Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломного проекту (роботи)

магістр
(освітній рівень)

на тему: *«Розробка математичного та програмного забезпечення для дослідження та виділення ключових слів у задачах виявлення екстремістської інформації в мережі інтернет»*

Виконав: студент (ка) *VI* курсу, групи *СБм-61*
Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Жаврук Р.А.
підпис (прізвище та ініціали)

Керівник *Карпінський М.П..*
підпис (прізвище та ініціали)

Нормоконтроль *Лобур Т.Б..*
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 20*20*

АНОТАЦІЯ

// Дипломна робота ОР «Магістр» // Жаврук Роман Андрійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2020 // С. 67, рис. – 12, табл. – 4, кресл. – 0, додат. – 1.

Ключові слова: КАРТИ КОХОНЕНА, КЛАСТЕРІЗАЦІЯ ТЕКСТІВ, ПУБЛІЧНІ ТЕКСТИ, МЕРЕЖА ІНТЕРНЕТ.

Дана магістерська кваліфікаційна робота присвячена дослідженню методів пошуку та виділення ключових слів для виявлення екстремістської інформації в публічних текстах з мережі інтернет. Проведено дослідження методів та інструментів для кластерізації публічних текстів з метою виявлення текстів екстремістської направленості.

Для визначення ключових слів з публічних текстів запропоновано використання карт Кохонена.

В роботі запропоновано програмне забезпечення, яке дозволяє проводити частоний аналіз публічних текстів з мережі інтернет з метою виявлення виявлення екстремістської інформації.

У першій главі наведено існуючі підходи до аналізу текстів..

У другій главі проведено порівняльний аналіз методів, які застосовуються для аналізу публічних текстів. Описано алгоритм використання карт Кохоненні для виявлення ключових слів для аналізу екстремистської інфоормації

У третій главі наведено опис програмного забезпечення, яке імплементує запропоновану методику.

У підрозділі "Охорона праці" розглянуто правила охорони праці під час експлуатації електронно-обчислювальних машин У підрозділі "Безпека життєдіяльності" описано окремі питання безпеки у виробничих приміщеннях.

ANNOTATION

Development of mathematical and software support for the key words study and highlighting in the problems on extremist information identification in the Internet // Thesis of "Master" Degree// Zhavruk Roman Andriiovych // Ternopil National Technical University named after Ivan Pulyuy, Faculty of Computer Information Systems and software engineering, Department of Cybersecurity, SBm-61 group // Ternopil, 2020 // P. 67, fig. - 12, table. - 4, chair. - 0, added. - 1.

Key words: COCHONEN MAPS, TEXT CLUSTERING, PUBLIC TEXTS, INTERNET.

This master's thesis is devoted to the study of methods of search and selection of keywords to identify extremist information in public texts on the Internet. A study of methods and tools for clustering public texts in order to identify extremist texts.

The use of Kohonen maps has been suggested to determine keywords from public texts.

The paper proposes software that allows frequent analysis of public texts from the Internet in order to detect the detection of extremist information.

The first chapter presents existing approaches to text analysis.

The second chapter provides a comparative analysis of the methods used to analyze public texts. An algorithm for using Kohoneni maps to identify keywords for analyzing extremist information is described

The third chapter describes the software that implements the proposed technique.

In the subsection "Occupational safety" the rules of occupational safety during operation of electronic computers are considered. In the subsection "Safety of life" separate questions of safety in industrial premises are described.

ЗМІСТ

ВСТУП 8
1 ТЕОРЕТИЧНА ЧАСТИНА 10
1.1 Огляд методів аналіз текстів 10
1.2 Огляд методів кластеризації текстової інформації 12
1.3 Висновки до розділу 1 18
2 ДОСЛІДЖЕННЯ АЛГОРИТМУ SOM ЗАСОБІВ ДЛЯ ВИЯВЛЕННЯ КЛЮЧОВИХ СЛІВ У ПУБЛІЧНИХ ТЕКСТАХ 20
2.1 Порівняльна характеристика алгоритмів кластеризації текстової інформації 20
2.2 Якісне введення в самоорганізовані карти 21
2.3 Вихідний покроковий алгоритм SOM 23
2.4 Алгоритм SOM, заснований на скалярному добутку 26
2.5 Налаштування алгоритму 27
2.5.1 Вибір початкового наближення 27
2.5.2 Вибір швидкості навчання мережі 28
2.5.3 Вибір функції сусідства між нейронами 28
2.5.4 Алгоритм роботи карти 29
2.6 Макроструктура алгоритму 29
2.7 Самоорганізовані карти для символьних рядків 30
2.7.1 Ініціалізація SOM для рядків 31
2.7.2 Пакетний варіант самоорганізуючої карти для рядків 32
2.7.3 Виявлення переможця в ситуації нерозрізненості порівнюваних рядків 33
2.8 Висновки до розділу 2 34
3 ПРАКТИЧНА ЧАСТИНА. проектування програмного забезпечення для аналізу публічних текстів 35
3.1 Проектування програмного забезпечення для аналізу публічних текстів 35
3.1.1 Функціональні вимоги 35
3.1.2 Нефункціональні вимоги 36
3.2 Розробка системної програмної архітектури 37
3.2.1 Вибір цільового варіанту архітектури ПЗ 38
3.3 Діаграма компонент 40
3.4 Вибір інструментальних програмних засобів та інформаційних технологій 40
3.3 Висновки до розділу 3 44

4 приклад застосування розробленого програмного забезпечення 46
4.1 Особливості реалізації програмного забезпечення 46
4.2 Аналіз результатів 50
4.3 Висновки до розділу 4 52
5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ 53
5.1 Охорона праці 53
5.2 Безпека в надзвичайних ситуаціях 56
5.2.1 Техніка безпеки 56
5.2.2 Пожежна безпека 58
ВИСНОВКИ 61
СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ 62
ДОДАТКИ

ВСТУП

В наш час тероризм є однією з найстрашніших загроз, тому що не зрозуміло хто ворог, звідки і коли буде нанесено удар, страждають невинні люди. Як показує практика боротьба з тероризмом – важка задача. Проблема тероризму є дуже актуальною в нашій країні, більше того судячи зі статистики за останні роки зріст кількості злочинів, які носять терористичний характер лише зростають. Однак за своїми масштабами, наслідками та своєю жорстокістю, тероризм перетворився сьогодні в одну з найстрашніших проблем всього людства.

Так як ми живемо в епоху інформаційних технологій в мережі все більше виникає інформації з приводу виникнення різної терористичної та екстремістської діяльності або ж потенційної загрози безпеки. Зазвичай дана інформація розміщується на різних публічних джерелах таких як новинні розсилки або ж канали новинних видавництв. Такі новини або коментарі до них також можуть в тому чи іншому вигляді виступати за пропаганду терористичної діяльності та переслідувати схожі незаконні цілі.

Тому виникає проблема як мінімізувати або зовсім уникнути виникнення потенційної загрози.

Для вирішення даної проблеми пропонується розробити програмне рішення за допомогою методів аналізу текстів, яке буде проводити аналіз публічних новинних каналів, каналів видавництв та коментарів користувачів на предмет виявлення повідомлень які несуть у собі терористичні загрози.

Основна ідея – правильно скласти словник, який буде використовуватись для виявлення потенційної загрози. Далі виділити потрібний масив текстів та провестио їх аналіз, в результаті цього можна виявити людей які схильні до участі в терористичній діяльності. В даній дипломній роботі будуть розглянуті основні методи аналізу текстів та кластеризації текстової інформації. Буде обрано методи, для вірішення поставленої проблеми та розроблене ПЗ для аналізу публічних текстових повідомлень в мережі Інтернет.

Метою даної роботи є зменшення витрат часу на аналіз публічних текстів з мережі інтернет з метою виявлення ключових слів екстремистської спрямованності на основі розробки відповідного математичного та програмного забезпечення.

Для досягнення поставленої мети необхідно вирішити наступні *завдання*:

1. Проаналізувати предметну область.
2. Проаналізувати основні методи аналізу тексту.
3. Проаналізувати методи кластеризації текстової інформації.
4. Обрати метод для дослідження.
5. Розробити архітектуру розроблюваного ПЗ.
6. Розробити основний пакет діаграм.
7. Розробити програмні компоненти для аналізу публічних текстових повідомлень в мережі Інтернет.
8. Протестувати розроблене програмне рішення.

Об'єктом досліджень є загроз в середовищі безпровідних мереж.

Предметом дослідження є методи та підходи до кластеризації текстів та виявлення ключових слів з публічних текстів мережі інтернет.

Наукова новизна та практична цінність роботи: запропоноване програмне забезпечення, що реалізує алгоритм карт Кохонена для виявлення екстремістської інформації в публічних текстах. Запропонований метод дозволяє проводити аналіз публічних текстів та виконувати їх класифікацію щодо приналежності до екстремістської інформації.

Апробація результатів роботи. Окремі результати роботи доповідались на VIII науково-технічній конференції «Інформаційні моделі, системи та технології», Тернопіль, ТНТУ, 9 – 10 грудня 2020 р.

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Огляд методів аналіз текстів

Інтент-аналіз – метод, що дозволяє реконструювати інтенції (суб'єктивна спрямованість на певний об'єкт) автора по його тексту, оскільки для виявлення і кваліфікації інтенцій опора на окремі слова і пропозиції малопродуктивна. Експертне виявлення і ідентифікація мовних інтенцій надає можливість окреслити їх коло в текстах різної тематики і спрямованості, тобто охарактеризувати їх якісно, тому дослідницька задача у тих хто використовує метод інтент-аналізу полягає в експертному (тобто по суті суб'єктивному) оцінюванні характеру інтенцій, їх розмитості і неясності розуміння [1].

Оскільки метод інтент-аналізу заснований на експертній оцінці тексту, його повна автоматизація не здійснена, однак, існують методики, де автоматизовані перші етапи інтент- аналізу, що дозволяють здійснювати операції виділення тем, кодування і пошуку [2].

Контент-аналіз – найпоширеніший метод, який має безліч варіацій в різних методиках, що дозволяє провести якісно-кількісний аналіз змісту текстових масивів з метою подальшої інтерпретації виявлених числових закономірностей. Полягає в оцінці частотного розподілу слів, словосполучень словоформ та інших одиниць аналізу (число їх варіацій теоретично безмежно) щодо тексту. Результатом є частота, відносна і питома вага, ймовірність виявлення та ін., на основі чого робиться якісний або кількісний висновок в залежності від висунутої гіпотези [2].

Фоносемантичний аналіз тексту або слова полягає в оцінці його звучання безвідносно до його змісту. Полягає в зіставленні системи сполучень фонем в конкретному тексті або слові з їх стандартизованими оцінками по ряду біполярних шкал. Результатом фоносемантичного аналізу є профіль вираженості оціночних шкал в стандартизованому семантичному просторі, на підставі якого робиться висновок про можливий ефект впливу тексту на читача [2].

Дискурс-аналіз або дискурсивний аналіз – сукупність методик і технік інтерпретації текстів чи висловлювань як продуктів мовленнєвої діяльності, здійснюваної в конкретних суспільно-політичних обставинах і культурно-історичних умовах. Цей метод орієнтований, перш за все, на вивчення лінгвістичного рівня в структурі соціальної комунікації як домінуючого протягом певного історичного періоду розвитку суспільства і культури [3].

Дискурс-аналіз дозволяє виділити не тільки істотні характеристики соціальної комунікації, а й другорядні, змістовні і формальні показники (наприклад, тенденції в варіативності мовних формул або побудові висловлювань). Дискурс-аналіз широко застосовується в соціологічних і політичних дослідженнях і частково реалізований в таких програмах, як CATPAC. Це методика аналізу тексту, написаного на будь-якій мові, заснована на системі Galileo, яка представляє собою комплекс теорій і методів, спрямованих на наукове вивчення когнітивних і культурних процесів. CATPAC дозволяє виявляти основні ідеї тексту без попереднього кодування і лінгвістичного аналізу [1].

Наративний аналіз – це метод узагальнення минулого досвіду за допомогою співвіднесення послідовності слів у реченні і послідовності реальних (як передбачається) подій. Дозволяє здійснювати кількісну оцінку тексту. На відміну від контент-аналізу, який може бути застосований до будь-яких текстів, наративний аналіз орієнтований на особливі тексти, що містять розповідь [4].

Перевагою наративного аналізу, в порівнянні з кластерним, є те, що оцінка проводиться за конкретними категоріями (Суб'єкт, Дія, Об'єкт), а не по довільно обраним дослідником виходячи з його завдань. В клас наративних текстів входять різноманітні історії від різноманітних художніх та історичних текстів (міфи, легенди, літописи та ін.) До статей газет, в яких описуються події, що відбулися [1].

Морфологічний аналіз спрямований на визначення безлічі морфологічних інтерпретацій кожного з слів тексту, що складається з таких параметрів, як лема, морфологічна частина мови; набір загальних граммем; безліч наборів граммем.

Морфологічний аналіз реалізований в більшості методик, так як є основою для інших видів аналізу тексту [5].

Синтаксичний аналіз – це метод зіставлення лінійної послідовності лексем мови з його формальною граматикою. Результатом аналізу стає синтаксична структура пропозиції, яка подається у вигляді дерева залежностей. Результати синтаксичного аналізу важливі для подальших етапів роботи з текстом [2].

Семантичний аналіз – метод, спрямований на побудову семантичної структури пропозиції, що складається з семантичних вузлів і семантичних відносин. Метою проведення аналізу є побудова цих вузлів, які утворюються із слів вихідного речення [5].

1.2 Огляд методів кластеризації текстової інформації

Метод Custom Search Folders. Ця технологія використана в пошуковому сервері NorthernLight. Суть її полягає в тому, що користувач може звузити результат пошуку, і розглядати об'єкти, розподілені по папках – folders. Вибором однієї із запропонованих папок користувач звужує діапазон аналізованих об'єктів. Об'єктами в даному випадку є HTML посилання. Папки мають ієрархічну структуру, що дає можливість все більше і більше звужувати результат пошуку [6].

LSA/LSI – як метод виявлення латентних зв'язків – відомий давно і застосовується в різних сферах науки [7-9]. В основі методу лежать принципи факторного аналізу, зокрема, виявлення латентної структури досліджуваних явищ або об'єктів.

Переваги методу:

- використовує інформацію матриці tf-idf;
- метод не потребує попереднього налаштування на специфічний набір документів, його не треба навчати;
- кращий метод для виявлення латентних залежностей.

Недоліки методу:

– велика кількість обчислень може призводити до того, що на результатах запитів, що містять сотні тисяч об'єктів система буде працювати дуже довго . Як стверджують автори [10], швидкість обчислення SVD відповідає порядку $N^2 \times k$, де $N = N_{docs} + N_{terms}$, k – розмірність простору чинників;

– відсутність відповідної назви для отриманих факторів. В описується один з алгоритмів пошуку відповідних назв, але це вимагає додаткових обчислювальних витрат;

– кластери не перетинаються.

Метод Suffix Tree Clustering. Спочатку суффіксні дерева – suffix trees – були розроблені і застосовувалися для швидкого пошуку підрядків. Суфіксне дерево – дерево, що містить всі суфікси даного рядка. Воно складається з вершин, гілок і додаткових вказівників, які називаються suffix pointers, за допомогою яких домагаються лінійної швидкості побудови дерева [11]. Гілки дерева позначаються буквами або буквосполученнями, які є частинами суфіксів рядка. Суфікс, відповідний певній вершині, можна отримати шляхом об'єднання всіх букв, які знаходяться на ребрах дерева, починаючи від кореневої вершини і закінчуючи даної [11].

Переваги методу:

– висока швидкість роботи. За часом і займаної пам'яті дерево будується пропорційно кількості документів. Найгірша теоретична верхня межа часу побудови – пропорційно квадрату кількості документів;

– хороша наочність представлення результатів. Загальні фрагменти текстів і фраз виступають в якості назви кластерів, – це має великий сенс, тому що не треба витрачати додаткових зусиль для визначення відповідного імені;

– алгоритм не має потреби в навчанні та завданні порогу спрацьовування;

– алгоритм інкрементний і допускає перетинаємість областей видимості кластерів;

– якщо є вже побудований індекс по корпусу документів, то можна зробити «арифметизацію» якщо всі тексти документів замінити на якесь числове уявлення слів, де кожне число посилається на вхід в словнику (або тезаурус), то

порівнювати доведеться не слова, а числа, і препроцессінг текстів стає не потрібним.

Недоліки методу:

– необхідність повторної обробки текстів документів;

– не використовується вже наявна інформація про значення близькості документів або значних td-idf;

– не виявляється прихована семантика серед документів, яка може бути присутня не тільки на текстовому рівні;

– проблеми синонімії і омонімії.

Методи Single Link, Complete Link, Group Average. Одні з найстаріших алгоритмів кластеризації даних. Особливістю цих методів, є те, що вони розбивають документи на кластери шляхом розбиття їх на ієрархічні групи, тобто отримується безліч кластерів які мають ієрархічну структуру. Вони називаються ще методами ієрархічної агломеративної кластеризації. Принцип роботи ієрархічних агломеративних процедур полягає в послідовному об'єднанні груп елементів, спочатку найближчих, а потім все більш віддалених один від одного [12].

Швидкість роботи алгоритмів Single Link і Group Average – O(n^2), а в Complete Link – O (n^3) [13], де n – кількість документів. Кількість займаної пам'яті алгоритмом Single Link – O (n).

Переваги методів:

– алгоритми не мають потреби в навчанні;

– використання матриці близькості між документами;

– алгоритми інкрементні.

Недоліки методів:

– необхідно задавати поріг – максимальна кількість документів в кластері;

– для отримання хороших результатів кластеризації значення близькості між парами документів повинні приходити в певному порядку, тобто робота алгоритму лише детермінована;

– кластери не перетинаються.

Scatter/Gather – це метод представлення результатів запитів користувачеві [14]. Спочатку система розбиває документи на невелике число груп – фаза scattering. Ґрунтуючись на коротких описах груп, користувач вибирає одну або кілька груп для подальшого розгляду. Документи об'єднуються – фаза gathering і розглядаються, як одна група, і процес повторюється вже над нею. Це схоже на послідовність штучних запитів відносно основних категорій.

На фазі розбиття метод може використовувати два алгоритму [14]: Buckshot і Fractionation. Алгоритм Buckshot швидший і підходить для швидкої рекластерізації при виконанні ітерацій в Scatter/Gather. Fractionation ж є більш точним і більш повільним алгоритмом і використовується в Scatter/Gather для попереднього розбиття на групи безлічі документів і виконується в режимі off-line.

Переваги методу:

– розбиття на кластери за допомогою алгоритму Buckshot має високу швидкість, яка лінійна по відношенню до числа документів. У той же час Fractionation володіє хорошою точністю визначення центроїду кластерів;

– хороша наочність представлення даних [14];

– алгоритм не має потреби в навчанні;

– використовується матриця близькості документів.

Недоліки методу:

– висока швидкість і точність не поєднуються в якомусь одному алгоритмі розбиття на кластери;

– потрібно завдання кількості кластерів, на які буде розбиватися безліч документів;

– метод не інкрементний;

– немає можливості отримувати пересічні кластери, тому що документ поміщається в найближчий кластер.

Метод K-means. В основі K-means лежить ітеративний процес стабілізування центроїду кластерів. Основною характеристикою кластера є його

центроїд і вся робота алгоритму спрямована на стабілізування або, в кращому випадку, повне припинення зміни центроїда кластера.

Переваги методу:

– лінійна швидкість роботи;

– використовує значення матриці tf-idf;

– метод не має потреби в навчанні та при необхідності може накопичувати відомості для подальшого збільшення точності роботи - використання Байєсовських оцінок параметрів кластеризації.

Недоліки методу:

– потрібно завдання кількості кластерів, як мінімум на початкових етапах – до використання апріорної інформації;

– в тому випадку, коли центроїди кластерів вибираються випадковим чином, результати, одержувані над однією і тією ж вибіркою документів, будуть відрізнятися. Це може відбуватися через незадовільну роботу генератора випадкових чисел і внаслідок рівномірного розподілу документів в просторі - без явних областей згущення;

– алгоритм не інкрементний;

– кластери не перетинаються.

Метод Concept Indexing. Цей метод використовується для зменшення розмірності простору ознак (див. LSA / LSI) [17]. У просторі ознак, розмірність якого зменшена, виконується стандартне відображення множини документів.

Основна відмінність методу від інших полягає в тому, що алгоритм може бути навчаним або нездатним до навчання.

Переваги методу:

– алгоритм рекурсивної бісекції має високу швидкість роботи – вище лінійної (O (N'logk)), де N – число документів, k – число кластерів;

– кількість документів, що належать класу d в «зменшеному» просторі ознак розмірності r;

– кількість документів, що належать класу d в оригінальному просторі ознак розмірності;

– використовує значення матриці tf-idf;

– в разі, коли не використовується рекурсивна бісекція для знаходження кластерів, алгоритм інкрементний.

Недоліки методу:

– один з варіантів методу – навчаний. Для автоматичних систем це, безумовно, недолік, однак навчання помітно покращує якість роботи;

– потрібно завдання кількості кластерів, на які буде розбиватися множина документів;

– в разі рекурсивної бісекції алгоритм не інкрементний, тому що відноситься до класу top-down, де спочатку розглядається вся колекція документів [17].

Метод SOM. По суті своїй SOM – це нейронна мережа Кохонена, що виконує завдання класифікації вхідних даних і навчається без вчителя (unsupervised learning) [18]. SOM – метод, розроблений для відображення багатовимірних даних на двовимірну площину - ще один спосіб візуалізації даних.

Мережа Кохонена навчається без учителя. Це означає, що на вхід надходять навчальні дані, і відбувається корекція синаптичних ваг нейронів. Зміни внутрішньої структури мережі не відбувається, тому необхідно заздалегідь знати її внутрішню структуру і кількість передбачуваних класів. Для навчання мережі алгоритм Кохонена використовує інформацію про попередній крок навчання, тому не можна чітко говорити про швидкість навчання такої мережі. Швидкість навчання мережі сильно залежить від порядку надходження навчальних даних на вхід мережі. Для будь-якої нейронної мережі є тестовий набір, який застосовується до мережі, для контролю за процесом навчання, це необхідно для того, щоб не сталося нестачі або перенасичення в процесі навчання [19].

Переваги методу:

– зручність представлення результатів кластеризації;

– метод простий у навчанні;

– при побудованій мережі робота буде проводитися досить швидко. Час порівняння визначається швидкістю проходу вхідного сигналу по мережі, тобто швидкості обробки вхідного вектора документа внутрішньою структурою мережі;

– в якості вхідних даних для мережі використовуються відстані між документами, тобто значення матриці близькості;

– процес побудови мережі інкрементний, але має значення порядок подачі навчальних сигналів;

– кластери перетинаються.

Недоліки методу:

– процес навчання мережі є центральною ідеєю будь-якої нейронної мережі. Від правильно організованого процесу залежить якість роботи мережі. Найчастіше більша увага приділяється саме йому: швидкість навчання, спосіб навчання. Однак в автоматичних системах повернення і групування інформації, необхідність навчати мережу для кожної нової множини результатів представляється неможливим. Можливий варіант, коли мережа будується один раз для всіх можливих результатів, однак це можливо для обмеженого і відносно невеликої кількості об'єктів. Мережа в цьому випадку буде будуватися дуже довго і буде дуже великою [18];

– як правило, процес навчання тривалий і вимагає фіксування числа класів і набору навчальних даних.

1.3 Висновки до розділу 1

У сучасному інформаційному просторі в умовах постійного нарощування обсягів текстів проблема продуктивного пошуку і швидкого орієнтування в масі текстів стоїть особливо гостро.

В даний час навіть фахівцеві досить вузької предметної області складно знайти необхідну інформацію. Одне з актуальних завдань в зв'язку з цим - вдосконалення нових методів і підходів до автоматичного екстрагуванню ключових слів, пошук моделей виділення ключових слів із публічних текстів з

метою визначення екстремистської направленості. Отже, якісне уявлення змісту тексту у вигляді мінімального набору лексичних одиниць є актуальною проблемою, розв'язуваної методами статистики, лінгвістики, автоматичного розпізнавання і реферування текстів Нейромережеві методи, до вирішення задачі виявлення ключових слів, почали застосовуватися порівняно недавно і використовують технології штучних нейронних мереж і їх можливість виділення і узагальнення прихованих залежностей стосовно до вхідних і вихідних даних. Нейромережеві методи можна виділяти як окремий клас методів серед виділення ключових слів, а можна віднести до класу гібридних методів.

2 ДОСЛІДЖЕННЯ АЛГОРИТМУ SOM ЗАСОБІВ ДЛЯ ВИЯВЛЕННЯ КЛЮЧОВИХ СЛІВ У ПУБЛІЧНИХ ТЕКСТАХ

2.1 Порівняльна характеристика алгоритмів кластеризації текстової інформації

В результати огляду методів кластеризації текстової інформації, які були розглянуті в 1 розділі дипломної роботи була зроблена порівняльна характеристика кожного методу, яка показана у таблиці 2.1

Таблиця 2.1 – Основні характеристики алгоритмів

Алгоритм кластеризації	Вид методу	Обмеження	Перетинаємість кластерів	Інкрементність алгоритму	Використовувані числові характеристики документів	Попереднє навчання	Швидкість роботи
LSI	Числовий, кластеризуємий	Кількість кластерів	-	+	tfidf	-	N^2xk, N=terms+docs, k-factors
STC	Нечисловий, кластеризуємий	Нема обмежень	+	+	-	-	$O(k^2N)$ N-число документів, k-число кластерів
Single Link, Complete Link, Group Average	Числовий, кластеризуємий, bottom-up	Кількість документів в кластері	-	+	Similarity matrix	-	Single Link ~ $O(N^2)$ Complete Link ~ $O(N^3)$, Group Average ~ $O(N^2)$

Продовження таблиці 2.1

Scatter/Gather	Числовий, кластеризуємий, bottom-up	Кількість кластерів	-	-	Similarity matrix	-	Buckshot ~ O(kN), Fractionation ~ O(mN), m=O(k), k-число кластерів
K-means	Числовий, кластеризуємий	Кількість кластерів, центроїди	-	-	tfidf	-	O(n)
CI – ненавчаємий варіант	Числовий, кластеризуємий, top-down	Кількість кластерів	-	-	Similarity matrix	-	O(N*log k), k-число кластерів
CI - навчємий варіант	Числовий, кластеризуємий	Кількість кластерів	-	-	Similarity matrix або tfidf	+	?
SOM	Числовий, кластеризуємий	Кількість кластерів	+	+	Similarity matrix або tfidf	+	?

В результаті порівняння алгоритмів було обрано для реалізації метод SOM він має ряд переваг над іншими алгоритмами, єдиним суттевим недоліком є довга швидкість навчання мережі.

2.2 Якісне введення в самоорганізовані карти

Самоорганізована карта Кохонена (SOM) представляє з себе обчислювальний метод, призначений для завдань, в першу чергу, кластеризації та візуалізації, а також аналізу даних з просторів високої розмірності (інакше, багатовимірних даних), отриманих експериментально. Метод був запропонований Туево Кохоненом (1982). Прабатьками моделі самоорганізованої мережі Кохонена були ранні нейромережеві моделі (зокрема, модель асоціативної пам'яті і модель адаптивного навчання) [18].

Метою застосування даного методу є пошук прихованих закономірностей в даних, грунтуючись на зниженні розмірності вихідного простору в простір меншої розмірності (на практиці найчастіше використовується двовимірне, з причини, зокрема, зручною візуалізації). При цьому топологія вихідного

простору залишається тією ж самою. В результаті навчання даної моделі виходить решітка, що складається з навчених нейронів, вона ж і називається "картою" вихідного простору.

Архітектура самоорганізуючої карти Кохонена наступна: є два шари - вхідний шар (розподільний) нейрони і вихідний шар (шар Кохонена) нейронів, при цьому нейрони другого шару розташовані у вигляді двовимірної решітки (зазвичай сітка або квадратна, або шестикутна, про це далі), так , що кожен нейрон з першого шару з'єднаний з кожним нейроном другого шару карти Кохонена наступна: є два шари - вхідний шар (розподільний) нейрони і вихідний шар (шар Кохонена) нейронів, при цьому нейрони другого шару розташовані у вигляді двовимірної решітки (зазвичай сітка або квадратна, або шестикутна, про це далі), так , що кожен нейрон з першого шару з'єднаний з кожним нейроном другого шару, що зображено на рисунку 2.1.

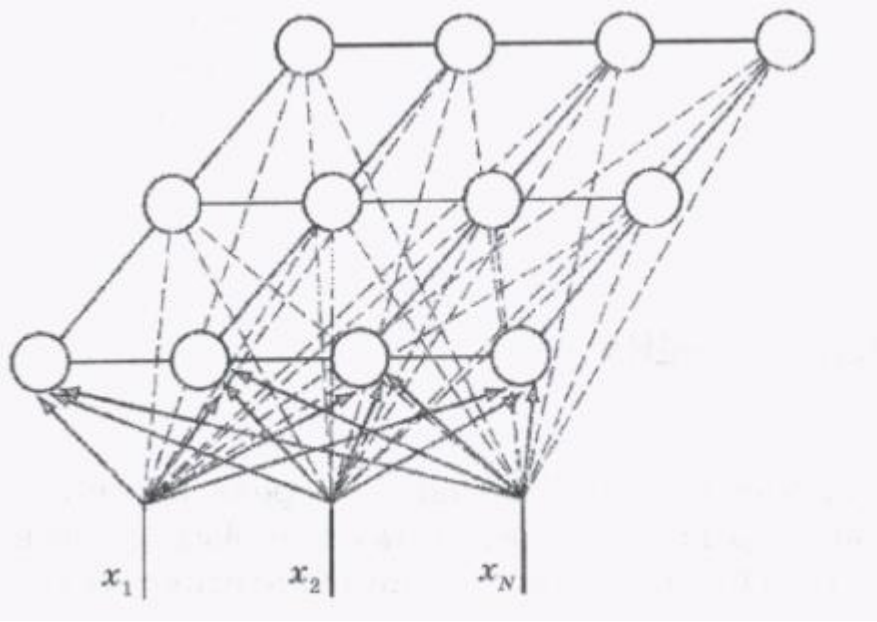

Рисунок 2.1 – Загальна структура карти Кохонена

Мережа, із запропонованою архітектурою, навчається на навчальній вибірці, навчання полягає в коригуванні ваг нейронів в шарі Кохонена. Після

того, як мережа навчена вона починає свою роботу, відносячи кожен нейрон з тестової вибірки до того чи іншого кластеру.

2.3 Вихідний покроковий алгоритм SOM

Приступимо до розгляду теорії самоорганізуючих карт, використовуючи вихідний алгоритм SOM як відправну точку. Цей алгоритм, як можна бачити, визначає регресійний рекурсивний процес спеціального виду, в якому на кожному кроці здійснюється обробка тільки частини моделей.

На рисунку 2.2 зображено масив вузлів в двовимірній решітці SOM.

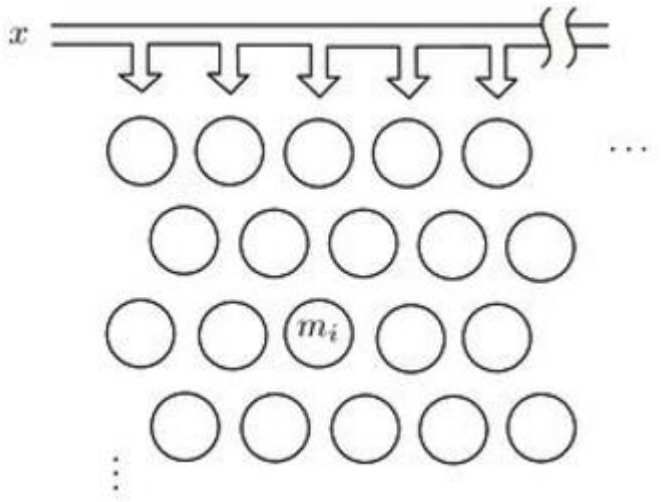

Рисунок 2.2 – Масив вузлів в двовимірній решітці SOM.

Алгоритм SOM визначає тут відображення вхідного простору даних R^n на двовимірну решітку вузлів. Кожному вузлу і ставиться у відповідність параметричний вектор моделі, званий також опорним вектором $m_i = [\mu_{i1}, \mu_{i2} \dots, \mu\ ] \in R^n$

Перед початком рекурсивного процесу всі вектори m_i повинні бути ініційовані. З цією метою будемо поки брати в якості їх компонент випадкові числа m_i. Це дає можливість продемонструвати той факт, що при старті з довільного початкового стану вектори m_i розподіляться впорядковано на двовимірній решітці, якщо число кроків процесу досить велике. У такій поведінці векторів m_i і полягає основний ефект самоорганізації.

Масив вузлів може утворювати решітку прямокутного, гексагонального і навіть нерегулярного типу. Гексагональна решітка ефективна для вирішення завдань візуального представлення даних. У найпростішому випадку вхідний вектор$x = [\xi_1, \xi_2, \dots, \xi_n]^T \in R^n$ пов'язаний одночасно з усіма нейронами через змінні скалярні ваги μ_{ij} які в загальному випадку різні для різних нейронів. В абстрактних термінах це можна уявити так. як якщо б вхідний вектор x за допомогою деяких паралельних обчислювальних механізмів порівнювався з усіма векторами m_i а шукане місце кращої відповідності в деякій метриці визначалося розташуванням отриманого «відгуку» [18].

Нехай $x \in R^n$ – випадковий вектор даних. Можна сказати, що SOM являє собою «нелінійну проекцію» щільності розподілу ймовірностей *p(x)* багатовимірного вхідного вектора даних x на двовимірний дисплей. Вектор *x* можна порівняти з усіма m_i в будь-якій метриці. У багатьох практичних додатках знаходження найбільш підходящого вузла можна виконати шляхом обчислення найменшої евклідової відстані $\|x - m_i\|$. Тогда индекс этого узла определяется выражением

$$c = arg\min_i\{\vee a - m_i \vee\},$$

що означає те ж саме, що і вираз

$$\|x - m_c\| = \min_i\{\|x - m_i\|\}$$

У процесі навчання або процесу, в якому формується «нелінійна проекція», ті вузли, які топографічно близькі в решітці в межах деякої геометричної відстані, будуть активувати один одного, навчаючись в певній мірі за рахунок цього на одному і тому ж вході x. Це веде до локальної релаксації або ефекту згладжування для вагових векторів нейронів в розглянутій околиці, що при тривалому навчанні призводить до глобального впорядкування. Розглянемо можливі межі збіжності для наступного навчального процесу:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)],$$

де t – 0,1,2,… цілочисельна величина, яка відіграє роль дискретного часу, а початкові значення $m_i(0)$ можуть бути довільними, зокрема, випадковими. У релаксаційному процесі функція $h_{ci}(t)$ відіграє центральну роль: вона діє як так

звана функція сусідства, згладжує ядро, визначене на точках решітки. Для збіжності необхідно, щоб виконувалася умова $h_{ci}(t) \to 0$ при $t \to \infty$.Зазвичай має місце співвідношення

$$h_{ci}(t) = h(\|r_c - r_i\|, t),$$

де $r_c \in R^2$та $r_i \in R^2$ – вектори, що визначають розміщення вузлів с та і, відповідно, в даній решітці. Із зростанням $\|r_c - r_i\|$ виконується умова $h_{ci} \to 0$. Середня ширина і форма функції $h_{ci}(t)$ визначають «жорсткість» тієї «еластичною поверхні», яка підганяється так, щоб найкращим чином відповідати оброблюваним даним.

У літературі часто зустрічаються дві простих реалізації для $h_{ci}(t)$. Більш проста з них визначає околиці вузла с, тобто. множина точок, що є сусідами з цим вузлом. Позначимо через N_c множину індексів цих точок (зауважимо, що можна визначити множину індексів як функцію часу $N_c = N_c(t)$, тоді $h_{ci}(t) = \alpha(t)$, якщо $i \in N_c$ та $h_{ci}(t) = 0$, якщо $i \notin N_c$. Значення $\alpha(t)$, визначається величиною коефіцієнта швидкості навчання <1). Як $\alpha(t)$ так і радіус множини $N_c(t)$ зазвичай монотонно зменшуються з плином часу (в ході процесу упорядкування).

Інший широко використовуваний варіант згладжуючого ядра може бути записаний в термінах функції Гаусса:

$$h_{ci}(t) = \alpha(t)e^{-\|r_c - r_i\|^2/[2\sigma(t)]},$$

Де $\alpha(t)$ – інший скалярний «коефіцієнт швидкості навчання», а параметр $\sigma(t)$ визначає ширину ядра, яка відповідає згаданому вище радіусу множини $N_c(t)$. Як $\alpha(t)$ так $\sigma(t)$ є монотонно спадними функціями часу.

Алгоритм, обраний тут для цілей попереднього моделювання, є тільки одним з багатьох можливих варіантів. Якщо мережа SOM не дуже велика (наприклад, в ній не більше декількох сотень вузлів), вибір параметрів процесу не дуже критичний. Можна використовувати також введене вище просте визначення функції $h_{ci}(t)$ як множини-околі [18].

Слід звернути особливу увагу на вибір розміру множини $N_c = N_c(t)$. Якщо початкова околиця занадто мала, не вдасться отримати глобально впорядковану

карту. Замість цього для даної карти будуть спостерігатися різного виду розбиття мозаїчного характеру, між якими напрямок упорядкування змінюється стрибкоподібно. Цього явища можна уникнути, якщо почати процес з досить широкою околиці $N_c = N_c(0)$ дозволяючи їй стискатися з часом. Початковий радіус даної околиці N_c може бути навіть більше, ніж половина діаметра мережі! Протягом перших 1000 кроків або близько того, коли має місце достатня впорядкованість, а значення $\alpha = \alpha(t)$ досить велике, радіус N_c може зменшуватися лінійно, наприклад, до одиниці. На етапі уточнення формованого відображення множина N_c може все ще містити найближчих сусідів вузла c.

Якщо початкові значення обрані випадковим чином, то приблизно протягом перших 1000 кроків величина $\alpha(t)$ повинна мати досить велике значення (близьке до одиниці), яке в подальшому буде монотонно зменшуватися. Вид залежності цієї величини від часу особливого значення не має: функція $\alpha = \alpha(t)$ може бути лінійною, експоненціальною або зворотно пропорційною t.

Впорядкування векторів m_i відбувається протягом початкового періоду роботи алгоритму, а решта кроків потрібні лише для точного підстроювання карти.

Після завершення етапу впорядкування функція $\alpha = \alpha(t)$ повинна приймати мінімальні значення (наприклад, порядку 0.02 або менше), але протягом значного числа кроків. На кінцевому етапі роботи алгоритму некритично, чи буде значення $\alpha(t)$ зменшуватися за лінійним або за експоненціальним законом. Однак для дуже великих карт може виявитися важливим мінімізувати повний час навчання. У цьому випадку вибір оптимального закону $\alpha(t)$може стати істотним.

Ефективний вибір цих функцій і їх параметрів виконується поки здебільшого експериментально [18] .

2.4 Алгоритм SOM, заснований на скалярному добутку

У ряді випадків величину x пропонується нормувати перед тим, як вона буде подана на вхід розглянутого алгоритму. Нормування, взагалі кажучи, не є необхідною, але може поліпшити точність розрахунків, оскільки в цьому випадку результуючі еталонні вектори мають один і той же динамічний діапазон зміни своїх значень.

Інший аспект полягає в тому. що при порівнянні векторів можна використовувати велику кількість різноманітних метрик. При цьому, однак закони порівняння і модифікації векторів повинні бути взаємно сумісними щодо однієї і тієї ж метрики.

Якщо як міри схожості векторів x та m_i застосовується величина їх скалярного добутку, рівняння, що описують процес навчання, набирають вигляду

$$x^T(t)m_c = \max_i\{x^T(t)m_i(t)\},$$

$$m_i(t+1) = \begin{cases} \dfrac{m_i(t) + \alpha\,(t)x(t)}{\|m_i(t) + \alpha\,(t)x(t)\|} \text{ при} i \in N_c(t), \\ m_i(t) \text{при} i \notin N_c(t), \end{cases}$$

де $0 < \alpha\,(t) < \infty$; наприклад, можна прийняти $\alpha\,(t) = 100/t$ [18]. Цей процес автоматично нормує еталонні вектори на кожному кроці. Обчислення, необхідні для проведення нормування, уповільнюють навчальний алгоритм.

Але з іншого боку, показник на основі скалярного добутку. застосовуваний для зіставлення векторів, дуже простий, швидко працює, а також легко реалізується за допомогою аналогових обчислень, як електронних, так і оптичних. Тут проглядається також зв'язок з фізіологічними процесами.

Крім показників, заснованих на евклідовій метриці і скалярному добутку, алгоритм SOM допускає використання і інших варіантів.

2.5 Налаштування алгоритму

2.5.1 Вибір початкового наближення

По-перше, ініціалізацію можна проводити об'єктами з навчальної вибірки, тобто обравши для кожного вагового вектора деякий елемент із навчальний

вибірки і встановити значення вагових компонент рівними компонентам обраного вектора. Плюсом такого підходу є те, що вектора ваг спочатку знаходяться в тому регіоні, де і об'єкти з вибірки, тобто потрібно менше ітерацій на етапі налаштування.

По-друге, ініціалізацію ваг можна проводити якимись випадковими значеннями. Наприклад можна взяти не близького розподілу, нехай рівномірний або нормальний, і для кожної згенерувати псевдовипадкове число з обраного розподілу, тобто $w_i^{(j)}\ P$, де P – це заданий розподіл. Плюсом даного підходу є простота однак, "налаштування" в даному випадку може бути довгим.

По-третє, можна використовувати лінійну ініціалізацію, тобто вагові вектора мають значення векторів з підпростору, натягнутого на два головних власних вектора простору вихідного набору даних, при цьому впорядкованим чином. Це найбільш практичний спосіб.

2.5.2 Вибір швидкості навчання мережі

Функцію швидкості навчання мережі $\alpha(t)$ можна вибрати, наприклад, наступним чином:

Найпростіший спосіб:

$$\alpha(t) := \frac{A}{B+t}, A = const, B = const$$

Експоненціальна швидкість навчання:

$$\alpha(t) \coloneqq \alpha_0 e^{\frac{-t}{}}, \in (0,1), = const, \alpha_0 = const$$

Показова швидкість навчання:

$$\alpha(t) \coloneqq \quad ^{\frac{t}{T}}, \in (0,1), = const$$

2.5.3 Вибір функції сусідства між нейронами

Зазвичай вважають $h_{cj}(t) \coloneqq \alpha(t)h(d,t)$, де $d = \left\| ne_c - ne_j \right\|, c$ – індекс нейрона переможця, при цьому:

$$h(d,t) = \begin{cases} const, d < \sigma(t) \\ 0, d > \sigma(t) \end{cases}$$

$h(d,t) = e^{\frac{-d^2}{2(\sigma(t)^2)}}$, де $\sigma(t)$ – це деякий співмножник, що зменшує кількість сусідів, які будуть піддані коректуванню ваг зі збільшенням ітерацій, ця функція монотонно убуває.

Функція $\sigma(t)$ також є параметром алгоритму. Можлива функціональна залежність для неї:

$$\sigma(t) = \frac{\sigma_0}{1 + \frac{t}{T}}$$

Варто відзначити, що є багато інших евристичних заходів подібностей які привласнюють ненульові ваги всередині деякого радіуса, це дозволяє робити обчислення не для всіх сусідів.

2.5.4 Алгоритм роботи карти

Крок 0. Поки є вектора в тестовій вибірці витягти (послідовно) $\vec{x}^{(i)}, K := K - 1, \Phi = \Phi\{\vec{x}^{(i)}$, в іншому випадку алгоритм завершує свою роботу.

Крок 1. Для $\vec{x}^{(i)}$ і $\vec{w}^{(j)} \forall j = 1, \dots, M$ знайти $p\left(\vec{x}^{(i)}, \vec{w}^{(j)}\right) = \left\|\vec{x}^{(i)} - \vec{w}^{(j)}\right\|^2$

Крок 2. Знаходимо нейрон-переможець $ne_c, c = arg \min_{\forall j \in \{1,..L\}} p\left(\vec{x}^{(i)} - \vec{w}^{(j)}\right)$, який лежить ближче до поточного об'єкту $\vec{x}^{(i)}$по даній метриці і відносимо $\vec{x}^{(i)}$ до кластеру, відповідного нейрона ne_c

Крок 3. Перехід до кроку 0.

2.6 Макроструктура алгоритму

В алгоритмі навчання можна виділити 4 макрофази:

— Початкова ініціалізація всіх ваг: побудова матриці $W(0) \in R^{N*M}$ стовпцями якої є вагові вектора всіх нейронів.

— Виконання наступних фаз T разів:

— Обчислення вектора відстаней $D \in R^M$, в якому зберігаються вектора відстаней від вектора *X* до векторів матриці W(t).

— Знаходження індексу мінімуму в векторі D – індексу нейрона переможця.

— Коригування всіх ваг: отримання матриці $W(t+1)$.

В алгоритмі роботи алгоритму можна виділити дві макрофази:

— Виконання наступних фаз K раз:

— Обчислення вектора відстаней $D \in R^M$, в якому зберігаються вектора відстаней від вектора X до векторів матриці W.

— Знаходження індексу мінімуму в векторі D індексу нейрона переможця і визначення вхідного вектора в кластер, відповідний нейрону-переможцю.

Також можна виділити математичні операції, використовувані в вищенаведених [18]:

— $\vec{x}^T\vec{y}$ – скалярний добуток векторів;

— $\vec{x} + \vec{y}$ – складання двох векторів;

— $\vec{x}$ – множення вектора на скаляр;

— $\min_i x_i$ – пошук мінімального елемента у векторі $\vec{x}$;

— Обчислення функції сусідства $h_{ci}(t)$

2.7 Самоорганізовані карти для символьних рядків

Самоорганізовані карти зазвичай визначаються в метричних векторних просторах. В такому випадку SOM являє собою діаграму подібності складних об'єктів. За допомогою SOM можна представляти багато різних видів векторних об'єктів. В першу чергу це, звичайно, впорядковані числові множини, що представляють набори сигналів, вимірювання або статистичних показників. Наприклад, текстові документи, можна описувати статистичними векторами, якщо останні відображають вживання слів: гістограми слів або їх стислі варіанти можна інтерпретувати як речові вектори.

Можна показати, проте, що об'єкти, які підлягають упорядкуванню. можуть мати набагато більш загальну природу. Якщо x і y є деякими об'єктами, достатня умова можливості відображення їх на діаграму SOM полягає в тому, що

для всіх пар(x, y) повинна бути визначена деякого виду симетрична функція відстані $d = d(x, y)$.

Продемонструємо відображення символьних рядків на решітку SOM. згідно з яким взаємне розташування «образів» рядків (точок) на SOM має відображати деяким способом задану відстань між ними, наприклад, відстань Левенштейна або відстань в просторі ознак між розглянутими рядками.

Якщо спробувати застосувати алгоритм SOM до таких об'єктів, відразу ж виявляється утруднення, що складається в тому. що правила покрокового навчання не можуть бути безпосередньо використані для символьних рядків, оскільки вони є дискретними об'єктами. Крім того, рядок не можна розглядати як вектор [19].

Підхід самоорганізованих карт, проте, легко можна застосувати для побудови упорядкованих діаграм подібності для об'єктів-рядків, якщо керуватися наступними ідеями.

1. Використовувати принципи побудови пакетного варіанту алгоритму SOM для визначення процесу навчання як послідовності обчислень деяких узагальнених умовних середніх за підмножини обраних рядків.

2. Ці «середні», певні на розглянутих рядках, розраховуються як узагальнені медіани цих рядків.

2.7.1 Ініціалізація SOM для рядків

Звичайну SOM визначену в векторному просторі, можна ініціалізувати за допомогою випадкових векторних значень. Точно так само можна формувати і SOM для строкових величин, починаючи процес навчання з випадкових еталонних рядків.

Однак є можливість значно прискорити цей процес, якщо початкові значення рядків будуть уже впорядкованими, хоча б і зовсім грубо. В цьому випадку можна використовувати область сусідства меншого розміру, навіть і фіксованого, що складається тільки з самого вузла і його найближчих сусідів.

Тепер можна скористатися тим фактом, що медіана розглянутого множини є одним з вхідних зразків. Тоді з'являється можливість визначити впорядковані

рядки для використання їх в якості початкових значень, якщо спочатку для вхідних зразків сформувати відображення Селшона.

Шляхом побудови проекцій для достатнього числа представницьких вхідних зразків можна потім вручну відібрати таку їх підмножину, яка виявиться двумірно впорядкованою.

Якщо символьні рядки досить довгі, попередній порядок для них можна визначити за допомогою аналізу гістограм символів в кожному рядку, що виконується відповідно до алфавітного порядку символів. Це означає, що будується попередній еталонний вектор для кожного елемента решітки SOM з тієї ж самої розмірністю, що і у гістограм.

В ході цього першого попереднього етапу ініціалізації спочатку формується традиційна SOM. Елементи решітки нейронів цієї SOM можна помітити рядками, гістограми яких відображаються на відповідні елементи, однак щоб отримати унікальні мітки для всіх елементів, стосовно кожного з них вибирається найбільш часто зустрічувана мітка. Після того як отримано попередній, приблизно упорядкований набір символьних рядків, якими позначені елементи карти, їх можна вважати початковими значеннями еталонних рядків і не використовувати більше векторні моделі гістограм.

Потім здійснюється процес більш точної самоорганізації розглянутих рядків, як це описується нижче.

2.7.2 Пакетний варіант самоорганізуючої карти для рядків

Обчислення, проведені при формуванні звичайного пакетного варіанту самоорганізуючої карти, майже безпосередньо застосовні і в разі строкових вхідних величин [20]:

1. Вибрати початкові еталонні рядки методом, описаним у. 2.4.1.
2. Для кожного і-го елемента карти скласти список тих вхідних рядків, для яких еталонний рядок і-го елемента є найближчим.
3. Для кожного і-го елемента карти взяти для нового еталонного рядку узагальнену медіану по об'єднанню списків, що належать топологічній околиці N_i розглянутого і-го елемента.

4. Повторювати п. 2 і 3 достатню кількість разів, поки еталонні рядки не перестануть змінюватися при подальших ітераціях.

Іноді трапляється так, що реалізовувані корекції призводять до появи «граничного циклу», внаслідок чого еталонні рядки осцилюють між двома альтернативними значеннями, це зазвичай означає неможливість віддати перевагу при виборі переможця. В даному випадку алгоритм слід якимось чином завершити, наприклад, вільним вибором однієї з наявних альтернатив.

2.7.3 Виявлення переможця в ситуації нерозрізненості порівнюваних рядків

Відзначимо, що всі рядки є об'єктами з дискретними значеннями. Через це часто при виконанні операцій порівняння може виникати «нічийний результат», тобто. Ситуація нерозрізненості порівнюваних рядків. Внаслідок цього збіжність до кінцевих значень при навчанні карт великої розмірності може сповільнитися, якщо не вжити деяких запобіжних заходів. Ці ситуації нерозрізненості. перш за все, можуть трапитися [20]:

1) при пошуку переможця;
2) при знаходженні медіани.

При пошуку переможця може легко виникнути ситуація нерозрізненості, якщо рядки дуже короткі. В цьому випадку відстані від декількох вхідних рядків до найближчого еталонного рядка можуть виявитися рівними, навіть якщо еталонні рядки сильно розрізняються.

Якщо використовувати зважену відстань Левенштейна, число ситуацій нерозрізненості цього типу буде невелика і одного з рівних переможців можна вибрати довільним чином.

Якщо в процесі формування карти для прискорення обчислень при отриманні нових значень еталонних рядків використовуються медіани множин і при цьому виявляється ситуація нерозрізненості, можна прийняти, що найкращий кандидат вибирається залежно від довжини рядка.

Після попереднього формування гістограм і подальшої розмітки елементів карти за допомогою всіх об'єктів-рядків, що відображаються на цю карту, вибір

за допомогою механізму голосування шляхом простої більшості може призводити до появи ситуацій нерозрізненості. У такій ситуації можна вибрати позначку згідно з її довжиною або ж випадковим чином.

2.8 Висновки до розділу 2

В роботі пропонується використання алгоритму SOM для виявлення ключових слів у публічних текстах. Даний метод використовує корпусу документів з розміченими ключовими словами, якийсь словник базових ключових слів. Помічені ключові слова вважаються позитивним прикладом, інші слова - негативним прикладом. Далі вираховується релевантність кожного слова тренувального тексту шляхом зіставлення йому вектора значень різних параметрів, наприклад, довжини слова, частини мови. Фіксуються відміну значень векторів цих параметрів для ключових слів і не ключових. Далі обчислюється ймовірність віднесення кожного слова до групи ключових і задається її поріг, т.ін. Модель навчається. Витяг ключових слів з нового документа відбувається шляхом обчислення релевантності слів і їх ймовірності віднесення до ключових відповідно до побудованої моделі.

3 ПРАКТИЧНА ЧАСТИНА. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ПУБЛІЧНИХ ТЕКСТІВ

3.1 Проектування програмного забезпечення для аналізу публічних текстів

3.1.1 Функціональні вимоги

Функціональна вимога – бажана поведінка системи з точки зору її користувача. Функціональні вимоги реалізуються функціями ПС. Функція системи - поведінка, яку необхідно реалізувати в розроблювальної програмної системі [21].

Описувати функціональні вимоги будемо за допомогою методології RUP.

Функціональні вимоги в RUP моделюються за допомогою варіантів використання. Варіант використання (Use Case, прецедент) – зовнішня специфікація послідовності дій, які система може виконувати в процесі взаємодії з діючими особами (actor) з метою отримання значимого для них результату.

Функціональні вимоги мігрують з legacy system (LS) до target system (TS).

Користувач ПС, тобто основний актор цього прецеденту – дослідник.

Основний вдалий сценарій – аналіз текстової інформації на предмет виявлення терористичних загроз:

- завантаженя файлів з потрібним текстом/текстами;
- проведення аналізу текстової інформації;
- отримання результатів аналізу.

Пост-умови:

- дослідник отримує гістограму, сгруповані тексти у кластери за небезпекою, та загальні оцінки по кожному тексту;
- дослідник отримує докладний звіт з результатами аналізу текстової інформації.

На рисунку 3.1 зображена діаграма варіантів використання. Use-case діарама в UML – діаграма, що відображає відносини між акторами і

прецедентами і є складовою частиною моделі прецедентів, що дозволяє описати систему на концептуальному рівні.

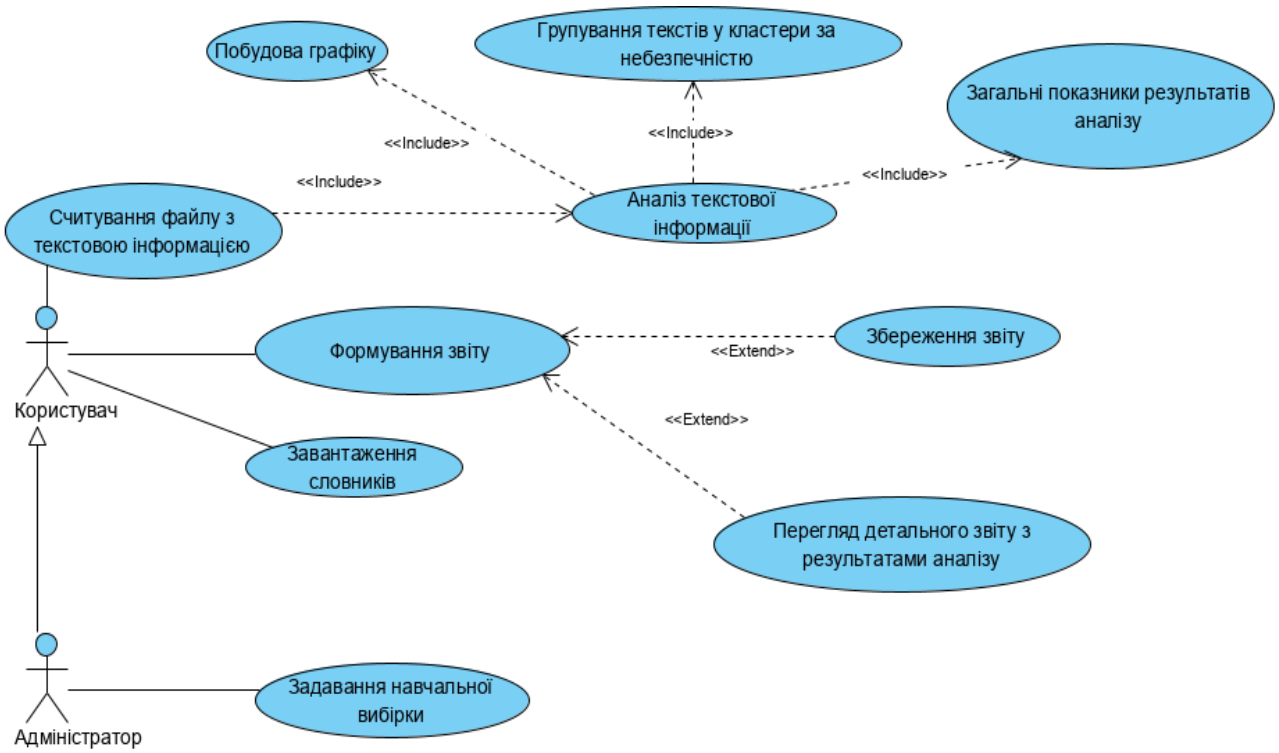

Рисунок 3.1 – Діаграма варіантів використання

3.1.2 Нефункціональні вимоги

Нефункціональні вимоги фіксують умови, які безпосередньо не пов'язані з поведінкою або функціональністю рішення, а скоріше описують умови навколишнього середовища, при яких рішення має залишатися ефективним, або якості, якими система повинні володіти. Вони також відомі як атрибути (показники) якості або додаткові вимоги. Вони можуть включати вимоги, пов'язані з пропускною здатністю, швидкістю, безпекою, доступністю, інформаційної архітектурою і поданням призначеного для користувача інтерфейсу [22].

Опираючись на стандарт ISO / IEC 9126 який пропонує комплексну ієрархічну структуру опису якісних характеристик. Основними характеристиками високого рівня є [23]:

– функціональність (functionality);

- ефективність (efficiency);
- супровід (maitainnability);
- надійність (realibility);
- переносимість (portability);
- зручність використання (usability).

Основними нефункціональними вимогами які були виділені при розробці вимог до системи такі:

- інтерфейс користувача повинен бути мінімальним та інтуїтивно зрозумілим;
- допустима кількість збоїв в системі не більше 1 збою в три місяці
- при виникненні збоїв в програмі система повинна відновлюватись до попереднього стану;
- при відмові системи дані не повинні втрачатись.
- середній час відновлення після збою не більше 1 години

3.2 Розробка системної програмної архітектури

Архітектура системи – принципова організація системи, втілена в її елементах, їх взаєминах один з одним і з середовищем, а також принципи, направляючі її проектування та еволюцію.

Логічна архітектура підтримує функціонування системи протягом усього її життєвого циклу на логічному рівні. Вона складається з набору пов'язаних технічних концепцій і принципів. Логічна архітектура представляється за допомогою методів, що відповідають тематичними групами описів, і як мінімум, включає в себе функціональну архітектуру, поведінкову архітектуру і тимчасову архітектуру [24].

Функціональна архітектура. Функціональна архітектура являє собою набір функцій та їх підфункцій, що визначають перетворення, здійснювані системою при виконанні свого призначення.

Поведінкова архітектура. Поведінкова архітектура - угода про функції та їх підфункції, а також інтерфейсах (входи і виходи), які визначають послідовність

виконання, умови для управління або потоку даних, рівень продуктивності, необхідний для задоволення системних вимог. Поведінкова архітектура може бути описана як сукупність взаємопов'язаних сценаріїв, функцій та / або експлуатаційних режимів.

Тимчасова архітектура. Тимчасова архітектура є класифікацією функцій системи, яка отримана відповідно до рівня частоти її виконання. Тимчасова архітектура включає в себе визначення синхронних і асинхронних аспектів функцій. Моніторинг рішень, який відбувається всередині системи, слідує тієї ж тимчасової класифікації

Системну архітектуру також можна представити у вигляді кортежу трьох множин :

$$\text{SA} \Rightarrow < C, F, I >,$$

де: SA (SystemArchitecture) – системна архітектура;

C– множина програмних компонентів(модулів), які реалізують функціональність даної системи .

F– множина допустимих форм взаємодії (конфігурацій), у якому можуть бути об’єднані елементи з множини компонентівC.

I– множина системних інтерфейсів, засобами яких, елементи множин $C_{\text{та}}F$взаємодіють один з одним, або звертаються до зовнішніх, по відношенню до даної системи, компонентам.

3.2.1 Вибір цільового варіанту архітектури ПЗ

Абсолютна більшість сучасних ПС є розподілені мережеві програмні рішення, тому в основу класифікації ECA може бути покладено уявлення про те, яким чином окремі компоненти ПО розташовуються на різних вузлах тієї чи іншої мережевої конфігурації (локальна мережа, корпоративна мережа, інтранет і т.д.). Крім того, можна стверджувати, що в складі ПО ПС в загальному випадку можуть бути виділені наступні групи програмних компонентів (сервісів)

1) сервіси представлення даних (data PResentation Services – PRS) – це різні діалогові форми, HTML-сторінки, меню тощо;

2) сервіси бізнес-логіки (Business Logic Services – BLS) – до них відносяться будь-які програмно-реалізовані алгоритми обробки даних і сервісні функції ПС;

3) послуги доступу до даних (Data Access Services – DAS) – це компоненти ПО для доступу до даних, а також і будь-які БД, які створюються з використанням відповідних СУБД (включаючи і їх самих) [24].

Зважаючи на те, що в програмному забезпеченні яке проектується відсутня модель даних замість неї використовуються словники, враховуючи позитивні та негативні сторони наведених вище системних архітектур, було вирішено використовувати дворівневу архітектуру типу «клієнт-сервер» » з тонким клієнтом, так як цей тип архітектури найбільш підходить для реалізації ПЗ аналізу публічних текстових повідомлень в мережі Інтернет.

Діаграма розгортання системи, що проектується наведено на рисунку 3.2.

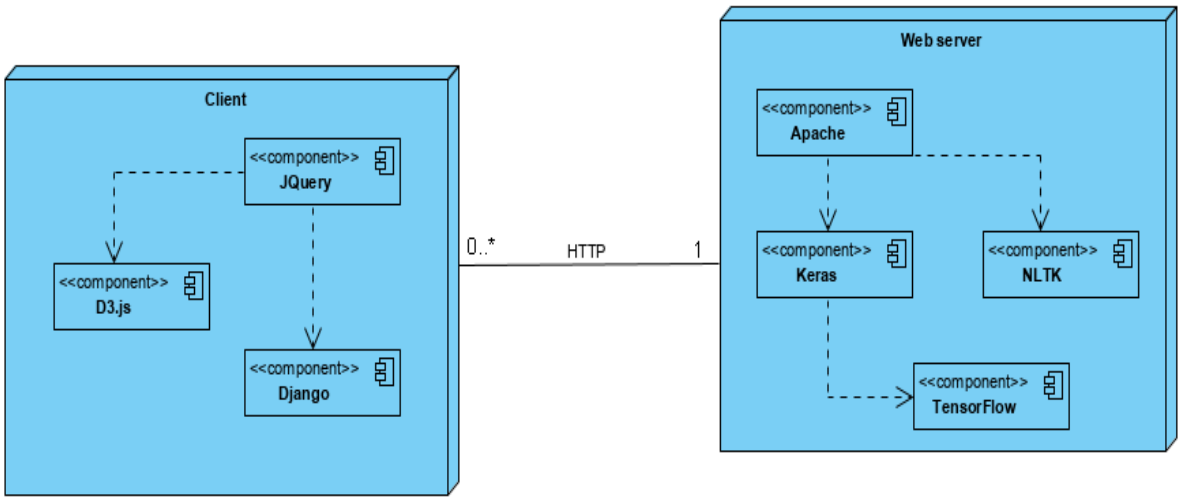

Рисунок 3.2 – Діаграма розгортання системи

3.3 Діаграма компонент

Діаграма компонентів описує особливості фізичного представлення системи, дозволяє визначити архітектуру розроблюваної системи, встановивши залежності між програмними компонентами, в ролі яких може виступати вихідний, бінарний і виконуваний код [25]. У багатьох середовищах розробки модуль або компонент відповідає файлу. Пунктирні стрілки, що з'єднують модулі, показують відношення взаємозалежності, аналогічні тим, які мають місце при компіляції початкового програмного коду. Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними.

На рисунку 3.3 зображена діаграма компонентів для розроблюваного програмного забезпечення

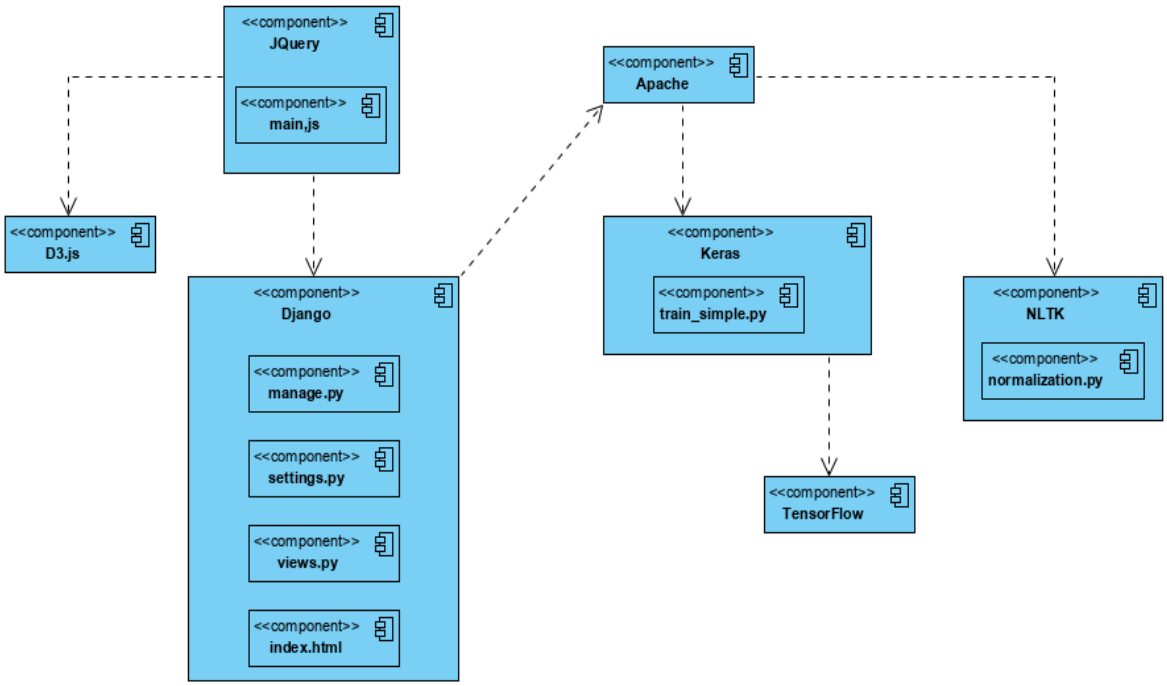

Рисунок 3.3 – Діаграма компонент

3.4 Вибір інструментальних програмних засобів та інформаційних технологій

Для розробки програмного забезпечення планується використовувати наступні інструментальні засоби та інформаційні технології.

Python – це універсальна сучасна мова програмування високого рівня, до переваг якого відносять високу продуктивність програмних рішень і структурований, добре читабельний код. Синтаксис Пітона максимально полегшений, що дозволяє вивчити його за порівняно короткий час. Ядро має дуже зручну структуру, а широкий перелік вбудованих бібліотек дозволяє застосовувати значний набір корисних функцій і можливостей. мова програмування може використовуватися для написання прикладних програм, а також розробки WEB-сервісів [26].

Python може підтримувати широкий перелік стилів розробки додатків, в тому числі, дуже зручний для роботи з ООП і функціонального програмування.

Один з найпопулярніших інтерпретаторів мови - CPython, написаний на Сі. Поширюється це середовище розробки безкоштовно за вільною ліцензією. Інтерпретатор підтримує більшість популярних платформ.

Пітон підтримує практично всі поширені операційні системи. Він може прекрасно працювати на кишенькових комп'ютерах, так і на великих серверах. У разі, якщо платформа значно застаріває, вона виключається з підтримки ядра. Наприклад, версії мови, починаючи від 2.6, вже не працюють з платформами Windows 95, 98 і ME. У разі необхідності можна скористатися більш старими версіями, відмовившись від застосування сучасних інструментів мови. І тоді додаток буде працювати в тому числі з цими ОС. Для старих версій періодично виходять патчі. Мова також може підтримувати роботу з віртуальною машиною Java. [26].

PyCharm – це інтелектуальна Python IDE з повним набором засобів для ефективної розробки на мові Python. Випускається в двох варіантах - безкоштовна версія PyCharm Community Edition і яка підтримує більший набір можливостей PyCharm Professional Edition. PyCharm виконує інспекцію коду на льоту, автодоповнення, в тому числі ґрунтуючись на інформації, отриманої під час виконання коду, навігацію по коду, забезпечує множину рефакторингів [27].

Ключові можливості [27]:

— Потужний і функціональний редактор коду з підсвічуванням синтаксису, авто-форматуванням і авто-відступами для підтримуваних мов.

— Проста і потужна навігація в коді.

— Допомога при написанні коду, що включає в себе автодоповнення, авто-імпорт, шаблони коду, перевірка на сумісність версії інтерпретатора мови, і багато іншого.

— Швидкий перегляд документації для будь-якого елемента прямо у вікні редактора, перегляд зовнішньої документації через браузер, підтримка docstring - генерація, підсвічування, автодоповнення і багато іншого.

— Потужний рефакторинг коду, який надає широкі можливості щодо виконання швидких глобальних змін у проекті.

— Повна підтримка свіжих версій Django фреймворка.

— Підтримка Google App Engine .

— Підтримка IronPython, Jython, Cython, PyPy, wxPython, PyQt, PyGTK і ін.

— Редактор Javascript, Coffescript, HTML / CSS, SASS, LESS, HAML.

— Інтеграція з системами контролю версій (VCS).

— UML діаграми класів, діаграми моделей Django і Google App Engine.

— Інтегроване Unit тестування.

— Інтерактивні консолі для Python, Django, SSH, відладчика і баз даних.

— PyCharm має кілька колірних схем, а також настроюється підсвічування синтаксису коду.

— Інтеграція з баг / issue-трекерами, такими як JIRA, Youtrack, Lighthouse, Pivotal Tracker, GitHub, Redmine

— Крос-платформеність (Windows, Mac OS X, Linux).

JQuery – це невелика за обсягом бібліотека, створена на основі JavaScript, яка сильно спрощує програмування на мові JavaScript. За словами творців цієї бібліотеки, це маленька, швидка і розширювана JS бібліотека [28].

З її допомогою можна набагато простіше переміщуватися по HTML елементам, управляти анімацією, обробляти події, працювати з Ajax запитами.

Все це завдяки API (Прикладний програмний інтерфейс) і підтримки цієї бібліотеки практично у всіх браузерах.

Django – це високорівневий Web-фреймворк, який реалізований на основі архітектури MVC. Django має прозорий дизайн, дає можливість для оперативної розробки Web-додатків, дозволяє розробляти динамічні Web-сайти [29].

Відмінні риси Django:

– будь-який запит обробляється програмно і перенаправляється на свою адресу (url);

– поділ контенту та подання за допомогою шаблонів;

– абстрагування від низького рівня баз даних.

Django-додаток складається з чотирьох основних компонентів.

1. Модель даних: дані є серцевиною будь-якого сучасного Web-додатку. Модель - найважливіша частина програми, яка постійно звертається до даних при будь-якому запиту з будь-якої сесії. Будь-яка модель є стандартним Python класом. Об'єктно-орієнтований маппер (ORM) забезпечує таким класам доступ безпосередньо до баз даних. Якби не було ORM, програмісту довелося б писати запити безпосередньо на SQL.

2. Представлення (view): в Django виконують різноманітні функції, в тому числі контролюють запити користувача, видають контекст в залежності від його ролі. View - це звичайна функція, яка викликається у відповідь на запит якоїсь адреси (url) і повертає контекст.

3. Шаблони: вони є формою представлення даних. Шаблони мають свою власну просту метамову і є одним з основних засобів виведення на екран.

4. URL: це всього лише механізм зовнішнього доступу до уявлень (view). Вбудовані в url регулярні вирази роблять механізм досить гнучким. При цьому одне подання може бути налаштоване до кількох url, надаючи доступ різним додаткам.

Apache – це HTTP сервер, що володіє високою надійністю і гнучкістю, під HTTP сервером слід розуміти програмне забезпечення для обробки HTTP запитів. Основна робота Apache це обробка і відповідь на HTTP запити і

генерувати динамічний зміст сторінок. Гнучкість досягається шляхом використання файлу .htaccess, завдяки якому можна перевизначати глобальні настройки сервера Apache [30].

Keras – відкрита нейромережева бібліотека, написана на мові Python. Вона являє собою надбудову над фреймворками Deeplearning4j, TensorFlow і Theano. Націлена на оперативну роботу з мережами глибинного навчання, при цьому спроектована так, щоб бути компактною, модульною та розширюваною. Вона була створена як частина дослідницьких зусиль проекту ONEIROS [29].

Ця бібліотека містить численні реалізації широко застосовуваних будівельних блоків нейронних мереж, таких як шари, цільові та передавальні функції, оптимізатори, і безліч інструментів для спрощення роботи з зображеннями і текстом.

TensorFlow – відкрита програмна бібліотека для машинного навчання, розроблена компанією Google для вирішення завдань побудови і тренування нейронної мережі з метою автоматичного знаходження та класифікації образів, досягаючи якості людського сприйняття [29].

Обчислення TensorFlow виражаються у вигляді потоків даних через граф станів. Назва TensorFlow походить від операцій з багатовимірними масивами даних, які також називаються «тензорами».

NLTK (Natural Language Toolkit) – провідна платформа для створення NLP-програм на Python. У неї є легкі у використанні інтерфейси для багатьох мовних корпусів, а також бібліотеки для обробки текстів для класифікації, токенізації, стемінг, розмітки, фільтрації і семантичних міркувань. Ну і ще це безкоштовний open source проект, який розвивається за допомогою ком'юніті [29].

3.3 Висновки до розділу 3

В розділі сформовано функціональні та нефункціональні вимоги до програмного забезпечення, що реалізує запропонований в роботі алгоритм SOM.

На основі сформованих вимог побудовано діаграми варіантів використання, що демонструють основний базовий функціонал програмного забезпечення.

Проведено аналіз архітектур сучасного програмного забезпечення. Зважаючи на те, що в програмному забезпеченні яке проектується відсутня модель даних замість неї використовуються словники, враховуючи позитивні та негативні сторони наведених вище системних архітектур, було вирішено використовувати дворівневу архітектуру типу «клієнт-сервер» » з тонким клієнтом, так як цей тип архітектури найбільш підходить для реалізації ПЗ аналізу публічних текстових повідомлень в мережі Інтернет.

Проведено аналіз та вибір інструментальних засобів розробки. Для розробки веб-застосування пропонується використовувати Django. Django – це високорівневий Web-фреймворк, який реалізований на основі архітектури MVC. Django має прозорий дизайн, дає можливість для оперативної розробки Web-додатків, дозволяє розробляти динамічні Web-сайти. Алгоритм SOM реалізується за допомогою Keras – відкритої нейромережевої бібліотеки, написаної на мові Python.

4 ПРИКЛАД ЗАСТОСУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Особливості реалізації програмного забезпечення

Опишемо основні сценарії використання програмної системи

Кожному користувачу системи необхідно мати веб–браузер та стабільний вихід до мережі інтернет .

Робота користувача з програмою починається з головного меню що зображено на рисунку 4.1.

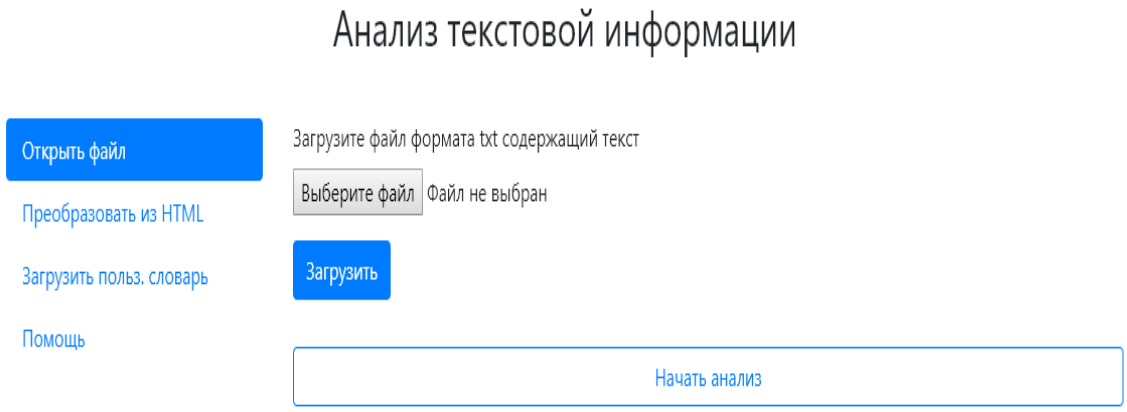

Рисунок 4.1 – Головне меню

Для проведення аналізу текстової інформації необхідно у вкладці «Відкрити файл» завантажити файл формату txt в якому знаходяться необхідні для аналізу тексти, та натиснути кнопку «Завантажити», якщо цього не зробити кнопка «Почати аналіз» буде неактивна. Після цього натиснути на кнопку «Почати аналіз», яка почне аналіз, процес аналізу тесстової інформації зображено на рисунках 4.2 – 4.6.

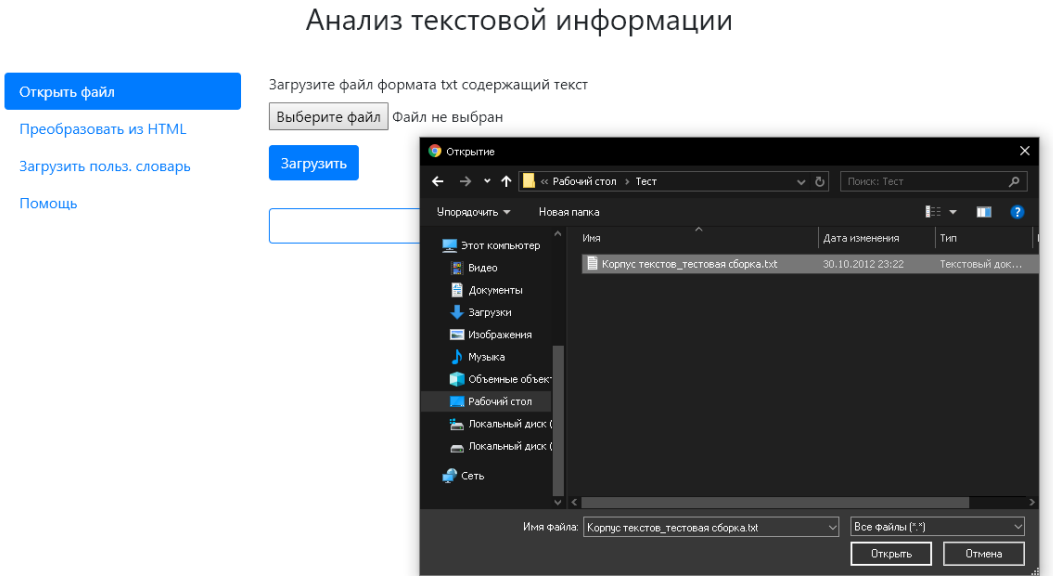

Рисунок 4.2 – Вибір файлу

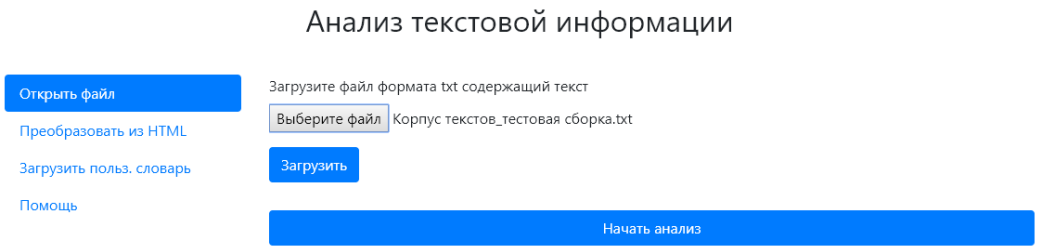

Рисунок 4.3 – Початок аналізу

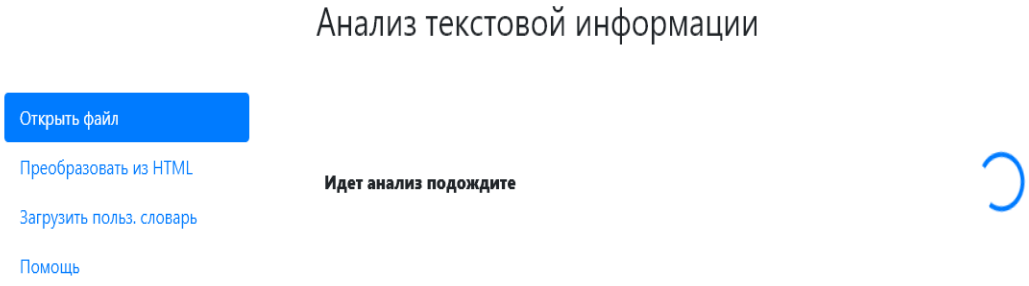

Рисунок 4.4 – Процес аналізу текстової інформації

Примітка: процес аналізу може займати до 1 хвилини, все залежить від кількості аналізуємої інформації.

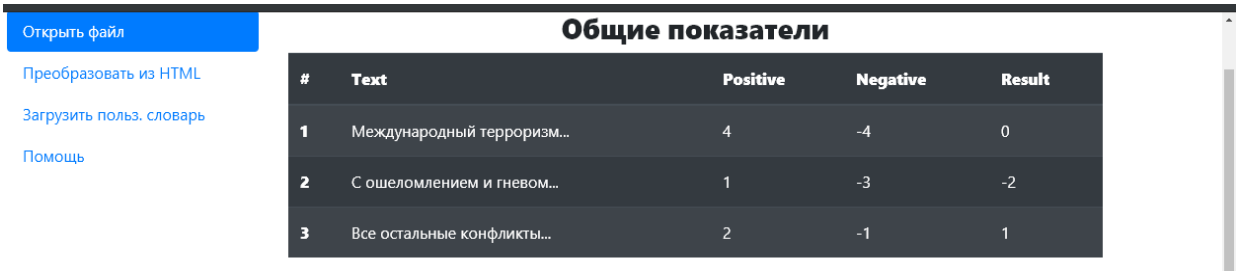

Рисунок 4.5 – Результат аналізу 1

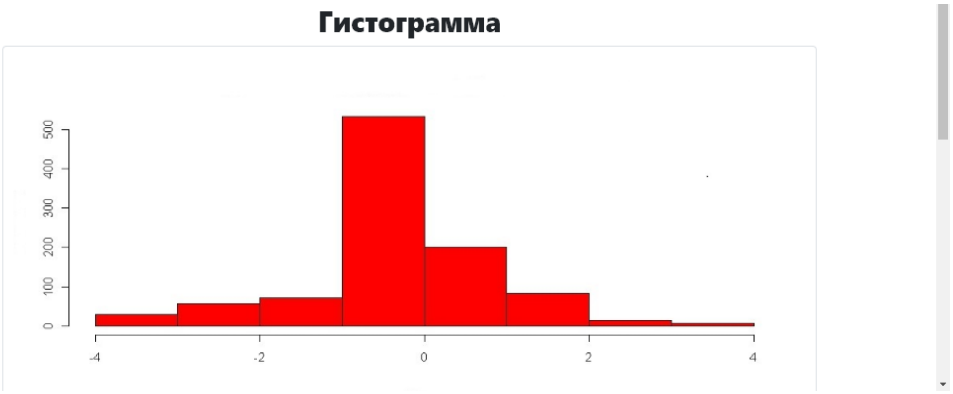

Рисунок 4.6 – Результат аналізу 2

Процес аналізу працює на основі словника емоційно забарвленої лексики,всі слова в якому закодовані від -5 до -1 для слів, що виражають негативні емоції і від 1 до 5 для слів, що виражають позитивні емоції.

Після проведення аналізу на єкрані з’являются результати, а саме таблиця з загальними показниками по кожному тексту, гістограма на я кій зображене частотне розподілення оцінок що зображено у таблиці 4.1, кластеризація, та кнопка для перегляду докладного звіту де показано розбір кожного тексту.

Таблиця 4.1 – Частотне розподілення оцінок

Оцінка	-5	-4	-3	-2	-1	0	1	2	3	4	5
Кількість текстів	0	7	20	50	74	476	219	111	25	17	0

Вкладка «Преобразити з HTML» проводить аналіз аналогічним чином, який показано вище, за одніє відмінністю що для початку аналізу необхідно витягнути текст з тегів HTML натиснувши на кнопку «Преобразити». Пункт меню «Преобразити з HTML» зображено на рисунку 4.7

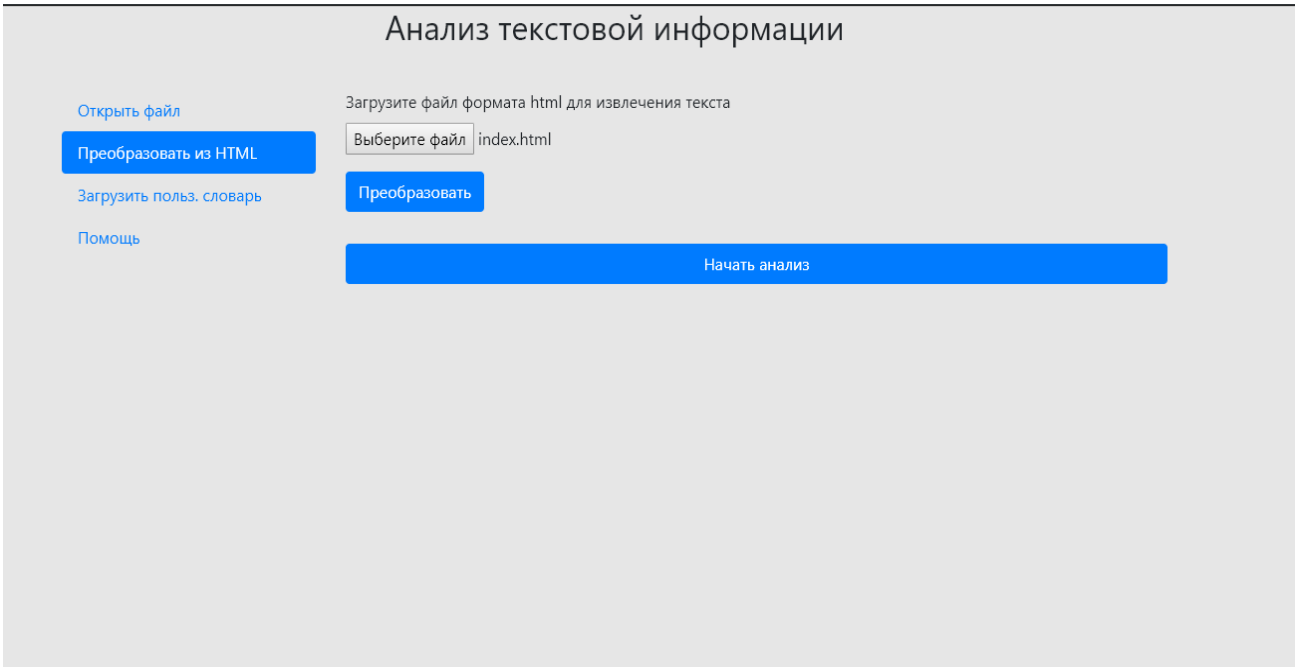

Рисунок 4.7 – Вкладка «Перетворити із HTML»

На вкладці «Завантажити користувацький словник» можна завантажити свій словник який оформлено згідно з умовами, які описані у пункті меню «Допомога», формат файла повинен буди csv. Словник буде працювати в рамках поточної сесії, в разі закриття системи словник буде необхідно завантажити знову Користувацький словник може бути необхідним в ситуаціях коли потрібно провести аналіз згідно своїх потреб для конкретної галузі дослідження. Пункт «Завантажити користувацький словник», зображено на рисунку 4.8.

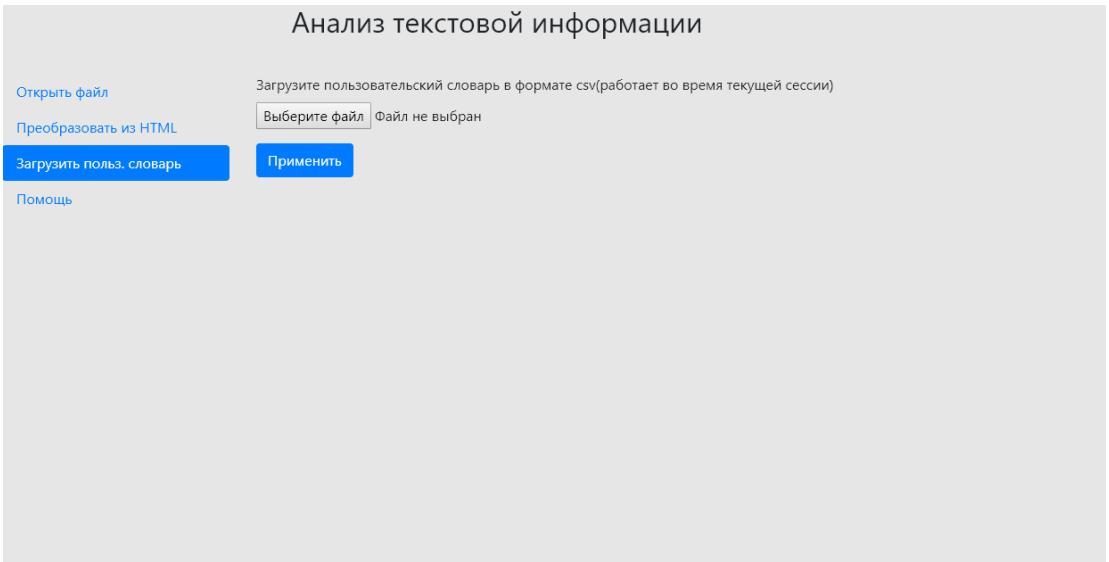

Рисунок 4.8 – Вкладка «Завантажити словник»

4.2 Аналіз результатів

Аналіз буде проводитись на основі отриманих результатів описаних вище на прикладі одного з текстів. Для цього буде використан фрагмент словника з вибіркою слів, які містяться в тексті прикладу на якому буде розглянуто принцип аналізу текстової інформації. Фрагмент словнику з закодованими оцінками показано у таблиці 4.2.

Таблиця 4.2 – Фрагмент словнику

Слово	Оцінка
Міжнародний	0
Тероризм	-2
Невід'ємна	1
Частина	1
Процес	0
Поширення	1
Транснаціональний	0
Злочин	-1
Організація	0
Підтримка	2
Корумпований	-2
Держава	0
Чиновник	0
Політик	0
Одержав	1
Популярність	2
Робота	0
Вчений	1
Негативні	-1
Терорист	-2
Кримінал	-1
Конфлікт	-1
Контрабандист	-1
Наркоторгівля	-2
Організована	2
Злочин	-1
Вирішення	1
Проблема	-1
Найважливіший	2
Уряд	1
Поліцейський	2
Сила	1
Світ	1

Далі буде наведено розбір частини статті з інтернету про міжнародний тероризм:

“Міжнародний(0) тероризм(-2) є(0) в(0) наш(0)і дні(0) невід'ємною(1) частиною(1) процесу(0) поширення(1) транснаціональних(0) злочинних(-1) організацій(0), підтримуваних(2) корумпованими(-1) державними(0) чиновниками(0) і(0) політиками(0) {Речення: -4; 5}. Так(0), роботі(0) англійських(0) вчених(1) «Глобальні(0) трансформації(0)» що(0) одержала(1) широку(0) популярність(2) зазначається(0): «Існують(0) і(0) негативні(-1) форми(0) міжнародних(0) організацій(0), такі(0) як(0) терористичні(-2) й(0) кримінальні(-1) організації(0) {Речення: -4; 4}. Незважаючи(0) на(0) триваючий(0) багато(2) століть(0) конфлікт(-1) між(0) контрабандистами(-1) і(0) владою(0), в(0) останні(0) роки(0) зростання(0) транснаціональних(0) кримінальних(-1) організацій(0) пов'язаний(0) з(0) наркоторгівлею(-2) широким(0) поширенням(0) організованої(0) злочинності(-1). {Речення: -5; 2} Вирішення(1) цих(0) проблем(-1) стало(0) одним(0) з(0) найважливіших(2) завдань(0) для(0) урядів(1) і(0) поліцейських(2) сил(1) в(0) усьому(0) світі(1) » {Речення: -2; 5}”

Загальна позитивна оцінка тексту: 4 .

Загальна негативна оцінка тексту: -4

Загальна оцінка тексту розраховується, як сума двох оцінок, в данному випадку вона дорівнює 0, тобто текст є нейтральним.

Для інтерпретації результатів аналізу текстової інофрмації використовується гистограмма частотного розподілення оцінок.

Якщо частотний розподіл за сумарною оцінкою позитивних і негативних емоцій більше нагадує нормальний розподіл, то це показник того, що в більшості випадків тексти з колекції отримали нейтральну оцінку (0 або 1 / -1). Перекошення розподілу в праву або ліву сторону говорить про те, що тексти більшою мірою отримали позитивні / негативні оцінки, а значить можна говорити про позитвне чи негативне відношення до досліджуваного об'єкта.

На рисунку 4.9 та таблиці 4.3 показано гістаграму частотного розподілення оцінок, на якій можна побачити що переважають тексти з нейтральною оцінкою.

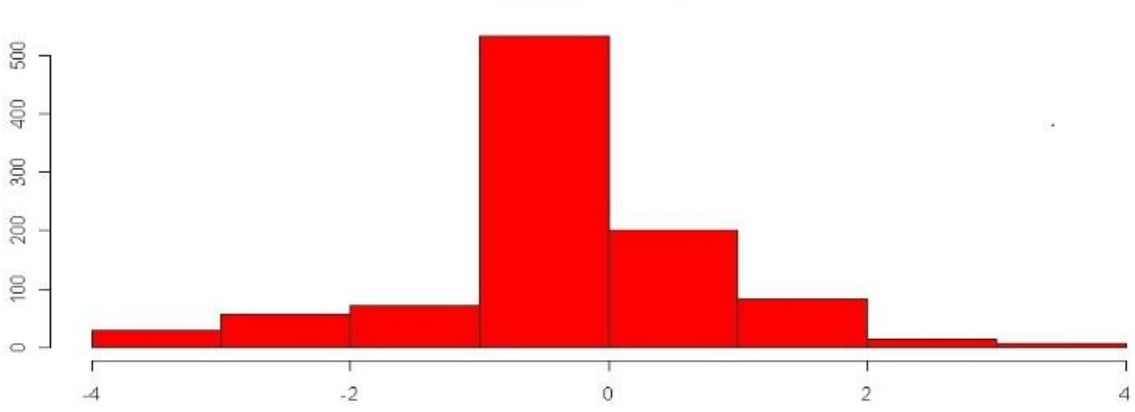

Рисунок 4.9 – Гістограма частотного розподілення

Таблиця 4.3 – Частотне розподілення оцінок

Оцінка	-5	-4	-3	-2	-1	0	1	2	3	4	5
Кількість текстів	0	7	20	50	74	476	219	111	25	17	0

4.3 Висновки до розділу 4

В розділі наведено приклад застосування розробленого програмного забезпечення. Розглянуто приклад аналізу текстів на виявлення в них ключових слів та класифікації тексту за екстремістською спрямованістю. Наведений приклад, показує що запропонований алгоритм на основі карт Кохонена є дієвим і може використовуватися для оцінки публічних текстів їх мережі інтернет.

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1 Охорона праці

Правила роботи за комп'ютером регламентуються законом України “Про охорону праці”, вимогами щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями, затверджені наказом Мінсоцполітики від 14.02.2018 р. № 207 та іншими нормативними документами.

Зокрема, ними передбачено, що на підприємстві незалежно від роду діяльності та форми власності має бути реалізована система управління охороною праці для організації виконання правових, організаційно-технічних, санітарно-гігієнічних, соціально-економічних і лікувально профілактичних заходів, спрямованих на запобігання нещасним випадкам, професійним захворюванням та аваріям у процесі праці.

Навчання та інструктаж персоналу з питань охорони праці є складовою частиною системи управління охороною праці підприємства і проводиться з усіма працівниками в процесі їх трудової діяльності.

Обов'язок по забезпеченню безпечних і нешкідливих умов праці покладається на власника підприємства. Виконання цього обов'язку вимагає від нього точного дотримання вимог нормативних актів (інструкцій, правил, стандартів) з охорони праці, розроблених на державному міжгалузевому і галузевому рівнях.

Основними потенційними небезпеками під час роботи за комп’ютером є:

- небезпека ураження електричним струмом, внаслідок недотримання правил електробезпеки або виходу з ладу електроприладів;

- порушення роботи кістково-м’язового апарату внаслідок тривалих статичних навантажень при роботі з ПК.

- нервово-психічні перевантаження внаслідок інцедентів порушень політики безпеки підприємства, контакту колегами по роботі, керівництвом при вирішенні робочих питань, які можуть носити конфліктний характер і призвести до емоційного дискомфорту, внутрішнього роздратування, емоційної нестабільності та захворювань нервової системи;

- незадовільні ергономічні характеристики робочого місця внаслідок нераціонального планування робочого місця, що може призвести до механічних травм, уражень електричним струмом та порушень кістково- м’язового апарату;

- негативний вплив недостатнього освітлення робочої зони на зір та продуктивність роботи працюючого, внаслідок несправності освітлювальних приладів або неправильного проектування освітлювальної системи;

- негативний вплив незадовільних параметрів повітряного середовища робочої зони на здоров’я працюючого, внаслідок неправильного проектування системи вентиляції або несправності її несправності;

- негативний вплив підвищеного рівня шуму на психоемоційний стан працюючого, який пов’язаний з використанням застарілої периферійної техніки, кондиціонерів, копіювальної техніки, освітлювальних приладів;

- небезпека загоряння у зв'язку із несправністю електричного обладнання, недотримання, або порушення правил протипожежної безпеки обслуговуючим персоналом, що може призвести до пожежі;

- неправильні дії персоналу у надзвичайних ситуаціях.

Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями, затверджені наказом Мінсоцполітики від 14.02.2018 р. № 207 прийшли на зміну Правилам охорони праці під час експлуатації електронно-обчислювальних машин, затверджені наказом Держгірпромнанагляду від 26.03.2010 р. № 65

Нові вимоги поширюються на всіх суб’єктів господарювання незалежно від форм власності, організаційно-правової форми і видів діяльності;

— встановлюють мінімальні вимоги безпеки і захисту здоров’я під час здійснення роботи, пов’язаної з використанням екранних пристроїв незалежно від їх типу і моделі.

Що таке «екранні пристрої»? Це електронні засоби для відтворення будь-якої графічної або алфавітно-цифрової інформації (на основі електронно-променевої трубки, рідкокристалічні, плазмові, проекційні, органічні світлодіодні монітори та інші новітні розробки у сфері інформаційних технологій).

Робочі місця, обладнані персональними комп’ютерами, заборонено облаштовувати у підвальних або цокольних приміщеннях будівель. При обладнанні приміщень забороняється використання полімерних матеріалів, що виділяють шкідливі хімічні речовини. Також слід приділити увагу забезпеченню достатнім для здійснення роботи рівнем освітлення (природного та штучного – у темну пору доби) та звукоізоляції. Для регуляції рівня освітлення природним світлом бажано застосовувати жалюзі. Окрім того, у приміщеннях, де здійснюється робота з комп’ютерами, щодня має здійснюватися вологе прибирання з метою недопущення запиленості підлоги та меблів.

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця операторів (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), мають бути надійно захищені діелектричними щиткам або сітками з метою недопущення потрапляння людини під напругу.

Особливої уваги заслуговують заходи дотримання протипожежної безпеки. Так, у всьому офісі лінії електромережі мають бути захищені від виникнення короткого замикання, а також від перепадів мережевої напруги, що може спричинити збої в роботі електронно–обчислювальної техніки. Приміщення (окрім тих, де розташовуються сервери) мають бути оснащені системою автоматичної пожежної сигналізації та вогнегасниками. Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією і, за можливості, застосовувати негорючу ізоляцію. У приміщенні, де одночасно експлуатуються понад п'ять комп’ютерів, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

Роботодавець повинен проінформувати працівників під розписку про умови праці та наявність на їх робочих місцях небезпечних і шкідливих факторів (фізичних, хімічних, біологічних, психофізіологічних), які виникають під час роботи з комп’ютером і ще не усунені, а також про можливі наслідки їх впливу на здоров’я працівників.

У разі дотримання всіх правил щодо охорони праці при роботі з комп'ютером (зокрема, ДСанПіН 3.3.2.007-98) її не вважають важкою або шкідливою. Тому жодні компенсації за «шкідливою» підставою працівникам, які працюють з комп'ютером, не надають. Але якщо все-таки небезпечні та/або шкідливі фактори на робочому місці присутні, роботодавець повинен повідомити про них працівникові під розписку. Ця вимога не нова і закріплена в ст. 5 Закону України «Про охорону праці» від 14.10.92 р. № 2694-XII

Роботодавець повинен:

- забезпечити навчання і перевірку знань працівників з питань охорони праці і безпечного використання екранних пристроїв до початку роботи з ними, а також у випадках модифікації й організації роботи обладнання;

- вживати відповідних заходів, щоб забезпечити відповідність робочого місця працівника Вимогам № 207, зокрема до організації робочого місця працівника, мікроклімату приміщень тощо;

- за рахунок тривалості робочої зміни організувати внутрішні регламентовані перерви для відпочинку відповідно до ДСанПіН 3.3.2.007-98.

5.2 Безпека в надзвичайних ситуаціях

5.2.1 Техніка безпеки

Згідно з ПУЕ всі приміщення підприємств поділяються на 3 класи:

- без підвищеної небезпеки (звичайні);
- з підвищеною небезпекою;
- особливо небезпечні.

До приміщень без підвищеної небезпеки відносять сухі приміщення без пилу з нормальною температурою повітря, з підлогою з ізоляційного матеріалу, у яких відсутні заземлені предмети або їх дуже мало (кімнати управління, офіси, кімнати майстрів, контори цехів, кабінети начальників).

Відділ матеріально-технічного забезпечення відноситься до приміщень без підвищеної небезпеки враження електричним струмом. Оскільки в аудиторії відсутні ознаки підвищеної небезпеки: відносна вологість повітря більше 75%,

температура повітря вище 35 С; та особливої небезпеки: відносна вологість повітря близька до 100%, хімічно активне середовище. Також дотримані всі вимоги щодо експлуатації електирчних мереж та підключеного до них обладнання.

Обладнання у приміщенні складається з 3 комп’ютерів, які розташовані на робочих місцях рівномірно; свіч для організації мережі знаходиться в оптимальному місці з точки зору безпеки та мінімізації впливу на працюючих – у дальньому кутку аудиторії. Відстань між робочими місцями більше 1 м. Проходи у приміщенні вільні від дротів та інших перешкод. Електромережа влаштована згідно зі стандартами[8].

Устаткування, що використовується підлягає постійній модернізації.

Для захисту працюючих в відділі від ураження електричним струмом використовуються наступні засоби захисту:

– У приміщенні організовано заземлення – це навмисне електричне з'єднання з землею, або її еквівалентом, металевих частин обладнання, що не проводять струму, але можуть опинитися під напругою.

Призначення захисного заземлення – захист від небезпеки ураження електричним струмом при дотику до металевих корпусів електрообладнання, яке внаслідок порушення електричної ізоляції опинилося під напругою.

Принцип дії захисного заземлення полягає в зниженні до безпечних значень напруги дотику, яка обумовлена замиканням на корпус.

– Забезпечене захисне занулення – навмисне електричне з'єднання металевих частин електричних установок, що не проводять струм, але можуть опинитися під напругою, з нульовим захисним провідником.

Призначення занулення – усунення небезпеки ураження струмом у випадку дотику до корпусу електричної установки та інших металевих частин, що не проводять струму та можуть опинитися під напругою, відносно землі внаслідок замикання на корпус та через інші причини.

Принцип дії занулення – перетворення замикання на корпус на однофазне коротке замикання, тобто замикання між фазним і нульовим захисним провідником з метою викликати великий струм, здатний забезпечити

спрацювання захисту і таким чином автоматично відключити пошкоджену електроустановку від мережі живлення.

– У мережаних фільтрах на вході у приміщення передбачено автоматичне вимкнення струму за необхідністю – захисне вимкнення – швидкодіючий захист, який забезпечує автоматичне вимкнення електроустановки при виникненні в ній небезпеки ураження струмом.

Для забезпечення техніки безпеки відділу проводяться інструктажі з техніки безпеки (для нових працівників проводиться вступний інструктаж та первинний інструктаж на робочому місці; повторний інструктаж для всіх працівників, який проводиться щоквартально; позаплановий інструктаж, який проводиться при впровадженні нового обладнання, нових технологій, або при отриманні травми), проводиться регулярне вологе прибирання приміщення. Як наслідок за роки роботи відділу не виникло жодного нещасного випадку, тому можна сказати, що техніка безпеки відділу знаходиться на достатньому рівні.

5.2.2 Пожежна безпека

Пожежна безпека – це стан об'єкту, при якому з регламентованою ймовірністю виключається можливість виникнення і розвитку пожежі, а також дії на людей її небезпечних факторів на підприємствах. Крім того, забезпечується захист матеріальних цінностей.

Основою аналізу протипожежних мір є визначення категорії приміщення за пожежною небезпекою відповідно до норм технологічного проектування. Всього визначено 5 категорій приміщень за пожежною безпекою – А, Б, В, Г та Д[2]. Приміщення відділу матеріально-технічного забезпечення центру відноситься до категорії В. Подібна класифікація зумовлена наявністю горючих речовини і матеріалів в холодному стані (столи, стільці, шафи, меблі, паперові документи) в аудиторії, та відсутністю вибухонебезпечних парів і концентратів.

Вогнестійкість – здатність будівельних конструкцій чинити опір дії високої температури, утворенню наскрізних тріщин та поширенню вогню в умовах пожежі і виконувати при цьому свої звичайні експлуатаційні функції. Вогнестійкість конструкцій будівель характеризується межею вогнестійкості.

По вогнестійкості будинку розділяють на п'ять ступенів в залежності від ступеня загоряння і межі вогнестійкості конструкцій. Найбільшу вогнестійкість мають будинку I ступеня, а найменшу – V ступеня. До будинків I, II й III ступенів вогнестійкості відносять кам'яні будинки, до IV – дерев'яні оштукатурені, до V – дерев'яні неоштукатурені будинки. У будинках I й II ступенів вогнестійкості стіни, опори, перекриття і перегородки неспалені. У будинках III ступеня вогнестійкості стіни й опори неспалені, а перекриття і перегородки важкозгораємі. Дерев'яні будинки IV і V ступенів вогнестійкості по протипожежних вимогах повинні бути не більш двох поверхів. Будівля, у якій знаходиться відділ матеріально-технічного забезпечення відноситься до споруд III ступеня вогнестійкості [2].

Відповідно до ПУЕ, приміщення поділяються на вибухонебезпечні (В- I, В-Ia, В-IБ, В-Iг, В-II, В-IIA) і пожежонебезпечні (П-I, П-II, П-IIA, П-III) зони. Дане приміщення відноситься до пожежонебезпечної зони класу П-IIA (це зони приміщень, в яких є тверді горючі речовини, які не здатні переходити в завислий стан).

У будівлі знаходяться наступні первинні засоби пожежогасіння:

- внутрішні пожежні крани;
- відра, кошми, лопати, пісок;
- вогнегасники(ВВ-3).

Всі вони характеризуються задовільним станом та готові до використання. Очевидним недоліком є те, що засоби та плани розташовані в будівлі на поверхах а не в приміщенні безпосередньо.

Можливими причинами виникнення пожеж у відділі матеріально-технічного забезпечення можуть бути: коротке замикання, недотримання правил експлуатації обладнання, газові розряди.

Засоби щодо профілактики виникнення пожеж полягають в наступному:

підтримка нормальної температури та вологості повітря; підтримка приміщення та обладнання в належному стані та чистоті; наявність первинних засобів пожежогасіння; проведення інструктажів з пожежної безпеки. У кожному корпусі будівлі є план евакуації людей при пожежі. Евакуаційні виходи

чітко позначені та мають належні розміри, знаходяться в місцях, доступних з будь-якої частини будівлі.

ВИСНОВКИ

В процесі виконання дипломної роботи було проведено аналіз існуючого алгоритмічного забезпечення та методів для вирішення задачі аналізу публічних текстових повідомлень з метою виявлення терористичних загроз.

Було проведено аналіз існуючих методів аналізу текстової інформації таких як інтент-аналіз, контент аналіз, феносемантичний аналіз, дискурсивний аналіз, наративний аналіз, морфологічний аналіз, синтаксичний аналіз та семантичний аналіз. Для кожного методу було розглянуто існуючи програмні реалізації та її особливості

Також були розглянуті сучасні методи кластеризації текстової інформації такі як: Concept Indexing,Complete Link, Group AverageLatent Semantic Analysis/Indexing, Scatter/Gather, Single Link, Self-Organizing Maps, Suffix Tree Clustering та були проаналізовані її переваги та недоліки.

Було обрано метод SOM та для вирішення поставленої задачі.

Під час виконання дипломної роботи були виконані всі поставлені задачі такі як:

– Розроблена архітектура розроблюваного ПЗ.

– Розроблений основний пакет діаграм.

– Розроблені програмні компоненти для аналізу публічних текстових повідомлень в мережі Інтернет.

– Протестоване розроблене програмне рішення.

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. О.В. Митина А.С Евдокименко Методы анализа текста: методологические основания и программная реализация. – Москва: «Вестник», 2017.

2. Alexa, M. Text Analysis Software: Commonalities, Differences and Limitations: The Results of a Review / M. Alexa, C. Zuell – Springer Netherlands, 2016. – P. 299-321.

3. Brown, G. Discourse analysis / G. Brown, G. Yule. – Cambridge, 2017.

4. Franzosi, R Quantitative Narrative Analysis (Quantitative Applications in the Social Sciences) / R. Franzosi. – Beverly Hills, CA: Sage, 2017. – 200 p.

5. Labov, W. Sociolinguistic patterns / W. Labov. — Pennsylvania: University of Pennsylvania, 2016.

6. Oren Eli Zamir. A Phrase-Based Method for Grouping Search Engine Results. University of Washington, Department of Science & Engineering. – 2018.

7. L.A Soshnikova, V.N. Tamashevich, G. Uebe, M. Sheffer. Multidimensional statistical analysis in economics – Uniti:Moscow, 2016.

8. А.М.Дубров, В.С.Мхитарян, Л.И.Трошин., Многомерные статистические методы. – Москва «Финансы и Статистика», 2016.

9. Michael W. Berry, Todd A. Letsche. Computational Methods for Intelligent Information Access. Department of Computer Science University of Tennessee Knoxville, TN 37996-13031 ./ Susan T. Dumains: Information science Research Group. Bellcore, 445 South Street Room 2L-371, Morristown, NJ 07962-1910. – 2018.

10. Susan T. Dumains, George W. Furnas, Thomas K.Landauer. Indexing by Latent Semantic Analysis. Bell Communications Research – 435 South St. Morristown, NJ 07960. – Richard Rashman:University Of Western Ontario., 2017.

11. Esko Ukkonen. On-line construction of suffix trees. – Department of Computer Science, University of Helsinki, PO Box 26 (Teollisuuskatu 23), FIN-00014 HUT, – Finland., 2017.

12. Alan Griffiths, H. Claire Luckhurst, and Peter Willett. Using Interdocument Similarity Information in Document Retrieval Systems. – Department of Information Studies, University of Sheffield, Western Bank, Sheffield S10 2TN, United Kingdom, 2017 p.365-373.

13. Voorhees E.M. Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. Information Processing and Management, 2016.

14. Douglass R.Cutting, David R.Karger, Jan O.Pedersen, John W.Turkey. Scatter/Gather: a Cluster-based Approach to Browsing Large Document Collections., 2016.

15. Dan Pelleg, Andrew Moore. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. School of Computer Science, Carnegie Mellon University, – Pittsburgh, PA 15213 USA., 2017.

16. С.А. Айвазян, В.С. Мхитарян. Прикладная статистика и основы эконометрики 3-е издание. Издательское объединение "ЮНИТИ", – Москва., 2017.

17. Eui-Hong (Sam) Han and George Kapyris. Concept Indexing. A Fast Dimensionality Reduction Algorithm with Application to Document Retrieval & Categorization. – University of Minnesota, Department of Computer Science. , 2017.

18. Teuvo Kohonen. Self-Organization of Very Large Document Collections: State of the Art. Helsinki University of Technology, – Neural Networks Research Center, PO Box 2200, FIN-02015 HUT,– Finland., 2017.

19. Jouko Lampinen and Erkki Oja. Clustering Properties of Self-Organizing Maps. – Lapperanta University of Technology, Department of Information Technology, PO box 20, SF-53851 Lapperanta, – Finland.,2017.

20. Т.Кохонен. Самоорганизующиеся карты.Адаптивные и интеллектуальные системы. – Москва, 2017.

21. Програмні проекти .Функціональні вимоги // https://project.dovidnyk.info/38funkcional_nye_trebovaniya, 15.10.2019.

22. Нефункціональні вимоги // https://habr.com/ru/post/415773/, 15.10.2019.

23. ISO/IEC 9126-1:2001 // https://www.iso.org/standard/22749.html, 16.10.2019

24. Ткачук М.В., Шеховцов В.А., Кукленко Д.В., Сокол В.Є. «Архітектури, моделі і технології програмного забезпечення інформаційно-керуючих систем» – Харків: НТУ «ХПІ», 2005. – 546 с.

25. Component diagram - діаграма компонентів // https://github.com/stankin/oop-2017/wiki/UML, 18.10.2019.

26. Python - короткий огляд мови // https://techrocks.ru/2019/01/21/about-python-briefly/, 19.10.2019.

27. PyCharm IDE // https://is42-2018.susu.ru/blog/2019/03/01/byil-zadan-vopros-chto-takoe-pycharm/, 19.10.2019.

28. Jquery //https://code.jquery.com/, 19.10.2019.

29. Бібліотеки та фреймфорки Python // https://proglib.io/p/50-python-projects/, 20.10.2019.

30. Apache – вільний веб сервер // https://habr.com/ru/hub/apache/, 19.10.2019.

31. Lenovo Y500 // https://ek.ua/LENOVO-Y500-59-359659.htm, 19.11.2019.

32. Маршрутизатор TP-LINK // https://secur.ua/marshrutizator-tp-link-tl-wr840n.htm, 19.11.2019.

33. Windows 10 Professional // https://soft.rozetka.com.ua/microsoft_fqc_10071_hav_00061/p4054370, 19.11.2019.

34. PyCharm Professional Edition // https://softlist.com.ua/catalog/product-jetbrains-pycharm/, 19.11.2019.

35. Керівник проекту// https://ua.trud.com/ua/salary/2.html?currency=UAH, 19.11.2019.

36. Програміст // https://tech.informator.ua/2019/07/24/skolko-poluchayut-programmisty-v-ukraine-i-gde-etomu-uchatsya/, 19.11.2019.

37. Тестувальник // https://www.work.ua/ru/salary-kharkiv, 19.11.2019

38. Інтернет-провайдер // http://triolan.com/articles.aspx?k=connections&lng=uk&reg=kh, 19.11.2019

39. «Хостинг Україна» // https://www.ukraine.com.ua/, 19.11.2019

40. Законодавство України // https://zakon.rada.gov.ua/laws/term/32604 , 13.11.2019.

41. Кібербезпека як важлива складова всієї системи захисту держави // http://www.mil.gov.ua/ukbs/kiberbezpeka-yak-vazhliva-skladova-vsiei-sistemi-zahistu-derzhavi.html, 13.11.2019.

42. Захист від кібератак // https://ucci.org.ua/press-center/expert-opinion/oleksandr-galushchenko-dlia-usvidomlennia-isnuiuchogo-stanu-it-skladovoyi-kompaniyi-neobkhidno-provesti-audit-informatsiinoyi-bezpeki, 13.11.2019.

43. Запорожець О.І., Михайлюк В.О., Халмурадов Б.Д. Цивільний захист. – Київ, 2017.

ДОДАТКИ