Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод нечіткої кластеризації коротких текстів
(тема)

Виконав:

студент II курсу, групи СПм-22-5

Стеблянко Б. О.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

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

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: проф. Кучук Г.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

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

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Стеблянку Богдану Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Метод нечіткої кластеризації коротких текстів

затверджена наказом по університету від " 01 " квітня 2024 р. № 257 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2024 р.

3. Вхідні дані до роботи операційна система – Windows, мова програмування – C#, Excel, класичне завдання класифікації та кластеризації "Іриси Фішера".

4. Перелік питань, що потрібно опрацювати у роботі

1) огляд сучасних методів чіткої та нечіткої кластеризації;
2) формулювання завдання мовного моделювання для коротких текстів;
3) подати опис роботи програми, її основні функції;
4) провести верифікацію методу на синтетичному наборі даних;
5) зробити висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз літературних джерел-	02.04.24-08.04.24	
2	Вибір та обгрунтування методики дослідження	09.04.24-16.04.24	
3	Вибір інструментальних засобів	17.04.24-22.04.24	
4	Розробка моделей протоколів	23.04.24-06.05.24	
5	Проведення експериментів	07.05.24-23.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	24.05.24-03.06.24	
7	Подання кваліфікаційної роботи керівникові та	04.06.24-07.06.24	
	її попередній захист		
8	Подання кваліфікаційної роботи на рецензу-	08.06.24-12.06.24	
	вання		

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Кучук Г.А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 58 с., 16 рис., 8 табл., 2 дод., 15 джерел.

КЛАСТЕРИЗАЦІЯ, КОРОТКІ ТЕКСТИ, МОДЕЛІ, МАШИННЕ НАВЧАННЯ.

Метою роботи є розробка методу системи підтримки прийняття рішень для кластеризації коротких текстів українською мовою з урахуванням експертної інформації. Ефективність визначається точністю кластеризації та скороченням часу та трудомісткості роботи виконуваної експертом при використанні запропонованого рішення.

Об'єктом дослідження є кластеризація наборів даних, що складаються з коротких текстів українською мовою та експертна інформація, що надходить під час інтерактивної обробки текстів.

Предметом дослідження є методи нечіткої кластеризації коротких текстів та обробки експертної інформації.

ABSTRACT

Master’s thesis: 58 pages, 16 figures, 8 tables, 2 appendices, 15 sources.

CLUSTERIZATION, SHORT TEXTS, MODELS, MACHINE LEARNING.

The purpose of the work is to develop a model in the decision support system for clustering short texts in the Ukrainian language taking into account expert information. Efficiency is determined by the accuracy of clustering and the reduction of the time and labor intensity of the work performed by the expert when using the proposed solution.

The object of the study is the clustering of data sets consisting of short texts in the Ukrainian language and expert information received during interactive text processing.

The subject of the research is methods of fuzzy clustering of short texts and processing of expert information.

.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API –Application Programming Interface

BERT – Bidirectional Encod-er Representations from Transformers

CNN – Convolutional Neural Network

CPU – Central Processing Unit

DEC – Unsupervised Deep Embedding for Clustering Analysis

ELMo – Embeddings від Language Models

GPU –Graphics Processing Unit

IDE –Integrated Development Environment

LDA – Latent Dirichlet allocation

LSTM –Long short-term memory

NER – Named Entity Recognition

NLP –Natural Language Processing

NLTK – один із провідних пакетів програмного забезпечення з обробки природної мови

REST – Representational State Transfer

RNN – Recurrent Neural Network

ВСТУП

Актуальність. Кластерний аналіз є одним з найважливіших розділів системного аналізу даних і застосовується в різних проблемних областях - технічних, природничих, соціальних. Кластеризація є прикладом завдання навчання без вчителя і зводиться до розбиття вихідної множини об'єктів на підмножини класів таким чином, щоб елементи одного класу були максимально схожі між собою, а елементи різних класів відрізнялися.

Традиційні методи кластерного аналізу працюють із об'єктами, заданими у вигляді векторів ознак. При роботі з текстами першим кроком алгоритму кластеризації є визначення простору ознак і побудова в ньому векторів наявних текстів. Як правило, одержувані вектори мають велику розмірність і при роботі з ними традиційні методи кластерного аналізу не забезпечують достатню ефективність. У разі роботи з короткими текстами розмірність векторів не зменшується, а лише додається властивість розрідженості до векторів ознак, що створює додаткові труднощі при їх обробці методами кластерного аналізу. Під короткими текстами в даному дослідженні маються на увазі тексти, що складаються з одного або декількох речень із загальним числом слів в діапазоні від 5 до 100. Крім того, додатковими факторами, що ускладнюють вирішення завдання кластеризації для коротких текстів, є такі: синонімія, омонімія, більше часте, порівняно із звичайними текстами, використання абревіатур, сленгових виразів і неологізмів і найголовніше – часткова чи повна відсутність контексту в коротких текстах.

Висока розмірність одержуваних просторів ознак у разі роботи з текстами об'єктивна, так як тексти – це складні багатовимірні і багатопланові структури, що потенційно містять різний сенс, емоційні відтінки, авторські характерні риси, стиль викладу і багато іншого . При великому розмаїтті можливих показників підхід чіткої кластеризації , у якому кожному об'єкту

зіставляється лише один кластер, є досить ефективним. Експерту, який проводить процедуру кластеризації, в ході аналізу результатів важливо знати і розуміти альтернативні варіанти співвіднесення об'єкта з кластером. Тому, у разі роботи з текстами, найбільш переважними є методи нечіткої класифікації.

Кластеризація текстів допускає значну кількість можливих принципів для розбиття на класи: тематика, автор, стиль, емоційне забарвлення, правовий статус і комбінація різних факторів. Методи, що не дозволяють врахувати інтенцію експерта, виявляються в загальному випадку не ефективними для вирішення описаного завдання. Альтернативним є підхід, у якому експерт входить у процес кластеризації і на різних її етапах задає обмеження з урахуванням проміжних результатів, які враховуються у подальших стадіях кластеризації. Такі методи класифікуються як методи інтерактивної кластеризації з використанням зворотного зв'язку від експерта. Інтерактивні методи забезпечують скорочення сумарних витрат часу експерта на обробку результатів кластеризації і дозволяють підвищити точність кластеризації за рахунок виявлення прихованого знання експерта на ранніх етапах кластеризації. Врахування додаткової інформації дозволяє алгоритму вибрати правильний напрямок ходу процесу розбиття на кластери.

Масиви інформації, що складаються з наборів коротких текстових фрагментів, сприяють інтенсифікації досліджень у розвитку методів обробки текстів із застосуванням машинного навчання. Цій проблемі щороку присвячується значна кількість досліджень. Більшість з проведених досліджень відноситься до текстів англійською мовою. Досліджень у сфері української мови значно менше, що пояснюється як меншим числом дослідників, що займаються питаннями української мови, та й об'єктивно більшою складністю української мови для автоматизованої обробки. Недостатня розробленість стандартних засобів кластеризації для коротких текстів та низька ефективність існуючих методів на текстах українською мовою підтверджується відсутністю стандартних засобів кластеризації для

коротких текстів у провідних пакетах NLP (Natural Language Processing, наприклад, NLTK).

Об'єктом дослідження є процес кластеризації наборів даних, що складаються з коротких текстів українською мовою та експертна інформація, що надходить під час інтерактивної обробки текстів.

Предметом дослідження є методи нечіткої кластеризації коротких текстів та обробки експертної інформації.

Метою роботи є розробка методу у системі підтримки прийняття рішень для кластеризації коротких текстів українською мовою з урахуванням експертної інформації. Ефективність визначається точністю кластеризації та скороченням часу і трудомісткості роботи, виконуваної експертом, при використанні запропонованого рішення.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести огляд сучасних методів чіткої та нечіткої кластеризації;
- розробити метод кластеризації коротких текстів;
- провести реалізацію та верифікацію методу нечіткої кластеризації коротких текстів.

Методи дослідження. При виконанні дипломної роботи було використано широкий спектр методів, таких як методи теорії масового обслуговування, теорії ймовірності, математичної статистики та комп'ютерного імітаційного моделювання.

Галузь застосування – інфокомунікаційні системи.

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

З стрімким розвитком технологій web 2.0, все більше коротких текстів генерується різними видами веб-сайтів. Facebook (пости та статуси з обмеженням у 142 символи, Twitter – з обмеженням у 140 символів, Windows Love Messenger з обмеженням у 128 символів та Instagramm – лише кілька прикладів таких веб-сайтів. Обсяги повідомлень на перерахованих web - ресурсах обчислюються мільйонами, при цьому регулярно виникають нові тематики, змінюються тренди, змінюються зміст слів і цілих фраз. Також традиційно до наборів даних коротких текстів відносяться набори заголовків новин . в ньому налічується близько 500 000 оголошень по 4-м великим категоріям і ряду підкатегорій. проведених змагань із класифікації оголошень.

1.1 Огляд сучасних методів чіткої та нечіткої кластеризації

Методи кластеризації належать до групи методів машинного навчання без учителя. "Чітка" кластеризація полягає в розбитті досліджуваного набору даних o = {o_1, o_2, o_3, ...} на групи класів з = {c_1, c_2, ...} – таким чином, щоб елементи одного класу істотно відрізнялися один від одного за заданим набором параметрів p = {p_1, p_2, p_3 , …. } – від елементів інших класів, і були максимально схожі з елементами свого класу [4,5].

Нечітка кластеризація (також звана м'якою кластеризацією) – це форма кластеризації, в якій кожна точка даних може належати більш ніж одному кластеру з певною мірою належності [2].

Класичні методи кластеризації успішно застосовуються практично [8] і показують високі результати. Крім класичного методу k- mean [10] і більш просунутих, таких як hdbscan [2,7], існують методи на основі роєвого та

генетичного алгоритмів [6] оптимізації, на основі методу головних компонентів [9], кластеризації графів та інших математичних моделей [8].

Методи та метрики, що застосовуються для оцінки ефективності алгоритмів кластеризації [5] дозволяють порівнювати роботу методів однієї та різної природи. На еталонних наборах даних, що використовуються для перевірки методів кластеризації, класичні методи показують 60-70 і більше відсотків точності [9]. Тим не менш, існує велика кількість сучасних методів демонструють набагато кращі результати (state-of-the-art results). Більшість цих методів ґрунтуються на використанні мереж з глибинним навчанням (deep neural network) [7]. Така перевага пояснюється здатністю мереж навчатися на суміжних предметних областях або подібних завданнях (transfer learning , learning to cluster) і будувати складні нелінійні перетворення для отримання простору ознак (representation learning , embedding learning) одночасно містить максимум інформації та "зручного" для алгоритму кластеризації (наприклад, сильне зниження розмірності вхідних даних) [11]. Але найголовнішим внеском використання нейронних мереж методи кластеризації є можливість побудови безперервної кластеризації (end-to-end clustering), в якій відсутня явне поділ алгоритму на дві фази: побудова простору ознак та розбиття на групи [4]. При такому підході навчання мережі відповідного подання даних відбувається одночасно з ітераціями розбиття множини на кластери або побудови ієрархії з них [10]. У ряді методів автори показують можливість подальшого перенесення отриманих знань мережі на суміжні завдання, наприклад використання навченої мережі для кластеризації одного виду зображень на інший вид зображень.

1.2 Аналіз сучасних моделей та методів обробки природної мови

Тексти, будучи багатовимірними об'єктами, становлять особливу складність алгоритмів кластеризації, тому їм у більшості підходів формуються простору ознак великої розмірності, із якими справляються

традиційні методи кластеризації. Наприклад, найбільш простими та поширеними способами обробки тексту природною мовою є методи, засновані на підході “мішка слів” [7,9].

Цей підхід полягає в тому, що всі слова, які використовуються в досліджуваному корпусі текстів, спочатку вважаються рівнозначними та незалежними. Це дозволяє перейти з роботи з природною мовою до роботи з векторним простором розмірності N, де |N| = числу різних слів у корпусі, слова упорядковуються в рамках словника корпусу текстів, таким чином, кожне слово можна однозначно ідентифікувати його номером у словнику. Кожному слову в такому просторі зіставляється кодуючий вектор (one-hot вектор) в якому всі компоненти дорівнюють 0, за винятком компоненти з номером, що відповідає номеру слова в словнику, ця компонента належить рівною 1. Іноді замість слів використовують токени (довільні частини слів, залежно від від алгоритму отримання токенів) [11] або леми (вихідні форми слова), що дозволяє знизити розмірність об'єкта, що досліджується. Тим не менш, навіть кількість лем у корпусах текстів обчислюється тисячами та десятками тисяч. Очевидно, що слова в тексті і загалом у природній мові не є незалежними, вони пов'язані синтаксично та семантично. Облік цього зв'язку дозволяє точніше моделювати текст, використовувати моделі менших розмірів та отримувати якісніші результати. Так було в 1998 року було представлено проект вирішальний завдання присвоєння семантичних ролей [9]. Ця форма поверхневого семантико-синтаксичного аналізу досі активно використовується та досліджується.

У 2001 році було представлено модель умовних випадкових полів [5]. Цей клас методів розмітки послідовностей “отримав винагороду test-of-time (випробування часом) на міжнародній конференції з машинного навчання (ICML) 2011. іменованих сутностей” [7]. Широко відомий метод латентного розміщення Діріхле [7] вперше опубліковано у 2003 році.

LDA – один з методів, що найбільш широко використовуються в машинному навчанні. У класифікації та кластеризації LDA є стандартним

способом тематичного моделювання. Разом з розвитком методів обробки штучної мови розвивалися і корпуси текстів. Наприклад, проект OntoNotes - великий багатомовний корпус із множинними інструкціями був представлений у 2006 році [5].

Корпус OntoNotes використовувався для навчання безлічі завдань, серед яких: синтаксичний аналіз на основі граматики залежностей та вирішення кореференції.

У 2008 Мілн і Віттен показали, як Wikipedia (онлайн енциклопедія Вікіпедія) може використовуватися для збагачення наборів даних для методів машинного навчання. З того часу Вікіпедія є одним із головних ресурсів для навчання моделей для обробки природної мови. У колекції збираються не лише тексти, а й результати їхньої обробки. Наприклад, у 2016 році у проекті Universal Dependencies [8] було зібрано багатомовні синтаксичні дерева.

До січня 2019 року Universal Dependencies налічував понад 100 синтаксичних дерев більш ніж 70 мовами. Таким чином, обсяг сучасних корпусів текстів та обчислювальні потужності сприяють тому, що сучасні методи обробки природної мови переходять від побудови приватних моделей для вирішення локальних завдань до побудови узагальнених мовних моделей для вирішення групи завдань для корпусу текстів або цілком природної мови. Ці підходи відображають загальну тенденцію переходу до багатозадачного навчання та перенесення знань, що особливо широко використовується в нейронних мережах.

1.3 Аналіз сучасних методів інтерактивної кластеризації

Навчання без вчителя можливе завдяки інформації, що міститься в самих даних, яку покликані виявити методи кластеризації [5]. Тим не менш, на практиці дослідник рідко не має жодних знань про досліджуваний набір даних [5], будь то економічні дані, дані зібрані з датчиків, приладів або якимось іншим чином комп'ютерною програмою. У більшості випадків

вирішення практичних завдань участь дослідника необхідна або для побудови коректного розбиття на групи, або прийняття рішення про структуру ієрархії [8], або сприяє суттєвому підвищенню якості результату за рахунок знань, не включених до 20-го простору ознак оброблюваних даних [4]. Особливо це актуально під час обробки текстової інформації. Тексти, будучи багатовимірними об'єктами, становлять особливу складність алгоритмів кластеризації [4]. Без участі експерта, без виявлення його прихованих інтенцій неможливо заздалегідь визначити, яке саме розбиття очікується в результаті алгоритму роботи [5]. Крім очевидного угруповання за тематикою, тексти можуть бути згруповані на підставу того від чиєї особи ведеться оповідання, за цільовою аудиторією тексту, за правовим статусом тексту чи комбінації різних ознак.

Таким чином, для отримання якісного результату роботи алгоритму кластеризації потрібно включення експерта до кластеризації як органічної частини алгоритму кластеризації. При цьому, бажано, щоб це не вимагало розуміння внутрішніх деталей роботи алгоритму від експерта, і причинно-наслідковий зв'язок між діями експерта та результатами роботи алгоритму був би явним [3]. У сучасній науковій літературі склалася практика позначення методів кластеризації, в яких використовується та чи інша додаткова інформація, не включена до набору даних, методами кластеризації з частковим залученням вчителя (semi-supervised) кластеризацією з обмеженнями (constrained clustering) [3]. При цьому у переважній більшості таких методів інформація дана a priori і подається на вхід алгоритму кластеризації спільно з набором даних у вигляді частково промаркованих об'єктів [4], заданих обмежень на пари об'єктів [3], обмеження на структуру ієрархії кластерів, перенесення знання у вигляді попередньої нейронної мережі (transfer learning) [6], наприклад, завдання класифікації у схожій предметної області тощо. При цьому і обмеження на об'єкти та мітки можуть бути задані не жорстко (soft labels) [8]. Однак, існують методи , що передбачають отримання додаткової інформації безпосередньо в процесі

кластеризації, їх докладний огляд зроблено в роботі [6]. Такі методи називають методами інтерактивної кластеризації. Одним із перших таких методів став нечіткий метод [7].

Залежно від характеру взаємодії та отриманої інформації вони поділяються на:

- активну кластеризацію як приклад активного навчання [3];
- кластеризація з підкріпленням, одержуваної у вигляді зворотного зв'язку від середовища в якому відбувається кластеризація [7];
- кластеризація із зворотним зв'язком (interactive clustering under feedback, mixed-initiative clustering), що передбачає отримання зворотного зв'язку від користувача у вигляді оцінки результатів або вказівок щодо коригування алгоритму.

Останні методи дозволяють виявити приховані інтенції користувача і отримати по-справжньому корисну кластеризацію, т.к. добре відповідають тезі: "користувач дізнається правильний результат, коли побачить його" [2,5].

Дослідники зазначають, що до інтерактивних методів часто помилково відносять і методи кластеризації з інтерактивними операціями: методи інтерактивної візуалізації результатів кластеризації, методи вибору алгоритмів кластеризації тощо. [6]. Для повноти картини слід згадати методи допоміжної кластеризації. clustering) [1], в яких провідна роль віддана досліднику, саме він визначає кількість кластерів та їх характеристики, а алгоритм пропонує варіанти їх наповнення та коригування структури. Однак цим методи на даний момент не набули значного поширення.

Методи інтерактивної кластеризації зі зворотним зв'язком можна розділити на дві множини тому, що спрямована зворотний зв'язок від дослідника. У першому численному сімействі методів дослідник інтерактивно і ітеративно може проводити параметри алгоритму кластеризації, метрику схожості (близькості), модифікувати простір ознак [7]. У другій множині методів дослідник взаємодіє безпосередньо з результатами кластеризації, вказуючи які кластери необхідно об'єднати або

роз'єднати, які елементи додати або виключити з кластера, яким чином утворити новий кластер або куди віднести елементи, що випадають із кластеризації [6, 9]. Такий підхід відноситься саме до другої множини, що дозволяє досліднику не занурюватися в деталі реалізації алгоритму і використовувати нові методи, що не з'являються, не змінюючи характер своєї роботи. Систематизація методів кластеризації за участю дослідника, які належать великому сімейству методів кластеризації із залученням вчителя (semi-supervised clustering), може бути представлена наступним чином:

- кластеризація з обмеженнями (constrained clustering);
- інтерактивна кластеризація (interactive clustering);
- активна кластеризація (active clustering);
- кластеризація з підкріпленням (reinforcement clustering);
- інтерактивна кластеризація із зворотним зв'язком від користувача (interactive clustering with user feedback);
- зворотній зв'язок як коригування параметрів або виду цільової функції;
- зворотній зв'язок у вигляді оцінки результатів кластеризації;
- допоміжна кластеризація (assisting clustering).

Першим етапом інтерактивної кластеризації, очевидно, є нормальна кластеризація без вчителя. Таким чином, всі методи інтерактивної кластеризації базуються на методах без вчителя, додаючи до них механізми роботи із зворотним зв'язком.

На рисунку 1.1 представлено динаміку кількості публікацій присвячених темі інтерактивної кластеризації згідно з дослідженням [6].

Дане дослідження дозволяє помітити, що більшість методів інтерактивної кластеризації ґрунтуються на класичних методах кластеризації, таких як:

к – means,

с – means, варіаціях ієрархічної кластеризації та кластеризації графів.

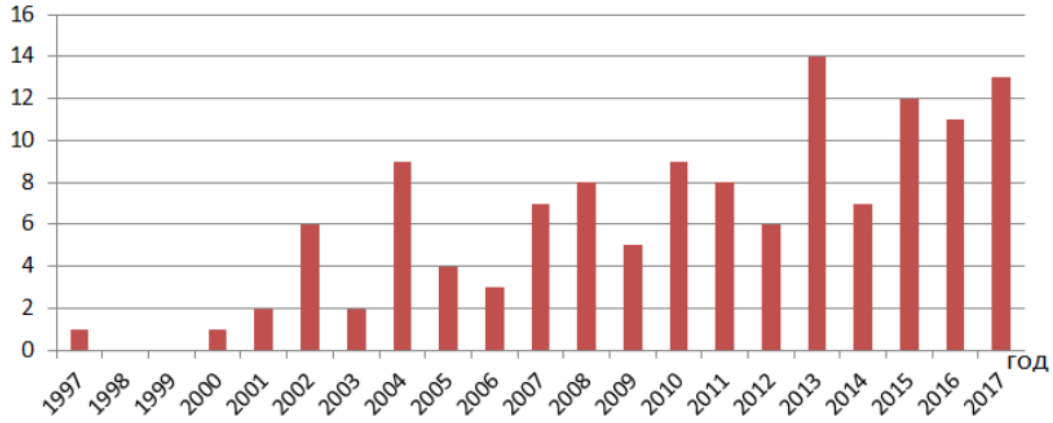


Рисунок 1.1 – Динаміка кількості публікацій, присвячених темі інтерактивної кластеризації

Невелика кількість методів використовує нейронні мережі, а в разі їх використання застосовуються штучні нейронні мережі SOM, що самоорганізуються (Kohonen self-organized maps).

1.4 Аналіз особливостей обробки коротких текстів

У таблиці 1.1 представлені основні завдання актуальні у сфері обробки коротких текстів.

Таблиця 1.1 – Основні завдання актуальні у сфері обробки коротких текстів

Завдання	Опис	Методи вирішення
Розпізнавання іменованих сутностей	Виділення з тексту дат, прізвищ, найменувань географічних об'єктів тощо.	Морфемний аналіз, Мовні моделі
Визначення тональності тексту	Поділ набору даних на тексти з позитивною та негативною тональністю, або за ширшим спектром емоцій	Визначення за ключовими словами
Класифікатори на базі мовних моделей	Класифікація текстів по заздалегідь визначеним тематкам	LDA, Мовні моделі

Ці завдання збігаються з основними завданнями у сфері обробки природної мови (NLP – Natural Language Processing) і вирішуються подібними способами. При цьому обробка коротких текстів відрізняється додатковою складністю. Нижче наведено та описано основні проблеми, що виникають при обробці коротких текстів. Розрідженість векторів ознак. При векторизації документів або великих текстів кожному документу зіставляється вектор ознак. Елементи цього вектора відповідають термінам корпусу текстів або іншим характерним ознаками, які підходять для використання в алгоритмах машинного навчання. При цьому саме числове значення елемента вектора є вагою відповідної ознаки. Ваги можуть бути розраховані у різний спосіб. Найбільш популярною є метрика t_f-id_f – добуток частоти f(t,d) – кількість вживань терміну 't' у документі 'd' та зворотної частоти id_f (t,D) – відношення загальної кількості документів 'D' до кількості документів, що містять термін 't'. Математично це виглядає наступним чином: $t_f\_id_f$ (t, d , D) = t_f (t, d) * id_f (t, D). У коротких текстах, кількість слів вкрай мало, і вектор ознак побудований таким чином буде сильно розрідженим. При цьому відомі алгоритми класифікації та кластеризації (k-means, HDBScan, тематичний аналіз [11] тощо) показують низьку ефективність при роботі з такими векторами.

Полісемія. Наявність більш ніж одного лексичного значення у слова (наприклад, 'коса' і менш очевидна 'павутина'). Таким чином, визначення категорії для цього слова вимагає аналізу контексту. У великому тексті завжди присутній необхідний контекст на відміну від коротких текстів, в яких може бути всього кілька слів і контекст певного слова може бути не розкритий або, має намір, залишений неясним (гра слів, алегоричні висловлювання).

Синонімія. Два і більше слів, які мають однакове або близьке лексичне значення. Наприклад: 'красивий', 'привабливий', 'симпатичний'. У обробці природної мови синоніми вимагають наявності словників синонімів. У коротких текстах виникає додаткова труднощі - синоніми ще сильніше

збільшують розрідженість векторів ознак, при цьому вони не можуть бути об'єднані в одну ознаку, так як при цьому може загубитися значення відтінків, яке могло бути важливо в даному тексті. Використання абревіатур, сленгових слів та неологізмів. Часто системи, в яких користувачі створюють короткі тексти, такі як Twitter, обмежують загальну довжину повідомлення, що мотивує користувачів на використання скорочень та абревіатур. Додаткова мотивація скорочувати слова у користувача виникає внаслідок внесення тексту через мобільні пристрої з незручною для введення довгих слів клавіатурою.

Проблема друкарських помилок, граматичні та пунктуаційні помилки. На відміну від великих текстів у різних виданнях, короткі тексти з соціальних мереж та інших систем, що передбачають генерацію текстів користувачами, не передбачають спеціальної фази рецензування та редактури, в результаті це призводить до того, що рівень грамотності формулювання таких текстів вкрай низький. Більшість об'єктів у наборах коротких текстів вводяться з мобільних пристроїв з незручною для введення клавіатурою, що призводить до великої кількості друкарських помилок. Інтелектуальні помічники можуть навіть погіршувати ситуацію, замінюючи слово з явною друкарською помилкою на інше слово без друкарської помилки, але не відповідне за змістом. У результаті автор просто не помічає друкарської помилки. Правила пунктуації здатні суттєво впливати на зміст висловлювань масово не дотримуються серед соціальних мереж.

2 РОЗРОБКА АЛГОРИТМУ КЛАСТЕРИЗАЦІЇ КОРОТКИХ ТЕКСТІВ

2.1 Формулювання завдання мовного моделювання для коротких текстів

Завдання мовного моделювання у вузькому значенні – спрогнозувати наступне слово у тексті, знаючи послідовність попередніх слів. Результат вирішення цього завдання має конкретне практичне застосування: інтелектуальні клавіатури, генерація відповіді на e-mail [7], виправлення друкарських помилок. Спочатку були запропоновані підходи, в основі яких лежить N- грамова модель. Методи згладжування дозволяють опрацювати N-грами, які модель не зустрічала [5]. Перша мовна модель на основі штучної нейронної мережі була запропонована Йошуа Бенжіо [6], у роботі запропоновано схему функціонування (рисунок 2.1).

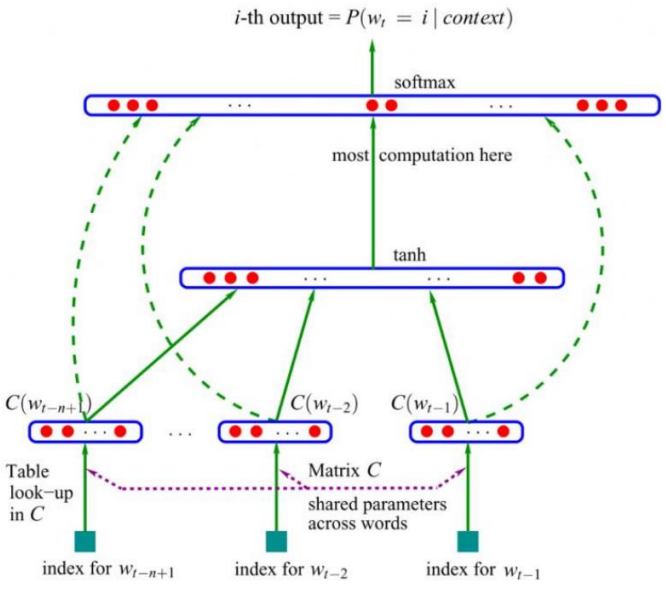


Рисунок 2.1 – Схема роботи першої нейронної мовної моделі

На вхід даної моделі подається набір векторних уявлень попередніх n слів. Стислі векторні уявлення називають ембеддингами (word embedding). Ці

вектори конкатенуються та передаються на прихований шар нейронної мережі. Вихідні дані прихованого шару потім передаються у шар із функцій softmax. Пізніше для вирішення задачі мовного моделювання замість нейронних мереж із прямим зв'язком почали використовуватися рекурентні нейронні мережі [7], а ще пізніше мережі з довгою короткостроковою пам'яттю [3]. В останні роки запропоновано багато нових мовних моделей, які розширюють можливості класичних мереж LSTM.

2.2 Особливості кластеризації коротких текстів

Сучасні методи кластеризації з використанням нейронних мереж зазвичай використовують нейронну мережу для підготовки векторів ознак і потім використовується аналітичний метод (заснований на формулах з гіперпараметрами) для кластеризації цих ознак. Через війну результат кластеризації обумовлений якістю отриманих векторів ознак, тобто якістю навчання нейронної мережі. При цьому останнім часом з'являються методи, що дозволяють вирішувати задачу кластеризації безпосередньо за допомогою нейронної мережі, що дозволяє об'єднати процес отримання векторів ознак і кластеризації. У своєму дослідженні [1] автори пропонують узагальнену схему побудови сучасних методів кластеризації з використанням нейронних мереж, в яку вкладається переважна більшість методів (рисунок 2.2) [3,11].

Схема побудови методу кластеризації принципово об'єднує етапи конструювання векторів ознак та угруповання об'єктів у кластери за рахунок використання загальної цільової функції.

На рисунку 2.2 наведена схема роботи алгоритму DEC та архітектура нейронної мережі, що використовується в ньому. Підхід, використаний DEC, повністю задовольняє вищевикладеної схемою. Початкова ідея методу викладена у статті [8].

Рисунок 2.2 – Узагальнена схема побудови методів кластеризації на базі нейронних мереж

Спочатку конструюється автоэнкодер, щоб отримати стислі векторні уявлення на виході блоку кодера. Автоэнкодер пропонувалося навчати на досліджуваному наборі даних, що як зазначалося вище, у разі коротких текстів який завжди можливо. Далі кодер поєднується з блоком, що відповідає за визначення центрів кластерів, і продовжується навчання мережі з використанням цільової функції у вигляді міри Кульбака-Лейблера.

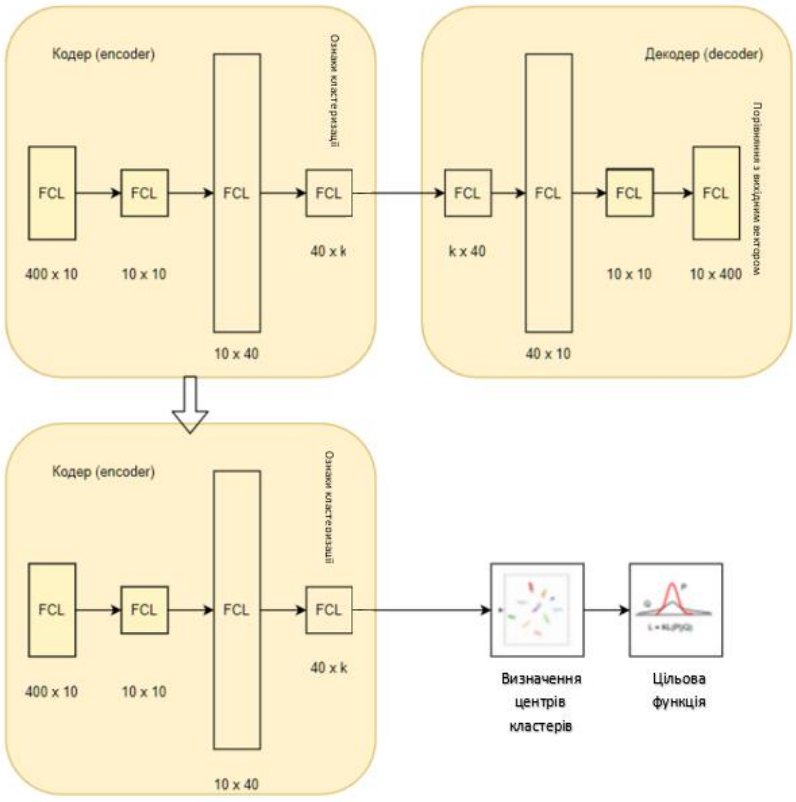


Рисунок 2.3 – Схема роботи та архітектура нейронної мережі алгоритму DEC

В результаті виконання дослідження пропонується для обробки текстів додатково перед кодером описаної архітектури використовувати кодер мовної моделі. Архітектура ропропонованої нейронної мережі представлена на рисунку 2.4. За кодером мовної моделі йдуть шари, які відповідають за побудову векторів ознак кластеризації з урахуванням векторів ознак текстів. При цьому шари, що відповідають за отримання векторів ознак для кластеризації, також ініціалізуються як частина автоенкодера.

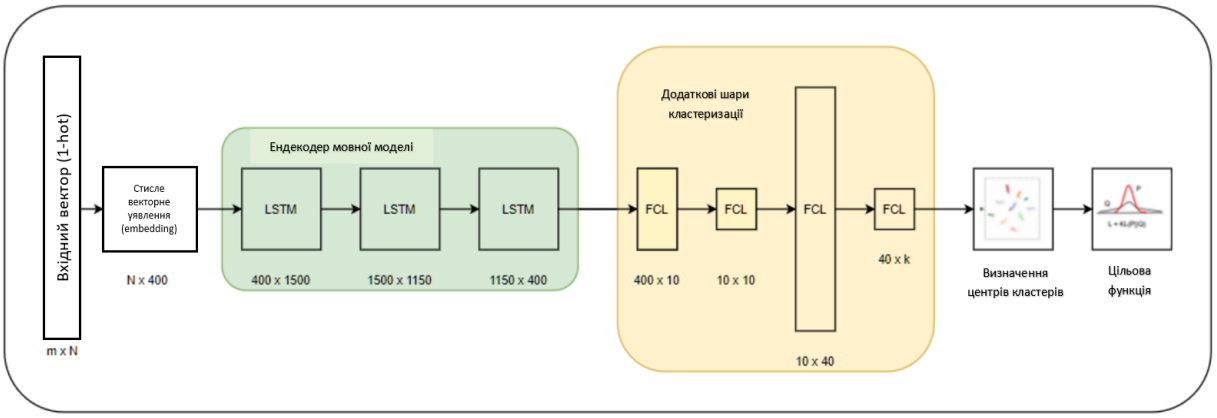


Рисунок 2.4 – Архітектура нейронної мережі для обробки текстів

Розглянемо докладніше розроблений під час виконання дипломного дослідження метод кластеризації. Як вже було зазначено вище, як базовий обраний метод кластеризації DEC (Unsupervised Deep Embedding for Clustering Analysis), при цьому аналогічний підхід може бути застосований до багатьох інших алгоритмів, наприклад, DEPICT [3].

Вхідний набір даних: $X = \{ x_i \mid i \in [0, N) \}$, де N – кількість елементів у наборі.

Ця множина за допомогою енкодера, що є частиною заздалегідь навченого автоенкодера, відображається в простір меншої розмірності:

- $f : X \rightarrow Z$, де X – параметри нейронної мережі;
- Z – прихований простір ознак.

Простір ознак називається у разі прихованим, тому його побудова відбувається в процесі навчання нейронної мережі і потім воно формується

неявним чином у процесі навчання автоенкодера та розв'язання задач кластеризації. У роботі використовується кодер наступної структури:

- d-10-10-40-k, де d-розмірність вхідного набору даних;
- k –число кластерів.

Результатом роботи алгоритму є набір центрів кластерів у просторі Z: $\{\mu_j \in Z | j \in [0, k)\}$, де k – задане число кластерів.

Для ініціалізації ваг додаткових шарів кластеризації використовується автоенкодер, декодері якого симетрично повторюються шари кодера. Застосовується типова схема навчання автоенкодера: спочатку кожен шар навчається окремо, потім додатково всі шари навчаються разом. Ініціалізація центрів кластерів відбувається за допомогою алгоритма k-means, який застосовується до представлення векторів, отриманого в результаті навчання автоенкодера. Процеси визначення оптимального розташування центрів кластерів та побудови простору ознак відбуваються одночасно за рахунок визначення загальної функції втрат. Для цього як міра відстані між елементом та центром кластера використовується метрика, заснована на розподілі Стьюдента з одним ступенем свободи.

$$\frac{(1 + ||z_i - \mu_j||^2)^{-1}}{\sum_{l=0}^{k}(1 + ||z_i - \mu_l||^2)^{-1}}). \quad (2.1)$$

Цільова функція (функція втрат, loss function) або штрафна функція будується як метрика Кульбака-Лейблера (Kullback-Leibler divergence) між фактичним та цільовим розподілом.

$$L = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \quad (2.2)$$

Як цільовий розподіл використовується наступний розподіл

$$p_{ij} = \frac{{q_{ij}}^2/f_j}{\sum_{l=0}^{k} {q_{il}}^2/f_l}, \qquad f_j = \sum_j q_{ij}. \tag{2.3}$$

Цей розподіл має наступні властивості: посилює внесок від елементів з великою часткою належності кластеру і нормалізує вплив великих кластерів, не дозволяючи їм надмірно притягувати до себе віддалені елементи за рахунок свого розміру. Для оновлення вагів нейронної мережі та перерахунку центрів кластерів використовуються такі приватні похідні цільової функції.

$$\frac{\partial L}{\partial z_i} = 2 \sum_j \left(1 + \left|\left|z_i - \mu_j\right|\right|^2\right)^{-1} * (p_{ij} - q_{ij}) * (z_i - \mu_j), \tag{2.4}$$

$$\frac{\partial L}{\partial \mu_j} = -2 \sum_i \left(1 + \left|\left|z_i - \mu_j\right|\right|^2\right)^{-1} * (p_{ij} - q_{ij}) * (z_i - \mu_j). \tag{2.5}$$

Найчастіше під час вирішення завдання кластеризації є гіпотеза чи вимоги до рівномірності розподілу елементів за кластерами. Для цього в цільову функцію можна додати доданок, що додає штраф за нерівномірність розподілу. Таким чином, цільова функція матиме вигляд

$$\begin{aligned} \mathcal{L} &= KL(\mathbf{Q}\|\mathbf{P}) + KL(\mathbf{f}\|\mathbf{u}) \\ &= \left[\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} q_{ik} \log \frac{q_{ik}}{p_{ik}}\right] + \left[\frac{1}{N}\sum_{k=1}^{K} f_k \log \frac{f_k}{u_k}\right] \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} q_{ik} \log \frac{q_{ik}}{p_{ik}} + q_{ik} \log \frac{f_k}{u_k}, \end{aligned} \tag{2.6}$$

2.3 Алгоритм інтерактивної кластеризації коротких текстів

Для проведення процедури кластеризації у результаті отримано класифікатор показників СП розроблено алгоритм кластеризації набору коротких текстів. Алгоритм складається з наступних кроків:

Крок 1. Передобробка текстів: виправлення орфографії, токенізація.

Крок 2. Розширення словника мовної моделі.

Крок 3. Точне налаштування мовної моделі (додаткове навчання).

Крок 4. Ініціалізація шарів нейронної мережі блоку кластеризації (навчання автоенкодера)

Крок 5. Ініціалізація центрів кластерів (з використанням k- means)..

Крок 6. Первинна кластеризація (синхронне навчання нейронної мережі кластеризації коротких текстів та визначення центрів кластерів).

Кроки інтерактивної кластеризації (цикл до досягнення прийнятної для експерта якості кластеризації):

- аналіз результатів кластеризації експертом;
- отримання матриці зворотній зв'язку від експерта;
- коригування ваг нейронної мережі та коригування центрів кластерів на підставі матриці множників зворотного зв'язку.

Слід зазначити, що основна обчислювальна складність посідає перші три кроки даного алгоритму.

Для виправлення орфографії необхідний спеціалізований зовнішній сервіс для мови набору даних. Такі послуги, крім виправлення типових орфографічних помилок, можуть виправляти помилки з урахуванням контексту, що позитивно позначається на якості кластеризації. Крок тонкої установки нейронної мережі займає значний час (годинник), т.к. сучасні мовні моделі містять мільйони параметрів та велику кількість шарів.

Для економії обчислювальних ресурсів може навчатися лише певна кількість останніх верств мовної моделі, але це, як правило, погіршує якість мовної моделі, знижуючи точність та перплексію. Кроки інтерактивної

кластеризації займають лише кілька епох, т.к. умовою припинення навчання є досягнення порога мінімальної кількості переміщених елементів із кластера до кластера. Ця операція займає час від секунд до кількох хвилин, таким чином інтерактивна робота з експертом може вестися в режимі реального часу.

3 РЕАЛІЗАЦІЯ ТА ВЕРИФІКАЦІЯ МЕТОДУ НЕЧІТКОЇ КЛАСТЕРИЗАЦІЇ КОРОТКИХ ТЕКСТІВ

3.1 Опис моделі нечіткої кластеризації коротких текстів

З урахуванням особливостей запропонованого методу програмний модуль повинен задовольняти наступним вимогам:

- інтерфейс програмного комплексу повинен підтримувати взаємодію з користувачем в інтерактивному режимі, запам'ятовувати введені користувачем обмеження та враховувати їх у наступних ітераціях кластеризації;
- програмний модуль має бути реалізований у триланковій архітектурі, для можливості розміщення серверної частини на продуктивних серверах та надання клієнтського доступу до програми через браузер ПК;
- програмний модуль має бути реалізований на open-source технологіях і під Windows, CentOs, Linux Astra, тобто відповідати вимогам щодо імпортозаміщення.

З урахуванням заданих обмежень для реалізації програмного модуля обрано мову Python, що дозволяє як реалізувати алгоритмічну частину за допомогою фреймворку FastAI [5], так і забезпечити потенційну реалізованість клієнт-серверної частини, що забезпечує взаємодію з користувачем за допомогою фреймворку Flask. Інтерпретатор цієї мови вбудований у всі ОС сімейства Linux, а для Windows є зручні IDE для розробників.

На рисунку 3.1 представлено архітектуру розробленого комплексу. В рамках виконання дипломної роботи розроблено блок машинного навчання, що дозволив провести всі необхідні експерименти та вирішити поставлені завдання.

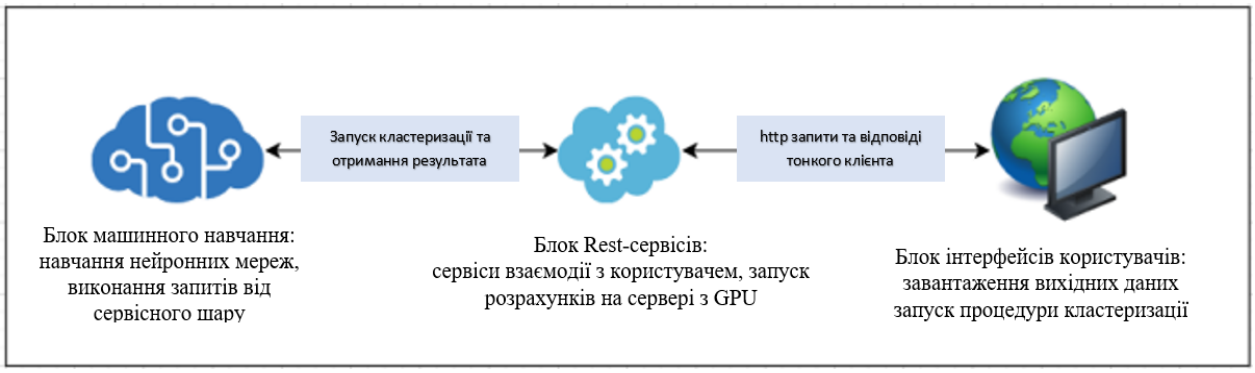


Рисунок 3.1 – Триланкова архітектура програмного комплексу для інтерактивної кластеризації коротких текстів

Для перевірки результатів дослідження потрібно було спроектувати блоки Rest сервісів та інтерфейсів користувача.

3.1.1 Блок Машинного навчання

Блок Машинного навчання, що відповідає за попередню обробку текстів, навчання мовної моделі та навчання нейронної мережі для кластеризації коротких текстів, має структуру, представлену на рисунку 3.2. Він складається з двох основних великих блоків, що відповідають за перший і другий етапи представленого методу відповідно. У межах кожного блоку навчається нейронна мережу, яка згодом використовується блоком Rest - сервісів на вирішення завдання кластеризації. У блоці попередньої обробки відбувається виправлення друкарських помилок за допомогою відкритого сервісу від Яндекс, видалення розділових знаків, цифр та інших символів не придатних для обробки в мовній моделі.

Для етапу навчання нейронних мереж та контролю якості навчання використовувався інструмент Jupyter Notebook дозволяє покроково виконувати команди з миттєвим виведенням результату команди. Цей інструмент є стандартом серед фахівців з обробки даних. Кожен блок з рисунка 3.2 відповідає файл у Jupyter Notebook.

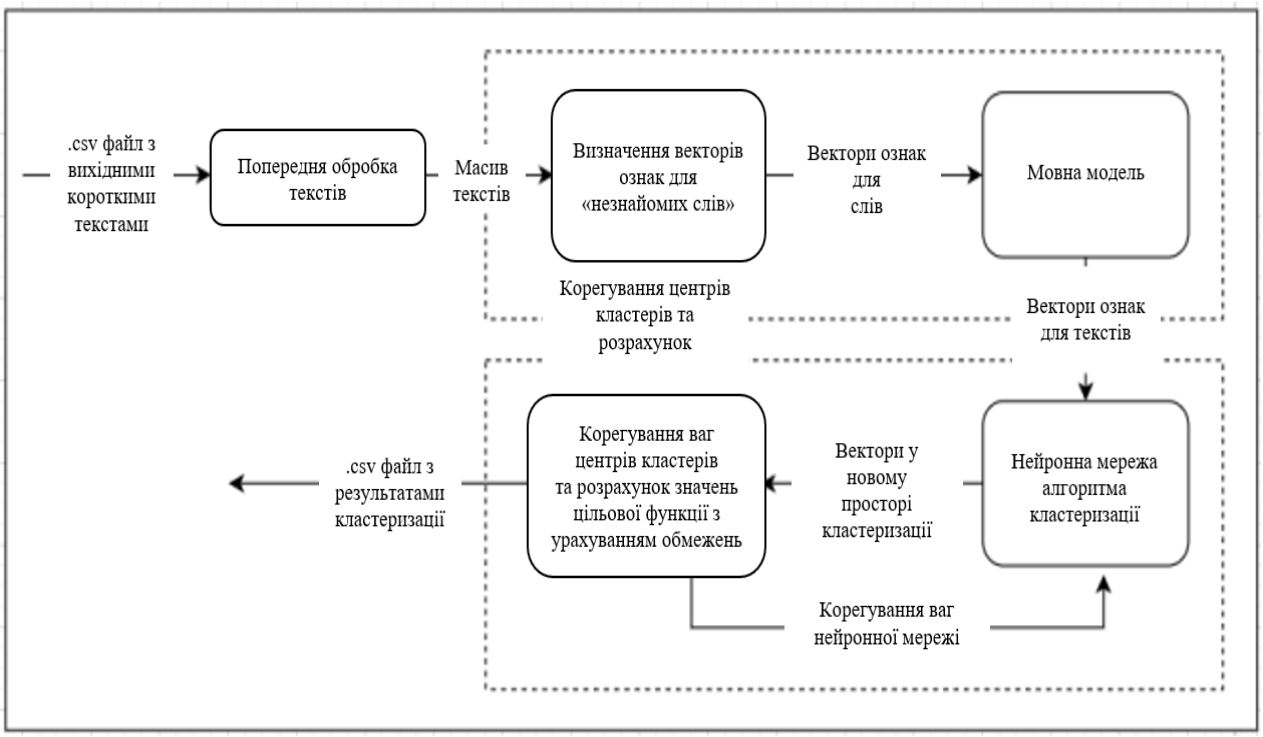


Рисунок 3.2 – Функціональна структура блоку машинного навчання

На рисунку 3.3 представлений файл із модулем побудови векторів ознак для “невідомих” слів.

Для імплементації нейронної мережі, що реалізує мовну модель, обраний фреймворк FastAI, на якому доступна модель ULMFit, що є однією з найсучасніших мовних моделей і найпоширенішою для мов слов'янської групи. Даний фреймворк є набором класів над популярним фреймворком PyTorch, що є стандартом для реалізації сучасних нейронних мереж поряд з Keras. При цьому PyTorch володіє більш гнучким API (application programming interface).

Імплементація нейронної мережі для кластеризації проводилася за допомогою фреймворку MXNet від Apache. За допомогою даного фреймворку реалізований метод кластеризації DEC взятий за основу в даному дослідженні та доопрацьований для застосування в інтерактивній кластеризації. Цей фреймворк також функціонує поверх фреймворку PyTorch

.

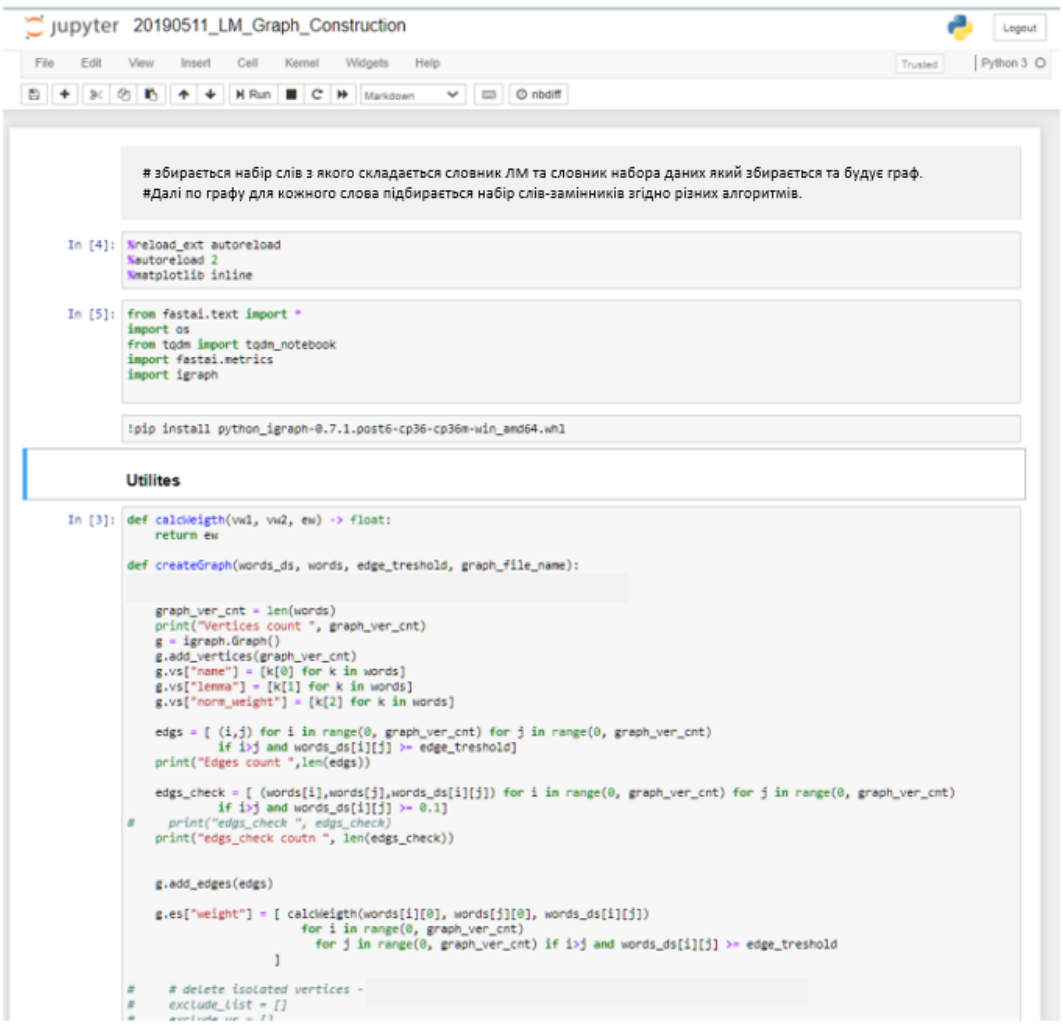


Рисунок 3.3 – Модуль побудови векторів ознак для “невідомих” слів у Jupyter Not

3.1.2 Блок Rest -сервісів

Проект блоку Rest – сервісів, що реалізує сервісний шар взаємодії з користувачем та запускає ітерацію алгоритму кластеризації, складається з одного сервісу з наступними методами:

- uploadShortTextDataset – завантаження csv – файлу з вихідними даними;

- startPreProcessing – запуск процедури попередньої обробки коротких текстів у блоці машинного навчання;

- startVocabularyUpdate – запуск у блоці машинного навчання процедури обробки "невідомих" (для мовної моделі) слів з вхідного набору

даних, повернення числа слів для яких вдалося побудувати вектори ознак на основі наявного словника мовної моделі;

- startLMTuning – запуск процедури додаткового навчання (тонкого налаштування) нейронної мережі, що реалізує мовну модель, повернення результатів за значенням цільової функції та точності роботи мовної моделі;
- downloadShortTextVec – передача csv -файлу з векторами ознак для вхідного набору даних для завантаження з браузера;
- u ploadShortTextVec – завантаження csv -файлу з векторами ознак для вхідного набору даних коротких текстів;
- startClusteringIteration – запуск алгоритму кластеризації з передачею введених обмежень, повернення значення функції втрат, індексів якості кластеризації та масиву списку кластерів з основними характеристиками;
- getClusterName – отримання імені кластера за ідентифікатором короткого тексту;
- getClusterContent - отримання списку елементів за ідентифікатором кластера та набором фільтрів (певний відсоток випадкових текстів, найбільш близькі до центру кластера, найбільш далекі від центру кластера).

3.1.3 Блок інтерфейсів користувача

Дизайн блоку інтерфейсів користувача, що відповідає за інтерактивну взаємодію з користувачем і візуалізацію результатів роботи запропонованого методу, складається з трьох екранних форм. Перша екранна форма представлена на рисунку 3.4 і містить такі основні керуючі та візуальні елементи:

- кнопка “Завантажити файл” призначена для завантаження csv- файлу з вихідним набором даних;
- кнопка “Переробка” призначена для запуску попередньої обробки вихідного набору даних (виправлення помилок, приведення до формату придатного для обробки мовною моделлю);

- поряд з кожною кнопкою, яка запускає тривалу процедуру на сервері, розташовується елемент “progress bar”, що відбиває динаміку роботи серверної процедури;

- кнопка “Підготовка словника” призначена для запуску процедури побудови векторів ознак для слів мовної моделі, що не входять до словника;

- візуальний елемент виведення числа слів доданих до словника мовної моделі відображає значення, повернене процедурою після натискання на кнопку “Підготовка словника”;

- кнопка “Відкрити папку” відкриває серверну папку з файлами для програми Gephi, що містять граф, побудований для кожного зі слів мовної моделі, що додається в словник;

- поле для введення розміру етапу навчання нейронної мережі дозволяє впливати на швидкість навчання в рамках однієї епохи;

- поле для введення кількості епох навчання нейронної мережі дозволяє проводити тривалість навчання;

- поле для введення кількості шарів для навчання дозволяє визначити кількість останніх шарів нейронної мережі, для яких застосовуватиметься коригування вагових коефіцієнтів;

- кнопка “До-навчання” дозволяє запустити процедуру тонкої настройки нейронної мережі, особливо предметної області та лексики вихідного набору даних.

- візуальні елементи виведення значення цільової функції та розрахованої точності роботи мовної моделі на валідаційній вибірці відображають значення, що повертаються в результаті роботи процедури кнопкою “До-навчання”;

- кнопка “Завантажити результат” дозволяє отримати csv -файл із векторами ознак для кожного тексту вихідного набору.

Цей файл є вихідним файлом для наступного етапу кластеризації.

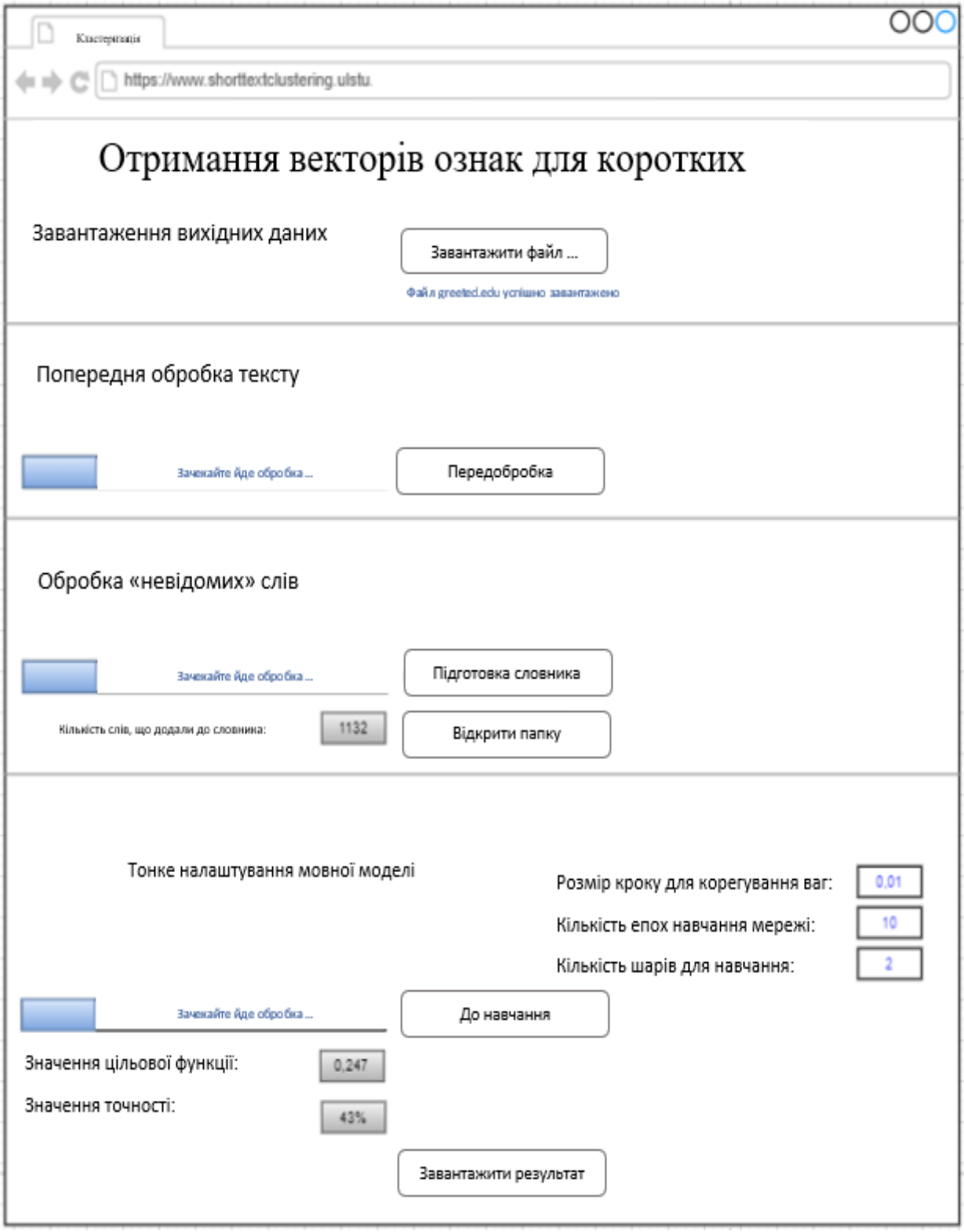


Рисунок 3.4 – Екранна форма отримання векторів ознак для коротких текстів

Друга екранна форма представлена на рисунку 3.3 і містить такі основні керуючі та візуальні елементи:

Кнопка “Завантажити файл” призначена для завантаження файлу csv з вихідним набором даних, що містить вектори ознак для коротких текстів.

Поле для введення числа кластерів задає гіперпараметр для алгоритму кластеризації.

Таблиця “Обмеження” за допомогою кнопок “+” та “–” дозволяє встановити обмеження для ітерації алгоритму кластеризації у форматі: (об'єкт; кластер; увімкнути/виключити). Також після виконання ітерації кластеризації в колонці "Новий кластер" буде виведено найменування кластера, до якого увійшов цей об'єкт (текст).

Кнопка “Кластеризація” запускає процедуру кластеризації із заданим числом кластерів та вказаним набором обмежень.

В результаті роботи процедури кластеризації будуть заповнені візуальні елементи зі значенням цільової функції та індексом якості кластеризації.

Таблиця “Результати кластеризації” також заповнюється після виконання процедури кластеризації та містить інформацію про отримані кластери, кількість елементів у кожному, щільність, мінімальну та максимальну відстань елементів від центру кластера, а також дозволяє вручну користувачу задати ім'я кластера. Введене ім'я буде запам'ятоване і на наступних ітераціях застосовуватиметься до кластера, що має найбільш загальну кількість елементів із кластером з попередньої ітерації.

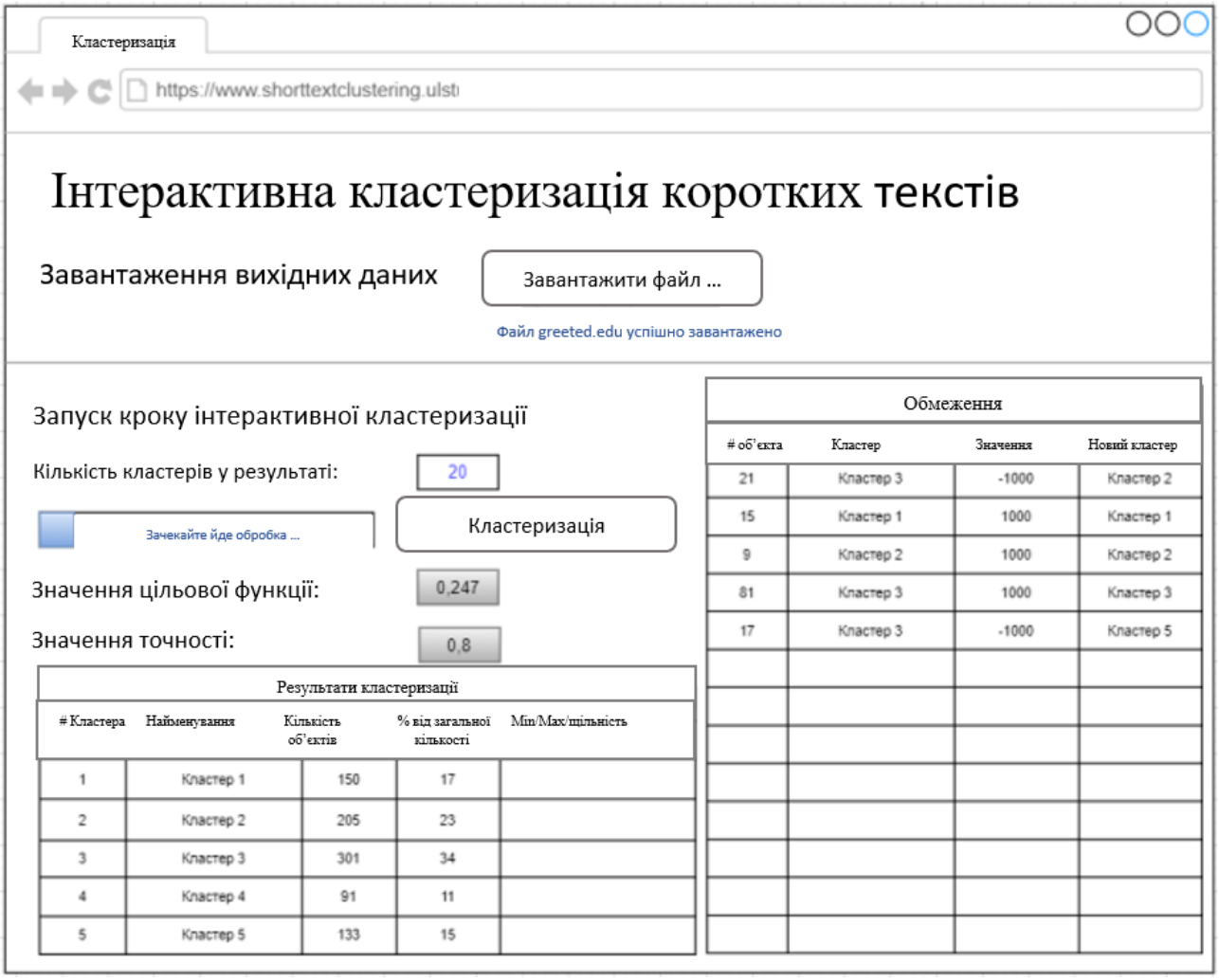


Рисунок 3.5 – Екранна форма кластеризації коротких текстів

Третя екранна форма представлена на рисунку 3.6, детально розглянемо її керуючі та візуальні елементи.

Блок перемикачів керуючих режимом відбору елементів кластера: може бути обраний відсоток випадкових об'єктів; певну кількість найближчих об'єктів чи найбільш далеких об'єктів.

Блок візуальних елементів у правій частині форми відображає статистичну інформацію про кластер: число елементів у кластері, відсоток від загального числа елементів та щільність об'єктів у кластері в евклідовому просторі кластеризації.

Кнопка "Показати" запускає процедуру відбору елементів кластера. Результат роботи процедури відбору елементів відображається у таблиці

“Об'єкти кластеризації”. Крім номера та найкоротшого тексту в таблиці відображається евклідова відстань до центру кластера.

Об’єкти кластеризації

# об’єкт	Текст	Близькість до центру
21	Доля педагогічних працівників загальноосвітніх організацій яким при проходженні атестації у відповідному році була призначена перша або вища категорія	0,05
15	Доля педагогічних працівників муніципальних загальноосвітніх установ яким при проходженні атестації у відповідному році була призначена перша або вища категорія	0,09
9	питома вага випускників класів отримавших атестат о середній загальній освіті у загальній кількості випускників класів	0,12
81	Кількість навчаючихся завершивших навчання по загальноосвітнім програмам основної загальної освіти, що підлягають державній підсумковій атестації	0,29
17	Доля педагогічних працівників загальноосвітніх установ яким при проходженні атестації у відповідному році була призначена перша або вища категорія	0,06

Рисунок 3.6 – Екранна форма перегляду вмісту кластеру

3.2 Верифікація методу на синтетичному наборі даних

Для демонстрації роботи згенеровано набір даних із 400 елементів, таким чином: 1. За основу взято 4 вектори {(1,0,0,0); (0,1,0,0); (0,0,1,0); (0,0,0,1)} 85 2. Для кожного з 4 векторів згенеровані 125 векторів додаванням до кожної компоненти випадкової величини з рівномірного розподілу U [0, 1/10]. Доданий випадковий шум невеликий, т.к. у цьому експерименті завданням ставиться показати вплив зворотний зв'язок, а чи не якість роботи алгоритму себе. 3. Вектори у вибірці розташовані послідовно четвірками, таким чином перші 4 вектори містять по 1 представнику від кожного

базового класу. У результатах експериментів детально будуть показані лише перші 12 векторів для стислості та ясності картини результату. Для зазначеного набору даних запущено алгоритм кластеризації з розбиттям множини на 2 кластери. Таблиця 3.1 містить результати роботи алгоритму кластеризації, причому для автоэнкодера досягнуте значення функції втрат на перевірочній вибірці дорівнює 0.000341. Рисунок 3.7 відображає розподіл перших 12 векторів за кластерами. Для отримання проекції на двовимірну площину використовувався метод t-SNE, значення відкладені по осях є безрозмірними величинами.

Далі будуть показані три експерименти для різних видів зворотного зв'язку:

- вказівка про необхідність включення вектора X_0 кластер C_0;
- вказівку про необхідність виключення вектора X_1 із кластера C_1;
- комплексний зворотний зв'язок щодо заміни векторів X_2 та X_3 місцями у кластерах C_0 та C_1 відповідно.

Усі експерименти виконуються як перша ітерація після початкової кластеризації, а чи не послідовно, виключно зручнішого порівняння. Послідовне застосування, очевидно, можливо, без будь-яких обмежень чи особливостей у роботі алгоритму.

Таблиця 3.1 – Список перших 4 векторів набору даних

№	Координати				Кластер
0	1.0191519	0.06221088	0.04377278	0.07853585	C_0
1	0.07799758	1.0272592	0.02764643	0.08018722	C_0
2	0.09581394	0.08759326	1.0357817	0.05009951	C_1
3	0.06834629	0.07127021	0.03702508	1.0561196	C_1

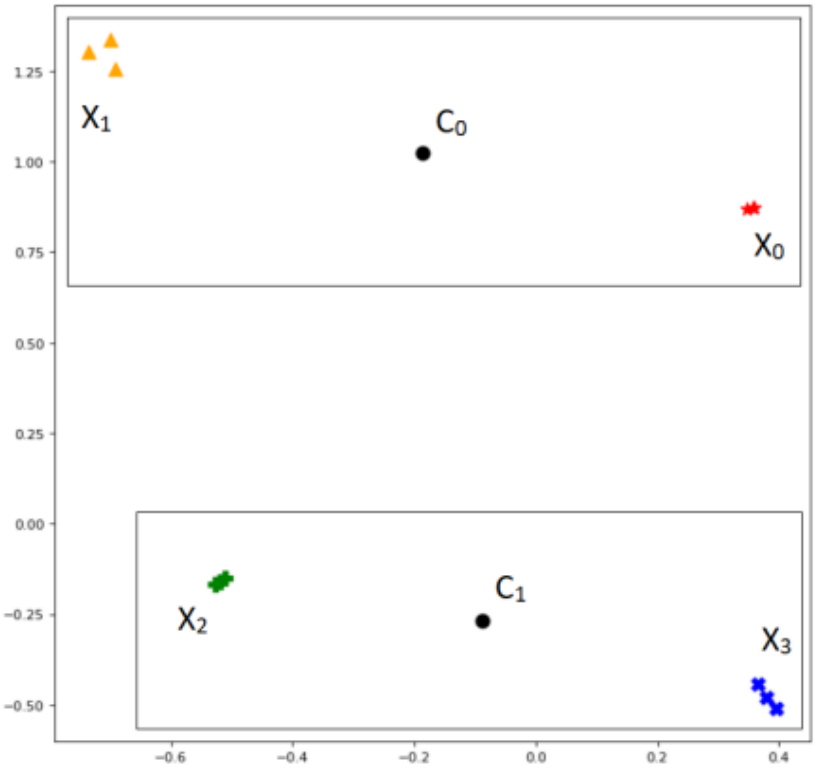


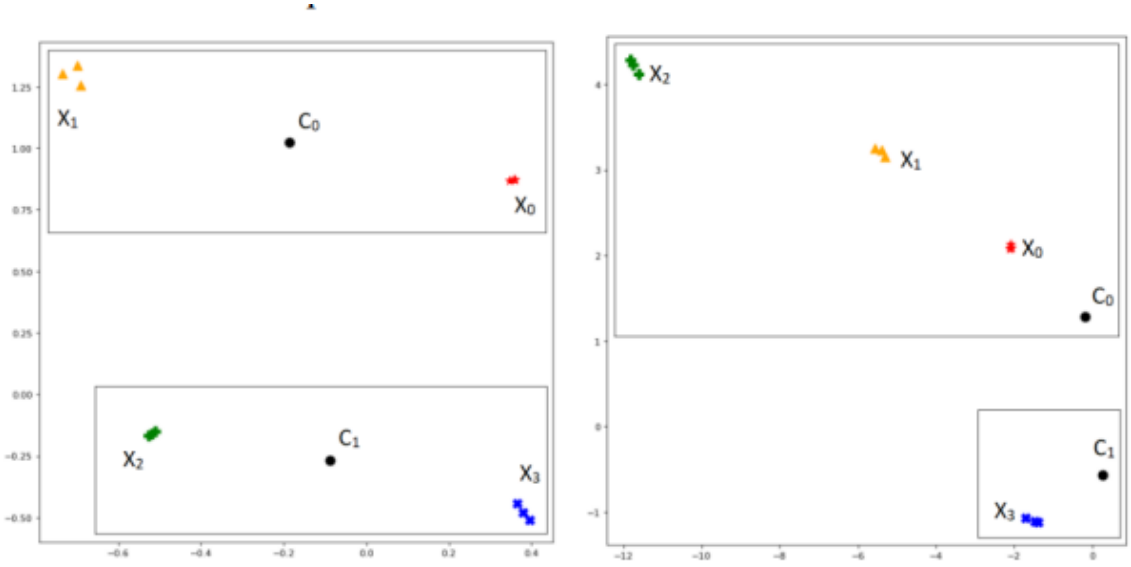
Рисунок 3.7 – Результат кластеризації для перших 12 векторів

У першому експерименті припустимо, що дослідник має інформацію, у тому, що вектор X_1 семантично ближче до векторів X_2 і X_3, ніж вектору X_0. Тож алгоритму кластеризації формується зворотний зв'язок як матриці T[500,4] = $\{t_{ij} \mid i \in [0, 500), j \in [0, 4)\}$, що вказує, що вектор X_1 повинен перейти в кластер C_1.

$$t_{ij} = \begin{cases} 1000, & i = 1, \quad j = 1 \\ 1, & \end{cases}$$

Результати 100 епох роботи алгоритму кластеризації представлені на рисунку. 3.7. Вектор X_2 залишив кластер C_1 і разом з вектором X_2 кластер C_1 перемістилися і всі інші вектори 3 класу. У цьому можна побачити, що т.к. виняток із кластера це, по суті, ослаблення сили тяжіння між вектором і центром кластера, то клас, що виключається, виявився максимально далеко від центру кластера C_1 і досить далеко від центру кластера C_0. Також можна

помітити передбачувано нижчу швидкість збіжності алгоритму кластеризації при операції виключення з кластера, ніж операції включення в кластер.



a b

Рисунок 3.8 –Результат кластеризації для виключення вектора X_2 із кластера C_1 (a – вихідні дані, b – результат кластеризації)

У третьому експерименті припустимо, що дослідник має інформацію про необхідність змінити розташування відразу двох векторів. Вектор X_1 потрібно перемістити в кластер C_1, а вектор X_2 перемістити в кластер C_0. Для алгоритму кластеризації формується зворотний зв'язок у вигляді матриці $T[500,4] = \{ t_{ij} \mid i \in [0, 500), j \in [0, 4)\}$ наступного виду:

$$t_{ij} = \begin{cases} 1000, & i = 1, \quad j = 1 \\ 1000, & i = 2, \quad j = 0 \\ 1, & \end{cases}$$

Результати 100 епох роботи алгоритму кластеризації представлені на рисунку 3.8. Класи 2 і 3 помінялися місцями за своїми представниками векторами X_1 і X_2.

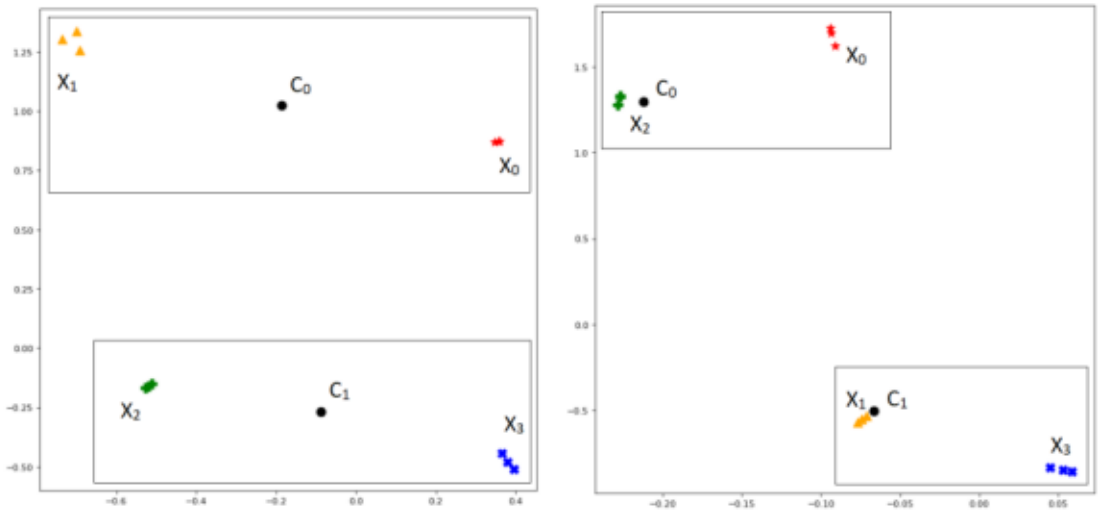


Рисунок 3.9 – Результат кластеризації заміни місцями векторів X_1 та X_2 у кластерах (a – вихідні дані; b – результат кластеризації)

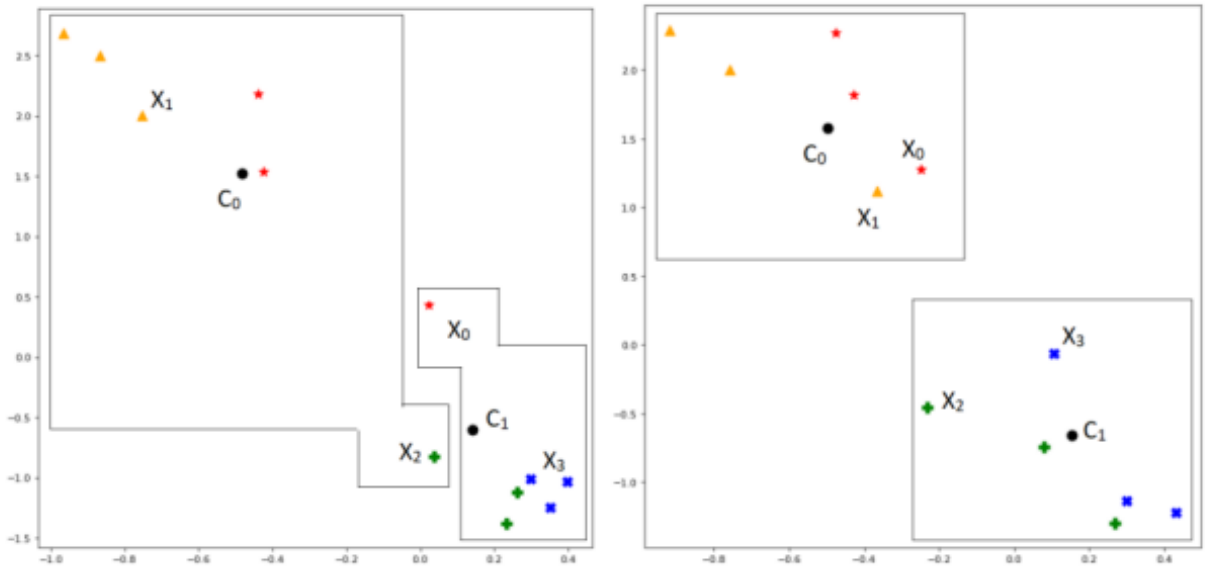
У четвертому експерименті змінимо набір даних, збільшивши рівень шуму у векторах у десять разів додаванням до кожної компоненти випадкової величини рівномірного розподілу U [0, 1]. Перші 4 вектори та результат кластеризації зазначені в таблиці 3.2. Рисунок 3.9 a відображає перші 12 векторів та результат кластеризації. Видно, що вектори X_0 і X_2 співвіднесені з кластерами не так. Для виправлення результату алгоритму кластеризації формується зворотний у вигляді матриці T[500,4] = { t_{ij} | i ∈ [0, 500), j ∈ [0, 4)} наступного виду:

$$t_{ij} = \begin{cases} 1000, & i = 0, \quad j = 0 \\ 1000, & i = 2, \quad j = 1 \\ 1, & \end{cases}$$

Результати 100 епох роботи алгоритму кластеризації представлені рисунку 3.9 b. Помилково співвіднесені вектори перейшли в коректні класи, причому інші представники класів, як і раніше, співвіднесені коректно, тому що внесені зміни об'єктивно покращили якість кластеризації та зміни надали точкову дію, на відміну від попередніх експериментів.

Таблиця 3.2 – Вектори зі збільшеним рівнем шуму

№	Координати				Кластер
0	**1.1915**	0.6221	0.4377	0.7853	C_0
1	0.7799	**1.2725**	0.2764	0.8018	C_0
2	0.9581	0.8759	**1.3578**	0.5009	C_1
3	0.6834	0.7127	0.3702	**1.5611**	C_1



a b

Рисунок 3.10 – Результат кластеризації векторів зі збільшеним шумом із точковим виправленням (a – вихідні дані, b – результат кластеризації)

3.3 Верифікація методу на класичному завданні класифікації та кластеризації «Іриси Фішера»

Набір даних «Іриси Фішера» є класичним для завдань класифікації та кластеризації [7]. У наборі представлені довжина і ширина зовнішньої та внутрішньої часток оцвітини для трьох видів ('setosa', 'versicolor', 'virginica'). Таблиця 3.3 представлені приклади 3-х векторів по одному для кожного виду. До вихідних даних до кожної компоненти додано випадкову величину з

рівномірного розподілу U [0, 1/10] для ускладнення задачі та генерації 600 прикладів з 150 наявних.

Таблиця 3.3 – Приклад векторів із набору даних “Іриси Фішера”

№	Координати				Кластер
0	5.119	3.562	1.443	0.278	C_0 (setosa)
1	7.077	3.227	4.727	1.480	C_1 (versicolor)
2	6.395	3.387	6.032	2.550	C_2 (virginica)

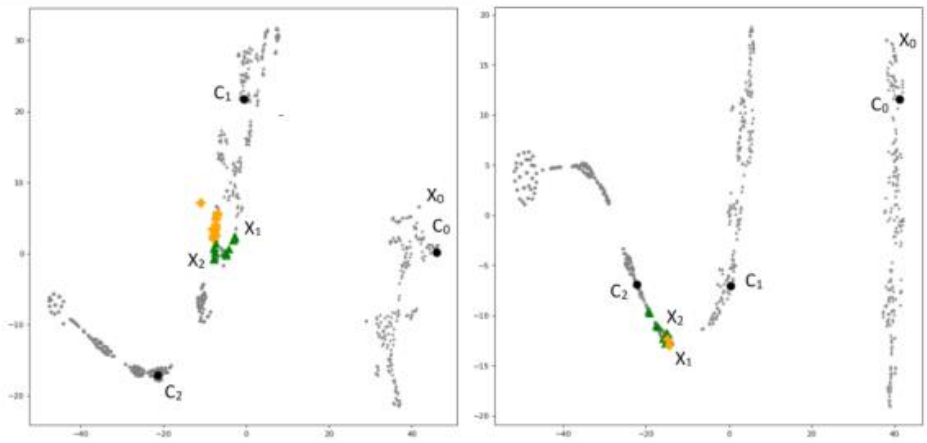
Рисунок 3.10 відображає результати роботи алгоритму кластеризації для 3-х кластерів. Кластер, відповідний виду ' setosa ', чітко і безпомилково відділений, а кластерах двох інших видів є 13 і 16 переплутаних місцями векторів. Види 'versicolor' і 'virginica' є близькими, і навіть завдання класифікації їх вдається однозначно розділити. Тим не менш, припустимо, що дослідник має інформацію про два вектори (примірниках квітки, видова приналежність яких йому цілком могла бути відома), які необхідно поміняти місцями. У цьому прикладі це вектори з порядковим номером 7 і 50.

І тому формується зворотний зв'язок як матриці

T[600,3] = { t_{ij} | i ∈ [0, 600), j ∈ [0, 3)} наступного виду:

$$t_{ij} = \begin{cases} 1000, & i = 7, \quad j = 2 \\ 1000, & i = 50, \quad j = 1 \\ 1, & \end{cases}$$

Результати 100 епох роботи алгоритму кластеризації представлені Рис. 3.10 b (для проекції на площину 3-мірних даних використовувався алгоритм t-SNE реалізований в python бібліотеці sklear). Помилково співвіднесені вектори перейшли в коректні класи, причому більшість помилок також виправилася.



a b

Рисунок 3.11 – Результат зміни розподілу кластерів у наборі “Іриси Фішера”

Переплутаними залишилися 8 і 2 векторів у 2-му та 3-му класах відповідно. Це дає точність (accuracy) рівну 0.98 (3). Такі результати важко досягнути навіть для алгоритмів класифікації, середнім результатом кращих алгоритмів є точність 0.971. Алгоритми кластеризації часто цілком не здатні розрізнити другий і третій види [8].

Кількість неправильно співвіднесених векторів зменшилася з 26 до 10.

ВИСНОВКИ

У кваліфікаційній роботі розроблено метод нечіткої кластеризації коротких текстів.

Проведено дослідження методів машинного навчання для обробки текстів, та розроблено архітектуру штучної нейронної мережі, що реалізує кластеризацію на базі простору ознак мовної моделі української мови.

Розроблено метод обробки текстів для розширення словника мовної моделі на базі нейронної мережі з використанням нечіткого ієрархічного класифікатора, який дозволяє підвищити точність кластеризації в середньому на 10%.

Також наводиться опис спроектованих блоків програмного модуль, що дозволяє автоматизувати роботу експерта для вирішення задачі нечіткої інтерактивної кластеризації коротких текстів. Цей програмний модуль використовувався для апробації моделі, оцінки продуктивності та ефективності нечіткої кластеризації наборів коротких текстів українською мовою без попередньої обробки.

Програмний модуль, що виконан у триланковій архітектурі, дозволив ефективно організувати спільну роботу експертів у ході рішення задачі нечіткої кластеризації набору даних коротких текстів

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Стеблянко Б. О., Ні О. В., Кучук Г. А., Волк Д.М. Метод нечіткої інтерактивної кластеризації. Системи управління, навігації та зв'язку. Полтава : Національний університет «Полтавська політехніка імені Юрія Кондратюка», 2024. Вип. 2(76). С. 149–154.

2. Wang Y. Explicit routing algorithms для Internet Traffic Engineering / Wang Y., Wang Z. // Proc . 8th International Conference on Computer Communications and Networks. – Paris, 2019. – P. 582-588.

3. Younis O. Constraint-based routing in the internet : basic principles and recent research. / Younis O. Fahmy S. // IEEE Communication Society Surveys & Tutorials-2003. -Vol.5, №3. – P. 42-56.

4. Wang Z. Quality- of - service routing for supporting multimedia applications . / Wang Z., Crowcroft J. // IEEE JSAC. - 2016. - Vol . 14, № 7. - P. 1228-1234.

5. N. Likhanov . Analysis of an ATM buffer with self-similar (" Fractal ") input traffic . / N. Likhanov , B. Tsybakov , ND Georganas . // Proc . IEEE INFOCOM'95, Boston, MA. - 1995. - pp . 985-992.

6. Performance Testing : What is , Types , Metrics & Example – Режим доступу: https://www.guru99.com/performance-testing.html

7. Best Performance Testing Tools (Load Testing Tools) In 2020 – Режим доступу: https://www.softwaretestinghelp.com/performance-testing-tools-load-testing-tools/

8. Key performance indicators for load testing – Режим доступу https://www.soapui.org/learn/load-testing/key-performance-indicators-for-load-testing/

9. Yang B. Towards K-means-friendly spaces: Simultaneous deep learning and clustering. / B. Yang, X. Fu, N.D. Sidiropoulos, M. Hong // Proceedings of the 34th International Conference on Machine Learning. – 2017. – Vol. 70, P. 44-60.

10. Sutskever I. On the importance of initialization and momentum in deep learn-ing. / I. Sutskever, J. Martens, G. Dahl, G. Hinton // Proceedings of the 30th Interna-tional Conference on Machine Learning, PMLR 28(3). – 2013. – P. 1139-1147.

11. Ramachandran P. Unsupervised Pretraining for Sequence to Sequence Learn-ing. / P. Ramachandran, P. Liu, Q. Le // Proceedings of the 2017 Conference on Em-pirical Methods in Natural Language Processing, Copenhagen, Denmark. - 2017.

12. Nivre J. Universal Dependencies v1: A Multilingual Treebank Collection. / J. Nivre, M.C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajič, C.D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty, D. Zeman // Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia. -2016. – P. 1659–1666.

13. Novák V. A general methodology for managerial decision making using intel-ligent techniques / V. Novák, I. Perfilieva, N.G. Jarushkina // Chapter Recent Ad-vances in Decision Making, Series Studies in Computational Intelligence. - 2009. - Vol., 222. – P. 103-120.

14. Pedregosa F. Scikit-learn: Machine Learning in Python / F. Pedregosa, G.Va-roquaux, A.Gramfort, V.Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay // Journal of Machine Learning Research. – 2011. - Vol. 12, P. 111-119.

15. McCann B. Learned in Translation: Contextualized Word Vectors. / B. McCann, J. Bradbury, C. Xiong, R. Socher // NIPS / I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R.Fergus, S.V.N. Vishwanathan, R. Garnett (eds). – 2017. - P. 6297-6308.