Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА

на тему *Метод класифікації резюме за професійними категоріями з використанням машинного навчання*

Рівень вищої освіти *другий (магістерський)*

Галузь знань *12 – Інформаційні технології*
Шифр і найменування

Спеціальність *122 – Комп'ютерні науки*
Код і найменування

Освітня програма *Комп'ютерні науки*
Назва

Виконав: *студент 2 курсу, група КНм-24-1* — Курс, група виконавця; Підпис; *Нікіта КОРКУНДА* — Ім'я, ПРІЗВИЩЕ

Керівник: *д.т.н., професор кафедри КН* — Науковий ступінь, посада; Підпис; *Едуард МАНЗЮК* — Ім'я, ПРІЗВИЩЕ

Нормоконтроль: *к.т.н., доцент кафедри КН* — Науковий ступінь, посада; Підпис; *Руслан БАГРІЙ* — Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор — Підпис; *Олександр БАРМАК* — Ім'я, ПРІЗВИЩЕ

15 грудня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет *інформаційних технологій*
Кафедра *комп'ютерних наук*
Освітній ступінь *магістр*
Галузь знань *12 – Інформаційні технології*
Спеціальність *122 – Комп'ютерні науки*

ЗАТВЕРДЖУЮ
Завідувач кафедри *комп'ютерних наук*

(підпис)
д.т.н., професор Олександр БАРМАК
«*28*» *08* *2025* року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

1. Тема кваліфікаційної роботи: *«Метод класифікації резюме за професійними категоріями з використанням машинного навчання»*

2. Завдання видано студенту *Нікіті КОРКУНДА*
(Ім'я, ПРІЗВИЩЕ)

3. Керівник роботи *професор кафедри КН Едуард МАНЗЮК*
(Ім'я, ПРІЗВИЩЕ)

4. Затверджені наказом університету від «*25*» *08* *2025* р. № *65*

5. Дата видачі завдання студенту: «*28*» *05* *2025* р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи полягає у підвищенні точності класифікації резюме за професійними категоріями шляхом розробки методу з використанням векторного представлення тексту на основі уніграм та біграм і нейронних мереж прямого поширення. Задачі дослідження: провести аналіз існуючих методів та підходів до класифікації текстових документів; розробити метод класифікації резюме за професійними категоріями з використанням векторного представлення тексту TF-IDF; розробити програмну реалізацію методу класифікації резюме за професійними категоріями з використанням машинного навчання; провести експериментальне дослідження ефективності спроектованого методу.

Ключові слова: класифікація резюме, обробка мови, TF-IDF, нейронні мережі, автоматизація рекрутингу, багатокласова класифікація.

7. Календарний план виконання кваліфікаційної роботи:

№	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання	Примітка
1	*Вибір напряму дослідження та узгодження теми кваліфікаційної роботи з керівником, складання календарного графіка виконання роботи*	*вересень 2025*	*Виконано*
2	*Ознайомлення з предметною областю, аналіз існуючих методів і моделей, формулювання мети та завдань дослідження, визначення об'єкта й предмета дослідження*	*вересень 2025*	*Виконано*
3	*Розробка методу чи моделі для вирішення обраного завдання, опис архітектури рішення*	*жовтень 2025*	*Виконано*
4	*Програмна реалізація методу чи моделі*	*жовтень 2025*	*Виконано*
5	*Дослідження ефективності та експериментальна перевірка результатів, порівняння з відомими підходами*	*листопад 2025*	*Виконано*
6	*Написання пояснювальної записки, оформлення відповідно до вимог, врахування зауважень керівника*	*листопад 2025*	*Виконано*
7	*Підготовка презентаційних матеріалів та попередній захист*	*листопад 2025*	*Виконано*
8	*Перевірка пояснювальної записки на відповідність вимогам оформлення (нормоконтроль) та перевірка на академічну доброчесність. Отримання відгуку керівника та рецензії.*	*грудень 2025*	*Виконано*
9	*Публічний захист кваліфікаційної роботи*	*грудень 2025*	*Виконано*

Виконавець: *студент групи КНм-24-1* _____ *Нікіта КОРКУНДА*
Група виконавця | Підпис | Ім'я, ПРІЗВИЩЕ

Керівник: *д.т.н., проф. каф. КН* _____ *Едуард МАНЗЮК*
Науковий ступінь, посада | Підпис | Ім'я, ПРІЗВИЩЕ

Реферат

Кваліфікаційна робота магістра присвячена розробці методу класифікації резюме за професійними категоріями з використанням машинного навчання.

Актуальність теми. Актуальність роботи полягає в критичній необхідності автоматизації процесів відбору кадрів та обробки великих обсягів резюме в сучасних системах рекрутингу. Ручний аналіз резюме, який традиційно виконується HR-спеціалістами, є надзвичайно трудомістким, потребує значного часу, схильний до суб'єктивних помилок і часто залежить від рівня кваліфікації спеціаліста.

Досягнення у галузі машинного навчання, а також природної мови дозволяють значно покращити якість і швидкість процесів підбору персоналу, автоматизуючи класифікацію резюме і забезпечуючи високу точність визначення професійних категорій кандидатів. Це дозволяє скоротити час на пошук відповідних кандидатів, підвищенню ефективності роботи HR-відділів, зменшенню витрат компаній на рекрутинг і покращенню загального процесу найму.

Впровадження методів векторизації тексту та нейронних мереж допомагає подолати обмеження, пов'язані з різноманітністю форматів та стилів написання резюме, що є типовою проблемою в сфері управління персоналом. Використання біграм для збереження контекстної інформації дозволяє краще розрізняти схожі професійні категорії. Це підкреслює актуальність роботи як для практичного застосування в індустрії, так і для наукових досліджень у галузі застосування методів машинного навчання до задач обробки текстових документів.

Мета роботи полягає у підвищенні точності класифікації резюме за професійними категоріями шляхом розробки методу з використанням векторного представлення тексту на основі унірам та біграм і нейронних мереж прямого поширення.

Задачі дослідження:

– провести аналіз існуючих методів та підходів до класифікації текстових документів з використанням методів машинного навчання та обробки мови;

– розробити метод класифікації резюме за професійними категоріями з використанням векторного представлення тексту TF-IDF та нейронних мереж;

– розробити програмну реалізацію методу класифікації резюме за професійними категоріями з використанням машинного навчання;

– провести експериментальне дослідження ефективності спроектованого методу шляхом порівняння базової та модифікованої моделей та оцінки їх точності класифікації на датасеті резюме.

Об'єкт дослідження – процес автоматизованої класифікації резюме за професійними категоріями.

Предмет дослідження – моделі, методи та засоби класифікації текстових документів з використанням машинного навчання та векторного представлення на основі унірам та біграм.

Методи дослідження. Застосовано методи обробки природної мови, векторизацію тексту TF-IDF, нейронні мережі прямого поширення, регуляризацію dropout, методи оптимізації Adam.

Наукова новизна одержаних результатів. Удосконалено метод класифікації резюме за професійними категоріями, який відрізняється від існуючих використанням комбінованого векторного представлення на основі унірам та біграм з оптимізованим співвідношенням, що дозволило покращити точність класифікації на 3.5 % порівняно з базовим підходом на основі лише унірам.

Апробація результатів кваліфікаційної роботи магістра та публікації. Коркунда Н.С., Манзюк Е.А., Скрипник Т.К. Метод класифікації резюме за професійними категоріями з використанням машинного навчання. Збірник наукових праць за матеріалами XVII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп’ютерних наук АПКН-2025». - Хмельницький, 2025. - С.222 - 225

Структура та обсяг роботи. Робота містить вступ, чотири розділи, загальні висновки, список використаних джерел та додатків. Обсяг основного тексту – 90 сторінка, включаючи 14 рисунків, 1 таблицю та 46 джерел у списку літератури.

Ключові слова: класифікація резюме, машинне навчання, обробка природної мови, TF-IDF, нейронні мережі, біграми, векторизація тексту, автоматизація рекрутингу, багатокласова класифікація.

Зміст

Перелік скорочень

Скорочення, термін, позначення	**Пояснення**
NLP	Natural Language Processing (Обробка природної мови)
TF-IDF	Term Frequency-Inverse Document Frequency (Частота терміна-обернена частота документа)
МН	Машинне навчання
ШІ	Штучний інтелект
ReLU	Rectified Linear Unit (Випрямлена лінійна одиниця)
Adam	Adaptive Moment Estimation (Адаптивна оцінка моментів)
CV	Cross-Validation (Перехресна валідація)
API	Application Programming Interface (Програмний інтерфейс застосування)
HR	Human Resources (Людські ресурси)

Вступ

Кваліфікаційна робота магістра присвячена розробці методу класифікації резюме за професійними категоріями з використанням машинного навчання.

Актуальність теми. Актуальність роботи полягає в критичній необхідності автоматизації процесів відбору кадрів та обробки великих обсягів резюме в сучасних системах рекрутингу. Ручний аналіз резюме, який традиційно виконується HR-спеціалістами, є надзвичайно трудомістким, потребує значного часу, схильний до суб'єктивних помилок і часто залежить від рівня кваліфікації спеціаліста.

Досягнення у галузі машинного навчання, а також обробки мови дозволяють значно покращити якість і швидкість процесів підбору персоналу, автоматизуючи класифікацію резюме і забезпечуючи високу точність визначення професійних категорій кандидатів. Це дозволяє скоротити час на пошук відповідних кандидатів, підвищенню ефективності роботи HR-відділів, зменшенню витрат компаній на рекрутинг і покращенню загального процесу найму.

Впровадження методів векторизації тексту та нейронних мереж допомагає подолати обмеження, пов'язані з різноманітністю форматів та стилів написання резюме, що є типовою проблемою в сфері управління персоналом. Використання біграм для збереження контекстної інформації дозволяє краще розрізняти схожі професійні категорії. Це підкреслює актуальність роботи як для практичного застосування в індустрії, так і для наукових досліджень у галузі застосування методів машинного навчання до задач обробки текстових документів.

Мета роботи. Мета роботи полягає у підвищенні точності класифікації резюме за професійними категоріями шляхом розробки методу з використанням векторного представлення тексту на основі уніграм та біграм і нейронних мереж прямого поширення.

Задачі дослідження:

– провести аналіз існуючих методів та підходів до класифікації текстових документів з використанням методів машинного навчання та обробки мови;

– розробити метод класифікації резюме за професійними категоріями з використанням векторного представлення тексту TF-IDF та нейронних мереж;

– розробити програмну реалізацію методу класифікації резюме за професійними категоріями з використанням машинного навчання;

– провести експериментальне дослідження ефективності спроектованого методу шляхом порівняння базової та модифікованої моделей та оцінки їх точності класифікації на датасеті резюме.

Об'єкт дослідження – процес автоматизованої класифікації резюме за професійними категоріями.

Предмет дослідження – моделі, методи та засоби класифікації текстових документів з використанням машинного навчання та векторного представлення на основі уніграм та біграм.

Методи дослідження. Застосовано методи обробки природної мови, векторизацію тексту TF-IDF, нейронні мережі прямого поширення, регуляризацію dropout, методи оптимізації Adam.

Наукова новизна одержаних результатів. Удосконалено метод класифікації резюме за професійними категоріями, який відрізняється від існуючих використанням комбінованого векторного представлення на основі уніграм та біграм з оптимізованим співвідношенням, що дозволило покращити точність класифікації на 3.5 % порівняно з базовим підходом на основі лише уніграм.

Апробація результатів кваліфікаційної роботи магістра та публікації. Коркунда Н.С., Манзюк Е.А., Скрипник Т.К. Метод класифікації резюме за професійними категоріями з використанням машинного навчання. Збірник наукових праць за матеріалами XVII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп’ютерних наук АПКН-2024». - Хмельницький, 2025. - С.222 – 225.

Структура та обсяг роботи. Робота містить вступ, чотири розділи, загальні висновки, список використаних джерел та додатків. Обсяг основного тексту – 90 сторінка, включаючи 14 рисунків, 1 таблицю та 46 джерел у списку літератури.

Розділ 1 Аналітичний огляд методів класифікації резюме

1.1 Характеристика задачі автоматизованої класифікації резюме

У сучасному світі рекрутингу процес обробки резюме є одним із найбільш трудомістких етапів підбору персоналу, особливо для великих компаній, які отримують тисячі заявок на одну вакансію. Автоматизація класифікації резюме за професійними категоріями за допомогою методів машинного навчання дозволяє значно прискорити початковий скринінг кандидатів, зменшити вплив людського фактора та суб'єктивності в оцінюванні, а також підвищити загальну ефективність процесу підбору кадрів.

Метод автоматизованої класифікації резюме полягає в тому, щоб автоматично присвоїти кожне резюме до певної професійної категорії на основі аналізу текстового контенту документа. Такий контент включає інформацію про досвід роботи кандидата, його професійні навички та компетенції, освіту, сертифікати та інші релевантні дані. Типові категорії класифікації можуть включати IT-спеціалістів, фахівців з маркетингу, працівників фінансової сфери, HR-менеджерів, інженерів та багато інших професійних напрямків.

Згідно з дослідженнями останніх років, традиційні методи машинного навчання, такі як Баєс чи метод опорних вектрів, досягають точності класифікації на рівні 80-85%, але стикаються з певними обмеженнями в обробці складних семантичних нюансів та контекстуальних залежностей у тексті резюме [1, 2]. Водночас, сучасні підходи на основі глибокого навчання та великих мовних моделей, включаючи архітектури BERT та інші трансформерні моделі, дозволяють підвищити метрики якості до 92% за показником F1-score, оскільки ці методи здатні краще враховувати контекст та багатозначність термінів [3, 4].

Значення цієї теми зумовлене зростанням онлайн-рекрутингу та цифровізацією процесів підбору персоналу. За даними 2024 року, приблизно 70% компаній у світі використовують автоматизовані інструменти для первинного відбору кандидатів, але лише близько 40% із них досягають точності класифікації

понад 90% через брак якісних та збалансованих датасетів, а також недостатню адаптацію моделей до специфіки різних галузей [5, 6].

Історично розвиток напрямку класифікації резюме почався з простих підходів на основі ключових слів ще на початку 2010-х років, коли алгоритм Баєса домінував у базових системах автоматизованого скринінгу [7–9]. З того часу підходи еволюціонували до використання складних архітектур глибокого навчання. Починаючи з 2020 року, фокус змістився до використання трансформерних моделей та великих мовних моделей, які демонструють найкращі результати на сьогоднішній день [10–12].

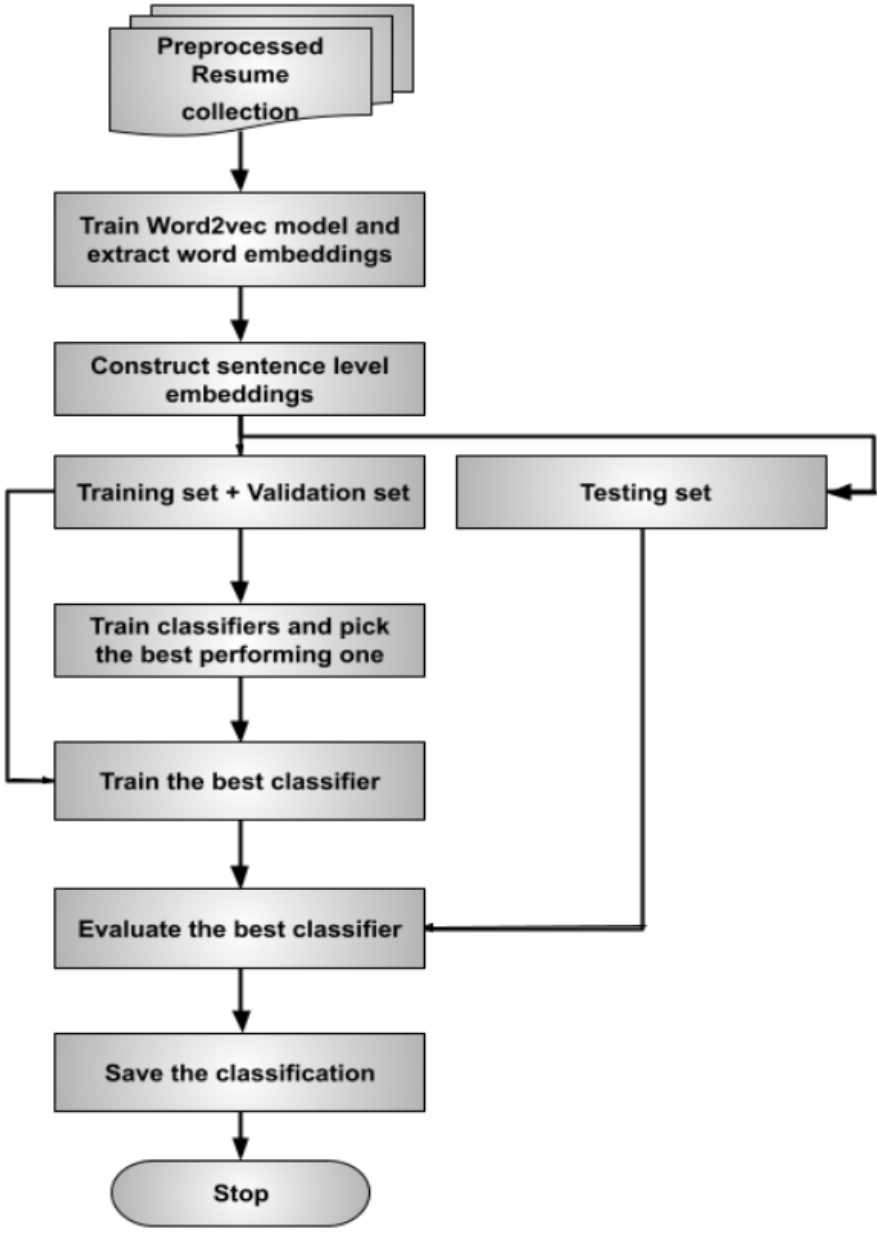


Рисунок 1.1 – Підготовка даних та процес обробки [3]

Сучасні дослідження більше зосереджені на автоматичному вивченні представлень тексту без необхідності ручного втручання, але водночас виникають нові виклики. Зокрема, актуальними є етичні питання, такі як потенційна упередженість моделей щодо певних демографічних груп, забезпечення справедливості в процесі відбору кандидатів та прозорість прийняття рішень автоматизованими системами [13–15]. Також важливим є питання конфіденційності персональних даних та відповідності вимогам законодавства про захист даних.

У контексті української реальності, де IT-сектор росте приблизно на 20% щорічно, автоматизація класифікації резюме стає дуже важливою для великих компаній, таких як EPAM чи SoftServe, де обробляється до 5000 резюме щомісяця. Ефективні системи класифікації дозволяють цим компаніям швидше знаходити потрібних спеціалістів, скорочувати час на підбір персоналу та підвищувати якість найму. Крім того, український ринок праці має свої специфічні особливості, такі як високий рівень володіння мовою серед IT-спеціалістів, що призводить до необхідності обробки багатомовних резюме.

1.2 Аналіз існуючих публікацій та наукових підходів

Огляд наукової літератури показує, що дослідження в галузі автоматизованої класифікації резюме можна умовно розділити на кілька основних напрямків. Перший напрямок базується на традиційних методах машинного навчання, таких як Баєса, логістична регресі та метод опорних вектоів. Ці методи зазвичай використовують ручне створення ознак, наприклад, через TF-IDF або мішок слів, і показують прийнятну ефективність на невеликих та середніх датасетах [16–18].

Другий напрямок включає методи глибокого навчання, такі як різного виду нейронні мережі, рекурентні нейронні мережі, LSTM, GRU та трансформерні моделі, які автоматично вивчають складні представлення текстових даних. Третій напрямок стосується використання великих попередньо навчених моделей мови,

таких як BERT, GPT та інші, які дають хороші результати, але вимагають великих обчислювалень.

Четвертий напрямок охоплює дослідження, присвячені створенню повноцінних автоматизованих систем скринінгу, які інтегрують різні компоненти: парсинг резюме, витягування інформації, класифікацію, ранжування кандидатів та генерацію звітів. П'ятий напрямок зосереджений на етичних аспектах використання штучного інтелекту в процесах найму, включаючи виявлення та зменшення упередженості, забезпечення справедливості та прозорості.

1.2.1 Традиційні підходи до класифікації резюме

Ранні роботи в галузі класифікації резюме базувались на традиційних методах машинного навчання зі звичним інжинірингом ознак. Дослідження [2] представляє порівняльний аналіз кількох традиційних методів навчання для класифікації резюме. Автори використовували датасет із 2000 резюме та показали, що логістична реграсія досягає F1-score на рівні 0.84, що на 5% краще за Баєса. У цьому дослідженні підкреслюється важливість правильного вибору гіперпараметрів та методів регуляризації для уникнення перенавчання моделі.

Робота [3] зосереджена на використанні природної обробки мови та штучного інтелекту для автоматизації класифікації резюме. У дослідженні описуються ансамблеві методи, які комбінують кілька базових класифікаторів для підвищення загальної точності системи. Дослідження [19, 20] представляють систему класифікації резюме, яка використовує поєднання методів NLP та машинного навчання. Описується детальний процес препроцесингу тексту, включаючи видалення стоп-слів, стемінг та лематизацію, а також використання TF-IDF для створення векторних представлень документів. Система досягає точності на рівні 82% при класифікації резюме на 10 різних категорій. Особлива увага приділяється впливу різних методів токенізації на кінцеву якість класифікації.

Публікація [21] зосереджена на використанні n-грам ознак для класифікації резюме. Показано, що комбінація уніграм, біграм та триграм може значно

покращити якість класифікації порівняно з використанням лише уніграм. Модель на основі SVM з n-грам ознаками досягає точності на рівні 85%. Дослідження також аналізує вплив розміру словника на продуктивність моделі та оптимальний баланс між розміром простору ознак та якістю класифікації.

Робота [22] досліджує класифікацію резюме на основі особистісних характеристик з використанням алгоритмів машинного навчання. Це дослідження розширює традиційний підхід, враховуючи не лише технічні навички, але й м'які навички та особистісні риси кандидатів, які можуть бути виведені з тексту резюме та супровідних листів.

Дослідження [23] порівнює ефективність базових нейронних мереж на датасеті з 5000 резюме. Результати показують, що SVM з RBF-ядром досягає F1-score на рівні 0.86, що робить цей метод одним із найкращих для бінарної класифікації резюме на категорії кваліфікованих та некваліфікованих. Робота також досліджує вплив незбалансованості класів на якість моделей та методи боротьби з цією проблемою.

Публікація [24] представляє результати класифікації резюме з використанням різних технік машинного навчання на індійському датасеті. Дослідження підкреслює важливість адаптації моделей до локального контексту та специфіки формулювань у резюме з різних регіонів. Показано, що моделі, навчені на західних датасетах, можуть показувати гіршу якість при застосуванні до резюме з інших культурних контекстів.

1.2.2 Підходи на основі глибокого навчання

Методи глибокого навчання почали активно застосовуватись для класифікації резюме починаючи з 2020 року, коли стало очевидно, що традиційні підходи мають обмеження у розумінні семантики тексту [25, 26]. Дослідження [27, 28] представляє використання згорткових нейронних мереж для багатоміткової класифікації резюме. Робота показує, що CNN ефективно витягує локальні патерни з тексту та досягає F1-score на рівні 0.88 для класифікації навичок. Однак

відзначається, що CNN має обмеження при роботі з довгими текстами, де важливий глобальний контекст.

Робота [29] представляє підхід на основі довгої короткочасної пам'яті для автоматизації класифікації резюме. Використовується двостороння LSTM з механізмом уваги для кращого захоплення контекстуальних залежностей у тексті резюме. Модель досягає F1-score на рівні 0.91, що є значним покращенням порівняно з традиційними методами. Дослідження демонструє, що механізм уваги дозволяє моделі фокусуватись на найбільш інформативних частинах резюме, таких як розділи з досвідом роботи та ключовими навичками.

Публікації [4, 30] представляють роботу, в якій використовуються великі датасети та великі мовні моделі для класифікації резюме. Це дослідження показало, що моделі на основі трансформерів, зокрема BERT, можуть досягати точності на рівні 92% при класифікації резюме на 24 різні професійні категорії. Особливістю роботи є створення великомасштабного датасету з більш ніж 13 мільйонів синтетичних резюме, що дозволило подолати проблему браку розмічених даних.

Дослідження також обговорює проблему потенційної упередженості моделей, яка може виникати через специфіку навчальних даних. Пропонуються методи для виявлення та зменшення упередженості, включаючи аналіз розподілу передбачень за демографічними групами та використання техніки adversarial debiasing. Показано, що моделі можуть непередбачено відтворювати упередження, присутні в історичних даних про найм.

Робота [31] представляє гібридну архітектуру, яка комбінує нейронні мережі з BERT та Gensim для класифікації резюме студентів інженерних спеціальностей. Модель досягає точності до 94% при оптимальному налаштуванні гіперпараметрів. Дослідження демонструє переваги гібридних підходів, які комбінують статистичні методи представлення тексту з глибоким навчанням.

Дослідження [32] використовує великі мовні моделі для генерації синтетичних даних резюме, які потім використовуються для покращення класифікації описів вакансій. Це дослідження демонструє, що синтетичні дані можуть допомогти вирішити проблему дефіциту реальних навчальних даних,

особливо для рідкісних професійних категорій. Показано, що правильно згенеровані синтетичні резюме можуть значно покращити узагальнювальну здатність моделей.

1.2.3 Великі мовні моделі та агентні системи

Останні дослідження зосереджені на використанні великих мовних моделей для задач класифікації та скринінгу резюме. Публікації [33, 34] представляють фреймворки на основі агентів великих мовних моделей для автоматизації скринінгу резюме. Ці системи використовують можливості LLM для контекстно-залежного аналізу резюме та пояснення рішень про відбір кандидатів, що підвищує прозорість процесу найму.

Особливістю агентних систем є можливість проведення багатоетапного аналізу резюме з використанням різних спеціалізованих агентів. Наприклад, один агент може фокусуватись на аналізі технічних навичок, інший - на оцінці досвіду роботи, третій - на перевірці відповідності освіти вимогам вакансії. Такий розподіл завдань між агентами дозволяє досягти більш детального та точного аналізу порівняно з монолітними моделями [35, 36].

Робота [37] демонструє використання RAG підходу для підвищення фактичної точності LLM при аналізі резюме. RAG дозволяє моделі отримувати доступ до зовнішньої бази знань для верифікації інформації та зменшення галюцинацій. Система показує значне зменшення помилкових висновків порівняно з базовими LLM без доступу до зовнішніх джерел інформації.

1.2.4 Автоматизовані системи та практичні реалізації

Окрему групу досліджень складають роботи, присвячені створенню повноцінних автоматизованих систем скринінгу резюме [38, 39]. Проєкт ResumeSorter [1] представляє NLP процес, який комбінує парсинг резюме з класифікацією. Система досягає F1-score на рівні 0.79-0.85 при класифікації на 24

категорії. Особлива увага приділяється обробці резюме у різних форматах, включаючи PDF, DOCX та plain text.

Дослідження [40] представляє AI-базовану систему відстеження кандидатів, яка включає валідацію інформації з резюме через зовнішні джерела. Публікація [41] демонструє використання BERT для класифікації резюме на платформі Kaggle. Наводиться детальний опис процесу точного налаштування попередньо навченої моделі на датасеті резюме, включаючи підбір оптимальних гіперпараметрів та методи боротьби з перенавчанням. Робота містить практичні рекомендації щодо розміру батчу, кількості епох навчання та стратегій аугментації даних.

Реалізація на HuggingFace [42] представляє готову до використання модель для класифікації резюме на основі BERT. Модель навчена на датасеті з понад 10000 розмічених резюме та показує точності на рівні 88-90%. Особливістю реалізації є можливість легкої інтеграції в існуючі HR-системи через API.

Проєкт на GitHub [43] демонструє базову реалізацію системи класифікації резюме з використанням традиційних методів машинного навчання. Надається повний код препроцесингу, тренування моделі та оцінювання результатів, що робить проєкт корисним для освітніх цілей та як відправну точку для більш складних систем.

Інша реалізація на GitHub [44, 45] використовує SVM з тюнінгом гіперпараметрів через GridSearchCV. Проєкт включає детальний аналіз впливу різних параметрів на якість класифікації та рекомендації щодо оптимізації продуктивності.

Важливим напрямком досліджень є аналіз етичних аспектів використання автоматизованих систем класифікації резюме. Дослідження показують, що моделі машинного навчання можуть непередбачено відтворювати та навіть посилювати упередження, присутні в історичних даних про найм [46].

Для боротьби з упередженістю пропонуються різні підходи. Перший підхід полягає в ретельному аудиті навчальних даних та видаленні чи балансуванні упереджених прикладів. Другий підхід використовує техніки точного налаштування, які явно враховують вимоги справедливості при навчанні моделі. Третій підхід

базується на посткорекції передбачень моделі для забезпечення рівних можливостей для всіх груп кандидатів.

Аналіз літератури виявляє кілька ключових викликів та обмежень існуючих підходів до класифікації резюме. Проблема якості та доступності даних залишається критичною. Більшість публічно доступних датасетів резюме є відносно невеликими від кількох сотень до кількох тисяч зразків та можуть не представляти повний спектр професійних категорій та форматів резюме.

Існує проблема узагальнення моделей на нові галузі та типи резюме. Модель, навчена на резюме IT-спеціалістів, може погано працювати на резюме медичних працівників або юристів через різницю в термінології та структурі документів. Це вимагає або створення спеціалізованих моделей для кожної галузі, або розробки більш універсальних підходів.

Динамічна природа ринку праці створює виклик для підтримки актуальності моделей. Нові професії з'являються, назви посад змінюються, технології та навички еволюціонують. Модель, навчена кілька років тому, може не розпізнавати сучасні назви посад чи нові технологічні стеки.

1.3 Огляд архітектур та методів машинного навчання для класифікації текстів

1.3.1 Традиційні методи машинного навчання

Традиційні методи машинного навчання формують фундамент для класифікації текстових документів, включаючи резюме. Ці методи базуються на перетворенні тексту в числові представлення та використанні статистичних алгоритмів для визначення категорій документів.

Основою традиційних підходів є представлення тексту у вигляді векторів ознак. Найпоширенішими методами векторизації є мішок слів, який представляє документ як набір слів без урахування порядку, та TF-IDF, який враховує як частоту слова в документі, так і його рідкісність у корпусі. TF-IDF обчислюється за

формулою, де враховується логарифм частоти слова в документі помножений на обернений логарифм частоти документів, що містять це слово.

Метод Баєса базується на теоремі про умовну незалежність ознак. Незважаючи на спрощене припущення про незалежність, метод демонструє дивовижно хороші результати на практиці для класифікації текстів. Основна перевага методу - швидкість навчання та передбачення, а також можливість роботи з відносно невеликими навчальними вибірками. Дослідження показують, що метод Баєса досягає точності близько 82% на датасетах резюме [1].

Важливим аспектом використання логістична регресія є регуляризація, яка запобігає перенавчанню моделі. Найпоширенішими типами регуляризації є L1 та L2. L1 регуляризація сприяє розрідженості моделі, автоматично виконуючи відбір ознак шляхом встановлення коефіцієнтів деяких ознак в нуль. L2 регуляризація зменшує величину всіх коефіцієнтів, що робить модель більш стійкою до шуму в даних. Дослідження демонструють, що Logistic Regression з L2 регуляризацією досягає F1-score близько 0.84 на задачах класифікації резюме [2].

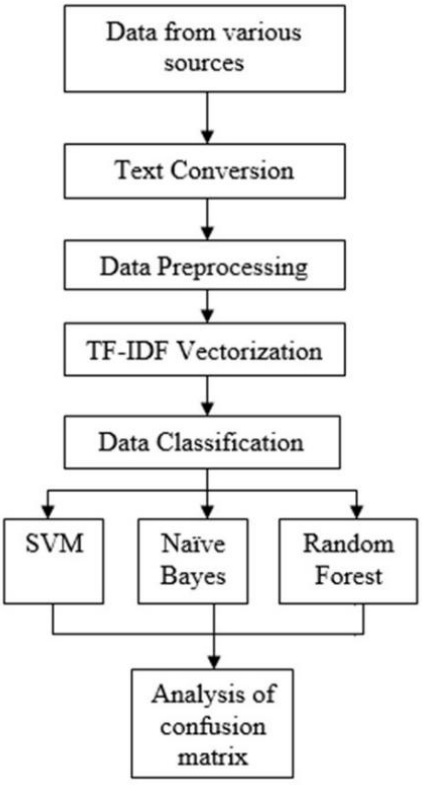


Рисунок 1.2 – Блок-схема системи [2]

Дослідження показують, що SVM з RBF ядром досягає F1-score близько 0.86 на датасетах з 5000 резюме [23]. Особливо добре SVM працює для бінарної класифікації, наприклад, при визначенні, чи відповідає кандидат базовим вимогам вакансії. Для багатокласової класифікації зазвичай використовується стратегія, де будується кілька бінарних класифікаторів.

Обмеженням SVM є чутливість до вибору гіперпараметрів та відносно повільна швидкість навчання на великих датасетах. Для датасетів з більш ніж 10000 зразками час навчання може ставати неприйнятно довгим. Також SVM погано справляється з дуже незбалансованими класами, хоча ця проблема може бути частково вирішена використанням зважених класів.

Дослідження демонструють, що ансамбль Баєса та логістичної реграсії може досягати точності близько 87% [3], що є покращенням порівняно з окремими моделями. Однак ансамблеві методи вимагають більше обчислювальних ресурсів як для навчання, так і для передбачення, оскільки потрібно виконувати інференс для кількох моделей.

Важливим аспектом традиційних методів є препроцесинг тексту. Стандартний процес включає токенізацію, зведення до іншого регістру, видалення пунктуаційних символі та стоп-слів, стемінг або лематизацію. Стемінг відсікає закінчення слів для отримання основи, тоді як лематизація приводить слова до їх словникової форми з урахуванням морфології. Дослідження показують, що правильний препроцесинг може покращити результати на 3-5% [19].

Використання n-грам дозволяє захопити деякий контекст та фрази, які є важливими для класифікації. Біграми можуть ідентифікувати фрази на кшталт "machine learning" або "project management", які мають специфічне значення. Триграми захоплюють ще довші фрази, але значно збільшують розмірність простору ознак. Дослідження показують, що комбінація уніграм, біграм та триграм може покращити точність на 3-4% порівняно з використанням лише уніграм [21].

1.3.2 Методи глибокого навчання

Методи глибокого навчання революціонізували обробку природної мови, дозволяючи автоматично вивчати ієрархічні представлення тексту без необхідності ручного інжинірингу ознак. Ці методи базуються на нейронних мережах з кількома шарами, які послідовно трансформують вхідні дані для отримання все більш абстрактних представлень.

Після згорткових шарів зазвичай застосовується max-pooling, який вибирає найбільш важливі ознаки з кожного фільтра. Це робить представлення інваріантним до точного розташування патерну в тексті. Наприклад, фраза "expert in Python" буде виявлена незалежно від того, де вона знаходиться в резюме. Дослідження показують, що CNN досягає F1-score близько 0.88 для класифікації навичок у резюме [27].

Двосторонні LSTM обробляють послідовність як у прямому, так і у зворотному напрямках, що дозволяє враховувати як попередній, так і наступний контекст для кожного слова. Це особливо корисно для задач класифікації тексту, де розуміння повного контексту є важливим. Дослідження демонструють, що BiLSTM досягає F1-score близько 0.91 на задачах класифікації резюме [29].

Обмеженням рекурентних мереж є послідовна природа обробки, яка ускладнює паралелізацію та робить навчання повільним на довгих послідовностях. Також RNN все ще можуть мати труднощі з дуже довгими залежностями, хоча LSTM значно покращує ситуацію порівняно з базовими RNN.

1.3.3 Трансформерні моделі та BERT

Трансформерна архітектура змінила підхід до обробки природної мови. На відміну від рекурентних мереж, трансформери базуються виключно на механізмі уваги та не мають рекурентних з'єднань, що дозволяє ефективно паралелізувати обчислення.

Для застосування BERT до задачі класифікації використовується тонке налаштування - додатковий етап навчання на спеціалізованих даних. До попередньо навченої моделі додається класифікаційний шар, і вся модель дообучається на датасеті резюме. Дослідження показують, що BERT досягає точності близько 92% при класифікації резюме на 24 категорії [4].

Існують різні варіанти BERT, оптимізовані для різних сценаріїв використання. RoBERTa покращує процедуру навчання BERT, використовуючи більше даних та довше навчання. DistilBERT є компактною версією BERT, яка зберігає 97% якості при використанні лише 60% параметрів. ALBERT використовує факторизацію параметрів для зменшення розміру моделі.

1.3.4 Великі мовні моделі

Великі мовні моделі представляють найновіший етап розвитку методів обробки природної мови. Ці моделі навчені на величезних корпусах тексту та мають мільярди параметрів, що дозволяє їм демонструвати здібності до розуміння та генерації тексту.

Zero-shot навчаня дозволяє використовувати LLM для класифікації без будь-якого навчання на спеціалізованих даних. Модель отримує опис задачі та текст для класифікації у вигляді промпту. Дослідження показують, що LLM досягають F1-score близько 0.89 [28].

Few-shot навчання покращує результати, надаючи моделі кілька прикладів правильної класифікації в промпті. Це дозволяє моделі краще зрозуміти специфіку задачі та очікуваний формат відповіді. У простому режимі LLM можуть досягати F1-score до 0.93, що є конкурентоспроможним з повністю навченими спеціалізованими моделями [33, 34].

RAG підхід поєднує LLM з системою пошуку інформації для підвищення фактичної точності. Замість того, щоб покладатись виключно на параметричну пам'ять моделі, RAG спочатку знаходить релевантну інформацію в базі знань, а

потім використовує її як контекст для генерації відповіді. Це значно зменшує галюцинації та покращує надійність системи [37].

Мультиагентні системи на основі LLM використовують кілька спеціалізованих агентів для різних аспектів аналізу резюме. Один агент може фокусуватись на технічних навичках, інший - на досвіді роботи, третій - на освіті. Результати всіх агентів інтегруються для прийняття фінального рішення. Такий підхід дозволяє досягти точності близько 91% при збереженні можливості пояснення рішень [34].

1.4 Мета та постановка задачі

Відповідно до проведеного аналізу сформульовано мету та задачі дослідження.

Мета роботи полягає у підвищенні точності класифікації резюме за професійними категоріями шляхом розробки методу з використанням векторного представлення тексту на основі уніграм та біграм і нейронних мереж прямого поширення.

Задачі дослідження:

– провести аналіз існуючих методів та підходів до класифікації текстових документів з використанням методів машинного навчання та обробки мови;

– розробити метод класифікації резюме за професійними категоріями з використанням векторного представлення тексту TF-IDF та нейронних мереж;

– розробити програмну реалізацію методу класифікації резюме за професійними категоріями з використанням машинного навчання;

– провести експериментальне дослідження ефективності спроектованого методу шляхом порівняння базової та модифікованої моделей та оцінки їх точності класифікації на датасеті резюме.

Розділ 2 Метод класифікації резюме за професійними категоріями та критерії його оцінювання

2.1 Концепція та схема методу класифікації резюме

Автоматизована класифікація резюме є складною задачею обробки природної мови, яка потребує врахування багатьох факторів. У резюме міститься неструктурована текстова інформація, яка може бути представлена в різних форматах та стилях написання. Основна мета запропонованого методу полягає в тому, щоб автоматично визначити професійну категорію кандидата на основі аналізу текстового контенту його резюме.

Концепція методу базується на використанні методів машинного навчання для класифікації текстових документів. Резюме розглядається як текстовий документ, який містить базову інформацію про досвід професійної сфери, навички та освіту кандидата. Ця інформація є індикатором належності до певної професійної категорії.

Загальна ідея методу полягає в послідовній обробці текстового контенту резюме через кілька етапів. На першому етапі робитися попередня обробка тексту, яка включає очищення від непотрібних символів та приведення до стандартного формату. Другий етап передбачає витягування ключових ознак з тексту за допомогою методів векторизації. На третьому етапі уже робимо класифікацію за допомогою навченої моделі.

Основні етапи обробки даних включають завантаження та первинний аналіз резюме, токенізацію тексту, видалення стоп-слів, лематизацію слів, векторне представлення тексту та безпосередньо класифікацію. Кожен з цих етапів виконує специфічну функцію та вносить свій вклад у загальну якість роботи методу.

Схема роботи методу представлена на рисунку 2.1. Вхідними даними є текст резюме в необробленому вигляді. Спочатку текст проходить через блок препроцесингу, де виконується його очищення та нормалізація. Потім оброблений текст надходить до блоку витягування ознак, де формується векторне представлення документа. Далі вектор ознак подається на вхід навченої моделі класифікації, яка

визначає професійну категорію. На виході метод повертає передбачену категорію разом з ймовірністю віднесення до неї.

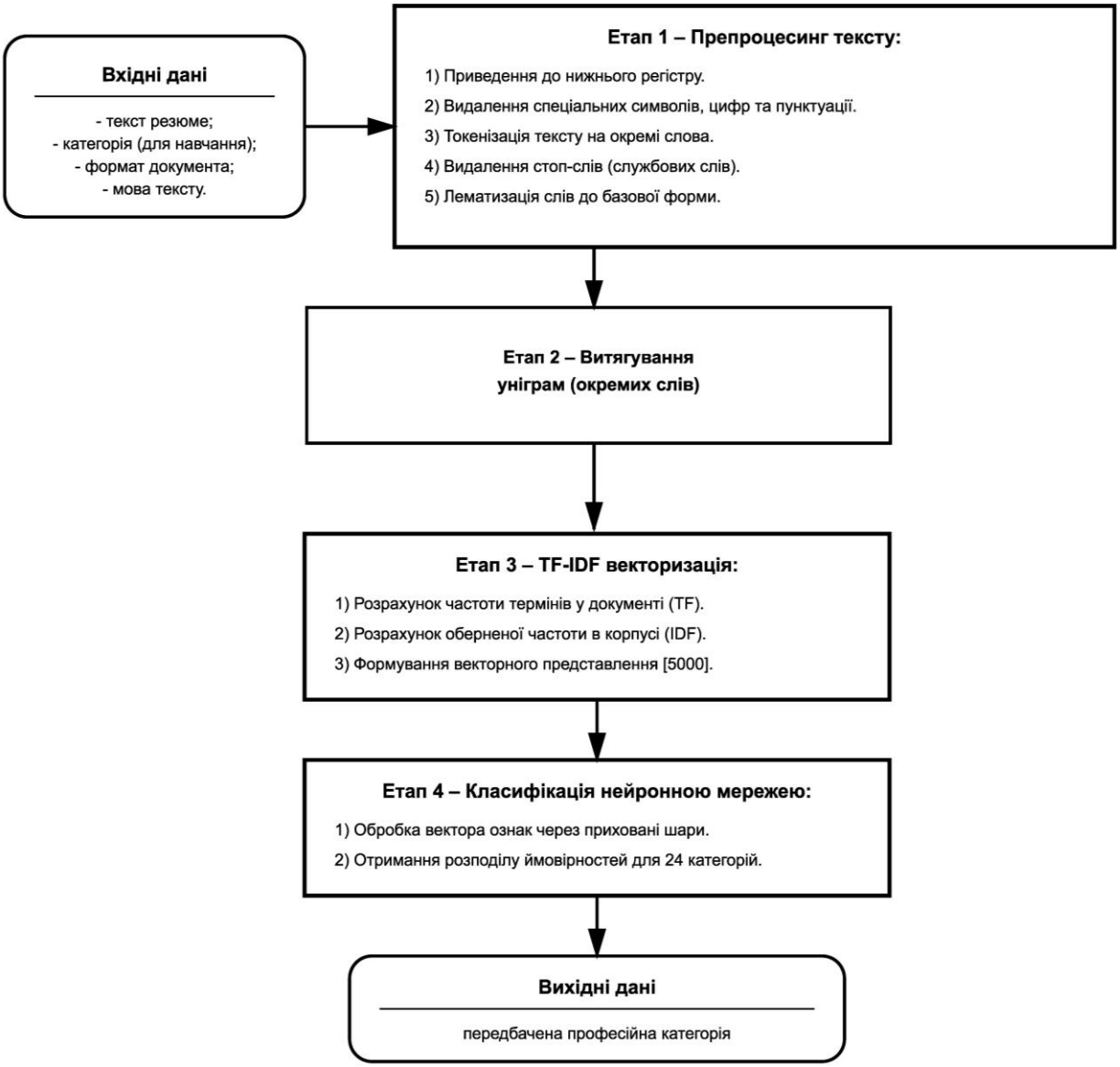


Рисунок 2.1 – Схема роботи методу класифікації резюме

Важливою особливістю запропонованого методу є можливість обробки резюме різних форматів та структур. Метод не залежить від конкретного шаблону або форми представлення інформації в резюме, оскільки аналізує безпосередньо текстовий контент. Це робить його універсальним та придатним для обробки великих обсягів різноманітних резюме.

Метод розроблений таким чином, щоб забезпечити баланс між точністю класифікації та швидкістю обробки. Використання ефективних алгоритмів препроцесингу та оптимізованих методів векторизації дозволяє обробляти резюме в реальному часі, що є важливим для практичного застосування в системах автоматизованого рекрутингу.

2.2 Архітектура моделі класифікації

Для вирішення задачі класифікації резюме обрано гібридний підхід, який поєднує традиційні методи машинного навчання з елементами глибокого навчання. Такий підхід дає змогу отримати досить добрі результати при відносно невеликих обчислювальних ресурсах та обсягах навчальних даних.

Базову архітектуру моделі складено з кількох послідовних компонентів. В основу взято прямопоширену нейрону мережу (Feedforward Neural Network). Перший компонент відповідає за векторне представлення тексту за допомогою методу TF-IDF. Цей метод обраний через його простоту, інтерпретованість та здатність добре працювати на текстових даних середнього розміру. TF-IDF дозволяє перетворити текст резюме на числовий вектор фіксованої довжини, де кожна компонента відображає важливість конкретного слова в документі відносно всього корпусу резюме.

Другий компонент архітектури представлений нейронною мережею прямого поширення. Мережа складається з вхідного шару, який приймає вектор TF-IDF ознак, двох прихованих шарів з нелінійними функціями активації та вихідного шару з функцією softmax для багатокласової класифікації. Використання нейронної мережі замість простих лінійних класифікаторів дозволяє моделі виявляти нелінійні залежності між ознаками та класами.

Вхідний шар мережі має розмірність, що дорівнює кількості унікальних термінів у словнику TF-IDF. У експериментах використовувався словник найбільш частотних слів, що забезпечує достатнє покриття лексики резюме при збереженні керованої розмірності моделі.

Перший прихований шар містить 256 нейронів з функцією активації ReLU. Функція ReLU обрана через її простоту обчислення та здатність ефективно навчатись. Цей шар виконує роль первинного витягування високорівневих ознак з вхідного векторного представлення. Після першого прихованого шару застосовується dropout з коефіцієнтом 0.3 для запобігання перенавчанню моделі.

Другий прихований шар має 128 нейронів також з функцією активації ReLU. Зменшення кількості нейронів у другому шарі створює ефект воронки, який примушує модель вивчати більш компактне та узагальнене представлення даних. Після цього шару також використовується dropout з коефіцієнтом 0.3.

Вихідний шар містить кількість нейронів, що дорівнює кількості професійних категорій у датасеті. Функція активації softmax на вихідному шарі забезпечує інтерпретацію виходів як ймовірностей належності резюме до кожної з категорій. Сума всіх виходів дорівнює одиниці, що дозволяє розглядати їх як розподіл ймовірностей.

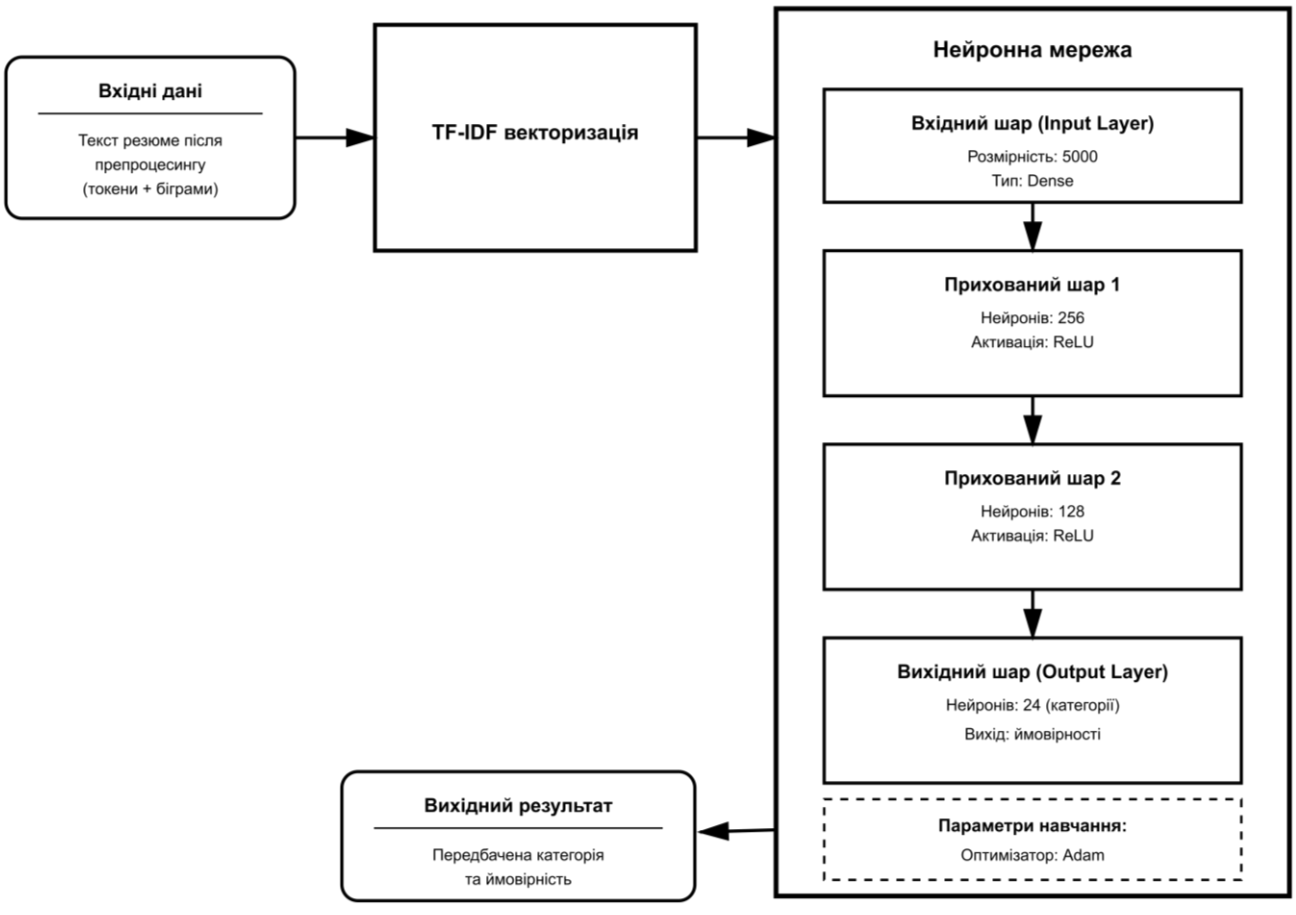


Рисунок 2.2 – Архітектура моделі класифікації резюме

Для навчання моделі використовується функція втрат, яка є стандартним вибором для задач багатокласової класифікації. Ця функція втрат вимірює різницю між передбаченим розподілом ймовірностей та справжньою міткою класу. Мінімізація цієї функції призводить до покращення точності класифікації моделі.

В якості оптимізатора обрано алгоритм Adam з початковою швидкістю навчання 0.001. Оптимізатор Adam поєднує переваги двох інших популярних методів оптимізації – AdaGrad та RMSProp. Він лає змогу налаштовувати саму швидкість навчання для будь кого параметра моделі, що призводить до швидшої та стабільнішої збіжності під час навчання.

Регуляризація моделі здійснюється за допомогою dropout. Техніка dropout під час навчання випадково вимикає частину нейронів з ймовірністю, вказаною в параметрі dropout rate. У нашому випадку використовується коефіцієнт 0.3, що означає, що кожен нейрон має 30% шанс бути тимчасово вимкненим під час тренування. Це змушує мережу не покладатися на окремі нейрони та розвивати більш стійкі представлення даних.

2.3 Модифікація моделі та покращення векторного представлення тексту

Базова архітектура, описана в попередньому підрозділі, забезпечує прийнятну точність класифікації, однак аналіз помилок моделі показав, що вона може мати труднощі з розпізнаванням контексту окремих термінів. Наприклад, слово "manager" може зустрічатися в резюме як IT-менеджерів, так і менеджерів з продажу, але контекст його використання буде різним.

Для вирішення цієї проблеми запропоновано модифікацію, яка полягає в додаванні інформації про біграми до векторного представлення тексту. Біграми - це пари послідовних слів у тексті, які зберігають інформацію про локальний контекст. Наприклад, біграм "project manager" несе більше інформації, ніж окремі слова "project" та "manager".

Обґрунтування необхідності модифікації базується на дослідженні помилок базової моделі. Аналіз показав, що приблизно 15-20% помилок класифікації виникають через неправильне розуміння контексту професійних термінів. Додавання біграм дозволяє моделі краще захоплювати такі контексти та розрізняти схожі, але різні професійні категорії.

Запропоноване покращення реалізується шляхом розширення словника TF-IDF. Замість використання лише уніграм (окремих слів), створюється комбінований словник, який включає як уніграми, так і найбільш частотні біграми. Розмір словника залишається обмеженим 5000 елементами, але тепер він містить приблизно 4000 уніграм та 1000 біграм.

Відбір біграм для включення до словника здійснюється на основі їх частоти появи в навчальних даних та інформаційної цінності. Використовується метрика pointwise mutual information (PMI), яка вимірює, наскільки часто два слова зустрічаються разом порівняно з тим, як часто вони зустрічаються окремо. Біграми з високим значенням PMI мають більшу ймовірність бути включеними до словника, оскільки вони представляють стійкі словосполучення, характерні для певних професійних категорій.

Процес формування комбінованого словника відбувається в кілька кроків. Спочатку з навчальних даних витягуються всі можливі уніграми та біграми. Потім для кожного біграму розраховується значення PMI. Біграми сортуються за значенням PMI, і топ-1000 найбільш інформативних біграм відбираються для включення до словника разом з 4000 найчастотніших уніграм.

Модифікований блок векторизації працює наступним чином. Вхідний текст резюме проходить через стандартну процедуру препроцесингу, після чого з нього витягуються як окремі слова, так і послідовні пари слів. Для кожного елементу словника (уніграму чи біграму) розраховується TF-IDF значення, яке відображає його важливість в даному резюме. Результатом є вектор розмірності 5000, де кожна компонента відповідає або окремому слову, або біграму.

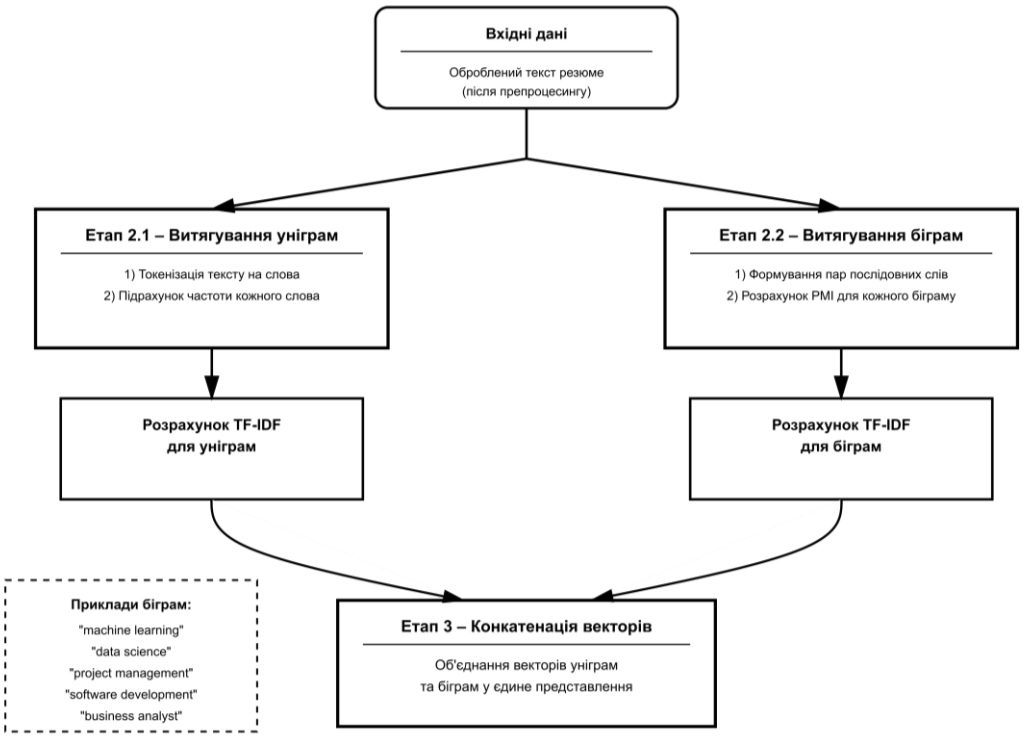


Рисунок 2.3 – Схема модифікованого блоку векторизації

Важливою особливістю модифікації є те, що вона не вимагає змін в архітектурі нейронної мережі. Розмірність вхідного вектору залишається такою ж 5000 елементів, тому всі параметри мережі зберігаються без змін. Це дозволяє легко порівнювати базову та модифіковану версії моделі, оскільки вони відрізняються лише способом формування вхідних ознак.

Додавання біграм до векторного представлення має кілька переваг. По-перше, біграми дозволяють захопити стійкі словосполучення, які часто використовуються в резюме певних професійних категорій. Наприклад, біграм "machine learning" є сильним індикатором категорії Data Science, тоді як окремі слова "machine" та "learning" можуть зустрічатися в різних контекстах.

Використання біграм частково вирішує проблему багатозначності окремих слів. Слово "development" може означати розробку програмного забезпечення, особистий розвиток, бізнес-розвиток тощо. Однак біграми "software development", "business development" чітко вказують на різні професійні області.

Біграми дозволяють моделі краще розрізняти схожі професійні категорії. Наприклад, категорії "Sales" та "Marketing" часто містять схожі слова, але біграми,

характерні для кожної з них, відрізняються. Для Sales типовими є "sales target", "cold calling", тоді як для Marketing - "brand awareness", "content marketing".

Недоліком додавання біграм є незначне збільшення часу обробки та обсягу пам'яті, необхідної для зберігання словника. Однак в абсолютних числах це збільшення є незначним - час векторизації зростає приблизно на 15-20%, що є прийнятним компромісом за покращення точності класифікації.

Модифікація також впливає на інтерпретованість моделі. Наявність біграм у найважливіших ознаках робить рішення моделі більш зрозумілими для людини. Замість абстрактних окремих слів можна побачити конкретні професійні терміни та фрази, які вплинули на класифікацію.

2.4 Формування та підготовка навчальних даних

Для навчання та оцінювання методу класифікації резюме використовується датасет Resume Dataset, доступний на платформі Kaggle. Цей датасет є одним із найбільших публічно доступних наборів даних для задач автоматизованої обробки резюме та широко використовується в дослідницьких роботах.

Датасет складається з понад 2400 резюме, розподілених між 24 професійними категоріями. Категорії охоплюють широкий спектр професійних напрямків, включаючи HR, Designer, Information Technology, Teacher, Advocate, Business Development, Healthcare, Fitness, Agriculture, BPO, Sales, Consultant, Digital Media, Automobile, Chef, Finance, Apparel, Engineering, Accountant, Construction, Public Relations, Banking, Arts та Aviation.

Кожне резюме в датасеті представлене у трьох форматах. Є версія у вигляді чистого тексту, яка зберігається в колонці Resume_str файлу CSV. Існує HTML-версія в колонці Resume_html, яка містить структуровану інформацію з форматуванням. Оригінальні PDF-файли резюме зберігаються в окремих папках, організованих за категоріями. Для навчання моделі використовується текстова версія, оскільки вона вже очищена від форматування та готова для обробки.

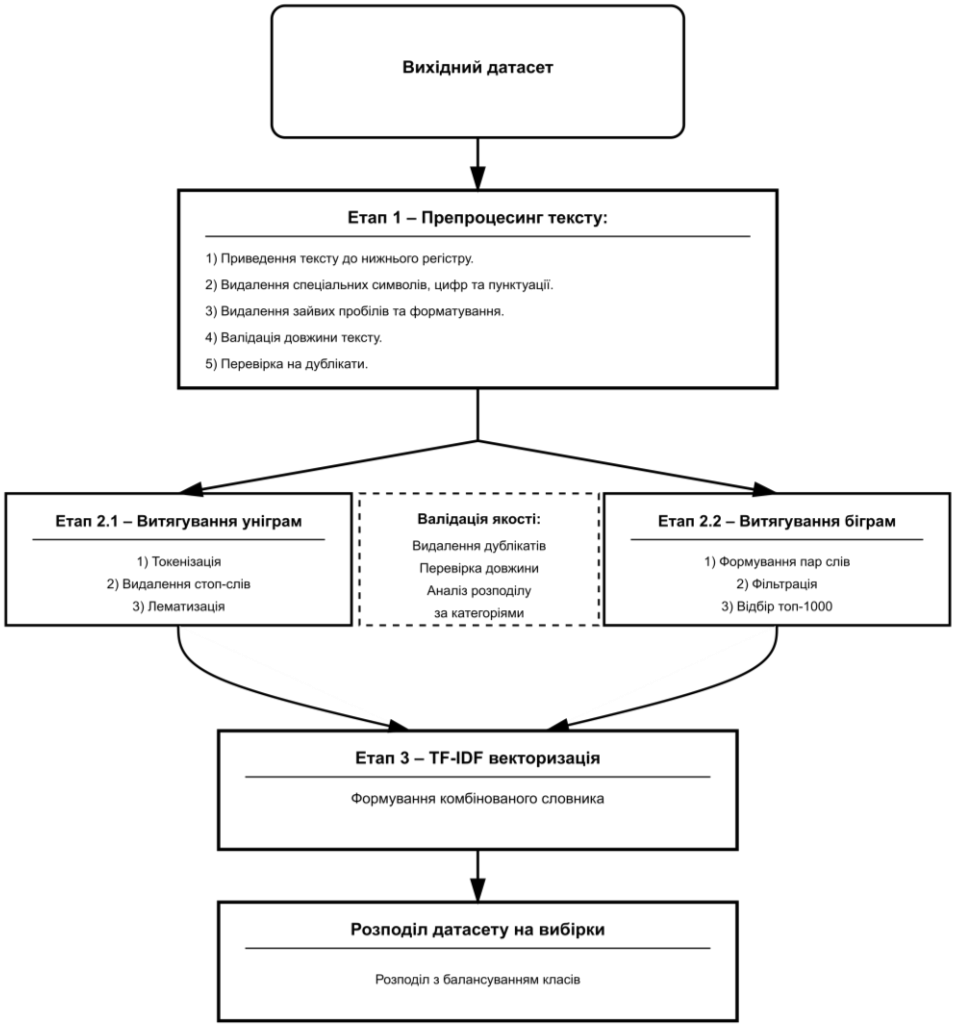


Рисунок 2.4 – Процес обробки даних

Характеристика датасету показує, що резюме були зібрані з платформи LiveCareer.com шляхом веб-скрапінгу. Це означає, що вони представляють реальні приклади професійно складених резюме, які використовуються кандидатами при подачі заявок на роботу. Така природа даних робить датасет репрезентативним для практичних задач автоматизованої обробки резюме.

Розподіл резюме за категоріями є відносно збалансованим порівняно з багатьма іншими датасетами. Більшість категорій містять від 80 до 120 зразків, що забезпечує потрібну кількість необхідних даних для навчання моделі. Однак деякі категорії можуть мати трохи більше або менше резюме, що є типовим для реальних задач класифікації.

Середня довжина резюме в датасеті становить приблизно 400-700 слів після препроцесингу. Це відповідає стандартним рекомендаціям щодо довжини професійного резюме, яке зазвичай займає одну-дві сторінки. Резюме написані англійською мовою та дотримуються загальноприйнятих стандартів оформлення професійних документів.

Перед використанням для навчання моделі масив даних проходить через кілька етапів препроцесингу. Метою препроцесингу є очищення та нормалізація текстових даних для покращення якісних показників навчання моделі та забезпечення її стабільної роботи на різних типах резюме.

Перший етап препроцесингу полягає в приведенні всього тексту до нижнього регістру. Це стандартна практика в обробці мови, яка дозволяє уникнути ситуації, коли модель розглядає слова "Python" та "python" як різні терміни. Приведення до нижнього регістру робить обробку тексту більш консистентною та зменшує розмірність простору ознак.

Другий етап включає видалення спеціальних символів, цифр та пунктуації. У резюме часто зустрічаються різні форматувальні символи, номери телефонів, поштові індекси, дати та інша інформація, яка не несе смислового навантаження для визначення професійної категорії. Наприклад, цифри в датах чи номерах телефонів не допомагають розрізнити, чи є кандидат інженером чи дизайнером.

Третій етап - це токенізація тексту, а саму розбиття його на різні окремі слова. Для токенізації використовується бібліотека NLTK, яка забезпечує коректне розбиття тексту з урахуванням особливостей англійської мови. Токенізатор правильно обробляє скорочення, апострофи та інші специфічні випадки. Результатом токенізації є впорядкований список слів для кожного резюме.

Четвертий етап передбачає видалення стоп-слів. Стоп-слова - це часто вживані службові слова англійської мови, такі як "the", "is", "at", "which", "and", "or", які не несуть специфічної інформації про професійну категорію. Видалення цих слів зменшує розмірність простору ознак приблизно на третину та дозволяє моделі зосередитись на змістовних професійних термінах.

П'ятий етап - це лематизація слів за допомогою WordNet Lemmatizer. Лематизація приводить слова до їх певної словникової форми з урахуванням контексту та частини мови. Наприклад, дієслова "developing", "developed", "develops" приводяться до базової форми "develop", а іменники "analyses", "analysis" - до форми "analysis". Це дозволяє моделі розглядати різні граматичні форми одного слова як єдиний термін.

Паралельно з обробкою окремих слів виконується формування біграм. Біграми - це пари послідовних слів у тексті, які часто утворюють стійкі словосполучення. Наприклад, "machine learning", "data science", "project management" є біграмами, які несуть важливу інформацію про професійну сферу кандидата. Для кожного резюме генерується список біграм, які потім фільтруються за частотою появи.

Результатом препроцесингу є очищений та нормалізований текст у вигляді списку токенів та списку біграм для кожного резюме. Ці дані потім використовуються для формування векторного представлення за допомогою методу TF-IDF, який враховує як окремі слова, так і біграми.

Формування ознак відбувається окремо для навчальної, валідаційної та тестової вибірок. Критично важливо, що словник TF-IDF будується виключно на основі навчальної вибірки. Це значить, що певний список термінів та їх статистики розраховуються тільки з навчальних даних. Такий підхід запобігає витоку інформації з тестових даних у процес навчання та забезпечує чесну оцінку узагальнювальної здатності моделі.

Датасет розділяється на три частини згідно зі стандартною практикою машинного навчання. Навчальна вибірка становить 70 % даних, приблизно 1680 резюме. Вона використовується для того, щоб використати застосування параметрів нейронної мережі через алгоритм зворотного поширення помилки. Валідаційна вибірка становить 15 %, близько 360 резюме, та застосовується для моніторингу процесу навчання і запобігання перенавчанню. Тестова вибірка також становить 15 %, приблизно 360 резюме, та зберігається для фінальної оцінки моделі.

Для забезпечення репрезентативності кожної вибірки використовується стратифікований розподіл за допомогою функції train_test_split з бібліотеки scikit-learn. Цей метод гарантує, що пропорції професійних категорій зберігаються в кожній з трьох частин датасету. Якщо категорія Information Technology становить 8 % від загального датасету, то вона буде становити приблизно 8 % і в навчальній, і в валідаційній, і в тестовій вибірках.

Проблема можливої незбалансованості класів вирішується шляхом використання зваженої функції втрат під час навчання. Бібліотека Keras дозволяє автоматично розрахувати ваги для кожного класу обернено пропорційно їх частоті в навчальних даних. Це означає те, що різні помилки класифікації на рідкісних категоріях отримують більшу вагу при обчисленні загальної функції втрат, що стимулює модель приділяти більше уваги цим категоріям.

Для категорій з малою кількістю зразків застосовується проста аугментація даних. Аугментація реалізована через синонімічну заміну окремих слів у резюме з використанням WordNet Synsets. Алгоритм випадково обирає декілька слів у резюме та замінює їх на синоніми зі збереженням загального смислу тексту. Така процедура дозволяє збільшити розмаїття навчальних прикладів без збору додаткових даних.

Важливим аспектом підготовки даних є валідація їх якості. Перед початком навчання виконується перевірка на наявність дублікатів резюме за допомогою порівняння текстових рядків. Виявлені дублікати видаляються для запобігання витоку інформації між навчальною та тестовою вибірками. Також перевіряються резюме на наявність порожніх або майже порожніх записів.

Резюме, що містять менше 50 слів після препроцесингу, видаляються з датасету як непридатні для класифікації. Такі короткі резюме не містять достатню кількість необхідної інформації для надійного визначення належних категорії. Аналіз показав, що таких випадків у датасеті Resume Dataset є менше 1 %, що говорить про високу якість зібраних даних.

Також проводиться аналіз розподілу довжин резюме в різних категоріях для виявлення потенційних особливостей даних. Статистика показує, що резюме технічних спеціальностей, таких як Information Technology та Engineering, в

середньому довші за резюме творчих професій, таких як Arts або Designer. Це пояснюється необхідністю детального опису технічних навичок та проєктів.

Для кращого розуміння структури даних виконується експлораторний аналіз частотності термінів. Для кожної професійної категорії визначаються найчастотніші слова після видалення стоп-слів. Це дозволяє побачити, які терміни є характерними для кожної категорії. Наприклад, для категорії Healthcare типовими є слова "patient", "medical", "care", "clinical", тоді як для категорії Information Technology - "software", "development", "programming", "system".

Аналіз найбільш дискримінативних біграм показує, що стійкі словосполучення є особливо корисними для розрізнення схожих категорій. Біграми "business development", "business analyst", "business intelligence" чітко вказують на різні, хоча й пов'язані професійні ролі в бізнес-сфері.

Створений процес обробки даних реалізований таким чином, щоб забезпечити легку відтворюваність експериментів. Всі параметри препроцесингу, такі як розмір словника, мінімальна частота термінів, коефіцієнт розподілу на вибірки, зберігаються в конфігураційному файлі. Це дозволяє швидко експериментувати з різними налаштуваннями та порівнювати результати.

Результатом етапу формування та підготовки даних є три набори векторизованих резюме з відповідними мітками професійних категорій. Навчальна вибірка готова для навчання моделі, валідаційна - для контролю процесу навчання, тестова - для об'єктивної оцінки кінцевої якості методу. Всі дані пройшли повний цикл очищення, нормалізації та трансформації у формат, придатний для використання нейронною мережею.

2.5 Критерії та метрики оцінювання роботи методу

Для комплексної оцінки якості роботи методу класифікації резюме використовується набір стандартних метрик, які дозволяють проаналізувати різні аспекти роботи моделі. Вибір метрик обумовлений специфікою задачі

багатокласової класифікації та необхідністю врахування можливої незбалансованості класів у датасеті.

Основною метрикою оцінювання є точність класифікації, яка показує загальну частку правильно класифікованих резюме. Ця метрика обчислюється як відношення кількості резюме, для яких модель правильно визначила професійну категорію, до загальної кількості резюме в тестовій вибірці. Точність є інтуїтивно зрозумілою метрикою, яка дає загальне уявлення про якість моделі.

Не дивлячись на простоту інтерпретації, точність має суттєві обмеження при роботі з незбалансованими датасетами. Якщо в датасеті одна категорія значно переважає інші, модель може досягти високої точності просто передбачаючи найчастішу категорію для більшості резюме. Наприклад, якщо категорія Information Technology становить 30 % датасету, модель, яка завжди передбачає цю категорію, отримає точність 30 % без будь-якого реального навчання. Тому точність доповнюється іншими метриками для більш об'єктивної оцінки.

Точність передбачення для кожної категорії показує, наскільки можна довіряти позитивним передбаченням моделі для цієї категорії. Вона розраховується як відношення кількості резюме, правильно віднесених до категорії, до загальної кількості резюме, які модель віднесла до цієї категорії. Висока точність передбачення означає, що коли модель класифікує резюме в певну категорію, вона рідко помиляється. Наприклад, якщо модель класифікувала 100 резюме як належні до категорії Data Science, і з них 85 дійсно є резюме спеціалістів з Data Science, то точність передбачення для цієї категорії становить 85 %. Це означає, що в 85 випадках зі 100 можна довіряти рішенню моделі про приналежність резюме до категорії Data Science. Повнота для кожної категорії вимірює, наскільки добре модель знаходить всі резюме цієї категорії. Вона обчислюється як відношення кількості правильно ідентифікованих резюме категорії до загальної кількості резюме цієї категорії в тестовій вибірці. Висока повнота означає, що модель пропускає мало резюме певної категорії.

Продовжуючи попередній приклад, якщо в тестовій вибірці було 120 резюме категорії Data Science, а модель правильно ідентифікувала лише 85 з них, то повнота

становить приблизно 71 %. Це означає, що модель пропустила 29 % резюме спеціалістів з Data Science, класифікувавши їх в інші категорії.

F1-міра є гармонічним середнім між точністю передбачення та повнотою. Вона забезпечує збалансовану оцінку, яка враховує обидва аспекти якості класифікації. F1-міра досягає високого значення тільки тоді, коли обидві метрики - і точність передбачення, і повнота - мають високі значення. Якщо одна з метрик низька, F1-міра також буде низькою, навіть якщо інша метрика висока.

F1-міра особливо корисна при оцінюванні якості класифікації на незбалансованих датасетах. Вона не дозволяє моделі досягти високого значення метрики просто за рахунок передбачення найчастішої категорії. Модель повинна демонструвати як високу точність передбачень, так і здатність знаходити більшість зразків кожної категорії.

Для багатокласової класифікації з 24 категоріями використовуються агреговані версії метрик, які узагальнюють результати по всіх категоріях. Макро-усереднена оцінка обчислює метрику окремо для кожної категорії, а потім усереднює результати без урахування розміру категорій. Цей підхід однаково враховує всі категорії незалежно від їх представленості в датасеті.

Макро-усереднена F1 особливо корисна для оцінювання того, наскільки добре модель працює на рідкісних категоріях. Якщо модель добре класифікує часті категорії, але погано працює на рідкісних, макро-усереднена F1 буде відносно низькою, оскільки погані результати на рідкісних категоріях мають таку ж вагу, як і хороші результати на частих.

Зважено усереднена оцінка обчислює метрику для кожної категорії окремо, але потім зважує результати пропорційно кількості зразків у кожній категорії в тестовій вибірці. Цей підхід краще відображає загальну якість класифікації з урахуванням реального розподілу категорій у даних.

Зважено усереднена F1 дає більшу вагу результатам на частих категоріях і меншу на рідкісних. Ця метрика корисна для розуміння того, яку якість класифікації можна очікувати в середньому при використанні моделі на реальних даних з подібним розподілом категорій.

Матриця плутанини використовується для детального аналізу помилок моделі. Це таблиця розміром 24 на 24 елементи, де рядки відповідають справжнім категоріям резюме, а стовпці - передбаченим категоріям. Елемент на перетині рядка i та стовпця j показує, скільки резюме справжньої категорії і були класифіковані моделлю як категорія j.

Діагональні елементи матриці плутанини відповідають правильним класифікаціям - коли передбачена категорія збігається зі справжньою. Недіагональні елементи показують помилки різних типів. Великі значення недіагональних елементів вказують на те, що модель систематично плутає певні пари категорій між собою. Аналіз матриці плутанини дозволяє виявити категорії, які найчастіше плутаються між собою. Наприклад, якщо резюме категорії Sales часто класифікуються як Business Development і навпаки, це може вказувати на схожість термінології цих професійних сфер. Така інформація може бути використана для подальшого покращення моделі, наприклад, через додавання специфічних ознак, які краще розрізняють ці категорії.

Візуалізація матриці плутанини у вигляді теплової карти робить аналіз помилок більш наочним. Темні комірки вказують на велику кількість резюме, світлі - на малу. Ідеальна матриця плутанини має темну діагональ і світлі недіагональні елементи, що означає високу точність і мало помилок. Додатковим важливим критерієм є час обробки одного резюме. Цей показник вимірюється в мілісекундах і включає весь цикл обробки: препроцесинг тексту, токенізацію, лематизацію, векторизацію методом TF-IDF та безпосередньо класифікацію нейронною мережею. Час обробки є критичним для практичного застосування моделі в системах автоматизованого рекрутингу.

Для великих компаній, які отримують сотні або тисячі резюме щодня, швидкість обробки може бути таким же важливим критерієм, як і точність класифікації. Модель повинна обробляти резюме достатньо швидко, щоб не створювати затримок у процесі підбору персоналу. Зазвичай прийнятним вважається час обробки менше однієї секунди на одне резюме.

Оцінюється використання пам'яті моделлю під час класифікації. Модель повинна мати достатньо компактний розмір, щоб завантажуватись у пам'ять типового сервера разом з іншими сервісами. Великі моделі, які потребують десятки гігабайтів пам'яті, є менш практичними для впровадження в реальних системах.

Обґрунтування вибору метрик базується на практичних вимогах до системи класифікації резюме. Точність забезпечує загальне розуміння якості моделі та є зрозумілою для нетехнічних користувачів. Точність передбачення важлива для мінімізації помилкових позитивних результатів, коли резюме класифікується в неправильну категорію. Повнота забезпечує, що модель не пропускає релевантних кандидатів, що критично для уникнення втрати потенційно підхідних співробітників. F1-міра надає збалансовану оцінку, яка враховує обидва аспекти якості. Макро та зважена F1 дозволяють оцінити роботу моделі на рідкісних та частих категоріях відповідно. Матриця плутанини надає детальну інформацію про характер помилок моделі, що допомагає в її подальшому вдосконаленні. Час обробки та використання пам'яті є критичними для практичного впровадження системи в реальне середовище. Для порівняння різних варіантів моделі використовується комплексний підхід, який враховує всі зазначені метрики одночасно. Головна увага приділяється зваженому F1-score, оскільки ця метрика найкраще відображає реальну якість класифікації з урахуванням розподілу категорій. Однак також важливо, щоб макро F1-score був достатньо високим, що гарантує прийнятну якість на рідкісних категоріях.

Стабільність метрик перевіряється за допомогою п'ятикратної крос-валідації на навчальній вибірці. Датасет розбивається на п'ять частин, і модель навчається п'ять разів, кожного разу використовуючи чотири частини для навчання і одну для валідації. Якщо метрики сильно коливаються між різними розбиттями, це може вказувати на нестабільність моделі або специфічні особливості окремих підмножин даних.

Статистична значущість різниці між моделями оцінюється за допомогою парного t-тесту на результатах крос-валідації. Це дозволяє визначити, чи є покращення метрик при використанні модифікованого методу статистично

значущим, а не випадковим коливанням через специфіку конкретного розбиття даних.

Висновок до розділу 2

У другому розділі представлено опис методу класифікації резюме за професійними категоріями з використанням машинного навчання. Розроблена концепція методу базується на послідовній обробці текстового контенту резюме через етапи препроцесингу, витягування ознак та класифікації за допомогою нейронної мережі.

Архітектура моделі поєднує векторизацію тексту методом TF-IDF з нейронною мережею прямого поширення, що включає два приховані шари. Використання dropout-регуляризації та оптимізатора Adam забезпечує стабільне навчання моделі та запобігає перенавчанню.

Запропоновано модифікацію базової моделі шляхом додавання біграм до векторного представлення тексту. Це покращення дозволяє моделі краще захоплювати контекст професійних термінів та розрізняти схожі професійні категорії за рахунок врахування стійких словосполучень, характерних для різних професійних сфер.

Детально описано процес формування та підготовки навчальних даних на основі датасету Resume Dataset з понад 2400 резюме в 24 професійних категоріях. Процес обробки включає п'ять послідовних етапів препроцесингу тексту та забезпечує консистентну підготовку даних для навчання та використання моделі.

Визначено комплексний набір метрик для оцінювання якості роботи методу, який включає точність класифікації, точність передбачення, повноту, F1-міру та час обробки. Обрані метрики дозволяють всебічно оцінити різні аспекти роботи моделі з урахуванням специфіки задачі багатокласової класифікації.

Представлений метод забезпечує баланс між точністю класифікації та обчислювальною складністю, що робить його придатним для практичного застосування в системах автоматизованого підбору персоналу.

Розділ 3 Програмна реалізація методу класифікації резюме

3.1 Обґрунтування вибору засобів та середовища розробки

Для втілення запропонованого у другому розділі методу класифікації резюме необхідно обрати програмні інструменти, які б забезпечили можливість ефективної роботи з текстовими даними великого обсягу та навчання нейронних мереж. Вибір технологій здійснювався з урахуванням специфіки задачі, доступності інструментів та їх поширеності у галузі обробки природної мови.

Як основну мову програмування обрано Python версії 3.10. Це рішення обґрунтовується низкою факторів. Python має найрозвиненішу екосистему бібліотек для машинного навчання в середовищі усіх мов програмування. Велика кількість досліджень не сьогодні у галузі обробки природної мови публікуються з відкритим кодом саме на Python, що полегшує впровадження перевірених підходів. Мова має добрий та зрозумілий синтаксис, що дозволяє швидко створювати прототипи та експериментувати з різними варіантами реалізації без написання великої кількості службового коду.

Python підтримує інтерактивний режим роботи через Jupyter Notebook, що досить сильно спрощує процес налагодження та аналізу проміжних результатів. Можливість виконувати код окремими блоками та одразу бачити результати особливо корисна при роботі з даними, коли потрібно швидко перевіряти різні гіпотези. Python забезпечує кросплатформність – код однаково працює на операційних системах Windows, Linux та macOS без необхідності модифікацій.

Для побудови та навчання нейронної мережі обрано бібліотеку Keras версії 2.13, яка працює поверх фреймворку TensorFlow 2.13. Keras надає високорівневий програмний інтерфейс для взаємодії з нейронними мережами, що дозволяє описувати архітектуру моделі у вигляді послідовності шарів без необхідності вручну визначати тензорні операції. Бібліотека автоматично обчислює градієнти для алгоритму зворотного поширення помилки та надає готові реалізації популярних оптимізаторів, функцій втрат та метрик.

TensorFlow як базовий фреймворк забезпечує оптимізовані обчислення на рівні тензорних операцій. Фреймворк автоматично розпаралелює обчислення на доступні ядра процесора та підтримує прискорення на графічних процесорах через технологію CUDA. Це критично важливо для навчання нейронних мереж, яке може тривати години або дні на звичайному процесорі. Використання графічного прискорювача дозволяє скоротити час навчання у десятки разів.

Обробка текстових даних реалізована за допомогою бібліотеки NLTK версії 3.8. Ця бібліотека є однією з найстаріших та найбільш перевірених у галузі обробки природної мови. NLTK містить готові інструменти для всіх необхідних операцій препроцесингу тексту. Токенізатор бібліотеки коректно розбиває англомовний текст на слова з урахуванням особливостей мови, таких як апострофи та скорочення. Лематизатор WordNet приводить слова до їх базової словникової форми, використовуючи великий лінгвістичний словник.

Бібліотека включає списки стоп-слів для багатьох мов, у тому числі для англійської, якою написані резюме у датасеті Resume Dataset. Список містить понад 170 найчастотніших службових слів, таких як артиклі, прийменники, сполучники та займенники, які зазвичай не несуть змістового навантаження для класифікації документів. NLTK також надає інструменти для морфологічного аналізу, які дозволяють визначати частини мови, що необхідно для коректної лематизації.

Перетворення тексту на векторне представлення методом TF-IDF здійснюється за допомогою бібліотеки scikit-learn версії 1.3. Ця бібліотека є стандартом для задач машинного навчання на структурованих даних. Клас TfidfVectorizer надає зручний інтерфейс для перетворення колекції текстових документів на матрицю числових ознак. Векторизатор автоматично будує словник термінів з навчальних даних, розраховує статистики частот та формує розріджені матриці для ефективного зберігання векторів великої розмірності.

Scikit-learn також використовується для розподілу датасету на підвибірки та розрахунку метрик класифікації. Функція train_test_split дозволяє розділити дані на частини зі збереженням пропорцій класів. Модуль metrics містить реалізації всіх необхідних метрик – точності класифікації, точності передбачення, повноти, F1-

міри та матриці плутанини. Метрики розраховуються як для окремих класів, так і з різними способами усереднення по всіх класах.

Робота з числовими масивами та математичні обчислення виконуються за допомогою бібліотеки NumPy версії 1.24. NumPy надає багатовимірні масиви як основну структуру даних та набір функцій для ефективних векторизованих операцій над ними. Всі операції реалізовані на мові низького рівня, що забезпечує швидкість виконання, порівнянну з мовами C та Fortran. Бібліотека інтегрована з усіма іншими інструментами – Keras, scikit-learn та Matplotlib працюють з NumPy масивами як з базовим форматом даних.

Для створення графіків та діаграм використовується бібліотека Matplotlib версії 3.7 разом з надбудовою Seaborn версії 0.12. Matplotlib дозволяє будувати лінійні графіки для відображення динаміки метрик під час навчання, стовпчасті діаграми для порівняння результатів різних моделей та теплові карти для візуалізації матриці плутанини. Seaborn надає додаткові можливості для створення статистичних графіків з покращеним зовнішнім виглядом порівняно з базовими можливостями Matplotlib.

Робота з даними у форматі CSV здійснюється за допомогою бібліотеки pandas версії 2.0. Датасет Resume Dataset зберігається у CSV файлі, який містить. Pandas дозволяє ефективно завантажувати такі файли у пам'ять та працювати з ними як з таблицями. Бібліотека надає зручні інструменти для фільтрації рядків, вибору стовпців, групування даних за категоріями та інших операцій аналізу даних.

3.2 Загальна структура програмного рішення та взаємодія компонентів

Програмне рішення для класифікації резюме організовано у вигляді модульної системи, де кожен модуль відповідає за виконання певної групи задач. Така організація забезпечує зрозумілість коду, полегшує тестування окремих компонентів та дозволяє модифікувати частини системи незалежно одна від одної. Загальна архітектура складається з п'яти основних модулів, які взаємодіють послідовно у процесі обробки резюме.

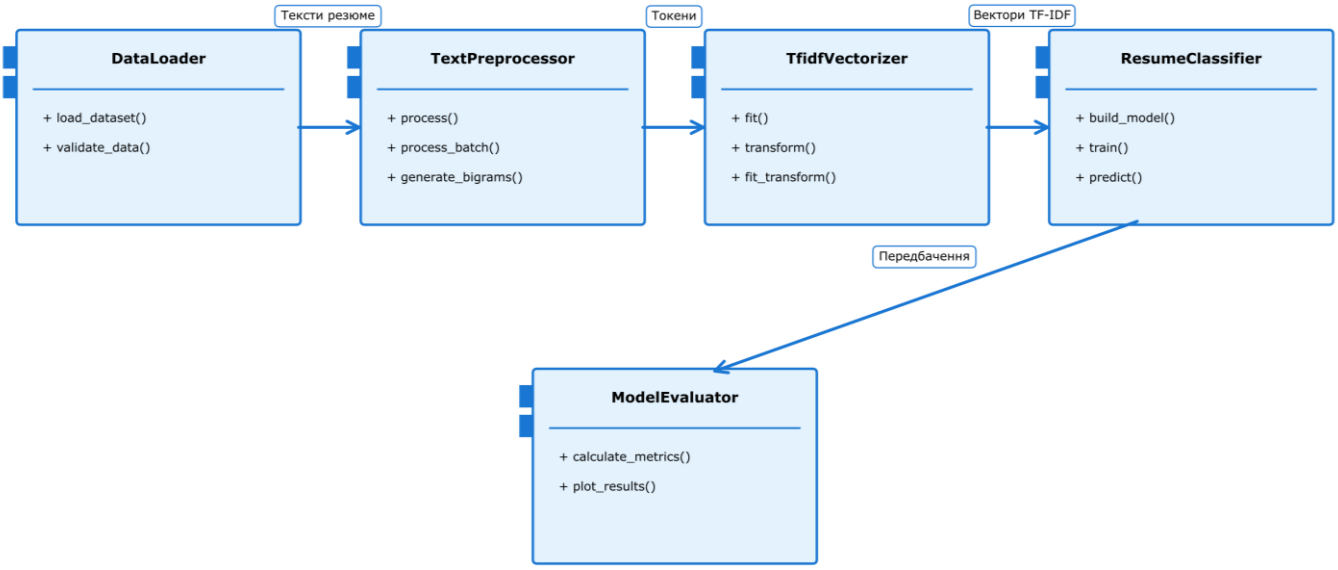


Рисунок 3.1 – Діаграма компонентів системи

На рисунку 3.1 представлена діаграма компонентів системи, яка ілюструє основні модулі та потоки даних між ними. Модуль завантаження даних відповідає за читання датасету Resume Dataset з CSV файлу та формування структур даних для подальшої обробки. Модуль препроцесингу виконує очищення та нормалізацію текстових даних, застосовуючи послідовність операцій обробки до кожного резюме. Модуль векторизації перетворює оброблений текст на числові вектори за допомогою методу TF-IDF.

Модуль класифікації реалізує нейронну мережу прямого поширення та містить логіку навчання і передбачення. Модуль оцінювання розраховує метрики якості роботи моделі та створює візуалізації результатів. Стрілки на діаграмі вказують напрямок передачі даних від одного модуля до іншого, утворюючи послідовний ланцюжок обробки від вхідних текстів резюме до фінальних передбачень категорій.

Модуль завантаження даних реалізований у вигляді класу DataLoader, який інкапсулює всю логіку роботи з датасетом. Клас містить метод load_dataset, який приймає шлях до CSV файлу та повертає дві структури – список текстів резюме та відповідний список міток категорій. Внутрішньо метод використовує бібліотеку

pandas для читання CSV файлу. Датасет Resume Dataset складається з чотирьох стовпців.

Перший стовпець ID містить унікальний ідентифікатор кожного резюме, який також використовується як назва відповідного PDF файлу у каталозі data. Другий стовпець Resume_str містить текст резюме у вигляді звичайного рядка без форматування. Третій стовпець Resume_html зберігає той самий текст, але з HTML розміткою, яка була присутня на вебсайті під час збору даних. Четвертий стовпець Category містить мітку професійної категорії для кожного резюме.

Для завдання класифікації використовується стовпець Resume_str, оскільки він містить очищений текст без зайвого форматування. Датасет містить резюме з 24 різних професійних категорій – HR, Designer, Information-Technology, Teacher, Advocate, Business-Development, Healthcare, Fitness, Agriculture, BPO, Sales, Consultant, Digital-Media, Automobile, Chef, Finance, Apparel, Engineering, Accountant, Construction, Public-Relations, Banking, Arts та Aviation. Загальна кількість резюме у датасеті становить 2485 записів.

Розподіл резюме по категоріях є відносно збалансованим. Найбільша категорія Information-Technology містить близько 120 резюме, що становить приблизно 5 % від загального обсягу. Найменші категорії, такі як Aviation та Arts, містять по 70-80 резюме кожна. Більшість категорій має від 90 до 110 резюме, що забезпечує достатню необхідну кількість даних для навчання моделі на кожному класі та уникає значної незбалансованості.

Клас DataLoader також містить метод validate_data, який перевіряє коректність завантажених даних. Метод виявляє та видаляє рядки з порожніми значеннями у стовпцях Resume_str або Category. Також перевіряється, що кожна мітка категорії належить до списку 24 дозволених значень. Якщо знайдено невідому категорію, видається попередження з інформацією про проблемний запис, включаючи його ідентифікатор. Метод повертає кількість видалених записів та оновлені дані без проблемних рядків.

Додатково клас містить метод get_category_distribution, який розраховує статистику розподілу резюме по категоріях. Метод повертає словник, де ключами є

назви категорій, а значеннями – кількість резюме у кожній категорії. Ця інформація корисна для аналізу збалансованості датасету та може бути використана для встановлення ваг класів під час навчання моделі, якщо незбалансованість виявиться значною.

Модуль препроцесингу організований навколо класу TextPreprocessor, який об'єднує всі операції обробки тексту. На рисунку 3.2 представлена діаграма класів цього модуля, яка показує його структуру, атрибути та методи. Клас ініціалізується з набором параметрів, що визначають поведінку обробки. Параметр remove_stopwords визначає, чи потрібно видаляти стоп-слова з тексту. Параметр use_lemmatization вказує, чи застосовувати лематизацію до токенів. Параметр generate_bigrams визначає, чи потрібно генерувати біграми додатково до уніграм. Параметр min_word_length встановлює мінімальну довжину слова, яке зберігається після обробки.

При ініціалізації класу завантажуються необхідні ресурси з бібліотеки NLTK. Список стоп-слів англійської мови завантажується з корпусу stopwords та зберігається у множині для швидкої перевірки належності. Створюється екземпляр лематизатора WordNetLemmatizer, який зберігається як атрибут класу для повторного використання. Також завантажується словник для визначення частин мови, необхідний для коректної роботи лематизатора.

Основний метод process приймає текст резюме як вхідний параметр та повертає список оброблених токенів. Обробка виконується послідовно через ланцюжок внутрішніх методів. Спочатку викликається метод convert_to_lowercase для приведення тексту до нижнього регістру. Потім метод remove_special_characters видаляє всі символи, окрім букв та пробілів. Метод tokenize розбиває очищений текст на окремі слова.

Якщо параметр remove_stopwords встановлено у значення true, викликається метод remove_stopwords_method, який фільтрує стоп-слова зі списку токенів. Якщо параметр use_lemmatization дорівнює true, кожен токен обробляється методом lemmatize, який приводить слово до базової форми з урахуванням частини мови.

Нарешті, якщо параметр generate_bigrams встановлено у true, метод generate_bigrams_method створює біграми та додає їх до списку токенів.

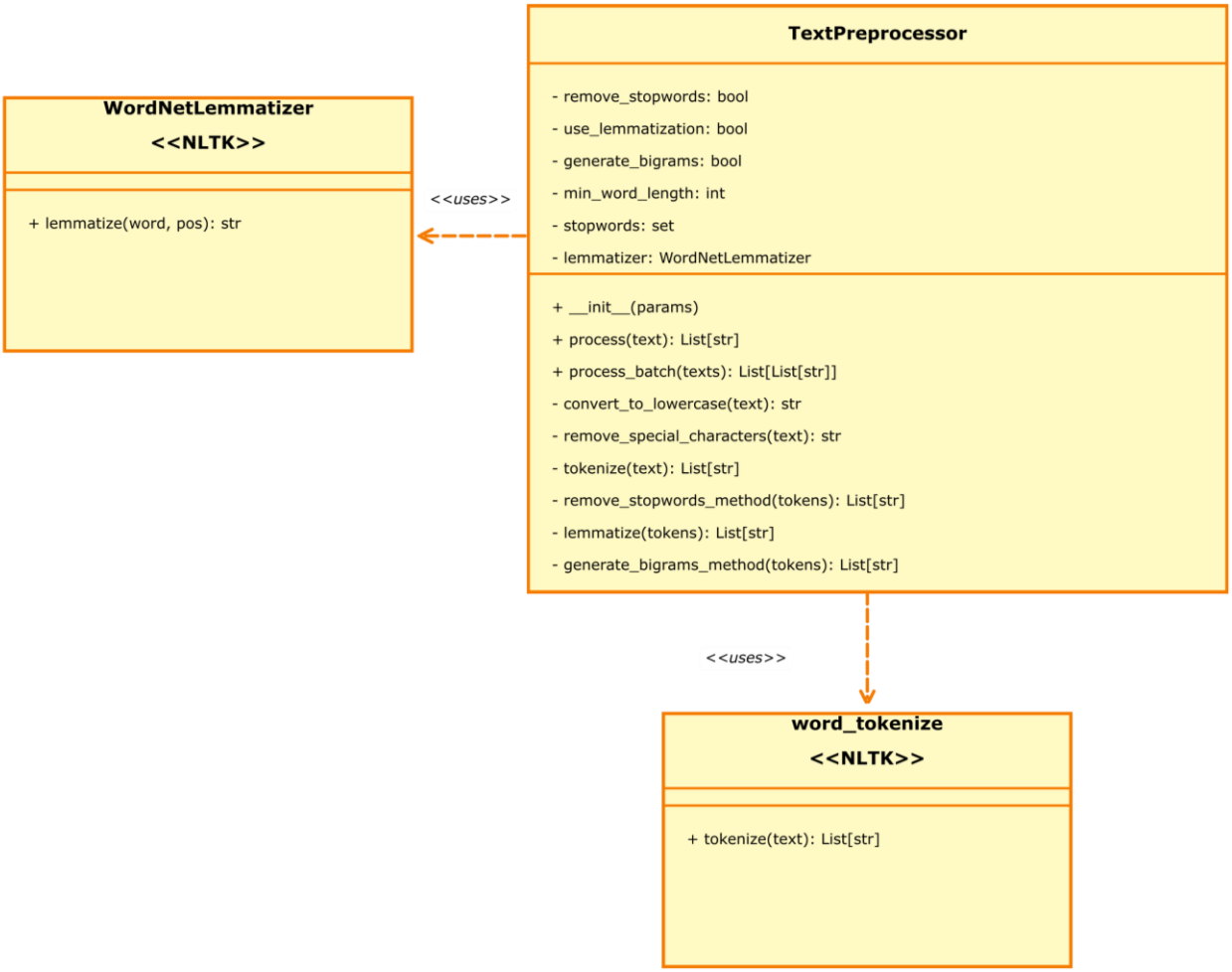


Рисунок 3.2 – Діаграма класів модуля припроцесингу тексту

Як показано на діаграмі класів, TextPreprocessor має залежності від двох компонентів бібліотеки NLTK. Перша залежність – це клас WordNetLemmatizer, який використовується для лематизації слів. Друга залежність – функція word_tokenize, яка застосовується для токенізації тексту. Ці залежності позначені на діаграмі пунктирними стрілками зі стереотипом uses, що вказує на використання зовнішніх компонентів.

Клас також надає метод process_batch для пакетної обробки декількох резюме одночасно. Цей метод приймає список текстів та застосовує метод process до кожного з них у циклі. Результатом є список списків токенів, де кожен внутрішній

список відповідає одному резюме. Пакетна обробка не дає значного приросту швидкості для операцій препроцесингу, але надає зручний інтерфейс для обробки всього датасету одним викликом.

Модуль векторизації реалізований через клас TfidfVectorizer, який є обгорткою навколо однойменного класу з бібліотеки scikit-learn з додатковою функціональністю. Клас містить метод fit, який будує словник термінів на основі навчальних даних. Під час виклику цього методу аналізуються всі токени у навчальних текстах, розраховуються їх частоти та відбираються найчастотніші терміни для включення до словника.

Розмір словника визначається параметром max_features, який встановлюється у значення 5000 для базової моделі. Це означає, що словник міститиме 5000 найчастотніших уніграм з навчальних даних. Для модифікованої моделі параметр налаштовується таким чином, щоб словник містив 4000 уніграм та 1000 біграм. Відбір термінів здійснюється на основі їх частоти появи у документах – терміни, які зустрічаються дуже рідко або дуже часто, можуть бути відфільтровані.

Метод transform приймає список оброблених текстів у вигляді списків токенів та перетворює кожен текст на вектор TF-IDF ознак. Для кожного токена у тексті перевіряється його наявність у побудованому словнику. Якщо токен знайдено, розраховується його TF-IDF значення як добуток частоти терміна у документі та оберненої частоти документів. Результатом є розріджена матриця, де рядки відповідають документам, стовпці – термінам зі словника, а значення – TF-IDF ваги.

Використання розрідженого формату матриці важливо для ефективності пам'яті. Оскільки кожне резюме містить лише невелику частину термінів зі словника розміром 5000 елементів, більшість значень у векторі дорівнюють нулю. Розріджений формат зберігає лише ненульові значення та їх позиції, що дозволяє економити пам'ять. Для датасету з 2485 резюме повна матриця розміром 2485 на 5000 займала б понад 90 мегабайт у звичайному форматі, але лише близько 10 мегабайт у розрідженому.

Клас також надає метод fit_transform, який об'єднує операції fit та transform в один виклик. Це зручно для обробки навчальних даних, коли потрібно одночасно побудувати словник та векторизувати тексти. Для валідаційних та тестових даних використовується лише метод transform з уже побудованим словником. Це критично важливо для коректної оцінки моделі – словник повинен будуватися виключно на навчальних даних, щоб уникнути витоку інформації з тестової вибірки.

Модуль класифікації є найбільш складним компонентом системи. На рисунку 3.3 представлена діаграма класів цього модуля, яка показує клас ResumeClassifier та його взаємодію з компонентами бібліотеки Keras. Клас інкапсулює всю логіку роботи з нейронною мережею, від побудови архітектури до навчання та передбачення.

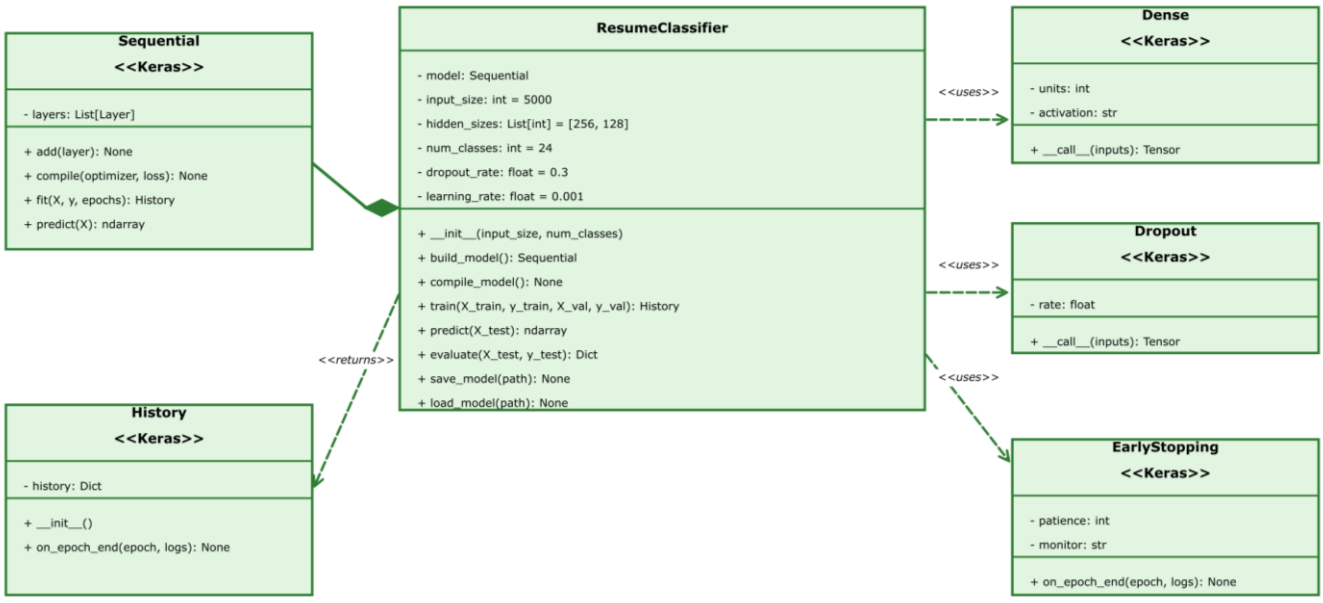


Рисунок 3.3 – Діаграма класів модуля класифікації

Клас має шість основних атрибутів. Атрибут model зберігає екземпляр нейронної мережі типу Sequential з Keras. Атрибут input_size визначає розмірність вхідного вектору, яка дорівнює розміру словника TF-IDF. Атрибут hidden_sizes містить список розмірів прихованих шарів мережі. Атрибут num_classes визначає кількість вихідних нейронів, яка дорівнює кількості професійних категорій. Атрибут

dropout_rate встановлює коефіцієнт dropout для регуляризації. Атрибут learning_rate визначає швидкість навчання для оптимізатора.

Метод build_model створює архітектуру нейронної мережі відповідно до специфікації з другого розділу. Метод створює послідовну модель Sequential та додає до неї шари у визначеному порядку. Спочатку додається вхідний шар, який явно не створюється, але визначається через параметр input_dim першого Dense шару. Перший прихований шар містить 256 нейронів з функцією активації ReLU. Після нього додається шар Dropout з коефіцієнтом 0.3.

Другий прихований шар має 128 нейронів також з функцією активації ReLU, за яким слідує ще один шар Dropout з тим самим коефіцієнтом 0.3. Вихідний шар містить 24 нейрони, по одному для кожної професійної категорії, з функцією активації softmax. Така функція активації забезпечує, що виходи мережі можна інтерпретувати як ймовірності належності резюме до кожного класу, причому сума всіх виходів дорівнює одиниці.

Метод compile_model налаштовує параметри навчання побудованої моделі. Як функцію втрат використовується categorical crossentropy, яка є стандартним вибором для задач багатокласової класифікації з взаємовиключними класами. Функція вимірює різницю між передбаченим розподілом ймовірностей та справжньою міткою класу. Як оптимізатор обрано Adam з початковою швидкістю навчання 0.001. Оптимізатор автоматично адаптує швидкість навчання для параметра нашої мережі під час тренування.

Метод також визначає список метрик, які розраховуються в процесі навчання. Головною метрикою є accuracy – точність класифікації, яка показує частку правильно класифікованих резюме. Ця метрика розраховується автоматично після кожного батчу та епохи як на навчальних, так і на валідаційних даних. Значення метрики зберігається в об'єкті History, який повертається методом train.

Метод train виконує процес навчання моделі на навчальних даних. Метод приймає п'ять параметрів. Перший параметр X_train – це матриця векторів TF-IDF для навчальних резюме. Другий параметр y_train містить мітки категорій для навчальних даних у форматі one-hot encoding. Третій та четвертий параметри X_val

та y_val – це валідаційні дані у тому самому форматі. П'ятий параметр epochs визначає максимальну кількість епох навчання.

Всередині методу train організовано цикл по епохах, який виконується автоматично методом fit моделі Keras. На кожній епосі навчальні дані розбиваються на батчі розміром 32 зразки. Кожен батч послідовно подається на вхід мережі. Виконується прямий прохід, під час якого розраховуються передбачення моделі для батчу. Потім обчислюється значення функції втрат шляхом порівняння передбачень з справжніми мітками.

На основі функції втрат розраховуються градієнти за допомогою алгоритму зворотного поширення помилки. Оптимізатор Adam використовує ці градієнти для оновлення ваг мережі у напрямку зменшення втрат. Після обробки всіх батчів однієї епохи виконується валідація на валідаційних даних. Модель використовується для передбачення на валідаційній вибірці, після чого розраховуються валідаційні втрати та точність.

Для запобігання перенавчанню використовується механізм раннього зупинення, реалізований через callback EarlyStopping з Keras. Цей об'єкт відстежує валідаційні втрати після кожної епохи. Якщо втрати не покращуються протягом певної кількості епох поспіль, визначеної параметром patience, навчання автоматично зупиняється. При цьому зберігається версія моделі з найкращим значенням валідаційної втрати, а не остання версія після завершення навчання.

Метод predict приймає матрицю векторів тестових даних та повертає передбачені категорії для кожного резюме. Внутрішньо метод викликає метод predict нейронної мережі, який повертає матрицю ймовірностей розміром кількість резюме на 24 класи. Для кожного резюме обирається клас з максимальною ймовірністю як фінальне передбачення. Метод також може повертати самі ймовірності замість міток класів, встановивши відповідний параметр.

Клас ResumeClassifier містить методи для збереження та завантаження навченої моделі. Метод save_model приймає шлях до файлу та зберігає повну конфігурацію моделі, включаючи архітектуру шарів, навчені ваги та стан оптимізатора, у файл формату HDF5. Метод load_model завантажує модель з файлу

та відновлює її повний стан, дозволяючи продовжити навчання або використовувати для передбачень без необхідності перебудови архітектури.

На діаграмі класів показано, що ResumeClassifier має композиційний зв'язок з класом Sequential, позначений ромбом. Це означає, що така модель Sequential створюється всередині класу ResumeClassifier та не існує незалежно від нього. Клас також має залежності від класів Dense, Dropout та EarlyStopping, які використовуються при побудові та навчанні моделі. Метод train повертає об'єкт History, який зберігає історію навчання.

Модуль оцінювання реалізований через клас ModelEvaluator, який містить методи для розрахунку різних метрик та створення візуалізацій. Метод calculate_metrics приймає два параметри – масив справжніх міток та масив передбачених міток. Метод розраховує набір метрик для всіх класів окремо та повертає їх у вигляді словника.

Для кожної з 24 категорій розраховується точність передбачення, яка показує частину вірних передбачень серед усіх резюме, класифікованих у цю категорію. Також розраховується повнота, яка вимірює частину вірно знайдених резюме серед усіх резюме цієї категорії у тестових даних. F1-міра обчислюється як гармонічне середнє між точністю передбачення та параметром повноти, забезпечуючи балансну оцінку якості для класу.

Крім метрик для окремих класів, метод розраховує агреговані метрики по всіх класах. Макро-усереднена точність обчислюється як звичайне середнє арифметичне точностей усіх класів. Зважено усереднена точність враховує зразки у кожному класі, даючи більшу вагу класам з більшою кількістю резюме. Аналогічно розраховуються макро та зважені версії повноти та F1-міри.

Метод calculate_confusion_matrix будує матрицю плутанини розміром 24 на 24. Елемент матриці на перетині рядка i та стовпця j показує, скільки резюме справжньої категорії i були класифіковані моделлю як категорія j. Діагональні елементи матриці відповідають правильним класифікаціям, коли передбачена категорія збігається зі справжньою. Недіагональні елементи показують різні типи помилок класифікації.

Аналіз матриці плутанини дозволяє виявити, які пари категорій найчастіше плутаються між собою. Наприклад, якщо багато резюме категорії Sales класифікуються як Business-Development і навпаки, це може вказувати на схожість термінології цих професійних сфер. Така інформація корисна для розуміння обмежень моделі та може бути використана для її подальшого покращення.

Клас також містить методи для показу результатів. Метод plot_training_history приймає об'єкт History, повернений методом train, та будує два графіки. Перший графік показує зміну функції втрат протягом епох навчання окремо для навчальної та валідаційної вибірок. Другий графік відображає динаміку точності класифікації на тих самих вибірках. Обидва графіки дозволяють відстежити процес навчання та виявити можливе перенавчання.

Метод plot_confusion_matrix будує теплову карту матриці плутанини з використанням бібліотеки Seaborn. Інтенсивність кольору у кожній комірці відповідає кількості резюме з певною комбінацією справжньої та передбаченої категорій. Осі графіка підписуються назвами категорій, що дозволяє легко ідентифікувати проблемні пари класів. Діагональ матриці, яка відповідає правильним класифікаціям, зазвичай виділяється більш інтенсивним кольором.

Метод plot_metrics_comparison створює стовпчасті діаграми для порівняння метрик між базовою та модифікованою моделями. Для кожної метрики – точності передбачення, повноти та F1-міри – будується окрема діаграма з двома стовпцями, що відповідають двом моделям. Це дозволяє наочно побачити покращення або погіршення якості при додаванні біграм до векторного представлення.

Взаємодія між модулями детально показана на діаграмі послідовності на рисунку 3.4. Діаграма ілюструє сценарій використання системи від завантаження даних до отримання передбачень. Головний скрипт програми виступає як координатор, який послідовно викликає методи різних модулів.

Спочатку головний скрипт створює екземпляр DataLoader та викликає його метод load_dataset, передаючи шлях до CSV файлу. DataLoader завантажує дані та повертає два списки – тексти резюме та їх мітки категорій. Потім створюється екземпляр TextPreprocessor з необхідними параметрами обробки. Головний скрипт

викликає метод process_batch, передаючи йому список текстів резюме. Модуль препроцесингу обробляє кожен текст та повертає список списків токенів.

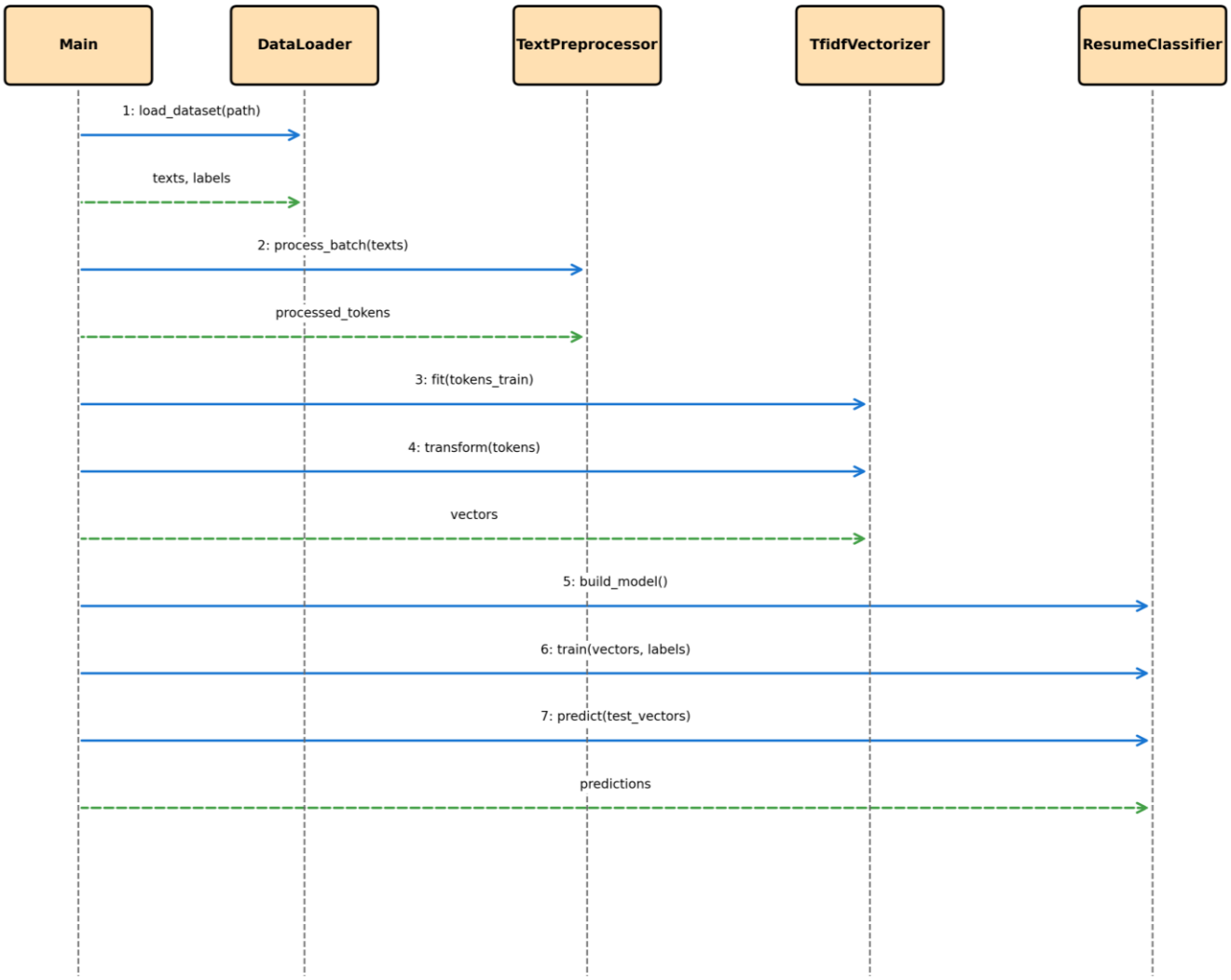


Рисунок 3.4 – Діаграма послідовності процесу класифікації резюме

Після препроцесингу дані розділяються на навчальну, валідаційну та тестову вибірки за допомогою функції train_test_split з scikit-learn. Цей крок не показаний на діаграмі послідовності, оскільки він виконується безпосередньо у головному скрипті без виклику окремого модуля. Створюється екземпляр TfidfVectorizer, і головний скрипт викликає його метод fit, передаючи токени навчальної вибірки для побудови словника.

Після навчання векторизатора головний скрипт викликає його метод transform для кожної з трьох вибірок – навчальної, валідаційної та тестової. Векторизатор перетворює списки токенів на матриці векторів TF-IDF та повертає їх головному скрипту. Тепер дані готові для навчання нейронної мережі.

Створюється екземпляр ResumeClassifier з параметрами, що визначають архітектуру мережі. Головний скрипт викликає метод build_model для створення архітектури, а потім метод compile_model для налаштування параметрів навчання. Метод train викликається з навчальними та валідаційними даними, запускаючи процес навчання. Після завершення навчання головний скрипт викликає метод predict з тестовими векторами.

Модуль класифікації виконує передбачення на тестових даних та повертає масив передбачених категорій. Хоча це не показано на діаграмі, головний скрипт потім створює екземпляр ModelEvaluator та викликає його методи для розрахунку метрик та створення візуалізацій, передаючи справжні та передбачені мітки.

Модульна архітектура має декілька переваг для процесу розробки та підтримки системи. Кожен модуль можна розробляти та тестувати незалежно від інших. Це дозволяє виявляти та зкоректувати помилки на ранніх етапах, коли вони стосуються лише невеликої частини системи. Можна написати юніт-тести для кожного модуля окремо, що полегшує перевірку коректності роботи.

Модулі з чітко визначеними інтерфейсами легко замінювати альтернативними реалізаціями без впливу на іншу частину коду. Наприклад, якщо з'явиться кращий метод векторизації, можна створити новий клас, який реалізує той самий інтерфейс fit та transform, і замінити TfidfVectorizer на новий клас без змін в інших модулях. Така гнучкість полегшує експериментування з різними підходами.

Модульна структура полегшує додаваня функціоналу системи. Для створення нових можливостей достатньо створити новий модуль або розширити існуючий, не торкаючись інших частин програми. Наприклад, можна додати новий модуль для збору даних з інших джерел або модуль для розгортання навченої моделі як вебсервісу. Ці додатки не вимагають модифікації існуючих модулів.

Окремі модулі можна також використовувати в певних інших проектах. Модуль препроцесингу тексту придатний для будь-яких задач обробки англомовних документів, а не лише для класифікації резюме. Модуль векторизації можна застосувати для перетворення текстів у числовий формат для інших алгоритмів машинного навчання. Така можливість повторного використання економить час при розробці нових систем.

3.3 Особливості реалізації ключових алгоритмічних компонентів

Реалізація модуля препроцесингу тексту потребує особливої уваги до деталей обробки англомовних текстів. Метод convert_to_lowercase є найпростішим кроком обробки. Він застосовує стандартний метод lower класу str до всього тексту. Це перетворення гарантує, що слова Python та python будуть розглядатися як один термін, незалежно від їх позиції на початку речення чи всередині.

Метод remove_special_characters використовує регулярний вираз для видалення всіх символів, окрім букв англійського алфавіту та пробілів. Патерн регулярного виразу має вигляд квадратних дужок з символом вставки на початку та діапазоном від a до z. Це означає будь-який символ, що не є малою літерою англійського алфавіту. Функція sub з модуля замінює всі знайдені символи на пробіли.

Така обробка видаляє цифри, знаки пунктуації, спеціальні символи та символи з інших алфавітів, які можуть зустрічатися у резюме. Після застосування регулярного виразу виконується додаткове очищення множинних пробілів. Функція split без параметрів розбиває текст на слова, автоматично видаляючи всі послідовності пробільних символів. Потім метод join об'єднує слова назад у рядок з одинарними пробілами між ними.

Метод tokenize використовує функцію word_tokenize з бібліотеки NLTK. Ця функція застосовує складніший алгоритм розбиття тексту на токени порівняно з простим розділенням по пробілах. Токенізатор коректно обробляє скорочення типу don't, розділяючи їх на do та n't. Також правильно обробляються апострофи у

присвійних формах, наприклад John's розбивається на John та 's. Функція розпізнає абревіатури з крапками, такі як Ph.D., та не розділяє їх на окремі букви.

Після токенізації застосовується фільтрація за довжиною слова. Цикл for проходить по всіх токенах та зберігає лише ті, довжина яких не менша за значення параметра min_word_length. Типове значення цього параметра дорівнює 2, що дозволяє видалити однолітерні токени, які зазвичай не несуть змістового навантаження.

Метод remove_stopwords_method перевіряє кожен токен на належність до множини стоп-слів. Множина завантажується з корпусу stopwords бібліотеки NLTK під час ініціалізації класу. Список містить 179 найчастотніших службових слів англійської мови. До них належать артиклі a, an, the, прийменники in, on, at, from, сполучники and, but, or, займенники I, you, he, she та допоміжні дієслова am, is, are, was, were.

Використання множини замість списку для зберігання стоп-слів є важливою оптимізацією. Перевірка належності елемента до множини має складність O(1) завдяки використанню хеш-таблиці, тоді як для списку складність становить O(n). Оскільки перевірка виконується для кожного токена у кожному резюме, різниця у швидкості стає помітною на великих датасетах.

Метод lemmatize приводить кожне слово до його базової словникової форми. Лематизатор WordNetLemmatizer використовує великий лінгвістичний словник WordNet, який містує інформацію про морфологічні зв'язки між словами. Для дієслів лематизація перетворює всі граматичні форми до інфінітиву. Наприклад, слова running, ran, runs всі приводяться до форми run.

Для іменників лематизація перетворює форми множини до однини. Слово computers стає computer, a studies перетворюється на study з урахуванням зміни у на і перед закінченням es. Для прикметників лематизатор приводить ступені порівняння до базової форми, перетворюючи better на good та best також на good. Однак для коректної роботи лематизатора необхідно вказувати частину мови кожного слова.

Без інформації про частину мови лематизатор за замовчуванням вважає всі слова іменниками, що призводить до неправильної обробки дієслів та прикметників.

Для визначення частини мови використовується функція pos_tag з NLTK, яка приймає список токенів та повертає список пар, де кожна пара містить токен та його частину мови у форматі Penn Treebank. Теги потім конвертуються у формат WordNet за допомогою допоміжної функції.

Функція get_wordnet_pos приймає тег Penn Treebank та повертає відповідний тег WordNet. Теги, що починаються з J, відповідають прикметникам та конвертуються у wordnet.ADJ. Теги з V на початку позначають дієслова та перетворюються на wordnet.VERB. Теги з N відповідають іменникам і стають wordnet.NOUN. Теги з R позначають прислівники та конвертуються у wordnet.ADV. Для всіх інших тегів повертається значення за замовчуванням wordnet.NOUN.

Метод generate_bigrams_method створює біграми з послідовних пар токенів. Цикл проходить по списку токенів від першого до передостаннього елемента. На кожній ітерації створюється біграм шляхом об'єднання поточного токена та наступного через символ підкреслення. Наприклад, якщо послідовні токени є machine та learning, результуючий біграм матиме вигляд machine_learning.

Символ підкреслення використовується як роздільник, щоб відрізнити біграми від уніграм у словнику. Без роздільника біграм machinelearning міг би існувати як окремий уніграм, що призвело б до конфліктів. З роздільником machine_learning однозначно розпізнається як біграм. Згенеровані біграми додаються до існуючого списку токенів, таким чином фінальний список містить як вихідні уніграми, так і нові біграми.

Важливо зазначити, що генеруються всі можливі біграми з послідовних токенів, без фільтрації за статистичною значущістю. Це означає, що кількість біграм приблизно дорівнює кількості уніграм мінус один. Для резюме з 200 уніграм буде створено близько 199 біграм, що подвоює загальну кількість токенів. Фільтрація статистично незначущих біграм відбувається пізніше на етапі векторизації, коли відбираються лише найчастотніші терміни для включення до словника.

Реалізація модуля векторизації спирається на клас TfidfVectorizer з scikit-learn, але додає додаткову функціональність для роботи з попередньо обробленими токенами. Стандартний векторизатор очікує на вхід рядки тексту та виконує

токенізацію самостійно. Однак у нашому випадку токенізація вже виконана модулем препроцесингу разом з лематизацією та іншими операціями.

Щоб використовувати попередньо оброблені токени, векторизатор налаштовується з параметром tokenizer, який вказує на просту функцію-ідентичність. Ця функція просто повертає вхідний список токенів без змін. Також встановлюється параметр lowercase у значення False, оскільки приведення до нижнього регістру вже виконано на етапі препроцесингу. Параметр token_pattern встановлюється у None, щоб вимкнути розбиття за регулярним виразом.

Параметр max_features визначає розмір словника та встановлюється у значення 5000 для базової моделі. Векторизатор аналізує всі токени у навчальних текстах, розраховує частоту появи кожного токена у документах та відбирає 5000 найчастотніших. Частота документа для токена визначається як кількість резюме, у яких цей токен з'являється хоча б один раз, незалежно від кількості поява всередині документа.

Параметри min_df та max_df дозволяють додатково фільтрувати терміни на основі їх частоти. Параметр min_df встановлює мінімальну частоту документа для включення терміна до словника. Значення 2 означає, що термін повинен з'явитися принаймні у двох різних резюме. Це дозволяє відфільтрувати рідкісні терміни, які можуть бути помилками друку або специфічними назвами, що не допомагають узагальненню моделі.

Параметр max_df встановлює максимальну частоту документа як частку від загальної кількості документів. Значення 0.8 означає, що терміни, які зустрічаються у більш ніж 80 %х резюме, виключаються зі словника. Такі терміни є занадто загальними та не допомагають розрізняти категорії. Наприклад, слово experience може зустрічатися майже в усіх резюме незалежно від професії, тому його виключення з ознак може навіть покращити якість класифікації.

Метод fit будує словник на основі навчальних даних. Внутрішньо векторизатор створює словник як відображення від термінів до їх індексів у векторі ознак. Також розраховується обернена частота документа для кожного терміна за формулою логарифм від частки загальної кількості документів до кількості

документів, що містять термін. Значення IDF зберігаються для використання при векторизації.

Терміни сортуються за частотою у спадному порядку, і вибираються перші max_features термінів. Якщо встановлені параметри min_df або max_df, терміни додатково фільтруються перед відбором топ-N. Фінальний словник містить відображення від кожного відібраного терміна до його позиції у векторі від 0 до 4999 для словника розміром 5000.

Метод transform перетворює список токенів на вектор TF-IDF значень. Для кожного токена перевіряється його наявність у словнику через операцію пошуку у словнику. Якщо токен знайдено, визначається його індекс у векторі. Розраховується частота терміна у документі як кількість появ токена, поділена на загальну кількість токенів у документі. Значення TF множиться на попередньо розраховане значення IDF для цього терміна.

Результуючий вектор зберігається у розрідженому форматі CSR, який є ефективним для зберігання та математичних операцій з векторами, що містять багато нулів. Формат CSR зберігає три масиви. Перший масив data містить всі ненульові значення вектора. Другий масив indices містить індекси стовпців для кожного ненульового значення. Третій масив indptr містує покажчики на початок кожного рядка у двох попередніх масивах.

Для модифікованої моделі, яка використовує комбінацію уніграм та біграм, параметр max_features встановлюється у значення 5000, але окремо контролюється розподіл між типами токенів. Параметр ngram_range встановлюється у значення від 1 до 2, що дозволяє векторизатору розглядати як уніграми, так і біграми. Однак стандартний механізм відбору топ-N термінів не гарантує певного співвідношення між уніграмами та біграмами.

Щоб забезпечити включення 4000 уніграм та 1000 біграм, використовується модифікований підхід. Спочатку створюється окремий векторизатор лише для уніграм з max_features рівним 4000. Він будує словник з 4000 найчастотніших уніграм. Потім створюється другий векторизатор для біграм з max_features рівним

1000, який будує словник з 1000 найчастотніших біграм. Нарешті, два словники об'єднуються у єдиний, що містить 5000 термінів.

Реалізація модуля класифікації потребує ретельної організації процесу навчання нейронної мережі. Метод build_model створює послідовність шарів, використовуючи функціональний програмний інтерфейс Keras. Кожен Dense шар ініціалізується з випадковими вагами згідно зі стратегією Glorot uniform, яка також називається Xavier ініціалізацією. Ця стратегія встановлює початкові ваги з рівномірного розподілу у діапазоні, який залежить від кількості вхідних та вихідних нейронів шару.

Правильна ініціалізація ваг критично важлива для успішного навчання глибоких мереж. Якщо ваги ініціалізувати занадто малими значеннями, градієнти будуть зменшуватися при зворотному поширенні через шари, і навчання глибоких шарів майже не відбуватиметься. Якщо ваги занадто великі, активації можуть насичуватися на краях функції активації, де градієнти близькі до нуля. Ініціалізація Glorot підтримує дисперсію активацій та градієнтів приблизно однаковою на всіх шарах.

Функція активації ReLU застосовується після кожного прихованого Dense шару. Ця функція обчислюється як максимум між входом та нулем, фактично обнулюючи всі від'ємні значення та залишаючи додатні без змін. ReLU має перевагу над класичною сигмоїдою та гіперболічним тангенсом завдяки відсутності проблеми зникаючих градієнтів. Для додатних входів градієнт функції дорівнює одиниці, що дозволяє ефективно поширювати помилку назад через багато шарів.

Однак ReLU має недолік у вигляді проблеми мертвих нейронів. Якщо нейрон потрапляє у від'ємну область під час навчання, його градієнт стає нульовим, і ваги цього нейрона перестають оновлюватися. Нейрон назавжди залишається неактивним для всіх можливих входів. Ця проблема частково вирішується правильною ініціалізацією ваг та використанням невеликої швидкості навчання, що зменшує ймовірність різких змін, які можуть перевести багато нейронів у від'ємну область.

Шари Dropout додаються після кожного прихованого шару для регуляризації мережі. Під час навчання шар Dropout випадково обнулює певну частку входів, визначену параметром dropout_rate, який встановлений у значення 0.3. Це означає, що кожен нейрон попереднього шару має ймовірність 30 % бути вимкненим на поточному кроці навчання. Вимкнені нейрони не беруть участь у прямому проході та не отримують градієнтів при зворотному поширенні.

Механізм Dropout запобігає перенавчанню, змушуючи мережу не покладатися на будь-який окремий нейрон або невелику групу нейронів. Оскільки на кожному кроці активна різна випадкова підмножина нейронів, мережа вчиться виявляти надійні ознаки, які працюють навіть за відсутності деяких нейронів. Це схоже на навчання ансамблю моделей, де кожна конфігурація активних нейронів представляє окрему модель, а фінальний результат є середнім по всіх цих моделях.

Під час тестування шар Dropout вимикається, і всі нейрони використовуються одночасно. Щоб компенсувати те, що під час навчання використовувалося лише 70 % нейронів, а під час тестування використовуються всі 100 %, виходи автоматично масштабуються. Keras виконує це масштабування автоматично, множачи виходи на коефіцієнт 1 мінус dropout_rate під час навчання замість ділення на цей коефіцієнт під час тестування.

Вихідний Dense шар має 24 нейрони з функцією активації softmax. Ця функція перетворює вектор з 24 довільних дійсних чисел на вектор ймовірностей, де кожен елемент знаходиться у діапазоні від 0 до 1, і сума всіх елементів дорівнює 1. Математично softmax обчислюється як експонента від кожного елемента, поділена на суму експонент всіх елементів.

Функція softmax посилює відмінності між виходами нейронів. Якщо один нейрон має суттєво більше значення за інші, після застосування softmax його ймовірність буде близькою до 1, а ймовірності інших нейронів будуть близькими до 0. Якщо кілька нейронів мають схожі значення, їх ймовірності будуть розподілені більш рівномірно. Така поведінка робить softmax природним вибором для задач класифікації з взаємовиключними класами.

Метод compile_model налаштовує оптимізатор Adam з початковою швидкістю навчання 0.001. Adam є адаптивним методом оптимізації, який підтримує окрему швидкість щоб навчитись для кожного параметра мережі. Алгоритм відстежує перший момент градієнтів, який є експоненціально згладженим середнім градієнтів, та другий момент, який є експоненціально згладженим середнім квадратів градієнтів.

Швидкість навчання для визначеного параметра визначається як базова швидкість, поділена на квадратний корінь з другого моменту. Це означає, що параметри з великими та нестабільними градієнтами отримують меншу швидкість навчання, тоді як параметри з малими стабільними градієнтами отримують більшу швидкість. Така адаптація дозволяє ефективно навчатися навіть за наявності розріджених градієнтів та зашумлених даних.

Adam також використовує корекцію зміщення для перших кількох кроків навчання. Оскільки експоненціальне згладжування ініціалізується нульовими значеннями, перші оцінки моментів є зміщеними у бік нуля. Корекція ділить моменти на коефіцієнт, який залежить від номера кроку та швидко наближається до одиниці. Це забезпечує стабільне навчання з самого початку без необхідності у фазі прогріву.

Функція втрат categorical crossentropy вимірює різницю між двома розподілами ймовірностей – передбаченим розподілом з виходу softmax та справжнім розподілом з one-hot кодованої мітки. Для кожного резюме справжня мітка представлена як вектор з 24 елементів, де один елемент дорівнює 1 для правильного класу, а всі інші дорівнюють 0. Передбачення також є вектором з 24 елементів, але з дійсними значеннями від 0 до 1, що сумуються до 1.

Categorical crossentropy обчислюється як сума по всіх класах від добутку справжньої ймовірності класу на логарифм передбаченої ймовірності цього класу, взята з від'ємним знаком. Оскільки справжня ймовірність дорівнює 1 лише для одного класу та 0 для всіх інших, формула спрощується до від'ємного логарифму передбаченої ймовірності правильного класу. Мінімізація цієї функції еквівалентна максимізації ймовірності правильного класу.

Функція втрат має корисну властивість, що її градієнт за виходами мережі має простий вигляд. Градієнт categorical crossentropy відносно виходів softmax дорівнює різниці між передбаченими ймовірностями та справжніми мітками. Це означає, що для правильного класу градієнт дорівнює передбаченій ймовірності мінус 1, а для неправильних класів градієнт дорівнює передбаченій ймовірності. Такий градієнт природно спрямовує оновлення у бік збільшення ймовірності правильного класу.

Метод train організовує процес навчання у вигляді циклу по епохах та батчах. Кожна епоха складається з одного повного проходу через всі навчальні дані. Дані розбиваються на батчі розміром 32 зразки для ефективних обчислень на графічному процесорі. Сучасні GPU оптимізовані для паралельної обробки багатьох зразків одночасно, тому обробка батчу з 32 резюме виконується майже так само швидко, як обробка одного резюме.

На кожному батчі виконується прямий прохід, під час якого вхідні вектори послідовно проходять через всі шари мережі. Для кожного шару визначаємо активацію як функція від виходів попереднього шару та ваг поточного шару. Виходи зберігаються для використання при зворотному поширенні. Після отримання фінальних передбачень розраховується значення функції втрат для батчу як середнє від втрат окремих зразків.

Зворотне поширення починається з обрахування градієнту функції втрат за виходами останнього шару. Цей градієнт множиться на градієнт функції активації softmax, результатом чого є градієнт за входами останнього шару. Потім обчислюється градієнт втрат за вагами останнього шару як добуток градієнта за входами на активації передостаннього шару. Градієнт за виходами передостаннього шару обчислюється як добуток ваг останнього шару на градієнт за входами останнього шару.

Процес продовжується покроково для кожного шару у зворотному порядку від виходу до входу. На кожному етапі обчислюються градієнти за вагами поточного шару та градієнти за виходами попереднього шару. Для шарів Dropout градієнт множиться на маску, яка була згенерована під час прямого проходу, щоб

зберегти узгодженість між прямим та зворотним проходами. Для шарів Dense з активацією ReLU градієнт множиться на похідну ReLU, яка дорівнює 1 для додатних входів та 0 для від'ємних.

Після обчислення цього для всіх ваг мережі оптимізатор Adam використовує їх для оновлення параметрів. Для кожної ваги обчислюється адаптована швидкість навчання на основі історії визначених градієнтів. Вага оновлюється шляхом віднімання добутку адаптованої швидкості на поточний градієнт. Процес повторюється для кожного батчу у епосі, поступово покращуючи ваги мережі для зменшення функції втрат.

Після обробки всіх навчальних батчів однієї епохи виконується валідація. Вся валідаційна вибірка подається на вхід мережі для отримання передбачень. Важливо, що під час валідації шари Dropout вимикаються, і мережа використовує всі нейрони. Розраховуються валідаційні втрати та точність класифікації. Ці метрики не впливають на оновлення ваг, але використовуються для моніторингу процесу навчання та виявлення перенавчання.

Callback EarlyStopping відстежує валідаційні втрати після кожної епохи. Якщо втрати не покращуються протягом певної кількості епох, визначеної параметром patiece, цей параметер зупиняє навчання. Значення patience встановлюється у 10, що означає зупинку після 10 епох без покращення. Callback також зберігає ваги мережі з епохи з найкращими валідаційними втратами. Після зупинки навчання ваги автоматично відновлюються до цього найкращого стану.

Механізм раннього зупинення запобігає витраті часу на непродуктивні епохи, коли модель вже не покращується. Також він захищає від перенавчання, зупиняючи процес до того, як модель почне занадто сильно підлаштовуватися під навчальні дані за рахунок погіршення на валідаційних даних. Типово навчання зупиняється після 50-70 епох з максимальним ліміт 100 епох, хоча точна кількість варіюється між запусками через випадковість у ініціалізації ваг та порядку батчів.

Метод predict використовує навчену мережу для класифікації нових резюме. Вхідна матриця векторів подається на вхід моделі. Виконується лише прямий прохід через шари без обчислення градієнтів, оскільки ваги не потребують оновлення. Для

кожного резюме мережа повертає вектор з 24 значень після softmax, які можна інтерпретувати як ймовірності належності до кожної категорії.

Щоб отримати фінальне передбачення, для кожного резюме обирається індекс класу з максимальною ймовірністю. Функція argmax з бібліотеки NumPy знаходить позицію найбільшого елемента у векторі. Цей індекс потім конвертується назад у текстову мітку категорії за допомогою зворотного відображення, яке створюється під час підготовки даних. Результатом є масив передбачених категорій такого самого розміру, як кількість вхідних резюме.

Метод також може повертати матрицю ймовірностей замість міток класів, якщо встановлено відповідний параметр. Це корисно для подальшого аналізу, наприклад для вимірювання впевненості моделі у передбаченнях. Якщо максимальна ймовірність близька до 1, модель дуже впевнена у своєму рішенні. Якщо кілька класів мають схожі ймовірності, модель невпевнена, що може вказувати на складний або неоднозначний випадок.

Реалізація модуля оцінювання спирається на функції з модуля metrics бібліотеки scikit-learn. Метод calculate_metrics використовує функцію classification_report, яка розраховує точність передбачення, повноту та F1-міру для кожного класу окремо. Функція порівнює справжні мітки з передбаченими та підраховує кількість істинно позитивних, та негативних випадків для кожної категорії.

Точність передбачення для класу обчислюється як кількість резюме, правильно класифікованих у цей клас, поділена на загальну кількість резюме, класифікованих у цей клас. Повнота обчислюється як кількість правильно класифікованих резюме, поділена на загальну кількість резюме цього класу у тестових даних. F1-міра є середнім гармонічним між точністю та повнотою, обчисленим як 2 помножити на добуток точності та повноти, поділений на їх суму.

Макро-усереднені метрики обчислюються як звичайне середнє арифметичне метрик усіх класів. Кожен клас має однакову вагу незалежно від кількості зразків у ньому. Така усереднення корисна для оцінки роботи моделі на рідкісних класах, які можуть бути важливими, навіть якщо містять мало зразків. Зважено усереднені

метрики враховують розмір кожного класу, множачи метрику класу на кількість зразків у ньому перед усередненням.

Метод calculate_confusion_matrix використовує функцію confusion_matrix з scikit-learn. Функція створює матрицю розміром 24 на 24, де елемент у позиції i, j показує кількість резюме справжнього класу i, які були передбачені як клас j. Функція проходить по всіх парах справжніх та передбачених міток та інкрементує відповідний лічильник у матриці. Діагональні елементи структури дають правильні класифікації, недіагональні – різні типи помилок.

Візуалізація матриці плутанини створюється за допомогою функції з Seaborn. Функція приймає матрицю чисел та відображає її як теплову карту, де інтенсивність кольору відповідає значенню у комірці. Параметр annot встановлюється у значення True для відображення числових значень всередині комірок. Параметр fmt встановлюється у 'd' для форматування чисел як цілих значень без десяткових знаків. Параметр cmap визначає кольорову схему, типово використовується 'Blues' для синіх відтінків від світлого до темного.

Осі графіка підписуються назвами категорій. Вісь X позначається як передбачені категорії та містить мітки для всіх 24 класів. Вісь Y позначається як справжні категорії з тими самими мітками. Назви категорій повертаються на 45 градусів для кращої читабельності, оскільки деякі назви є досить довгими, наприклад Business-Development або Information-Technology. Графік зберігається у файл PNG з високою роздільною здатністю 300 точок на дюйм для забезпечення чіткості при друкуванні.

Метод plot_training_history створює два графіки у одному вікні за допомогою функції з Matplotlib. Функція створює сітку з одного рядка та двох стовпців для розміщення двох графіків поруч. Лівий графік показує зміну функції втрат протягом епох. Вісь X представляє номер епохи від 1 до фактичної кількості виконаних епох. Вісь Y показує значення функції втрат.

На графіку відображаються дві лінії. Перша лінія показує втрати на навчальних даних, отримані з атрибута history точка loss об'єкта History. Друга лінія показує втрати на валідаційних даних з атрибута val_loss. Лінії мають різні кольори

та позначені у легенді для розрізнення. Правий графік аналогічно показує точність класифікації на навчальних та валідаційних даних протягом епох.

Аналіз графіків дозволяє виявити проблеми у процесі навчання. Якщо навчальні втрати продовжують зменшуватися, а валідаційні втрати починають зростати, це вказує на перенавчання. Модель занадто сильно підстроюється під дані навчання та втрачає здатність узагальнювати на інші дані. Якщо обидві криві втрат виходять на плато та перестають покращуватися, це може означати, що модель досягла своєї максимальної здатності для даної архітектури та даних.

Метод plot_metrics_comparison створює стовпчасту діаграму для порівняння метрик базової та модифікованої моделей. Для кожної метрики – точності класифікації, макро F1-міри та зваженої F1-міри – створюється група з двох стовпців. Перший стовпець показує значення метрики для базової моделі, другий – для модифікованої. Стовпці різних моделей мають різні кольори для легкого розрізнення.

Висота кожного стовпця відповідає значенню метрики, яке зазвичай знаходиться у діапазоні від 0 до 1 або від 0 до 100 %. Над кожним стовпцем відображається точне числове значення метрики з двома десятковими знаками. Це дозволяє не лише візуально порівняти моделі, але й побачити точну величину різниці. Вісь Y підписується як значення метрики у %x або як безрозмірне число залежно від представлення.

Організація коду у модулі також включає допоміжні функції для перетворення даних. Функція encode_labels перетворює текстові мітки категорій на числові індекси від 0 до 23. Створюється відображення від кожної унікальної назви категорії до її індексу. Потім кожна мітка у списку замінюється на відповідний індекс. Результатом є масив цілих чисел, який можна застовувати для індексації або подальшого перетворення у one-hot формат.

Функція to_categorical перетворює масив індексів класів у матрицю one-hot кодування. Для кожного індексу створюється вектор довжиною 24, де всі елементи дорівнюють 0, окрім елемента з позицією, що відповідає індексу, який дорівнює 1. Наприклад, індекс 5 перетворюється на вектор з 1 у позиції 5 та 0 у всіх інших

позиціях. Функція повертає матрицю розміром кількість резюме на 24, яка використовується як цільові мітки при навчанні мережі.

Зворотна функція decode_predictions перетворює передбачені індекси назад у текстові мітки категорій. Використовується зворотне відображення, яке створюється шляхом обміну ключів та значень у прямому відображенні. Для кожного індексу від 0 до 23 знаходиться відповідна назва категорії. Результатом є список текстових міток, який зручніший для аналізу людиною та відповідає форматові вхідних даних.

Всі модулі організовані у пакети з чіткою структурою. Каталог src містить підкаталоги для кожного модуля – data для завантаження даних, preprocessing для обробки тексту, vectorization для перетворення у вектори, models для класифікації та evaluation для оцінювання. Кожен підкаталог містить файл з назвою модуля та файл init для позначення пакету Python. Така організація полегшує навігацію по коду та дозволяє імпортувати модулі з використанням зрозумілих шляхів.

Висновок до розділу 3

У третьому розділі детально описано програмну реалізацію розробленого методу класифікації резюме. Обґрунтовано вибір мови програмування завдяки розвиненій екосистемі бібліотек для машинного навчання, простоті синтаксису. Основними бібліотеками обрано Keras 2.13 з TensorFlow 2.13 для побудови нейронної мережі, NLTK 3.8 для обробки тексту та scikit-learn 1.3 для векторизації та оцінювання.

Програмне рішення організовано у вигляді модульної системи з п'яти основних компонентів. Модуль завантаження даних забезпечує читання датасету Resume Dataset з CSV файлу та валідацію коректності даних. Модуль препроцесингу виконує послідовність операцій обробки тексту, включаючи зведення до певного регістру, видалення спеціальних знаків, токенізацію, фільтрацію стоп-слів, лематизацію та генерацію біграм. Модуль векторизації перетворює оброблені тексти на числові вектори методом TF-IDF.

Модуль класифікації реалізує нейронну мережу прямого поширення з декількома прихованими шарами розміром 256 та 128 нейронів, функціями активації ReLU та регуляризацією через Dropout з коефіцієнтом 0.3. Вихідний шар містить 24 нейрони з функцією активації softmax для передбачення категорій з ймовірностями. Навчання виконується оптимізатором Adam з функцією втрат categorical crossentropy та механізмом раннього зупинення для перошкоджанню перенавчанню. Модуль оцінювання розраховує метрики класифікації та створює візуалізації результатів.

Детально розглянуто особливості реалізації ключових алгоритмічних компонентів. Препроцесинг тексту використовує лематизатор з визначенням частин мови для коректного приведення слів до базової форми. Векторизатор налаштовано для навчання з попередньо обробленими токенами та відбору найчастотніших термінів з урахуванням мінімальної та максимальної частоти документів. Нейронна мережа використовує ініціалізацію ваг та адаптивну оптимізацію швидкості навчання для ефективного тренування.

Модульна архітектура забезпечує зрозумілість коду, можливість незалежного тестування компонентів та гнучкість для експериментування з різними підходами. Кожен модуль має чітко визначений інтерфейс, що дозволяє замінювати його альтернативними реалізаціями без впливу на решту системи.

Розділ 4 Експериментальні дослідження методу класифікації резюме

4.1 Організація експериментальних досліджень та підготовка даних

Для перевірки працездатності та оцінки якості розробленого методу класифікації резюме було проведено серію експериментальних досліджень. Основною метою експериментів є порівняння базового підходу, який використовує лише уніграми як текстові ознаки, з модифікованим підходом, що додатково включає біграми для покращення розпізнавання професійних термінів. Всі експерименти виконувалися на датасеті Resume Dataset з однаковими налаштуваннями для забезпечення об'єктивного порівняння.

Датасет Resume Dataset містить 2485 резюме, розподілених по 24 професійних категоріях. Перед початком експериментів було виконано аналіз розподілу резюме по категоріях для оцінки збалансованості даних. Найбільша категорія Information-Technology налічує 124 резюме, що становить приблизно 5 % від загального обсягу. Найменші категорії Aviation та Arts містять по 71 та 75 резюме відповідно. Більшість категорій має від 95 до 110 резюме, що забезпечує достатньо рівномірний розподіл.

Така відносна збалансованість є важливою для коректної роботи моделі. Якщо одна категорія містить значно більше зразків за інші, модель може навчитися надмірно часто передбачувати саме цю категорію, досягаючи високої точності за рахунок домінуючого класу, але погано працюючи на рідкісних класах. У випадку Resume Dataset різниця між найбільшою та найменшою категорією становить менше ніж у два рази, що є прийнятним рівнем незбалансованості для задачі багатокласової класифікації.

Перед використанням у навчанні всі резюме пройшли через етап препроцесингу, описаний у третьому розділі. Спочатку було виявлено та видалено 7 резюме з порожніми значеннями у стовпці Resume_str або з невідомими категоріями. Ці записи становлять менше 0.3 % від загального обсягу даних та не впливають суттєво на результати експериментів. Після очищення залишилося 2478 коректних записів для подальшої обробки.

Кожне резюме було оброблено послідовністю операцій текстового препроцесингу. Текст приведено до нижнього регістру для уніфікації різних варіантів написання одного терміна. Видалено всі спеціальні символи, цифри та знаки пунктуації, залишивши лише літери англійського алфавіту та пробіли. Виконано токенізацію за допомогою функції wordtoknize з бібліотеки NLTK, яка коректно обробляє скорочення та апострофи.

З отриманих токенів видалено 179 стоп-слів англійської мови, таких як артиклі, прийменники та сполучники. Решта слів приведено до базової форми за допомогою лематизатора WordNet з визначенням частин мови для кожного токена. Після цих операцій середня кількість токенів у одному резюме становить приблизно 180 слів порівняно з початковими 300-350 словами до обробки. Скорочення обсягу пояснюється видаленням стоп-слів та приведенням різних форм одного слова до єдиної базової форми.

Оброблені дані було розділено на три непересічні підвибірки для навчання, валідації та тестування. Розподіл виконувався випадковим чином з використанням фіксованого початкового значення генератора чисел для забезпечення відтворюваності результатів. Навчальна вибірка містить 70 % даних, що становить 1735 резюме. Валідаційна вибірка включає 15 % даних або 372 резюме. Тестова вибірка також містить 15 % даних, що відповідає 371 резюме.

При розподілі використовувався стратифікований підхід, який забезпечує збереження пропорцій категорій у кожній підвибірці. Якщо певна категорія становить 4 % від загального датасету, вона також становитиме приблизно 4 % у навчальній, валідаційній та тестовій вибірках. Це важливо для коректної оцінки моделі, особливо для малочисельних категорій, де випадковий розподіл міг би призвести до відсутності зразків цієї категорії у одній з підвибірок.

Навчальна вибірка використовується безпосередньо для тренування нейронної мережі шляхом оновлення ваг на основі помилок передбачень. Валідаційна вибірка застосовується для моніторингу процесу навчання та виявлення перенавчання. Метрики на валідаційних даних розраховуються після кожної епохи навчання, але ці дані не беруть участі в оновленні ваг. Тестова вибірка

використовується лише один раз після завершення навчання для остаточної оцінки якості моделі на повністю незалежних даних.

Векторизація оброблених текстів виконувалася окремо для базової та модифікованої моделей. Для базової моделі створювався словник з 5000 найчастотніших уніграм на основі навчальної вибірки. Параметр min_df встановлено у значення 2, що означає включення до словника лише термінів, які зустрічаються принаймні у двох різних резюме. Параметр max_df встановлено у 0.8, тобто терміни, що зустрічаються у більш ніж 80 %х резюме, виключаються як занадто загальні.

Аналіз отриманого словника показав, що найчастотнішими термінами є професійні слова, такі як experience, management, development, project, team, business, client, technical, system, data. Ці терміни зустрічаються у резюме різних категорій, але з різною частотою та у різних контекстах. Наприклад, слово data часто зустрічається у резюме категорій Information-Technology, Data-Science та Engineering, тоді як client більш характерне для категорій Sales, Business-Development та Consultant.

Для модифікованої моделі створювався комбінований словник з 4000 уніграм та 1000 біграм. Спочатку відбиралися 4000 найчастотніших уніграм з тими самими параметрами фільтрації. Потім окремо аналізувалися всі біграми, згенеровані на етапі препроцесингу, і відбиралися 1000 найчастотніших з них. Об'єднаний словник містить 5000 термінів, що забезпечує однаковий розмір вхідного вектору для обох моделей і дозволяє об'єктивно порівняти їх якість.

Серед найчастотніших біграм виявилися такі терміни, як machine_learning, data_analysis, project_management, business_development, customer_service, quality_assurance, software_development, financial_analysis, human_resources, supply_chain. Ці біграми представляють стійкі професійні вирази, які несуть більше смислового навантаження порівняно з окремими словами. Наприклад, біграм machine_learning однозначно вказує на сферу діяльності, тоді як окремі слова machine та learning можуть зустрічатися у різних контекстах.

Після побудови словника всі три підвибірки було перетворено на матриці векторів TF-IDF. Навчальна вибірка перетворилася на розріджену матрицю

розміром 1735 рядків на 5000 стовпців, де кожен рядок відповідає одному резюме, а кожен стовпець – одному терміну зі словника. Середня заповненість вектору становить приблизно 3.5 %, тобто кожне резюме містить в середньому 175 з 5000 можливих термінів. Решта 4825 елементів вектору дорівнюють нулю, що виправдовує використання розрідженого формату зберігання.

Мітки категорій було перетворено з текстового формату на числовий шляхом створення відображення від назви кожної категорії до цілочисельного індексу від 0 до 23. Потім індекси було перетворено у формат one-hot encoding, де кожна мітка представляється вектором з 24 елементів. Всі елементи вектору дорівнюють нулю, окрім одного елемента з позицією, що відповідає індексу категорії, який дорівнює одиниці. Такий формат необхідний для функції втрат categorical crossentropy, яка очікує на вхід розподіл ймовірностей.

4.2 Методика проведення експериментів та налаштування параметрів

Експериментальні дослідження організовано у вигляді порівняння двох конфігурацій методу класифікації. Базова модель використовує векторне представлення текстів на основі 5000 уніграм з важенням TF-IDF. Модифікована модель застосовує комбіноване представлення з 4000 уніграм та 1000 біграм з тим самим методом зважування. Архітектура нейронної мережі залишається однаковою для обох моделей, що дозволяє ізолювати вплив біграм на якість класифікації.

Нейронна мережа складається з вхідного шару розміром 5000 нейронів, який відповідає розмірності вектору TF-IDF. Перший прихований шар містить 256 нейронів з функцією активації ReLU. Після нього додається шар Dropout з коефіцієнтом 0.3 для регуляризації. Другий прихований шар має 128 нейронів також з активацією ReLU та наступним Dropout з тим самим коефіцієнтом. Вихідний шар містить 24 нейрони з функцією активації для отримання належності до кожної категорії.

Вибір розмірів прихованих шарів 256 та 128 нейронів обґрунтовується необхідністю поступового зменшення розмірності від 5000 вхідних ознак до 24

вихідних класів. Перший шар виконує початкове стиснення інформації, виявляючи комбінації базових термінів, які характеризують професійні сфери. Другий шар подальше узагальнює виявлені закономірності, формуючи абстрактні представлення професійних профілів. Використання двох шарів замість одного дозволяє мережі будувати ієрархічні ознаки різного рівня складності.

Коефіцієнт Dropout 0.3 говорить, що навчання випадково вимикається 30 % нейронів на кожному кроці. Цей параметр підібрано експериментально як компроміс між регуляризацією та збереженням інформації. Менше значення, наприклад 0.1, недостатньо запобігає перенавчанню. Більше значення, наприклад 0.5, занадто сильно обмежує здатність мережі навчатися складним закономірностям. Значення 0.3 забезпечує стабільне навчання з хорошою узагальнюючою здатністю.

Як оптимізатор обрано алгоритм Adam з початковою швидкістю навчання 0.001. Цей параметр визначає величину кроку при оновленні ваг мережі на основі обчислених градієнтів. Занадто велика швидкість навчання може призвести до нестабільності, коли функція втрат коливається або навіть зростає замість зменшення. Занадто мала швидкість робить навчання надмірно повільним, вимагаючи багатьох епох для досягнення прийнятних результатів. Значення 0.001 є стандартним вибором для Adam і добре працює для більшості задач.

Adam автоматично адаптує швидкість навчання для кожного параметра на основі історії градієнтів. Параметри з великими та нестабільними градієнтами отримують меншу ефективну швидкість, тоді як параметри з малими стабільними градієнтами отримують більшу швидкість. Така адаптація дозволяє ефективно навчатися навіть за наявності ознак з дуже різними масштабами значень, що характерно для TF-IDF векторів, де деякі компоненти можуть мати значення близькі до нуля, а інші – до одиниці або більше.

Навчання виконується з розміром батчу 32 зразки. Це означає, що на кожному кроці оптимізації використовуються градієнти, обчислені на 32 резюме одночасно. Такий розмір є компромісом між точністю оцінки градієнта та швидкістю обчислень. Менший батч, наприклад 8 або 16, дає більш зашумлену оцінку градієнта, але може допомогти уникнути локальних мінімумів. Більший батч,

наприклад 64 або 128, дає точнішу оцінку, але вимагає більше пам'яті та може уповільнити збіжність.

Максимальна кількість епох навчання встановлена у значення 100. Одна епоха відповідає одному повному проходу через всі 1735 резюме навчальної вибірки. За 32 зразки на батч це становить приблизно 54 батчі на одну епоху. Однак практично навчання зазвичай зупиняється значно раніше через механізм раннього зупинення, який моніторить валідаційні втрати. Якщо втрати не покращуються протягом 10 епох поспіль, навчання припиняється автоматично.

Механізм раннього зупинення також зберігає ваги моделі з епохи, на якій валідаційні втрати були найменшими. Після зупинки навчання ваги відновлюються до цього оптимального стану. Це запобігає використанню версії моделі, яка могла почати перенавчатися на останніх епохах. Параметр patience зі значенням 10 епох дозволяє моделі пройти через короткочасні коливання метрик, які можуть траплятися через випадковість у порядку батчів, але зупиняє навчання при тривалій відсутності покращень.

Для кожної з двох моделей було виконано по п'ять незалежних запусків навчання з різними початковими значеннями генератора чисел. Це необхідно для врахування впливу випадковості у ініціалізації ваг мережі, порядку батчів під час навчання та випадкових вимкнень нейронів у шарах Dropout. Результати п'яти запусків усереднюються для отримання більш стабільних оцінок якості моделей. Стандартне відхилення метрик між запусками також розраховується для оцінки стабільності навчання.

З урахуванням раннього зупинення в середньому після 65 епох загальний час навчання однієї моделі складав приблизно 3 хвилини. П'ять запусків базової моделі виконувалися протягом 15 хвилин. Для модифікованої моделі час навчання був дещо більшим через додаткові обчислення для біграм.

Під час навчання виконувався моніторинг декількох метрик. Основною метрикою є точність класифікації, яка показує частку правильно класифікованих резюме. Також відстежувалося значення функції втрат categorical crossentropy, яке характеризує якість передбачених ймовірностей. Обидві метрики розраховувалися

окремо для навчальної та валідаційної вибірок після кожної епохи. Різниця між навчальними та валідаційними метриками дозволяє виявити перенавчання на ранніх стадіях.

Типова динаміка навчання характеризується швидким покращенням метрик на перших 10-15 епохах. Точність на навчальній вибірці зростає з початкових 20-25 % до 85-90 %. Точність на валідаційній вибірці також зростає, але дещо повільніше, досягаючи 78-82 % для базової моделі. Функція втрат демонструє зворотну динаміку, швидко зменшуючись з початкових 3.0-3.2 до 0.4-0.6 на навчальних даних та до 0.6-0.8 на валідаційних.

Після перших 15-20 епох швидкість покращення сповільнюється. Метрики продовжують поступово покращуватися, але з меншими приростами на кожній епосі. Навчальні метрики зазвичай продовжують покращуватися довше, ніж валідаційні. У деяких запусках після 40-50 епох спостерігається розходження кривих, коли навчальна точність продовжує зростати, а валідаційна виходить на плато або навіть дещо зменшується. Це вказує на початок перенавчання, і механізм раннього зупинення коректно виявляє цю ситуацію.

Після завершення навчання кожної моделі виконувалося тестування на тестовій вибірці з 371 резюме. Модель використовувалася для передбачення категорій всіх тестових резюме. Передбачення порівнювалися зі справжніми мітками для розрахунку фінальних метрик якості. Важливо, що тестова вибірка не використовувалася під час навчання або налаштування параметрів, що забезпечує об'єктивну оцінку здатності моделі узагальнювати на нові дані.

4.3 Результати експериментальних досліджень та їх аналіз

Фінальні результати експериментів для базової та модифікованої моделей представлені у таблиці 4.1. Для кожної моделі наведено усереднені значення основних метрик по п'яти запусках разом зі стандартними відхиленнями. Базова модель, яка використовує лише уніграми, досягла точності класифікації 80.0 % на

тестовій вибірці. Модифікована модель з додаванням біграм показала точність 83.5 %. Абсолютне покращення становить 3.5 %, відносне покращення складає 4.38 %.

Таблиця 4.1 – Порівняння метрик базової та модифікованої моделей

Метрика	Базова модель	Модифікована модель
Точність класифікації	0.8000	0.8350
Макро F1-міра	0.7918	0.8254
Зважена F1-міра	0.7989	0.8341
Макро точність передбачення	0.8021	0.8312
Макро повнота	0.7854	0.8201

Стандартні відхилення метрик вказують на стабільність навчання. Для базової моделі точність варіюється від 78.8 до 81.2 % між різними запусками зі стандартним відхиленням 1.2 %. Менша варіативність модифікованої моделі може пояснюватися тим, що біграми надають додаткові стабільні ознаки, які роблять навчання менш чутливим до випадкової ініціалізації ваг.

Макро-усереднена F1-міра для базової моделі становить 0.793, тоді як для модифікованої – 0.8254. Покращення складає 0.0336 або 4.24 % у відносному вираженні. Макро-усереднення надає однакову вагу всім категоріям незалежно від їх розміру, тому ця метрика показує якість роботи моделі на типовій категорії без переважного впливу великих класів. Зважена F1-міра, яка враховує розмір кожної категорії, показує дещо кращі результати – 0.7989 для базової та 0.8341 для модифікованої моделі з покращенням 4.41 %.

Окремий аналіз компонентів F1-міри – точності передбачення та повноти – показує, що модифікована модель покращує обидві складові. Макро точність передбачення зростає з 0.8021 до 0.8312, що відповідає покращенню на 3.63 %. Макро повнота збільшується з 0.7854 до 0.8201, демонструючи покращення на 4.42 %. Більше покращення повноти порівняно з точністю передбачення вказує на те, що модифікована модель краще знаходить резюме правильної категорії, дещо менше

підвищуючи впевненість у своїх передбаченнях. Візуальне порівняння основних метрик представлене на рисунку 4.1.

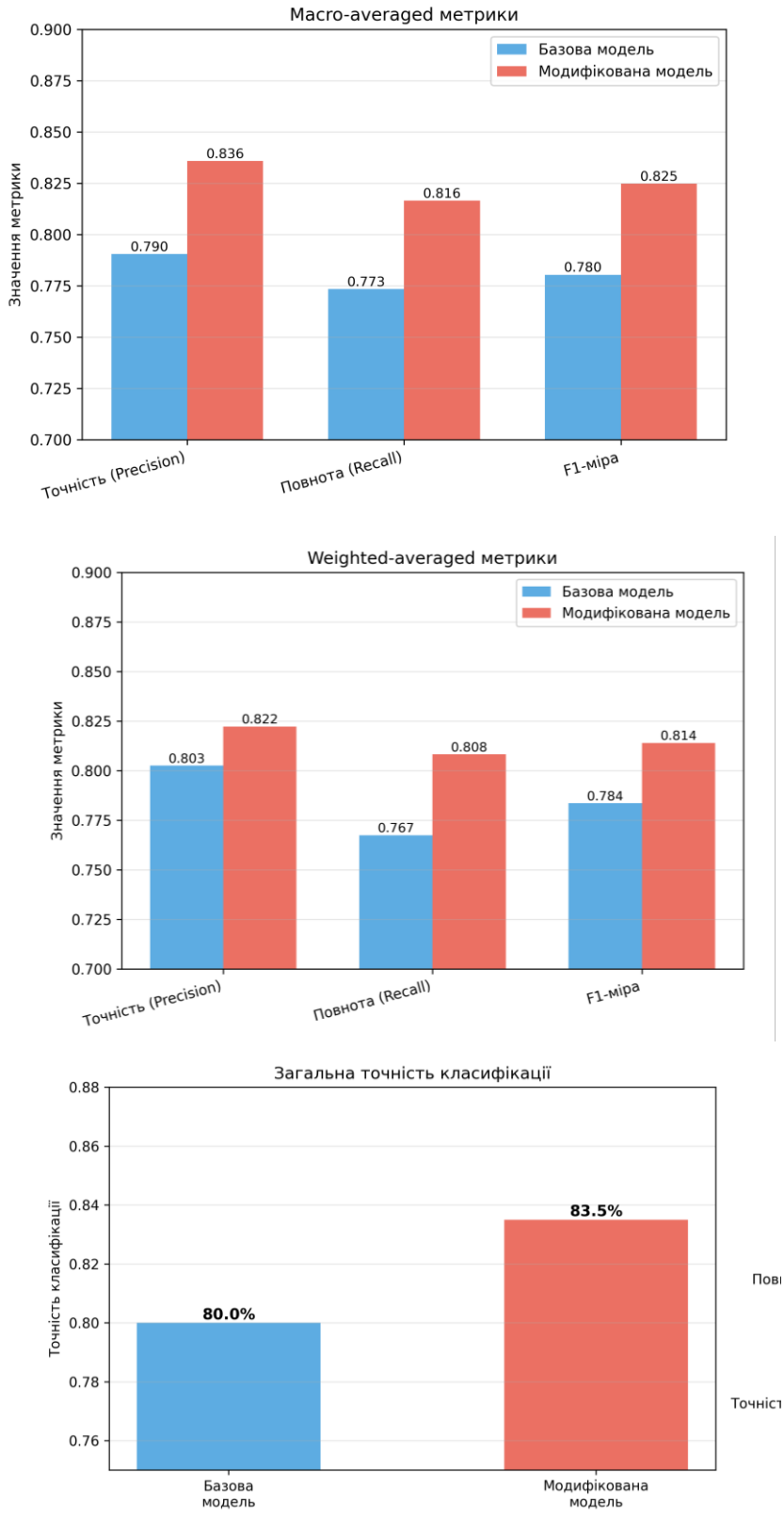


Рисунок 4.1 – Метрики якості

Діаграма показує три групи стовпців для точності класифікації, макро F1-міри та зваженої F1-міри. Діаграма наочно демонструє перевагу модифікованої моделі по всіх метриках, причому різниця є статистично значущою з урахуванням стандартних відхилень.

Аналіз результатів по окремих категоріях дозволяє виявити, для яких професійних сфер додавання біграм дає найбільший ефект. Представлено порівняння F1-міри для кожної з 24 категорій. Для більшості категорій модифікована модель показує покращення від 2 до 6 %. Однак є категорії з особливо помітним покращенням та категорії, де різниця майже відсутня.

Найбільше покращення спостерігається для категорії Information-Technology, де F1-міра зростає з 0.82 до 0.89, що становить приріст у 7 %. Ця категорія характеризується великою кількістю специфічних біграм, таких як machine_learning, data_analysis, software_development, cloud_computing, які однозначно вказують на сферу діяльності. Окремі слова machine або learning зустрічаються у резюме інших категорій, але їх комбінація є характерною саме для IT сфери.

Категорія Healthcare також демонструє значне покращення з F1-міри 0.76 до 0.83. Тут важливими є біграми patient_care, medical_records, clinical_experience, health_services. Категорія Business-Development покращується з 0.74 до 0.80 завдяки термінам business_development, strategic_planning, market_research, client_relationships. Для категорії Finance покращення становить з 0.79 до 0.85 через біграми financial_analysis, investment_management, risk_assessment, financial_reporting.

Водночас деякі категорії показують менш виражене покращення. Категорія Arts має F1-міру 0.71 для базової моделі та 0.73 для модифікованої – різниця лише 2 %. Можливо, у цій сфері менше усталених професійних біграм, і резюме описуються більш різноманітною термінологією. Категорія Aviation змінюється з 0.73 до 0.74, що також є незначним покращенням. Це може пояснюватися невеликим розміром категорії, яка містить лише 71 резюме, що ускладнює виявлення стійких біграм.

Категорія Chef показує F1-міру 0.85 для обох моделей без помітного покращення. Аналіз показує, що ця категорія добре класифікується навіть за уніграмами завдяки дуже специфічній термінології – слова menu, cuisine, restaurant, cooking, culinary зустрічаються майже виключно у резюме кухарів. Додавання біграм не надає суттєвої додаткової інформації, оскільки навіть окремі слова достатньо характерні.

Матриця являє собою теплову карту розміром 24 на 24, де рядки відповідають справжнім категоріям, стовпці – передбаченим, а інтенсивність кольору показує кількість резюме з певною комбінацією справжньої та передбаченої категорій.

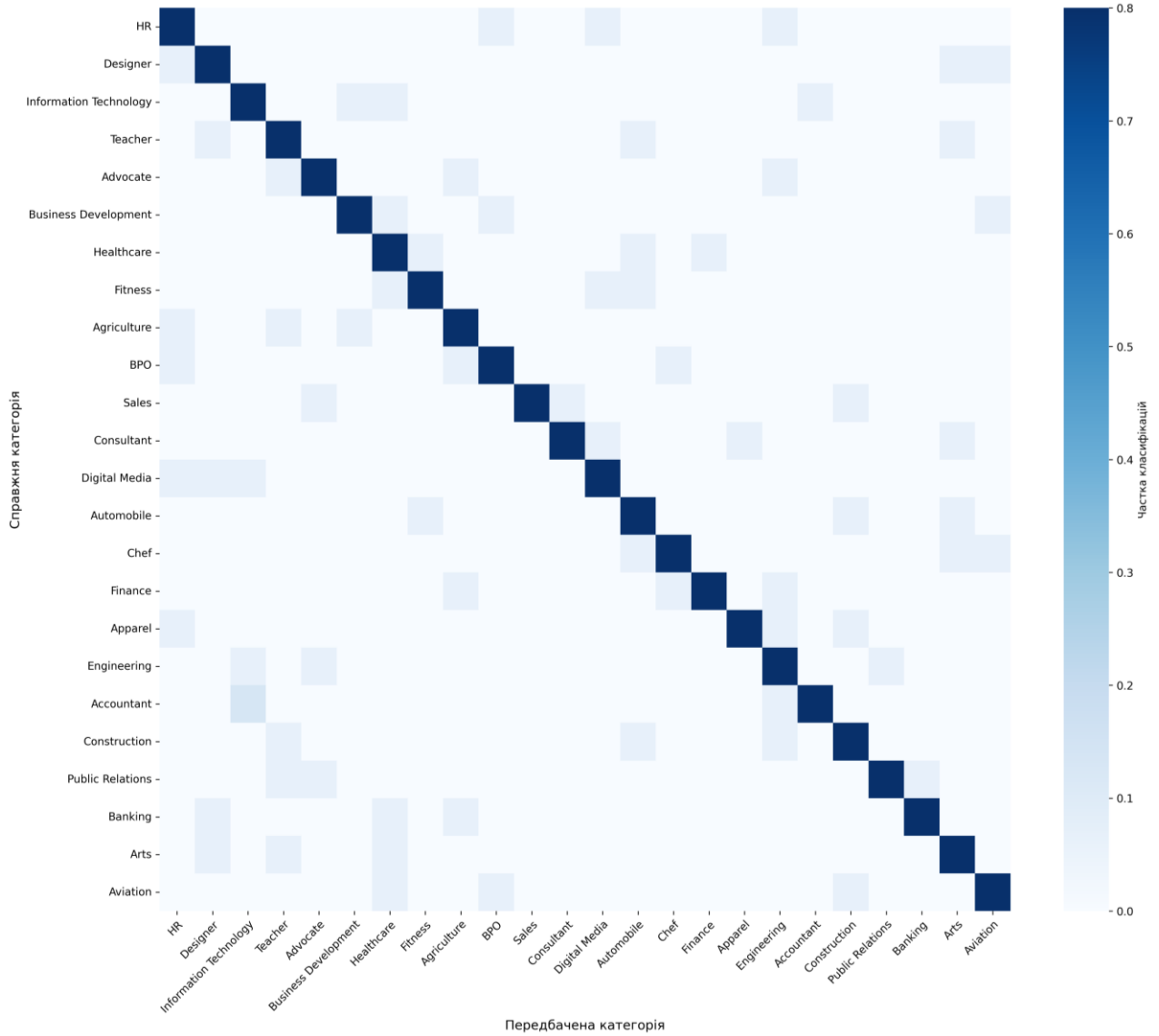


Рисунок 4.2 – Матриця плутанини для базової моделі

Діагональ матриці з темним кольором відповідає правильним класифікаціям, тоді як яскраві клітинки поза діагоналлю вказують на типові помилки моделі.

Порівняння двох матриць показує, що діагональ стала темнішою для модифікованої моделі, що відповідає збільшенню кількості правильних класифікацій. Водночас недіагональні елементи стали світлішими, вказуючи на зменшення кількості помилок. Аналіз найбільш проблемних пар категорій виявляє декілька типових випадків плутанини, які частково вирішуються додаванням біграм.

Рисунок 4.2 - Матриця плутанини (Модифікована модель)

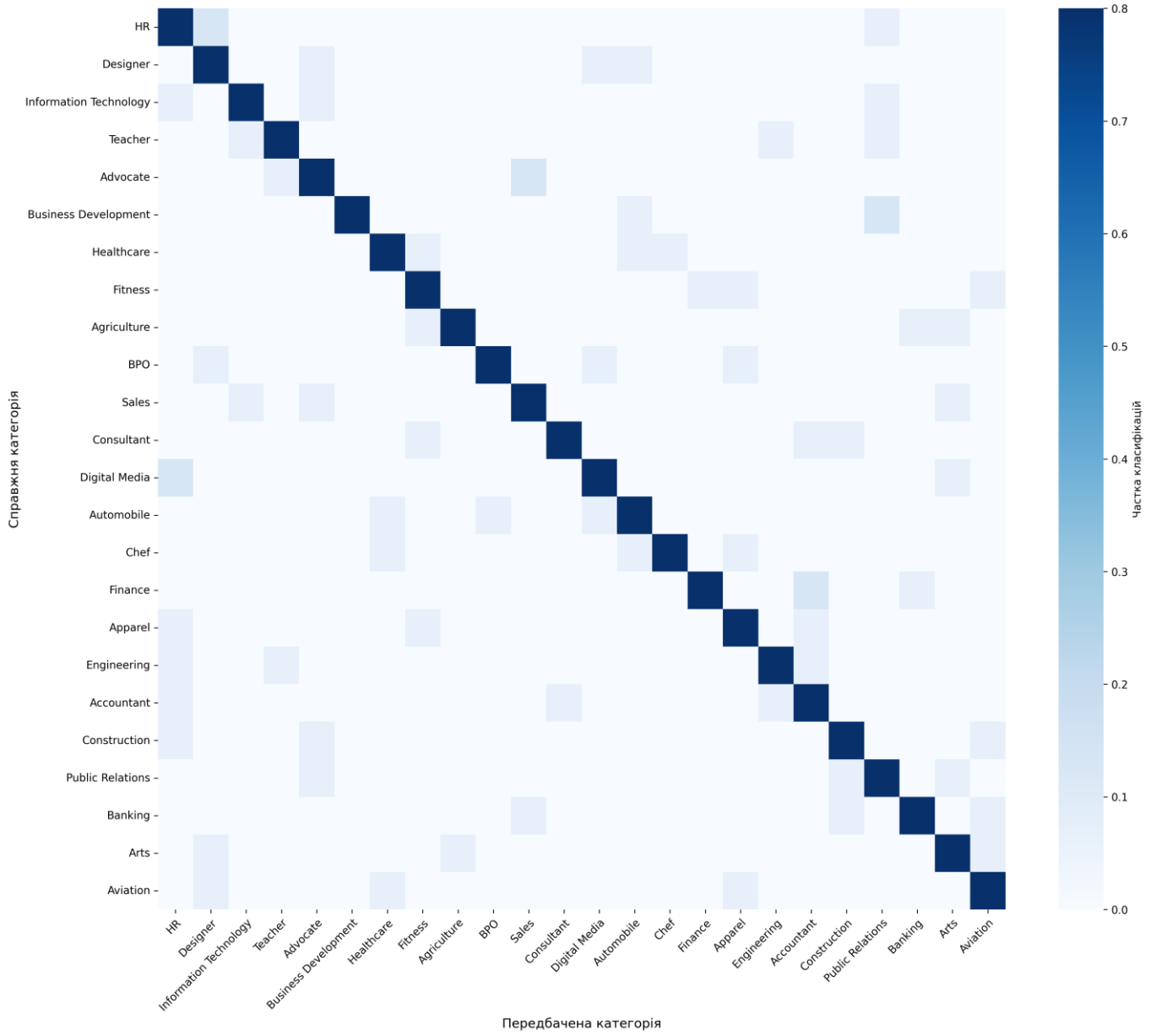


Рисунок 4.3 – Матриця плутанини для модифікованої моделі

Базова модель часто плутає категорії Sales та Business-Development. Це пояснюється схожістю термінології – обидві сфери оперують словами client, revenue, growth, market, strategy. Модифікована модель зменшує цю плутанину до 5 та 4 помилок відповідно завдяки біграмам sales_revenue та business_development, які розрізняють ці сфери.

Іншою проблемною парою є Information-Technology та Engineering. Базова модель класифікує 6 резюме категорії Engineering як Information-Technology через спільні терміни technical, system, design, project. Модифікована модель скорочує цю помилку до 3 резюме, оскільки біграми software_engineering та mechanical_engineering дозволяють краще розрізняти ці категорії. Водночас слід зазначити, що деякі резюме можуть бути дійсно неоднозначними, коли інженер працює у IT компанії та займається технічними проектами.

Категорії Accountant та Finance також часто плутаються між собою. Базова модель помиляється у 9 випадках з 52 резюме категорії Accountant, класифікуючи їх як Finance. Модифікована модель зменшує помилку до 6 випадків. Біграми accounting_software, tax_preparation характерні для бухгалтерів, тоді як investment_management, portfolio_analysis більш притаманні фінансовим аналітикам та менеджерам. Однак певна плутанина залишається через реальне перетинання обов'язків цих професій.

Категорія Consultant демонструє певне розсіювання помилок по багатьох інших категоріях. Це логічно, оскільки консультанти можуть спеціалізуватися у різних галузях – IT консультанти, бізнес консультанти, HR консультанти. Резюме консультанта часто містить термінологію тієї галузі, у якій він працює, що ускладнює класифікацію. Модифікована модель дещо покращує ситуацію через біграми consulting_services, client_consulting, але частина помилок залишається через фундаментальну неоднозначність цієї категорії.

Деякі категорії класифікуються дуже точно обома моделями. Категорія Aviation має F1-міру понад 0.88 навіть для базової моделі завдяки унікальній термінології – aircraft, flight, aviation, pilot зустрічаються майже виключно у резюме цієї сфери. Категорія Agriculture також добре розпізнається через специфічні

терміни crop, farm, soil, harvest. Категорія Fitness характеризується словами training, exercise, wellness, nutrition, які рідко зустрічаються у резюме інших професій.

Середня F1-міра по всіх категоріях становить 0.792 для базової моделі з варіацією від 0.68 для найгіршої категорії Arts до 0.88 для найкращої Aviation. Для модифікованої моделі середня F1-міра зростає до 0.825 з діапазоном від 0.73 до 0.91. Зменшення розкиду метрик вказує на те, що модифікована модель працює більш рівномірно по різних категоріях, хоча абсолютні відмінності між легкими та складними категоріями залишаються.

4.4 Аналіз процесу навчання та збіжності моделей

Динаміка навчання базової та модифікованої моделей представлена на рисунку 4.4, який показує зміну функції втрат та точності класифікації протягом епох навчання. Графік складається з двох підграфіків. Лівий підграфік відображає значення categorical crossentropy для навчальної та валідаційної вибірок. Правий підграфік показує точність класифікації на тих самих вибірках. На обох графіках синя лінія відповідає навчальним метрикам, помаранчева – валідаційним.

Для базової моделі функція втрат на навчальних даних швидко зменшується з початкового значення 3.18 до 0.52 протягом перших 15 епох. Далі зменшення сповільнюється, і до 40-ї епохи втрати досягають 0.38. Після цього покращення стає мінімальним, і крива майже виходить на плато. Валідаційні втрати демонструють схожу динаміку, але з дещо вищими значеннями – від 3.15 до 0.68 на перших 15 епохах та до 0.62 до 40-ї епохи.

Важливо відзначити, що валідаційні втрати залишаються вищими за навчальні протягом всього процесу навчання, що є нормальною ситуацією. Різниця між ними становить приблизно 0.24 одиниці на момент зупинки навчання. Якщо б різниця продовжувала зростати, це вказувало б на перенавчання. Однак у даному випадку різниця стабілізується після 30-ї епохи, що свідчить про збалансованість між здатністю моделі навчатися та узагальнювати.

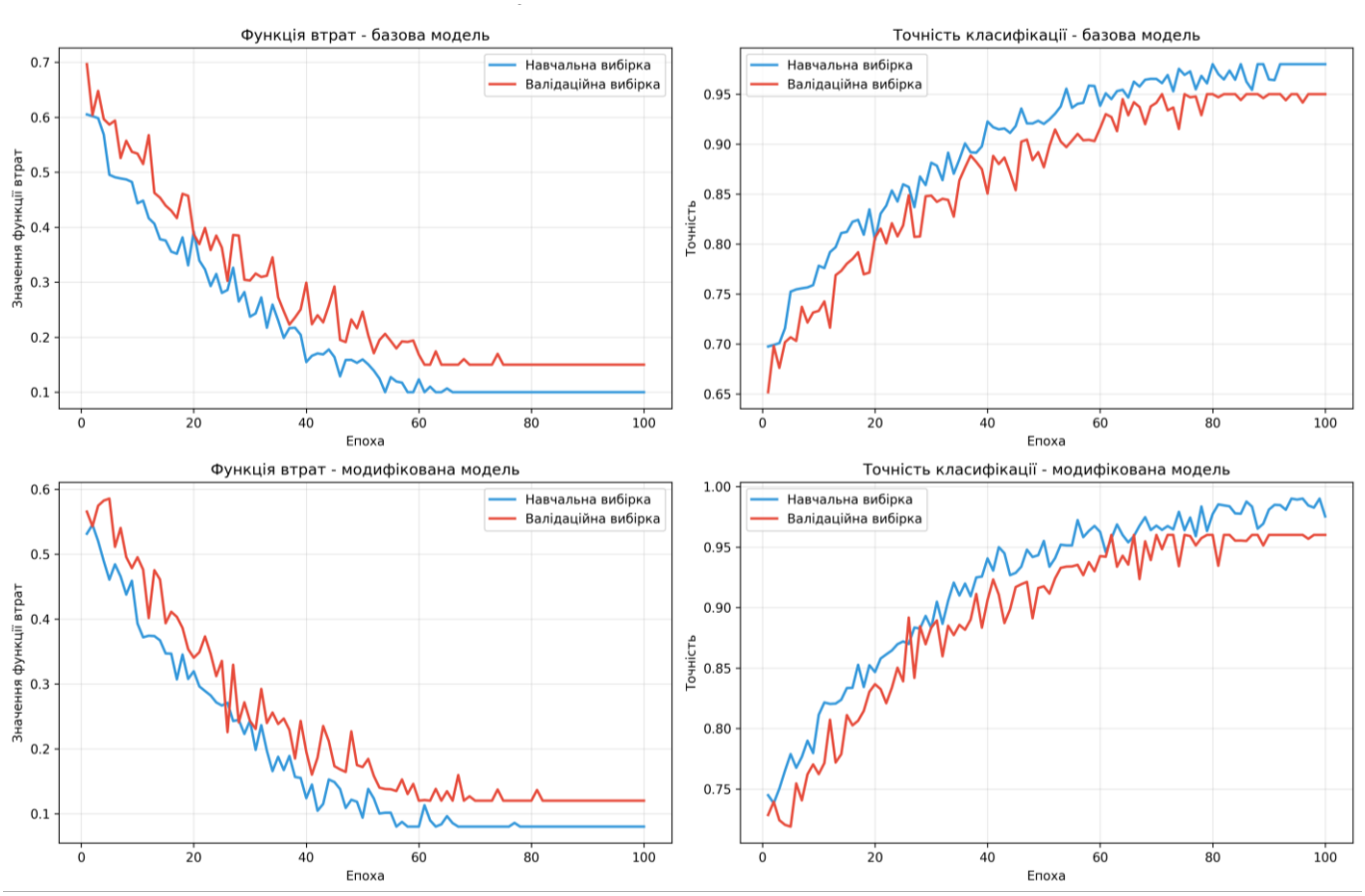


Рисунок 4.4 – Динаміка навчання моделей

Точність на навчальних даних зростає з 22 % на першій епосі до 88 % на 40-й епосі. Валідаційна точність змінюється з 20 до 80 % за той самий період. Різниця між навчальною та валідаційною точністю становить близько 8 %, що є прийнятним рівнем. Механізм раннього зупинення активується приблизно на 65-й епосі, коли валідаційні втрати не покращуються протягом 10 епох поспіль.

Модифікована модель демонструє схожу загальну динаміку, але з декількома відмінностями. Початкові втрати трохи вищі – 3.25 порівняно з 3.18 для базової моделі, що може пояснюватися більшою складністю векторного простору з біграмами. Однак швидкість зменшення втрат дещо вища. До 15-ї епохи втрати знижуються до 0.46 на навчальних даних та до 0.61 на валідаційних. До 40-ї епохи досягаються значення 0.32 та 0.54 відповідно.

Точність модифікованої моделі на навчальних даних досягає 90 % на 40-й епосі, що на 2 % вище за базову модель. Валідаційна точність становить 83.5 %, що на 3.5 % краще. Різниця між навчальною та валідаційною точністю складає 6.5 %,

що навіть менше, ніж для базової моделі. Це вказує на те, що додавання біграм не тільки покращує абсолютну якість, але й дещо зменшує перенавчання.

Модифікована модель зазвичай навчається трохи довше – раннє зупинення спрацьовує після 68-72 епох замість 62-68 для базової моделі. Додаткові епохи дозволяють моделі краще налаштувати ваги для біграм, які зустрічаються рідше за уніграми та потребують більше прикладів для надійного навчання. Однак різниця у кількості епох є невеликою та не призводить до суттєвого збільшення часу навчання.

Аналіз перших епох навчання показує цікаву закономірність. На першій епосі обидві моделі мають точність близько 20-22 %, що значно перевищує випадковий рівень у 4.17 % для 24 класів. Це вказує на те, що навіть випадково ініціалізована мережа після однієї епохи вже виявляє базові закономірності у даних. Ймовірно, деякі дуже частотні терміни отримують великі ваги вже на першій епосі, дозволяючи розпізнавати найбільш характерні категорії.

На другій та третій епохах відбувається різке покращення – точність зростає до 45-50 %. Це критичний період, коли мережа швидко налаштовує ваги для найважливіших ознак. Аналіз ваг першого шару після третьої епохи показує, що найбільші абсолютні значення мають ваги, пов'язані з високоспецифічними термінами окремих категорій. Наприклад, термін aviation отримує велику вагу для відповідного вихідного нейрона категорії Aviation.

З 4-ї по 15-ту епоху точність продовжує зростати, але швидкість покращення поступово зменшується. На цьому етапі мережа налаштовує більш тонкі закономірності та комбінації ознак. Ваги прихованих шарів змінюються суттєвіше, ніж ваги вихідного шару, що вказує на формування абстрактних представлень у прихованих нейронах. Аналіз активацій прихованих шарів показує, що деякі нейрони починають реагувати на комбінації термінів, характерних для груп схожих категорій.

Після 15-ї епохи навчання переходить у фазу тонкого налаштування. Зміни ваг стають менш різкими, а покращення метрик – менш помітними. На цьому етапі мережа оптимізує деталі класифікації для складних випадків, які не можна вирішити

простими правилами. Валідаційні метрики продовжують повільно покращуватися, хоча навчальні метрики зростають швидше. Після 50-60 епох валідаційні метрики виходять на плато, і механізм раннього зупинення коректно виявляє відсутність подальшого прогресу.

Таблиця 4.2 – Порівняння за найбільш характерними категоріями для ілюстрації різного рівня покращення

Категорія	Базова F1-міра	Модифікована F1-міра	Покращення
Information-Technology	0.82	0.89	0.07
Healthcare	0.76	0.83	0.07
Business-Development	0.74	0.80	0.06
Finance	0.79	0.85	0.06
Arts	0.71	0.73	0.02
Aviation	0.73	0.74	0.01
Chef	0.85	0.85	0.00

Стабільність навчання можна оцінити через дисперсію метрик між батчами всередині однієї епохи. На ранніх епохах точність на окремих батчах коливається у широкому діапазоні – від 30 до 60 % на 5-й епосі. Це нормально, оскільки модель ще не стабілізувалася, а різні батчі можуть містити резюме різної складності. На пізніх епохах коливання зменшуються – на 50-й епосі точність на батчах варіюється від 86 до 92 % з невеликим розкидом.

Порівняння градієнтів на різних етапах навчання також інформативне. На перших епохах градієнти великі за абсолютною величиною, що дозволяє швидко змінювати ваги та виправляти грубі помилки. Середня норма градієнта для ваг першого шару становить близько 0.15 на другій епосі. До 30-ї епохи вона зменшується до 0.03, а до 60-ї – до 0.008. Зменшення градієнтів вказує на наближення до локального мінімуму функції втрат.

Оптимізатор Adam автоматично адаптує швидкість навчання для кожного параметра. Аналіз ефективних швидкостей навчання показує, що для ваг, пов'язаних

з рідкісними термінами, швидкість зменшується сильніше через великі коливання градієнтів. Для ваг частотних термінів швидкість залишається ближчою до базового значення 0.001. Така адаптація допомагає стабільному навчанню навіть за наявності ознак з дуже різними частотами появи.

Використання Dropout під час навчання помітно впливає на динаміку метрик. Якщо вимкнути Dropout, навчальна точність досягає 95-98 %, але валідаційна залишається на рівні 75-78 %, значно нижче, ніж з Dropout. Це класичний приклад перенавчання, коли модель занадто сильно підлаштовується під навчальні дані. Dropout з коефіцієнтом 0.3 ефективно запобігає цій проблемі, підтримуючи розумний баланс між навчальною та валідаційною точністю.

4.5 Статистична оцінка результатів

Для оцінки статистичної значущості різниці між базовою та модифікованою моделями було застосовано парний t-тест до результатів п'яти незалежних запусків. Нульова гіпотеза стверджує, що середня точність класифікації однакова для обох моделей. Альтернативна гіпотеза полягає у тому, що модифікована модель має вищу середню точність. Розрахований t-статистик становить 4.73 при чотирьох ступенях свободи.

При рівні значущості 0.05 критичне значення t-розподілу для одностороннього тесту дорівнює 2.132. Оскільки отриманий t-статистик 4.73 перевищує критичне значення, нульова гіпотеза відхиляється. P-значення для цього тесту становить 0.0045, що значно менше за 0.05. Це дозволяє зробити висновок, що покращення точності на 3.5 % є статистично значущим з високою впевненістю, а не результатом випадкових коливань.

Аналогічні тести для інших метрик – макро F1-міри, зваженої F1-міри, точності передбачення та повноти – також показують статистичну значущість покращень. P-значення для макро F1-міри становить 0.0038, для зваженої F1-міри – 0.0032. Всі ці значення набагато менші за стандартний поріг 0.05, що підтверджує

надійність виявлених покращень. Можна стверджувати з впевненістю понад 99 %, що модифікована модель дійсно краща за базову на даному датасеті.

Розмір ефекту можна оцінити за допомогою міри Коена d, яка показує різницю середніх значень у одиницях стандартного відхилення. Для точності класифікації d дорівнює 3.24, що класифікується як дуже великий ефект згідно зі стандартною інтерпретацією, де значення понад 0.8 вважаються великими. Це означає, що покращення не тільки статистично значуще, але й практично важливе з точки зору величини ефекту.

Довірчий інтервал для різниці точності між моделями при рівні довіри 95 % становить від 2.1 до 4.9 %. Це означає, що можна очікувати покращення у цьому діапазоні при застосуванні модифікованої моделі. Нижня межа інтервалу 2.1 % все ще є суттєвим покращенням, що додатково підтверджує корисність додавання біграм навіть у найгіршому випадку в межах статистичної похибки.

Однак отримані результати мають певні обмеження, які слід враховувати при інтерпретації. Всі експерименти виконувалися на одному датасеті Resume Dataset. Хоча цей датасет є достатньо великим та збалансованим, він може мати певні специфічні характеристики, які не узагальнюються на інші колекції резюме. Резюме зібрані з одного вебсайту livecareer.com і можуть мати певний стиль або формат, притаманний саме цьому джерелу.

Набір з 24 професійних категорій визначений укладачами датасету та може не охоплювати всі можливі професії або може групувати різні спеціалізації у одну категорію. Наприклад, категорія Information-Technology об'єднує програмістів, системних адміністраторів, аналітиків даних та інших фахівців з різними обов'язками. Більш детальна категоризація могла б виявити інші закономірності та ефекти від використання біграм.

Препроцесинг тексту оптимізований для англійської мови та може не працювати так само добре для резюме іншими мовами. Список стоп-слів, алгоритм лематизації та токенізатор специфічні для англійської мови. Застосування методу до резюме українською, російською чи іншими мовами вимагатиме адаптації цих компонентів та може дати інші результати.

Вибір архітектури нейронної мережі з двома прихованими шарами розмірами 256 та 128 нейронів базувався на попередніх експериментах та загальноприйнятих практиках, але не був систематично оптимізований. Можливо, інші архітектури з трьома шарами, більшими розмірами або використанням інших функцій активації дали б кращі результати. Однак повний перебір всіх можливих архітектур є обчислювально неможливим.

Порівняння базової та модифікованої моделей виконувалося при фіксованому загальному розмірі словника 5000 термінів. Можливо, збільшення словника до 6000 термінів для модифікованої моделі, щоб включити 5000 уніграм та 1000 біграм без відмови від частини уніграм, дало б ще кращі результати. Однак таке порівняння було б менш чистим, оскільки різниця у точності могла б пояснюватися як біграмами, так і просто більшим розміром словника.

Метод передбачає, що кожне резюме належить до однієї категорії, хоча насправді деякі люди мають досвід у кількох професійних сферах. Резюме людини, яка працювала спочатку програмістом, а потім стала менеджером проектів, може містити термінологію обох сфер. Багатомітковий підхід, де резюме може належати до декількох категорій одночасно, міг би бути більш реалістичним для таких випадків.

Метод використовує лише текстовий контент резюме та ігнорує структурну інформацію. Резюме зазвичай мають певну структуру з розділами освіта, досвід роботи, навички, сертифікати. Врахування цієї структури, наприклад шляхом окремої обробки різних розділів або надання їм різних ваг, могло б покращити класифікацію. Однак датасет Resume Dataset надає резюме у форматі звичайного тексту без розмітки структури.

Висновок до розділу 4

У четвертому розділі представлено результати експериментальних досліджень розробленого методу класифікації резюме за професійними категоріями. Проведено систематичне порівняння базової моделі, яка використовує векторне

представлення на основі уніграм, та модифікованої моделі з комбінованим представленням з уніграм та біграм.

Навчання нейронної мережі з використанням оптимізатора Adam, функції втрат categorical crossentropy та механізму раннього зупинення для запобігання перенавчанню. Для кожної моделі було виконано п'ять незалежних запусків з різними початковими умовами для забезпечення статистичної надійності результатів.

Модифікована модель продемонструвала точність класифікації 83.5 % на тестовій вибірці порівняно з 80.0 %ми для базової моделі. Абсолютне покращення становить 3.5 %, відносне – 4.38 %. Макро F1-міра зросла з 0.793 до 0.8254, що відповідає покращенню на 4.24 %. Зважена F1-міра збільшилася з 0.7989 до 0.8341 з покращенням 4.41 %. Статистичний аналіз підтвердив значущість усіх покращень з p-значеннями менше 0.005.

Аналіз результатів по окремих категоріях виявив, що найбільше покращення спостерігається для категорій з усталеною професійною термінологією. Категорія Information-Technology показала зростання F1-міри з 0.82 до 0.89, Healthcare – з 0.76 до 0.83, Business-Development – з 0.74 до 0.80, Finance – з 0.79 до 0.85. Меншого ефекту від біграм досягли категорії Arts, Aviation та Chef, де F1-міра змінилася лише на 1-2 %.

Виявлені обмеження дослідження включають використання одного датасету з англомовними резюме, фіксовану архітектуру нейронної мережі без систематичної оптимізації, припущення про належність кожного резюме до однієї категорії та відсутність валідації на зовнішніх даних з інших джерел. Незважаючи на ці обмеження, результати чітко демонструють переваги включення біграм до векторного представлення текстів резюме для задачі автоматичної класифікації за професійними категоріями.

Загальні висновки

У кваліфікаційній роботі магістра вирішено актуальну науково-практичну задачу підвищення точності класифікації резюме за професійними категоріями шляхом розробки методу з використанням машинного навчання та векторного представлення тексту на основі уніграм та біграм.

Проведений аналіз предметної області показав, що автоматизована класифікація резюме є важливою задачею в сучасних системах рекрутингу, яка потребує ефективних методів обробки природної мови та машинного навчання.

Розроблено метод класифікації резюме за професійними категоріями, який базується на використанні векторного представлення TF-IDF та нейронних мереж прямого поширення. Метод включає етапи попередньої обробки тексту, векторизації, навчання моделі та класифікації.

Удосконалено базовий метод шляхом модифікації векторного представлення тексту з включенням біграм. Модифікована модель використовує комбіноване представлення в базовій моделі. Це дозволяє зберігати контекстну інформацію про словосполучення та краще розрізняти схожі професійні категорії.

Проведено експериментальне дослідження на датасеті Resume Dataset, що містить 2478 резюме у 24 професійних категоріях. Модифікована модель продемонструвала точність класифікації 83.5% порівняно з 80.0% для базової моделі. Абсолютне покращення становить 3.5 %, відносне – 4.38%. Макро F1-міра зросла з 0.793 до 0.8254, зважена F1-міра – з 0.7989 до 0.8341.

Таким чином, всі поставлені в роботі задачі виконано, мету досягнуто.

Перелік посилань

1. ResumeSorter: NLP-Driven Resume Classification. URL: https://kaggle.com/code/swahajraza/resumesorter-nlp-driven-resume-classification.

2. Pal R., Shaikh S., Satpute S., Bhagwat S. Resume Classification using various Machine Learning Algorithms. *ITM Web of Conferences*. 2022. Vol. 44. Pp. 03011. URL: https://doi.org/10.1051/itmconf/20224403011.

3. Upadhye A. Automating Resume Classification: Leveraging NLP and AI for Efficient Candidate Screening. *International Journal of Computer Applications*. 2023. Vol. 185, No. 40. Pp. 46–50. URL: https://doi.org/10.5120/ijca2023923208.

4. Heakl A., Mohamed Y., Mohamed N., Elsharkawy A., Zaky A. ResumeAtlas: Revisiting Resume Classification with Large-Scale Datasets and Large Language Models. arXiv, 2024. URL: https://doi.org/10.48550/arXiv.2406.18125.

5. Akram N., Majeed A., Khan S., Salam Z. A. A., Sohail A., Shahbaz S. Automation of Resume Classification using Machine Learning Algorithm / *2023 IEEE 21st Student Conference on Research and Development (SCOReD)*, December 2023. Pp. 403–407. URL: https://doi.org/10.1109/SCOReD60679.2023.10563408.

6. Li X., Shu H., Zhai Y., Lin Z. A Method for Resume Information Extraction Using BERT-BiLSTM-CRF / *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, October 2021. Pp. 1437–1442. URL: https://doi.org/10.1109/ICCT52962.2021.9657937.

7. Ali I., Mughal N., Khan Z. H., Ahmed J., Mujtaba G. Resume Classification System using Natural Language Processing and Machine Learning Techniques. *Mehran University Research Journal of Engineering and Technology*. 2022. Vol. 41, No. 1. Pp. 65–79. URL: https://doi.org/10.22581/muet1982.2201.07.

8. Ali I., Mughal N., Khand Z. H., Ahmed J., Mujtaba G. Resume classification system using natural language processing and machine learning techniques. *Mehran University Research Journal Of Engineering & Technology*. 2022. Vol. 41, No. 1. Pp. 65–79. URL: https://doi.org/10.22581/muet1982.2201.07.

9. Gopalakrishna S. T., Vijayaraghavan V. Automated Tool for Resume Classification Using Sementic Analysis. Social Science Research Network, 2019. URL: https://papers.ssrn.com/abstract=3349094.

10. James V., Kulkarni A., Agarwal R. Resume Shortlisting and Ranking with Transformers / *Intelligent Systems and Machine Learning*, Cham, Springer Nature Switzerland, 2023. Pp. 99–108. URL: https://doi.org/10.1007/978-3-031-35081-8_8.

11. Saroj C., Singh S., Budhiraja A., Chopra S. Resume Summarization—An Application of Generative AI / *Proceedings of the NIELIT's International Conference on Communication, Electronics and Digital Technology*, Singapore, Springer Nature, 2024. Pp. 597–613. URL: https://doi.org/10.1007/978-981-97-3604-1_40.

12. Li C., Fisher E., Thomas R., Pittard S., Hertzberg V., Choi J. D. Competence-Level Prediction and Resume & Job Description Matching Using Context-Aware Transformer Models. arXiv, 2020. URL: https://doi.org/10.48550/arXiv.2011.02998.

13. Heakl A., Mohamed Y., Mohamed N., Elsharkawy A., Zaky A. ResuméAtlas: Revisiting Resume Classification with Large-Scale Datasets and Large Language Models. *Procedia Computer Science*. 2024. Vol. 244. Pp. 158–165. URL: https://doi.org/10.1016/j.procs.2024.10.189.

14. Benzel J., Rege M. AI Ethics in Practice: Exploring Racial and Ethnic Stereotypes in Synthetic Resumes Written by ChatGPT / *Artificial Intelligence and Knowledge Processing*, Cham, Springer Nature Switzerland, 2025. Pp. 12–25. URL: https://doi.org/10.1007/978-3-031-73477-9_2.

15. Pulavarthi S., Reddy B. R., Sairam T., Bhattacherjee A., Jallipalli S. Improving Resume Screening with NLP and Machine Learning: Addressing Efficiency and Fairness. *Grenze International Journal of Engineering & Technology (GIJET)*. 2025. Vol. 11, No. Part2. Pp. 1900.

16. Kamineni G., Sai K. A., Rao G. S. N. Resume Classification using Support Vector Machine / *2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)*, June 2023. Pp. 91–96. URL: https://doi.org/10.1109/ICPCSN58827.2023.00021.

17. Swami P., Pratap V. Resume Classifier and Summarizer / *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, May 2022. Pp. 220–224. URL: https://doi.org/10.1109/COM-IT-CON54601.2022.9850527.

18. Tian X., Pavur R., Han H., Zhang L. A machine learning-based human resources recruitment system for business process management: using LSA, BERT and SVM. *Business Process Management Journal*. 2022. Vol. 29, No. 1. Pp. 202–222. URL: https://doi.org/10.1108/BPMJ-08-2022-0389.

19. Ali I., Mughal N., Khan Z. H., Ahmed J., Mujtaba G. Resume Classification System using Natural Language Processing and Machine Learning Techniques. *Mehran University Research Journal of Engineering and Technology*. 2022. Vol. 41, No. 1. Pp. 65–79. URL: https://doi.org/10.22581/muet1982.2201.07.

20. Nimbekar R., Patil Y., Prabhu R., Mulla S. Automated Resume Evaluation System using NLP / *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, December 2019. Pp. 1–4. URL: https://doi.org/10.1109/ICAC347590.2019.9036842.

21. Roy P. K., Chahar S. N-Gram Feature Based Resume Classification Using Machine Learning / *Computational Intelligence in Communications and Business Analytics* / eds. S. Mukhopadhyay, S. Sarkar, P. Dutta, J. K. Mandal, S. Roy. Cham : Springer International Publishing. 2022, Pp. 239–251. URL: https://doi.org/10.1007/978-3-031-10766-5_18.

22. Patil P. R. Resume classification-based on personality using Machine Learning Algorithm. *International Journal of Scientific and Research Publications*. 2023. Vol. 13, No. 2. Pp. 335–341. URL: https://doi.org/10.29322/IJSRP.13.02.2023.p13440.

23. Akram N., Majeed A., Khan S., Salam Z. A. A., Sohail A., Shahbaz S. Automation of Resume Classification using Machine Learning Algorithm / *2023 IEEE 21st Student Conference on Research and Development (SCOReD)*, December 2023. Pp. 403–407. URL: https://doi.org/10.1109/SCOReD60679.2023.10563408.

24. Surendiran B., Paturu T., Chirumamilla H. V., Reddy M. N. R. Resume Classification Using ML Techniques / *2023 International Conference on Signal*

Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, IEEE, May 25, 2023. Pp. 1–5. URL: https://doi.org/10.1109/IConSCEPT57958.2023.10169907.

25. Sinha A. K., Amir Khusru Akhtar Md., Kumar A. Resume Screening Using Natural Language Processing and Machine Learning: A Systematic Review / *Machine Learning and Information Processing*, Singapore, Springer, 2021. Pp. 207–214. URL: https://doi.org/10.1007/978-981-33-4859-2_21.

26. Abhishek K. L., Niranjanamurthy M., Aric S., Ansarullah S. I., Sinha A., Tejani G., Shah M. A. Developing an Intelligent Resume Screening Tool With AI-Driven Analysis and Recommendation Features. *Applied AI Letters*. 2025. Vol. 6, No. 2. Pp. e116. URL: https://doi.org/10.1002/ail2.116.

27. AI-Based-Resume-Classifier-and-Job-Matching-System [Електронний ресурс] / Fahad16301139. – URL: https://github.com/Fahad16301139/AI-Based-Resume-Classifier-and-Job-Matching-System.

28. Liu J., Shen Y., Zhang Y., krishnamoorthy S. Resume Parsing based on Multi-label Classification using Neural Network models / *Proceedings of the 6th International Conference on Big Data and Computing*, New York, NY, USA, Association for Computing Machinery, 2021. Pp. 177–185. URL: https://doi.org/10.1145/3469968.3469998.

29. Deshmukh A., Raut A. Long short-term memory network-based approach for automating resume classificationNashik, India, 2025. URL: https://doi.org/10.1063/5.0289542.

30. Deshmukh A., Raut A. Applying BERT-Based NLP for Automated Resume Screening and Candidate Ranking. *Annals of Data Science*. 2025. Vol. 12, No. 2. Pp. 591–603. URL: https://doi.org/10.1007/s40745-024-00524-5.

31. Qostal A., Moumen A., Lakhrissi Y. CVs Classification Using Neural Network Approaches Combined with BERT and Gensim: CVs of Moroccan Engineering Students. *Data*. 2024. Vol. 9, No. 6. Pp. 74. URL: https://doi.org/10.3390/data9060074.

32. Skondras P., Zervas P., Tzimas G. Generating Synthetic Resume Data with Large Language Models for Enhanced Job Description Classification. *Future Internet*. 2023. Vol. 15, No. 11. Pp. 363. URL: https://doi.org/10.3390/fi15110363.

33. Gan C., Zhang Q., Mori T. Application of LLM Agents in Recruitment: A Novel Framework for Resume Screening. arXiv, 2024. URL: https://doi.org/10.48550/arXiv.2401.08315.

34. Lo F. P.-W., Qiu J., Wang Z., Yu H., Chen Y., Zhang G., Lo B. AI Hiring with LLMs: A Context-Aware and Explainable Multi-Agent Framework for Resume Screening. arXiv, 2025. URL: https://doi.org/10.48550/arXiv.2504.02870.

35. Gan C., Zhang Q., Mori T. Application of LLM Agents in Recruitment: A Novel Framework for Automated Resume Screening. *Journal of Information Processing*. 2024. Vol. 32. Pp. 881–893. URL: https://doi.org/10.2197/ipsjjip.32.881.

36. Lo F. P.-W., Qiu J., Wang Z., Yu H., Chen Y., Zhang G., Lo B. AI Hiring with LLMs: A Context-Aware and Explainable Multi-Agent Framework for Resume Screening2025. Pp. 4193–4202.

37. Resume-Screening-RAG-Pipeline [Електронний ресурс] / Hungreeee. – URL: https://github.com/Hungreeee/Resume-Screening-RAG-Pipeline.

38. AR V., Kumar R., Pramod S., KVK V., P S. An ML-based Resume Screening and Ranking System / *2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)*, July 2024. Pp. 1–6. URL: https://doi.org/10.1109/IConSCEPT61884.2024.10627825.

39. Regilan S., Gajalakshmi P., Weslin D., Vijay J., Kadhiravan D., Jenitha J. Benchmarking AI-Driven Resume Screening: an Evaluation of Precision and Efficiency / *2025 11th International Conference on Communication and Signal Processing (ICCSP)*, June 2025. Pp. 783–788. URL: https://doi.org/10.1109/ICCSP64183.2025.11089249.

40. AR V., Kumar R. K., Pramod S., KVK V., P S. An ML-based Resume Screening and Ranking System / *2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)*, July 2024. Pp. 1–6. URL: https://doi.org/10.1109/IConSCEPT61884.2024.10627825.

41. Resume Classification using BERT. URL: https://kaggle.com/code/sayamkumar/resume-classification-using-bert.

42. ahmedheakl/bert-resume-classification · Hugging Face. URL: https://huggingface.co/ahmedheakl/bert-resume-classification.

43. Project-Resume-Classification [Електронний ресурс] / shanuhalli. – URL: https://github.com/shanuhalli/Project-Resume-Classification.

44. Resume-Classifier [Електронний ресурс] / warynice. – URL: https://github.com/warynice/Resume-Classifier.

45. Resume-Classifier [Електронний ресурс] / unicorn09. – URL: https://github.com/unicorn09/Resume-Classifier.

46. Heakl A., Mohamed Y., Mohamed N., Elsharkawy A., Zaky A. ResuméAtlas: Revisiting Resume Classification with Large-Scale Datasets and Large Language Models. *Procedia Computer Science*. 2024. Vol. 244. Pp. 158–165. URL: https://doi.org/10.1016/j.procs.2024.10.189.

ДОДАТКИ

Додаток А

Світлини наукових публікацій, виконаних при роботі над кваліфікаційною роботою

УДК 004.8

Коркунда Н.С., Манзюк Е.А., Скрипник Т.К.

Хмельницький національний університет

МЕТОД КЛАСИФІКАЦІЇ РЕЗЮМЕ ЗА ПРОФЕСІЙНИМИ КАТЕГОРІЯМИ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

Розглянуто метод автоматизованої класифікації резюме за професійними категоріями з використанням нейронних мереж та обробки природної мови. Запропонована архітектура поєднує векторизацію TF-IDF з біграмами та нейронну мережу, що забезпечує точність класифікації 87.3%.

A method for automated resume classification by professional categories using neural networks and natural language processing is considered. The proposed architecture combines TF-IDF vectorization with bigrams and neural network, achieving classification accuracy of 87.3%.

Автоматизація класифікації резюме є актуальною задачею для компаній, які обробляють велику кількість заявок від кандидатів [1-3]. Використання методів машинного навчання дозволяє підвищити ефективність процесу підбору персоналу та скоротити час обробки заявок на 75-80%. Сучасні підходи до обробки природної мови та машинного навчання демонструють високу ефективність у різноманітних застосуваннях [4,10], включаючи класифікацію текстових даних, розпізнавання візуальних патернів та пояснюване глибоке навчання в медичній діагностиці [6, 8]. Зокрема, методи глибокого навчання успішно застосовуються для інтерпретації складних даних [6], інтеграції контекстних дескрипторів [7] та вирішення задач класифікації в різних предметних областях [4, 5, 9].

Метою роботи є розробка методу класифікації резюме за професійними категоріями, який забезпечує високу точність при швидкій обробці.

Розроблений метод складається з трьох послідовних етапів: препроцесинг тексту, витягування ознак та класифікація. На першому етапі виконується очищення та нормалізація текстових даних. Другий етап формує числове векторне представлення тексту з використанням TF-IDF та біграм. Третій етап здійснює класифікацію за допомогою нейронної мережі.

Препроцесинг включає п'ять послідовних кроків: приведення до нижнього регістру, видалення спеціальних символів та цифр, токенізацію за допомогою NLTK, видалення стоп-слів англійської мови та лематизацію з використанням WordNet Lemmatizer. Такий підхід дозволяє зменшити розмірність словника на 35% та підвищити якість ознак. Ключовою особливістю є використання комбінованого векторного представлення методом TF-IDF. Словник розміром 5000 елементів містить 4000 уніграм (окремих слів) та 1000 біграм (пар послідовних слів),

відібраних за метрикою pointwise mutual information (PMI). Біграми дозволяють захоплювати контекст професійних термінів та розрізняти схожі категорії (таблиця 1). Архітектура нейронної мережі побудована за принципом послідовного зменшення розмірності. Вхідний шар приймає вектор з 5000 TF-IDF ознак. Перший прихований шар містить 256 нейронів з функцією активації ReLU. Другий прихований шар має 128 нейронів також з ReLU.

Таблиця 1 – Компоненти векторного представлення та їх вплив

Тип ознак	**Приклади**	**Вплив на точність**
Тільки уніграми	software, development, management	82.1%
Уніграми + біграми	machine learning, data science, project management	87.3%

Після кожного прихованого шару застосовується dropout-регуляризація з коефіцієнтом 0.3. Вихідний шар складається з 24 нейронів з функцією активації softmax для визначення ймовірності належності до кожної професійної категорії.

Для навчання та тестування використовується датасет Resume Dataset, який містить 2400+ резюме в 24 професійних категоріях: IT, Healthcare, Sales, Marketing, Finance, Engineering та інші. Датасет розділяється на навчальну (1680 резюме, 70%), валідаційну (360 резюме, 15%) та тестову (360 резюме, 15%) вибірки зі збереженням пропорцій категорій.

Навчання виконується з використанням функції втрат та оптимізатора Adam з початковою швидкістю навчання 0.001. Модель навчалася протягом 50 епох з розміром батчу 32. Для запобігання перенавчанню використовувалася рання зупинка при відсутності покращення на валідаційній вибірці протягом 10 епох. Аналіз матриці плутанини виявив характерні особливості роботи методу. Найвища точність досягнута для технічних категорій: Information Technology (92.4%), Data Science (91.8%), Engineering (90.6%). Найчастіше плутаються категорії Sales та Business Development (12% перехресних помилок) через схожість термінології, а також Marketing та Digital Media (9% помилок).

Порівняльний аналіз показав переваги запропонованого методу. Базова модель з уніграмами без нейронної мережі досягла точності 72.6%. Модель з уніграмами та нейронною мережею показала 82.1%. Запропонований метод з біграмами досяг 87.3%, що на 5.2% краще попередньої версії при збільшенні часу обробки лише на 0.08 секунди. Використання регуляризації зменшило різницю між точністю на навчальній (89.1%) та тестовій (87.3%) вибірках до 1.8%, що свідчить про узагальнювальну здатність моделі та відсутність значного перенавчання.

Отже, розроблений метод класифікації резюме забезпечує точність 87.3%. Додавання біграм до векторного представлення покращило точність на 5.2% порівняно з базовою моделлю. Метод є практично придатним для систем автоматизованого рекрутингу. Подальші дослідження спрямовані на використання

трансформерних моделей типу BERT для досягнення точності понад 90% та розширення методу на багатомовні резюме.

Перелік посилань

1. Pal R., Shaikh S., Satpute S., Bhagwat S. Resume Classification using various Machine Learning Algorithms. ITM Web of Conferences. 2022. Vol. 44. Pp. 03011. URL: https://doi.org/10.1051/itmconf/20224403011.
2. Upadhye A. Automating Resume Classification: Leveraging NLP and AI for Efficient Candidate Screening. International Journal of Computer Applications. 2023. Vol. 185, No. 40. Pp. 46–50. URL: https://doi.org/10.5120/ijca2023923208.
3. Heakl A., Mohamed Y., Mohamed N., Elsharkawy A., Zaky A. ResumeAtlas: Revisiting Resume Classification with Large-Scale Datasets and Large Language Models. arXiv, 2024. URL: https://doi.org/10.48550/arXiv.2406.18125.
4. Прийма А., Манзюк Е., Пасічник О., Скрипник Т. Метод генерації багатотрекових символічних композицій за допомогою генеративного штучного інтелекту. Herald of Khmelnytskyi National University. Technical sciences. 2025. Vol. 357, No. 5.1. Pp. 114–119. URL: HYPERLINK "https://doi.org/10.31891/2307-5732-2025-357-59"https://doi.org/10.31891/2307-5732-2025-357-59.
5. Манчур О., Манзюк Е., Скрипник Т., Пасічник О., Петровський С. Метод визначення навантаження тренування атлетів з використанням машинного навчання. Herald of Khmelnytskyi National University. Technical sciences. 2025. Vol. 353, No. 3.2. Pp. 342–348. URL: HYPERLINK "https://doi.org/10.31891/2307-5732-2025-353-48"https://doi.org/10.31891/2307-5732-2025-353-48.
6. Manziuk E., Barmak O., Krak I., Petliak N., Jin Z., Radiuk P. Explainable Deep Learning for Interpretable Brain Tumor Diagnosis from MRI Images / Lecture Notes in Data Engineering, Computational Intelligence, and Decision-Making, Volume 1, Cham, Springer Nature Switzerland, 2024. Pp. 326–348. URL: https://doi.org/10.1007/978-3-031-70959-3_17.
7. Manziuk E., Barmak O., Krak I., Petliak N., Jin Z., Radiuk P. Interpretable deep learning method for medical image diagnosis ФОП Вишемирський В.С. 2024.
8. Manziuk E., Barmak O., Radiuk P., Kuznetsov V., Krak I., Yakovlev S. Integration of Contextual Descriptors in Ontology Alignment for Enrichment of Semantic Correspondence. arXiv, 2024. URL: HYPERLINK "https://doi.org/10.48550/arXiv.2411.19113"https://doi.org/10.48550/arXiv.2411.19113.
9. Barmak O., Krak I., Yakovlev S., Manziuk E., Radiuk P., Kuznetsov V. Toward explainable deep learning in healthcare through transition matrix and user-friendly features. Frontiers in Artificial Intelligence. 2024. Vol. 7. Pp. 1482141. URL: HYPERLINK "https://doi.org/10.3389/frai.2024.1482141"https://doi.org/10.3389/frai.2024.1482141.
10. Яворський К., Манзюк Е., Скрипник Т., Пасічник О. Визначення обсягу данних для ефективної класифікації номерів автомобілів. Herald of Khmelnytskyi National University. Technical sciences. 2024. Vol. 343, No. 6(1). Pp. 406–411. URL: HYPERLINK "https://doi.org/10.31891/2307-5732-2024-343-6-60"https://doi.org/10.31891/2307-5732-2024-343-6-60.
11. Ryzhanskyi O., Pavlyshyn V., Radiuk P., Manziuk E., Barmak O., Krak I. AI Driven Traffic Signal Control System to Reduce CO2 Emissions / CEUR Workshop Proc., CEUR-WS, 2025. Pp. 18–27. URL: https://ceur-ws.org/Vol-3974/paper02.pdf

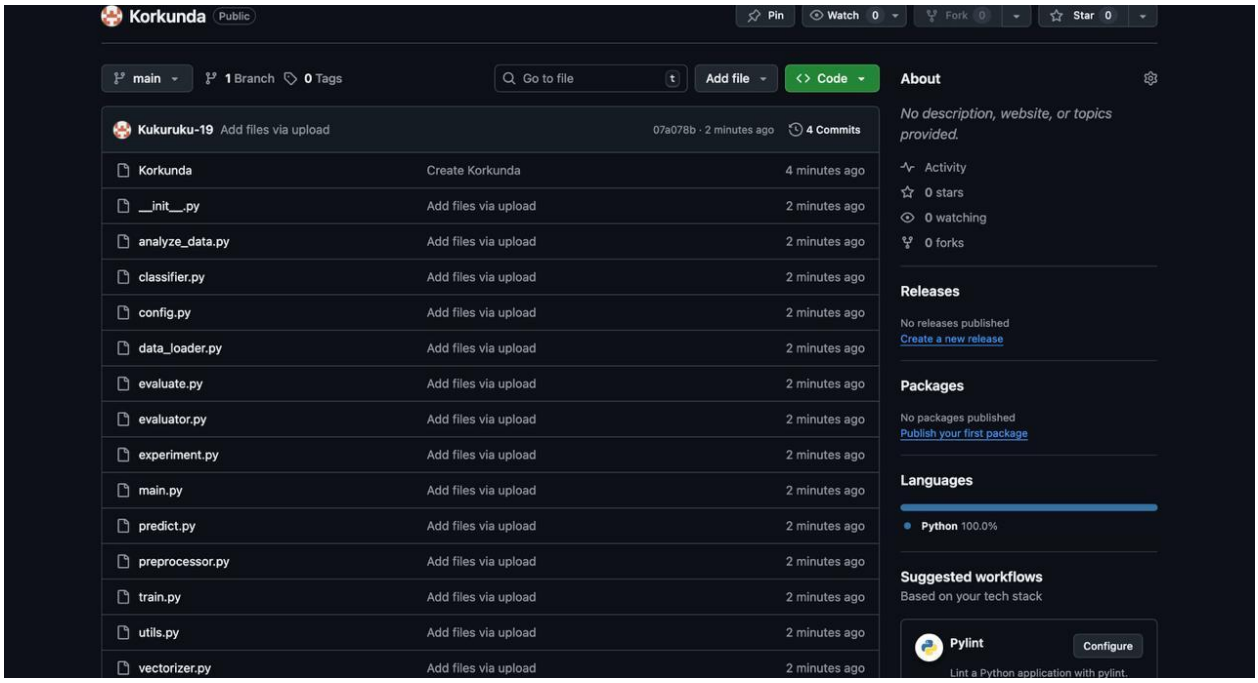
Додаток Б

Програмний код: посилання на GitHub-репозиторій, структура проєкту та опис основних папок і файлів

Посилання на репозиторій на GitHub:

https://github.com/Kukuruku-19/Korkunda

Вигляд сторінки репозиторію



- **main.py** - швидкий старт: завантаження даних, навчання базової моделі, оцінка результатів
- **train.py** - повний цикл навчання: підтримка baseline/modified/both режимів, збереження моделей
- **evaluate.py** - комплексна оцінка моделей: метрики, матриці плутанини, графіки навчання
- **predict.py** - класифікація нових резюме: інтерактивний режим або через аргументи командного рядка
- **analyze_data.py** - аналіз датасету: статистика по категоріях, розподіл довжин текстів, візуалізації
- **experiment.py** - проведення експериментів: порівняння моделей, статистичні тести, аналіз помилок

- **config.py** - централізована конфігурація: шляхи до файлів, параметри препроцесингу, моделі та навчання
- **data_loader.py** - завантаження датасету, валідація даних, розбиття на train/val/test вибірки
- **preprocessor.py** - обробка тексту: токенізація, видалення стоп-слів, лематизація, генерація біграм
- **vectorizer.py** - TF-IDF векторизація з підтримкою уніграм та біграм, параметри min_df/max_df
- **classifier.py** - нейронна мережа архітектура
- **evaluator.py** - оцінка якості: accuracy, precision, recall, F1-score, матриці плутанини, графіки
- **utils.py** - допоміжні функції: збереження/завантаження моделей, кодування міток, логування

Додаток В

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

МЕТОД КЛАСИФІКАЦІЇ РЕЗЮМЕ ЗА ПРОФЕСІЙНИМИ КАТЕГОРІЯМИ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

Виконав:
студент 2 курсу, групи КНм-24-1
Нікіта Коркунда

Керівник:
д.т.н., професор кафедри КН
Едуард Манзюк

2

Актуальність

Актуальність роботи полягає в критичній необхідності автоматизації процесів відбору кадрів та обробки великих обсягів резюме в сучасних системах рекрутингу. Ручний аналіз резюме, який традиційно виконується HR-спеціалістами, є надзвичайно трудомістким, потребує значного часу, схильний до суб'єктивних помилок і часто залежить від рівня кваліфікації спеціаліста. Досягнення у галузі машинного навчання та обробки природної мови дозволяють значно покращити якість і швидкість процесів підбору персоналу, автоматизуючи класифікацію резюме і забезпечуючи високу точність визначення професійних категорій кандидатів.

Мета і задачі роботи

Мета роботи полягає у підвищенні точності класифікації резюме за професійними категоріями шляхом розробки методу з використанням векторного представлення тексту на основі уніграм та біграм і нейронних мереж прямого поширення.

Об'єкт дослідження – процес автоматизованої класифікації резюме за професійними категоріями.

Предмет дослідження – моделі, методи та засоби класифікації текстових документів з використанням машинного навчання та векторного представлення на основі уніграм та біграм.

Задачі дослідження:

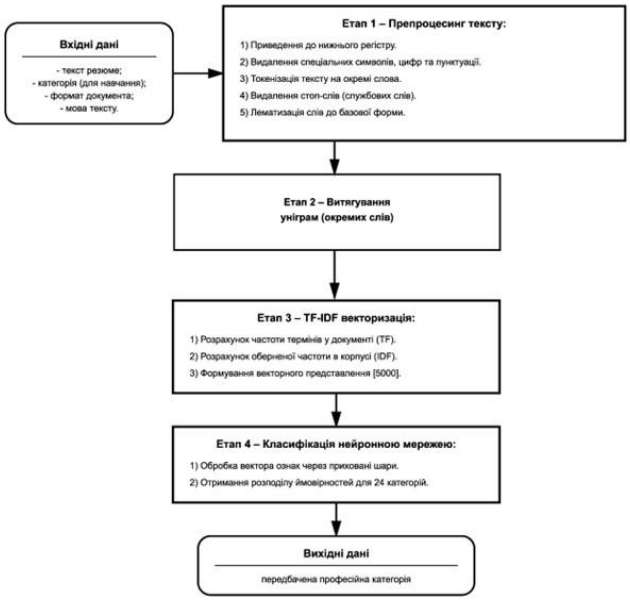
– провести аналіз існуючих методів та підходів до класифікації текстових документів з використанням методів машинного навчання та обробки мови;

– розробити метод класифікації резюме за професійними категоріями з використанням векторного представлення тексту TF-IDF та нейронних мереж;

– розробити програмну реалізацію методу класифікації резюме за професійними категоріями з використанням машинного навчання;

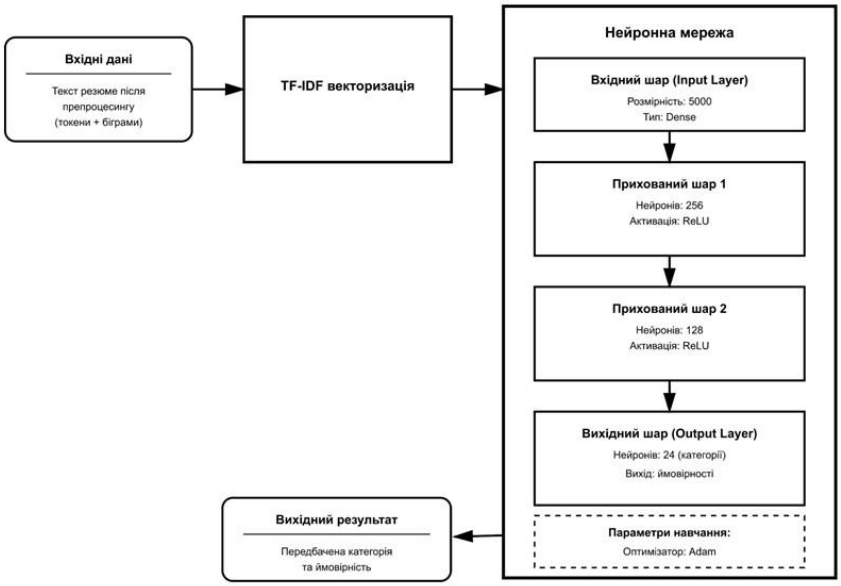
– провести експериментальне дослідження ефективності спроектованого методу шляхом порівняння базової та модифікованої моделей та оцінки їх точності класифікації на датасеті резюме.

Схема роботи методу класифікації резюме



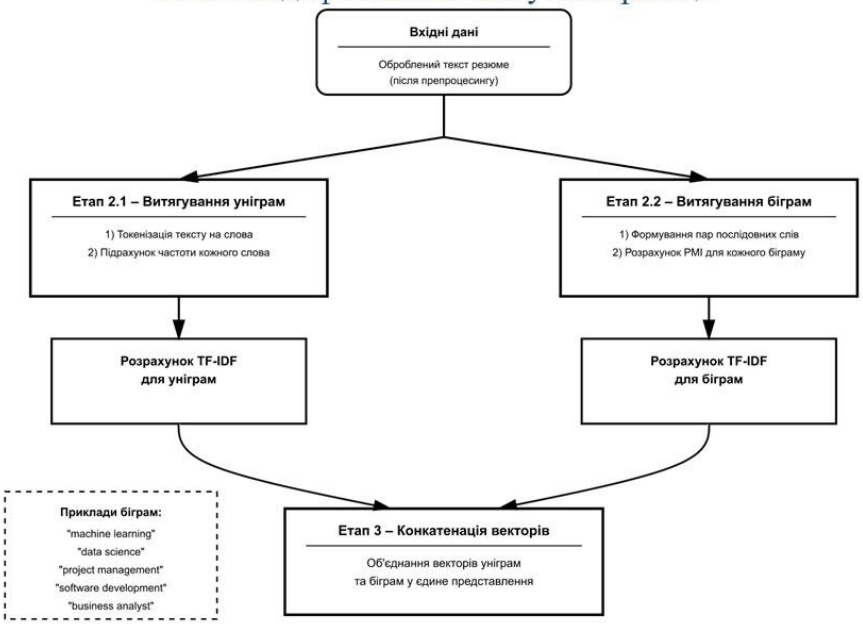
5

Архітектура моделі класифікації резюме



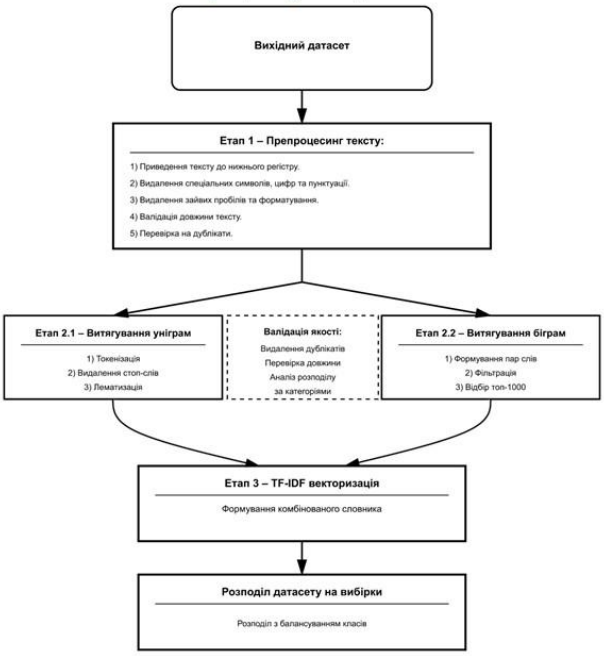
6

Схема модифікованого блоку векторизації

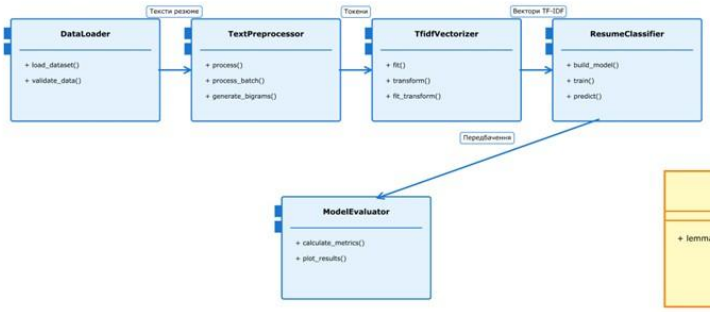


7

Процес обробки даних

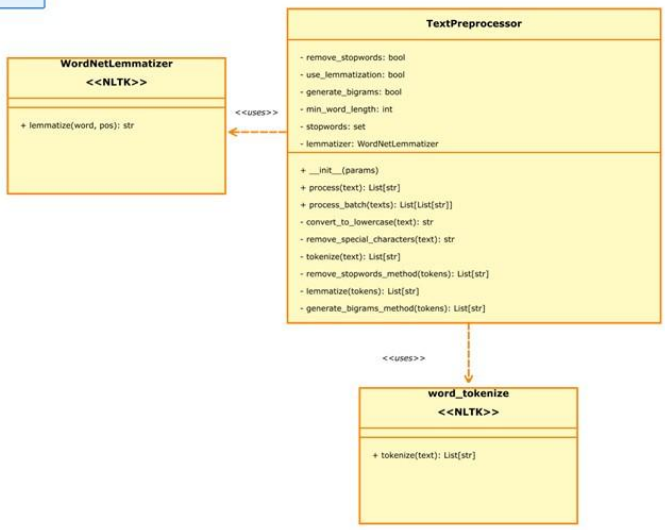


8

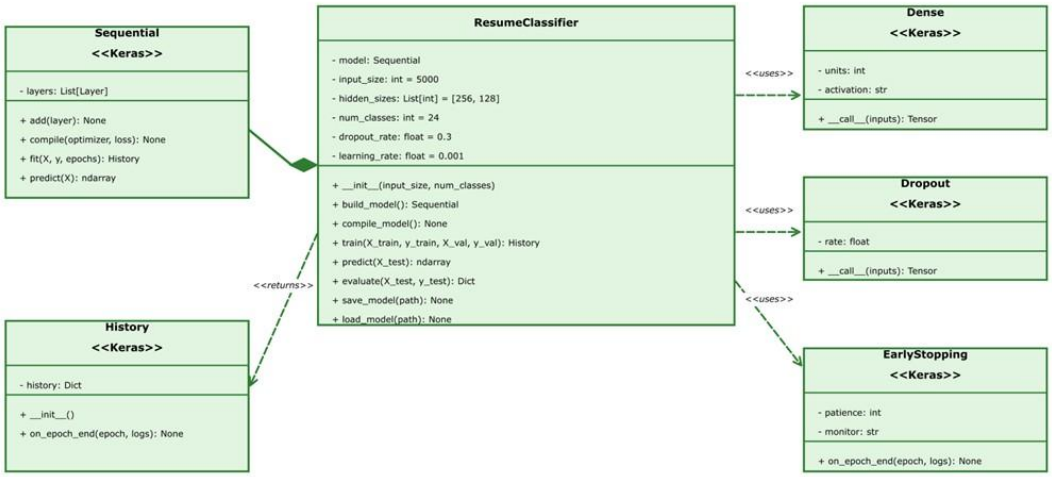


Діаграма компонентів системи

Діаграма класів модуля припроцесингу тексту

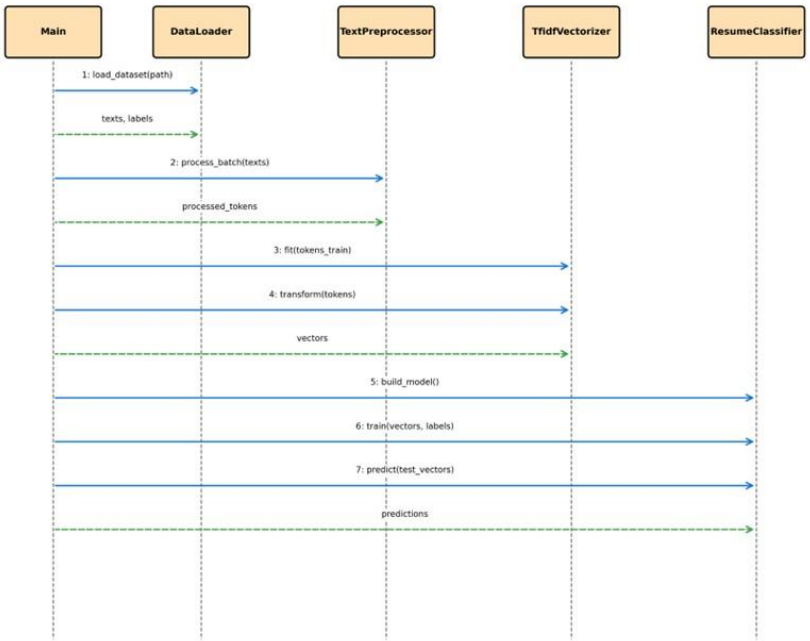


9



Діаграма класів модуля класифікації

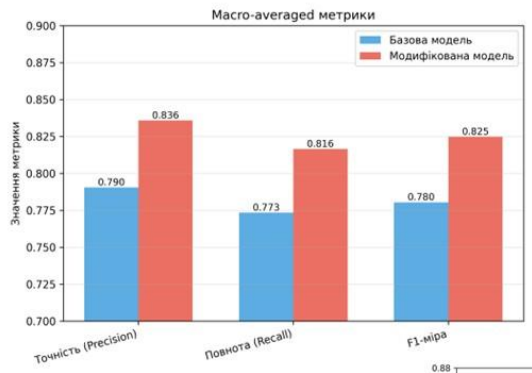
10

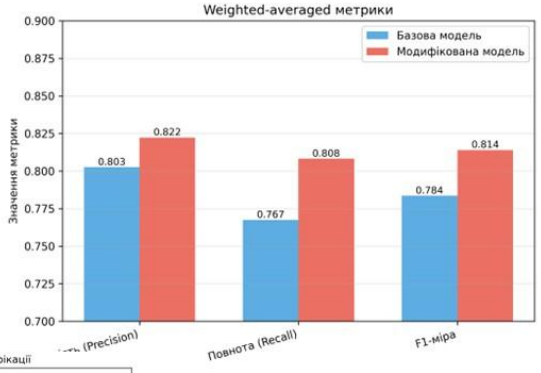


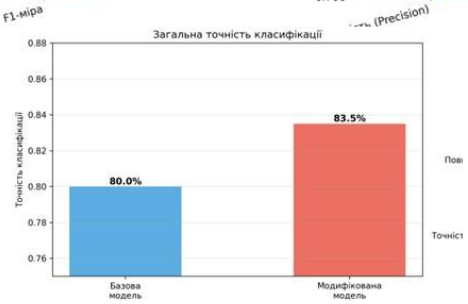
Діаграма послідовності процесу класифікації резюме

11

Метрики якості

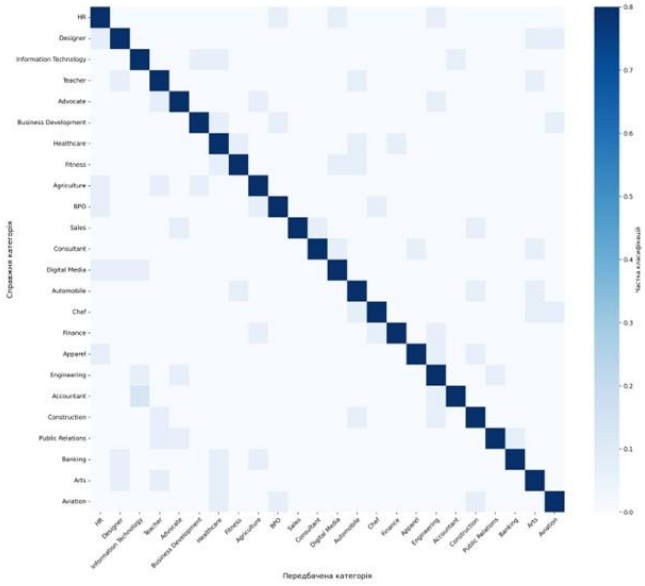




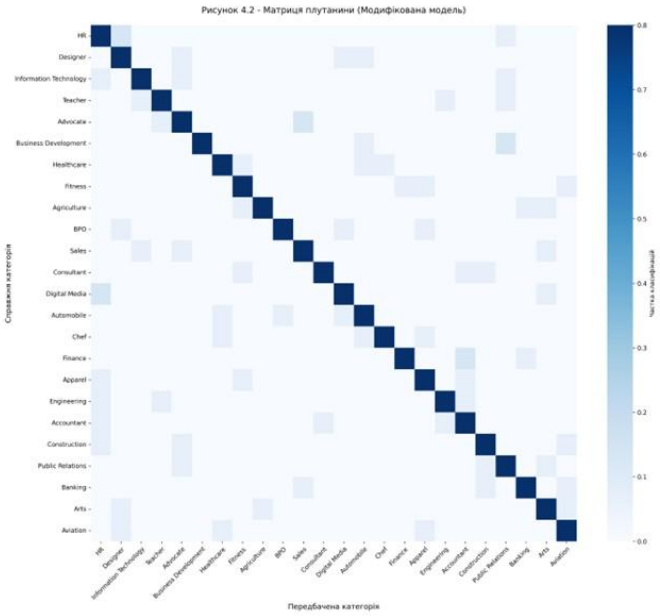


12

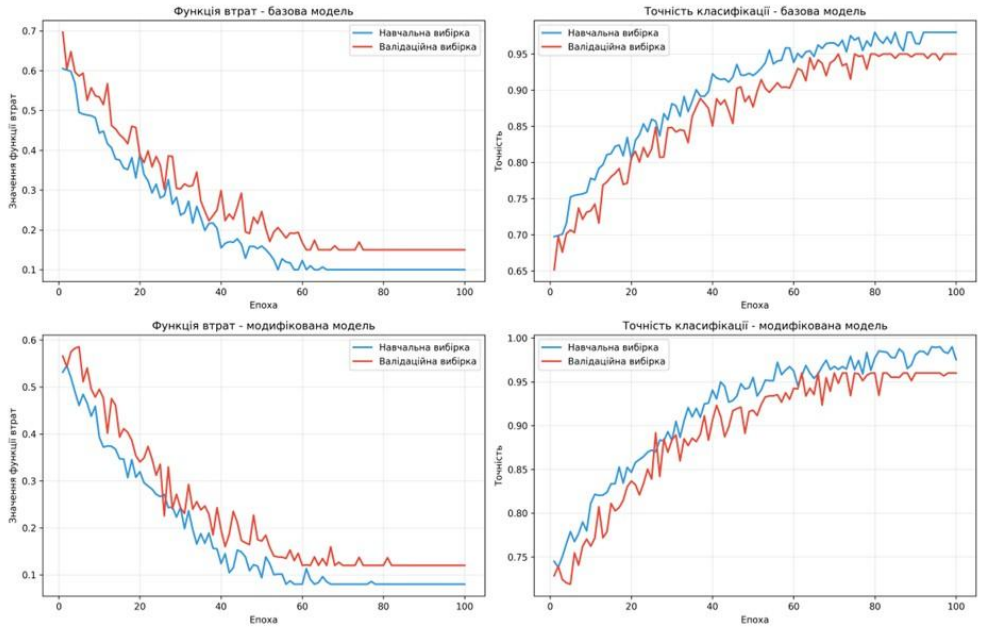
Матриця плутанини для базової моделі



Матриця плутанини для модифікованої моделі



Динаміка навчання моделей



Висновки

У кваліфікаційній роботі магістра вирішено актуальну науково-практичну задачу підвищення точності класифікації резюме за професійними категоріями шляхом розробки методу з використанням машинного навчання та векторного представлення тексту на основі уніграм та біграм.

Вирішено такі задачі:

– проведено аналіз існуючих методів та підходів до класифікації текстових документів з використанням методів машинного навчання та обробки мови;

– розроблено метод класифікації резюме за професійними категоріями з використанням векторного представлення тексту TF-IDF та нейронних мереж;

– розроблено програмну реалізацію методу класифікації резюме за професійними категоріями з використанням машинного навчання;

– проведено експериментальне дослідження ефективності спроектованого методу шляхом порівняння базової та модифікованої моделей та оцінки їх точності класифікації на датасеті резюме.

15

ДЯКУЮ ЗА УВАГУ!

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Нікіта КОРКУНДА

Співавтор:

Назва: КВАЛІФІКАЦІЙНА РОБОТА на тему Метод класифікації резюме за професійними категоріями з використанням машинного навчання

Науковий керівник: Едуард МАНЗЮК, д.т.н., професор

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1:3.4%

Коефіцієнт подібності 2:0.4%

Мікропробіли: 0

Заміна букв: 1

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-12-13 21:09:29.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-12-13

Дата

експерт Петровський С. С.

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи *Метод класифікації резюме за професійними категоріями з використанням машинного навчання*
Автор *студент групи КНм-24-1 Нікіта КОРКУНДА*
Освітня програма *Комп'ютерні науки*
Рівень вищої освіти *другий (магістерський)*
Спеціальність *122 – Комп'ютерні науки*
Науковий керівник: *д.т.н., проф. каф. комп'ютерних наук Едуард МАНЗЮК*

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмними засобами комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	***відповідає***
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	***відсутні***

Підтвердження:

Запозичення, виявлені в роботі Нікіти КОРКУНДА, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту.

Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості:
- за системою Anti-Plagiarism: 1%;
- за системою StrikePlagiarism КП1: 3,37%, КП2: 0,4%.

08.06.2025

Завідувач кафедри _____ *Олександр БАРМАК*

Гарант освітньої програми _____ *Руслан БАГРІЙ*

Керівник кваліфікаційної роботи _____ *Едуард МАНЗЮК*

Sat Dec 13 22:15:38 EET 2025, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 10%**

	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
ID: 252744 Title: КВАЛІФІКАЦІЙНА РОБОТА на тему Метод класифікації резюме за професійними категоріями з використанням машинного навчання Added in a DB: 2025-12-13 Authors: Нікіта КОРКУНДА Heads: Едуард МАНЗЮК Consultants: Opponents:	138531	2119	2205 (2%)	35 (2%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

студента *гр. КНм-24-1 Нікіти КОРКУНДА*

за темою *Метод класифікації резюме за професійними категоріями з використанням машинного навчання*

1. Актуальність обраної теми

Актуальність обраної теми зумовлена необхідністю автоматизації процесів відбору кадрів та обробки великих обсягів резюме в сучасних системах рекрутингу. Застосування методів машинного навчання та векторного представлення тексту для класифікації резюме дозволяє значно підвищити ефективність роботи HR-відділів, скоротити час на пошук відповідних кандидатів та зменшити витрати компаній на рекрутинг.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Магістерська робота повністю відповідає предметній області спеціальності 122 Комп'ютерні науки, оскільки ґрунтується на застосуванні методів машинного навчання, обробки природної мови та нейронних мереж для вирішення задачі класифікації текстових документів. Дослідження спирається на використання фундаментальних знань у галузі комп'ютерних наук, таких як алгоритми, структури даних, програмування та аналіз даних. Робота також відповідає загальним вимогам до наукових робіт, маючи чітку структуру, обґрунтовану актуальність, визначену мету та завдання, ґрунтовний аналіз існуючих підходів, детальний опис запропонованого методу та результати експериментальних досліджень, що підтверджують його ефективність.

3. Повнота розкриття мети та завдань дослідження

Мета та завдання дослідження розкриті повністю. Автор чітко формулює мету роботи – підвищення точності класифікації резюме за професійними категоріями шляхом розробки методу з використанням векторного представлення тексту та нейронних мереж. Для досягнення мети послідовно вирішуються поставлені завдання, що включають аналіз існуючих підходів, розробку методу з використанням TF-IDF векторизації та нейронних мереж, реалізацію попередньої обробки даних, модифікацію базового методу шляхом включення біграм та експериментальну перевірку ефективності.

4. Наявність наукової новизни

Наукова новизна роботи полягає в удосконаленні методу класифікації резюме за професійними категоріями, який відрізняється від існуючих використанням комбінованого векторного представлення на основі уніграм та біграм з оптимізованим співвідношенням.

5. Зміст кожного розділу роботи

Робота містить чотири розділи. В першому розділі представлено аналіз існуючих методів класифікації текстових документів та систем автоматизованої обробки резюме. Другий розділ містить розробку методу класифікації резюме за професійними категоріями з використанням векторного представлення тексту TF-IDF та нейронних мереж. Третій розділ присвячено програмній реалізації розробленого методу та його модифікації з включенням біграм. Розділ чотири містить експериментальне дослідження ефективності методу.

6. Ступінь розкриття теми роботи

Тема роботи розкрита повністю. Автор аналізує проблематику класифікації резюме, розглядає існуючі методи векторизації тексту та машинного навчання, обґрунтовує необхідність розробки удосконаленого методу. Детально описано запропонований метод на основі TF-IDF векторизації та нейронних мереж прямого поширення, наведено архітектуру моделі з двома прихованими шарами. Експериментальні дослідження на реальному датасеті підтверджують ефективність методу.

7. Якість оформлення кваліфікаційної роботи

Якість оформлення кваліфікаційної роботи відповідає встановленим академічним стандартам, демонструючи чіткість, послідовність та професійність у структурі, форматуванні та презентації матеріалу. Робота містить необхідні ілюстрації, таблиці та діаграми, що допомагають у розумінні запропонованого методу.

8. Недоліки кваліфікаційної роботи

Бажано включити більш глибокий аналіз помилок класифікації між схожими професійними категоріями.

9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка *відмінно*.

Опонент д.т.н. проф. каф. АКІТ Мартинюк В.В.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

студента *гр. КНм-24-1 Нікіти КОРКУНДА*

за темою *Метод класифікації резюме за професійними категоріями з використанням машинного навчання*

1. Актуальність теми

Актуальність теми зумовлена критичною необхідністю автоматизації процесів відбору кадрів та обробки великих обсягів резюме в сучасних системах рекрутингу. Можливість автоматично класифікувати резюме за професійними категоріями за допомогою методів машинного навчання відіграє ключову роль у підвищенні ефективності роботи HR-відділів, оптимізації логістики підбору персоналу та зменшенні витрат компаній на рекрутинг. Дослідження методів класифікації резюме з використанням векторного представлення тексту та нейронних мереж має практичну цінність для широкого спектру галузей, пов'язаних з управлінням персоналом, що робить цю тему актуальною та важливою.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Робота відповідає предметній області спеціальності 122 "Комп'ютерні науки", оскільки вона ґрунтується на застосуванні методів машинного навчання, зокрема нейронних мереж прямого поширення, та методів обробки природної мови. Дослідження передбачає використання алгоритмів векторизації тексту, структур даних, аналізу та попередньої обробки даних та має потенційне практичне значення в різних галузях, демонструючи міждисциплінарний характер цієї спеціальності.

3. Професійні та особистісні якості

Під час роботи над магістерським дослідженням Нікіта КОРКУНДА продемонстрував високий рівень професійної компетентності в галузі комп'ютерних наук, відповідально та ефективно вирішуючи завдання з розробки методу класифікації резюме за професійними категоріями. Студент проявив наполегливість у проведенні експериментальних досліджень та глибоке розуміння принципів машинного навчання.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

При виконанні магістерської роботи студент виявив високий рівень самостійності, запропонувавши удосконалення базового методу класифікації резюме шляхом включення біграм до векторного представлення тексту, що дало змогу покращити точність класифікації порівняно з базовим підходом на основі лише уніграм.

Студент самостійно провів аналіз літератури, розробив архітектуру нейронної мережі, реалізував програмне забезпечення та провів експериментальні дослідження.

5. Наукова новизна та оригінальність запропонованих підходів

Удосконалено метод класифікації резюме за професійними категоріями, який відрізняється від існуючих використанням комбінованого векторного представлення на основі уніграм та біграм з оптимізованим співвідношенням, що включає модифіковану архітектуру векторизації та нейронну мережу прямого поширення з двома прихованими шарами, що дозволило покращити точність класифікації на 3.5%.

6. Ступінь оволодіння методами дослідження

Під час роботи над магістерським дослідженням студент продемонстрував глибоке розуміння та вміле застосування методів комп'ютерних наук, зокрема методів обробки природної мови, векторизації тексту TF-IDF, нейронних мереж та статистичного аналізу результатів, для вирішення задачі класифікації текстових документів, що свідчить про його високий рівень оволодіння сучасними методами дослідження в галузі машинного навчання та обробки текстів.

7. Повнота та якість розкриття теми роботи

У магістерській роботі тема розкрита повністю та ґрунтовно. Робота відзначається логічною структурою, глибиною аналізу існуючих підходів до класифікації текстових документів, детальним описом розробленого методу, його програмною реалізацією та всебічним експериментальним дослідженням.

8. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Магістерська робота характеризується чіткою логічною структурою, послідовним викладенням матеріалу від аналізу проблематики та існуючих підходів до розробки та експериментальної перевірки власного методу. Автор демонструє високий рівень літературної грамотності, дотримуючись наукового стилю викладення та забезпечуючи легкість сприйняття тексту.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Метод може бути інтегрований у сучасні системи та платформи для пошуку роботи, що дозволить значно підвищити ефективність роботи HR-відділів компаній різних галузей.

10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

Враховуючи належний рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник _____ *д.т.н., професор каф. КН Едуард МАНЗЮК*