Ця сторінка містить вже знайоме користувачу фіксоване бокове меню, книгу с публікаціями, та новий елемент - кнопка з розділами журналів видання ОНАХТ.

Також необхідно відзначити, що було розроблено веб-додаток який задовольняє майже всі потреби для людей які будуть його використовувати з урахуванням предметної області.

4.Практичне значення отриманих результатів

Результатом дослідження є електронний науковий дайджест, який включає в себе реалізацію всіх поставлених вимог, рішення побудованих задач та усунення виявлених проблем. Розроблений електронний науковий дайджест є повноцінним, адаптивним і функціональним ПЗ, який може бути запровадженим для використання у роботі ОНАХТ для співробітників та здобувачів вищої освіти.

Список використаних джерел:

[1] Kwanya T., Stilwell C., Peter G. Library 3.0 Intelligent Libraries and Apomediation. Chandos Publishing, 2015. 190 p.
[2] B. van Wyk, H. Geldenhuys. Learn 3.0 Meets Library 3.0: A Case Study / International Conference on e-Learning. 2018. 480 p.
[3] A Beginner's Guide to Neural Networks and Deep Learning // Skymind: [Веб-сайт]. URL: https://skymind.ai/wiki/neural-network (дата звернення: 07.11.2019).
[4] Object Relational Tutorial [Електронний ресурс] – Режим доступу до ресурсу: https://docs.sqlalchemy.org/en/latest/orm/tutorial.html - Назва з екрану. - Дата перегляду: 15.04.2018.
[5] The Architecture of Open Source Applications (Volume 2) SQLAlchemy [Електронний ресурс] – Режим доступу до ресурсу: http://aosabook.org/en/sqlalchemy.html - Назва з екрану. - Дата перегляду: 15.04.2018.
[6] Науменко Д. HTML, CSS, PHP, JavaScript, SQL – что и зачем? [Електронний ресурс] / Дмитрий Науменко – Режим доступу до ресурсу: http://codeharmony.ru/materials/125.
[7] Язык HTML 5 – преимущества и недостатки [Електронний ресурс] – Режим доступу до ресурсу: https://seodirection.ru/html5/.

References:

[1] T. Kwanya, C. Stilwell, and P. G. Underwood, *Library 3.0: intelligent libraries and apomediation*. Amsterdam: Chandos Publishing is an imprint of Elsevier, 2015.
[2] B. van Wyk, H. Geldenhuys, “Learn 3.0 Meets Library 3.0: A Case Study”. International Conference on e-Learning, 2018.
[3] “A Beginner's Guide to Neural Networks and Deep Learning,” *Pathmind*. [Online]. Available: https://skymind.ai/wiki/neural-network. [Accessed: 07-Nov-2019].
[4] “SQLAlchemy 1.3 Documentation,” *Object Relational Tutorial - SQLAlchemy 1.3 Documentation*. [Online]. Available: https://docs.sqlalchemy.org/en/latest/orm/tutorial.html. [Accessed: 15-Apr-2018].
[5] “SQLAlchemy,” *The Architecture of Open Source Applications (Volume 2): SQLAlchemy*. [Online]. Available: http://aosabook.org/en/sqlalchemy.html. [Accessed: 15-Apr-2018].
[6] “HTML, CSS, PHP, JavaScript, SQL – chto I zachem?,” [Online]. Available: http://codeharmony.ru/materials/125. [Accessed: 10-Mar-2019].
[7] “SEO Direction,” *Jazik HTML 5 – preimushestva I nedostatki*. [Online]. Available: https://seodirection.ru/html5/. [Accessed: 05-Mar-2019].

УДК 004.383.2:004.738.5:004.771

ПРИНЦИПИ ПОБУДОВИ ХМАР ТЕГІВ ДАНИХ

Хараш К. М.[1], Ольшевська О. В.[2], Титуренко Ж. А.[3]

[1,2,3]Одеська національна академія харчових технологій, Одеса, Україна
ORCID: [2] http://orcid.org/0000-0002-4512-3915, [3] http://orcid.org/0000-0001-6774-1688
E-mail: [2] olshevska.olga@gmail.com, [3] janettrnk@gmail.com

Copyright © 2018 by author and the journal “Automation of technological and business - processes.
This work is licensed under the Creative Commons Attribution International License (CC BY).
http://creativecommons.org/licanses/by/4.0

DOI: https://doi.org/10.15673/atbp.v12i1.1699

Анотація. *Розглядаються механізми візуалізації для побудови термінологічних хмар. На прикладі JSON, HTML, CSV, XLSX, XML, TXT наведений перелік типів файлів та ресурсів. Проаналізовано можливості добування та збереження вхідних даних. Проведено дослідження аналогічних систем, на основі якого було обрано два оптимальні типи файлів, а саме CSV та TXT. Виявлено підхід формування списку ключових слів для наукових публікацій або виокремлення провідної тематики різних текстів. Встановлено, що у разі необхідності обробки великих текстів спільної спрямованості, якими наприклад можуть бути літературні твори, наукові статті, судові вироки тощо, достатнім буде використання малих веб-додатків для побудови тегових хмар. Тегові хмари на основі алгоритму k-середніх здатні досить ефективно виявити ключові поняття, найбільш уживані слова та провідні концепції. При порівнянні між собою форматів CSV та TXT, було підтверджено, що швидкість обробки залежить скоріше від об'єму вхідної інформації, ніж від структури файлу. Звідси, можна стверджувати, що використання одного або іншого формату зумовлено вибором користувача. Проведено аналіз з якого відзначено, що формат CSV потребує верхнього рядка, в якому вказують атрибути. Для більшої коректності подальшого аналізу, атрибути слід вказати і формувати кожний наступний рядок даних строго по черзі. Така незначна особливість структури допомагає досліднику орієнтуватися серед набору текстової інформації, а при подальшій обробці перший рядок можна не враховувати. На відміну від попереднього формату, формат TXT не потребує формування першого рядка атрибутів. Це ускладнює візуальне сприйняття наявної інформації. Не рекомендовано вводити атрибути самостійно, в подальшому при обробці це буде впливати на коректність результатів кластеризації в негативний бік.*

Abstract. *Visualization mechanisms for constructing terminological clouds are considered. An example of JSON, HTML, CSV, XLSX, XML, TXT is a list of file types and resources. Possibilities of extraction and storage of input data are analyzed. Studies of similar systems were performed, on the basis of which two optimal file types were selected, namely CSV and TXT. The approach of forming a list of keywords for scholarly publications or distinguishing the leading topics of different texts was discovered. If the need is to handle large collaborative texts, such as literary works, scientific articles, judgments, etc., it will be sufficient to use small web applications to build tag clouds. K-mean tag clouds are able to effectively identify key concepts, most commonly used words, and leading concepts. When comparing CSV and TXT formats, it was confirmed that the processing speed depends more on the amount of input than on the file structure. Hence, it can be argued that the use of one or the other format is conditioned by the user's choice. An analysis has been conducted that noted that the CSV format needs an upper line that specifies attributes. For the sake of correctness of the further analysis, the attributes should be specified and formed each successive row of data in strict order. Such a slight feature of the structure helps the researcher to navigate among the set of textual information, and in further processing the first line can be ignored. Unlike the previous format, the TXT format does not require the formation of the first line of attributes. This complicates the visual perception of the information available. It is not recommended to enter the attributes yourself, in the future, when processing it will affect the correctness of the clustering results in the negative.*

Ключові слова: Термінологічні хмари, наукометричні системи, хмари тегів, JSON, HTML, CSV, XLSX, XML, TXT, аналітика великих даних, кластерний аналіз, кластерізація, алгоритм Дейкстри, манхеттенська відстань, відстань міських кварталів, регресійні дерева (CART), евклідова відстань, алгоритм DBSCAN, алгоритм k-means.

Keywords: Terminological clouds, scientometric systems, tag clouds, JSON, HTML, CSV, XLSX, XML, TXT, Data Mining, bag of words, bag of terms, cluster analysis, clustering, Dijkstra's algorithm, Manhattan distance, city block distance, regression trees (CART), Euclidean distance, DBSCAN algorithm, k-means algorithm.

Актуальність теми. В науковій, як і в освітній сфері неможливо впоратись без візуального представлення матеріалу, що досліджується або викладається. Науковці оперують великим набором вихідних даних, або результатів досліджень, які варто кластерізувати і представити у більш адаптованому вигляді. Візуалізація даних є одним з методів опрацювання наукових доробків. Методи візуалізації потрібно впровадити глибше та розширити охоплювані області.

Метою даної роботи є дослідження принципів побудови термінологічних хмар та моделювання системи обробки даних, яка може представити велику кількість даних візуально у вигляді понятійних кластерів або тегових хмарових структур. Система повинна базуватися на досвіді та додатках, які вже використовуються в даній області знань. Окрім того, необхідно змоделювати не лише процес обробки та кластеризації даних, а й способи його візуального представлення.

Візуалізація може бути досягнена шляхом впровадження обробки розрізнених даних для формування термінологічних хмар. Термінологічні хмари являють собою структури схожі на хмари тегів, які часто можна зустріти на веб-ресурсах. Термінологічні хмари є допоміжним інструментом аналізу відносно малих обсягів даних, однак тяжко сприйманих людиною. Перевага таких хмар у меншій кількості ресурсів, необхідних для побудови у порівнянні з повноцінними графічними моделями. Отже, термінологічні хмари є більш доцільним методом візуалізації у сфері навчання та локальних дослідів у малих організаціях. Аби дослідити термінологічні хмари с точки зору принципів їх створення, необхідно розібрати поняття кластерного аналізу даних.

Вибір, які елементи врахувати у формуванні термінологічної хмари, а які пропустити, лежить на користувачеві. Реалізація такої структури включає синтаксичну фільтрацію та опущення непотрібних елементів. Непотрібними елементами можна вважати знаки пунктуації, числа або похідні форми конкретних термінів [1].

Мета і задачі дослідження. Метою роботи є дослідження та моделювання системи обробки даних, яка реалізує алгоритм побудови термінологічних хмар, адаптованої до використання у малих організаціях, учбових установах та з навчальною метою. Система повинна базуватися на досвіді та додатках, які вже використовуються в даній області знань.

Для досягнення поставленої мети в роботі поставлені і розв'язані наступні задачі: визначити ключові концепції для побудови термінологічних хмар; проаналізувати ключові концепції та відокремити провідний механізм побудови термінологічних хмар; визначити механізми візуалізації для побудови термінологічних хмар; визначити перелік типів файлів та ресурсів на прикладі JSON, HTML, CSV, XLSX, XML, TXT; дослідити можливості форматів та вивчити аналоги; побудувати тестовий файл для завантаження його на ресурс; провести порівняльний аналіз розробленого файлу з аналогічними; обрати найбільш доцільний тип файлу.

1. Теоретичні складові

До базових методів Data Mining відносять алгоритми, засновані на переборі, елементи теорії статистики, основний недолік яких – усереднення значень, яке може привести до втрати інформативності даних [3-4]. А наразі в технології Data Mining використовують методи нечіткої логіки, генетичні алгоритми, нейронні мережі та ін.

Робота [5] від самого початку акцентує увагу читача на тому, що обробка великих даних являє собою велику проблему. Таке ставлення автор пояснює тим фактом, що здатність створювати дані зростає значно швидше ніж здатність їх аналізувати. Комп'ютерні додатки сильніше наближуються до росту об'ємів даних для аналізу. Тенденція до потреби аналітики даних разом із більшою швидкістю обробки привели до виникнення такого поняття як «Аналітика великих даних».

Існують три види задач, які тісно пов'язані з проблемою великих даних. Перша задача полягає у зберіганні та керуванні цими даними. Друга ставить питання про те, як можна організувати неструктуровані дані. Третя є найцікавішою і полягає саме у аналізі великих даних. Як можна на основі Великих Даних будувати прості наочні звіти, будувати та впроваджувати поглиблені прогностичні моделі? У цій саме роботі автор намагається відповісти на поставлені питання. Зокрема, питання аналізу великих даних потребує використання нових технологій комп'ютерної графіки, середовищ віртуальної та доповненої реальності. А тому виникає необхідність проведення поглиблених досліджень не лише з точки зору інформаційних систем та математики, а й у області когнітивної психології. Вивчення особливостей сприйняття інформації людиною дозволить ліпшим чином адаптувати аналітичні системи саме під потреби окремої людини, що дозволить опрацьовувати більше даних і робити це значно швидше, а отже, і прискорювати технологічний прогрес. Тема когнітивної психології є безумовно цікавою, однак не стосується даного дослідження. З точки зору інформаційних систем можна зробити висновок, що проблема обробки великих даних залишиться невирішеною до кінця доти, доки не з'явиться система, здатна адаптуватися до перманентного збільшення об'єму вхідних даних. Це питання лежить вже у площині штучного інтелекту та на стику біологічної свідомості та кремнієвих обчислювальних потужностей [2].

Ще одна проблема, яка виникає в процесі індексування документів полягає у виборі структури списку ключових слів. Питання полягає в тому, чи повинен такий список складатися виключно з одиночних слів або може включати в себе і складні вирази. Справедливим є факт, що складні вирази краще описують предметну область і досліджуване питання, однак морфологічно обробити такі ключові слова важче. На таку проблему звернув увагу автор у роботі [12] і продемонстрував практичні переваги кластеризації документів на базі ключових словосполучень.

Зазвичай така проблема вирішується використанням засобу аналізу, який засновано на тезаурусі досліджуваної предметної області. Однак такий підхід має великий недолік – з використанням тезаурусу неможливо індексувати тексти вільної тематики. Автор має рацію, однак крім такого зауваження, слід звернути увагу ще й на той факт, що категоризація інформації наукометричних баз даних значно ускладнюється. Це зумовлено тим, що бази містять у собі наукові доробки з різних наукових областей. Наскільки точним буде загальний тезаурус або наскільки релевантною буде така оцінка?

Оскільки слов'янські мови більш варіативні за формами слів, ніж, наприклад, мови германської групи, виникає питання як категоризувати спільнокореневі ключові слова. Одним із варіантів є ведення наукової роботи виключно одною мовою у всій світовій науковій спільноті, однак таке вирішення не є правильним.

Логічно кластеризація текстових даних поділяється на два етапи. На першому етапі текстові представлення документів переводять у векторні, а на другому до отриманих векторних представлень застосовують методи кластеризації, які базуються на пошуку відстані між векторами. У статті [6] автор розглядає такі методи на основі «пакунку слів» (bag of words), «пакунку термінів» (bag of terms), тематичного моделювання, векторних систем, семантичної кластеризації та оглядає міри оцінки ефективності. Досвід автора показує, що найкращим методом кластеризації даних є алгоритм k-means для всіх наборів даних при виконанні деяких умов. Окрім того, автор аналізував кластеризацію на наборах анотацій до наукових публікацій і на повних текстах публікацій, що показало

більшу ефективність у випадку повних текстів. Такий висновок є логічним, адже більше заглиблення в тему дає більше даних для аналізу, а отже кластеризація надасть точніший результат с точки зору глибини вивчення

2. Дослідження проблеми:

Було проаналізовано основні властивості найбільш поширених форматів файлів JSON, XML, HTML, CSV, XLSX, TXT:

JSON – формат, що має структуру ключ-значення. Є найбільш поширеним на сьогодні, широко використовується для обміну даними, у випадках, коли необхідно надіслати запит і отримати відповідь. Наприклад, в обміні REST-запитами. Будь-яка сучасна мова програмування здатна оброблювати такий формат завдяки простоті реалізації. Файли типу JSON також легкі для сприйняття людиною, однак за умови наявності кольорової розмітки. Інакше, редагування файлу значно ускладнюється. До недоліків варто віднести неможливість потокової обробки даних.

XML – формат файлів, заснований на парах тег-значення. Був значно поширений раніше, але зараз з кожним днем все більше витісняється іншими форматами. Це зумовлено наявністю доволі складної граматики мови, яку вже не можна спростити. Головним чином, XML використовують у системах, які були спроектовані і впроваджені на початку століття, через складність перепроектування. Оскільки, складність синтаксису XML перевищує аналогічний показник JSON, то і швидкість обробки таких файлів менша, що є однією з причин втрати популярності форматом. Формат не відрізняється гнучкістю. Також, до недоліків можна віднести складність редагування та візуального сприйняття людиною і відсутність потокової обробки даних.

HTML являє собою стандартну веб-сторінку. Такий формат майже не використовується для передачі даних, а для обробки такого типу файлів існують веб-браузери.

XLSX заснований на форматі Open XML і за своєю суттю є електронною таблицею. Дані у файлах такого типу прив’язані до комірок, координати яких складаються з номеру рядка та стовпця. Формат використовують у випадках, коли необхідно зберігати саме таблиці з даними, а також для автоматизації розрахунків та спрощеної обробки даних. Для обробки використовуються спеціалізовані офісні програми.

CSV – текстовий файл, у якому міститься таблична інформація. Дуже широко використовується для автоматизованого заповнення даних, виконання операцій на основі певних даних, побудови графіків та інших дій, які імітують користувацьке введення.

На відміну від XLSX такий файл можна продивитися у будь-якому стандартному редакторі без втрати дружнього до користувача вигляду. Рядки таблиці у CSV розділяють переносом на новий рядок, а стовпці – роздільниками, якими можуть слугувати кома, пробіл, табуляція, або крапка з комою. Формат дуже легкий для людського сприйняття, легко редагується, не має жорстких обмежень за суттю. Всі обмеження задає конкретна програма, яка оброблює такий файл.

TXT – файл для зберігання текстових даних. Є найбільш поширеним для зберігання різноманітної інформації. Не має жодних обмежень відносно структури, не має синтаксису. Користувач вільний обирати як такий файл структурувати, що у ньому зберігати та чим оброблювати. Доступний до обробки всіма мовами програмування, а прочитати файли такого формату можна будь-яким текстовим редактором. Недоліком є неможливість автоматизованої структуризації, якщо необхідно зберігати конкретний вид даних, однак за умови уважного підходу до формування та структуризації даних у такому файлі, в даному розширенні можна зберігати будь-яку інформацію так, як зручно користувачеві.

Аби дослідити термінологічні хмари с точки зору принципів їх створення, необхідно розібрати поняття кластерного аналізу даних.

Кластерний аналіз або кластеризація – це задача розбиття множини об’єктів на групи, що називають кластерами. В кожній групі мають опинитися об’єкти найбільш схожі між собою. Так само у різних групах мають опинитися об’єкти найбільш віддалені один від одного.

Однією з цілей кластеризації є знаходження внутрішніх зв’язків між даними шляхом визначення кластерної структури. Це може бути використано для вирішення задачі стискання даних або виявлення ступеню новизни.

Складність кластеризації та її відмінність від класифікації полягає у тому, що перелік кінцевих груп не заданий наперед, а визначається у ході роботи алгоритму.

Використання кластерного аналізу можна звести до таких етапів [7]:

- Виділення множини об’єктів для кластеризації.
- Виділення множини змінних, за якими об’єкти будуть оцінюватися.
- Виявлення ступені схожості об’єктів між собою.
- Використання методу кластерного аналізу для створення кластерів.
- Представлення результатів роботи.

Постановка задачі:

Нехай множина об'єктів $I=\{x_i\}_{i=[1, \dots, n]}$, представлених набором атрибутів за формулою (1):

$$x_i = \{t_1^i, t_2^i, \dots, t_m^i\}_{i=1}^n \quad (1)$$

t_ν^i де приймає значення із заданої множини *T*. Задача кластеризації полягає у побудові множини *C*, представленою формулою (2) і відображення заданої множини об'єктів на множину кластерів.

$$F : \mathfrak{I} \to \mathbf{C} \tag{2}$$

Кластер містить об'єкти із загальної множини, схожі між собою за заданим критерієм, що записується формулою (3):

$$x_i \in C_\nu, x_j \in C_\nu \Rightarrow d(x_i, x_j) < \varepsilon, \tag{3}$$

де *d* – ступінь схожості між об'єктами, а ε – найбільше значення порогу, що формує кластер [8].

Аби розв'язати таку задачу потрібно вміти визначати ступінь схожості об'єктів. Для цього необхідно визначити критерії, за якими буде проводитися порівняння – спільні характеристики для всіх об'єктів. Далі ці характеристики потрібно нормалізувати. Аби програма могла обробити вхідну інформацію, потрібно цю інформацію звести до єдиного вигляду. Наприклад, вписати всі дані у певний діапазон величин. І останній етап – знайти «відстань між об'єктами».

Оскільки способів знаходження «відстані» між об'єктами (або ступеня їх відмінності) існує багато, важливо заздалегідь визначити мету кластеризації даних і, в залежності від мети, обирати спосіб знаходження бажаної величини. Серед таких способів варто відзначити нижче приведені.

Вираховується за допомогою формули Піфагора Евклідову відстань, що представляє собою відстань між об'єктами у евклідовому просторі. Іншими словами це відстань між двома об'єктами у тривимірному просторі. Формула являє собою корінь суми квадратів різності. Математичне представлення записується формулою (4):

$$\sqrt{\sum_{k=1}^{n}(p_k - q_k)^2} \tag{4}$$

де *p* та *q* – точки для яких шукається відстань.

З цим методом можна поєднати і квадрат евклідової відстані. Такий метод обчислення використовують аби надати більшу вагу сильно віддаленим один від одного об'єктам.

Алгоритм Дейкстри для знаходження найкоротшої відстані це алгоритм побудований на теорії графів. Велика перевага даного алгоритму у його наочності. Алгоритм Дейкстри покликаний знайти найкоротшу відстань від однієї вершини графа до всіх інших.

В першу чергу треба створити набори даних пар відстань-«відвідано». Де відстань – натуральне число, а значення відвідуваності є булевим показником. Початкова відстань є нульовою і приймається за поточну. У ході роботи алгоритм ітеративно вимірює відстань між поточним вузлом та сусідами і, якщо нова відстань більше за початкову, поточна відстань оновлюється більшим показником. Ітерація завершується оновленням значення «відвідано». Наступна ітерація починається з вибору нового поточного вузла – з найменшим значенням відстані і нульовим значенням відвідуваності. Знову рахується відстань від поточного вузла до сусідніх, і якщо вона більша за фактичну, значення відстані оновлюється більшим показником. У загальному вигляді алгоритм пошуку найкоротшого шляху можна представити таким чином:

Крок 1. Створити набори даних «відстань» та «відвідано»

Крок 2. Ініціалізувати набори

Крок 3. Обрати початковий вузол і присвоїти йому нульове значення відстані, вважати даний вузол за поточний

Крок 4. Розрахувати відстань від поточного вузла до сусідніх, оновити значення відстані якщо нове значення більше за поточне

Крок 5. Змінити значення «відвідано» для вузла

Крок 6. Змінити поточний вузол на не відвіданий з найменшою відстанню

Крок 7. Повторювати доти, доки всі вузли не стануть відвіданими [9].

Манхеттенська відстань або відстань міських кварталів. Найчастіше результатом пошуку відстані даним методом стають ті ж величини, що і у випадку використання методу звичайної евклідової відстані, однак вага окремих великих різниць у даній формулі зменшено, адже відстань не підноситься у квадрат. Така відстань являє собою середнє значення різності координат. Математично записується формулою (5) де *p* та *q* – вектори.

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i| \tag{5}$$

Всі алгоритми кластеризації можна класифікувати кількома способами. У першому випадку алгоритми можна поділити на ієрархічні та неієрархічні. Різниця полягає у тому, що ієрархічні алгоритми не враховують кінцевої кількості кластерів, в той час, як неієрархічні одним із вхідних параметрів мають умову зупинки роботи та кількості кластерів. Неієрархічні алгоритми засновані на тезі: зв'язок між ознаками схожості визначається кінцевою кількістю прихованих змінних.

Ієрархічні алгоритми визначають кінцевий результат у ході роботи. У свою чергу, такі алгоритми розділяються на агломеративні та дивізимні. Агломеративні будують кластери за принципом зменшення кількості кластерів шляхом з'єднання схожих елементів. Дивізимні, навпаки, розділяють існуючі великі кластери на менші.

У другому випадку алгоритми поділяються на чіткі та нечіткі. Чіткі алгоритми кожному об'єкту вибірки ставлять у відповідність номер кластеру, тобто кожний об'єкт належить лише одному кластеру. Нечіткі алгоритми кожному об'єкту виборки ставлять у відповідність сукупність значень, які вказують ступінь відношення об'єкта до кластерів, тобто кожен об'єкт має відношення до кожного кластеру з певною вірогідністю [10].

Варто розглянути деякі алгоритми. Найпопулярнішим є метод k-means, який покликаний опрацьовувати динамічні дані. Ідея методу полягає у тому, що на початку роботи обирається певна кількість k елементів з початкової множини об'єктів. Потім всі об'єкти розбивають на таку ж кількість груп і знаходять центри знайдених кластерів. Дії повторюються ітераційно, доки не буде досягнуто заданої кількості ітерацій.

Кроки алгоритму можна визначити наступним чином:

Крок 1. Задати кількість груп k, яка відповідає кількості початково обраних елементів. Ці довільні об'єкти стануть центрами кластерів.

Крок 2. Пронумерувати групи за мінімальною нормальною відстанню між об'єктом і центром відповідного кластеру.

Крок 3. Перерахувати центри нових кластерів.

Крок 4. Повторити кроки 2 і 3 доті, доки центри не стабілізують свої значення.

Головна перевага даного алгоритму полягає у його простоті та ефективності. Через простоту обчислень, алгоритм працює досить швидко навіть з великим обсягом даних. До недоліків варто віднести чуттєвість до початкового вибору центральних елементів кластерів, якщо вони підібрані не вірно, результати роботи будуть суттєво спотворені.

Алгоритм DBSCAN. У основі методу лежить об'єднання деяких об'єктів відповідно до їх внутрішньогрупового з'єднання. Аби коректно провести процедуру кластеризації, необхідно задати критерії, у відповідності до яких об'єкти будуть об'єднуватися у кластери.

Для визначення густини об'єктів для певної точки X рішучими відіграють два параметри. Перший параметр це радіус наближеності α, який характеризує ступінь наближеності точки X. Тоді множина сусідніх точок для X включатиме в себе такі точки, де відстань між об'єктами буде менша або дорівнюватиме радіусу наближеності між об'єктами певної виборки. Математично функція записується формулою (6):

$$\mathrm{dist}(X, f_i) \le \alpha, \quad \left(i = \overline{1,n}\right) \tag{6}$$

де функція *dist(var1, var2)* є відстанню між об'єктами виборки. Дану відстань можна обчислити як евклідову відстань.

Другий параметр це найменша кількість точок, які найближче розташовані до даної точки відповідно радіусу α. Алгоритм виглядає так:

Крок 1. Виділити точки з множини D, які є оточеними.

Крок 2. Для кожної точки визначити:

чи належить дана точка до кластеру

чи є дана точка оточеною точкою

Крок 3. Якщо точка є оточеною, тоді всі об'єкти, які є досяжними, з'єднати у новий кластер. Інакше, якщо точка не є оточеною і не є досяжною від іншої точки, вважати такий об'єкт викидом.

Даний алгоритм на сьогодні є достатньо ефективним і перспективним. До переваг алгоритму можна віднести його нечуттєвість до викидів, це означає, що у процесі кластеризації всі викиди одразу виносять в окремий кластер із заданою заздалегідь позначкою. Окрім того, метод не потребує наперед заданої кількості кластерів і його використання дозволяє опрацьовувати кластери будь-якої форми. До недоліків можна віднести трудомісткість пошуку необхідних параметрів для коректної роботи алгоритму [11].

На основі отриманих даних було прийнято рішення апробувати модель обраного рішення. З огляду на це, була побудована візуальна модель програмними засобами додатку Orange (рисунок 1).

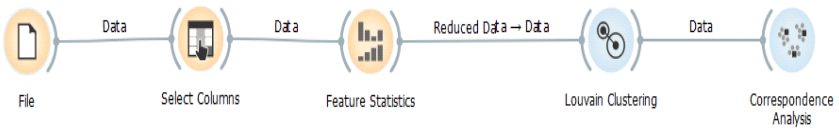

Рис. 1 – Візуальне представлення моделі обраного рішення

Дана модель відображує дії та структури даних, над якими вони будуть здійснюватися. Після того, як вхідні дані буде зчитано, вони направляються до вихідного каналу. Модель здатна зчитувати дані форматів XLSX, TXT, CSV.

3. Проектування системи

Термінологічні хмари аналогічні за принципом побудови до звичних всім тегових хмар, які часто зустрічаються на веб-ресурсах. Відмінність полягає у природі вхідних даних. Для даного проекту доцільним є використання алгоритму

k-means для обробки вхідних даних, оскільки алгоритм є достатньо швидким для досягнення поставленої у завдані мети і одночасно сбалансованим с точки зору трудомісткості.

Варто описати процес створення термінологічної хмари:

1) Задати кількість бажаних кластерів.

2) Для кожного кластеру визначити центр мас, або ключове слово, навколо котрого бущде створюватися кластер. У даному випадку таким центром буде виступати частота зустрічності терміну серед ключових слів у наукових публікаціях.

3) Розрахувати відстань від кожного центру мас до кожного терміну з використанням функції евклідової відстані.

4) Сгрупувати терміни у кластери. Тобто, визначити який кластер є найближчим до певного терміну та додати даний термін у відповідний кластер.

5) Перерахувати центри мас для стабілізації кластеру. Для цього потрібно визначити середнє значення частоти використання тих термінів, які увійшли до кластеру.

6) Пропускати кластери, до яких не увійшов жоден термін.

7) Ітеративно повторювати дії до моменту, коли центри мас перестануть змінюватися.

У загальному випадку ступінь розповсюдженності терміну у вхідному наборі даних сильно впливає на вагу терміна в кінцевій хмарі. Якщо взяти за приклад наукову публікацію, частота терміну відповідатиме кількості вживання терміну та його похідних форм у роботі. На основі такої характеристики формують набір ключових слів для кожної публікації.

Так, на малих наборах даних кількість врахованих елементів зростає відповідно до експоненційного закону розподілення. Якщо розмір вхідного набору зростає, є сенс використати логарифмічне представлення.

Вибір, які елементи врахувати у формуванні термінологічної хмари, а які пропустити, лежить на користувачеві. Реалізація такої структури включає синтаксичну фільтрацію та опущення непотрібних елементів. Непотрібними елементами можна вважати знаки пунктуації, числа або похідні форми конкретних термінів. У рамках даної роботи всі такі елементи не приймають участі у розрахунках.

4. Результати проведенного експерименту

У ході дослідження було встановлено оптимальний формат для зберігання вхідних даних і завантаження їх у програмний додаток для обробки. Серед більшості найчастіше використовуваних форматів для зберігання і передачі даних, було обрано два як найбільш доцільні, а саме формати CSV та TXT. Такий вибір зумовлено простотою створення, обробки та редагування. Такі файли не потребують спеціального ПЗ, а форматування даних проводити простіше, оскільки відсутні строгі синтаксичні обмеження.

Другим етапом було виявлено два підходи до реалізації візуального представлення розрізнених текстових даних у вигляді термінологічних хмар або термінологічних кластерів.

Перший підхід полягає у використанні веб-додатків на базі алгоритмів кластеризації k-means або k-середніх. Такі веб-додатки здатні формувати термінологічні хмари, або тегові хмари, або кластерні представлення на основі окремих слів.

Робота таких веб-додатків базується на виокремленні найбільш уживаних слів впродовж аналізованого тексту. Частота вживання візуально відображується розміром шрифту написання слова у вихідній хмарі. Для дослідження провідної тематики тексту такого аналізу достатньо. Недоліком такого підходу є неможливість його використання для більш поглибленого аналізу. Оскільки такі системи дають лише асоціативний наближений результат.

Для визначення зв'язків між кластерами варто використовувати другий з виявлених підходів. Такий підхід базується на використанні спеціалізованого ПЗ, яке здатне на основі вхідних даних не лише формувати кластери, а й встановлювати зв'язки між ними. Окрім того, таке ПЗ враховує особливості сприйняття інформації людиною і здатне формувати візуалізоване представлення у вигляді кольорових плям, або так званих мап щільності.

Останній підхід ефективно себе проявляє для обробки відносно великих, і не пов'язаних спільною тематикою, наборів даних. Більш розширений функціонал та засоби налаштування дають досліднику змогу обирати параметри, за якими буде проведено аналіз, кількість вихідних кластерів тощо. Суттєвою перевагою перед використанням відносно простих веб-додатків на базі алгоритмів k-середніх є можливість формувати кластери, спираючись не лише на слова, а й на словосполучення. У подальшому це надає змогу проводити більш точний та поглиблений аналіз предметної області.

5. Наукова новизна проекту

Для досягнення поставленої мети було визначено ключові концепції для побудови термінологічних хмар. Проаналізовано ключові концепції та виокремлено провідний механізм побудови термінологічних хмар. Також було визначено механізми візуалізації для побудови термінологічних хмар. Дослідження саме візуалізації даних необхідно для кращого розуміння оптимальної побудови структури. Дана задача включала в себе і дослідження особливостей сприйняття інформації, спрямованість на які дозволить налаштувати систему у відповідності до потреб цільової аудиторії.

Для підготовки до практичної реалізації було визначено перелік типів файлів та ресурсів на прикладі JSON, HTML, CSV, XLSX, XML, TXT. Для коректної постановки експерименту, вхідні дані необхідно було зберігати або добувати

динамічно. Оскільки, динамічний спосіб добування даних перевантажує систему, формувати початковий набір даних було вирішено сторонньо.

За результатами дослідження можна стверджувати, що найбільш придатними до швидкої обробки неструктурованих даних є файли у форматі CSV та TXT. Різниця у швидкості обробки є несуттєвою.

Список використаних джерел

[1] Современные методы создание мультипредметных веб-ресурсов на базе визуализации и обработки формализованной семантики / В. В.Диковицький, П. А. Ломов, Р. Р. Сепеда-Еррера, М. Г. Шишаєв. // Вісник Кольського наукового центру РАН. – 2011. – С. 63–73.
[2] Кислова О. Н. Интеллектуальный анализ данных: история становления термина / О. Н. Кислова. // Український соціологічний журнал. – 2011. – №1. – С. 83–94.
[3] Нечипорук Д. В. Особенности технологии Data Mining / Д. В. Нечипорук. – 2017. – №1.
[4] Барсегян А. А. Анализ данных и процессов / А. А. Барсегян. – Санкт-Петербург: БХВ-Петербург, 2009. – 512 с.
[5] Malyarova M. Analysis and visualization "Big Data": why "Big data" is a "Big Problem"? / Maria Malyarova. // International Scientific review. – 2016. – С. 66–68.
[6] Пархоменко П. А. Обзор и экспериментальное сравнение методов кластеризации текстов / П. А. Пархоменко, А. А. Григор'єв, Н. А. Астраханцев. // Труды института системного программирования РАН. – 2017. – С. 161–188.
[7] Библиометрические инструменты в помощь исследователю. Ключевые слова. Часть третья: VOSviewer [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://www.eco-vector.com/single-post_lutay4.
[8] Clusterization [Електронний ресурс] – Режим доступу до ресурсу: http://pzs.dstu.dp.ua/DataMining/cluster/index.html.
[9] David P. Graphs & Paths: Dijkstra. [Електронний ресурс] / Pynes David. – 2018. – Режим доступу до ресурсу: https://towardsdatascience.com/graphs-paths-dijkstra-4d8b356ad6fa.
[10] Обзор алгоритмов кластеризации данных [Електронний ресурс]. – 2010. – Режим доступу до ресурсу: https://habr.com/ru/post/101338/.
[11] Чапланов А. П. Кластеризация с помощью алгоритмов DBSCAN / А. П. Чапланов, О. Б. Чапланова. // Системи обробки інформації. – 2006. – №9. – С. 82–85.
[12] Барахнин В. Б. Кластеризация текстовых документов на основе составных ключевых термов / В. Б. Барахнин, Д. А. Ткачев. // Вестник Новосибирского государственного университета. Серия: Информационные технологии. – 2010.

References

[1] V. V. Dykovytskyi et al., “Sovremennye metody sozdanye mul’typredmetnykh veb-resursov na baze vyzualyzatsyy y obrabotky formalyzovannoy semantyky,” (in Russian) *Visnyk Kol’s’koho naukovoho tsentru RAN*, pp. 63–73, 2011
[2] O. N. Kyslova, “Intellektual’nii analiz dannykh: istoriia stanovleniia termina,” (in Ukrainian) *Ukrayins’kii sotsiolohichnyi zhurnal*, no. 1, pp. 83-94, 2011.
[3] D. V. Nechyporuk, “Osobennosti tekhnolohii Data Mining,” *Molodoi issledovatel Dona*, no. 1(4), 2017.
[4] A. A. Barsehyan, *Analiz dannykh i protsessov*, (in Russian), Sankt-Peterburh, Russia: BKhV-Peterburh, 2009, 512 p.
[5] M. Malyarova, “Analysis and visualization "Big Data": why "Big data" is a "Big Problem"?,” *International Scientific review*, pp. 66-68, 2016.
[6] P. A. Parkhomenko et al., “Obzor i eksperimentalnoe sravnenie metodov klasterizatsi tekstov,” *Trudy instituta sistemnogo programmirovaniia RAN*, pp. 161-188, 2017.
[7] *Bibliometricheskie instrumenty v pomoshch issledovateliu. Kliuchevye slova. Chast tretia:* VOSviewer (2018) [Online]. Available: https://www.eco-vector.com/single-post_lutay4.
[8] *Clusterization* [Online]. (Dniprovsk State Technical University). Available: http://pzs.dstu.dp.ua/DataMining/cluster/index.html.
[9] P. David. (2018). *Graphs & Paths: Dijkstra* [Online]. Available: https://towardsdatascience.com/graphs-paths-dijkstra-4d8b356ad6fa.
[10] *Obzor algoritmov klasterizatsii dannykh*. (2010). [Online]. Available:https://habr.com/ru/post/101338.
[11] A. P. Chaplanov and O. B. Chaplanova., “Klasterizatsiia s pomoshchiu algoritmov DBSCAN,” *Systemy obrobky informatsii*, no. 9, pp. 82-85, 2006.
[12] V. B. Barakhnyn and D. A. Tkachev, “Klasterizatsia tekstovykh dokumentov na osnove sostavnykh kliuchevykh termov,” (in Russian), *Vestnik Novosibirskogo gosudarstvennogo universiteta*. Seryya: Informatsionnye tekhnologii, 2010.