НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

На правах рукопису
УДК 004.852

До захисту допущено
В.о. зав. кафедри ШІ
_____О.І. Чумаченко
«___»_____2022 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 122 «Комп'ютерні науки»

на тему: «Система класифікації вакансій відповідно до класифікатора професій методами штучного інтелекту»

Виконала:
студентка ІІ курсу, групи КІ-11мп
Цимбал Юлія Олександрівна _____

Керівник:
доцент кафедри ММСА,
д.т.н., доц. Недашківська Н. І. _____

Рецензент:
доцент кафедри системного проектування
КПІ ім. Ігоря Сікорського, к.т.н., Безносик О. Ю. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ
2022

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

Рівень вищої освіти – другий (магістерський)

Спеціальність – 122 «Комп'ютерні науки »

ЗАТВЕРДЖУЮ
В. о. зав. кафедри
_____ О.І. Чумаченко
«___» _____ 2022 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Цимбал Юлії Олександрівні

1. Тема дисертації: «Система класифікації вакансій відповідно до класифікатора професій методами штучного інтелекту», науковий керівник роботи Недашківська Надія Іванівна, доцент кафедри ММСА, д.т.н., доц. затверджено наказом по університету від «03» листопада 2022 р. № 4046-с.
2. Термін подання студентом дисертації 15.12.2022
3. Об'єкт дослідження: онлайн вакансії, які представлені на українському ринку праці.
4. Предмет дослідження: застосування методів штучного інтелекту для багатокласової класифікації текстових даних.
5. Перелік завдань, які потрібно зробити:
    1) здійснити огляд технічної літератури за темою роботи;
    2) дослідити актуальність обраної теми;
    3) ознайомитись із існуючими методами та моделями класифікації неструктурованих даних;

4) здійснити порівняльний аналіз наявних методів, виявити їх переваги та недоліки;

5) розробити та реалізувати систему, що використовує апарат нейрониих мереж, та вирішує задачу багатокласової класифікації вакансій зібраних у мережі;

6) провести експеримент, що засвідчує працеспроможність запропонованої моделі, виконати аналіз результатів;

7) провести аналіз ринкових можливотсей запуску стартап проекту;

8) розробити концептуальні висновки;

9) підготувати ілюстративний матеріал;

10) оформити пояснювальну записку.

6. Перелік ілюстративного матеріалу.

7. Дата видачі завдання: 1 вересня 2022 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів роботи	Примітка
1.	Вивчення літератури за темою роботи.	01.09.2022 – 14.09.2022	Виконано
2.	Підготовка першого розділу.	14.09.2022 – 16.09.2022	Виконано
3.	Підготовка другого розділу.	16.09.2022 – 23.09.2022	Виконано
4.	Розробка програмного продукту.	01.10.2022 – 15.11.2022	Виконано
5.	Підготовка третього розділу	16.11.2022 – 20.11.2022	Виконано
6.	Підготовка четвертого розділу	21.11.2022 – 25.11.2020	
7.	Підготовка частини стартап-проєкту	26.11.2022 – 30.11.2022	Виконано
8.	Концептуальні висновки. Перспективи розвитку отриманих рішень	01.12.2022 – 03.12.2022	Виконано
9.	Оформлення пояснювальної записки	04.12.2022 – 05.12.2022	Виконано

Студент Юлія ЦИМБАЛ

Керівник Надія НЕДАШКІВСЬКА

РЕФЕРАТ

Магістерська дисертація: 110 с., 24 табл., 17 рис., 27 джерел, 1 додаток.

РИНОК ПРАЦІ, ОНЛАЙН СЕГМЕНТ РИНКУ ПРАЦІ, ВЕЛИКІ ДАННІ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ, ОБРОБКА ПРИРОДНОЇ МОВИ, КЛАСИФІКАЦІЯ.

Об’єкт дослідження – онлайн вакансії, які представлені на українському ринку праці.

Предмет дослідження – застосування методів штучного інтелекту та обробки природної мови для класифікації вакансій.

Мета роботи – розробка нового, прикладного інструментарію збирання, обробки та класифікації різнорідної інформації щодо класифікатора професій задля структуризації та кращого моніторингу ситуації на ринку праці України.

В роботі проведено огляд ринку праці в Україні за останні роки, розглянуто проблематику питання класифікації професії та його роль у подальшому розширенні завдання отримання актуальної інформації про ринок праці. Розглянуті сучасні методи штучного інтелекту для обробки природної мови та тексту. Розроблено програмний продукт для збору, обробки та класифікації специфічних даних.

Основні наукові результати та їх новизна. Зібрано, очищено та систематизовано інформацію онлайн джерел щодо пропозиції робочих місць на українському ринку праці за рахунок методичної адаптації, та практичного освоєння технології Big Data Mining. Реалізовані методи збору за рахунок інструментів скрейпінгу, очищення даних та класифікації тематично неструктурованих текстів опису вакансій; методичний підхід до відбору релевантних джерел онлайн ринку праці, що дозволяє знайти їх оптимальний мінімум для надійної репрезентації повного онлайн ринку праці в частині пропозиції робочих місць; вперше в Україні розроблена комплексна система класифікації пропозицій робочих місць методами штучного інтелекту.

ABSTRACT

Master’s thesis: 109 p., 24 tab., 17 fig., 27 references, 1 appendix

LABOR MARKET, ONLINE SEGMENT OF THE LABOR MARKET, BIG DATA, DATA MINING, NATURAL LANGUAGE PROCESSING, CLASSIFICATION.

The object of the research is online vacancies that are presented on the Ukrainian labor market.

The subject of the research is the application of artificial intelligence methods and natural language processing for job classification.

The purpose of the master's thesis is to develop a new, applied toolkit for collecting, processing, and classifying heterogeneous information regarding the profession classifier for structuring and better monitoring of the situation on the labor market of Ukraine.

The paper reviewed the labor market in Ukraine in recent years, discussed the issue of the classification of the profession and its role in further expanding the task of obtaining up-to-date information about the labor market. Modern methods of artificial intelligence for natural language and text processing are considered. A software product has been developed for the collection, processing, and classification of specific data.

Main scientific results and their novelty. Collected, cleaned, and systematized information from online sources regarding the offer of jobs on the Ukrainian labor market due to methodical adaptation and practical mastering of Big Data Mining technology. Implemented methods of collecting by scraping tools, data cleaning and classification of thematically unstructured job description texts; a methodical approach to the selection of relevant sources of the online labor market; for the first time in Ukraine, a comprehensive system for classifying job offers using artificial intelligence methods was developed.

ЗМІСТ

ВСТУП

Обробка природної мови за останні роки невпинно нарощує масштаби застосування. Зараз важко уявити сучасну систему без використання NLP методів, за винятком дуже консервативних галузей. У більшості технологічних рішень давно реалізовано розпізнавання та обробка «людських» мов.

Обробка природної мови (NLP) – це технологія машинного навчання, яка дає комп'ютерам можливість інтерпретувати, маніпулювати та розуміти людську мову. Сьогодні організації мають великі обсяги голосових та текстових даних із різних каналів зв'язку, таких як електронні листи, текстові повідомлення, стрічки новин соціальних мереж, відео, аудіо та багато іншого. Вони використовують програмне забезпечення NLP для автоматичної обробки цих даних, аналізу намірів чи настроїв у повідомленні та реагування на людське спілкування в режимі реального часу.

Обробка природної мови має вирішальне значення для ефективного аналізу текстових та мовних даних. Таким чином, можна долати відмінності в діалектах, сленгу та граматичних порушеннях, типових для повсякденних розмов. Компанії використовують цей метод для кількох автоматизованих завдань, таких як:

- обробка, аналіз та архівування великих документів
- аналіз відгуків клієнтів або записів кол-центру
- запуск чат-ботів для автоматизованого обслуговування клієнтів
- відповіді на запитання «хто, що, коли та де»
- класифікація та вилучення тексту

Наразі сфера застосування машинного навчання постійно розширюється. Повсюдна інформатизація призводить до накопичення величезних об’ємів даних в науці, виробництві, бізнесі, транспорті, медицині. Задачі прогнозування, управління та прийняття рішень, які виникають при

цьому дуже часто зводяться до навчання по прецедентам. Раніше, коли таких даних не було, ці задачі або ж вирішувалися іншими методами, або ж взагалі залишалися нерозв’язаними.

Основним напрямком цієї роботи є застосування різних методів штучного інтелекту та обробки природної мови як потужного інструменту інтелектуального аналізу даних у демографічній сфері населення, а саме ринку праці. Розвиток економічних відносин стрімко зростає, відповідно і робочі місця масштабуються, змінюють вимоги до працівників та шукачів роботи, створюються нові професії та трансформуються існуючі. Наразі в демографічних дослідженнях в Україні науковці тільки починають використовувати різноманітні інструменті машинного навчання та штучного інтелекту, тому дана тема магістерської дисертації є дуже актуальною. Такий автоматизований метод аналізу ринку праці дасть змогу більш детально, актуально та прицільно досліджувати професійну сферу. Очищена та належним чином систематизована інформація онлайн джерел щодо вакансій статистично значуще доповнює наявну інформацію про офіційні вакансії Державного центру зайнятості, їх вимог щодо знань і навичок робочої сили, типу трудових контрактів, досвіду, трудових обов’язків та особистих якостей. Головна мета та ціль даного продукту це можливість оцінити та аналізувати ринок праці в Україні у реальному часі та базуючись на реальних даних. Такий аналіз наддасть змогу ширше та глибше зрозуміти ситуацію працевлаштування.

Магістерська дисертація містить чотири розділи.

Перший розділ магістерської дисертації присвячений огляду предметної області та детальнішого опису проблематики питання.

Другий розділ присвячено огляду методів технічної реалізації класифікації. Детальний розбір наявних методів штучного інтелекту та обробки мови та вибір найбільш ефективніших для даної задачі.

У третьому розділі викладено практичну роботу з даними, їх збір, аналіз, підготовка та опис інформації, що міститься зазвичай на порталах пошуку роботи.

Четвертий розділ описує роботу алгоритму класифікації та результати дослідження.

П’ятий розділ присвячено розробці власного стартап-проєкту.

РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1 Стан та проблеми інформаційного забезпечення функціонування українського ринку праці

Інформаційне забезпечення дослідників, управлінців та учасників українського ринку праці сьогодні задовольняється за рахунок:

– діяльності регулярних державних інституцій, а саме: обстежень Державної служби статистики в сфері статистики праці; обов'язкової податкової звітності підприємств, підприємців та організацій, що збирається податковою службою України та пенсійним фондом України; оперативними даними та звітами Державної служби зайнятості;

– діяльності регулярних недержавних інституцій: спеціалізованих ЗМІ, де висвітлюються окремі аспекти функціонування ринку праці та публікуються інформація про вакансії та конкурси для заміщення вакансій; даними інтернет посередників щодо вакансій та резюме пошукувачів роботи (далі онлайн ринок праці), а також канали й чат-боти соціальних мереж та месенджерів;

– нерегулярних агентів: міжнародних дослідників та організацій; окремих соціологічних опитувань.

Основними державними інституціями в сфері інформаційного забезпечення відносин щодо ринку праці є Державна служба статистики України (далі ДССУ) та Державний центр зайнятості (ДЦЗ).

ДССУ здійснює свою діяльність на основі закону України «Про державну статистику» (Відомості Верховної Ради України (ВВР), 1992, № 43, ст.608) [1]. Суттєвою перевагою ДССУ є те, що вона має перевірений та регулярний доступ до:

– первинних даних щодо респондентів, які підлягають статистичним спостереженням: населенню та підприємствам;

– адміністративних даних інших державних органів, органів місцевого самоврядування;

– статистичну інформацію міжнародних організацій та статистичних служб інших країн щодо ринку праці;

– знеособлені дані реєстрів територіальних громад та інших інформаційних систем, що містять інформацію щодо населення України та його демографічних характеристик.

Крім первинних даних ДССУ має змогу статистично коректно здійснювати оцінки та розрахунки агрегованих та репрезентативних показників на основі первинних даних. Оброблені дані та синтезована на них інформація ДССУ характеризується високим ступенем достовірності, надійності та визначеної точності. Вона доступна перевірці, а методологія їх підготовки відповідає сучасним науковим досягненням, міжнародним рекомендаціям та досвіду статистичної практики з урахуванням національно-історичних особливостей країни. Основні положення статистичної методології підлягають опублікуванню тому з ними можна ознайомитися та краще зрозуміти зміст, призначення, логіку та таксономію представленої інформації.

У сфері ринку праці ДССУ збирає дані підприємств та організацій щодо середньооблікової кількості штатних працівників, обсягів прийнятих та звільнених, розмір використаного та невикористаного робочого часу, обсяги вимушеної неповної зайнятості, середню заробітну плату, фонд оплати праці та його структуру, розмір заборгованості працівникам заробітної плати. Всі ці показники можуть бути кореспондованими між собою та представлені за видами економічної діяльності, регіонами, організаційно-правовими формами господарювання, формами зайнятості, статтю працівників.

Для представлення у збірнику первинна статистична інформація згрупована згідно з такими статистичними класифікаторами:

– видів економічної діяльності (КВЕД) ДК 009:2010;

– організаційно-правових форм господарювання (КОПФГ) ДК 002:2004;

– кодифікатор адміністративно-територіальних одиниць та територіальних громад (КАТОТТГ).

Отже, інформація, яку збирає ДССУ по підприємствах та організаціях щодо питань праці представляє чималий масив перевіреної та надійної інформації, однак вона не містить відомостей щодо професійно-кваліфікаційних особливостей штатних працівників, а також жодним чином не торкається кількості незайнятих робочих місць на підприємствах та пропонованих на ринку праці вакансій.

Частково нестачу даних щодо професійно-кваліфікаційних особливостей штатних працівників ДССУ заповнює в результаті державне статистичне спостереження підприємств: «Рівень заробітної плати працівників за статтю, віком, освітою та професійними групами» [2].

Недоліком згаданого обстеження є те, що воно проводиться один раз на 4 роки, а його дані не можуть бути безпосередньо інтегрованими до регулярних даних ДССУ щодо статистики праці. Крім того, як і дані ДССУ щодо статистики праці, результати окремого статистичного спостереження заробітної плати не містить даних про вакансії. Слід також враховувати, що згідно особливостей проведення статистичних спостережень підприємств, до кола обстежуваних робочих місць не потрапляють ті, які функціонують без оформлення та у сфері неформальної зайнятості, а також робочі місця в секторі зайнятості фізичних осіб підприємців без створення юридичної особи. В результаті, згідно оцінок визнаних експертів, статистика праці ДССУ охоплює від 60-75% повного ринку праці і не може таким чином претендувати на повне представлення.

Ще одним потужним джерелом державних статистичних даних є матеріали обстеження робочої сили України. В даному випадку одиницями обстеження виступають не підприємств, а населення України. Обстеження є максимально репрезентативним, оскільки представляє розподіл усього

населення у віці старше 15 років стосовно його стану щодо ринку праці: зайняте, безробітне, економічно неактивне (не входить до складу робочої сили). Це найтриваліше та найрегулярніше статистичне спостереження в Україні. Воно здійснюється з 1995 року, а з 2004 проводиться щомісячно. Згадане обстеження є унікальним в Україні на масштабами охоплення населення за видами економічної діяльності та регіонами. Так упродовж 2021р. було опитано 151,7 тис. респондентів віком 15 років і старше, що складає 0,48% постійного населення України зазначеного віку. Кількість проведених інтерв'ю становила 265,6 тис [3].

Обстеження робочої сили дає змогу встановити чисельність зайнятого та безробітного населення на повному ринку праці України. Щодо них надаються відомості про вік, стать, рівень здобутої освіти, рівень професійної кваліфікації поточного заняття. Щодо зайнятого населення надається інформація про вид економічної діяльності, регіональну локацію, особливості зайнятості щодо рівня оформленості трудових відносин, тривалості робочого часу, відповідності кваліфікаційних вимог робочого місця та набутої кваліфікації в процесі професійної підготовки та освіти. Отже, відповідний блок даних обстеження робочої сили надає відомості про сферу зайнятості як і дані спостереження підприємств щодо статистики праці. Вони певним чином можуть вивчатися спільно, однак не можуть бути не тільки інтегрованими безпосередньо, а й навіть забезпечені проведенням надійної консистенції чи кореляції. Крім того, матеріали обстеження зайнятої робочої сили не дають жодних відомостей про вакансії, що пропонуються на ринку праці.

Унікальна інформація надається обстеженням робочої сили України щодо безробітних. Гіпотетично це ті самі особи, які пропонують свої резюме на сайтах онлайн посередників та соціальних мережах. Однак, згідно даних обстеження робочої сили у 2021 році серед безробітних таких осіб було лише 3,3%, тоді як 18,6% лише вивчали відповідні оголошення в інтернеті. Скоріш за все такий низький відсоток активних учасників онлайн платформ з пошуку роботи пов’язаний з тим, що значна чисельність населення розміщує резюме

ще протягом періоду зайнятості. Згідно методології обстеження робочої сили подібне питання не задається зайнятим особам, що є недоліком з точки зору завдань встановлення ступеню репрезентативності тих резюме, що представлені на сайтах інтернет посередників.

Отже, аналіз інформації, що оприлюднює ДДСУ, методології її збору та обробки, дозволяє зафіксувати відсутність інформації про вакансії, що пропонуються на повному ринку праці України, обмеженням можливостей встановити професійно-кваліфікаційну структуру штатних працівників у прив'язці до підприємств, неможливості проєкції даних обстеження робочої сили на дані спостереження підприємств для заповнення цієї прогалини. Разом з тим, перевагою ДССУ є те, що вона користується міжнародно визнаними класифікаціями щодо видів економічної діяльності (гармонізовано із Statististical classification of economic activities in the Europeenne Communaute / Nomenclature statistique des Activites economiques dans la Communaute Europeenne (NACE Rev.2).), професій (гармонізовано із Міжнародною стандартною класифікацією занять (ISCO-88: International Standard Classification of Occupations /ILO, Geneva)), інституційних секторів економіки України (гармонізовано із Системою національних рахунків (СНР 2008); Європейською системою національних та регіональних рахунків 2010 року (ESA 2010). Це сприяє розвитку міжнародного співробітництва в сфері регулювання ринку праці та трудової міграції, а також сприяє можливості гармонізації інституційних та організаційних структур ринку праці в процесі інтеграції українського до країн ЄС.

Єдиним офіційним джерелом про вакансії на українському ринку праці є Державний центр зайнятості (далі ДЦЗ), який входить до складу державної служби зайнятості – централізованої системи державних установ, діяльність якої спрямовується та координується Міністерством соціальної політики України.

Державний центр зайнятості виконує дві основні функції: він є єдиним державним посередником з працевлаштування, і одночасно – виконавчим

органом Фонду загальнообов'язкового державного соціального страхування України, який здійснює державну реєстрацію незайнятих осіб як безробітних, сплачує їм страхові виплати та надає повний цикл послуг з пошуку підходящої роботи, профорієнтації, професійного підготовки тощо.

На сьогодні державна служба зайнятості крім інших функцій є активним посередником на ринку праці між роботодавцями і шукачами роботи, вона на безоплатній основі надає послуги із пошуку підходящої роботи не тільки особам, що офіційно зареєструвалися як безробітні, а будь-кому, хто шукає роботу. Значний обсяг робіт виконує ДЦЗ і в сфері підбору персоналу згідно вакансій, поданих роботодавцями.

У ДЦЗ створена уніфікована оперативна база вакансій, шукачів роботи та можливостей проходження професійного навчання по всій країні. Доступ до оперативної бази вакансій забезпечено створенням спеціального онлайн-порталу вакансій та резюме робочої сили [4].

Державний центр зайнятості є досить популярним ресурсом посередництва у пошуку роботи на ринку праці загалом, а не тільки для тих осіб, які зареєструвалися як безробітні. Так згідно державного статистичного обстеження робочої сили у 2021 р. найчастішим методом пошуку роботи для українських безробітних було звернення по допомогу до державного центру зайнятості (37,9%) (таблиця 1.1), в той час як пошук за допомогою інтернет-ресурсів був у рейтингу третім (21,9%) після пошуку роботи за допомогою персональних зв’язків (31,0%) (до того ж певна частина пошуку через інтернет-ресурси також була за допомогою онлайн-порталу державного центру зайнятості).

Таблиця 1.1 Рівень безробіття та методи знайти роботу в 2021 році, %

	Усе населення	Жінки	Чоловіки	Міська місцевість	Сільська місцевість
Безробітне населення у віці 15–70 років, усього, тис. осіб	1 711,6	841,6	870,0	1 132,2	579,4
з них особи, які шукали роботу або намагались організувати власну справу, тис. осіб	1 667,7	824,6	843,1	1 104,4	563,3
у тому числі за способами пошуку роботи, у відсотках					
вивчали оголошення в пресі	3,6	4,0	3,2	4,1	2,4
вивчали оголошення в Інтернеті	18,6	16,1	21,0	23,3	9,5
шляхом розміщення або оновлення анкет у професійних чи соціальних мережах в Інтернеті	3,3	2,8	3,7	4,2	1,4
через особисті зв’язки (друзів, родичів чи інших посередників)	31,0	25,3	36,6	30,9	31,2
зверталися до роботодавців, шукали роботу на ринках чи інших громадських місцях	4,7	5,3	4,0	3,6	6,9
зверталися до державної служби зайнятості	37,9	45,8	30,1	32,7	48,0
зверталися до приватної фірми з працевлаштування	0,6	0,3	1,0	0,7	0,5
інше	0,3	0,4	0,4	0,5	0,1

Джерело: Робоча сила України 2021 / Статистичний збірник. Держстат України. – Київ, 2022. – 216 с.

Крім розгалуженої мережі відділень, де в режимі офлайн надаються послуги незайнятому населенню, ДЦЗ підтримує роботу інтернет-порталу вакансій. Сильний бік інформації цього порталу в тому, що вакансії там представлені проходять процедуру верифікації, як щодо реальності так і відповідності вимогам законам України «Про зайнятість населення» [5] (стаття 11 і Розділ VI) та «Про рекламу» (стаття 24) [6].

Портал ДЦЗ також підтримує власні норми, яких повинні дотримуватися роботодавці коли користуються його послугами:

– пропонуватися має тільки трудовий договір;

– відповідність професій у вакансіях до професійних назв робіт КП ДК 003:2010;

– максимальна перевірка даних про роботодавця, що обмежує можливості шахрайства;

– максимально широке регіональне охоплення;

– у всіх вакансіях вказаний рівень заробітних плат, причому вона має дорівнювати або перевищувати національну мінімальну заробітну плату;

– оголошення про вакансію має відповідати усім правилам недискримінації, зокрема за статтю, віком, віросповіданням.

Для пошукувачів роботи портал ДЦЗ дозволяє подати документи в службу зайнятості, отримати консультації, подати заявку на пошук робочого місця, оновити інформацію у своєму профілі CV. Це дозволяє роботодавцям зв'язуватися з потенційними кандидатами на роботу, вивчаючи їхні профілі CV, публікувати пропозиції та подавати заявки на програми, що фінансуються державою. Служби ДЦЗ перевіряють достовірність інформації, що міститься в заявах, а оголошення про роботу, які пропонуються до публікації, повинні бути заповнені у формі, що містить обов'язкові поля.

Попри законодавчо визначене право співпрацювати із приватними агенціями з працевлаштування та інтернет-ресурсами державний центр зайнятості цього не робить, оскільки вони не відповідають усім обов'язковим елементам, необхідним для публікації на публічному порталі.

Недоліком інформації, що надається про вакансії ДЦЗ є те, що вони не охоплюють неформальні робочі місця як на підприємствах, які зареєстрували свою діяльність так і підприємств неформального сектору. Крім того, на думку більшості експертів ринку праці, сегмент підприємств, які пропонують вакансії через ДЦЗ охоплює переважно робочі місця із невисоким рівнем привабливості. Найбільш привабливі робочі місця традиційно заповнюються через особисті зв'язки, пошук працівників власними силами, або за допомогою кваліфікованих рекрутінгових агенцій. Судячи із загальних тенденцій поширення інтернет технологій та цифровізації, використання онлайн

ресурсів для пропозиції вакансій і пошуку резюме потенційних працівників є домінуючим порівняно із іншими способами широкої публікації інформації про свої потреби у робочій силі.

Можна зробити перший висновок: найвагомішим джерелом інформації про фактичний обсяг вакансій на українському ринку праці виступає онлайн-середовище, представлене як недержавними посередниками так і ДЦЗ. Відповідно до цього без використання таких даних інформаційна картина українського ринку праці є неповною та викривленою.

1.2. Актуальність систематизації та аналізу вакансій українського онлайн сегменту ринку праці

Представлення вакансій українського ринку праці в онлайн середовищі дозволяє громадянам та підприємствам України оперативно та зручно вирішувати завдання пошуку один одного. З іншим завданням: накопичення, зберігання та універсалізацією даних про вакансії та резюме пошукувачів роботи, а також правильним позиціонуванням онлайн пропозицій по відношенню до повного ринку праці України онлайн-середовище не справляється. Причин тому декілька. Перша – онлайн пропозиції зберігається порівняно короткий час, і накопичення відповідних даних не підтримується більшістю онлайн-посередників. Друга – як підприємства так і населення намагаються подавати онлайн пропозиції на максимально більшій кількості платформ, що не дозволяє елементарними підрахунками оцінити загальні та часткові обсяги вакансій. Третя – різні онлайн посередники по різному форматують дані, використовують різні алгоритми та синтаксис організації роботи з даними, різну їх рубрикацію. Четверта – жоден із досліджених онлайн посередників не використовує для категоризації власних даних та опису офіційні стандарти та класифікації.

По ідеї нормативна база подання і використання інформації у сфері посередництва у працевлаштуванні має спиратися на використання Класифікатора професій: КП ДК 003:2010 [7] та Класифікатора видів економічної діяльності: КВЕД-2010 [8]. Це обумовлюється тим, що відповідно до положень чинного законодавства, національні класифікатори прирівнюються до державних стандартів України [9]. Однак нормативні вимоги щодо сфер їх використання досить суперечливі. Те, що певний класифікатор є національним стандартом, згідно закону України «Про стандартизацію» [10], не означає його обов'язкового використання. Він застосовуються на добровільній основі, крім випадків, якщо їх обов'язковість встановлена нормативно-правовими актами. Разом з тим, прямої вимоги про використання КП та КВЕД у публічному представлені вакансій не існує, що дозволяє ігнорувати їх використання не тільки інтернет-посередникам, а й іншим публічним ресурсам, які розміщують інформацію про вакансії та пошук роботи.

В результаті інтерфейс із користувачами різними інтернет-порталами організовано по різному. Головні причини три: перша – низька обізнаність із цими класифікаторами та відсутність стимулів (як і дієвих механізмів примусу) їх дотримуватися; друга – бажання формулювати маркетингово привабливі назви посад і навичок з метою максимізації кількості претендентів; третє – інерція сформованих шаблонів та традицій назв посад, навичок, функцій тощо.

Згідно результатів дослідження національних експертів Соріогло В.Г. та Михайлова С.Л. [11] для усіх приватних інтернет-посередників є характерною заплутана і нетипова класифікація назв вакансій та їх ключових елементів (видів економічної діяльності, описів функцій та вимог до кваліфікації і навичок, режимів роботи; типу трудового договору; терміну дії трудового договору тощо). У більшості українських сайтів посередників провідна категорізація є стихійною сумішшю професійних назв робіт, професій, видів та секторів економічної діяльності. Найбільша проблема не з тими назвами

вакансій, що не представлено у класифікаторах, а в тому, що назва така як у класифікаторах, однак опис її змісту розкриває, що мається на увазі зовсім інша категорія того ж класифікатору або їх суміш. Наприклад, розповсюджена назва роботи «менеджер» може в одному випадку означати дійсно управлінця і бути віднесеною до 1 групи КП, а іншому бути просто привабливою назвою звичайного продавця і за функціоналом відповідати роботам 4 групи КП. Є розповсюдженим прийом, коли задля підвищення привабливості частину назви вакансії, або її окремі елементи опису пишуть англійською мовою, або навіть усталеними українським або російським калькуванням. Наприклад, вакансія «project manager» буде в одному випадку перекладена правильно: «менеджер проектів», в іншому буде використана українізована калька англійського озвучення назви вакансії: «проджект менеджер», або взагалі може бути вигадана нова варіація «проектний менеджер» [12].

Одна із найбільш поширених проблем опрацювання бази вакансій – це дублювання записів. Досить заплутаною є технологія точної територіальної локалізації вакансій. Є поширеною практика коли фірма, що пропонує вакансію розташована в одному регіоні, але пропонує роботу в інших регіонах або за кордоном. Наприклад, маляр (адресат вакансії локалізований у м. Київ) - в описі уточняється, що вакансія у Словаччині.

Обмежень, рекомендації чи настанов щодо використання російської чи української мови приватні інтернет ресурси не мають. Однак, можна спостерігати певні закономірності використання мови:

А) вакансії роботодавців локалізованих в західних регіонах переважно публікуються української мовою;

Б) північних та центральних регіонів – практично однаково російською та українською;

В) південних та східних – переважно російською.

Досить часто зустрічаються вакансії, які опубліковані англійською мовою. В більшості випадків такі вакансії публікують компанії орієнтовані на

роботу із закордонними партнерами (найбільш часто в IT сфері), або ж безпосередньо іноземні підприємства.

Аналіз порталів вакансій щодо переліку та формату даних дає підстави для висновку, що дані цих джерел мало узгоджуються між собою, часто надають неактуальну інформацію, багато в чому дублюються, в той же час як важливі моменти ринку праці залишаються без відповіді. В результаті сформулювати єдину і несуперечливу картину українського ринку праці, якою б могли одночасно користуватися державні структури, дослідники, населення та підприємства, сьогодні неможливо. Формування єдиної інформаційної системи ринку праці, де усі могли б знайти актуальну для себе інформацію, вимагає розробки максимально простих алгоритмів систематизації та класифікації публічних вакансій, розміщених в онлайн середовищі.

Розроблення такого алгоритму дозволило б здійснити на аналітичному рівні об’єднання даних державної статистики, даних ДЦЗ та приватних онлайн посередників. Побувати на об’єднаних даних показники українського ринку праці, які точніше показують ринкову ситуацію, перспективи його розвитку. Крім того, забезпечення єдиної класифікаційної платформи для державних і недержавних даних дозволить доповнити знання експертів та державного регулювання новою інформацією щодо нових навичок і компетенцій, що стають актуальними на ринку праці. Це дозволить більш обґрунтовано формувати систему професійної підготовки та вищої освіти, планувати державне замовлення на фахівців, вчасно заповнювати прогалини дефіцитних спеціалістів.

1.3 Обстеження робочої сили

Обстеження робочої сили надає інформацію переважно з позицій пропозиції робочої сили, тому структурування показників здійснюється

передусім за соціально-демографічними характеристиками респондентів. Для оцінки структури робочих місць більшої ваги набувають характеристики виробничих одиниць, які ці робочі місця утворили.

З цієї позиції, для структурування результатів обстеження робочої сили якнайкраще підходить ознака виду економічної діяльності, оскільки вона визначається саме для виробничої одиниці, де працює респондент. Більше того, Держстат України визначає розподіл зайнятого населення за видами економічної діяльності з використанням комплексної оцінки, яка передбачає інтеграцію даних обстеження робочої сили, державних статистичних спостережень підприємств та адміністративної звітності [13]. Гармонізація цих даних додає упевненості, що вид економічної діяльності характеризує саме специфіку виробничих одиниць, а не професійну приналежність зайнятих осіб.

Чисельність зайнятих осіб не дорівнює точній кількості зайнятих робочих місць, з причини змінності, а також різних варіантів поділ робочого місця між працівниками та навпаки. Але структура зайнятих осіб за видами економічної діяльності достатньо повно відображає структуру задоволеної зустрічної пропозиції робочих місць за секторами економіки.

Чисельність безробітних осіб відображає частину робочої сили, яка перебуває в пошуку чи очікуванні своїх робочих місць, які наразі є вакантними або лише стадії створення. Результати обстеження робочої сили дають інформацію для розподілу за видами економічної діяльності безробітних, які раніше мали зайнятість. Звичайно, вид економічної діяльності виробничої одиниці на їх наступному робочому місці може не збігатися з попереднім, навіть за незмінної професії. Тим не менше, існує велика інерційність в переходах, особливо за відсутності цілеспрямованих структурних реформ в економіці та нерозвинутості сфери освіти дорослих. Тому цілком логічно припускати, що в більшості випадків наступне робоче місце буде пов’язане з тим самим сектором економіки, що й попереднє. Ця гіпотеза менш вірогідна у випадку молодих пошукувачів роботи, особливо, якщо вони знаходяться на

стадії переходу від тимчасових підробітків, поєднаних з професійним навчанням, до стабільної зайнятості за отриманим фахом [14]. Повна невизначеність існує у випадку безробітних, які шукають свої перше робоче місце, і не мають досвіду попередньої роботи.

З урахуванням вище сказаного, розподіл за видами економічної діяльності дає змогу отримати характеристику структури повної пропозиції робочої сили, і порівняти наскільки збігаються чи різняться розподіли зайнятих та безробітних, виявляючи попутно конфігурацію робочих місць за секторами економіки.

У 2019 р. 91,8% загальної пропозиції робочої сили становили зайняті особи і 8,2% – безробітні (табл. 1.2). Найбільшими секторами економіки України, які формують понад половину всіх робочих місць, традиційно залишаються: оптова та роздрібна торгівля, ремонт автотранспортних засобів і мотоциклів (4,1 млн осіб або 22,5% робочої всієї сили); сільське господарство, лісове господарство та рибне господарство (3,2 млн осіб або 17,8%); промисловість (2,7 млн осіб або 15,0%). Найменш поширені робочі місця у видах діяльності, що належать до сфери комерційних послуг: тимчасове розміщування й організація харчування; діяльність у сфері адміністративного та допоміжного обслуговування; інформація та телекомунікації; операції з нерухомим майном; фінансова та страхова діяльність; мистецтво, спорт, розваги та відпочинок (до 2% робочої сили в кожному). Низька ємність цих видів діяльності пояснюється як нерозвинутістю ринку таких послуг, так і загалом обмеженою місткістю в плані поглинання робочої сили.

Таблиця 1.2 – Структура робочої сили за статусом на ринку праці та видами економічної діяльності підприємств у 2019 р [13].

Види економічної діяльності	Зайняті		Безробітні		Робоча сила	
	тис.	%	тис.	%	тис.	%
Зайняте населення у віці 15-70 років	16578,3	91,8	1487,7	8,2	18066,0	100,0
Сільське господарство, лісове господарство та рибне господарство	3010,4	16,7	209,7	1,2	3220,1	17,8
Промисловість	2461,5	13,6	252,9	1,4	2714,4	15,0
Будівництво	699,0	3,9	129,6	0,7	828,6	4,6
Оптова та роздрібна торгівля; ремонт автотранспортних засобів і мотоциклів	3801,3	21,0	263,1	1,5	4064,4	22,5
Транспорт, складське господарство, поштова та кур'єрська діяльність	999,0	5,5	55,9	0,3	1054,9	5,8
Тимчасове розміщування й організація харчування	304,0	1,7	41,9	0,2	345,9	1,9
Інформація та телекомунікації	289,2	1,6	14,0	0,1	303,2	1,7
Фінансова та страхова діяльність	211,6	1,2	19,1	0,1	230,7	1,3
Операції з нерухомим майном	259,7	1,4	0,0	0,0	259,7	1,4
Професійна, наукова та технічна діяльність	421,6	2,3	14,0	0,1	435,6	2,4
Діяльність у сфері адміністративного та допоміжного обслуговування	317,9	1,8	0,0	0,0	317,9	1,8
Державне управління й оборона; обов'язкове соціальне страхування	870,5	4,8	78,8	0,4	949,3	5,3
Освіта	1388,7	7,7	96,6	0,5	1485,3	8,2
Охорона здоров'я та надання соціальної допомоги	974,2	5,4	50,8	0,3	1025,0	5,7
Мистецтво, спорт, розваги та відпочинок	197,6	1,1	0,0	0,0	197,6	1,1
Інші види економічної діяльності	372,1	2,1	44,5	0,2	416,6	2,3
Не працювали раніше	x	x	216,8	1,2	216,8	1,2

Обстеження робочої сили також надає інформацію про способи пошуку роботи, що дає можливість скласти уявлення про основні моделі взаємодії між відкритим попитом та пропозицією на ринку праці. Зі сторони пропозиції робочої сили, найпоширенішими способами пошуку роботи є звернення до державної служби зайнятості (34,6% безробітних), задіяння особистих зв’язків (29,3%), вивчення оголошень в інтернеті (16,5%). Водночас лише 0,9% зверталися до приватної фірми з працевлаштування, 2,6% розмістили власне резюме в Інтернеті на спеціальних або професійних сайтах, 6,0% вивчали

оголошення в пресі і 6,8% зверталися безпосередньо до роботодавців (рис. 1.1) [13].

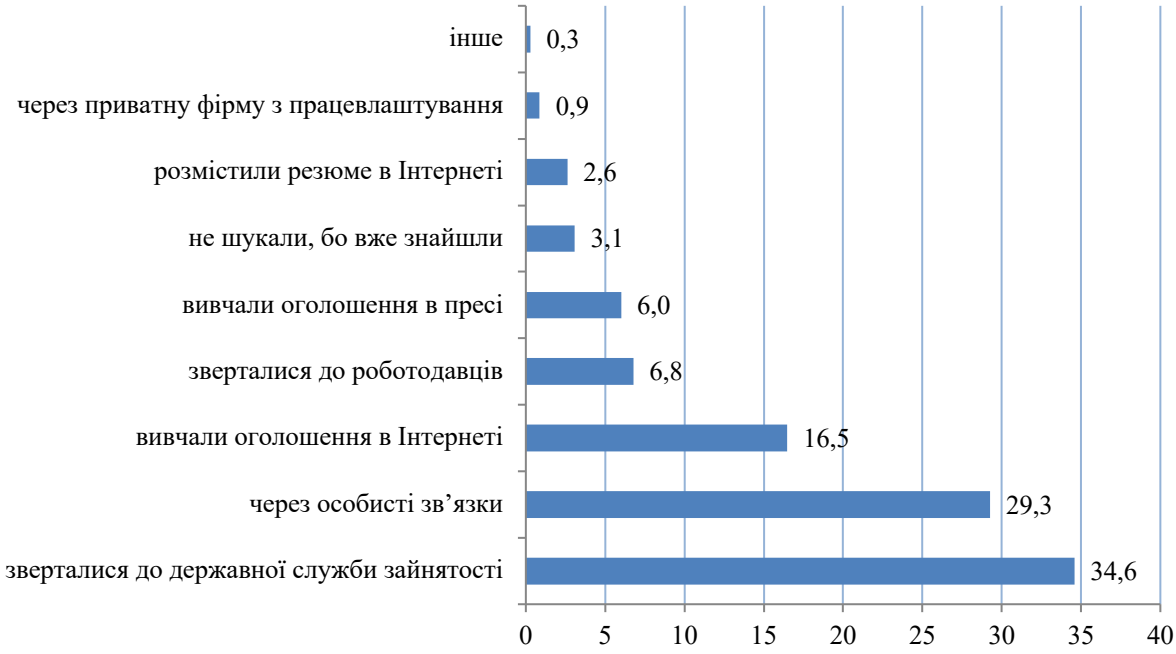

Рисунок 1.1. – Розподіл безробітних за способи пошуку роботи у 2019 р., %

Як свідчить розподіл, безробітні звертаються до державної служби зайнятості дійсно в пошуках роботи, і для багатьох це найбільш доступний і прийнятний варіант заявити про себе. Тобто роботодавцям не варто дотримуватися упереджень, що зареєстровані безробітні не хочуть працювати, а лише отримувати допомогу по безробіттю.

Дуже мала кількість звернень до приватних агентств з працевлаштування не свідчить про недоступність їхніх послуг через дорожнечу, адже для шукачів роботи ці послуги безкоштовні. Скоріше це свідчить, що шукачі роботи користуються послугами приватних агентств з працевлаштування дистанційно і без особистого контакту – переглядаючи оголошення про вакансії в Інтернеті чи в пресі. Причому самі шукачі майже припинили розміщувати свої резюме на сайтах чи іншим способом, що може свідчити про певну пасивність і готовність підлаштовуватися під існуючий запит з боку роботодавців.

Висновок до розділу 1

В першому розділі магістерської дисертації було детально обстежено предметну область обраної теми. Проблематика задачі збору, зберігання, уніфікування та аналізу даних щодо вільних вакантних місць на ринку праці України дослідження з різних боків. Було проведено загальний аналіз ринку праці за останні роки, а саме робочу силу. Визначено в чому актуальність систематизації та аналізу вакансій українського онлайн сегменту ринку праці.

Загалом складається враження, що інформування зацікавлених осіб про український ринок праці є досить насиченим і розмаїтим, торкається різних аспектів та відбувається досить регулярно. Однак, аналіз згаданих джерел щодо переліку та формату даних дає підстави для висновку, що дані цих джерел мало узгоджуються між собою, часто надають неактуальну інформацію, багато в чому дублюються, в той же час як важливі моменти ринку праці залишаються без відповіді. В результаті сформулювати єдину і несуперечливу картину українського ринку праці, якою б могли одночасно користуватися державні структури, дослідники, населення та підприємства, сьогодні неможливо. Формування єдиної інформаційної системи ринку праці, де усі могли б знайти актуальну для себе інформацію, вимагає значних методичних та організаційних зусиль, подолання численних інституційних бар'єрів, узгодження методології збору та обробки даних, їх таксономії, а також застосування нових наукових інформаційних методів та технологій.

Отже, поставлена ціль цієї роботи – розробка системи класифікації вакансій згідно класифікатора професії начасі та актуальна. Така система наразі – це єдина можливість отримувати актуальну інформацію щодо ринку праці в Україні регулярно, комплексно і в реальному часі.

РОЗДІЛ 2 ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Методи штучного інтелекту для обробки природної мови

Під штучним інтелектом розуміють комплекс технологічних рішень, який дозволяє імітувати когнітивні (розумні) функції людини та отримувати результати, які можна порівняти, як мінімум, з результатами інтелектуальної діяльності людини. При цьому імітація включає самонавчання та пошук рішень без заздалегідь заданого алгоритму.

Комплекс технологічних рішень включає інформаційно-комунікаційну інфраструктуру, програмне забезпечення (у тому числі в якому використовуються методи машинного навчання), процеси та послуги з обробки даних та пошуку рішень.

Принцип роботи штучного інтелекту полягає у поєднанні великого обсягу даних із можливостями швидкої, ітеративної обробки цих даних інтелектуальними алгоритмами, що дозволяє програмам автоматично навчатися на базі закономірностей та ознак, що містяться в даних.

Штучний інтелект є комплексною дисципліною з безліччю теорій, методик і технологій. Ключовими поняттями в штучного інтелекту є:

Машинне навчання - алгоритми аналізу даних з метою знайти в них закономірності. У ньому використовуються методи нейромереж, статистики, дослідження операцій тощо. для виявлення прихованої корисної інформації даних; при цьому явно не програмуються інструкції, що вказують, де шукати дані та як робити висновки.

Нейромережа – це один із методів машинного навчання; математична модель, і навіть її програмне чи апаратне втілення, побудована за принципом організації та функціонування біологічних нейронних мереж — мереж нервових клітин живого організму. У загальному випадку штучна нейронна мережа (ІНС) може складатися з декількох шарів найпростіших процесорів

(нейронів), кожен з яких здійснює деяке математичне перетворення (обчислює результат математичної функції) над вхідними даними та передає отриманий результат на наступний шар або вихід мережі.

2.1.1 Обробка природної мови

Обробка природної мови (Natural Language Processing, NLP) - перетин машинного навчання та математичної лінгвістики, спрямований на вивчення методів аналізу та синтезу природної мови. NLP застосовується у багатьох сферах, зокрема голосових помічниках, автоматичних перекладах тексту і фільтрації тексту.

Основними трьома напрямками є: розпізнавання мови (Speech Recognition), розуміння природної мови (Natural Language Understanding) та генерація природної мови (Natural Language Generation).

Глибоке навчання суттєво підвищило (і продовжує підвищувати) якість машинної обробки мови, для деяких завдань – уже впритул до людського рівня.

Організації можуть використовувати NLP-додатки двома способами:

– для розуміння запитів користувача, сформульованих природною мовою (як текстових, так і голосових):

– забезпечення більш якісної та таргетованої відповіді за допомогою розуміння питань користувачів та їх бажань;

– виявлення зовнішніх запитів та подання інтелектуальних альтернатив.

Для розуміння змісту текстів (витяг інформації з масивів неструктурованих даних):

– вилучення з текстових документів юридичної інформації для ідентифікації співробітників, клієнтів, продукції, процедур тощо;

– ідентифікація та розуміння значення контенту природною мовою (документи, звіти, електронна пошта тощо) з метою надання відповідей природною мовою.

NLP дозволяє краще розуміти запити користувачів та аналізувати корпоративну інформацію. Використання даної технології гарантує, що у кожного користувача є доступ до найбільш актуальних, корисних джерел інформації, які б інакше залишилися прихованими у величезних обсягах даних.

Основні типи обробки природної мови:

– оптичне розпізнавання символів (Optical Character Recognition): механічний або електронний переклад зображень рукописного, машинописного або друкованого тексту в текстові дані, що використовуються для представлення символів на комп'ютері.

– розпізнавання мови (Speech Recognition): перетворення сказаних слів на дані, які може зрозуміти комп'ютер. Це NLP-технологія, яка використовується у голосових помічниках типу Siri, Аліса, Cortana, Echo або Google Assistant.

– машинний переклад (Machine Translation): переклад тексту з однієї мови іншою. Ця технологія є основою таких програм для перекладу, як Google Translate або Яндекс Перекладач.

– виведення інформації людською мовою (Natural Language Generation): ця технологія використовується, коли Аліса, Siri або Cortana відповідають на ваше запитання.

– аналіз настроїв (Sentiment Analysis): вилучення даних з контексту (часто «великого тексту», big text) та оцінка того, чи ці дані є емоційно негативними чи позитивними.

– Семантичний пошук (Semantic Search): тісно пов'язана з розпізнаванням мови технологія, яка дозволяє ставити питання голосовим помічникам як при розмові з іншою людиною.

– програмування природною мовою (Natural Language Programming): це інструменти, які дозволяють користувачам створювати програми та програмне забезпечення, використовуючи команди природною мовою (замість програмування традиційним способом).

– афективні обчислення (Affective Computing): використання NLP та інших технологій для розуміння та відтворення людських емоцій.

NLP вирішує великий набір завдань, який можна розбити за рівнями (у дужках). Серед цих завдань можна виділити такі:

– Розпізнавання тексту, мови, синтезу мови (сигнал);

– Морфологічний аналіз, канонізація (слово);

– POS-тегування, розпізнавання іменованих сутностей, виділення слів (словосполучення);

– Синтаксичний розбір, токенізація речень (пропозиція);

– Вилучення відносин, визначення мови, аналіз емоційного забарвлення (абзац);

– Анотація документа, переклад, аналіз тематики (документ);

– Дедублікація, інформаційний пошук (корпус).

Термін корпус, що є основним та найуживанішим терміном на позначення підібраної та обробленої за певними правилами сукупності текстів, що використовуються як база для вивчення мови.

2.1.2 Дистрибутивна семантика

Дистрибутивна семантика (distributional semantics) - це область лінгвістики, яка займається обчисленням ступеня семантичної близькості між лінгвістичними одиницями на підставі їх розподілу (дистрибуції) у великих масивах лінгвістичних даних (текстових корпусах).

Розподільна семантична модель (distributional semantic model - DSM) — це масштабована та/або трансформована матриця спільного входження M, у якій кожен рядок x представляє розподіл цільового терміна в різних контекстах.

Кожному слову надається свій контекстний вектор. Безліч векторів формує словесний векторний простір.

DSM – вектор як субсимволічне представлення значення використовується як:

– вектор ознак для машинного навчання;

– вхідні дані для нейронної мережі.

Модель розподілу (distributional model):

– фіксує лінгвістичний розподіл кожного слова у формі багатовимірного числового вектора;

– зазвичай (але не обов’язково) базується на підрахунку одночасних випадків I гіпотезі розподілу:

– дистрибутивна подібність/відстань ≥ семантична подібність/відстань.

Розподілене представлення (distributed representation)

– субсимволічне представлення слів як багатовимірних числових векторів;

– подібність векторів зазвичай (але не обов'язково) відповідає семантичній подібності слів;

– unsupervised neural word embeddings.

Як спосіб представлення моделі використовуються векторні простори з лінійної алгебри. Інформація про дистрибуцію лінгвістичних одиниць подається у вигляді багаторозрядних векторів, які утворюють векторний словесний простір. Вектори відповідають лінгвістичним одиницям (словам чи словосполученням), а виміри відповідають контекстам. Координати векторів є числами, що показують, скільки разів це слово або словосполучення зустрілося в даному контексті.

Приклад словесного векторного простору, що описує дистрибутивні характеристики слів tea і coffee, в якому контекстом є сусіднє слово:

$$A_{m,n} = \begin{array}{c} \\ \textit{coffee} \\ w_2 \\ \textit{tea} \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{cccccc} w_1 & \textit{drink} & w_3 & \dots & w_m \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 2 & \dots & 0 \\ 0 & 2 & 0 & \dots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix} \end{array}$$

Рисунок 2.1 - Розподільна семантична модель для слів tea і coffee.

Розмір контекстного вікна визначається цілями дослідження:

– встановлення синтагматичних зв'язків -1-2 слова;

– встановлення парадигматичних зв'язків – 5-10 слів;

– встановлення тематичних зв'язків – 50 слів і більше [15].

Семантична відстань або семантична схожість між поняттями, вираженими словами природної мови, зазвичай обчислюється як косинусна відстань між векторами словесного простору. Косинусна міра обчислюється за формулою:

$$\frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \qquad (2.1)$$

Після проведення такого аналізу стає можливим виявити найближчі за змістом слова стосовно досліджуваного слова.

Також важливим терміном в дистрибутивній семантиці є дистрибутивна гіпотеза. Вона полягає у тому, що схожі слова схильні з’являтися в схожому контексті.

У графічному вигляді слова можуть бути представлені як точки на площині, при цьому точки, що відповідають близьким за змістом словами, розташовані близько один до одного. Приклад словесного простору, що

описує предметну область суперкомп'ютери, із роботи Генріха Шутце (рис. 2.2) [16]:

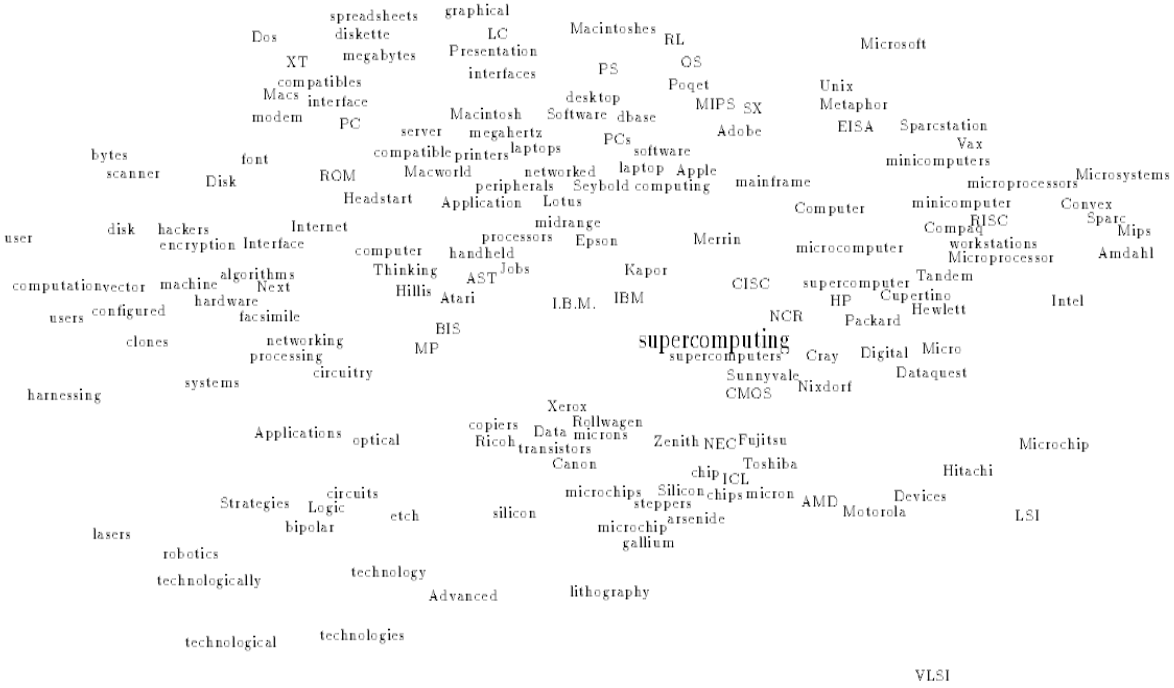

Рисунок 2.2 – Графічне зображення слів відповідно до семантичної відстані.

2.1.3 Word embedding метод Word2Vec

У обробці природної мови (NLP) word embedding — це термін, який використовується для представлення слів для аналізу тексту, як правило, у формі дійсного вектора, який кодує значення слова таким чином, що слова, розташовані ближче у простірному векторі, як очікується, буде подібним за значенням. Тобто word embedding це техніка отримання DSM – вектору. Word embedding можна отримати за допомогою набору методів моделювання мови та вивчення функцій, де слова чи фрази зі словника зіставляються з векторами дійсних чисел.

Методи генерації цього відображення включають нейронні мережі, зменшення розмірності на матриці спільного входження слів, імовірнісні

моделі, пояснюваний метод бази знань і явне представлення в термінах контексту, в якому з’являються слова.

Було показано, що вбудовування слів і фраз, які використовуються як базове представлення вхідних даних, покращує продуктивність у таких завданнях NLP, як синтаксичний аналіз і аналіз настроїв.

Word2vec — це техніка обробки природної мови, опублікована в 2013 році. Алгоритм word2vec використовує модель нейронної мережі для вивчення асоціацій слів із великого корпусу тексту. Після навчання така модель може виявляти слова-синоніми або пропонувати додаткові слова для часткового речення. Як випливає з назви, word2vec представляє кожне окреме слово певним списком чисел, який називається вектором. Вектори вибираються ретельно, щоб вони відобразили семантичні та синтаксичні якості слів; проста математична функція (косинусна подібність) може вказати рівень семантичної подібності між словами, представленими цими векторами.

Word2vec — це група пов’язаних моделей, які використовуються для створення word embedding. Ці моделі являють собою неглибокі двошарові нейронні мережі, які навчені реконструювати лінгвістичні контексти слів. Word2vec приймає як вхідні дані великий корпус тексту та створює векторний простір, як правило, розмірністю кілька сотень, причому кожному унікальному слову в корпусі призначається відповідний вектор у просторі. Вектори слів розташовані у векторному просторі таким чином, що слова, які мають спільні контексти в корпусі — тобто вони семантично й синтаксично подібні — розташовані близько одне до одного в просторі. Більш різнорідні слова розташовані далі одне від одного в просторі [17].

У Word2vec представлено дві модельні архітектури для створення цих розподілених представлень слів: безперервний пакет слів (CBOW) або безперервний скіп-грам. В обох архітектурах word2vec розглядає як окремі слова, так і ковзаюче вікно контекстних слів, що оточують окремі слова, під час ітерації по всьому корпусу.

У безперервній архітектурі пакета слів (Continuous bag-of-words model) модель прогнозує поточне слово з вікна навколишніх контекстних слів. На вхід у нейронну мережу подається набір векторів слів заданого контексту і на виході отримуємо вектор цільового слова (рис. 2.3). Порядок контекстних слів не впливає на прогноз [18].

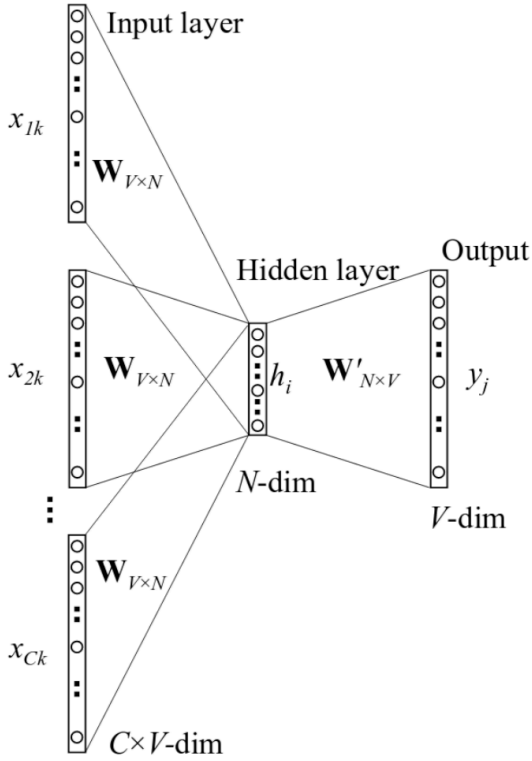

Рисунок 2.3 – Схема нейронної мережі CBOW

У безперервній архітектурі skip-gram модель використовує поточне слово для прогнозування навколишнього вікна контекстних слів. Архітектура skip-gram зважує найближчі контекстні слова на стільки більше, на скільки більш віддалені контекстні слова (рис. 2.4). Відповідно до примітки авторів, CBOW працює швидше, тоді як skip-gram краще справляється з нечастими словами (словами, що рідко зустрічаються у корпусі) [18].

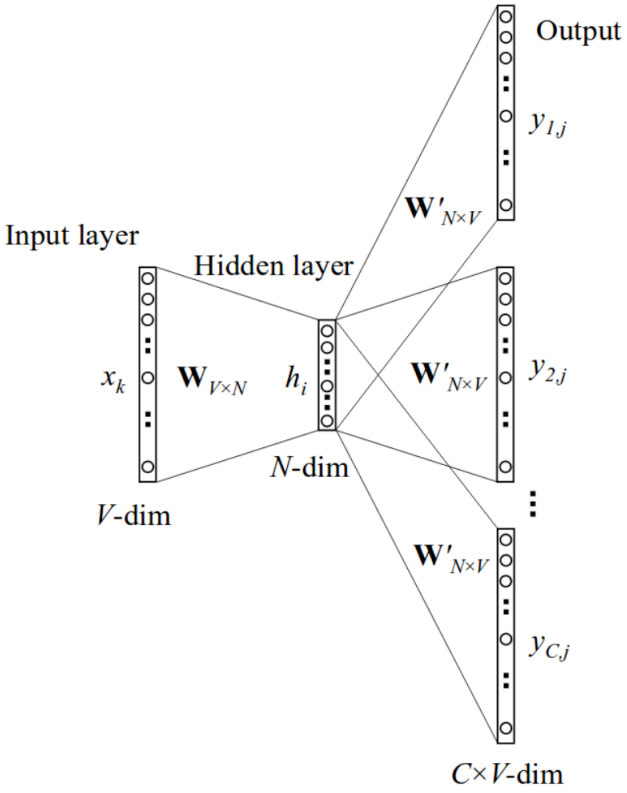

Рисунок 2.4 – Схема нейронної мережі skip-gram.

2.1.4 Основні підходи препроцесінгу мови

Передобробка тексту перекладає текст природною мовою у зручний для подальшої роботи формат. Передобробка складається з різних етапів, які можуть відрізнятися залежно від завдання та реалізації.

1. Стемінг

Кількість коректних словоформ, значення яких схожі, але написання відрізняються суфіксами, приставками, закінченнями та іншим, дуже велике, що ускладнює створення словників та подальшу обробку. Стеммінг дозволяє привести слово до його основної форми. Суть підходу у знаходженні основи слова, при цьому з кінця початку слова послідовно відрізаються його частини. Правила відсікання для стеммера створюються заздалегідь, і найчастіше являють собою регулярні висловлювання, що робить даний підхід

трудомістким, тому що при підключенні чергової мови потрібні нові лінгвістичні дослідження. Другим недоліком підходу є можлива втрата інформації при відрізанні частин, наприклад ми можемо втратити інформацію про частину мови.

2. Лематизація

Цей підхід є альтернативою стеммінгу. Основна ідея у приведенні слова до словникової форми – леми. Наприклад:

– для іменників – називний відмінок, однина;

– для прикметників — називний відмінок, однина, чоловічий рід;

– для дієслів, дієприкметників, дієприслівників — дієслово в інфінітиві недоконаного виду.

3. Векторизація

Більшість математичних моделей працюють у векторних просторах великих розмірностей, тому необхідно відобразити текст у просторі. Основним походом є мішок слів (bag-of-words): для документа формується вектор розмірності словника, для кожного слова виділяється своя розмірність, для документа записується ознака, наскільки часто слово зустрічається в ньому, отримуємо вектор. Найбільш поширеним методом для обчислення ознаки є TF-IDF [19] (TF – частота слова, term frequency, IDF – зворотна частота документа, inverse document frequency). TF обчислюється, наприклад, лічильником входження слова. IDF зазвичай обчислюють як логарифм від числа документів у корпусі, поділений на кількість документів, де це слово представлене. Таким чином, якщо якесь слово зустрілося у всіх документах корпусу, то таке слово не буде додане нікуди. Плюсами мішка слів є проста реалізація, проте цей метод втрачає частину інформації, наприклад, порядок слів. Для зменшення втрати інформації можна використовувати мішок N-грам (додавати не тільки слова, але й словосполучення), або використовувати методи векторних уявлень слів, це, наприклад, дозволяє знизити помилку на словах з однаковими написаннями, але різними значеннями.

4. Дедублікація

Оскільки кількість подібних документів у великому корпусі може бути велика, потрібно позбавлятися дублікатів. Так як кожен документ може бути представлений як вектор, ми можемо визначити їх близькість, взявши косинус або іншу метрику. Мінусом є те, що для великих корпусів повний перебір по всіх документах буде неможливим. Для оптимізації можна використовувати локально чутливий хеш, який помістить близько схожі об'єкти.

5. Семантичний аналіз

Семантичний (смисловий) аналіз тексту - виділення семантичних відносин, формування семантичного уявлення. У випадку семантичне уявлення є графом, семантичної мережею, що відбиває бінарні відносини між двома вузлами — смисловими одиницями тексту. Глибина семантичного аналізу може бути різною, а в реальних системах найчастіше будується лише синтаксико-семантичне подання тексту або окремих речень. Семантичний аналіз застосовується у завданнях аналізу тональності тексту[20](Sentiment analysis), наприклад, для автоматизованого визначення позитивності відгуків.

2.1.5 Бібліотеки для NLP

1. Natural Language ToolKit

NLTK є провідною платформою для створення програм Python для роботи з даними людської мови. Він надає прості у використанні інтерфейси для більш ніж 50 корпусів і лексичних ресурсів, таких як WordNet, а також набір бібліотек для обробки тексту для класифікації, токенізації, сформування основи, тегування, синтаксичного аналізу та семантичного міркування, оболонки для індустріальних бібліотек NLP, і активний дискусійний форум.

Завдяки практичному посібнику, який представляє основи програмування разом із темами з комп’ютерної лінгвістики, а також вичерпну

документацію API, NLTK підходить як для лінгвістів, інженерів, студентів, викладачів, дослідників, так і для промислових користувачів. NLTK доступний для Windows, Mac OS X і Linux. Найкраще те, що NLTK є безкоштовним проектом із відкритим кодом, керованим спільнотою.

NLTK називають «чудовим інструментом для навчання та роботи в комп'ютерній лінгвістиці з використанням Python» і «дивовижною бібліотекою для гри з природною мовою».

Плюси:

- найбільш відома та багатофункціональна бібліотека для NLP;
- велика кількість сторонніх розширень;
- швидка токенізація пропозицій;
- підтримується багато мов.

Мінуси:

- повільна;
- складна у вивченні та використанні;
- працює з рядками;
- не використовує нейронні мережі;
- нема вбудованих векторів слів [21].

2. SpaCy

SpaCy — це безкоштовна бібліотека з відкритим кодом для розширеної обробки природної мови (NLP) на Python. Віна розроблена спеціально для використання у виробництві та допомагає створювати програми, які обробляють і «розуміють» великі обсяги тексту. Його можна використовувати для створення систем вилучення інформації або систем розуміння природної мови.

Бібліотека, розроблена за методологією SCRUM мовою Cypthon, позиціонується як найшвидша бібліотека NLP. Має безліч можливостей, у тому числі, аналіз залежностей на основі міток, розпізнавання іменованих сутностей, позначка частин мови, вектори розміщення слів.

Плюси:

- найшвидша бібліотека для nlp;
- проста у вивченні та використанні;
- працює з об'єктами, а чи не рядками;
- є вбудовані векторні слова;
- використовує нейронні мережі для тренування моделей.

Мінуси

- менш гнучка порівняно з nltk;
- токенізація пропозицій повільніша, ніж у nltk;
- підтримує невелику кількість мов [22].

3. Gensim

Найшвидша бібліотека для навчання векторних кодувань слів (word embedding). Основні алгоритми в Gensim високо оптимізовані та розпаралелені процедури C.

Gensim може обробляти довільно великі корпуси, використовуючи алгоритми потокової передачі даних. Немає обмежень таких як "набір даних повинен поміститися в оперативну пам'ять".

Gensim працює на Linux, Windows і OS X, а також на будь-якій іншій платформі, яка підтримує Python і NumPy.

Завдяки тисячам компаній, які використовують Gensim щодня, понад 2600 наукових цитат і 1 мільйону завантажень на тиждень, Gensim є однією з найдосконаліших бібліотек ML.

Весь вихідний код Gensim розміщено на Github під ліцензією GNU LGPL, яка підтримується спільнотою з відкритим кодом.

Спільнота Gensim також публікує попередньо підготовлені моделі для певних сфер, як-от право чи охорона здоров’я, через проект Gensim-data [23].

Python бібліотека, розроблена за методологією SCRUM, для моделювання, тематичного моделювання документів та отримання подібності

для великих корпусів. У gensim реалізовані популярні NLP алгоритми, наприклад, word2vec. Більшість реалізацій можуть використати кілька ядер.

Плюси:

– Працює з великими датасетами;

– Підтримує глибоке навчання;

– word2vec, tf-idf vectorization, document2vec.

Мінуси

– Заточена під моделі без учителя;

– Не містить достатнього функціоналу, необхідного для NLP, що змушує використовувати її разом із іншими бібліотеками.

4. FastText

NLP-бібліотека FastText від Facebook Research стала наступним після Word2Vec великим кроком у розвитку векторних семантичних моделей та машинного навчання в обробці тексту.

FastText – це бібліотека, що містить передбачені готові векторні уявлення слів і класифікатор, тобто алгоритм машинного навчання, що розбиває слова на класи.

Для отримання векторного подання слів одночасно використовуються моделі skipgram та CBOW (Continuous Bag-of-Words).

Швидша робота в порівнянні з іншими пакетами та моделями. Для моделі векторних слів використовується skip-gram з негативним семплюванням. Негативне семплювання – це спосіб створити для навчання векторної моделі негативні приклади, тобто показати їй кілька слів, які не є сусідами по контексту. Для кожного позитивного прикладу (коли слова в тексті стоять поруч, наприклад, «пухнастий котик») ми підбираємо кілька негативних («пухнаста праска», «пухнастий радіосигнал», «пухнаста втеча»). Усього підбирається від 3 до 20 випадкових слів. Такий випадковий підбір кількох прикладів не потребує багато комп’ютерного часу і дозволяє прискорити FastText [24].

Skip-gram ігнорує структуру слова, але деякі мови мають складні слова, як, наприклад, німецькою. Тому до основної моделі було додано subword-модель. Subword-модель – це уявлення слова через ланцюжки символів (n-грами) з n від 3 до 6 символів від початку до кінця слова плюс саме слово цілком. Наприклад, слово замок з n = 3 буде представлено n-грамами <за, зам, амо, мок, ок > і послідовністю <замок>. Таким чином, для моделі є різниця між послідовністю <зам> у слові зам – і n-грамою зам із слова замок. Такий підхід дозволяє працювати і з тими словами, які модель раніше не зустрічала.

Ознаки, отримані з допомогою розбиття на n-грами, мають величезну розмірність (тобто виходить величезна важка таблиця). Це може уповільнити роботу моделі, що навчається на цих ознаках. Для фіксування розмірності ознак застосовується хешування ознак (спеціальна процедура, що дозволяє кодувати об’єкти різних розмірів за допомогою символьних ланцюжків однакової довжини). Ознаки набувають хеш-індекси, що допомагає зчитувати їх швидше.

В основі класифікатора лежить модель лінійної класифікації, що по архітектурі схожа з моделлю CBOW. Чим більша кількість класів, тим більше часу роботи лінійної моделі. Для оптимізації класифікатора використовується ієрархічний софтмакс, заснований на алгоритмі кодування Хаффмана.

Дерево класів складається з гілок, де в самому верху клас, що найбільш зустрічається, а його діти – пов’язані менш зустрічаються класи. Наприклад, у класу «Біологія» будуть діти вузли класів «Тварини» та «Рослини» тощо. Імовірність класів визначається для кожного вузла від першого вузла-батька до останнього вузла-дитини. Таким чином, у дочірнього вузла ймовірність завжди менша, ніж у його вузла-батька [25].

2.2 Інструменти та методи збору даних

Сформулювала задача дослідження полягає в тому, що необхідно створити технологічно-методичний інструментарій, який би дозволив зібрати усі дані щодо вакансій, які розміщені на відібраних сайтах протягом деякого фіксованого періоду часу.

Вихідним джерелом відбору аналогів для вирішення цього завдання було обрано сучасні інструменти та методи збору великих даних. Їх застосування для вивчення пропозиції робочих місць в Україні стримується відсутністю науково обґрунтованої адаптації відповідного методичного та технологічного забезпечення до специфіки завдань вивчення обсягів і структури вакансій. Переважна більшість технологічних і методичних рішень у сфері збору та приймання онлайн даних (ЗПоД) представляє собою універсальні та потужні платформи, заточені на реалізацію масштабних бізнес-проектів, з чималими програмними, консультаційними й інформаційними ресурсами. Для таких технологій і платформ фокусування на специфіці точкового й експериментально-пошукового застосування інструментів є нерентабельним та недостатньо цільовим.

У результаті пошуку способу збору інформації вирішено використовувати методологію вебскрапінгу. Оскільки дана методика є доволі гнучкою, вона допоможе вирішити специфічні завдання оцінки обсягів і структури пропозиції робочих місць та фактичні умови функціонування сайтів із пошуку роботи.

2.2.1 Вебскрапінг як джерело текстових даних

Вебскрапінг використовується для отримання тексту з веб-сторінок (рис. 2.5). Програмне забезпечення для сканування веб-сторінок розробляється для розпізнавання різних типів вмісту на веб-сайті та для отримання та зберігання лише типів вмісту, указаних користувачем, напр. назви статей чи авторів із веб-сайту новин або ціни й описи продуктів із комерційного веб-сайту. Можна використовувати комерційне програмне забезпечення або мови програмування.

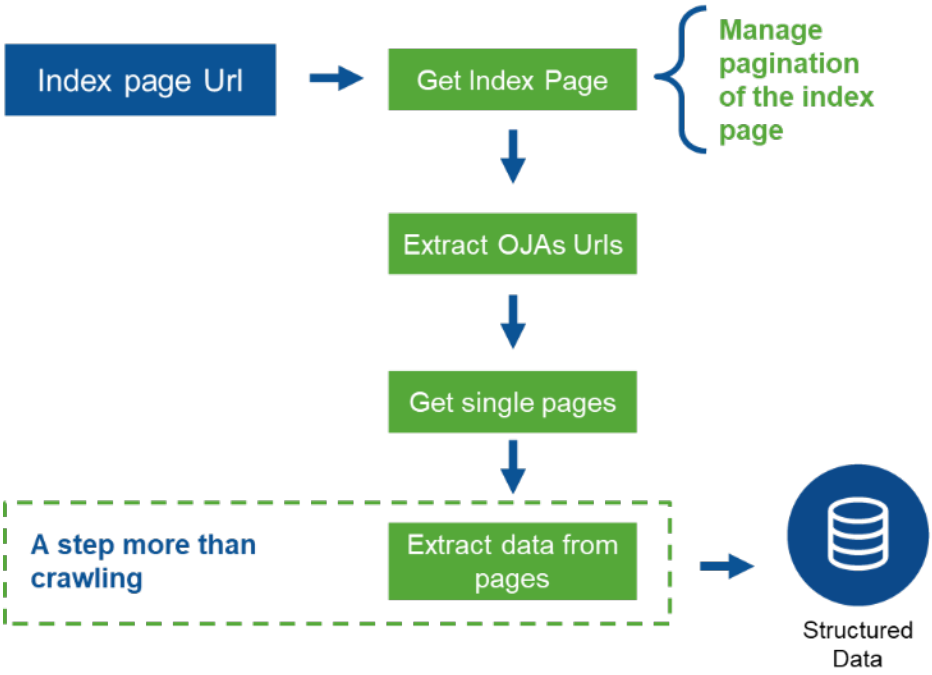

Рисунок 2.5 – Процес вебскрапінгу

Програмне забезпечення для сканування веб-сторінок може мати прямий доступ до всесвітньої павутини за допомогою протоколу передачі гіпертексту (HTML) або веб-браузера. Хоча вебскрапінг може виконуватися користувачем програмного забезпечення вручну, цей термін зазвичай стосується автоматизованих процесів, реалізованих за допомогою бота або веб-сканера. Це форма копіювання, за якої певні дані збираються та

копіюються з Інтернету, як правило, у центральну локальну базу даних або електронну таблицю, для подальшого пошуку чи аналізу.

Збирання веб-сторінки передбачає її отримання (fetching) та вилучення інформації з неї (рис. 2.6). Отримання – це завантаження сторінки (яке робить браузер, коли користувач переглядає сторінку). Таким чином, веб-сканування є основним компонентом веб-збирання для отримання сторінок для подальшої обробки. Після отримання можна розпочати вилучення.

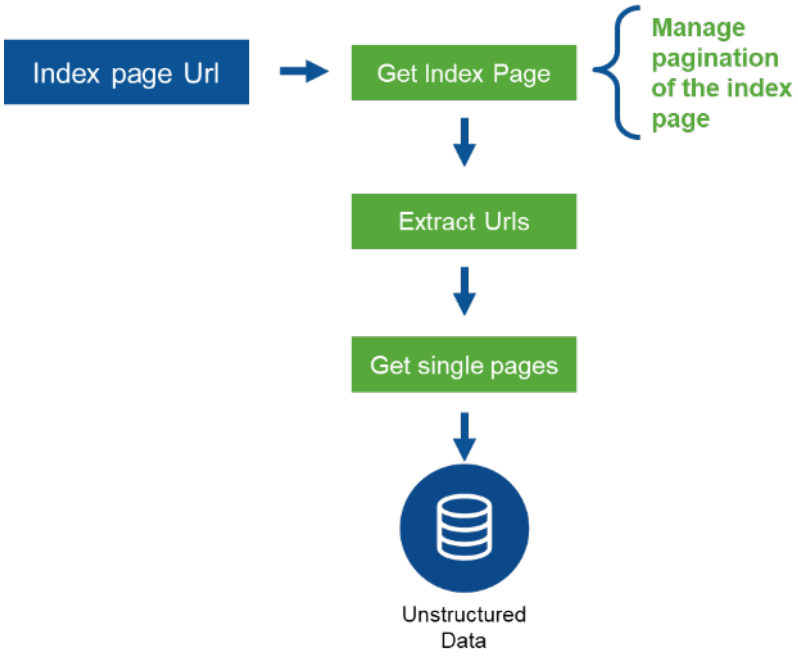

Рисунок 2.6 – Процес веб-сканування (crawling)

Вміст сторінки можна аналізувати, шукати та переформатувати, а її дані скопіювати в електронну таблицю або завантажити в базу даних. Програми вебскрапінгу зазвичай беруть щось зі сторінки, щоб використати це для іншої мети десь в іншому місці. Крім контактного сканування, вебскрапінг використовується як компонент програм, що використовуються для веб-індексування, веб-аналізу та інтелектуального аналізу даних, порівняння цін, аналіз огляду продукту (для спостереження за конкуренцією), збір списків нерухомості, моніторинг погодних даних, виявлення змін на веб-сайті, дослідження, відстеження присутності в Інтернеті та репутації.

2.2.2 Техніки вебскрапінгу

Вебскрапінг – сфера активних розробок, яка має спільну мету з баченням семантичної мережі, амбітною ініціативою, яка все ще потребує прориву в обробці тексту, семантичному розумінні, штучному інтелекті та взаємодії людини з комп'ютером.

1. Копіювання та вставка людини

Найпростіша форма копіювання веб-сайтів — це ручне копіювання та вставлення даних із веб-сторінки в текстовий файл або електронну таблицю. Іноді навіть найкраща технологія веб-збирання не може замінити ручне дослідження людини та копіювання та вставлення, а іноді це може бути єдиним працездатним рішенням, коли веб-сайти для збирання явно встановлюють бар'єри для запобігання машинній автоматизації.

2. Зіставлення шаблону тексту

Простий, але потужний підхід до отримання інформації з веб-сторінок може базуватися на команді UNIX grep або засобах зіставлення регулярних виразів мов програмування (наприклад, Perl або Python).

3. HTTP програмування

Статичні та динамічні веб-сторінки можна отримати, надсилаючи HTTP-запити на віддалений веб-сервер за допомогою програмування сокетів.

4. Розбір HTML

Багато веб-сайтів мають великі колекції сторінок, які генеруються динамічно з основного структурованого джерела, наприклад бази даних. Дані однієї категорії зазвичай кодуються на схожих сторінках за допомогою загального сценарію або шаблону. У інтелектуальному аналізі даних програма, яка виявляє такі шаблони в певному джерелі інформації, витягує його вміст і переводить його в реляційну форму, називається оболонкою. Алгоритми генерації оболонки припускають, що вхідні сторінки індукційної

системи оболонки відповідають загальному шаблону і що їх можна легко ідентифікувати за допомогою загальної схеми URL.

5. Аналіз веб-сторінки комп'ютерним зором

Існують спроби з використанням машинного навчання та комп’ютерного зору, які намагаються ідентифікувати та витягти інформацію з веб-сторінок шляхом візуальної інтерпретації сторінок, як це може зробити людина [26].

2.2.3 Бібліотека Selenium

Selenium - це набір програм з відкритим вихідним кодом, які застосовують для тестування веб-додатків та адміністрування сайтів локально та в мережі. Selenium спочатку був інструментом, створеним для тестування поведінки веб-сайту, але швидко став загальним інструментом автоматизації веб-браузера, який використовувався для веб-перегляду та інших завдань автоматизації.

Цей інструмент досить поширений і здатний автоматизувати різні браузери, такі як Chrome, Firefox, Opera і навіть Internet Explorer, за допомогою проміжного програмного забезпечення під назвою Selenium Webdriver.

Webdriver — це перший протокол автоматизації браузера, розроблений організацією W3C, і це, по суті, служба протоколу проміжного програмного забезпечення, яка знаходиться між клієнтом і браузером, перетворюючи команди клієнта на дії веб-браузера.

Переваги бібліотеки Selenium:

– Портативна веб-платформа для тестування з відкритим кодом.

– Selenium є комбінацією інструментів і DSL (domain-specific language) для проведення різних типів тестів.

– Простіше для розуміння та реалізації, команди Selenium класифікуються за різними класами, що полегшує розуміння та впровадження.

– Менше навантаження та стресу для тестувальників. Як згадувалося вище, кількість часу, необхідного для тестування повторних тестових сценаріїв для кожної нової збірки, скорочується майже до нуля. Таким чином, навантаження на тестера зменшується.

Ця бібліотека Python обертає Selenium WebDriver і надає методи автоматизації цілого ряду завдань, таких як заповнення форм, вхід на сайт, натискання на кнопки і багато іншого [27].

Висновки до розділу 2

В даному розділі був спрямований на дослідження можливих технічних інструментів, що допоможуть побудувати систему класифікації робочих місць в Україні, розміщених на онлайн сервісах. Технічна сторона даної роботи розбита на 2 окремі блоки.

Перший – це збір даних про вакансії представлені на українських онлайн ресурсах пошуках роботи. Для цього було вирішено використовувати методо вебскрапінгу. В розділі було детально розглянуто чому саме таких метод та які його практичні реалізації доступні для використання.

Другий блок – це розробка класифікатора текстової неструктурованої інформації. Найкращим рішенням для цього в ході дослідження різноманітних методів штучного інтелекту є алгоритми опрацювання та обробки природної мови. Розглянуто найпопулярніші методи NLP та досліджено, які проблеми та завдання вони можуть вирішувати. Приділено багато уваги поняттю дистрибутивної семантики, оскільки методи та алгоритмами саме з цієї області штучного інтелекту будуть використані у даній роботі. Розглянуто такі основні поняття у цій області як word embedding та архітектури нейронних

мереж Word2Vec. Реалізація програмного продукту буде виконана за допомогою таких бібліотек як nltk, gensim, FastText.

РОЗДІЛ 3 РОЗРОБЛЕННЯ ІНСТРУМЕНТАРІЮ ЗБОРУ Й ОБРОБКИ ДАНИХ ПРО ВАКАНСІЇ ОНЛАЙН СЕГМЕНТУ УКРАЇНСЬКОГО РИНКУ ПРАЦІ

3.1 Підбір джерел збору даних у вебмережі

Процедура відбору сайтів з вакансіями онлайн посередників для збору інформації про пропозицію робочих місць є надзвичайно важливим і кропітким етапом. Причинами цього є:

–значний перелік сайтів, які розміщують вакансії на своїх ресурсах (їхня кількість в Україні фахівцями в цій сфері оцінюється у понад 2 тис.);

–наявність як універсальних, так і спеціалізованих сайтів, які, у першому випадку, розміщують на свої ресурсах широкий перелік вакансій, а в другому – лише за окремими регіонами, видами діяльності, професіями тощо;

–висока конкуренція на ринку онлайн посередників, що зумовлює постійну появу нових і зникнення частини старих сайтів. Це потребує постійної актуалізації переліку сайтів, які задіяні для збору даних.

Ці причини не є специфічними для України. З ними стикаються дослідники ринку праці, які використовують для цього великі дані в усьому світі. Узагальнюючи існуючі зарубіжні розробки, можна виділити ряд дієвих та ефективних критеріїв щодо методики відбору сайтів онлайн посередників, які розміщують на своїх ресурсах пропозиції робочих місць (рис. 3.1).

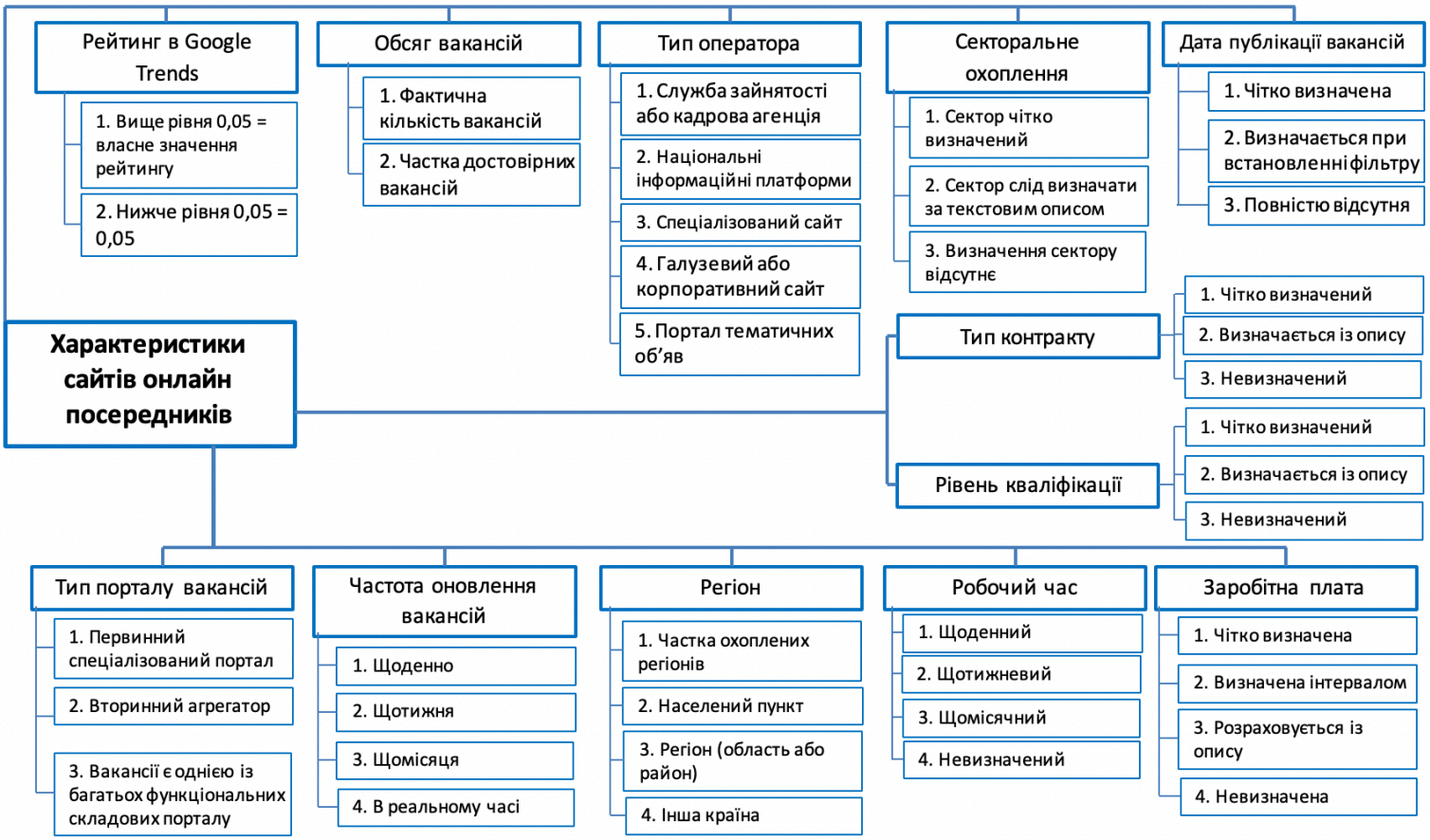

Рисунок 3.1 - Дерево критеріїв та їх варіацій для відбору релевантних сайтів, які використовуються для збору даних про вакансії

Дані критерії відбору можуть бути застосовані і в українських реаліях, однак важливим при цьому є й урахування вітчизняної специфіки. Унаслідок обмеженості часу та можливостей на практиці реалізувати повний перелік критеріїв, нами було визначено та реалізовано дієвий набір тих параметрів, які дозволяють відібрати найбільш інформативні сайти онлайн посередників.

Головним завданням при відборі сайтів онлайн посередників для нашого дослідження стала необхідність отримання максимально можливої кількості чітко визначеної та структурованої інформації про пропозицію робочих місць. Для цього першим критерієм, який застосовувався для відбору сайтів, став їхній рейтинг, який визначався за першочерговістю та частотою згадування у пошуковій системі Google та її додатку Google Trends. Це дало можливість скласти рейтинг із кількох десятків сайтів онлайн посередників, які згадувались на перших десяти сторінках пошуку. З'ясувалось, що

першочерговість та частота згадувань у значній мірі визначається кількістю вакансій, розміщених на сайтах онлайн посередників, їх актуальністю й універсальністю.

Важливим критерієм відбору сайтів стала достовірність даних про вакансії, які були розміщені на них. Достовірність визначалась шляхом перевірки кількості заявлених онлайн посередниками вакансій по тому чи іншому параметру із їхньою реальною кількістю, розміщеною на досліджуваному ресурсі. Окрім того, попередньо визначалась питома вага вакансій, які повторно розміщувались на сайті протягом короткого періоду. Дані критерії дали можливість відібрати ті сайти, в яких найменше відбувається спотворення інформації про структуру пропозиції робочих місць.

Для визначення актуальності інформації про пропозицію робочих місць важливим критерієм стала наявність програмних можливостей відбору найбільш свіжих публікацій. Нами було встановлено, що найбільш якісну інформацію оприлюднюють сайти, в яких кожна вакансія містить або дату публікації, або ж програмний модуль, який дозволяє відібрати вакансії, опубліковані не пізніше останніх 30 днів.

Слід зазначити, що значний вплив на відбір сайтів мали також:

– наявність назви установи, підприємства чи організації, яка пропонує робоче місце через онлайн посередників, що надає можливість перевірити актуальність і достовірність заявленої у вакансіях інформації, а також полегшує подальшу обробку сформованої бази даних, приведення її у відповідність до діючих державних класифікаторів;

– фіксація регіону розташування підприємств, які пропонують робочі місця, що використовується як у подальшій обробці бази даних вакансій, так і в аналізі регіонального зрізу пропозиції робочих місць;

– визначення рівня заробітної плати на пропонованих робочих місцях, що надає можливість визначити ринкові ціни пропозиції робочих місць та чинники, які її визначають;

– зазначення типу зайнятості, що дозволяє провести первинну оцінку умови зайнятості на пропонованих робочих місцях, її тривалість та напруженість;

– наявність визначеного досвіду роботи, що сприяє первинній оцінці рівня кваліфікації, необхідного на пропонованому робочому місці;

– присутність у вакансіях неструктурованого широкого опису вакансій, що є вкрай необхідним для її подальшої деталізації та структурування. Надалі наявність такої інформації є однією з базових умов для проведення комплексного аналізу й оцінки масштабів та структури пропозиції робочих місць в Україні.

З урахуванням визначених критерії для збору даних стосовно пропозиції робочих місць в Україні були відібрані наступні сайти: dcz.gov.ua, rabota.ua, work.ua, trud.ua, jobs.ua, olx.ua/rabota/, ua.jooble.org.

3.2 Збір та попередня очистка даних

Збір та приймання даних відібраних сайтів онлайн посередників із працевлаштування традиційним способам копіювання вручну є абсолютно непридатним для вчасного отримання й обробки релевантної інформації щодо робочих місць. Вихідним джерелом відбору аналогів для вирішення цього завдання було обрано сучасні інструменти та методи збору великих даних. Специфіка завдань оцінки обсягів і структури пропозиції робочих місць та фактичні умови функціонування сайтів із пошуку роботи визначає інструмент, який буде використано для її вирішення. Найкращим варіантом є використання методології вебскрапінгу. Розроблене програмне забезпечення з використанням бібліотеки Selenium.

На першому етапі розробки інструментів збору даних планувалося, що відповідне програмне забезпечення буде універсальним для усіх веб ресурсів.

У ході подальшої роботи виникли проблеми, що унеможливили забезпечити універсальний продукт для будь-якого сайту. Причинами цього стали особливості реалізації веб-ресурсів. Кожен сайт відображається у браузері користувача за допомогою HTML розмітки – це стандартна мова розмітки веб-сторінки, де кожен текстовий або медіа елемент заключений у відповідний тег, що визначає початок і кінець елемента. Згадані теги обов'язково мають ієрархію на сторінці, де кожен елемент вкладеній в інший. Модель побудови – стандарт для усіх веб ресурсів, але конкретне наповнення цієї ієрархії, порядок елементів, теги, які використані для їх означення, є індивідуальними для кожного сайту. Це стало на заваді для реалізації універсального інструменту збору даних. Для того, щоб вирішити цю проблему, було створено індивідуальне програмне забезпечення для кожного з обраних для дослідження сайтів. Алгоритм залишається незмінним, проте спосіб звернення до елементів сайту різний.

На другому етапі розробки – практичному, виникла інша проблема. Оскільки немає однієї встановленої домовленості в оформленні сайту пошуку вакансій, власники таких ресурсів відображають і називають блоки з інформацію про вакансію за власними вподобаннями, іноді вони навіть різняться на одному ж сайті. Для подолання цього необхідно розробити дискретований доступ програми до окремих блоків даних. Причому програма не повинна зациклюватися та запинятися навіть за відсутності відповідних блоків, або пустої множини їх заповнення.

Провідна логіка алгоритму, що покладено в основу класу програм, здатних вирішити обговорювані задачі, показана на рисунку 3.2. Згідно з ним, збір даних умовно поділено на 2 процеси: перший – робота з веб сторінкою, другий – запис добутої інформації у текстовому вигляді. Ці два процеси працюють почергово, оскільки кожен етап залежить від попереднього.

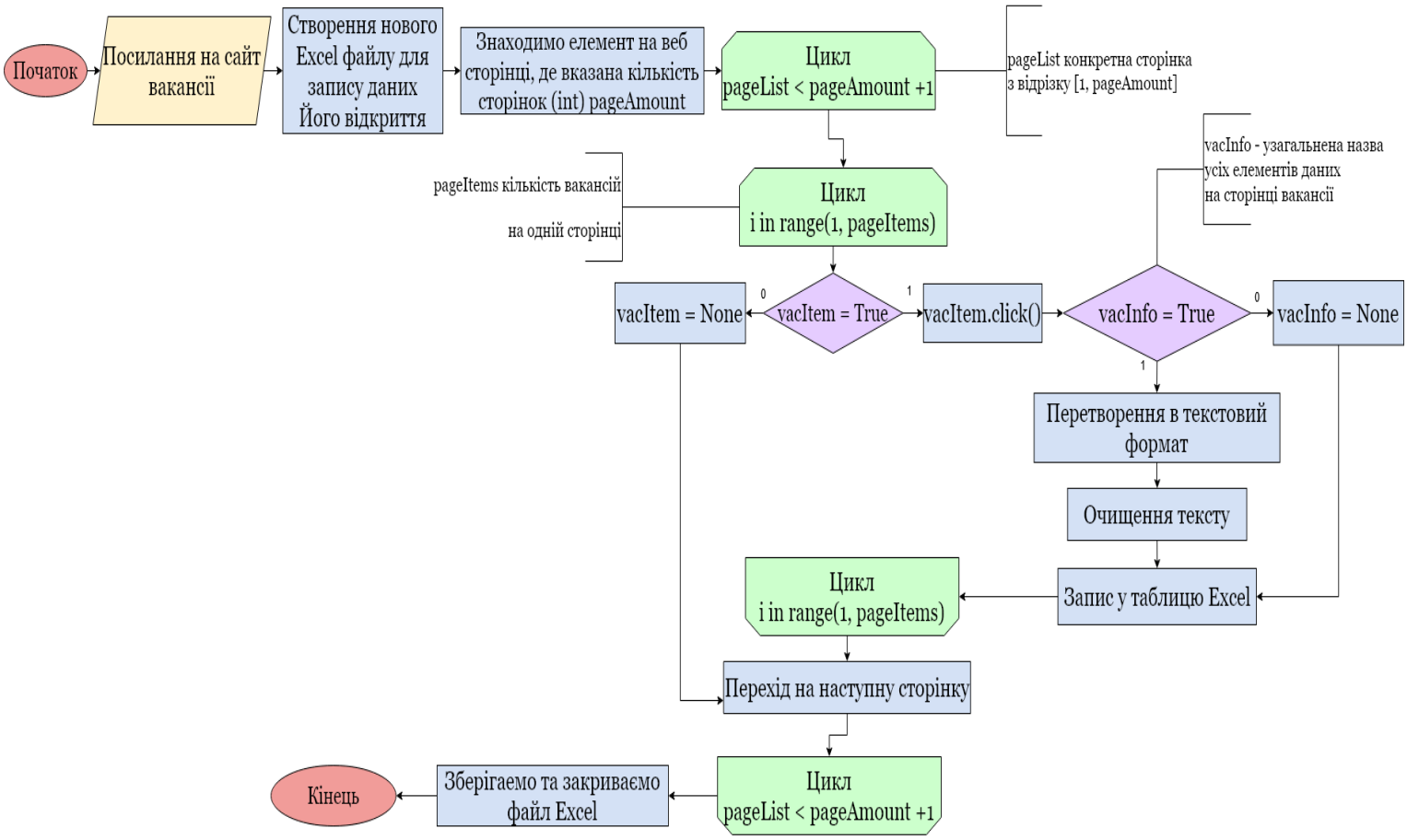

Рисунок 3.2 – Схема розробленого та реалізованого алгоритму скрапінгу онлайн даних щодо вакансій приватних посередників на ринку праці України

Перш за все необхідно підготувати файл у табличному форматі, куди надалі запишуться дані. Важливо забезпечити відповідність назв заголовків і даних, що туди надійдуть. Далі реалізується робота з веб браузером. Створюється об’єкт веб драйверу, за допомогою якого надалі буде відбуватися керування сторінкою в мережі. Навігація по сайту прописується безпосередньо у коді програми, для цього використовується імітація натискання клавіш. Відкривши безпосередньо сторінку зі списком вакансій, записуємо в таку структуру даних як список посилання на усі вакансії, що містяться на даній сторінці. Такий список зберігається у пам’яті, допоки не оновиться даними, або програма не закінчить своє виконання.

У роботі програми конче важливо отримати кількість сторінок у даній категорії вакансій, аби забезпечити циклічний прохід по сторінках. Це

завдання вирішується за допомогою циклу для зчитування даних з усіх вакансій на поточній сторінці. Цей цикл буде прокручуватися по кожному елементу зі заздалегідь підготовленого списку. На сторінці відкритої вакансії почергово збирається цільова інформація, така як назва вакансії, заробітна плата, місто, назва компанії, дата публікації вакансії, тип зайнятості, опис і т.д. Кожен із цих записів перетворюється у текст. Необхідно зазначити, що ця інформація не завжди подана на сайті в належному вигляді, тому далі за необхідності проходить форматування. Отриманий текст записується у підготовленій на початку файл. На цьому внутрішній цикл закінчується. Відбувається перехід на наступну сторінку. Якщо це неможливо, було пройдено усі сторінки, то програма закінчує виконання, закриває та зберігає файл Excel, закриває веб браузер.

У результаті роботи програми замість розрізнених і хаотично структурованих даних про вакансії різних сайтів, отримуємо базу даних, де в розрізі уніфікованих полів (назва вакансії; оплата; назва підприємства тощо) записано вектори опису всіх поточних вакансій із похибкою +/- 0,8% на тих, що існували або були прибрані протягом періоду між запуском програми скринінгу та її закінченням на одному сайті.

Важливим етапом формування бази даних про пропозицію робочих місць є первинна обробка великих даних, отриманих у результаті скрейпінгу. Уніфікація отриманої інформації у вигляді універсальної таблиці дає можливість дослідникам провести очищення первинно структурованих даних.

Головною причиною, яка викликає необхідність такого очищення, є наявність значної кількості повторів записів про вакансії. Наявність повторних записів у первинно структурованій базі даних значно спотворює інформацію про обсяги пропозиції робочих місць та їхню структуру, а тому потребує значної уваги по видаленню дублюючої інформації. Після пошуку та видалення усієї дублюючої інформації, можна перейти до уніфікування форматів колонок з даними, перевірити чи всі записи повні, якщо десь є критична інформація, що відсутня таких запис видаляється. Обробка першого

ступеню виконана. Далі датасет готовий до використання у класифікаторі вакансій.

3.3 Огляд змісту зібраних даних

Коректна робота алгоритму класифікації залежить не тільки від інструментів та підходів, які використовуються для його реалізації, але й також від врахування структури початкових даних та інформації, яку можна з них отримати. В цьому параграфі пропонується детальніше розглянути структуру та зміст зібраного датасету.

Як було описано вище, сайти для збору даних обирались з максимально можливою структурою, для легшої конкатенації датасету. Зазвичай онлай портали для пошуку роботи надають таку типову інформацію:

- назва вакансії;
- дата публікації вакансії;
- розмір зарплати;
- місто, в якому ведеться пошук робітника;
- назва компанії;
- сфера економічної діяльності компанії або підприємства;
- повний опис вакансії;
- вимоги до освіти;
- вимоги до досвіду;
- графік роботи.

На рисунку 3.3 представлено приклад датасету. З більшістю колонок проблем немає, окрім колонки з описом вакансії. Її особливість в тому, що кожний запис містить дуже великий об’єм текстової інформації, яка здебільшого не структурована від однієї вакансії до іншої, а також зміст. наповнення цього тексту змінюється. Зазвичай в даній колонці міститься

інформація про вимоги до працівника, необхідні навички, обов’язки та умови праці. Ця інформація є також корисною для дослідження ринку праці, але в дещо іншому направленні. Такі данні про умови, навички та вимоги можуть бути корисними для розширення класифікатора професій по горизонталі. Тобто це дасть змогу визначити, які навички та вимоги має кожна професія класифікатора, ввести альтернативні назви, або навіть додати нові професії. Така робота вже ведеться у європейських класифікаторах професій і активно розвивається у цьому напрямку. Це дає можливість не тільки слідкувати за актуальною ситуацією на ринку праці, але й налагодити комунікацію між ринком праці та освітою.

	A	B	C	D	E	F	G	H	I	J
1	vacancy	time	salary	city	company	industry	description	education	experience	timetable
2	Прибиральниця	5 серпня		Київ	Шен-Сервис	Інші пропозиції; Людям з обме	ОПИС ВАКАНСІЇКомпанія SHEI	не має значення	бажано	плаваючий графік роботи
3	Водитель на авто компании	5 серпня	30 000	Київ	ФЛП Соломко Д.С.	Транспорт, автосервіс	ОПИС ВАКАНСІЇСамые Лучши	не має значення	не вимагається	повний робочий день
4	Водитель на авто компании	5 серпня	25 000	Київ	Bolt	Транспорт, автосервіс; Робота	ОПИС ВАКАНСІЇАвтопарк боле	не має значення	не вимагається	повний робочий день
5	Водитель на авто компании	5 серпня	20 000	Харків	Bolt	Транспорт, автосервіс; Робота	ОПИС ВАКАНСІЇКомпания Бол	не має значення	не вимагається	повний робочий день
6	Водій на авто компанії Renau	5 серпня	20 000	Львів	Bolt	Транспорт, автосервіс; Робота	ОПИС ВАКАНСІЇАвтопарк, що н	не має значення	не вимагається	повний робочий день
7	Водитель на авто компании	5 серпня	20 000	Одеса	Bolt	Транспорт, автосервіс; Робота	ОПИС ВАКАНСІЇКомпания Бол	не має значення	не вимагається	повний робочий день
8	Водитель на авто компании	5 серпня	20 000	Дніпро	Bolt	Транспорт, автосервіс; Робота	ОПИС ВАКАНСІЇКомпания Бол	не має значення	не вимагається	повний робочий день
9	Донор яйцеклітин	7 липня	30 000	Київ	Клініка Генетики Репро	Інші пропозиції; Робота для ст	ОПИС ВАКАНСІЇПРОПОНУЄМС	не має значення	не вимагається	часткова зайнятість
10	Товарознавець	5 серпня	12 000	Київ	Мережа продуктових м	Торгівля, продажі, закупівлі	ОПИС ВАКАНСІЇТоварознавец	не має значення	обов'язковий	повний робочий день
11	Менеджер по продажам	5 серпня	20 000	Київ	Мехбуд	Торгівля, продажі, закупівлі; О	ОПИС ВАКАНСІЇВ связи с расш	не має значення	не вимагається	повний робочий день
12	Менеджер по продажам	5 серпня	20 000	Одеса	Мехбуд	Торгівля, продажі, закупівлі; О	ОПИС ВАКАНСІЇВ связи с расш	не має значення	не вимагається	повний робочий день

Рисунок 3.3 - Зразок зібраних даних

Задача обраної теми полягає у класифікації запропонованих робочих місць відповідно до класифікатора, тому для цього надалі буде використана колонка назви вакансії. Це спростить контроль якості класифікатора та дасть можливість зосередитися на технічній реалізації алгоритму, проте не на екстрактингу корисної інформації з опису вакансії.

Алгоритм класифікації працює не тільки з датасетом вакансій, але й з датасетом класифікатора професій. Він містить в собі назву професії, яку виконує працівник і спеціальний ієрархічний код.

Приклад датасету класифікатора професій наведено на рисунку 3.4.

	A	B
1	Code	Prof_job_title
2	7346	Авербандник
3	7232	Авіаційний механік з планера та двигунів
4	7241	Авіаційний механік з приладів та електроустаткування
5	7243	Авіаційний механік з радіоустаткування
6	7232	Авіаційний технік (механік) з парашутних та аварійно-рятувальних засобів
7	8155	Авіаційний технік з паливно-мастильних матеріалів
8	3115	Авіаційний технік з планера та двигунів
9	3115	Авіаційний технік з приладів та електроустаткування
10	3115	Авіаційний технік з радіоустаткування
11	8139	Автоклавник (виробництво скла та скловиробів)
12	8212	Автоклавник (виробництво теплоізоляційних матеріалів)
13	8154	Автоклавник (виробництво цитринової та виннокам'яної кислот)
14	8122	Автоклавник лиття під тиском

Рисунок 3.4 – Класифікатор професій

3.4 Підготовка даних

У процесі передобробки даних проводиться їх підготовка до аналізу, в результаті якої вони приводяться у відповідність до вимог, що визначаються специфікою завдання, що розв'язується.

Передобробка є найважливішим етапом Data Mining, і якщо вона не буде виконана, то подальший аналіз у більшості випадків неможливий через те, що аналітичні алгоритми просто не зможуть працювати або результати їхньої роботи будуть некоректними. Іншими словами, реалізується принцип GIGO - garbage in, garbage out (сміття на вході, сміття на виході).

Передобробка даних включає два напрями: очищення та оптимізацію.

Очищення проводиться з метою виключення різноманітних факторів, що знижують якість даних і заважають роботі аналітичних алгоритмів. Вона включає обробку дублікатів, протиріч та фіктивних значень, відновлення та заповнення перепусток, згладжування, придушення шуму та редагування аномальних значень. Крім цього, у процесі очищення відновлюються

порушення структури, повноти та цілісності даних, перетворюються некоректні формати.

Оптимізація даних як елемент передобробки включає зниження розмірності, виявлення та виключення незначних ознак. Основна відмінність оптимізації від очищення в тому, що фактори, що усуваються в процесі очищення, суттєво знижують точність розв'язання задачі чи роблять роботу аналітичних алгоритмів неможливою. Проблеми, які вирішуються при оптимізації, адаптують дані до конкретної задачі та підвищують ефективність їх аналізу.

Перший етап підготовки зібраних даних полягає у видаленні специфічних символів із заголовків вакансій. До таких символів відносяться знаки пунктуації, спеціальні символи, цифри та можливо логотип сайтів, що поміщені у назву вакансії.

	general_id	title	description
0	1001708134	менеджер (управитель) з постачання	Здійснює прийом товару, контроль якості та к...
1	1025025985	Мастер по ремонту телефонов, ноутбуков (ж/м Фр...	Компания «Твой Мастер» специализируется на обс...
2	973834515	Монтажник з монтажу сталевих та залізобетонних...	Виконує складні монтажні роботи при збиранні р...
3	1002450963	прибиральник службових приміщень	Повернутись\nNº вакансії: 20392202180099\n\nЯкщ...
4	973498359	Механік підвісних будівельних колисок в Вишнев...	Вход в аккаунт Войти Запомнить Забыли пароль? ...
...	...	...	...
243261	975098462	Sales-manager, контейнерные перевозки 28000 гр...	Вход в аккаунт Войти Запомнить Забыли пароль? ...
243262	999377591	менеджер (управитель) з постачання	Укладення договорів на постачання продукції, д...
243263	985600554	Водій-експедитор кат. С, С1 в Кременчуге - Гал...	Вход в аккаунт Войти Запомнить Забыли пароль? ...
243264	1015631299	водій автотранспортних засобів	Вернуться к поиску: [[backlinkLabel]]\nводій а...
243265	1007477983	В архиве с 27.02.2022.	Вход в аккаунт Войти Запомнить Забыли пароль? ...

243266 rows × 3 columns

Рисунок 3.5 – Приклад початкового датасету

Другий крок – це видалення стоп слів української мови. До стоп слів відносяться наприклад: а, аби, будь ласка, буває, дедалі, кожний, не, не можна і тд. Зазвичай це прийменники, вигуки, займенники та прислівники.

Третій крок – це пошук та видалення тих записів, де немає назва, але замість цього міститься стороння інформація. Наприклад у датасеті зустрічалися рядки, де замість назви була вказана заробітна плата.

Після наведених етапів отримуємо датасет з 4 колонок: ідентифікаційний номер, початкова назва, очищена назва, та індекс кроку класифікації, рис. 3.6.

	general_id	title	title_clean	StepMatched
0	1001708134	менеджер (управитель) з постачання	менеджер управитель постачання	0
1	1025025985	Мастер по ремонту телефонов, ноутбуков (ж/м Фр...	мастер ремонту телефонов ноутбуков фрунзенский	0
2	973834515	Монтажник з монтажу сталевих та залізобетонних...	монтажник монтажу сталевих залізобетонних кон...	0
3	1002450963	прибиральник службових приміщень	прибиральник службових приміщень	0
4	973498359	Механік підвісних будівельних колисок в Вишнев...	механік підвісних будівельних колисок вишнево...	0
...	...	...	...	...
243261	975098462	Sales-manager, контейнерные перевозки 28000 гр...	sales manager контейнерные перевозки грн одес...	0
243262	999377591	менеджер (управитель) з постачання	менеджер управитель постачання	0
243263	985600554	Водій-експедитор кат. С, С1 в Кременчуге - Гал...	водій експедитор кат кременчуге галактика пп	0
243264	1015631299	водій автотранспортних засобів	водій автотранспортних засобів	0
243265	1007477983	В архиве с 27.02.2022.	архиве	0

242653 rows × 4 columns

Рисунок 3.6– Приклад датасету після перших кроків очистки

Наступний етап – визначення мови за допомогою Python LangDetect.

LangDetect – найпопулярніша бібліотека Python, призначена для визначення мови. Вона одночасно швидка та точна. Вона може легко виявити кілька мов в тому самому тексті. Цю бібліотеку використано для заданого датасету. Оскільки як було описано раніше, вакансії представлені на онлайн сервісах пишуться на 3 різних мовах: українською, російською, англійською та іноді використовується комбінація цих мов в одній мові. Для подальшого аналізу буде використана наразі тільки українська мова, оскільки класифікатор професій написаний тільки на державній мові. В подальшому дослідженні можна використовувати інструменти автоматичного перекладу

для тої частини датасету, що написана на інших мовах. Але вакансії, назва яких мультилінгвальна доведеться виключити з дослідження.

На рис. Представлено екземпляр датасету після детекції мови. Далі необхідно відфільтрувати датасет тільки за українською мовою. Частина вакансій написаних державною мовою складає 38% від усього обсягу датасету. Російська мова 41%, та англійська 4%, залишок припадає на список мов, яку було визначені некоректно, саме через комбінації кількох мов у назвах. Цей відсоток не критичний, тому з цими даними можна працювати пізніше, для більш детальної обробки. Розмір початкового датасету це 243266 записів. Після першого етапу обробки стало 242653 записи. Записів на український мові 93093.

В даній роботі використовується не тільки датасет з даними про вакансії, але й класифікатор професій. Для використання у пайплайні програмного продукту його також потрібно було очистити за всіма етапами, що використані для вакансій. На рис 3.7 зображено приклад класифікатора професій після обробки.

	Code	Prof_job_title	name_clean
0	7346	Авербандник	авербандник
1	7232	Авіаційний механік з планера та двигунів	авіаційний механік планера двигунів
2	7241	Авіаційний механік з приладів та електроустатк...	авіаційний механік приладів електроустаткування
3	7243	Авіаційний механік з радіоустаткування	авіаційний механік радіоустаткування
4	7232	Авіаційний технік (механік) з парашутних та ав...	авіаційний технік механік парашутних аварійно ...

Рисунок 3.7 - дані класифікатора після обробки

Висновок до розділу 3

В третьому розділі магістерської дисертації було детально розглянуто засоби та спосіб збору даних для поставленої задачі. Приділено багато уваги проблематиці відбору онлайн ресурсів з пошуку роботи. Важливо заздалегідь визначити структуру і методологію, за якою збираються дані, аби в результаті

отримати узгоджені дані. Для цього треба розуміти організацію онлайн ресурсів та опубліковану на них інформацію. Детально описано розроблений алгоритм програми, що скачує інформацію онлайн. Розглянуто дані, що отримані за результатами роботи програмного забезпечення, їх специфіку та можливості їх використання. Остання частина цього розділу присвячена розбору датасету, визначення проблем та прогалин, що містять дані, очистці зібраних даних та підготовки їх для використання у системі класифікації.

РОЗДІЛ 4 СИСТЕМА КЛАСИФІКАЦІЇ ТА АНАЛІЗ РЕЗУЛЬТАТІВ КЛАСИФІКАЦІЇ ВАКАНСІЙ

4.1 Алгоритм розробленого класифікатора

Задача магістерської дисертації спрямована на розробку та налаштування інструментів, необхідних для класифікація вакансій за професіями

Для класифікації професій було розроблено гібридний підхід, щоб покращити ефективність класифікації та зменшити необхідні мануальні зусилля. Завдання також спрямоване на вирішення всіх лінгвістичних специфічних проблем, забезпечуючи унікальну структуру для сприяння ефективній класифікації. Щоб зменшити втручання людини в процес класифікації, були прийняті різні методології штучного інтелекту, що включають методи обробки природної мови. Процес також розроблятиме оригінальну онтологію, яка розвиватиметься з часом разом із розвитком проекту.

Фреймворк використовує гібридний підхід для поєднання потужності онтологій і словників із гнучкістю алгоритмів машинного навчання (рис. 4.1)

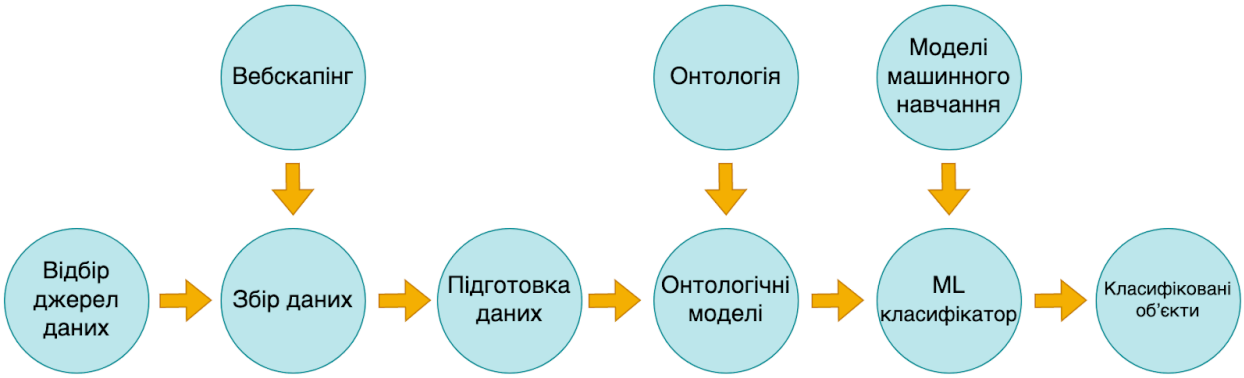

Рисунок 4.1 – Алгоритм класифікатора

Розроблена система спочатку намагається класифікувати вакансію шляхом пошуку відповідності термінів, що містяться в онтології

(класифікаторі), метаданих або в описі вакансії; згодом, якщо результату не було отримано, намагається класифікувати його за допомогою алгоритму машинного навчання.

Кожна оброблена вакансія проходить наступні етапи для отримання інформації. Усі кроки застосовуються за допомогою моделі водоспаду: якщо крок дає один або кілька результатів, вакансія класифікується, а конвеєр зупиняється, інакше алгоритм переходить до наступного кроку.

Визначення мови та попередня обробка: ці кроки, детально описані в попередніх розділах, ідентифікують мову вакансії і готують текст для наступних кроків

1. Моделі на основі онтології: на цьому кроці намагається класифікувати вакансію відповідно до термінів, що містяться у відповідній онтології. Він розділений на послідовність упорядкованих підкроків, від найбільш точних до менш конкретних, які виконуються один за одним, лише якщо попередній не дає результатів:

– пошук точного збігу термінів, що містяться в онтології та назві вакансії;

– пошук відповідності схожості термінів, що містяться в онтології, в назві вакансії;

– пошук за точним збігом однокореневих термінів у назві вакансії;

– пошук відповідності схожості однокореневих термінів у назві вакансії;

2. Класифікатор машинного навчання: після відпрацювання моделей онтології, які дають зазвичай велику кількість збігів, попередньо навчений алгоритм машинного навчання застосовується для класифікації вакансії відповідно до навчального набору.

Наприкінці процесу вакансія класифікується, якщо один із описаних раніше кроків дав принаймні один результат. Зазвичай кожна вакансія класифікується один раз за кожним атрибутом, таким чином зупиняючи конвеєр, як тільки досягається результат.

Техніка відповідності (match), подібності (similarity) та стемінг.

Під час оцінювання подібності між онтологією та термінами, що містяться у вакансіях, було оцінено багато підходів (додаткову інформацію про оцінені підходи див. (Bharti та ін., 2017) «Automatic Keyword Extraction for Text Summarization: A Survey» (Gomaa and Fahmy, 2013).) «A Survey of Text Similarity Approaches», (Peng та ін., 2012) «A Comparison of Techniques for Name Matching»):

– Підхід, заснований на рядках (string based approach): міри подібності рядків працюють на послідовності рядків і композиції символів (подібність Джаро-Вінклера, Жаккара, косинуса);

– Підхід, заснований на корпусі: це міра семантичної подібності, яка визначає подібність між словами відповідно до інформації, отриманої з великих корпусів (латентний семантичний аналіз, явний семантичний аналіз, розподіллено схожі слова з використанням матриць співпадінь);

– На основі знань: ґрунтується на визначенні ступеня подібності між словами за допомогою інформації, отриманої із семантичних мереж. У моделях на основі онтології подібність Жаккара була обрана як краща.

4.2 Результати роботи алгоритму класифікації

В даній роботі розглядається задача багатокласової класифікації. Розглянемо для початку зразок класів, до яких необхідно віднести кожну зібрану вакансію. Класи – перелік професій у Державному класифікаторі професій, їх кількість зазначена у даному класифікаторі є 8995, тобто стільки ж класів буде мати на вихід алгоритм класифікації. Кожна вакансія буде віднесена одній з категорій професій. На рисунку 4.2 наведено зразок професій та відповідний до них код.

8139	Фритівник (виробництво керамічних, фарфорових та фаянсових виробів)
8152	Фритівник (емалювання)
7436	Фурнітурник
7122	Футерувальник (кислототривник)
7122	Футерувальник-муляр
7122	Футерувальник-шамотник на ремонті ванн
7432	Футлярник
7412	Халвомісильник
8131	Хальмівник
8212	Хвилювальник азбестоцементних листів
2113.2	Хімік

Рисунок 4.2 – Професії та відповідний до них код

Оскільки в даній роботі використовуються сирі нові дані, вони не мають ніяких заздалегідь отриманих міток приналожності до класів. Тому використовувати навчання з учителем на першому етапі немножливо. Комплексний підхід до класифікації полягає в тому, що ми використовуємо семантичні порівняння для отримання навчального датасету, що далі можна використовувати для навчання з учителем. З іншого боку підходи та алгоритми, що призначені для роботи з живою мовою передбачають навчання без учителя. Тому стає можливим скомбінувати два підходи та порівняти результати.

Після проведення перелічених вище етапів, було отримано позитивні проміжні результати. До прикладу на рисунку 4.3 відображено хмару слів, що допомагає зрозуміти, які семантичні конструкції найчастіше зустрічаються в тексті (обрано категорію інформаційно-комунікаційні технологій згідно результатів класифікації).

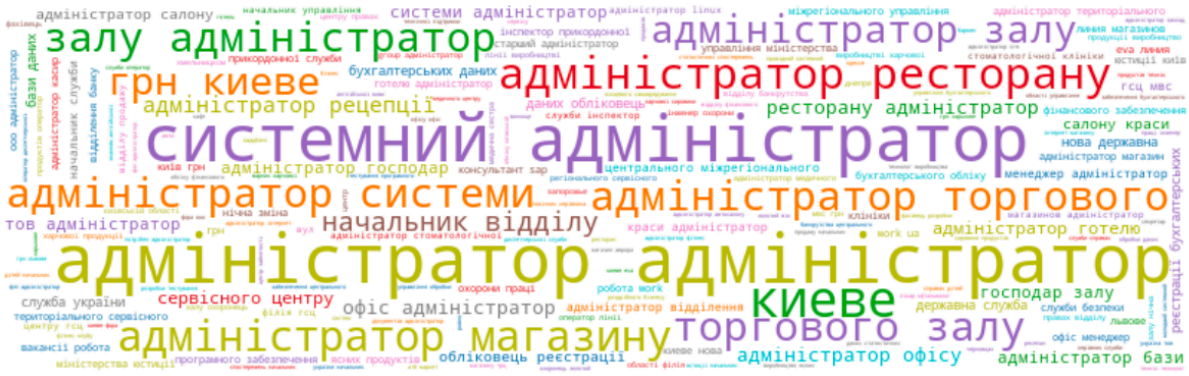

Рисунок 4.3 – Хмара слів за рівнем частоти згадування у тексті

Результати класифікації наведені у таблиці 4.1.

Таблиця 4.1. Відсоткове співвідношення співпадінь назви вакансії та назви професії в класифікаторі згідно з кожним методом перевірки

№	Метод визначення семантичної подібності	Відсоткове відношення співпадінь
1	Exact match	12%
2	Direct match	65%
3	Regular expression match	3%
4	FastText match (unsupervised)	16%

Згідно з наведених результатів можна відмітити, що значну частину вакансій можна класифікувати різними комбінаціями порівнянь назв, але в будь-якому разі завжди залишаться ті назви вакансій, що ніяким чином не узгоджені з класифікатором професій. Саме тому є сенс використовувати методи машинного навчання для досягнення повноцінного результату. Надалі та частина класифікованих вакансій може бути використана як навчальна вибірка для методів навчання з учителем. Це розширить спектр алгоритмів, що можна застосувати та дасть змогу порівняти результати. Навчання без учителя в алгоритмі FastText знаходить смислові схожості, використовуючи word embedding для обчислення косинусної відстані між закодованим словами. Таким чином є можливість встановити схожість між шматками тексту, навіть написані різним лексиконом.

Похибка класифікації складає 4%. В даному випадку похибка використовується для позначення некласифікованих вакансій. Головна причина того, що деякі вакансії не можуть бути класифіковані полягає у тому,

що на ринку кожного дня з’являються нові професії та нові формулювання. Оскільки класифікатор професій опубліковано останній раз у 2010 році, він є доволі застарілим, особливо якщо розглядати сферу інформаційних технологій. Тому завжди буде відсоток вакансій, які не можуть бути віднесені до відповідного коду у класифікаторі згідно з встановленим порогом подібності, що складає 0.95%. Це саме і є одна з цілей розробки класифікатора вакансій, аби мати змогу актуалізувати та доповнювати класифікатор новими професіями, що представлені на ринку.

У таблиці 4.2 наведені результати класифікації алгоритмом навчання нейронної мережі без учителя у процентному співвідношенні відповідно до коду класифікатора професій.

Таблиця 4.2. відсоткове співвідношення класифікації FastText (unsupervised).

Назва групи класифікатора професій	Код групи	Кількість віднесених вакансій	Відсоток віднесених вакансій
Законодавці, вищі державні службовці, керівники, менеджери (управителі)	1	1191	0.08
Професіонали	2	2681	0.18
Фахівці	3	2979	0.2
Технічні службовці	4	744	0.05
Працівники сфери торгівлі та послуг	5	2532	0.17

Кваліфіковані робітники сільського та лісового господарств, риборозведення та рибальства	6	149	0.01
Кваліфіковані робітники з інструментом	7	1788	0.12
Робітники з обслуговування, експлуатації та контролювання за роботою технологічного устаткування, складання устаткування та машин	8	2085	0.14
Найпростіші професії	9	744	0.05

Другий підхід класифікації полягає у використанні попередньо класифікованої частини вакансій методами лексичних порівнянь в якості тренувального набору даних. У таблиці 4.3 наведена кількість класифікованих вакансій таким підходом з різними порогами відсікання для значення косинусної подібності. Базовий поріг був встановлений 0,9.

Таблиця 4.3. Результати класифікації FastText (supervised)

Кількість співпадінь	Загальна кількість вак.	Відсоткове відношення	Threshold
8444	14895	56.69	0.9
8931	14895	59.96	0.8
9230	14895	61.97	0.7
9592	14895	64.4	0.6
9900	14895	66.47	0.5

Відповідно до наведених результатів класифікація навчанням з учителем не дає гарних результатів. Це легко пояснити тим, що датасет тренувальних та тестовий не однорідні, не зважені та мають рівномірного представлення тих семантичних структур, що є у другій частині датасету. Це відбулося тому, що лексичними співпадіннями було знайдено такі назви вакансій, які не мають невизначеностей та не містять в собі тих специфічних слів та змістів, що є у вакансіях з невизначеністю. Відповідно нейронна мережа мала не достатньо інформації та прикладів, щоб навчитися розпізнавати вакансії з невизначеностями.

Приклади класифікованих вакансій на кожному етапі системи наведено у таблиці 4.4.

Таблиця 4.4. Приклади класифікованих вакансій

Назва кроку класифікації	Назва вакансії	Назва професії	Код професії
Exact match	Фахівець розробки тестування програмного забезпечення	Фахівець розробки тестування програмного забезпечення	3121
Direct match	Начальник служби	Начальник медичної служби	1232
Regular expression match	Адміністратор	Адміністратор рецепції новояворівськ	4222
FastText (unsup.)	Касир торговельного залу	Продавець магазину продуктів	4211

Висновок до розділу 4

Було реалізовано систему збору, обробки та класифікації вакантних робочих місць на ринку праці України. Детально описані етапи алгоритму класифікації та викладені причини використання саме цих підходів. Головна ідея алгоритму це використання комбінації підходів для отримання найкращого результату. Спершу було використано синтаксичне порівняння, тобто онтологічні моделі. Цей етап логічний згідно до задачі та допоможе отримати тренувальний набір даних, оскільки дані зібрані власноруч і вони не

розмічені експертом попередньо. Таким чином на другому етапі, де використовуються нейронні мережі є змога використати два типи навчання нейронних мереж: навчання з учителем та без учителя. Завдяки цьому ми отримали різні результати класифікації і можемо порівняти їх. Результатом класифікації є датасет, де кожній з вакансій поставлено у відповідність код з класифікатора професій. Також розглянуто питання похибки або залишку вакансій, які не були класифіковані. Далі отриманий датасет необхідно передати на оцінку експерта у цій галузі для смислової перевірки коректності розподілення міток. Надалі датасет можна використовувати для подальшої аналітики.

РОЗДІЛ 5 РОЗРОБКА ВЛАСНОГО СТАРТАП ПРОЕКТУ

Сучасні інформаційні та цифрові технології інтегрувалися чи не в кожну сферу нашого життя. Найбільше різноманітні методи автоматизації використовуються у бізнесі та сфері інформаційних технологій. Наукові галузі, такі як комп'ютерна та інженерна, також використовують та найголовніше дають нові інструменти та підходи для вирішення все більш різноманітних завдань. Проте досі такі більш гуманітарні та соціальні напрями залишаються мало або взагалі не оснащеними новітніми інформаційними інструментами. Головна перепона – це те, що в цій галузі не достатньо людей з потрібною освітою аби імплементувати та використати інформаційні технології у прикладних задачах гуманітарних напрямів, демографії та соціології до прикладу. Це в свою чергу дало б змогу науковцям мати повніше та детальніше представлення про ті чи інші соціальні явища. Основним напрямком, де необхідне використання інформаційних технологій – це збір нетипових даних та їх обробка.

У цій роботі розглядається реальна проблема, з якою стикаються науковці і дослідники ринку праці України. Проблема полягає у тому, що інформаційне забезпечення ринку праці не повноцінно покриває цю сферу та не дає змогу скласти актуальне представлення та оцінити ситуацію на ринку праці. Головний чинник цього – відсутність зібраної, генералізованої та класифікованої інформації про поточні актуальні вакантні робочі місця в Україні.

Мета стартапу – створити уніфіковану систему для збору та класифікації вакансій, що публікуються на українських онлайн ресурсах.

Проблема, якій присвячено цей стартап є надзвичайно актуальною та корисною для розвитку української науки у галузі демографії, зокрема для дослідження ринку праці України.

5.1 План розробки стартапу та масштабування його на ринок

Наведемо план розробки стартапу та виведення його на ринок.

Спочатку треба провести маркетинговий аналіз, який включає в себе:

- конкурентний аналіз, щоб зрозуміти, якими методами вирішення проблем вже користуються люди;
- формування ідеї самого проекту та виділення цільової аудиторії;
- розробити стратегію виведення товару на ринок, базуючись на аналізі ринкового середовища.

Наступним кроком являється організація самого стартапу. На цьому етапі мають бути:

- складений весь план та побудований таймлайн розробки та запуску продукту;
- запланований обсяг виробництва та оцінений потенційний обсяг ресурсу, який буде потрібен для виконання плану;
- розраховані витрати, необхідні для реалізації проекту, та витрати на запуск проекту.

Далі необхідно виконати фінансово-економічний аналіз та оцінити ризики стартап-проекту, в межах якого:

- визначити обсяг інвестиційних втрат;
- розрахувати основні фінансово-економічні показники проекту (собівартість, ціну продукту/послуги, податковий збір та чистий прибуток) та визначити показники інвестиційної привабливості проекту (рентабельність продажів, період окупності проекту);
- визначити основні ризики проекту та способи для їх запобігання.

Фінальним кроком являється розробка заходів з комерціалізації продукту. Цей крок являється важливим для масштабування та збільшення розмірів продукту.

Для того, щоб залучити інвесторів та знайти різні способи фінансування проекту, необхідно:

- провести дослідження на предмет інтересів потенційних інвесторів та бізнесів;
- скласти інвестиційну пропозицію, яка включає в себе як опис самого продукту та його теперішні розміри, так і можливі шляхи розширення та розвитку;
- обрати канали комунікації із потенційно зацікавленими персонами.

Далі наведемо результати виконання кожного з описаних кроків.

5.2 Опис ідеї стартап-проекту

Стартап-проект полягає у вирішенні проблеми збору та класифікаціх українських вакансій. Суть продукту стартапу полягає у тому, що розроблена система збирає актуальні пропозиції вакантних робочих місць, що опубліковані на порталах з пошуку роботи, оброблює ці дані, підготовлює до використання методів класифікації та власне класифікує зібрані вакансії згідно з державним класифікатором професій.

У таблиці 5.1 наведена інформаційна карта стартапу.

Таблиця 5.1 – Інформаційна карта стартап-проекту

Назва проекту	VacancyClassifier
Автор проекту	Цимбал Юлія Олександрівна
Коротка анотація	Продукт буде збирати, підготовлювати та класифікувати вакансії

Термін реалізації проекту	12 місяців
Необхідні ресурси	Приміщення з комп'ютерами, доступом до Інтернету, доступ до електромережі. Програмне забезпечення для розробки, хмарне сховище для даних, антивірусні програми. Фінансові кошти на оплату заробітної плати виконавцям на термін 12 місяців, а також на такі витрати як: оренда приміщення, комунальні послуги, оренда хмарного сховища тощо.
Опис проблеми, яку вирішує проект	Продукт вирішує задачу класифікації вакансій, що представлені на українському ринку пошуку роботи.
Головні цілі та завдання проекту	Метою проекту є створення системи, яка буде автоматично збирати, обробляти та класифікувати вакансії.
Очікувані результати	Реалізована система та залучення інших департаментів освіти і науки для її використання.

5.3 Технологічний аудит ідеї проекту

Тепер можна розібрати ідею стартапу та провести конкурентний аналіз. У таблиці 5.2 наведений опис ідеї стартапу.

Таблиця 5.2 – Опис ідеї стартапу

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Основна ідея полягає у створенні комплексної системи, яка буде збирати, обробляти та класифікувати текстові дані про вакансії	Заповнення прогалин у інформаційному забезпеченні ринку праці та автоматизація для подальших наукових досліджень, що можуть базуватися на зібраних даних	В даному випадку користувачі – науковці. Для них це спрощення процесу збору та аналізу даних та автоматизація для зменшення часових ресурсів та людських.

Порівняльний аналіз конкурентів проекту провести наразі неможливо, оскільки в Україні поки що не існує подібного інструменту.

Далі аналізуємо реальність технічно здійснити ідею проекту (таблиця 5.3).

Таблиця 5.3 – Технологічна здійсненність продукту

№ п/п	Ідея проекту	Технології і реалізації	Наявність технологій	Доступність технологій
1	Створення комплексної системи, яка буде збирати, обробляти та класифікувати вакансії.	Використання мови програмування Python	Наявні	Доступні
2		Використання мови програмування C#	Не наявні, необхідні доопрацювання	Доступні

3		Використання мови програмування R	Наявні, необхідні доопрацювання	Доступні
Обрана технологія реалізації ідеї проекту: Python				

5.4 Аналіз ринкових можливостей запуску стартап-проекту

Далі проведемо попередній аналіз ринку для запуску стартап-проекту (таблиця 5.4).

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники ринку (найменування)	Характеристика
1	Кількість головних гравців, од	4
2	Загальний обсяг продаж, грн/ум.од	5000
3	Динаміка ринку (якісна оцінка)	Позитивна, зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Відсутні
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні

6	Середня норма рентабельності в галузі (або по ринку), %	14%

Тепер проведемо характеристику потенційних клієнтів, які можуть бути зацікавлені в проекті (таблиця 5.5).

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреби, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Аналіз інформації про опубліковані вакансії	Держава, використання науковцями у сфері демографії	Переслідують різні цілі використання готової бази даних	Простота та автоматизація використання
2	Сформулювати нові види професій та доповнити КП на базі актуальних запитів роботодавців	Держава, науковці	Переслідують різні цілі використання готової бази даних	Автоматизація та експертна оцінка

Обрахуємо фактори загроз (таблиця 5.6) та можливостей (таблиця 5.7). Проаналізуємо загрози, щоб зрозуміти можливі перешкоди при запуску

продукт на ринок. Фактори можливостей же треба обрахувати, щоб знати усі сприятливі умови та по можливості ними скористатися.

Таблиця 5.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Хоча ринок є відкритим і неосвоєним, можуть швидко з’явитися продукти, розроблені відомішими компаніями чи зроблені на конкретне замовлення	Знайти точки додаткової цінності для користувача
2	Ціна збуту	Небізнесова сфера збуту може бути не достатньо лояльною та пропонувати занизьку ціну	Сфокусуватися на якості роботи продукту та продумати маркетингову стратегію
3	Якість аналізу	Оскільки аналіз вакансій зараз виконується тільки для назв, які написані українською мовою, користувачі може мати запит розширювати можливості продукту для узагальнення та незалежності від мови	Мати достатній штаб і ресурси, для побудови різних моделей для різних текстових корпусів

Таблиця 5.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Автоматизація	Продукт дає змогу замістити вели обсяг мануальної роботи	Зробити акцент при маркетингу, продовжувати розвиток як окремого продукту
2	Якість фінального результату	Висока точність класифікатора дає змогу одразу переходити до подальшого наукового аналізу	Реалізувати зручний інтерфейс для завантаження
3	Швидкість та простота	Етап збору даних займає деякий час, але час класифікації зібраної інформації дуже короткий, а сам класифікатор простий у використанні	Пропонувати моделі з найкращими результатами, а також надавати усю необхідну технічну підтримку

Далі розглянемо питання конкуренції, а саме визначимо її тип та рівень (таблиця 5.8).

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)

1. Вказати тип конкуренції: недосконала конкуренція	Не представлено продуктів та експертів	Зробити максимальним збут застосунку
2. За рівнем конкурентної боротьби: міжнародний	Наявні проекти, розроблені та можуть бути доступні у всьому світі	Розширити цільову аудиторію, розробити варіант обробки вакансій на різних мовах
3. За галузевою ознакою: внутрішньогалузева	Не можуть працювати з різними галузями	Покращити персоналізацію
4. Конкуренція за видами товарів: товарно-родова	Конкуренція з аналізами інших систем та експертів	Підтримувати та покращувати якість існуючих функцій
5. За характером конкурентних переваг: нецінова	Компанії не пропонують різну якість	Розробляти якісніші алгоритми і моделі
6. За інтенсивністю: немарочна	Не представлені компанії із сильним брендом	Предметно створити комунікаційну стратегію для свого бренду

Далі необхідно виконаємо аналіз конкуренції за моделлю 5 сил конкуренції Майкла Портера (таблиця 5.9).

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти у галузі	Потенційні конкуренти	Постачальники	Клієнти	Товарозамінники
	Наразі конкуренті в на ринку не представлено	Якість, ціни, кількість користувачів, капіталовкладення	Фактори сили постачальників	Контроль якості, порівняння цін	Сила бренду, якість, ціна, масштаби
Висновки	Конкуренція з мінімально інтенсивністю, а також запит користувачів на ринку	Можливості входження на ринок, нові потенційні конкуренти	Постачальники відсутні	Клієнти не диктують умови роботи на ринку	Товаро замінники відсутні

Маючи результати аналізу конкуренції (таблиця 5.9), характеристики ідеї стартап-проекту (таблиця 5.4), характеристики потенційних клієнтів і їх вимоги до продукту (таблиця 5.5) та фактори ринкового середовища (таблиці 5.6 і 5.7) було сформульовано та обґрунтовано перелік факторів конкурентоспроможності (таблиця 5.10).

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Автоматизація	Продукт дає змогу замістити вели обсяг мануальної роботи
2	Якість фінального результату	Висока точність класифікатора дає змогу одразу переходити до подальшого наукового аналізу
3	Швидкість та простота	Етап збору даних займає деякий час, але час класифікації зібраної інформації дуже короткий, а сам класифікатор простий у використанні

Тепер можна провести аналіз сильних та слабких сторін продукту (таблиця 5.11).

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін системи

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів					
			-3	-2	-1	0	1	2
1	Автоматизація	20	+					
2	Якість фінального результату	17	+					
3	Швидкість та простота	10	+					

Далі проведемо SWOT-аналіз продукту (таблиця 5.12).

Таблиця 5.12 – SWOT-аналіз стартап-проекту

Сильні сторони Автоматизація Якість фінального результату Швидкість та простота	**Слабкі сторони** Відсутність сильного бренду Не сформована база клієнтів Не підключені альтернативні канали маркетингу
Можливості Покращення системи Розширення можливостей класифікації вакансій на різних мовах Інтеграція з науковими центрами	**Загрози** Нові системи та експерти Збут

Дякуючи проведенню SWOT-аналізу, ми змогли визначити сильні та слабкі сторони, можливості та загрози, пов'язані з конкуренцією та плануванням стартап-проекту. Далі спроектуємо альтернативну ринкову поведінку для інтеграції стартап-проекту на ринок та приблизний час реалізації системного комплексу, з урахуванням потенційних проектів, що можуть бути виведені на ринок та наведемо результати у таблиці 5.13.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Вихід на ринок з нижчою якістю	70%	3 місяці
2	Пропонувати одразу платне використання	80%	4 місяців

3	Представлення користувачам системи без інтерфейсу	40%	7 місяці

У даному пункті був проведений детальний аналіз ринку та продукту. Також відповідно до результатів проведеного конкурентного аналізу, визначених факторів ринку та його сприятливість, описання ідеї та характеристик стартап-проекту, робимо висновок, що існують дуже сприятливі умови для виходу продукту на ринок.

5.5 Розроблення ринкової стратегії стартап-проекту

Для розробки ринкової стратегії продукту, у першу чергу, необхідно проаналізувати цільову аудиторію проекту (таблиця 5.14).

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит у межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Держава, дослідницькі центри	Висока	70%	Низька	Середня
2	Великі бізнеси	Низька	10%	Низька	Середня

3	Малі та середні бізнеси	Низька	10%	Низька	Висока
4	Приватні користувачі	- (продукт не призначено до використання приватними користувачами)	-	-	-
Які цільові групи обрано: 1					

Маючи аналіз цільових груп, далі визначимо базову стратегію розвитку продукту (таблиця 5.15).

Таблиця 5.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Постійне оновлення і покращення продукту	Ринкове позиціонування на представників державної діяльності	Масштабування та максимізація	Оптимальних витрат

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиці 5.16, 5.17).

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
Так	Так	Ні	Заповнення конкурентної ніші

Таблиця 5.17 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Універсальність Простота у використанні Якість результатів	Оптимальних витрат	Автоматизація Якість результатів Швидкість та простота	Система, яка автоматично збирає та класифікую вакансії з веб мережі

5.6 Розроблення маркетингової програми стартап-проекту

Після проведеного комплексного аналізу, можемо повноцінно описати ключові переваги концепції потенційного товару (таблиця 5.18) та побудувати концепцію маркетингових комунікацій (таблиця 5.19).

Таблиця 5.18 – Ключові переваги концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Швидкий збір даних та їх аналіз	Автоматизація мануальних процесів	Постійне покращення та перенавчання моделей, які зможуть детальніше класифікувати
2	Простота у використанні	Система є максимально спрощеною для використання	Такого виду систему може використовувати користувач, після мінімального інструктажу

Таблиця 5.19 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються	Ключові позиції, обрані для позиціонування	Завдання рекламно о повідомлення	Концепція рекламного звернення

		цільові клієнти			
1	Пошук спеціалізованих систем	Державні замовлення розробних або дослідницьких проектів	Автоматизація Простота Швидкість Якість	Поєднати повідомлення про те, що це якісна система, яка є незалежною	Пропозиції співпраці з науковими департаментами

Висновки до розділу 5

Даний розділ був присвячений дослідженню стартап-проекту. В якості такого була представлена система збору та класифікації вакансій згідно з Класифікатором професій.

У рамках розділу було досліджено розробку стратегій виходу на ринок та маркетинг-стратегії для цього. Зокрема, даний ринок являється сприятливим та без компаній конкурентів. Запропонована система є універсальною та доступною, то у стартап-проекту є всі шанси стати монополістами на ринку.

Також були опрацьовані сильні та слабкі сторони проекту, SWOT аналіз, аналіз конкурентів та цільової аудиторії. На основі всіх досліджень був сформований концепт маркетингової стратегії для обраних цільових аудиторій.

ВИСНОВКИ

Технологія видобутку інформації з великих даних та відповідні методи її застосування розширюють можливості моніторингу й аналізу ринку праці, оцінки його динаміки своєчасно (майже в реальному часі), індуктивно, тобто використовуючи дані для формулювання та підтвердження гіпотез, і на дуже детальному рівні. Сучасна парадигма інформаційної системи ринку праці повинна випливати з необхідності інтеграції різних типів даних та поєднання як класичних, так і інноваційних методів роботи з такими даними.

Аналіз світового досвіду засвідчує можливість використання великих даних для дослідження в реальному часі широкого кола проблем функціонування ринку праці: від оцінки загальної онлайн пропозиції робочих місць і їх класифікації за професіями та видами діяльності, до більш специфічних питань таксономії навичок, виявлення нових професій та розробки модельних траєкторій побудови кар'єри.

Встановлено, що в Україні існує дві основні категорії джерел даних для забезпечення оцінки кількості та структури робочих місць: офіційні й альтернативні. Офіційні джерела – це дані офіційної статистики, адміністративні дані та державні моніторинги. Вони збираються з конкретною, заздалегідь визначеною метою, передбачають потрібне охоплення населення, чіткі визначення, методологію, якість і часові рамки, щоб задовольнити аналітичні потреби певних зацікавлених осіб; здебільшого бувають структурованими та легко піддаються зберіганню за допомогою класичних реляційних парадигм. Водночас їхніми суттєвими обмеженнями є агрегація та жорстка структурованість, що унеможливлює відстеження важливих деталей і зв'язків. Альтернативні джерела представлені незалежною аналітикою, маркетинговими дослідженнями та інтернет-даними. Ці дані безперервно генеруються різними джерелами. Їхня якість часто залежить від здатності користувача виявляти їх особливості, а також від охоплення, яке повинне бути розраховане та виміряне користувачем, нерідко для поєднання великої кількості різних джерел даних.

Процедура відбору джерел даних для збору інформації про онлайн пропозицію робочих місць передбачає застосування методики ранжування на основі набору критеріїв, серед яких рейтинг частоти згадування джерела в пошукових системах, достовірність даних про вакансії, наявність дати публікації вакансії, інформативність опису вакансії. Для збору та приймання даних із відібраних джерел в Україні найбільшу ефективність за співвідношенням показників продуктивності та вартості фактичних і часових витрат продемонструвало спеціалізоване програмне забезпечення вебскрапінгу на мові Python. Важливим етапом формування бази даних про пропозицію робочих місць є первинна обробка даних, отриманих у результаті вебскрапінгу, яка передбачає усунення помилок, «шумів», дублювань тощо, а також уніфікацію даних відповідно до вибраних класифікаторів.

В роботі розглянуто основні найвідоміші методи штучного інтелекту для обробки текстової інформації. Зараз ця область технологій стрімко розвивається, оскільки даних у форматі живої мови безліч у будь-якій інформаційній системі. NLP дозволяє краще розуміти запити користувачів та аналізувати корпоративну інформацію. Використання даної технології гарантує, що у кожного користувача є доступ до найбільш актуальних, корисних джерел інформації, які б інакше залишилися прихованими у величезних обсягах даних. Неодмінна складова обраної області штучного інтелекту це дистрибутивна семантика. За час проведення дослідження було детально розглянуто принципи та підходи для роботи з неструктурованим текстом. Загалом було обрано методи, що найкраще підходять для реалізації системи класифікації за критеріями ефективності, швидкості та зручності використання існуючих модулів.

Розроблено спеціальний алгоритм класифікації спираючись на специфіку зібраних даних на онлайн ресурсах пошуку роботи. Етапи та причини використання саме таких кроків були розкриті у роботі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Верховна рада України Закон Про державну статистику URL: https://zakon.rada.gov.ua/laws/show/2614-12#Text (дата звернення: 03.09.2022)

2. Державна служба статистики України URL: https://ukrstat.gov.ua/ (дата звернення: 05.09.2022)

3. Робоча сила України 2021 / Статистичний збірник. Держстат України. – Київ, 2022. – 216 с.

4. Державна служба зайнятості URL: https://www.dcz.gov.ua/ (дата звернення: 07.09.2022)

5. Верховна рада України Закон Про зайнятість населення URL: https://zakon.rada.gov.ua/laws/show/5067-17#Text (дата звернення: 08.09.2022)

6. Верховна рада України Закон Про рекламу URL: https://zakon.rada.gov.ua/laws/show/270/96-%D0%B2%D1%80#Text (дата звернення: 10.09.2022)

7. Harmonized with ISCO-88: International Standard Classification of Occupations/ILO, Geneva.

8. Harmonized with Statististical classification of economic activities in the Europeenne Communaute/ Nomenclature statistique des Activites economiques dans la Communaute Europeenne (NACE Rev.2).

9. Національні (державні) статистичні класифікації (класифікатори) URL: https://ukrstat.gov.ua/work/klass200n.htm (дата звернення: 15.09.2022)

10. Верховна рада України Закон Про стандартизацію URL: https://zakon.rada.gov.ua/laws/show/1315-18#Text (дата звернення: 18.09.2022)

11. Саріогло В.Г. "Великі дані" як джерело інформації та інструментарій для офіційної статистики: потенціал, проблеми, перспективи. Статистика України. - 2016. - № 4. - С. 12-19. URL: http://nbuv.gov.ua/UJRN/su_2016_4_5 (дата звернення: 18.09.2022)

12. Саріогло В.Г. "Великі дані" як джерело інформації та інструментарій для офіційної статистики: потенціал, проблеми, перспективи. Статистика України. - 2016. - № 4. - С. 12-19. URL: http://nbuv.gov.ua/UJRN/su_2016_4_5 (дата звернення: 19.09.2022)

13. Професійний склад зареєстрованих безробітних та вакансій у 2019 році та на 1 січня 2020 року URL: https://www.dcz.gov.ua/sites/default/files/infofiles/1._profesiynyy_sklad_bezrobitnyh_ta_vakansiy_2019.xlsx (дата звернення: 19.08.2022)

14. Саріогло В. Г. Проблеми комплексного використання соціально-демографічних даних. Демографія та соціальна економіка. 2004. № 1—2. С. 37—44.

15. Шарнін М. М., Сомін Н. В., Кузнєцов І. П., Морозова Ю. І., Галина І. В., Козеренко Є. Б. Статистичні механізми формування асоціативних портретів предметних областей на основі природно-мовних текстів великих обсягів для систем здобуття знань // Інформатика та її застосування: журнал. - 2013. - Т. 7, вип. 2. - С. 92-99.

16. Schutze H. Dimensions of meaning // Proceedings of Supercomputing'92. — 1992. — С. 787—796.

17. Mikolov, Tomas; et al. (2013). "Efficient Estimation of Word Representations in Vector Space"

18. Xin Rong. “word2vec Parameter Learning Explained” URL: https://arxiv.org/pdf/1411.2738.pdf (дата звернення: 23.09.2022)

19. TF-IDF URL: https://ru.wikipedia.org/wiki/TF-IDF (дата звернення: 06.10.2022)

20. Sentiment analysis URL: https://en.wikipedia.org/wiki/Sentiment_analysis (дата звернення: 09.10.2022)

21. NLTK URL: https://www.nltk.org/ (дата звернення: 11.10.2022)

22. SpaCy Facts and Figures URL: https://spacy.io/usage/facts-figures (дата звернення: 12.10.2022)

23. Gensim URL: https://radimrehurek.com/gensim/ (дата звернення: 18.10.2022)

24. P. Bojanowski, E. Grave, A. Joulin, T. Mikolov. Enriching Word Vectors with Subword Information URL: https://arxiv.org/abs/1607.04606 (дата звернення: 19.10.2022)

25. A. Joulin, E. Grave, P. Bojanowski, T. Mikolov. Bag of Tricks for Efficient Text Classification URL: https://arxiv.org/abs/1607.01759 (дата звернення: 20.10.2022)

26. Web scraping URL: https://en.wikipedia.org/wiki/Web_scraping (дата звернення: 25.10.2022)

27. Selenium project description URL: https://pypi.org/project/selenium/ (дата звернення: 27.10.2022)

ДОДАТОК А ЛІСТИНГ ПРОГРАМНОГО ПРОДУКТУ

```
# Libraries needed
from typing import DefaultDict
import pandas as pd
from collections import Counter
import csv, codecs
import string
import time
from tqdm import tqdm
from rapidfuzz import process, fuzz
from multiprocessing import Pool
import re
import multiprocessing
import pickle
import logging
import numpy as np
import gc
import nltk
import string
import sys
import multiprocessing
from multiprocessing import Pool
from tqdm import tqdm
from pandarallel import pandarallel
import fasttext
import warnings
warnings.filterwarnings("ignore")

#Automatic garbage collection
#gc.enable()
gc.disable()

# Test for multiproc available
print('Cpus available: ', + multiprocessing.cpu_count())

# Set initial variables
lang = 'it'
ListLevel = '3' # (5 = classificazione Esco >= 5, 4 = solo 4, 3 = idem, 2 = idem , 1 =
idem) cambia parametro per scalare nel più generale
Run = 2

if lang!= 'en':
    country = lang.upper()
else:
    country = 'UK'
year = '21'
print('Country: ', country, ', year: ', year)

# Tries and fuzzy parameters
Threshold_Tries_Lower = 0.75
Threshold_Tries_Upper = 1.35
```

```
Threshold_Fuzzy = 94.0

#Fasttext parameters
ndimensions = 100
Threshold_FastText = 0.92

# Variables for load input files process
myroot = "/home/giabelli/5d_classification/Data/"
myrootCommon = myroot + 'Common/'
myrootResultsUnique = myroot + 'Pipeline_' + country + '/'

# Performance indicator
StartTimeEntireTask = time.time()

if Run == 0:
    corpus = myrootCommon + "title_" + country + "_" + year + "_selected.csv"
    # Import corpus JA
    df_title = pd.read_csv(corpus, on_bad_lines ='skip', engine='python',
dtype={'StepMatched':'str'})
    df_title = df_title[~df_title.title_clean.isna()]
    # df_title = df_title.iloc[2000:3000,:]
    df_title['profession'] = ''
    df_title['profession_clean'] = ''
    df_title['code'] = ''
    df_title['predscore'] = ''
elif Run == 1:
    corpus = myrootResultsUnique + "df_title_" + year + "_matched_Level5.csv"
    # Import corpus JA
    df_title = pd.read_csv(corpus, on_bad_lines ='skip', engine='python',
dtype={'StepMatched':'str'})
elif Run == 2:
    corpus = myrootResultsUnique + "df_title_" + year + "_matched_Level4.csv"
    # Import corpus JA
    df_title = pd.read_csv(corpus, on_bad_lines ='skip', engine='python',
dtype={'StepMatched':'str'})

# Esco classification
lvl5_professions_file = myroot + 'esco/v1.1.0/occupations_' + lang + '.csv'
lvl5_professions = pd.read_csv(lvl5_professions_file, usecols=['preferredLabel',
'code'])
isco_professions_file = myroot + 'esco/v1.1.0/ISCOGroups_' + lang + '.csv'
isco_professions = pd.read_csv(isco_professions_file, usecols=['preferredLabel',
'code'])
professions = lvl5_professions.append(isco_professions)

# preprocessed professions
if ListLevel == '5':
    list_professions = pd.read_pickle(myrootResultsUnique + 'list_professions.pkl')
else:
    list_professions = pd.read_pickle(myrootResultsUnique +
'list_isco_professions.pkl')
    list_professions = [x for x in list_professions if str(x[3])==ListLevel]
```

```
list_professions_nolen1 = [x for x in list_professions if len(x[0].split(' '))>1]
list_professions_len1 = [x for x in list_professions if len(x[0].split(' '))==1]

def imap_bar(func, args, n_processes = (multiprocessing.cpu_count()-1)):
    p = Pool(n_processes,maxtasksperchild=5000)
    res_list = []
    pbar = tqdm( total = len(args))
    for res in tqdm(p.imap(func, args)):
        pbar.update()
        res_list.append(res)
    pbar.close()
    p.close()
    p.join()
    return res_list

## 1 - Exact Matching

wrkdf = pd.DataFrame()
def find_best_match_lvl5_6_Exact(profession):
    list_titles = []
    list_titles_id = []
    list_output = []
    for title, general_id in zip(wrkdf['title_clean'], wrkdf['general_id']):
        try:
            if title.strip() == profession[0].strip():
                list_titles.append(title)
                list_titles_id.append(general_id)
        except Exception as e:
            logging.error(e)
            continue
    list_best_titles = []
    list_best_titles_unmatched = []
    for i, j in zip(list_titles, list_titles_id):
        list_best_titles.append([i, j])
    list_output.append([profession, list_best_titles])
    return list_output

print('Exact Matching')
starttime = time.time()
wrkdf = df_title.query("StepMatched == '0'")
list_output_s =  imap_bar(find_best_match_lvl5_6_Exact, list_professions)
print('That took {} seconds'.format(time.time() - starttime))

# Matching elements
def matchElements (list_output):
    list_professions_1 = []
    list_professions = []
    list_similars = []
    list_professions_code = []

    for element in tqdm(list_output):
        try:
            for i in element[0][1]:
```

```
                list_professions_1.append(element[0][0][0])
                list_professions.append(element[0][0][1])
                list_professions_code.append(element[0][0][2])
                list_similars.append(i)
        except:
            continue

    df_similarity_match = pd.DataFrame()
    df_similarity_match['profession_clean'] = list_professions_1
    df_similarity_match['profession'] = list_professions
    df_similarity_match['code'] = list_professions_code
    df_similarity_match['title'] = [x[0] for x in list_similars]
    df_similarity_match['general_id'] = [x[1] for x in list_similars]

    #calcolo lunghezza professions
    list_len_professions = []
    for p in df_similarity_match['profession_clean']:
        list_tokens = p.split(' ')
        list_len_professions.append(len(list_tokens))
    df_similarity_match['Length Profession'] = list_len_professions

    #calcolo lunghezza codes
    list_len_codes = []
    for p in df_similarity_match['code']:
        list_len_codes.append(len(p))
    df_similarity_match['Length Code'] = list_len_codes

    # privilegio che non sia alt e che abbia meno numeri e poi che sia più lunga come
numero di parole
    df_similarity_match = df_similarity_match.sort_values(by=['code', 'Length Code',
'Length Profession'], ascending=[True, False,
False]).drop_duplicates(subset=['general_id'], keep='first')

    return df_similarity_match

def SaveMatching(df_similarity_match, df_Title):

    df_matched = df_Title.drop(columns=['profession', 'profession_clean',
'code']).merge(df_similarity_match[['profession', 'profession_clean', 'code',
'general_id']], left_on='general_id', right_on='general_id',
how='right').set_index('general_id')

    df_Title = df_Title.set_index('general_id')
    df_Title.loc[df_matched.index, :] = df_matched[:]
    df_Title = df_Title.reset_index()

    return df_Title

df_similarity_match = matchElements(list_output_s)
indexes =
df_title.loc[df_title.general_id.isin(df_similarity_match.general_id.values)].index
print('Matched step 1: ', len(indexes))
df_title.loc[indexes,'StepMatched'] = '1'
```

```
df_title = SaveMatching(df_similarity_match, df_title)


## 1B - Direct Matching (len > 1)

wrkdf = pd.DataFrame()
def find_best_match_lvl5_6_Direct(profession):
    list_titles = []
    list_titles_id = []
    list_output = []
    for title, general_id in zip(wrkdf['title_clean'], wrkdf['general_id']):
        try:
            if len(profession[0])<=(len(title)):
                mypos = title.find(profession[0].strip()) # se trova la professione
nel titolo della ojv
                if (mypos != -1):
                    # a condizione che a sx e a dx non ci sia carattere (ovvero in
('', ' ') perchè punct rimossi)
                    leftchar = title[mypos-1:mypos]
                    rightchar =
title[mypos+len(profession[0].strip()):mypos+len(profession[0].strip())+1]
                    if leftchar in ('',' ') and rightchar in ('',' '):
                        list_titles.append(title)
                        list_titles_id.append(general_id)
        except Exception as e:
            logging.error(e)
            continue
    list_best_titles = []
    for i, j in zip(list_titles, list_titles_id):
        list_best_titles.append([i, j])
    list_output.append([profession, list_best_titles])
    return list_output

print('Direct Matching')
starttime = time.time()
wrkdf = df_title.query("StepMatched == '0'")
list_output_s = imap_bar(find_best_match_lvl5_6_Direct, list_professions_nolen1)
print('That took {} seconds'.format(time.time() - starttime))

df_similarity_match = matchElements(list_output_s)
indexes =
df_title.loc[df_title.general_id.isin(df_similarity_match.general_id.values)].index
print('Matched step 1B: ', len(indexes))
df_title.loc[indexes,'StepMatched'] = '1B'
df_title = SaveMatching(df_similarity_match, df_title)


## 1B - Direct Matching (len = 1)

print('Direct Matching len=1')
wrkdf = pd.DataFrame()
starttime = time.time()
wrkdf = df_title.query("StepMatched == '0'")
list_output_s = imap_bar(find_best_match_lvl5_6_Direct, list_professions_len1)
```

```
print('That took {} seconds'.format(time.time() - starttime))

df_similarity_match = matchElements(list_output_s)
indexes =
df_title.loc[df_title.general_id.isin(df_similarity_match.general_id.values)].index
print('Matched step 1B_bis: ', len(indexes))
df_title.loc[indexes,'StepMatched'] = '1B_bis'
df_title = SaveMatching(df_similarity_match, df_title)


## 1C - Direct Matching Inverse

wrkdf = pd.DataFrame()
def find_best_match_lvl5_6_DirectInv(profession):
    #profession_list = profession[0].split(' ')
    list_titles = []
    list_titles_id = []
    list_output = []
    for title, general_id in zip(wrkdf['title_clean'], wrkdf['general_id']):
        try:
            #if title.strip().isin(profession[0]):
            #if (profession[0].find(title.strip()) != -1):
            if len(profession[0])>=(len(title)):
                prof2 = profession[0].strip()
                title2 = title.strip()
                #mypos = title.find(profession[0].strip())
                mypos = prof2.find(title2) # se trova la professione nel titolo della
ojv
                if (mypos != -1):
                    # a condizione che a sx e a dx non ci sia carattere (ovvero in
('', ' ') perchè punct rimossi)
                    #leftchar = title[mypos-1:mypos]
                    #rightchar =
title[mypos+len(profession[0].strip()):mypos+len(profession[0].strip())+1]
                    leftchar = prof2[mypos-1:mypos]
                    rightchar = prof2[mypos+len(title2):mypos+len(title2)+1]

                    #print('Titolo = ' + title2 + ' - Prof = ' + prof2)

                    if leftchar in ('',' ') and rightchar in ('',' '):
                        #print('Titolo = ' + title + ' - Prof = ' + prof2)
                        list_titles.append(title)
                        list_titles_id.append(general_id)
        except Exception as e:
            logging.error(e)
            continue
    list_best_titles = []
    list_best_titles_unmatched = []
    for i, j in zip(list_titles, list_titles_id):
        list_best_titles.append([i, j])
    list_output.append([profession, list_best_titles])
    return list_output

print('Direct Inverse Matching')
```

```
starttime = time.time()
wrkdf = df_title.query("StepMatched == '0'")
#wrkdf = wrkdf.head(10)
list_output_s = imap_bar(find_best_match_lvl5_6_DirectInv, list_professions)
print('That took {} seconds'.format(time.time() - starttime))

# Matching elements (keep the occ with minor lenght)
def matchElementsInv(list_output):
    list_professions_1 = []
    list_professions = []
    list_similars = []
    list_professions_code = []

    for element in tqdm(list_output):
        try:
            for i in element[0][1]:
                list_professions_1.append(element[0][0][0])
                list_professions.append(element[0][0][1])
                list_professions_code.append(element[0][0][2])
                list_similars.append(i)
        except:
            continue

    df_similarity_match = pd.DataFrame()
    df_similarity_match['profession_clean'] = list_professions_1
    df_similarity_match['profession'] = list_professions
    df_similarity_match['code'] = list_professions_code
    df_similarity_match['title'] = [x[0] for x in list_similars]
    df_similarity_match['general_id'] = [x[1] for x in list_similars]

    #calcolo lunghezza professions
    list_len_professions = []
    for p in df_similarity_match['profession_clean']:
        list_tokens = p.split(' ')
        list_len_professions.append(len(list_tokens))
    df_similarity_match['Length Profession'] = list_len_professions

    #calcolo lunghezza codes
    list_len_codes = []
    for p in df_similarity_match['code']:
        list_len_codes.append(len(p))
    df_similarity_match['Length Code'] = list_len_codes

    # privilegio che non sia alt e che abbia meno numeri e poi che sia più corta come
numero di parole
    df_similarity_match = df_similarity_match.sort_values(by=['code', 'Length Code',
'Length Profession'], ascending=[True, False,
True]).drop_duplicates(subset=['general_id'], keep='first')

    return df_similarity_match

df_similarity_match = matchElementsInv(list_output_s)
```

```
indexes =
df_title.loc[df_title.general_id.isin(df_similarity_match.general_id.values)].index
print('Matched step 1C: ', len(indexes))
df_title.loc[indexes,'StepMatched'] = '1C'
df_title = SaveMatching(df_similarity_match, df_title)


## 2 - Regular expressions

# Pattern matching with 1 word distance maximum
def find_best_match_lvl5_6_Regex(profession):
    profession_list = profession[0].split(' ')
    if len(profession_list) == 1:
        pattern = fr'[\s*\w*]*\b' + profession_list[0] + fr'\b[\s*\w*]*'
    else:
        pattern = fr'[\s*\w*]*\b'
        for i in profession_list:
            pattern = pattern + i + fr'\b[\s*\w*]*\b'
    #re_pattern = re.compile(pattern)
    list_titles = []
    list_titles_id = []
    list_output = []
    for title, general_id in zip(wrkdf['title_clean'], wrkdf['general_id']):
        # Regex pattern matching
        result = re.match(pattern, title)
        if result != None:
            list_titles.append(title)
            list_titles_id.append(general_id)
    list_best_titles = []
    for i, j in zip(list_titles, list_titles_id):
        list_best_titles.append([i, j])
    list_output.append([profession, list_best_titles])
    return list_output
# REGEX + comunque ORDER DEPENDENT

print('Regular expressions')
starttime = time.time()
wrkdf = df_title.query("StepMatched == '0'")
list_output1_s = imap_bar(find_best_match_lvl5_6_Regex, list_professions)
print('That took {} seconds'.format(time.time() - starttime))

df_similarity_match_1s = matchElements(list_output1_s)
indexes =
df_title.loc[df_title.general_id.isin(df_similarity_match_1s.general_id.values)].index
print('Matched step 2A: ', len(indexes))
df_title.loc[indexes,'StepMatched'] = '2A'
df_title = SaveMatching(df_similarity_match_1s, df_title)


## 2B - Tries <a name='Tries'></a>

#- A second method that uses Trie trees. Questa procedura restituisce anche le istanze
singole di oggetti a qualsiasi distanza. Per cui confrontiamo un result match
percentuale con una soglia basata sul numero di parole distinte fra professione e
```

```
title; qui non possiamo pilotare molto il concetto di distanza con pattern
(f'\\b{tre.regex()}\\b') è una sorta di fuzzy con soglia (aggiungiamo incertezza
considerando valido anche il match con singole parole).
#- E' DIPENDENTE dall'ordine con pattern
(f'[\\s*\\w*\\s*]*\\s{tre.regex()}\\s*[\\w*\\s*]*\\s]') è simile a regex,
discriminiamo in più la soglia.
#- estende regex in performances e in subsuquenzial dai due lati.
#- Usiamo il primo pattern.

from trieregex import TrieRegEx as TRE
import re
def find_best_match_lvl5_6_Trie(profession):
    lower_bound = Threshold_Tries_Lower
    upper_bound = Threshold_Tries_Upper # Also upper bound because of match too large
(prof on title)
    strpattern = ''
    profession_list = profession[0].split(' ')
    if len(profession_list) == 1:
        pattern = fr'[\s*\w*]*\b' + profession_list[0] + fr'\b[\s*\w*]*'
    else:
        pattern = fr'[\s*\w*]*\b'
        for i in profession_list:
            pattern = pattern + i + fr'\b[\s*\w*]*\b'
    list_titles = []
    list_titles_id = []
    list_output = []
    tre = TRE()
    tre = TRE(*profession_list)

    # Add boundary context and compile for matching
    pattern = re.compile(f'\\b{tre.regex()}\\b')  # OR rf'\b{tre.regex()}\b'
    #pattern = re.compile(f'[\\s*\\w*\\s*]*\\s{tre.regex()}\\s*[\\w*\\s*]*\\s]')

    #pattern = re.compile(strpattern) #rex classic
    for title, general_id in zip(wrkdf['title_clean'], wrkdf['general_id']):
            result = pattern.findall(title)
            num_matches = len(result)
            if num_matches > 0:
                #Soglia: se > 70% delle parole matchate (attenzione alle parole
distinte fare disctinct prima)
                num_title = len(title.split())
                rate = num_matches/num_title
                if (rate >= lower_bound) and (rate <= upper_bound):
                    list_titles.append(title)
                    list_titles_id.append(general_id)
    list_best_titles = []
    for i, j in zip(list_titles, list_titles_id):
        list_best_titles.append([i, j])
    list_output.append([profession, list_best_titles])
    return list_output

    # https://stackoverflow.com/questions/42742810/speed-up-millions-of-regex-
replacements-in-python-3
```

```
    # https://leetcode.com/problems/word-break/discuss/1481584/regex-trie-linear-time
    # https://stackoverflow.com/questions/43628742/find-lots-of-string-in-text-python

    # Utilizza un partial matching di parole intere order independent

print('Regular expressions TRIES')
starttime = time.time()
wrkdf = df_title.query("StepMatched == '0'")
list_output2_s = imap_bar(find_best_match_lvl5_6_Trie, list_professions)
print('That took {} seconds'.format(time.time() - starttime))

df_similarity_match_2s = matchElements(list_output2_s)
indexes =
df_title.loc[df_title.general_id.isin(df_similarity_match_2s.general_id.values)].index
print('Matched step 2B: ', len(indexes))
df_title.loc[indexes,'StepMatched'] = '2B'
df_title = SaveMatching(df_similarity_match_2s, df_title)


## 3 - Fuzzy Matching

def find_best_match_lvl5_6_Fuzz(profession): #profession = string
    list_titles = []
    list_titles_id = []
    list_output = []
    list_scores = []
    for title, general_id in zip(wrkdf['title_clean'], wrkdf['general_id']):
        result1 = fuzz.ratio(title[1:-1], profession[0]) # title[1:-1] per eliminare
il primo e l'ultimo spazio
        result2 = fuzz.token_sort_ratio(title[1:-1], profession[0]) # title[1:-1] per
eliminare il primo e l'ultimo spazio
        result3 = fuzz.token_set_ratio(title[1:-1], profession[0]) # title[1:-1] per
eliminare il primo e l'ultimo spazio
        if (result1 >= Threshold_Fuzzy) or (result2 >= Threshold_Fuzzy) or (result3 >=
Threshold_Fuzzy):
            list_titles.append(title)
            list_titles_id.append(general_id)
            list_scores.append(str(max(result1, result2, result3)) + (str(result1 >=
Threshold_Fuzzy)) + (str(result2 >= Threshold_Fuzzy)) + (str(result3 >=
Threshold_Fuzzy)))
    list_best_titles = []
    for i, j, k in zip(list_titles, list_titles_id, list_scores):
        list_best_titles.append([i, j, k])
    list_output.append([profession, list_best_titles])
    return list_output

print('Fuzzy Matching')
starttime = time.time()
wrkdf = df_title.query("StepMatched == '0'")
list_output30_s = imap_bar(find_best_match_lvl5_6_Fuzz, list_professions_nolen1)
print('That took {} seconds'.format(time.time() - starttime))

# Matching elements
def matchElementsFuzzy(list_output):
```

```
    list_professions_1 = []
    list_professions = []
    list_similars = []
    list_professions_code = []

    for element in tqdm(list_output):
        try:
            for i in element[0][1]:
                list_professions_1.append(element[0][0][0])
                list_professions.append(element[0][0][1])
                list_professions_code.append(element[0][0][2])
                list_similars.append(i)
        except:
            continue

    df_similarity_match = pd.DataFrame()
    df_similarity_match['profession_clean'] = list_professions_1
    df_similarity_match['profession'] = list_professions
    df_similarity_match['code'] = list_professions_code
    df_similarity_match['title'] = [x[0] for x in list_similars]
    df_similarity_match['general_id'] = [x[1] for x in list_similars]
    if Run==0:
        df_similarity_match['similarity'] = [x[2] for x in list_similars]
    else:
        df_similarity_match['similarity_' + str(Run)] = [x[2] for x in list_similars]

    #calcolo lunghezza professions
    list_len_professions = []
    for p in df_similarity_match['profession_clean']:
        list_tokens = p.split(' ')
        list_len_professions.append(len(list_tokens))
    df_similarity_match['Length Profession'] = list_len_professions

    #calcolo lunghezza codes
    list_len_codes = []
    for p in df_similarity_match['code']:
        list_len_codes.append(len(p))
    df_similarity_match['Length Code'] = list_len_codes

    # privilegio che non sia alt e che abbia meno numeri e poi che sia più lunga come
numero di parole
    df_similarity_match = df_similarity_match.sort_values(by=['code', 'Length Code',
'Length Profession'], ascending=[True, False,
False]).drop_duplicates(subset=['general_id'], keep='first')

    return df_similarity_match


def SaveMatchingDirectFuzzy(df_similarity_match, df_Title):

    if Run==0:
        df_matched = df_Title.drop(columns=['profession', 'profession_clean', 'code',
'similarity']).merge(df_similarity_match[['profession', 'profession_clean', 'code',
```

```
'similarity', 'general_id']], left_on='general_id', right_on='general_id',
how='right').set_index('general_id')
    else:
        df_matched = df_Title.drop(columns=['profession', 'profession_clean',
'code']).merge(df_similarity_match[['profession', 'profession_clean', 'code',
'similarity_' + str(Run), 'general_id']], left_on='general_id', right_on='general_id',
how='right').set_index('general_id')

    df_Title = df_Title.set_index('general_id')
    df_Title.loc[df_matched.index, :] = df_matched[:]
    df_Title = df_Title.reset_index()

    return df_Title

df_similarity_match = matchElementsFuzzy(list_output30_s)
indexes =
df_title.loc[df_title.general_id.isin(df_similarity_match.general_id.values)].index
print('Matched step 3: ', len(indexes))
df_title.loc[indexes,'StepMatched'] = '3'
if Run==0:
    df_title['similarity'] = ''
df_title = SaveMatchingDirectFuzzy(df_similarity_match, df_title)


## 3 - FastText supervised

def SaveMatchingClass(df_new, df_Title):

    df_matched = df_Title.drop(columns=['profession', 'code',
'predscore']).merge(df_new[['profession', 'code', 'predscore', 'general_id']],
left_on='general_id', right_on='general_id', how='right').set_index('general_id')

    df_Title = df_Title.set_index('general_id')
    df_Title.loc[df_matched.index, :] = df_matched[:]
    df_Title = df_Title.reset_index()

    return df_Title

for x in ['2', '3', '4', '5', '7', '8', '9']:

    df_train = df_title[df_title.StepMatched!='0'].reset_index(drop=True)
    df_train = df_train[df_train.code.apply(lambda x: x[0])==x]
    if ListLevel == '5':
        df_train.code = df_train.code.apply(lambda x: x.strip('alt_'))
    df_new = df_title[df_title.StepMatched=='0'].reset_index(drop=True)

    dflabel = '__label__' + df_train['code']  + ' ' + df_train['title_clean']
    frames = [dflabel]
    dflabel = pd.concat(frames, ignore_index = True)
    dflabel.to_csv(myrootResultsUnique + 'TextLabeled.txt', index=False )
    model = fasttext.train_supervised(myrootResultsUnique + 'TextLabeled.txt',
wordNgrams=4, minCount=5, minCountLabel=1)

    # custom predict
```

```
    import numpy as np
    def predict(row):
        try:
            res = model.predict(row['title_clean'])
            score = round(res[1][0],2)
            label = np.char.replace(res[0],'__label__','')[0]

        except:
            score = 0
            label = ''
        return label, score

    def GetClassDes(code):
        try:
            if len(code)>0:
                a = professions[['preferredLabel']][(professions['code']==code)]
                a = a.values.tolist()[0][0]
            else:
                a = ''
        except:
            a = ''
        if not a:
            a = ''
        return a

    pandarallel.initialize(progress_bar=True, nb_workers=40, use_memory_fs=True,
verbose = 0)
    df_new['predscore'] = 0
    df_new['code'], df_new['predscore'] = df_new.parallel_apply(predict,axis=1,
result_type='expand').T.values
    df_new['profession'] = df_new.parallel_apply(lambda x: GetClassDes(x['code']),
axis=1)

    df_new09 = df_new[df_new.predscore>=Threshold_FastText]
    indexes = df_title.loc[df_title.general_id.isin(df_new09.general_id.values)].index
    print('Matched semantic step: ', len(indexes))
    df_title.loc[indexes,'StepMatched'] = '4'
    df_title = SaveMatchingClass(df_new09, df_title
)
dfCount = pd.DataFrame(df_title.groupby('StepMatched')['general_id'].count())
dfCount.columns = ['StepMatched']
dfCount['StepMatchedPerc'] = round(dfCount['StepMatched'] /
dfCount['StepMatched'].sum(),4)
dfCount.to_csv(myrootResultsUnique + "Results_syntPip_" + year + "_Level" + ListLevel
+ ".csv")
dfCount

## Add idesco_level_5 column
if ListLevel == '5':
    df_title['idesco_level_5'] = df_title.code.apply(lambda x: x.strip('alt_'))

# Save dataframe matched
```

```
filename = myrootResultsUnique + 'df_title_' + year + '_matched_Level' + ListLevel +
'.csv'
df_title.to_csv(filename, index=False)
if df_title.shape[0]<1999999:
    filename_xlsx = myrootResultsUnique + 'df_title_' + year + '_matched_Level' +
ListLevel + '.xlsx'
    writer = pd.ExcelWriter(filename_xlsx, engine='xlsxwriter')
    df_title.iloc[0:1000000,:].to_excel(writer, sheet_name="1", index=False)
    df_title.iloc[1000000:,:].to_excel(writer, sheet_name="2", index=False)
    writer.save()

#To clean memory!!
if gc.isenabled() == False:
    gc.collect()
    print('Memory collected!')
else:
    print('Automatically Memory collected!')

print('Entire task took {} minutes'.format((time.time() - StartTimeEntireTask)/60))
logging.shutdown()
```