DOI: https://doi.org/10.36910/6775-2524-0560-2024-56-24
УДК: 004.932
Коваль Ігор Михайлович, аспірант
https://orcid.org/0009–0001–2083–1747
Суринович Олена Миколаївна, к.т.н., доцент
https://orcid.org/0000–0002–9300–0039
Луцький національний технічний університет, м. Луцьк, Україна

ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ МЕТОДІВ ПОПЕРЕДНЬОЇ ОБРОБКИ ТЕКСТУ В ORANGE DATA MINING ТА KNIME

Коваль І.М., Суринович О.М. Порівняльне дослідження методів попередньої обробки тексту в Orange Data Mining та KNIME. У роботі досліджується порівняння результатів попередньої обробки текстових даних у програмних системах Knime Analytics та Orange Data Mining. Представлено детальний опис методології дослідження, включаючи інструменти попередньої обробки текстових даних, налаштування та створення моделей відповідно до можливостей кожної з програм. Отримані результати аналізуються з використанням інструментів візуалізації, при цьому використовуються різні формати представлення результату. Переваги та недоліки кожного інструменту виявляються у рекомендаціях щодо застосування тієї чи іншої програмної системи у різних умовах. Результати попередньої обробки текстових даних в обох програмних системах показали, що дані були очищені від шуму, небажаних слів та синтаксичних елементів, що дозволило виділити ключові теми і тенденції із тестового матеріалу.
Ключові слова: Text Preprocess, Clustering, Orange Data Mining, Data mining, Knime Analytics, Word Cloud, попередня обробка тексту, Text to Vector

Koval I., Surynovych O. A comparative study of text preprocessing methods in Orange Data Mining and KNIME. The paper examines the comparison of the results of preprocessing of text data in the Knime Analytics and Orange Data Mining software systems. A detailed description of the research methodology is presented, including tools for preprocessing textual data, setting up and creating models according to the capabilities of each of the programs. The obtained results are analyzed using visualization tools, while different formats of the result presentation are used. The advantages and disadvantages of each tool are revealed in the recommendations for the use of this or that software system in different conditions. The results of pre-processing of text data in both software systems showed that the data were cleaned of noise, unwanted words and syntactic elements, which made it possible to highlight key themes and trends from the test material.
Keywords: Text Preprocess, Clustering, Orange Data Mining, Data mining, Knime Analytics, Word Cloud, text pre–processing, Text to Vector

Постановка проблеми. У сучасну епоху цифрових технологій аналітика даних стала одним із основних напрямків інформаційних технологій. Текстові дані становлять чи не найбільший відсоток усіх даних у світі, і кожного дня користувачі стикаються з аномально великою кількістю таких даних. Завдання кластеризації текстових даних полягає в тому, щоб впорядкувати хаотичний та непідготовлений матеріал таким чином, аби можна було отримати з нього корисну інформацію. Кластеризація та класифікація дозволяють виявити приховані патерни та структури у великих обсягах даних. Проте ефективність кластеризації значною мірою залежить від якості попередньої обробки тексту, яка може включати в себе токенізацію (Tokenization), видалення стоп-слів (Stop-Word Filter), стемінг (Stemming), лематизацію (Lemmatization) та інші методи. Попередня обробка тексту (Text Preprocessing) є обов'язковою складовою для підвищення точності кластеризації, зокрема при використанні таких програмних систем, як Knime та Orange Data Mining.

Сучасні інформаційні системи потребують автоматизованої обробки, особливо коли йдеться про великі обсяги текстових даних. Інструменти Orange та Knime надають можливість автоматизації Text Preprocessing, що, в свою чергу, зменшує час та зусилля, необхідні для роботи з текстовими даними, та підвищує продуктивність роботи. У кожному конкретному датасеті вибір оптимальних методів Text Preprocessing є критично важливим для отримання точних кластерів після застосування алгоритму. Недоліки або неправильна обробка тексту гарантовано призведуть до спотворених кластерів, які не дадуть жодної корисної інформації. Дослідження, присвячені аналізу ефективності різних методів та моделей попередньої обробки текстових даних, мають високу практичну цінність для науковців та аналітиків. Отримані результати з цієї статті можуть бути використані для розробки рекомендацій та побудови моделей, що дозволять успішно провести Text Preprocessing у запропонованих програмних системах.

Формулювання мети дослідження. Метою даного дослідження є аналіз ефективності різних методів попередньої обробки текстових даних (Text Preprocessing) та формування висновків щодо доцільності використання тієї чи іншої програмної системи у різних ситуаціях.

Аналіз останніх досліджень і публікацій. У сучасній науковій літературі активно досліджується питання попередньої обробки текстових даних як обов'язкового етапу перед кластеризацією. Багато дослідників підкреслюють важливість якісного Text Preprocessing для підвищення точності кластеризації та зниження рівня шуму в текстових даних. Зокрема у дослідженнях [1] акцентується увага на використанні алгоритмів лематизації для підвищення релевантності кластерів. У науковій праці [2] авторами відводиться велике дослідження впливу тих чи інших маніпуляцій з текстом на вихідні дані.

Orange Data Mining – це потужний інструмент для аналізу даних і машинного навчання, розроблений у вигляді відкритого програмного забезпечення [3]. Він надає користувачам можливість створювати візуальні робочі процеси для аналізу даних за допомогою графічного інтерфейсу користувача. Інструмент підтримує різні методи аналізу даних, включаючи кластеризацію, класифікацію, регресію та візуалізацію. Orange дозволяє інтегрувати різні методи попередньої обробки тексту, такі як токенізація, видалення стоп-слів, стемінг та лематизація, що робить його хорошим для текстового майнінгу та аналізу. Традиційно вважається, що Orange краще підходить для початківців, через зручність та інтуїтивність інтерфейсу.

KNIME (Konstanz Information Miner) – це інтегрована платформа для аналізу даних, звітності та інтеграції, яка підтримує весь процес аналізу даних через візуальне програмування [4]. KNIME дозволяє користувачам створювати робочі процеси для підготовки даних, аналізу та візуалізації без необхідності писати код. Платформа підтримує широкий спектр методів машинного навчання та статистичного аналізу, а також інтеграцію з іншими інструментами та мовами, такими як Python та R. Традиційно вважається, що Knime більше підходить для досвідчених користувачів.

Попередня обробка тексту Orange Data Mining представлена єдиним віджетом [5] який включає у себе усі можливі процедури для підвищення вихідного результату.

Схема зв'язку між віджетом Corpus (Відповідає за завантаження вхідних тексових даних) та віджетом Preprocess Text (відповідає за попередню обробку тексту) (рис. 1).

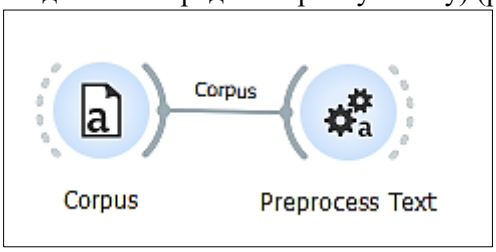

Рис 1. Віджети Orange

Функціонал (рис.2), який надає програмна система Orange Data Mining для попередньої обробки текстових даних. Представлені секції:

– Transformation (трансформація):
- Lowercase – перетворює всі символи тексту в нижній регістр;
- Remove accents – видаляє діакритичні знаки з тексту;
- Parse html – видаляє HTML–теги з тексту;
- Remove urls – видаляє URL–адреси з тексту;

– Tokenization (токенізація):
- Word Punctuation – токенізує текст, враховуючи пунктуацію;
- Whitespace – токенізує текст за пробілами;
- Sentence – токенізує текст за реченнями;
- Regexp – токенізує текст за допомогою регулярних виразів (наприклад, \w+ для слів);
- Tweet – токенізує текст, оптимізуючи для аналізу твітів;

– Filtering (фільтрація):
- Stopwords – видаляє стоп–слова. Можна вибрати мову і список стоп–слів;
- Lexicon – фільтрує текст за допомогою певного лексикону;
- Numbers – включає або виключає числа з тексту;
- Regexp – використовує регулярні вирази для фільтрації тексту;
- Document frequency – фільтрує токени за їх частотою в документі;
- Most frequent tokens – вибирає певну кількість найчастіших точені;
- POS tags – вибирає токени за їх частинами мови;

– POS Tagger (позначення частин мови):

- Averaged Perceptron Tagger – використовує алгоритм середнього перцептрону для позначення частин мови;
- Treebank POS Tagger (MaxEnt) – використовує Treebank POS Tagger з MaxEnt.

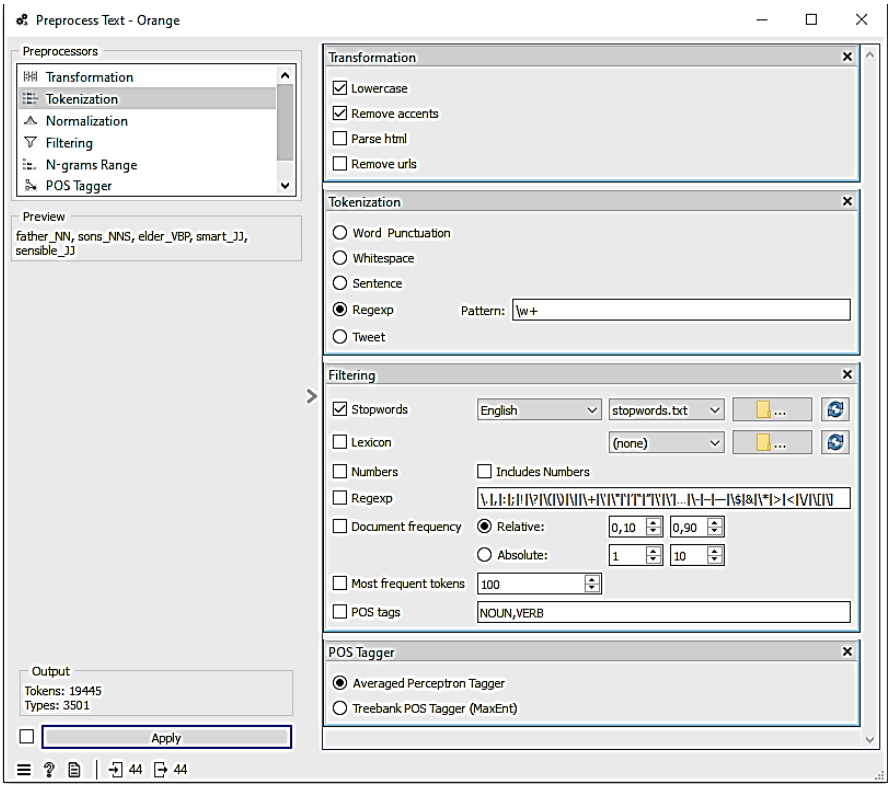

Рис 2. Інтерфейс віджету Preprocess Text

Попередня обробка тексту Knime Analytics представлена нодами [6] кожна з яких виконує відповідну функцію.

За допомогою нод, представлених на рис. 3, можна виконати основні текстові перетворення для очищення та підготовки до подальшої кластеризації. В залежності від бажаних перетворень програма пропонує відповідний функціонал:

– Case Converter – перетворює текст у верхній або нижній регістр;
– Diacritic Remover – видаляє діакритичні знаки з тексту;
– Dictionary Filter – фільтрує текст за допомогою словника;
– Dictionary Replacer – замінює слова в тексті на основі словника;
– Dictionary Replacer (File–based) – замінює слова в тексті, використовуючи словник із файлу;
– Hyphenator – розбиває слова на склади за допомогою дефісів;
– Kuhlen Stemmer – виконує стемінг (зведення слів до кореневої форми);
– Modifiable Term Filter – фільтрує змінні терміни;
– N Chars Filter – видаляє слова, довжина яких менше певної кількості символів;
– Number Filter – видаляє числа з тексту;
– Porter Stemmer – виконує стемінг за допомогою алгоритму Портера;
– Punctuation Erasure – видаляє розділові знаки з тексту;
– RegEx Filter – фільтрує текст за допомогою регулярних виразів;
– Replacer – замінює текст на основі заданих правил або регулярних виразів;
– Snowball Stemmer – виконує стемінг, використовуючи алгоритм Snowball;
– Stanford Lemmatizer – виконує лематизацію за допомогою Stanford NLP;
– Stop Word Filter – видаляє стоп–слова з тексту;
– Tag Filter – фільтрує текст за певними тегами;
– Tag Stripper – видаляє теги з тексту).

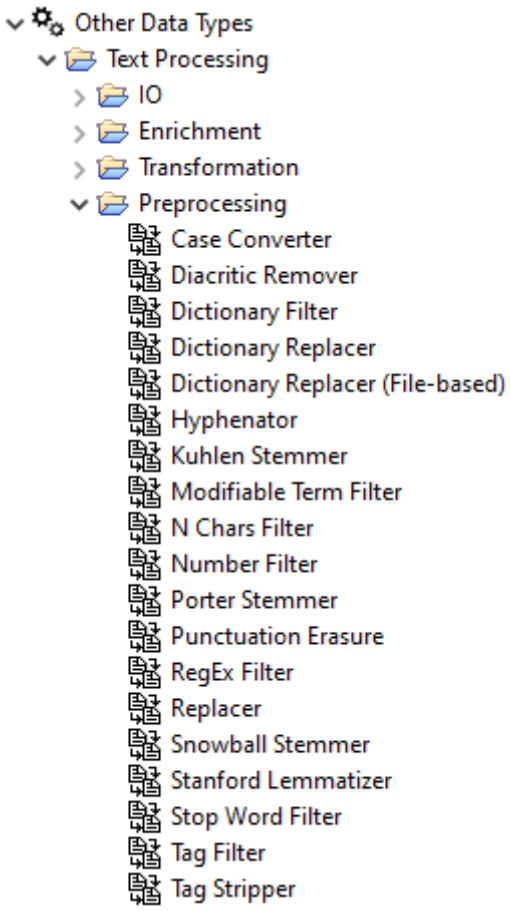

Рис. 3 Ноди гілки Preprocessing

Виклад основного матеаріалу. У ході експеременту використано 2500 заголовків новинних статтей. Для чистоти експерименту використано лише взаємовідповідний функціонал обох програм.

Опишемо створення та тестування моделі попреденьої обробки текстових даних у Orange Data Mining. Модель попередньої обробки текстових даних з використанням віджетів Text Mining відображено на рис.4.

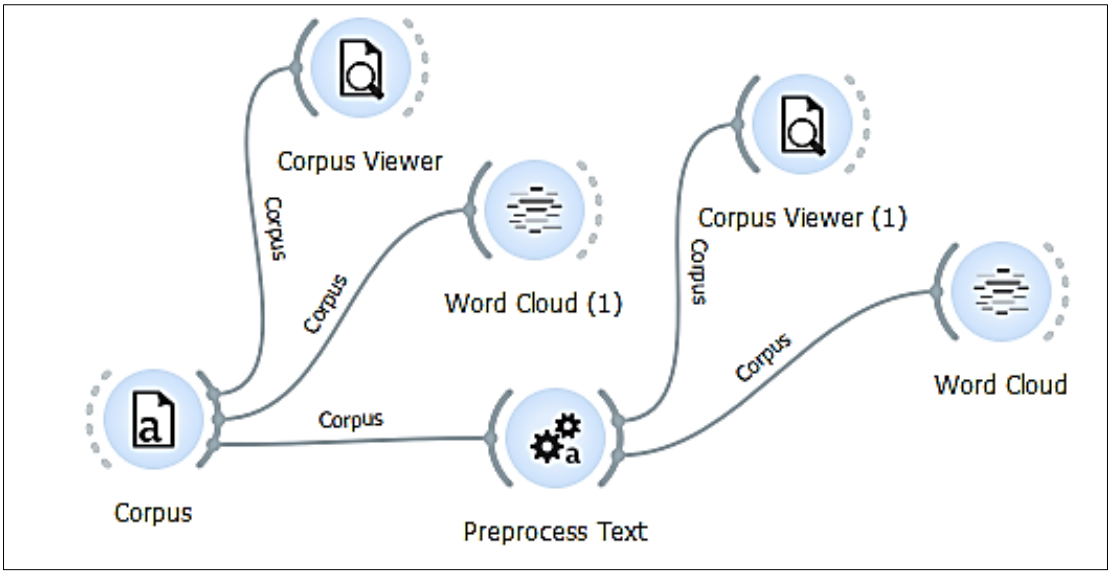

Рис. 4. Готова модель Orange Data Mining

В ході експеременту було виявлено певну проблему (рис. 5) у роботі віджету Corpus Viewer, а саме віджет Corpus Viewer (1) не відображає актуальні зміни які були застосовані інструментом Preprocess Text.

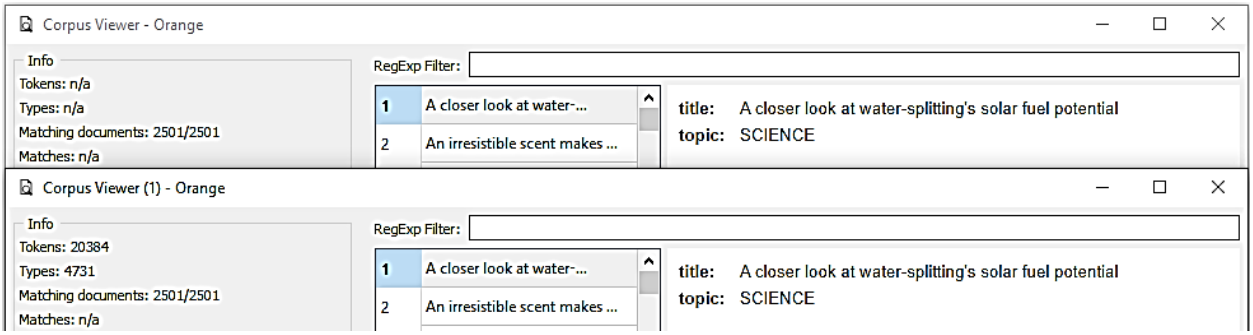

Рис. 5 Віджети Corpus Viewer

Для демонстрації результатів роботи Preprocess Text було прийняте рішення використати віджет Word Cloud, що являє собою хмару слів. Результати Word Cloud до застосування віджету Preprocess Text відображено на рис.6.

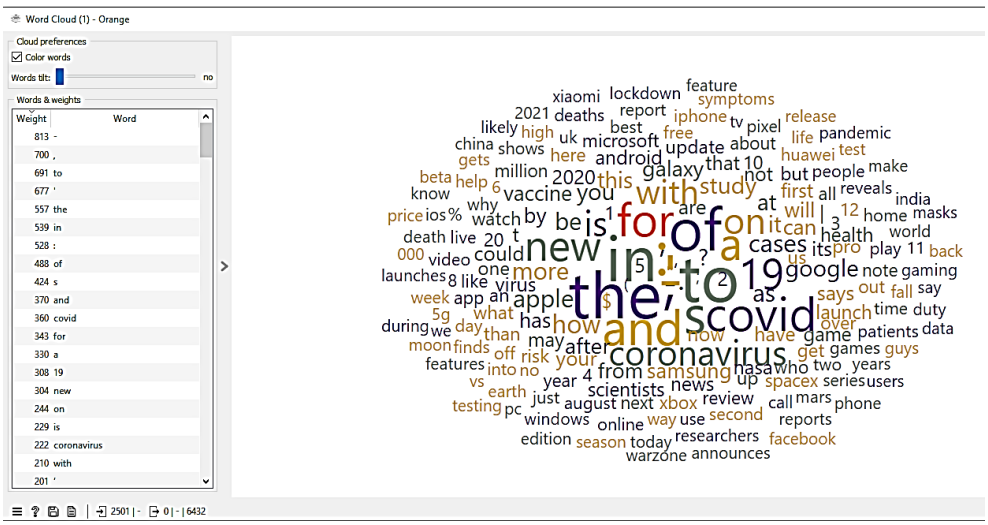

Рис. 6. Візуалізація Word Cloud

Результати Word Cloud після застосування віджету Preprocess Text відображено на рис.7.

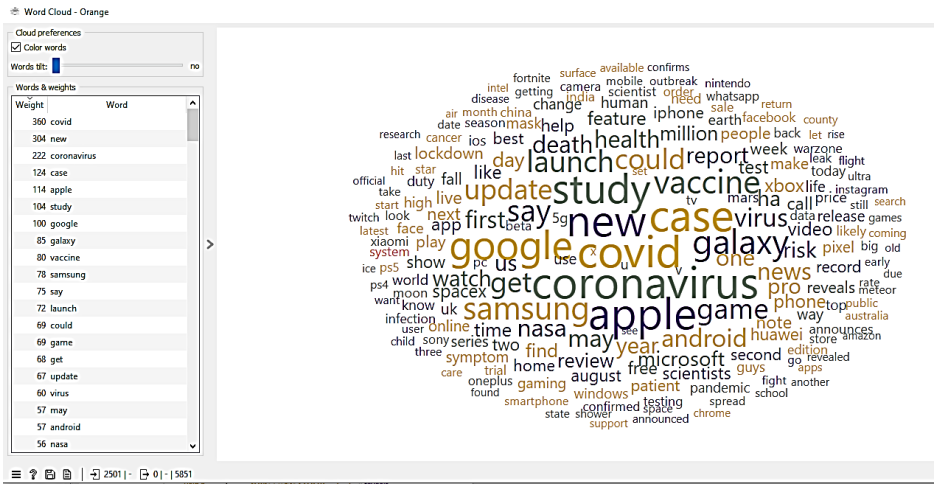

Рис. 7. Візуалізація Word Cloud після застосування віджету

Слід зазначити, що Orange Data Mining не вистачає віджетів які можна було би використовувати для перегляду проміжного результату після застосування Preprocess Text, адже Corpus Viewer не відображає зміни, а Word Cloud дає лише поверхневе уявлення та не показує результат роботи наприклад Pos Tagger. Для того, щоб побачити результат роботи Pos Tagger –

доводиться додатково представляти текстові дані у вигляді векторів. Принагідно варто додати, що на кінцевий результат кластеризації це не впливає.

Застосувавши інструменти Tokenization (Regexp), Normalization (WordNet Lematizer), Transformation (Lowercase, Remove accents), Filtering (Stop-word, Numbers), Pos Tagger – вдалось отримати якісний, відфільтрований результат. З даними (рис. 6) можна здійснити успішну кластеризацію текстових даних.

Опишемо створення та тестування моделі попередньої обробки текстових даних у Knime Analytics. Кінцева модель попередньої обробки текстових даних з використанням nodes (нод) у Knime (рис.8).

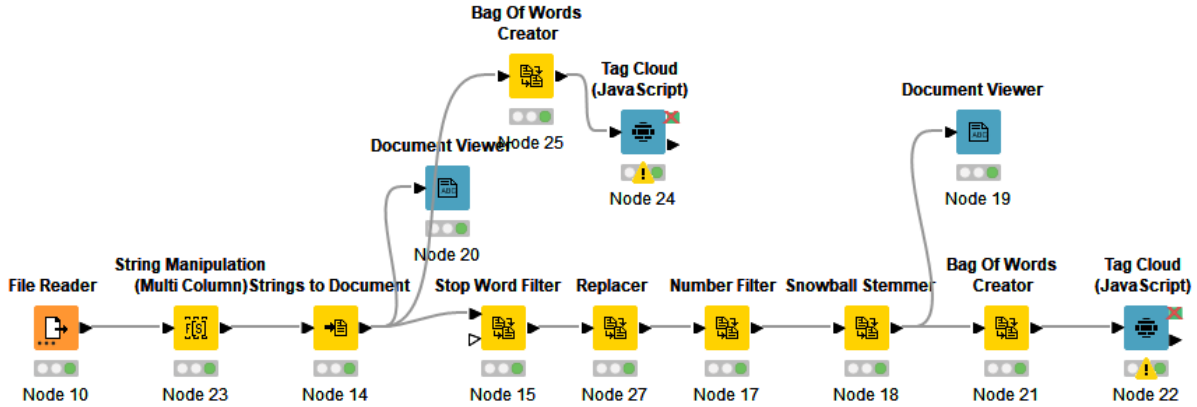

Рис. 8. Готова модель Knime Analytics

Елементарні текстові перетворення наприклад зведення до нижнього регістру, або використання регулярних виразів для перетворення даних формату string. Не зрозумілою залишається специфіка необхідності застосування нод Stop-Word, Number Filter, Snowball Stemmer оскільки для успішної реалізації усіх процесів очищення тексту вимагається перетворити вхідні текстові дані у документ застосувавши ноду Strings to Document.

На рис. 9 зображена ефективність перетворення тексту за допомогою нод з гілки Text Preprocess. За допомогою ноди Document Viewer аналізується вихідний формат строки перед кластеризацією. Як видно з рис. 9 усі інструменти застосовані успішно.

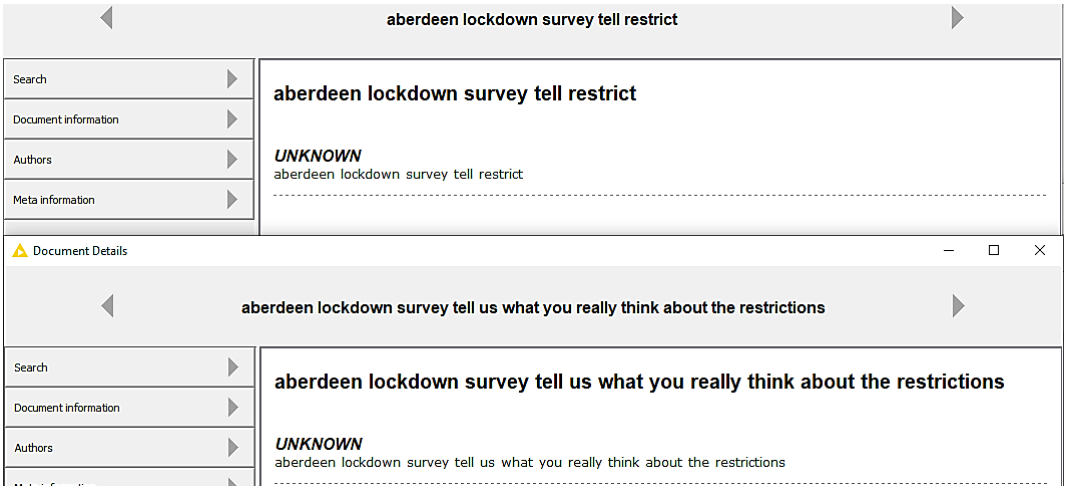

Рис. 9. Ноди Document Viewer

Для візуалізації тексту який представлений у вигляді векторів та готовий до подальшої кластеризації застосовано ноду JavaScript Tag Cloud View, що дозволяє оцінити якість початкового та кінцевого результату обробки тексту.

Хмара слів до застосування нод попередньої обробки тексту (рис. 10).

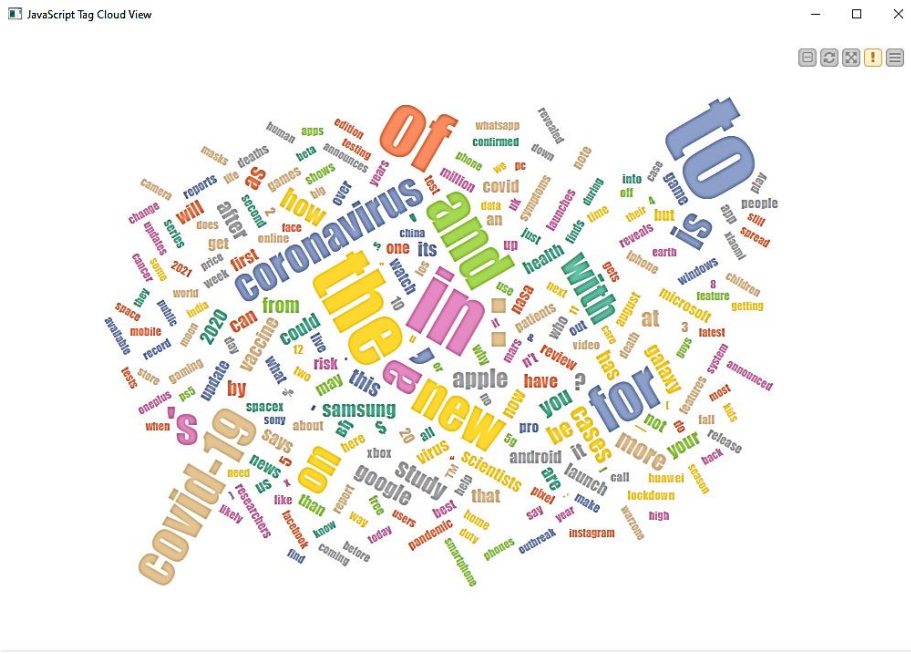

Рис. 10. Нода JavaScript Tag Cloud View

Єдиною маніпуляцією з вхідним текстом на даному етапі є застосування ноди String Manipulation (Multicolumn) всередині якої задано регулярний вираз що зводить всі складові тексту до нижнього регістру.

Хмара слів яку було отримано в результаті комплексної попередньої обробки тексту (рис. 11).

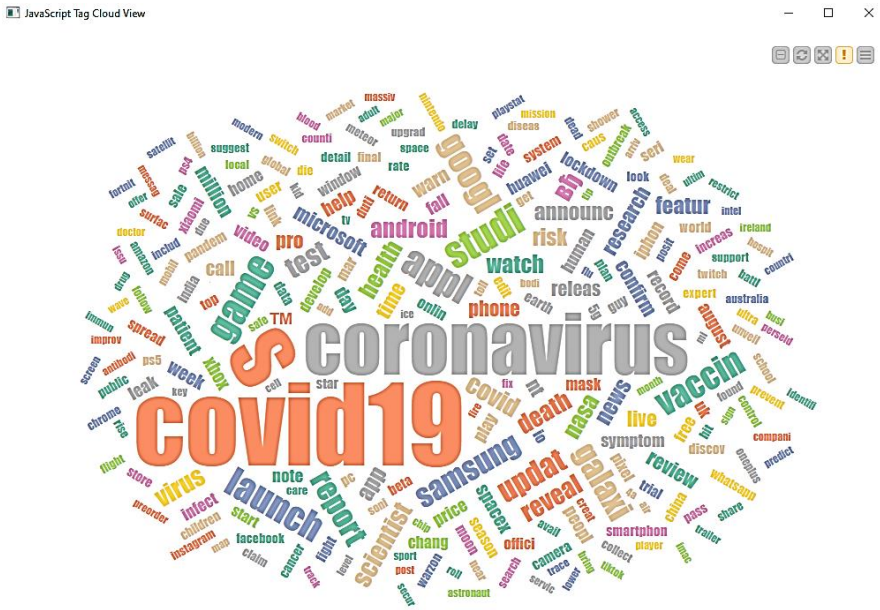

Рис 11. Нода JavaScript Tag Cloud View

Виходячи з цього, можна зробити висновок, що дані готові для подальшої кластеризації, оскільки основні теми вже виділені та очищені від шуму. Ця хмара слів наочно демонструє, що проведена попередня обробка тексту була успішною і дозволила виявити ключові тенденції та теми, присутні у вихідних даних.

У ході цього дослідження було проаналізовано та порівняно процеси попередньої обробки текстових даних, що є критичним етапом для досягнення якісних результатів кластеризації.

Висновки та перспективи подальшого дослідження. В Orange Data Mining процес попередньої обробки тексту виявився більш інтуїтивним та простим для реалізації завдяки зручному інтерфейсу та інтегрованим віджетам. Однак, обмежені можливості віджета Corpus Viewer у

відображенні проміжних результатів змусили використати альтернативні методи для перевірки якості очищення даних, такі як хмара слів. Це є недоліком Orange, оскільки відсутність повного перегляду проміжних результатів ускладнює аналіз окремих етапів обробки тексту.

У KNIME попередня обробка тексту потребує додаткових кроків, таких як обов'язкове перетворення текстових даних у формат документів, що додає складності у процесі. Незважаючи на це, KNIME надає більше можливостей для контролю над кожним етапом обробки тексту і дозволяє отримати більш детальний аналіз результатів. Проте, цей підхід потребує більше часу і навичок, що може бути недоліком для початківців.

Результати попередньої обробки текстових даних в обох програмних системах показали, що після застосування методів попередньої обробки тексту, дані були очищені від шуму, небажаних слів та синтаксичних елементів, що дозволило виділити ключові теми і тенденції із тестованого матеріалу.

Відповідно, можна зробити висновок, що обидві програмні системи мають свої переваги та недоліки, і вибір інструменту залежить від досвіду користувача та конкретних завдань. Orange Data Mining більше підходить для швидкої та інтуїтивної обробки даних, тоді як KNIME забезпечує більшу гнучкість та контроль на всіх етапах, але вимагає більше часу та знань. Подальші дослідження можуть бути спрямовані на застосування алгоритмів кластеризації (наприклад, K-Means, DBSCAN, Hierarchical Clustering) та формування висновків щодо ефективності тих чи інших методів попередньої обробки текстових даних у контексті точності та чистоти отриманих кластерів.

Список бібліографічного опису

1. Manning C. D., Raghavan P., & Schütze H. Introduction to Information Retrieval, 2022. URL: https://nlp.stanford.edu/IR–book/pdf/irbookonlinereading.pdf (дата звернення: 05.08.2024).
2. Charu C. Aggarwal, ChengXiang Zhai. Mining Text Data, 2012. URL: https://doi.org/10.1007/978–1–4614–3223–4 (дата звернення: 05.08.2024).
3. Orange Data Mining. URL: https://orangedatamining.com/ (дата звернення: 05.08.2024).
4. KNIME Analytics Platform.URL: https://www.knime.com/knime–analytics–platform (дата звернення: 17.08.2024).
5. Text Preprocessing Orange Blog. URL:https://orangedatamining.com/blog/text–preprocessing/ (дата звернення: 05.08.2024).
6. From data collection to text mining. URL: https://www.knime.com/blog/data–collection–to–text–mining (дата звернення: 17.08.2024).

References

1. Manning C. D., Raghavan P., & Schütze H. Introduction to Information Retrieval, 2022. URL: https://nlp.stanford.edu/IR–book/pdf/irbookonlinereading.pdf (access date: 05.08.2024).
2. Charu C. Aggarwal, ChengXiang Zhai. Mining Text Data, 2012. URL: https://doi.org/10.1007/978–1–4614–3223–4 (access date: 05.08.2024).
3. Orange Data Mining. URL: https://orangedatamining.com/ (access date: 05.08.2024).
4. KNIME Analytics Platform.URL: https://www.knime.com/knime–analytics–platform (access date: 17.08.2024).
5. Text Preprocessing Orange Blog. URL:https://orangedatamining.com/blog/text–preprocessing/ (access date: 05.08.2024).
6. From data collection to text mining. URL: https://www.knime.com/blog/data–collection–to–text–mining (access date: 17.08.2024).