificial Intelligence and Intelligent Systems (AIIS)*. 2023. Volume 23. P. 16-21.
87. Artificial Intelligence Act: MEPs adopt landmark law, European Parliament. URL: <https://www.europarl.europa.eu/news/en/press-room/20240308IPR19015/artificial-intelligence-act-meps-adopt-landmark-law> (Accessed: 21.05.2024).
88. Blueprint for an AI Bill of Rights, The White House. URL: <https://www.whitehouse.gov/ostp/ai-bill-of-rights/> (Accessed: 21.05.2024).
89. Business Analyst, Amazon Jobs. URL: <https://www.amazon.jobs/en/jobs/2608538/business-analyst> (Accessed: 21.05.2024).
90. Senior Software Engineer Manager, Embedded Systems/Firmware, Platforms Infrastructure Engineering, Google Careers. URL: <https://www.google.com/about/careers/applications/jobs/results/131951529977356998-senior-software-engineer-manager-embedded-systemsfirmware-platforms-infrastructure-engineering> (Accessed: 21.05.2024).
91. Anonymisation and pseudonymisation, Data Protection Commission. URL: <https://www.dataprotection.ie/en/dpc-guidance/anonymisation-pseudonymisation> (Accessed: 21.05.2024).

ДОДАТКИ

Додаток А

Код програмного рішення

```
# Import all necessary libraries
import re
import nltk
import torch
import warnings
import pandas as pd
import numpy as np
import tensorflow as tf
import tensorflow_hub as hub
from pypdf import PdfReader
from itertools import product
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from scipy import stats
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from gensim.models import KeyedVectors
from transformers import BertTokenizer, BertModel
tf.compat.v1.enable_eager_execution()
warnings.filterwarnings('ignore')

# Download all necessary sets of words
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')
nltk.download('omw-1.4')

# Text preprocessing functions

# Read resume
def read_resume(resume) -> str:
    try:
        reader = PdfReader(resume)
        number_of_pages = reader.get_num_pages()
        text = ""
        for page_number in range(number_of_pages):
            page = reader.get_page(page_number)
            page_content = page.extract_text()
```

```

        text+=page_content
    return text
except Exception as e:
    return f'The resume {resume} was not scraped due to {e}.'

# Read position description
def read_position(position) -> str:
    try:
        with open(position) as file:
            return file.read()
    except Exception as e:
        return f'The position {position} was not scraped due to {e}.'

# Tokenize description
def tokenize_description(description) -> list:
    try:
        return nltk.sent_tokenize(description)
    except Exception as e:
        return f'The description {description} was not scraped due to {e}.'

# Lemmatization; stopwords, punctuation, and whitespace removal
def preprocessing_text(text) -> str:
    try:
        lemmatizer = WordNetLemmatizer()
        text = text.split()
        text = [lemmatizer.lemmatize(word) for word in text if not word in set(stopwords.words('english'))]
        text = ''.join(text)
        text = re.sub(r'\d+', '', text)
        text = re.sub(r'^\w\s', '', text)
        text = re.sub(r'\s+', '', text)
        text = text.lower().strip()
        return text
    except Exception as e:
        return f'The text was not preprocessed due to {e}.'

# Create one corpus from smaller corpuses
def create_corpus(text) -> str:
    try:
        return ''.join(text)
    except Exception as e:
        return f'The corpus was not created due to {e}.'

# Metrics and helping functions

# Cosine similarity - Euclidean distance
def cos_sim(a, b):
    try:

```

```

    return np.dot(a, b) / (np.linalg.norm(a) * np.linalg.norm(b))
except Exception as e:
    return f'The cosine similarity was not calculated due to {e}.'

# Sqrt-cos similarity - Hellinger distance -> needs to be normalized on the scale from 0 to 1 due to non-support of
negative values
def sqrt_cos_sim(a, b):
    try:
        def hellinger_distance(a, b):
            return np.sqrt(np.sum((np.sqrt(a) - np.sqrt(b))**2)) / np.sqrt(2)
        return np.sqrt(1 - hellinger_distance(a, b)**2)
    except Exception as e:
        return f'Sqrt-cos similarity was not calculated due to {e}.'

# MinMax scaler
def normalize_vector(vector):
    try:
        return (vector - np.min(vector)) / (np.max(vector) - np.min(vector))
    except Exception as e:
        return f'Vector normalization was not calculated due to {e}.'

# Improved sqrt-cos similarity - Manhattan distance
def improved_sqrt_cos_sim(a, b):
    try:
        return np.sqrt(np.dot(a, b)) / (np.sqrt(np.linalg.norm(a)) * np.sqrt(np.linalg.norm(b)))
    except Exception as e:
        return f'Improved sqrt-cos similarity was not calculated due to {e}.'

# Jaccard similarity
def jaccard_sim(a, b):
    try:
        set1 = set(a)
        set2 = set(b)
        intersection = len(set1.intersection(set2))
        union = len(set1.union(set2))
        similarity = intersection / union
        return similarity
    except Exception as e:
        return f'Jaccard similarity was not calculated due to {e}.'

# Dice similarity
def dice_sim(a, b):
    try:
        set1 = set(a)
        set2 = set(b)
        intersection = len(set1.intersection(set2))
        dice = (2.0 * intersection) / (len(set1) + len(set2))

```

```

    return dice
except Exception as e:
    return f'Dice similarity was not calculated due to {e}.'

# Cosine similarity for tensors - Euclidean distance
def cosine_similarity_tensorflow(a, b):
    try:
        a = tf.reshape(a, [-1])
        b = tf.reshape(b, [-1])
        dot_product = tf.reduce_sum(a * b)
        magnitude1 = tf.sqrt(tf.reduce_sum(a * a))
        magnitude2 = tf.sqrt(tf.reduce_sum(b * b))
        similarity = dot_product / (magnitude1 * magnitude2)
        return similarity
    except Exception as e:
        return f'The cosine similarity was not calculated due to {e}.'

# Sqrt-cos similarity for tensors - Hellinger distance -> needs to be normalized on the scale from 0 to 1 due to non-
support of negative values
def sqrt_cos_sim_tensorflow(a, b):
    try:
        a = tf.clip_by_value(a, 0.0, 1.0)
        b = tf.clip_by_value(b, 0.0, 1.0)
        def hellinger_distance(a, b):
            return tf.sqrt(tf.reduce_sum(tf.square(tf.sqrt(a) - tf.sqrt(b)))) / tf.sqrt(2.0)
        similarity = tf.sqrt(1 - tf.square(hellinger_distance(a, b)))
        return similarity
    except Exception as e:
        return f'Sqrt-cos similarity was not calculated due to {e}.'

# Improved sqrt-cos similarity for tensors - Manhattan distance
def improved_sqrt_cos_sim_tensorflow(a, b):
    try:
        a = tf.reshape(a, [-1])
        b = tf.reshape(b, [-1])
        dot_product = tf.sqrt(tf.reduce_sum(tf.multiply(a, b)))
        magnitude1 = tf.sqrt(tf.norm(a))
        magnitude2 = tf.sqrt(tf.norm(b))
        similarity = dot_product / (magnitude1 * magnitude2)
        return similarity
    except Exception as e:
        return f'Improved sqrt-cos similarity was not calculated due to {e}.'

# Models word embeddings functions

# Bag-of-words model
def bag_of_words(resume_corpus, position_corpus) -> list:

```

```

try:
    resume_tokens = word_tokenize(resume_corpus)
    position_tokens = word_tokenize(position_corpus)
    vocabulary = set(resume_tokens + position_tokens)
    vectorizer = CountVectorizer(vocabulary=vocabulary)
    vectorizer.fit([resume_corpus, position_corpus])
    resume_bow = vectorizer.transform([resume_corpus])
    position_bow = vectorizer.transform([position_corpus])
    feature_names = vectorizer.get_feature_names_out()
    df = pd.DataFrame(
        data=[resume_bow[0].toarray()[0], position_bow[0].toarray()[0]],
        columns=feature_names,
        index=[resume_corpus, position_corpus])
    return [df, resume_bow[0].toarray()[0], position_bow[0].toarray()[0]]
except Exception as e:
    return f'Bag-of-words model was not created due to {e}.'

```

TF-IDF model

```

def tf_idf(resume_corpus, position_corpus) -> list:
    try:
        resume_tokens = word_tokenize(resume_corpus)
        position_tokens = word_tokenize(position_corpus)
        vocabulary = set(resume_tokens + position_tokens)
        vectorizer = TfidfVectorizer(vocabulary=vocabulary)
        vectorizer.fit([resume_corpus, position_corpus])
        resume_bow = vectorizer.transform([resume_corpus])
        position_bow = vectorizer.transform([position_corpus])
        feature_names = vectorizer.get_feature_names_out()
        df = pd.DataFrame(
            data=[resume_bow[0].toarray()[0], position_bow[0].toarray()[0]],
            columns=feature_names,
            index=[resume_corpus, position_corpus])
        return [df, resume_bow[0].toarray()[0], position_bow[0].toarray()[0]]
    except Exception as e:
        return f'Tf-idf model was not created due to {e}.'

```

Word2Vec, GloVe, and fasttext build model

```

def build_model_ncontext(model_path) -> KeyedVectors:
    try:
        return KeyedVectors.load(model_path)
    except Exception as e:
        return f'The model {model_path} was not built due to {e}.'

```

Word2Vec, GloVe, and fasttext word embeddings

```

def ncontext_word_embeddings(resume_tokens, position_tokens, model, vectors_size) -> list:
    try:
        p_sen1 = [item for sublist in resume_tokens for item in sublist]

```

```

p_sen2=[item for sublist in position_tokens for item in sublist]
sen_vec1 = np.zeros(vectors_size)
sen_vec2 = np.zeros(vectors_size)
for val in p_sen1:
    try:
        sen_vec1 = np.add(sen_vec1, model[val])
    except:
        sen_vec1 = np.add(sen_vec1, 0)
        continue
for val in p_sen2:
    try:
        sen_vec2 = np.add(sen_vec2, model[val])
    except:
        sen_vec2 = np.add(sen_vec2, 0)
        continue
return [sen_vec1, sen_vec2]
except Exception as e:
    return f'Word embeddings were not obtained by model {model} due to {e}.'
```

Build BERT model and obtain word embeddings

```
def context_word_embeddings_bert(model_name, resume_corpus, position_corpus, tokenizer, model) -> list:
```

```

    try:
        return [get_word_embeddings(resume_corpus, tokenizer, model), get_word_embeddings(position_corpus, tokenizer,
model)]
    except Exception as e:
        return f'Word embeddings were not obtained by BERT model due to {e}.'
```

Helping function to obtain word embeddings from BERT model

```
def get_word_embeddings(sentence, tokenizer, model):
```

```

    try:
        tokens = tokenizer(sentence, return_tensors='pt', padding=True, truncation=True)
        with torch.no_grad():
            outputs = model(**tokens)
            hidden_states = outputs.hidden_states
            embeddings = hidden_states[-1].squeeze(0) # Embeddings from last layer
        if embeddings.size(0) > 100:
            embeddings = embeddings[:100, :]
        return embeddings
    except Exception as e:
        return f'Word embeddings were not obtained by BERT model due to {e}.'
```

Build ElMo model and obtain word embeddings

```
def context_word_embeddings_elmo(elmo_model, resume_corpus, position_corpus):
```

```

    try:
        return elmo_model.signatures["default"]([resume_corpus, position_corpus])["elmo"]
    except Exception as e:
        return f'Word embeddings were not obtained by ElMo model due to {e}.'
```

```

# Read resume and position descriptions
resume = read_resume("Business analyst Male.pdf")
position = read_position("Business Analyst.txt")

# Tokenize descriptions into list of sentences
resume = tokenize_description(resume)
position = tokenize_description(position)

# Preprocess sentences in descriptions
resume = [preprocessing_text(sentence) for sentence in resume]
position = [preprocessing_text(sentence) for sentence in position]

# Tokenize words in sentences in descriptions
resume_tokens = [word_tokenize(sentence) for sentence in resume]
position_tokens = [word_tokenize(sentence) for sentence in position]

# Create resume and position corpuses
corpus_resume = create_corpus(resume)
corpus_position = create_corpus(position)

# Binary similarity
flatten_position_tokens = sum(position_tokens, [])
flatten_resume_tokens = sum(resume_tokens, [])
print("Jaccard similarity:", jaccard_sim(flatten_resume_tokens, flatten_position_tokens))
print("Dice similarity:", dice_sim(flatten_resume_tokens, flatten_position_tokens))

# Bag-of-words word embeddings
bow_resume = bag_of_words(corpus_resume, corpus_position)[1]
bow_position = bag_of_words(corpus_resume, corpus_position)[2]

# Bag-of-words word embeddings similarity
print("Bag-of-words cosine similarity:", cos_sim(bow_resume, bow_position))
print("Bag-of-words sqrt-cos similarity:", sqrt_cos_sim(normalize_vector(bow_resume),
normalize_vector(bow_position)))
print("Bag-of-words improved sqrt-cos similarity:", improved_sqrt_cos_sim(bow_resume, bow_position))

# Tf-idf word embeddings
tf_idf_resume = tf_idf(corpus_resume, corpus_position)[1]
tf_idf_position = tf_idf(corpus_resume, corpus_position)[2]

# Tf-idf word embeddings similarity
print("Tf-idf cosine similarity:", cos_sim(tf_idf_resume, tf_idf_position))
print("Tf-idf sqrt-cos similarity:", sqrt_cos_sim(normalize_vector(tf_idf_resume), normalize_vector(tf_idf_position)))
print("Tf-idf improved sqrt-cos similarity:", improved_sqrt_cos_sim(tf_idf_resume, tf_idf_position))

# Build Deep Learning models (without context)

```

```

model_w2v = build_model_ncontext(model_path="word2vec-google-news-300.model")
model_glove = build_model_ncontext(model_path="glove-wiki-gigaword-300.model")
model_fasttext = build_model_ncontext(model_path="fasttext-wiki-news-subwords-300.model")

# Resume and position word embeddings by Word2Vec model (without context)
w2v_resume = ncontext_word_embeddings(resume_tokens, position_tokens, model_w2v, 300)[0]
w2v_position = ncontext_word_embeddings(resume_tokens, position_tokens, model_w2v, 300)[1]

print("Word2Vec cosine similarity:", cos_sim(w2v_resume, w2v_position))
print("Word2Vec sqrt-cos similarity:", sqrt_cos_sim(normalize_vector(w2v_resume), normalize_vector(w2v_position)))
print("Word2Vec improved sqrt-cos similarity:", improved_sqrt_cos_sim(w2v_resume, w2v_position))

# Resume and position word embeddings by GloVe model (without context)
glove_resume = ncontext_word_embeddings(resume_tokens, position_tokens, model_glove, 300)[0]
glove_position = ncontext_word_embeddings(resume_tokens, position_tokens, model_glove, 300)[1]

print("GloVe cosine similarity:", cos_sim(glove_resume, glove_position))
print("GloVe sqrt-cos similarity:", sqrt_cos_sim(normalize_vector(glove_resume), normalize_vector(glove_position)))
print("GloVe improved sqrt-cos similarity:", improved_sqrt_cos_sim(glove_resume, glove_position))
# Resume and position word embeddings by fasttext model (without context)
fasttext_resume = ncontext_word_embeddings(resume_tokens, position_tokens, model_fasttext, 300)[0]
fasttext_position = ncontext_word_embeddings(resume_tokens, position_tokens, model_fasttext, 300)[1]

print("fasttext cosine similarity:", cos_sim(fasttext_resume, fasttext_position))
print("fasttext sqrt-cos similarity:", sqrt_cos_sim(normalize_vector(fasttext_resume),
normalize_vector(fasttext_position)))
print("fasttext improved sqrt-cos similarity:", improved_sqrt_cos_sim(fasttext_resume, fasttext_position))

# Build BERT model (with context)
bert_tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")
bert_model = BertModel.from_pretrained("bert-base-uncased", output_hidden_states=True)
bert_embeddings = context_word_embeddings_bert("bert-base-uncased", corpus_resume, corpus_position,
bert_tokenizer, bert_model)

# Resume and position word embeddings by BERT model (with context)
bert_resume = tf.convert_to_tensor(bert_embeddings[0].numpy())
bert_position = tf.convert_to_tensor(bert_embeddings[1].numpy())

print("BERT cosine tensor similarity:", cosine_similarity_tensorflow(bert_resume, bert_position).numpy())
print("BERT sqrt-cos tensor similarity:", sqrt_cos_sim_tensorflow(bert_resume, bert_position).numpy())
print("BERT improved sqrt-cos tensor similarity:", improved_sqrt_cos_sim_tensorflow(bert_resume,
bert_position).numpy())

# Build ELMo model (with context)
elmo_model = hub.load("https://tfhub.dev/google/elmo/3")
elmo_embeddings = context_word_embeddings_elmo(elmo_model, corpus_resume, corpus_position)

```

```

# Resume and position word embeddings by ELMo model (with context)
elmo_resume = elmo_embeddings[0]
elmo_position = elmo_embeddings[1]

print("ELMo cosine tensor similarity:", cosine_similarity_tensorflow(elmo_resume, elmo_position).numpy())
print("ELMo sqrt-cos tensor similarity:", sqrt_cos_sim_tensorflow(elmo_resume, elmo_position).numpy())
print("ELMo improved sqrt-cos tensor similarity:", improved_sqrt_cos_sim_tensorflow(elmo_resume,
elmo_position).numpy())
# Build Deep Learning models
model_w2v = build_model_ncontext(model_path="word2vec-google-news-300.model")
model_glove = build_model_ncontext(model_path="glove-wiki-gigaword-300.model")
model_fasttext = build_model_ncontext(model_path="fasttext-wiki-news-subwords-300.model")

bert_tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")
bert_model = BertModel.from_pretrained("bert-base-uncased", output_hidden_states=True)
elmo_model = hub.load("https://tfhub.dev/google/elmo/3")

# Define list of resumes and positions
resume_list = ["Business analyst Male.pdf", "Junior Frontend Developer Male.pdf", "Junior Full Stack Developer
Female.pdf",
               "Junior Software Developer Female.pdf", "Middle .NET Developer Male.pdf", "Middle Java Engineer
Male.pdf",
               "Product Marketing Manager Female.pdf", "QA Specialist Female.pdf", "Senior .NET Developer Male.pdf",
               "Senior Electrical Engineer Male.pdf", "Senior Software Engineer Male.pdf", "Trainee UX-UI Designer
Female.pdf"]
position_list = ["Business Analyst.txt", "Senior Software Developer.txt"]

# Initialize an empty list to store the results
results = []

# Iterate over all combinations of resumes and positions
for resume, position in product(resume_list, position_list):
    # Read and preprocess resume and position
    resume_text = read_resume(resume)
    position_text = read_position(position)
    resume_text = tokenize_description(resume_text)
    position_text = tokenize_description(position_text)
    resume_text = [preprocessing_text(sentence) for sentence in resume_text]
    position_text = [preprocessing_text(sentence) for sentence in position_text]
    position_tokens = [word_tokenize(sentence) for sentence in position_text]
    resume_tokens = [word_tokenize(sentence) for sentence in resume_text]
    flatten_position_tokens = sum(position_tokens, [])
    flatten_resume_tokens = sum(resume_tokens, [])
    corpus_resume = create_corpus(resume_text)
    corpus_position = create_corpus(position_text)

    # Calculate and store characteristics

```

```

characteristics = {}

characteristics['Resume'] = resume
characteristics['Position'] = position
characteristics['Jaccard similarity'] = jaccard_sim(flatten_resume_tokens, flatten_position_tokens)
characteristics['Dice similarity'] = dice_sim(flatten_resume_tokens, flatten_position_tokens)

bow_resume = bag_of_words(corpus_resume, corpus_position)[1]
bow_position = bag_of_words(corpus_resume, corpus_position)[2]
characteristics['Bag-of-words cosine similarity'] = cos_sim(bow_resume, bow_position)
characteristics['Bag-of-words sqrt-cos similarity'] = sqrt_cos_sim(normalize_vector(bow_resume),
normalize_vector(bow_position))
characteristics['Bag-of-words improved sqrt-cos similarity'] = improved_sqrt_cos_sim(bow_resume, bow_position)

tf_idf_resume = tf_idf(corpus_resume, corpus_position)[1]
tf_idf_position = tf_idf(corpus_resume, corpus_position)[2]
characteristics['Tf-idf cosine similarity'] = cos_sim(tf_idf_resume, tf_idf_position)
characteristics['Tf-idf sqrt-cos similarity'] = sqrt_cos_sim(normalize_vector(tf_idf_resume),
normalize_vector(tf_idf_position))
characteristics['Tf-idf improved sqrt-cos similarity'] = improved_sqrt_cos_sim(tf_idf_resume, tf_idf_position)

for model_name, model in [("Word2Vec", model_w2v), ("GloVe", model_glove), ("FastText", model_fasttext)]:
    embeddings_resume = ncontext_word_embeddings(resume_tokens, position_tokens, model, 300)[0]
    embeddings_position = ncontext_word_embeddings(resume_tokens, position_tokens, model, 300)[1]
    characteristics[f'{model_name} cosine similarity'] = cos_sim(embeddings_resume, embeddings_position)
    characteristics[f'{model_name} sqrt-cos similarity'] = sqrt_cos_sim(normalize_vector(embeddings_resume),
normalize_vector(embeddings_position))
    characteristics[f'{model_name} improved sqrt-cos similarity'] = improved_sqrt_cos_sim(embeddings_resume,
embeddings_position)

bert_embeddings = context_word_embeddings_bert("bert-base-uncased", corpus_resume, corpus_position,
bert_tokenizer, bert_model)
bert_resume = tf.convert_to_tensor(bert_embeddings[0].numpy())
bert_position = tf.convert_to_tensor(bert_embeddings[1].numpy())
characteristics['BERT cosine tensor similarity'] = cosine_similarity_tensorflow(bert_resume, bert_position).numpy()
characteristics['BERT sqrt-cos tensor similarity'] = sqrt_cos_sim_tensorflow(bert_resume, bert_position).numpy()
characteristics['BERT improved sqrt-cos tensor similarity'] = improved_sqrt_cos_sim_tensorflow(bert_resume,
bert_position).numpy()

elmo_embeddings = context_word_embeddings_elmo(elmo_model, corpus_resume, corpus_position)
elmo_resume = elmo_embeddings[0]
elmo_position = elmo_embeddings[1]
characteristics['ElMo cosine tensor similarity'] = cosine_similarity_tensorflow(elmo_resume, elmo_position).numpy()
characteristics['ElMo sqrt-cos tensor similarity'] = sqrt_cos_sim_tensorflow(elmo_resume, elmo_position).numpy()
characteristics['ElMo improved sqrt-cos tensor similarity'] = improved_sqrt_cos_sim_tensorflow(elmo_resume,
elmo_position).numpy()

```

```

results.append(characteristics)

df = pd.DataFrame(results)

# Define lists of expert rankings
ba_ideal_rank = pd.Series([1, 7, 3, 12, 10, 9, 5, 3, 4, 8, 3, 11], name="Business Analyst profession ranking").astype(dtype=np.float64)
se_ideal_rank = pd.Series([7, 8, 5, 10, 2, 7, 8, 3, 3, 9, 1, 12], name="Senior Software Engineer profession ranking").astype(dtype=np.float64)

# Calculate Kendall's Tau weighted coefficients for every metrics and model - Business Analyst Amazon
weightedtau_coefficients = {}
for coefficient in df[df['Position'] == 'Business Analyst.txt'].columns[2:]:
    weightedtau_coefficients[coefficient] = stats.weightedtau(df[df['Position'] == 'Business Analyst.txt'][coefficient].rank(ascending=False), ba_ideal_rank)[0]
sorted_weightedtau_coefficients = sorted(weightedtau_coefficients.items(), key=lambda x: (np.isnan(x[1]), -x[1] if not np.isnan(x[1]) else np.inf))
for coefficient, value in sorted_weightedtau_coefficients:
    if np.isnan(value):
        print(f"Weighted Tau coefficient for {coefficient} equals NaN")
    else:
        print(f"Weighted Tau coefficient for {coefficient} equals {value}")

# Calculate Kendall's Tau weighted coefficients for every metrics and model - Senior Software Engineer Google
weightedtau_coefficients = {}
for coefficient in df[df['Position'] == 'Senior Software Developer.txt'].columns[2:]:
    weightedtau_coefficients[coefficient] = stats.weightedtau(df[df['Position'] == 'Senior Software Developer.txt'][coefficient].rank(ascending=False), se_ideal_rank)[0]
sorted_weightedtau_coefficients = sorted(weightedtau_coefficients.items(), key=lambda x: (np.isnan(x[1]), -x[1] if not np.isnan(x[1]) else np.inf))
for coefficient, value in sorted_weightedtau_coefficients:
    if np.isnan(value):
        print(f"Weighted Tau coefficient for {coefficient} equals NaN")
    else:
        print(f"Weighted Tau coefficient for {coefficient} equals {value}")

```

Короткий звіт подібності



Ім'я користувача: Комп'ютерної математики та інформаційної безпеки...	ID перевірки: 1016327497
Дата перевірки: 06.06.2024 13:16:44 EEST	Тип перевірки: Doc vs Internet + Library
Дата звіту: 06.06.2024 13:17:31 EEST	ID користувача: 100005746

Назва документа: Диплом_Жук Д.В

Кількість сторінок: 84 Кількість слів: 14410 Кількість символів: 111906 Розмір файлу: 4.41 MB ID файлу: 1016126338

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

3%
Схожість

Найбільша схожість: 0.67% з джерелом з Бібліотеки (ID файлу: 1016102183)

1.54% Джерела з Інтернету 232 Сторінка 86

2.37% Джерела з Бібліотеки 304 Сторінка 87

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 2

Підозріле форматування 19 сторінок

Копії тез доповідей та статей

Моделювання та інформ. системи в економіці : зб. наук. праць / відп. ред. О. Є. Камінський. № 102 [Електронний ресурс]. – К.: КНЕУ, 2022 . – 184 с. – Режим доступу: https://kneu.edu.ua/userfiles/zb_mise/102/23-5758.pdf

УДК: 5.519.2

DOI 10.33111/mise.102.5

Галіцин В.К., д.е.н., професор,
професор кафедри комп'ютерної математики та інформаційної безпеки,
Київський національний економічний університет
імені Вадима Гетьмана

Жук Д.В., студент 3 курсу спеціальності «Системний аналіз»,
кафедра комп'ютерної математики та інформаційної безпеки,
Київський національний економічний університет
імені Вадима Гетьмана

Петриченко А.В., студентка 3 курсу спеціальності «Системний аналіз»,
кафедра комп'ютерної математики та інформаційної безпеки,
Київський національний економічний університет
імені Вадима Гетьмана

Galicin V.K., Doctor of Economic Sciences,
Professor, Professor Department of Computer Mathematics
and Information Security,
KNEU named after Vadym Hetman

Zhuk D.V., Student of the 3rd year, specialization «Systems analysis»,
Department of Computer Mathematics and Information Security,
KNEU named after Vadym Hetman

Petrychenko A.V., Student of the 3rd year, specialization «Systems
analysis»,
Department of Computer Mathematics and Information Security,
KNEU named after Vadym Hetman

ВИПАДКОВІ ПРОЦЕСИ В МЕТЕОРОЛОГІЇ

RANDOM PROCESSES IN METEOROLOGY

Анотація. Зміна клімату — це спостережувані та прогнозовані довгострокові зміни середнього клімату, а також мінливість клімату, викликана діяльністю людини, включаючи такі аномалії, як посухи, сильні шторми та повені. Клімат нашої планети постійно змінювався протягом усієї геологічної історії Землі, і ці зміни супроводжувалися значними коливаннями середніх глобальних температур. Одним із головних чинників зміни глобальної середньої температури є спалювання викопного палива, вугілля, газу та нафти, що збільшило концентрацію парникових газів, таких як вуглекислий газ, у нашій атмосфері. Ця робота є першою чергою спрямованою на вивчення проблем прогнозування та моніторингу такого погодного явища, як локальна зміна температури — на основі моделей часових рядів, представлених у цій роботі, можна зробити необхідні прогнози також для вимірювання рівня снігового покриву, рівень опадів. Використання моделей прогнозування температури (таких як метод Холта-Вінтера або бібліотека Пророк) має велике практичне значення, починаючи від прогнозування температури на наступний рік

для себе, вразливості культур тощо. У даній роботі було проведено довгостроковий аналіз середньодобової температури у Київській області мовою програмування Python та її бібліотеками для аналізу даних Pandas, Numpy, Scikit-Learn тощо за допомогою моделей експоненційного залякування та бібліотеки Prophet. За допомогою цих моделей можна обробляти та прогнозувати різні типи даних, такі як: температура, рівень опадів, глибина снігового покриву. Крім того, прогнози можуть бути використані для різноманітних цілей людської діяльності. Матеріали статті мають науково-методичний характер.

Ключові слова: випадковий процес, часові ряди; метеорологія; зміна клімату; модель прогнозування; наванчання під наглядом; експоненціальне залякування; метод Холта-Вінтера; бібліотека Пророк.

Abstract. Climate change is the observed and projected long-term changes in average climate, as well as climate variability, caused by human activity, including such anomalies as droughts, severe storms and floods. The climate of our planet has been constantly changing throughout the entire geological history of the Earth, and these changes were accompanied by significant fluctuations in average global temperatures. One of the main drivers of global average temperature change is the burning of fossil fuels, coal, gas and oil, which has increased the concentration of greenhouse gases such as carbon dioxide in our atmosphere. This work is primarily aimed at studying the problems of predicting such a weather phenomenon as a local change in temperature — based on the time series models presented in this work, it is possible to make the necessary predictions also for measuring the level of snow cover, precipitation level. The use of temperature prediction models (such as Holt-Winter's method or Prophet library) is of great practical importance, starting from predicting the temperature for the next year for sowing, crop yields, etc. In this work, a long-term analysis of the average daily temperature in the Kyiv region was carried out using the Python programming language and its data analysis libraries Pandas, Numpy, Scikit-Learn, etc. using exponential smoothing models and the Prophet library. These models can be used to process and forecast different types of data, such as temperature, precipitation, snow depth. In addition, forecasts can be used for various purposes of human activity. The materials of the article have a scientific and methodological nature.

Keywords: Random process; time series; meteorology; climate change; prediction model; supervised learning; exponential smoothing; Holt-Winter's method; Prophet.

Постановка проблеми. Зміна клімату та наслідки цього процесу для планети — один з важливих напрямів сучасних метеорологічних досліджень. Метеорологія наука про будову і властивості земної атмосфери та фізичних процесів, що в ній відбуваються. Перша офіційна згадка про цю науку належить Аристотелю, який приблизно в 340 р. до н. е. написав книгу під назвою «Метеорологія», що містила все, що було відомо на той час про погоду та клімат. Назва походила від грецького слова «Μετεωρο», що означало «щось високо» і стосувалося усього, що спостерігалося в атмосфері [2].

Математичні методи та моделі використовують для розв'язання побудови прогнозу погоди і загальної циркуляції атмосферних процесів, чисельних алгоритмів для їх вирішення. Рівняння гідротермодинаміки атмосферних процесів є досить скла-

Системи та засоби штучного інтелекту: тези доповідей Міжнародної наукової молодіжної школи. “Наука і освіта” [Електроний ресурс]. – К.: ІПШ, 2023 . – 22 с. – Режим доступу: <https://www.ipai.net.ua/docs/AIIS-2023-school.pdf>

«SYSTEMS AND MEASURES OF ARTIFICIAL INTELLIGENCE» AIIS'2023
International scientific youth school

UDC 004.8

THE ETHICS OF ARTIFICIAL INTELLIGENCE: CHALLENGES AND FRAMEWORKS

D. Zhuk

Kyiv National Economic University,
Kyiv, Ukraine

The ethics of artificial intelligence delves into the ethical considerations and principles that guide the development, deployment, and impact of artificial intelligence systems on individuals, society and the global landscape. I explored the intersection between responsible artificial intelligence and the artificial intelligence ethics, unraveling the intricacies of fostering innovation while upholding ethical standards. A framework is presented, outlining key ideas to guide the creation of responsible artificial intelligence, contributing to the ongoing discourse on the responsible use of artificial intelligence.

Introduction

Over the past years, artificial intelligence has gained enormous popularity. Developments in this area can revolutionize and change our everyday life. Despite the hype surrounding artificial intelligence, there are also arising concerns about the ethical implications of its developments. The discussion has been fueled by the growing popularity of generative artificial intelligence and its deriving services such as ChatGPT or Stable Diffusion, capable of producing believable texts and images. Responsible artificial intelligence practices are vital to address potential misuse and malicious threats as they have increased 26 times since 2012 [1]. These systems can craft convincing content, raising ethical concerns about deception, manipulation and the spread of misinformation. Oversight, ethical deployment and regulatory frameworks are crucial to

Цифрова екосистема сучасного університету: адаптація й розвиток за умов турбулентності освітнього середовища: Зб. матеріалів науково-методичної конференції [Електронний ресурс]. – К.: КНЕУ, 2024 . – 359 с. – Режим доступу: https://drive.google.com/file/d/1_18WXLInpiz5jCgl3AFO2d2b2YfOxUiG/view?usp=drive_link

УДК 378:316.422
Ц 75

Оргкомітет конференції

Лук'яненко Д.Г. д.с.н., проф., в.о. ректора КНЕУ ім. В. Гетьмана;
Колод А.М. д.с.н., проф., проректор з науково-педагогічної роботи КНЕУ ім. В. Гетьмана;

Сісчченко М.В. к.с.н., доцент, проректор з науково-педагогічної роботи та цифрової трансформації КНЕУ ім. В. Гетьмана;

Антонюк Л.І. д.с.н., проф., проректор з наукової роботи, директор Інституту вищої освіти КНЕУ ім. В. Гетьмана;

Фролова Т.О. д.с.н., професор, начальник навчально-методичного відділу забезпечення якості освіти та акредитації КНЕУ ім. В. Гетьмана;

Горбова Ю.С. к.соц.н., директор Інституту соціологічних досліджень КНЕУ ім. В. Гетьмана;

Котенок Д.М. к.с.н., доцент, заступник начальника навчально-методичного відділу забезпечення якості освіти та акредитації КНЕУ ім. В. Гетьмана;

Красюк Ю.М., к.пед.н., доцент, директор Інституту дистанційної освіти КНЕУ ім. В. Гетьмана;

Суриунюк Г.М., начальник відділу навчально-методичної роботи із супроводження дистанційного навчання Інституту дистанційної освіти КНЕУ ім. В. Гетьмана.

*Матеріали друкуються в авторській редакції.
Відповідальність за науковий рівень публікацій, обґрунтованість висновків,
достовірність результатів, наявність плагіату несуть автори.*

Цифрова екосистема сучасного університету: адаптація й розвиток за умов турбулентності освітнього середовища: Зб. матеріалів науково-методичної конференції [Електронний ресурс]. – К.: КНЕУ, 2024 . – 359 с.
ISBN 978-966-926-478-7

Представлено уявлення стейкхолдерів освітнього процесу про особливості адаптації та розвитку цифрової екосистеми університету за умов турбулентного освітнього середовища у світоінтеграційному та цифровому контекстах; репрезентовано країні практики навчально-дослідницької взаємодії учасників освітнього процесу за надзвичайних умов роботи та охарактеризовано вектори цифрової гармонізації підготовки здобувачів вищої освіти різних форм навчання та розвитку цифрової культури якості освітньої програми підготовки здобувачів, а також викладацької майстерності.

УДК 378:316.422

*Розповсюджувати та тиражувати
без офіційного дозволу КНЕУ заборонено*

ISBN 978-966-926-478-7

© КНЕУ, 2024

*Dmytro Zhuk Student 4-th year of specialization "System analysis"
Irada Dzhalladova Doctor of Physical and Mathematical Sciences, Professor
Department of Systems Analysis and Cybersecurity*

**USING RESPONSIBLE ARTIFICIAL INTELLIGENCE IN
EDUCATION: DIGITAL TRANSFORMATION OF ACQUISITION OF
PROFESSIONAL SKILLS**

In the fast-evolving landscape of education, the integration of artificial intelligence has emerged as a transformative force, promising to revolutionize the acquisition of professional skills. Formal and non-formal education are of great importance now, recognizing one of the main roles of artificial intelligence in shaping skills development.

Nowadays, there is a huge shortage of digital skills worldwide – for example, only 55 percent of people aged 16 to 74 in the European Union have at least basic overall digital skills. [1] However, according to the Society for Human Resource Management (SHRM), about 90 percent of job seekers said that general digital skills are vital, regardless of industry, occupation, or demographics. [2] This highlights the necessity for individuals to possess digital competencies in order to be competitive at local and global scale through continuous formal and non-formal training.

Adoption of artificial intelligence is simply crucial and financially prospective as the projected growth for the artificial intelligence market in education is remarkable, with estimates suggesting it will reach \$80 billion by the year 2032. [3] In response to the growing demand for digital skills, educational institutions and training programs are increasingly adopting artificial intelligence methods and technologies into their curricula. For example, artificial intelligence becomes more present with global trends in education [4], incorporating into subjects like STEM sciences, social sciences, health education et cetera.

In general, digital skills development falls into 3 completely different categories such as cognitive skills, technical skills, and interpersonal skills. Developing a well-rounded skill set that encompasses all of these categories is essential for success in today's interconnected world for building a personal brand.