

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Інститут (факультет) Автоматизації і комп'ютерних систем

«До захисту в ЕК»

«До захисту допущено»

«13» лютого 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

зі спеціальності 122 «Комп'ютерні науки»

(код та назва спеціальності)

освітньо-професійної програми Інформаційні управляючі системи та технології

на тему: Дослідження та аналіз інформації з соціальної мережі Twitter на основі технології Text Mining

Виконав: здобувач 2 курсу, групи ІС-2-3М

Кривець Олександр Юрійович

Харкянєн Олена Валеріївна

Консультанти

(ім'я та прізвище)

(ім'я та прізвище)

(ім'я та прізвище)

Рецензент

(ім'я та прізвище)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) незарплатованої допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач

(підпис)

Київ - 2023р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

Освітній ступінь магістр

Спеціальність 122 «Комп'ютерні науки»

(код і назва)

Освітньо-професійна програма Інформаційні управляючі системи та технології

(назва)

ЗАТВЕРДЖУЮ

Завідувач

кафедри Інформаційних технологій,
штучного інтелекту і кібербезпеки

Грибков С.В.

“ 11 ” листопада 2022 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Кривець Олександр Юрійович

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та аналіз інформації з соціальної мережі Twitter на основі технології Text Mining

керівник роботи Харкянен Олена Валеріївна, доцент, к.т.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 11 листопада 2022 року №820-кс

2. Строк подання здобувачем роботи 17.01.2023

3. Вихідні дані до роботи 1. Статті зарубіжних авторів про технологію Text Mining.

2. Відомості про сучасні програмні продукти для роботи з Text Mining. 3. Доступ до особистого кабінету розробника соціальної мережі Twitter.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

1. Дослідження технології Text Mining та можливості її використання для статистичного аналізу та прогнозування. 2. Дослідження варіантів отримання даних з соціальної мережі Twitter та варіантів взаємодії з нею. 3. Обґрунтування можливості та доцільності використання технології Text Mining в задачах статистичного аналізу та прогнозування. 4. Виявленні проблеми використання програмних продуктів для роботи з Text Mining та знайдені шляхи їх вирішення.

5. Перелік графічного матеріалу Фрагменти коду.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Харкянен О.В., к.т.н., доц.	14.11.2022	25.11.2022
2	Харкянен О.В., к.т.н., доц.	28.11.2022	09.11.2022
3	Харкянен О.В., к.т.н., доц.	12.12.2022	06.01.2023

7. Дата видачі завдання 11 листопада 2022 року

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження технології Text Mining та можливості її використання для статистичного аналізу та прогнозування	14.11.2022 - 25.11.2022	Виконано
2	Дослідження варіантів отримання даних з соціальної мережі Twitter та варіантів взаємодії з нею	28.11.2022 - 09.12.2022	Виконано
3	Обґрунтування можливості та доцільності використання технології Text Mining в задачах статистичного аналізу та прогнозування.	12.12.2022 - 23.12.2022	Виконано
4	Виялення проблем використання програмних продуктів для роботи з Text Mining та пошук шляхів їх вирішення	23.12.2022 – 28.1.2023	Виконано
5	Оформлення пояснювальної записки та створення презентації	09.01.2023 – 06.01.2023	Виконано

Здобувач

(підпис)

Керівник роботи

(підпис)

Кривець О.Ю.

(прізвище та ініціали)

Харкянен О.В.

(прізвище та ініціали)

АНОТАЦІЯ

Дипломна робота «Дослідження та аналіз інформації з соціальної мережі Twitter на основі технології Text Mining», складається з 65 сторінок, 3 розділів, 27 малюнків, 2 таблиць, та 12 літературних джерел.

В магістерській роботі було досліджено та проаналізовано методи Text Mining та способи їх використання. Досліджено соціальну мережу Твітер як джерело даних та можливості взаємодії з ним за допомогою Twitter API.

Досліджено функціональні можливості програмних продуктів для роботи з методами Text Mining.

Проаналізовано можливості використання методів Text Mining для статистичного аналізу даних отриманих з соціальної мережі Twitter.

Реалізовано декілька прикладів з використанням методів технології Text Mining для вирішення актуальних задач.

КЛЮЧОВІ СЛОВА: TEXT MINING, TWITTER, АНАЛІЗ ТЕКСТУ, JAVASCRIPT.

ANNOTATION

The thesis "Research and analysis of information from the Twitter social network based on Text Mining technology" consists of 65 pages, 3 chapters, 27 images, 2 tables, and 12 literary sources.

Text Mining methods and methods of their use were researched and analyzed in the master's thesis. The social network Twitter as a source of data and the possibilities of interacting with it using the Twitter API were studied.

The functional capabilities of software products for working with Text Mining methods have been studied.

The possibility of using Text Mining methods for statistical analysis of data obtained from the Twitter social network was analyzed.

Several examples using the methods of Text Mining technology to solve actual problems have been implemented.

KEYWORDS: TEXT MINING, TWITTER, TEXT ANALYSIS, JAVASCRIPT

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ TEXT MINING	12
1.1. Загальна інформація про технологію Text Mining.....	12
1.2. Загальна характеристика методів текстового аналізу.	13
1.2.1. Класифікація.	14
1.2.2. Класифікація за допомогою дерева рішень.	15
1.2.3. Метод Байєсової (наївної) класифікації (Naive Bayes).....	16
1.2.4. Метод опорних векторів	17
1.2.5. Метод k-найближчого сусіда.	18
1.2.6. Штучні нейронні мережі	19
1.2.7. Кластеризація.....	20
1.2.8.Розпізнавання іменованих сутностей	21
1.2.9. Усунення неоднозначності	21
1.2.10. Аналіз тональності тексту.....	22
1.3.Прогнозування.....	25
1.4. Дослідження завдань Text Mining.	25
1.5. Галузі використання Text Mining.	26
1.6. Дослідження програмних рішень для Text Mining.	27
1.6.1. Intelligent Miner.....	28
1.6.2 TextAnalyst.....	29
1.6.3. WebAnalyst.....	30
1.6.4. Text Miner.....	30
1.6.5. SemioMap.	31

1.6.6. Oracle Text.....	32
1.6.7. Orange.....	34
1.6.8. Weka.	34
1.7. Порівняльна характеристика програмних продуктів.	35
РОЗДІЛ 2. ВИЯВЛЕННЯ ПРОБЛЕМ ВИКОРИСТАННЯ TEXT MINING ТА ДОСЛІДЖЕННЯ ШЛЯХІВ ЇХ ВИРІШЕННЯ	37
2.1. Дослідження етапів процесу Text Mining в Orange.	37
2.2. Пошук інформації та виявлення вихідних даних.	37
2.3. Отримання доступу до Twitter API.....	38
2.4. Підключення та робота з Twitter API в Orange	40
2.5. Проблеми використання Twitter API.....	43
2.6. Попередня обробка документів.	43
2.7. Аналіз проблем попередньої обробки документів	45
2.8. Використання методів аналізу тексту та інтерпретація результатів.	46
2.9. Проблеми на етапі інтерпретації результатів	47
2.10. Розв'язок проблем використання програмних засобів для текстового аналізу	48
2.10.1 Труднощі отримання доступу до Twitter API.	48
2.10.2 Відсутність інтеграції з Twitter API.....	49
2.10.3. Обмежені можливості попередньої обробки текстів.....	49
2.10.4. Автоматична інтерпретація результатів текстового аналізу та реагування	50
2.11 Дослідження способів використання JavaScript для Text Mining.....	50
2.12. Висновок до розділу 2.	51
РОЗДІЛ 3. ВИКОРИСТАННЯ JAVASCRIPT ДЛЯ TEXT MINING	52

3.1. Постановка задачі.....	52
3.2. Розробка Телеграм бота	52
3.3. Висновки до розділу 3.....	57
ВИСНОВКИ	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТКИ	60
ДОДАТОК А «ФРАГМЕНТИ КОДУ ПРОГРАМ»	60

ВСТУП

Актуальність теми.

Аналіз тексту визначається як процес виявлення прихованого, корисного і цікавого зразка з неструктурованих текстових документів. Аналіз тексту також відомий як процес пошуку знань у текстовому інтелектуальному аналізі даних. Приблизно 80 % корпоративних даних знаходиться в неструктурованому форматі. Інформаційний пошук у неструктурованому тексті дуже складний, оскільки він містить велику кількість інформації, що вимагає використання специфічних методів і алгоритмів обробки для отримання корисних знань. Оскільки найбільш поширеною формою для зберігання інформації є текст, інтелектуальний аналіз тексту видається більш важливим процесом, ніж інтелектуальний аналіз даних (data mining). Інтелектуальний аналіз тексту є міждисциплінарною областю, яка включає збір даних, обробку web-даних, інформаційний пошук, комп'ютерну лінгвістику і обробку природної мови.

Зв'язок роботи з науковими програмами, планами, темами. Наукова робота виконувалася згідно з планами науково-дослідних робіт кафедри інформаційних систем Національного університету харчових технологій (НУХТ).

Об'єктом дослідження є текстові дані отримані з соціальної мережі Twitter.

Предметом дослідження є методи інтелектуального аналізу тексту в задачах статистичного аналізу та прогнозування.

Мета й завдання дослідження. Метою магістерської роботи є дослідження та застосування методів Text Mining в задачах статистичного аналізу на основі даних отриманих з соціальної мережі Twitter. Виявлення проблем використання програмних продуктів для Text Mining та дослідження шляхів їх вирішення.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- дослідити технологію Text Mining та можливості її використання для статистичного аналізу та прогнозування;

- дослідити варіанти отримання даних з соціальної мережі Twitter та варіанти взаємодії з нею;
- обґрунтувати можливість та доцільність використання технології Text Mining в задачах статистичного аналізу та прогнозування;
- дослідити існуючі програмні продукти для роботи з методами Text Mining;
- виявити проблем використання програмних продуктів для роботи з Text Mining та знайти шляхи їх вирішення;

Методи дослідження. Дані для аналізу отримані із соціальної мережі Twitter, з використанням Twitter API. Проектування структури даних та текстовий аналіз було здійснено методами такого програмного продукту як Orange і методами мови програмування JavaScript.

Наукова новизна одержаних результатів. Виявленні проблеми використання програмних продуктів для Text Mining та знайдені шляхи їх вирішення. Запропоновані способи використання мови програмування JavaScript для розв'язання задач Text Mining.

Практичне значення отриманих результатів. Практичне значення роботи полягає у адаптації методів інтелектуального аналізу тексту для статистичного аналізу та прогнозування на основі даних отриманих з соціальної мережі Twitter, використовуючи мову програмування JavaScript.

Особистий внесок здобувача.

- Досліджено технологію Text Mining, її завдання, етапи, методи, програмні рішення, типи задач які можуть бути вирішені.
- Досліджено соціальну мережу Twitter, її можливості, в тому числі взаємодія з Twitter API.
- Налаштовано отримання даних із соціальної мережі Twitter за допомогою мови програмування JavaScript та програмного продукту Orange;
- Виявлені проблеми використання Twitter API та програмних продуктів для роботи з Text Mining, та знайдені шляхи їх вирішення.
- Обґрунтовано доцільність використання мов програмування для вирішення

задач Text Mining.

- Реалізовано приклад використання методів технології Text Mining та взаємодії з Twitter API засобами Javascript.

Апробація результатів роботи. Матеріали до даної роботи опубліковані у збірниках тез:

1. IX Міжнародної науково-технічної Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 25 листопада 2022 [Електронний ресурс]. – К: НУХТ, 2022. – 110 с.

2. LXXIX наук. конф. професорсько-викладацького складу, аспірантів, студ. та працівників відокремлених структурних підрозділів університету. – К.: НТУ, 2023. – (у друці)

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ TEXT MINING

1.1. Загальна інформація про технологію Text Mining.

Вперше «ручні» техніки Text Mining з'явилися в середині 1980-х, а наступне десятиліття розвиток технологій дозволив значно їх удосконалити. У міждисциплінарному сенсі Text Mining лежить на стику пошуку інформації, Data Mining, машинного самонавчання, статистики та комп'ютерної лінгвістики.

Необхідність використання величезних обсягів корпоративної інформації, яка існує в неструктурованій формі, відома вже давно – ще з 80-х років минулого століття. Але спеціальні технології, які дозволяють працювати саме з текстами, а не з кількісними даними, з'явилися лише наприкінці 90-х років.

Технологія Text Mining є одним з різновидів методів Data Mining і має на увазі процеси отримання знань і високоякісної інформації з текстових масивів. Це зазвичай відбувається за допомогою виявлення шаблонів та тенденцій за допомогою засобів статистичного вивчення шаблонів.

Така технологія глибинного аналізу текстів здатна «просіювати» великі обсяги неструктурованої інформації та виявляти з них лише найвагоміше, щоб людині не доводилося витрачати час на видобуток цінних знань «вручну».

Text Mining – це алгоритмічне виявлення та виявлення в «сирих» даних досі невідомих кореляцій та зв'язків, раніше невідомих практично корисних знань, які можна інтерпретувати та використовувати для прийняття рішень у різних сферах діяльності людини. Результати Text Mining можуть бути використані для математичного прогнозування, аналізу соціальної ситуації та аналізу ринків.

Подібні технології незамінні для здобуття знань і відіграють важливу роль у всій системі керування знаннями.

Як правило, Text Mining передбачає процес структурування текстових даних, вилучення шаблонів із вже структурованих даних, і фінальну оцінку та інтерпретацію отриманих результатів.

Коли ми говоримо про «високоякісну» інформацію, мається на увазі, що вона має бути осмисленою, релевантною та цікавою дослідникам. Типові завдання Text

Mining включають категоризацію, кластеризацію текстів, вилучення концептів та об'єктів, створення таксономій, смисловий аналіз, узагальнення документації та моделювання об'єктів, тобто встановлення зв'язків між різними відомими об'єктами.

Аналіз текстів включає вилучення інформації та лінгвістичний аналіз для виявлення частоти входження різних слів, виявлення шаблонів, розставлення тегів та анотування, техніки Data Mining, включаючи аналіз зв'язків та асоціацій, візуалізацію та прогностичний аналіз. Зрештою, загальна мета всього цього полягає в тому, щоб перетворити текст на дані, доступні для аналізу.

Прикладом найчастішого використання Text Mining може послужити сканування набору документів, щоб скласти модель для прогностичної класифікації, або спростити за допомогою виявленої інформації пошук по базах даних.

По суті, аналіз текстів та Text Mining – це набір лінгвістичних, статистичних технік, а також технік машинного самонавчання, які здатні моделювати та структурувати інформаційний контент та текстові джерела з метою бізнес-аналітики, аналізу даних, досліджень. Ці технології, окремо чи разом із іншими засобами, застосовують у корпоративній практиці управління знаннями на вирішення тих чи інших бізнес-проблем. Відомо, що близько вісімдесяти відсотків важливої для бізнесу інформації існує у неструктурованій текстовій формі. Технології, що обговорюються нами, дозволяють отримати з цих даних цінні знання – факти, бізнес-правила, взаємозв'язки – які неможливо отримати будь-якими іншими автоматизованими засобами.

1.2. Загальна характеристика методів текстового аналізу.

- Класифікація – за деякою контрольною вибіркою система розпізнає та відносить новий об'єкт до тієї чи іншої категорії.
- Кластеризація - це процес виділення компактних підгруп об'єктів з близькими властивостями.
- Розпізнавання іменованих сутностей — це використання географічних

довідників або статистичних методів для визначення властивостей названих у тексті: людей, організацій, топонімів, біржових символів, певних скорочень, тощо.

- Усунення неоднозначності — використання контекстних підказок — може знадобитися, щоб вирішити, де, наприклад, слово «Форд» може посилатися на колишнього президента США, виробника транспортного засобу, кінозірку, переправу через річку, або інший об'єкт.
- Аналіз тональності тексту передбачає розрізнення суб'єктивного (на відміну від фактичного) матеріалу і вилучення різних форм інформаційної поведінки: почуттів, думок, настроїв і емоцій. Методи аналізу текстів є корисними для аналізу, настрою на рівні суб'єкта, а також для розрізнення власника думки та об'єкта думки.

1.2.1. Класифікація.

Класифікація або рубрикація інформації (віднесення порції інформації до однієї або декількох категорій з обмеженої множини) є традиційним завданням організації знань і обміну інформацією.

Запропоновано багато методів для вирішення даного завдання за допомогою автоматичних процедур. Основне призначення даних методів — аналіз, класифікація та виявлення прихованих закономірностей у великих обсягах різномірних складно структурованих даних. Існуючі методи доцільно поділити на два принципово різних класи: методи машинного навчання і методи, засновані на знаннях (так званий “інженерний підхід”). При використанні методів машинного навчання для побудови класифікатора використовується колекція документів, заздалегідь відрубрикована людиною. Алгоритм машинного навчання будує процедуру класифікації документів на основі автоматичного аналізу заданої множини відрубрикованих текстів. При використанні методів, заснованих на знаннях, правила віднесення документа до тієї або іншої рубрики задаються експертами на основі аналізу рубрикатора і, можливо, частини текстів, що належать рубрикуванню.

Незважаючи на велику кількість запропонованих методів класифікації текстів, питання обрання такого методу, який був би простим в реалізації, ефективним, мав низькі обчислювальні витрати при навчанні та класифікації і високу якість класифікації в більшості реальних завдань, є актуальним завданням.

Варіанти використання для аналізу постів Twitter.

Метод класифікації можна використати для наступних задач:

- Сортування твітів за рубриками.
- Пошук твітів за конкретною темою.

Задача методів класифікації текстів полягає в тому, щоб якнайкраще обрати такі ознаки і сформулювати правила, опираючись на які буде прийматися рішення щодо віднесення документа до рубрики. Визначимо й опишемо основні з цих методів, до яких відносяться:

- класифікація за допомогою дерев рішень;
- Байєсівська (наївна) класифікація;
- класифікація методом опорних векторів;
- класифікація за допомогою методу найближчого сусіда;
- класифікація за допомогою штучних нейронних мереж.

1.2.2. Класифікація за допомогою дерева рішень.

Дерева рішень (decision trees) розбивають дані на групи на основі значень змінних простору ознак, внаслідок чого виникає ієрархія операторів “якщо-то”, які класифікують дані [3]. Для прийняття рішення, до якої категорії віднести даний документ, потрібно відповісти на питання, що знаходяться у вузлах цього дерева, починаючи з його кореня. Питання мають вигляд – “значення змінної x більше порога b ?”. Якщо відповідь позитивна, здійснюється перехід до правого вузла цього дерева, якщо негативна до лівого вузла. Наступне питання пов'язане з відповідним вузлом.

Для автоматичної побудови дерев рішень за допомогою навчання на прикладах розроблено ряд алгоритмів [4]. Розглянемо один з таких алгоритмів CLS. Цей алгоритм циклічно розбиває навчальні приклади на класи відповідно до змінної, що має найбільшу класифікуючу силу. Кожна підмножина прикладів, що виділяється такою змінною, знову розбивається на класи з використанням змінної з найбільшою класифікуючою здатністю і т. д. Розбиття закінчується тоді, коли в підмножині виявляються лише елементи з одного класу. Таким чином, утворюється дерево рішень.

Для визначення змінної з найбільшою класифікуючою силою використовується критерій інформаційної ваги слова в рубриці. Зазвичай після побудови “точного” дерева рішень до отриманого дерева застосовуються різні процедури усічення і перетворення дерева для того, щоб забезпечити баланс між складністю дерева (кількістю вузлів) і якістю навчання.

Одним з головних недоліків методу дерев рішень для задач класифікації текстів є той факт, що алгоритм побудови дерев рішень надає однакову вагу “позитивним” і “негативним” розгалуженням у вузлах. Велика кількість “негативних” гілок в описі рубрики може призводити до важко інтерпретуємих правил і “перенавчанню” алгоритму класифікації.

До переваг зазначеного методу слід віднести той факт, що побудоване дерево легко піддається аналізу, результат роботи алгоритму можна інтерпретувати в наочних термінах. Існують програми наочного графічного відображення дерев рішень.

1.2.3. Метод Байєсової (наївної) класифікації (Naive Bayes).

Наївний Байєсівський класифікатор традиційно використовується в задачах класифікації текстів, таких як фільтрація спаму, автоматична рубрикація або визначення тональності документа. Набули поширеного розвитку два його різновиди: багатомірний (multivariate) та мультиноміальний (multinomial). У загальному вигляді визначення найбільш вірогідного класу алгоритмом наївною байєсівської класифікації виглядає наступним чином:

$$C = \arg \max_{c \in C} P(C | o_1, o_2, \dots, o_n) = \arg \max_{c \in C} P(c) \prod P(o_i | c),$$

де C – набір класів, а o – набір ознак. Класифікація зводиться до обчислення максимального значення аргументу, при відомому наборі незалежних ознак.

При цьому:

$$P(c) \prod P(o_i | c) = P(C) P(o_1 | c) P(o_2 | c) \dots P(o_n | c).$$

Обчислення ймовірності класу $P(C)$ при відомих ознаках o зводиться до наступного:

$$P(C | o_1 o_2 \dots o_n) = \sum (o_1 o_2 \dots o_n) + 1 / \sum (C | A) + \sum A,$$

де A – набір відомих ознак, отриманих при навчанні класифікатора.

Класифікація тексту при цьому виглядає наступним чином:

$$C(T) = \max \sum (t_1, t_2 \dots t_{n1} | C),$$

де T – текст, що класифікується, а t – набір речень тексту. Так, приналежність тексту до того чи іншого класу зводиться до обчислення максимального значення суми коефіцієнтів приналежності реченням. Зазначений метод використовує ймовірнісну модель, в якій класифікація та включення у відповідну категорію документів проводиться шляхом оцінювання ймовірності появи слів у документі. Ймовірності можуть бути використані для оцінки найбільш близьких категорій тестового документа. Основні переваги наївного байєсівського класифікатора простота реалізації і низькі обчислювальні витрати при навчанні та класифікації. У тих рідкісних випадках, коли ознаки дійсно незалежні (або майже незалежні), наївний байєсівський класифікатор (майже) оптимальний. Основним недоліком методу є відносно невисока якість класифікації в більшості реальних завдань. Зазначений метод часто використовується в якості базового методу при порівнянні різних методів машинного навчання.

1.2.4. Метод опорних векторів

Метод опорних векторів (Support Vector Machine, SVM) використовує процес пошуку площини вирішення, яка може розділити позитивні і негативні приклади в багатовимірному просторі функції, в якому навчальні документи представлені як вектори. Цей метод розроблений В. Вапником в 1995 році, був вперше застосований до задачі класифікації текстів Торстеном Джохімсом. У своєму первинному вигляді алгоритм вирішував завдання розрізнення об'єктів двох класів. Метод набув величезну популярність завдяки своїй високій ефективності. Багато дослідників використовують його в роботах, присвячених класифікації текстів. Підхід, запропонований В. Вапником для визначення того, до якого з двох заздалегідь визначених класів повинен належати аналізований зразок, заснований на принципі структурної мінімізації ризику. Результати класифікації текстів за допомогою методу опорних векторів є одними з найкращих, у порівнянні з іншими методами машинного навчання [9]. Однак, швидкість навчання даного алгоритму одна з найнижчих. Метод опорних векторів вимагає великого обсягу пам'яті і значних витрат машинного часу на навчання, що знижує його масштабованість. Проте даний алгоритм можна використовувати як еталон з точки зору якості класифікації [10], так метод буде працювати ефективно, якщо опорних векторів буде порівняно не багато, якщо ж їх кількість зростає, метод стає малоефективний через значно збільшену складність.

1.2.5. Метод k-найближчого сусіда.

Метод k-найближчого сусіда (k-nearest neighbor). Цей метод є одним з найбільш вивчених і високоточних алгоритмів, що використовуються при створенні автоматичних класифікаторів. Вперше він був запропонований ще в 1952 році для вирішення завдань дискримінантного аналізу. У дослідженнях, присвячених аналізу роботи різних алгоритмів машинного навчання для задачі класифікації текстів, цей метод демонструє одні з найкращих результатів. В основі методу лежить досить проста ідея: знаходити та відрубрикувати колекції найбільш схожі на аналізований текстовий документи і, спираючись на знання про їх категоріальну приналежність, класифікувати невідомий документ. З метою

визначення рубрики, релевантною документу d , цей документ порівнюється з усіма документами з навчальної вибірки. Для кожного документу e з навчальної вибірки знаходиться вибірка косинус кута між векторами ознак:

$$\rho(d, e) = \cos(d, e)$$

Надалі з навчальної вибірки обираються k документів, найближчих до d (k -параметри). Для кожної рубрики обчислюється релевантність за формулою:

$$s(c_j, d) = \sum_{e \in \{k \text{ найближчих сусідів}\} \wedge c_j \in \text{Rub}(e)} \cos(d, e)$$

Рубрики, що мають релевантність вище деякого заданого порогу, вважаються відповідними документу. Параметр k зазвичай обирається в інтервалі від 0 до 100. При монотематичній категоризації обирається рубрика з максимальним значенням. Якщо ж документ може бути приписаний до декількох рубрик (випадок мультитематичної категоризації), класи вважаються відповідними, якщо значення перевищує деякий наперед заданий поріг. Головною особливістю, що виділяє зазначений метод серед інших, є відсутність у нього стадії навчання. Іншими словами, належність документа рубрикам визначається без побудови класифікуючої функції. Основною перевагою такого підходу є можливість оновлювати навчальну вибірку без перенавчання класифікатора. Ця властивість може бути корисною, наприклад, у випадках, коли навчальна колекція часто поповнюється новими документами, а перенавчання займає надто багато часу. Класичний алгоритм пропонує порівнювати аналізований документ з усіма документами з навчальної вибірки і тому головний недолік описаного методу полягає в тривалості часу роботи рубрикатора на етапі класифікації.

1.2.6. Штучні нейронні мережі

Штучні нейронні мережі (Artificial Neural Network). Штучні нейронні мережі набули широкого вивчення в галузі штучного інтелекту для аналізу даних з 1986 року. Вони представляють собою математичну модель, а також її програмні або апаратні реалізації, побудовані за подібністю мереж нервових клітин живого організму. Нейронні мережі це один з найбільш відомих і старих методів

машинного навчання. Штучні нейронні мережі це адаптивна система, яка складається з групи з'єднаних штучних нейронів. Система може бути навчена для зміни її внутрішніх станів, відображення зв'язків документів та їх категорій. Для ефективного проведення класифікації текстів необхідно визначити раціональну структуру і топологію нейронної мережі. Основні топології класифікуючих нейронних мереж це одно- і багат шаровий персептрон, нейромережевий Гаусів класифікатор, мережа Кохонена, мережа вбудованого розповсюдження, каскадна мережа. Всі вищевказані топології мають високу точність в обробці одночасно лінійних і нелінійних прикладів, але прийняття рішень щодо класифікації важко формалізуються у зв'язку з природою організації нейронної мережі та представляють нетривіальну задачу з урахуванням масштабованості з обмеженими обчислювальними ресурсами.

1.2.7. Кластеризація.

Одним із методів семантичної структуризації текстів є кластеризація. Метою кластеризації документів є автоматичне виявлення груп семантично схожих текстів серед заданої фіксованої множини. Кластеризація результатів пошуку дозволяє представити отримані результати у загальному вигляді, що спрощує процес виділення області, яка відповідає інформаційним потребам користувача. Тобто кластеризація представляє собою розбиття множини документів на кластери – підмножини, змістовні параметри яких наперед відомі. Сам кластер можна розглядати як групу документів зі схожими ознаками. Метою всіх методів кластеризації є те, щоб семантична схожість документів, які потрапляють до кластеру, була максимальною.

Числові методи кластеризації базуються на визначенні кластера як множини документів: а) значення семантичної близькості між будь-якими двома елементами якого не менше визначеного «порогу»; б) значення близькості між будь-яким документом множини і центроїдом (вектор, який обчислюється як середнє арифметичне векторів всіх документів кластера) цієї множини не менше визначеного «порогу».

В області інформаційного пошуку кластеризація застосовується для вирішення двох задач: групування документів та результатів пошуку. Для інформаційно-пошукових систем важливе зниження розмірності, що підвищує швидкість обробки та виконання швидкого пошуку. Якщо розбити цю множину документів на кластери, які містять семантично близькі один одному документи, то можна організувати процедуру порівняння документу-запиту з усіма текстами у вибраних кластерах за змістом.

Варіанти використання для аналізу постів Twitter. Метод кластеризації можна використати для задач виявлення груп семантично схожих твітів та сортування твітів.

1.2.8. Розпізнавання іменованих сутностей

Розпізнавання іменованих сутностей (РІС) (також відоме як ідентифікація об'єктної сутності, фрагментація об'єктної сутності та видобуток об'єктної сутності) — це підзадача видобування інформації, яка намагається знайти і класифікувати іменовані сутності в неструктурованому тексті в заздалегідь визначені категорії, такі як імена людей, організації, місця, медичні коди[en], час, кількості, грошові значення, відсотки тощо.

Більшість досліджень у системах РІС було структуровано як отримання не коментованого блоку тексту, такого як: “Джим купив 300 акцій корпорації Асме у 2006”. І створення коментованого блоку тексту, який виділяє імена об'єктів:

“[Джим](Особа) купив 300 акцій [корпорації Асме](Організація) у [2006](Час)”.

У цьому прикладі було виявлено та класифіковано ім'я особи, що складається з одного токена, назва компанії з двох токенів та часового виразу.

Варіанти використання для аналізу постів Twitter. Метод розпізнавання іменованих сутностей можна використати для виявлення іменованих сутностей в твітах, таких як: імена людей, місця, організації, аббревіатури, грошові значення та інше.

1.2.9. Усунення неоднозначності

Ми розуміємо, що слова мають різні значення залежно від контексту їх використання в реченні. Якщо говорити про людські мови, то вони теж неоднозначні, оскільки багато слів можна тлумачити різними способами залежно від контексту їх виникнення.

Розпізнавання сенсу слова в обробці природної мови (NLP) можна визначити як здатність визначити, яке значення слова активується вживанням слова в певному контексті. Лексична неоднозначність, синтаксична чи семантична, є однією з перших проблем, з якою стикається будь-яка система НЛП. Теги частини мови (POS) з високим рівнем точності можуть вирішити синтаксичну неоднозначність Word. З іншого боку, проблема вирішення семантичної неоднозначності називається WSD. Вирішити семантичну неоднозначність важче, ніж синтаксичну неоднозначність. Наприклад, розглянемо два приклади чіткого значення слова “кран” – “Кран підіймає вантаж”, “Ми не закрили кран, тому підлога стала мокрою”. Поява слова “кран” чітко вказує на відмінне значення. У першому реченні це означає будівельну машину, яка підіймає вантажі, а в другому – кран водопостачання, що знаходиться в будинку. За допомогою WSD можна визначити ці неоднозначності.

Варіанти використання для аналізу постів Twitter. Метод усунення неоднозначностей можна використати для усунення неоднозначностей (омограф) перед виявленням іменованих сутностей у твітах, такі як: імена людей, місця, організації, контактні дані, адреси, пошти та інше.

1.2.10. Аналіз тональності тексту.

Аналіз тональності тексту — клас методів контент-аналізу в комп'ютерній лінгвістиці, призначений для автоматизованого виявлення в текстах емоційно забарвленої лексики і емоційної оцінки авторів (думок) по відношенню до об'єктів, мова про які йде в тексті.

В сучасних системах автоматичного визначення емоційної оцінки тексту найчастіше використовується одномірний емотивний простір: позитив чи негатив

(добре або погано). Однак відомі успішні випадки використання і багатовимірних просторів.

Основним завданням в аналізі тональності є класифікація полярності даного документа, тобто визначення, чи є виражена думка в документі або пропозиції позитивними, негативними або нейтральними. Більш розгорнуто, «поза полярності» - класифікація тональності виражається, наприклад, такими емоційними станами, як «злий», «сумний» і «щасливий».

Класифікація за бінарною шкалою. Полярність документа можна визначати за бінарною шкалою. У цьому випадку для визначення полярності документа використовується два класи оцінок: позитивна чи негативна. Одним з недоліків цього підходу є те, що емоційну складову документа не завжди можна однозначно визначити, тобто документ може містити ознаки позитивної оцінки, так і негативної ознаки. Ранні роботи в цій царині включають в себе праці Терні і Панга, які застосовують різні методи розпізнавання полярності оглядів товару і відгуків про фільмах відповідно. Це приклад роботи на рівні документа.

Класифікація за багатосмуговою шкалою. Можна класифікувати полярність документа по багатосмуговій шкалою, що було зроблено Пангом і Снайдером (серед інших). Ними було розширене основне завдання класифікації кіно-відгуків від оцінки «позитивний або негативний» в бік прогнозування рейтингу по 3-х або 4-бальною шкалою. У той же час Снайдер провів поглиблений аналіз оглядів ресторанів, пророкуючи рейтинги їх різних властивостей, таких як їжа і атмосфера (за 5-бальною шкалою).

Системи шкалювання. Іншим методом визначення тональності є використання систем шкалювання, за допомогою чого словами, зазвичай пов'язаних з негативними, нейтральними або позитивними тональностями, ставляться відповідно числа за шкалою від -10 до 10 (від негативного до самого позитивного). Спочатку фрагмент неструктурованого тексту досліджується з допомогою інструментів та алгоритмів обробки природної мови, а потім виділені з цього тексту об'єкти та терміни аналізуються з метою розуміння значення цих слів.

Суб'єктивність/об'єктивність. Інший дослідницький напрямок - це ідентифікація суб'єктивності/об'єктивності. Це завдання зазвичай визначається як віднесення даного тексту в один з двох класів суб'єктивний й або об'єктивний. Ця проблема іноді може бути більш складною, ніж класифікація полярності: суб'єктивність слів і фраз може залежати від контексту, а об'єктивний документ може містити в собі суб'єктивні пропозиції (наприклад, новинна стаття, цитує думки людей). Більш того, результати більшою мірою залежать від визначення суб'єктивності, що вживається в рамках анотації текстів.

Модель більш докладного аналізу називається аналізом на основі функції/аспекту. Ця модель посиляється на ухвалу думок або настроїв, виражених різними функціями або аспектами сутностей, наприклад, у стільникового телефону, цифрової камери або банку. Властивість/аспект — це атрибут або компонент сутності, досліджуваної на тональність, наприклад, екран мобільного телефону або ж якість зйомки камери. Ця проблема вимагає вирішення ряду завдань, наприклад, ідентифікація актуальних сутностей, витяг їх функцій аспектів та визначення, є думка, висловлена по кожній функції/аспекту, позитивним, негативним або нейтральним. Більш докладні дискусії на цей рахунок можуть бути знайдені в довіднику з NLP, у главі «Аналіз тональності та суб'єктивності».

Підходи до класифікації тональності. Комп'ютери можуть виконувати автоматичний аналіз цифрових текстів, використовуючи елементи машинного навчання, такі як прихований семантичний аналіз, метод опорних векторів, «мішок слів». Більш складні методи намагаються визначити володаря настроїв (тобто людини) і мета (тобто сутність, щодо якої виражаються почуття). Щоб визначити думку з урахуванням контексту, використовують граматичні відносини між словами.

Відносини граматичної пов'язаності отримують на основі глибокого структурного розбору тексту. Аналіз тональності може бути розділений на дві окремі категорії:

- ручний (або аналіз тональності експертами);

- автоматизований аналіз тональності.

Найбільш помітні відмінності між ними лежать в ефективності системи і точності аналізу. У комп'ютерних програмах автоматизованого аналізу тональності застосовують алгоритми машинного навчання, інструменти статистики і обробки природної мови, що дозволяє обробляти великі масиви тексту, включаючи веб-сторінки, онлайн-новини, тексти дискусійних груп в мережі Інтернет, онлайн-огляди, веб-блоги та соціальні медіа.

Варіанти використання для аналізу постів Twitter. Метод аналізу тональності можна використати для наступних задач:

- Виявлення емоційного забарвлення твітів.
- Виявлення реакцій людей на певні новини, закони і т. д.

1.3.Прогнозування

Прогнозування (Forecasting). На основі особливостей існуючих даних оцінюються пропущені або ж майбутні значення цільових чисельних показників. Для вирішення таких задач широко застосовуються методи математичної статистики, нейронні мережі, методи Text Mining та інше.

Методи Text Mining в парі з методами Data Mining часто застосовуються для вирішення задач прогнозування. Є багато методів статистичного аналізу даних, наприклад, кореляційно-регресійний аналіз, кореляція рядів динаміки, виявлення тенденцій динамічних рядів, гармонійний аналіз. Методи Text Mining використовуються на стадії підготовки вхідних даних. Існуючі методи, які ми розглянули вище, дають можливість провести певний статистичний аналіз, наприклад, відсіяти непотрібні текстові дані. А потім зібрані статистичні дані які ми можемо використати для прогнозування.

1.4. Дослідження завдань Text Mining.

Як і більшість когнітивних технологій, Text Mining – це алгоритмічне виявлення раніше не відомих зв'язків та кореляцій у вже наявних текстових даних. Важливим завданням технології Text Mining є вилучення з тексту його характерних

елементів чи властивостей, які можна використовувати як метадані документа, ключових слів, анотацій.

Інше важливе завдання полягає у віднесенні документа до деяких категорій із заданої схеми їх систематизації. Text Mining також забезпечує новий рівень семантичного пошуку документів.

Можливості сучасних систем Text Mining можуть застосовуватися при управлінні знаннями для виявлення шаблонів у тексті, для автоматичного сортування інформації за заданими користувачем правилами та багато іншого.

1.5. Галузі використання Text Mining.

Технологія Text Mining використовується для управління знаннями в різних напрямках та галузях, і тому в кожному випадку має свою специфіку – це може бути використання в урядових, дослідних, корпоративних цілях.

Найбільш типові галузі використання Text Mining:

- Корпоративна бізнес-аналітика та Data Mining, корпоративна розвідка.
- Діловодство та електронні дослідження.
- Національна безпека та розвідка.
- Наукові дослідження, особливо екологія.
- Медицина.
- Дослідження мов та семантики.
- Видавнича справа.
- Автоматизоване розміщення реклами.
- Пошук інформації та надання доступу до інформації.
- Моніторинг соціальних медіа.

Пропоную детальніше розглянути декілька сфер використання.

Багато програмних пакетів інтелектуального аналізу тексту використовуються в системах безпеки, особливо для моніторингу та аналізу текстових джерел в Інтернеті, таких як інтернет-новини, блоги, тощо. Інтелектуальний аналіз тексту також присутній у вивченні шифрування/дешифрування тексту.

Гарним прикладом використання аналізу тексту є медицина. За наявності великих наборів даних пацієнтів у клінічній сфері, наборів даних демографічної інформації в популяційних дослідженнях та у звітах про побічні ефекти, інтелектуальний аналіз тексту може полегшити проведення клінічних досліджень та якість лікування. Алгоритми інтелектуального аналізу тексту можуть полегшити стратифікацію та індексацію конкретних клінічних подій у великих текстових наборах даних пацієнтів з симптомами, побічними ефектами та супутніми захворюваннями з електронних медичних записів, звітів про події та звітів з конкретних діагностичних тестів. Одним з он-лайн застосування інтелектуального аналізу текстів у біомедичній літературі є PubGene, загальнодоступна пошукова система, яка поєднує в собі біомедичний інтелектуальний аналіз тексту з візуалізацією мережі. GoPubMed — це пошукова система на основі знань для біомедичних текстів. Методи інтелектуального аналізу тексту також дозволяють витягувати невідомі знання з неструктурованих документів у клінічній сфері.

1.6. Дослідження програмних рішень для Text Mining.

В наш час багато провідних виробників програмного забезпечення пропонують свої продукти та рішення в області Text mining. Як правило - це масштабовані системи, в яких реалізовано різні математичні та лінгвістичні алгоритми аналізу текстових даних. Деякі мають графічні інтерфейси, багаті можливості візуалізації та маніпулювання з даними, надають доступ до різних джерел даних, функціонують в архітектурі клієнт-сервер. Приклади таких систем:

- Intelligent Miner for Text (IBM)
- TextAnalyst,
- WebAnalyst (Мегап'ютер Інтелідженс)
- Text Miner (SAS)
- SemioMap (Semio Corp.)
- Oracle Text (Oracle)
- Knowledge Server (Autonomy)
- InfoStream (Інформаційний центр "ЕЛВІСТІ")

- Orange
- Weka

Розглянемо деякі з цих систем докладніше.

1.6.1. Intelligent Miner.

Продукт фірми IBM Intelligent Miner for Text це набір окремих утиліт, що запускаються з командного рядка або скриптів незалежно один від одного. Ця система є одним із найкращих інструментів глибинного аналізу текстів. Система містить такі основні утиліти (Tools) для побудови програм управління знаннями:

Language Identification Tool - утиліта визначення мови - для автоматичного визначення мови, якою складено документ.

Categorisation Tool – утиліта класифікації – автоматичного віднесення тексту до деякої категорії (вхідною інформацією на навчальній фазі роботи цього інструменту може бути результат роботи наступної утиліти – Clusterisation Tool).

Clusterisation Tool - утиліта кластеризації - розбиття великої множини документів на групи по близькості стилю, форми, різних частотних характеристик ключових слів, що виявляються.

Feature Extraction Tool – утиліта визначення нового – виявлення у документі нових ключових слів (власні імена, назви, скорочення) на основі аналізу заданого заздалегідь словника.

Annotation Tool - утиліта "виявлення сенсу" текстів та складання рефератів - анотацій до вихідних текстів.

IBM Intelligent Miner for Text поєднує потужну сукупність інструментів, що базуються в основному на механізмах пошуку інформації (information retrieval), що є специфікою всього продукту. Система включає ряд базових компонентів, які мають самостійне значення поза межами технології "видобування текстів" - це інформаційно-пошукова система Text Search Engine, утиліта сканування Web-простору Web crawler, Net Question Solution - рішення для пошуку в локальному

Web-сайті або на декількох intranet/Internet-серверах, Java Sample GUI - набір інтерфейсів

Java Beans для адміністрування та організації пошуку на основі Text Search Engine.

Intelligent Miner for Text як продукт IBM включений в комплекс "Information Integrator for Content" для СУБД DB2 як засіб Information Mining ("глибинного аналізу інформації").

Вартість продуктів різних рівнів сімейства Intelligent Miner складає від 18 до 75 тисяч доларів.

1.6.2 TextAnalyst.

Компанія Мегап'ютер Інтеллідженс, відома своєю системою PolyAnalyst класу Data Mining, розробила також систему TextAnalyst, яка вирішує такі завдання Text Mining, як: створення семантичної мережі великого тексту, підготовка резюме тексту, пошук за текстом та автоматична класифікація та кластеризація текстів. Побудова семантичної мережі - це пошук ключових понять тексту та встановлення взаємовідносин між ними. За такою мережею можна зрозуміти, про що йдеться в тексті, і здійснити контекстну навігацію. Підготовка резюме - це виділення у тексті речень, у яких найчастіше зустрічаються значущі цього тексту слова. У 80% випадків цього цілком достатньо для отримання уявлення про текст. Для пошуку інформації в системі передбачено використання запитів природною мовою. На запит будується унікальна семантична мережа, яка при взаємодії з мережею документа дозволяє виділити потрібні фрагменти тексту. Кластеризація та класифікація проводяться стандартними методами видобутку даних.

Система Text-Analyst розглядає Text Mining як окремий математичний апарат, який розробники програмного забезпечення можуть вбудовувати у свої продукти, не опираючись на платформи інформаційно-пошукових систем або СУБД. Основна платформа для застосування системи

MS Windows 9x/2000/NT. Існує плагін TextAnalyst для браузера Microsoft Internet Explorer.

1.6.3. WebAnalyst.

Система WebAnalyst, продукт "Мегап'ютер Інтеллідженс", є інтелектуальним масштабованим клієнтом/серверним рішенням для компаній, які бажають максимізувати ефект аналізу даних у Web-середовищі.

Сервер WebAnalyst функціонує як експертна система збирання інформації та управління контентом Web-сайту. Модулі WebAnalyst вирішують три завдання: збір максимальної кількості інформації про відвідувачів сайту та запитуваних ними ресурсів; дослідження зібраних даних та генерація персоналізованого, на основі результатів досліджень, контенту. Вирішення цих завдань у сукупності має на думку розробників системи призвести до максимізації кількості нових відвідувачів Web-сайту та утримання вже наявних, а отже, збільшення популярності ресурсу. WebAnalyst здатний інтегрувати можливості Text Mining безпосередньо на web-сайт організації. Це дозволяє запропонувати індивідуалізований, автоматизований та цільовий маркетинг, автоматичний пошук та реалізацію перехресних продажів та розширити набір даних, що настроюються користувачем. По суті WebAnalyst є інтелектуальним сервером для додатків електронної комерції.

1.6.4. Text Miner.

Американська компанія SAS Institute випустила систему SAS Text Miner для порівняння певних граматичних та словесних рядів у письмовій мові. Text Miner – дуже універсальна, оскільки може працювати з текстовими документами різних форматів – у базах даних, файлових системах і навіть у Web.

Text Miner забезпечує логічну обробку тексту серед потужного пакету SAS Enterprise Miner. Це дозволяє користувачам збагачувати процес аналізу даних, інтегруючи неструктуровану текстову інформацію з існуючими структурованими даними - такими, як вік, прибуток і характер попиту.

Приклад успішного застосування логічних можливостей Text Miner демонструє компанія Compaq Computer Corp., яка в наш час тестує Text Miner, аналізуючи більше 2,5 гігабайт текстових документів, отриманих електронною

поштою та зібраних представниками компанії. Раніше обробити такі дані було практично неможливо.

Програма Text Miner дозволяє визначати, наскільки правдивим є той чи інший текстовий документ. Виявлення брехні в документах здійснюється шляхом аналізу тексту та виявлення змін стилю письма, які можуть виникати при спробі спотворити чи приховати інформацію. Для пошуку таких змін використовується принцип, що полягає у пошуку аномалій та трендів серед записів баз даних без з'ясування їхнього сенсу. При цьому Text Miner включений великий набір документів різного ступеня правдивості, структура котрих приймається як шаблони. Кожен документ, що "проганяється" на детекторі брехні аналізується і порівнюється з цими стандартами, після чого програма надає документу той чи інший індекс правдивості. Особливо корисною програма може стати в організаціях, які одержують великий обсяг електронної кореспонденції, а також у правоохоронних органах для аналізу показань нарівні з детекторами брехні, чия дія ґрунтується на спостереженні за емоційним станом людини.

Цікавим є приклад застосування Text Miner у медицині: в одній з американських національних лабораторій з охорони здоров'я було зібрано понад 10 тис. лікарських записів про захворювання серця, зібрані з клінік по всій країні. Аналізуючи ці дані за допомогою Text Miner, фахівці виявили деякі адміністративні порушення у звітності, а також змогли визначити взаємозв'язок між серцево-судинними захворюваннями та іншими недугами, які не були визначені традиційними методами.

Разом з тим компанія SAS зазначає, що випустить свій продукт Text Miner в основному для привернення уваги бізнес-інтелігенції.

1.6.5. SemioMap.

SemioMap - це продукт компанії Entrieva, створений в 1996 р. вченим семіотиком Клодом Фогелем (Claude Vogel). У травні 1998 р. продукт був випущений як промисловий комплекс SemioMap 2.0 – перша система Text Mining, що працює в архітектурі клієнт-сервер. Система SemioMap складається із двох

основних компонентів - сервера SemioMap і клієнта SemioMap. Робота системи протікає у три фази:

- Індекссування; сервер SemioMap автоматично читає масиви неструктурованого тексту, витягує ключові фрази (поняття) та створює з них індекс;
- Кластеризація понять; сервер SemioMap виявляє зв'язок між витягнутими фразами і будує їх, з урахуванням спільної зустрічаємості, лексичну мережу (" понятійну карту ");
- Графічне відображення та навігація; візуалізація карт зв'язків, що забезпечує швидку навігацію за ключовими фразами та зв'язками між ними, а також можливість швидкого звернення до конкретних документів.

SemioMap підтримує розбиття матеріалу за папками, створення окремої бази даних для кожної папки. Зв'язки між поняттями, які виявляє SemioMap, базуються на спільній зустрічності фраз в абзацах вихідного текстового масиву.

Центральним блоком SemioMap є лексичний екстрактор - програма, яка отримує фрази з текстової сукупності і виявляє спільну зустрічальність цих фраз (їх взаємні зв'язки). Лексичний екстрактор базується на патентованій технології SEMIOLEX. Вона реалізує ідеї обчислювальної семіотики, науки про знаки у мовній комунікації, розробленої Клодом Фогелем.

1.6.6. Oracle Text.

Засоби Text Mining, починаючи з Text Server у складі СУБД Oracle 7.3.3 та картриджа interMedia Text у Oracle8i, є невід'ємною частиною продуктів Oracle. У Oracle9i ці кошти розвинулися і отримали нову назву – Oracle Text – програмний комплекс, інтегрований у СУБД, що дозволяє ефективно працювати із запитамі, що належать до неструктурованих текстів. При цьому обробка тексту поєднується з можливостями, наданими користувачеві для роботи з реляційними базами даних. Зокрема, при написанні програм для обробки тексту стало можливим використовувати SQL.

Основним завданням, на вирішення якого націлені кошти Oracle Text, є завдання пошуку документів за їх змістом - словами або фразами, які при потребі комбінуються з використанням булевих операцій. Результати пошуку ранжуються за релевантністю, з урахуванням частоти слів запиту в знайдених документах. Для підвищення повноти пошуку Oracle Text надає низку засобів розширення пошукового запиту, серед яких можна виділити три групи. По-перше, це розширення слів запиту всіма морфологічними формами, що реалізується залученням знань про морфологію мови. По-друге, Oracle Text допускає розширення слів запиту близькими за змістом словами за рахунок підключення тезаурусу - семантичного словника, а також розширення запиту словами, близькими за написанням та звучанням - нечіткий пошук і пошук співзвучних слів. Нечіткий пошук доцільно застосовувати при пошуку слів з друкарськими помилками, а також у тих випадках, коли виникають сумніви в правильному написанні - прізвища, назви організації тощо.

Система Oracle Text забезпечує проведення тематичного аналізу текстів англійською мовою. Під час обробки текст кожного документа піддається процедурам лінгвістичного та статистичного аналізу, внаслідок чого визначаються його ключові теми та будуються тематичні резюме, а також загальне резюме – реферат.

Всі ці засоби можуть використовуватися спільно, що підтримується мовою запитів у поєднанні з традиційним синтаксисом SQL і PL/SQL для пошуку документів. Oracle Text надає можливість працювати з сучасними реляційними СУБД у контексті складного багатоцільового пошуку та аналізу текстових даних.

Можливості обробки текстової інформації російською мовою в Oracle Text досить обмежені. Для вирішення цієї проблеми компанією ГарантПарк-Інтернет було розроблено модуль Russian Context Optimizer (RCO), призначений для спільного використання з interMedia Text (або Oracle Text). Крім підтримки російськомовної морфології, RCO включає засоби нечіткого пошуку, тематичного аналізу та реферування документів.

1.6.7. Orange.

Orange – це компонентний програмний пакет (ПП) для візуального програмування, візуалізації даних, машинного навчання, обміну та аналізу даними з відкритим вихідним кодом. Використання візуального програмування дозволяє обробляти, моделювати та досліджувати дані на основі використання різноманітних функцій методами інтелектуального аналізу даних та машинного навчання.

Orange складається з інтерфейсу Canvas, використовуючи який користувач може створювати робочий процес аналізу даних на основі поєднання різноманітних віджетів.

Основні компоненти ПП Orange називаються віджетами, і варіюються від простої візуалізації даних, вибору підмножини та попередньої обробки, до емпіричної оцінки алгоритмів навчання та прогнозного моделювання.

Візуальне програмування реалізується через інтерфейс, в якому робочі процеси створюються за допомогою об'єднання заздалегідь визначених або призначених для користувача віджетів. Комбінації віджетів утворюють структури аналітики даних. Існує більше 100 віджетів, які охоплюють більшість стандартних і спеціалізованих задач аналізу даних.

Досвідчені користувачі можуть використовувати Orange як бібліотеку Python для маніпулювання даними та редагування віджетів.

Віджети пропонують базові функції такі як читання даних, відображення таблиць, вибір функцій, порівняння алгоритмів навчання, візуалізація елементів даних тощо. Користувач може інтерактивно досліджувати візуалізації або передавати вибрану підмножину даних в інші віджети.

1.6.8. Weka.

Weka — це набір засобів візуалізації та алгоритмів для аналізу даних і вирішення задач прогнозування, разом з графічною оболонкою для доступу до них.

Weka дозволяє виконувати такі завдання аналізу даних, як підготовку даних (preprocessing), відбір ознак (feature selection), кластеризацію, класифікацію, регресійний аналіз та візуалізацію результатів.

Основним інтерфейсом користувача є Explorer, хоча ті ж функціональні можливості підтримуються з командного рядка та інтерфейсу Knowledge Flow. Для систематичного порівняння різних алгоритмів машинного навчання використовується інтерфейс Experimenter. Він дозволяє порівнювати результати не лише різних алгоритмів на одному наборі даних, а й одного алгоритму на різних наборах даних.

Інтерфейс Explorer містить наступні панелі:

Панель попереднього опрацювання уможливорює імпорт даних з бази даних, текстових файлів у форматі CSV, а також попереднє опрацювання цих даних за допомогою різноманітних алгоритмів (фільтрів). Ці фільтри використовуються для трансформування даних, а також для видалення певних атрибутів.

Панель класифікації надає можливість застосувати алгоритми класифікації та регресійного аналізу до обраного набору даних, візуалізувати та оцінити результати, відобразити ROC криві тощо.

Панель асоціації надає доступ до методів, які дозволяють оцінити взаємозв'язки між атрибутами.

Панель кластеризації містить різноманітні методи кластеризації, наприклад метод кластеризації методом k-середніх, ЕМ-алгоритм тощо.

Панель вибору атрибутів дозволяє ідентифікувати атрибути, які найбільш впливають на якість прогнозування.

Панель візуалізації відображає точкові діаграми.

1.7. Порівняльна характеристика програмних продуктів.

У таблиці 1 наведено порівняння програмних продуктів Text Mining.

Таблиця 1. Порівняння програмних продуктів

Програмний продукт	Preprocess	Associate	Cluster	Summarize	Categorize	API
Комерційні програми						

Intelligent Miner for Text (IBM)	+	+		+		
TextAnalyst	+	+	+		+	+
WebAnalyst	+		+	+		
SemioMap	+	+	+		+	
Oracle Text	+	+	+	+		+
Text Miner	+	+		+	+	
Програми у вільному доступі						
Orange	+	+	+	+	+	+
Weka	+	+	+	+	+	

Аналіз характеристик програмних продуктів показав, що кожен ПП має свої переваги, але беззаперечним лідером, що відповідає всім характеристикам є програмний продукт Orange, окрім цього він є у вільному доступі, тому пропонуємо розглянути його детальніше.

РОЗДІЛ 2. ВИЯВЛЕННЯ ПРОБЛЕМ ВИКОРИСТАННЯ TEXT MINING ТА ДОСЛІДЖЕННЯ ШЛЯХІВ ЇХ ВИРІШЕННЯ

2.1. Дослідження етапів процесу Text Mining в Orange.

В розділі 2.1 досліджено етапи Text Mining з використанням програмного продукту Orange, та виявлено проблеми з якими можуть зіштовхуватись аналітики, а також наведено шляхи їх вирішення.

Розглянемо *використання методу кластеризації для аналізу та групування твітів по ключових словах*, на прикладі пошуку твітів на тему "машинне навчання", що дасть можливість звужити область пошуку завдяки створеним групам. Дана тема є актуальною, оскільки, в останні роки напрямок машинного навчання розвивається дуже стрімко. Відносно нещодавно у вільному доступі з'явився ресурс під назвою GPT chat. Це чат-бот зі штучним інтелектом, розроблений лабораторією OpenAI. Являє собою велику статистичну модель мови, оптимізовану для ведення діалогів та відлагоджену завдяки технікам навчання з учителем та навчання з підкріпленням. Це великий крок в розвитку машинного навчання, саме тому є активною темою для обговорень у Twitter.

Розглянемо етапи процесу Text Mining пошуку твітів на тему "машинне навчання". Можна поділити наступні етапи технології:

- Пошук інформації та виявлення вихідних даних.
- Попередня обробка документів.
- Використання методів аналізу тексту.
- Інтерпретація результатів.

2.2. Пошук інформації та виявлення вихідних даних.

Першим етапом Text Mining є пошук інформації та виявлення вихідних даних. Це підготовчий крок, який включає збирання або виявлення набору текстових матеріалів для аналізу. Такі матеріали можуть міститися в Інтернеті, базах даних, файлових системах або системах керування контентом.

В роботі інформація отримана за допомогою Twitter API, тому перш за все пропоную ознайомитися з Twitter як джерелом даних, та зручністю його використання.

Твіттер — соціальна мережа, яка є мережею мікроблогів, дає змогу користувачам надсилати короткі текстові повідомлення (твіти) (до 280 символів, до 2017 року — до 140 символів), використовуючи SMS, служби миттєвих повідомлень і сторонні програми-клієнти. Літературний сегмент твіттера породив такий різновид короткотекстової літератури, як твіттература (англ. Twitterature). Твітер є неймовірно великим джерелом інформації. Через різке зростання повідомлень, сервіс «Твіттер» вже двічі пережив твітапокаліпсис. Перший в червні 2009, коли ID-номери записів почали плутатися у зв'язку з перевищенням максимального значення 32-розрядних цілих чисел зі знаком (більше за $2^{31} - 1 = 2\,147\,483\,647$). Другий у вересні 2009-го, коли значення тих же записів перевищило найбільше можливе значення беззнакового цілого числа ($2^{32} - 1 = 4\,294\,967\,295$). Кількість користувачів вже налічую більше 1.3 млрд. і вона продовжує збільшуватися. За статистикою кожного дня публікується понад 500 млн. нових твітів. Саме тому Twitter є неймовірно великим джерелом даних які можна використати для обробки та статистичного аналізу тексту з використанням технології Text Mining. Взаємодія з соціальною мережею може здійснюватись за допомогою API (Application Programming Interface). Деякі програмні продукти, наприклад, Orange, для роботи з алгоритмами Text Mining мають інтеграцію з Twitter через Twitter API. Це дозволяє без додаткових маніпуляцій отримати потрібні для дослідження дані. Нажаль, не всі програмні продукти мають подібні можливості, тому була розроблена власна програма для отримання, обробки та збереження твітів з використанням мови програмування JavaScript. Для її розробки було одержано доступ до акаунту розробника твітер - <https://developer.twitter.com>.

2.3. Отримання доступу до Twitter API

Twitter API можна використовувати для програмного отримання й аналізу даних Twitter. Протягом багатьох років API Twitter розширювався, додаючи додаткові рівні доступу для розробників і академічних дослідників, щоб мати можливість масштабувати свій доступ для покращення та вивчення публічних дискусій.

Нещодавно було випущено Twitter API v2. Twitter API версії 2 містить сучасну основу, нові та вдосконалені функції та швидкий доступ до Essential access.

Щоб отримати доступ до Twitter API необхідно зареєструвати акаунт розробника. Для цього необхідно подати заявку з інформацією про себе та про цілі використання Twitter API на розгляд спеціалістів з Twitter. Після розгляду заявки, якщо вона відповідає умовам використання та політиці компанії, буде надано доступ. Існує чотири рівні доступу, загалом вони обмежують об'єми використання такі як: кількість символів в пошуковому запиті, кількість можливих пошуків та інше. Кожен рівень доступу має власні правила використання. Наразі аналізувати їх не має сенсу, оскільки вони постійно змінюються. Для дослідження використано рівень доступу “Academic Research”, який надається науковцям та студентам.

	Essential	Elevated	Elevated+ (coming soon)	Academic Research
Getting access	More information soon	Apply for additional access within the developer portal	Need more? Sign up for our waitlist	Apply for additional access
Price	TBD	TBD		TBD
Access to Twitter API v2	✓	✓		✓
Access to standard v1.1	✓ (Limited access - only media endpoints)	✓		✓
Access to premium v1.1	✗	✓		✓
Access to enterprise	✗	✓		✓
Project limits	1 Project	1 Project		1 Project
App limits	1 App per Project	3 Apps per Project		1 App per Project
Tweet caps	Retrieve up to 500k Tweets per month	Retrieve up to 2 million Tweets per month		Retrieve up to 10 million Tweets per month
Filtered stream rule limit	5 rules	25 rules		1000 rules
Filtered stream rule length	512 characters	512 characters		1024 characters
Filtered stream POST rules rate limit	25 requests per 15 minutes	50 requests per 15 minutes		100 requests per 15 minutes
Search Tweets query length	512 characters	512 characters		1024 characters
Access to full-archive search Tweets	✗	✗		✓
Access to full-archive Tweet counts	✗	✗		✓

Рис.2.1. Рівні доступу

Після отримання доступу необхідно створити додаток у власному кабінеті. Це необхідно зробити для отримання ключа та пароля, які будуть використовуватися для роботи з Twitter API в рамках стороннього програмного продукту чи засобами мов програмування.

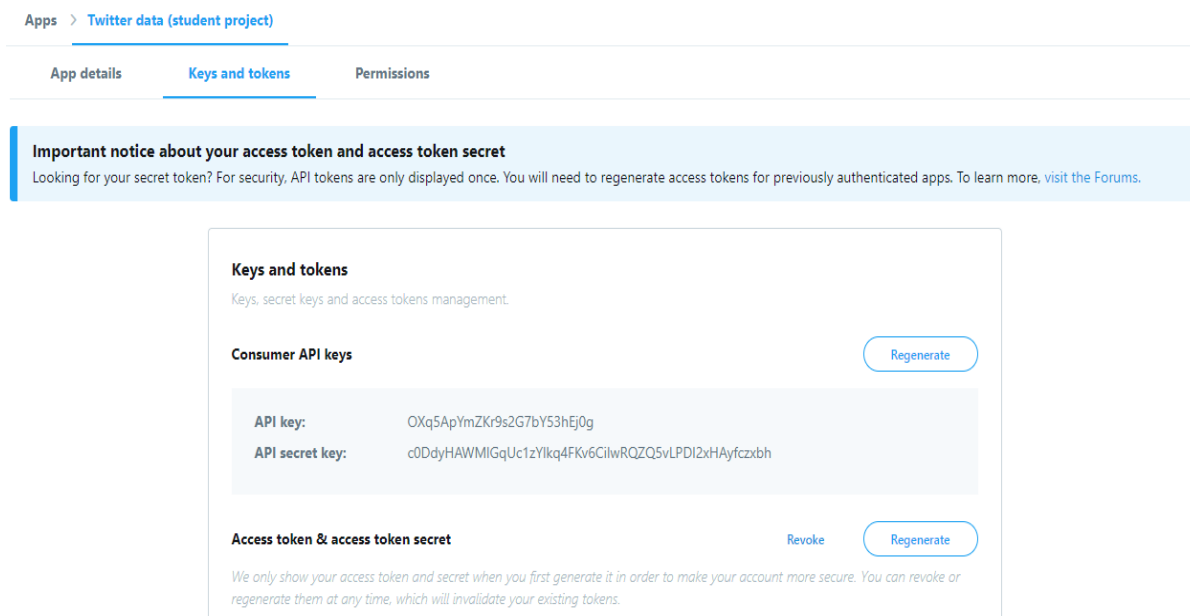


Рис.2.2. Отримані ключ та пароль

2.4. Підключення та робота з Twitter API в Orange

На початку роботи з Orange, перш за все, необхідно створити робочу модель, для цього підключаємо модуль Orange3 Text.

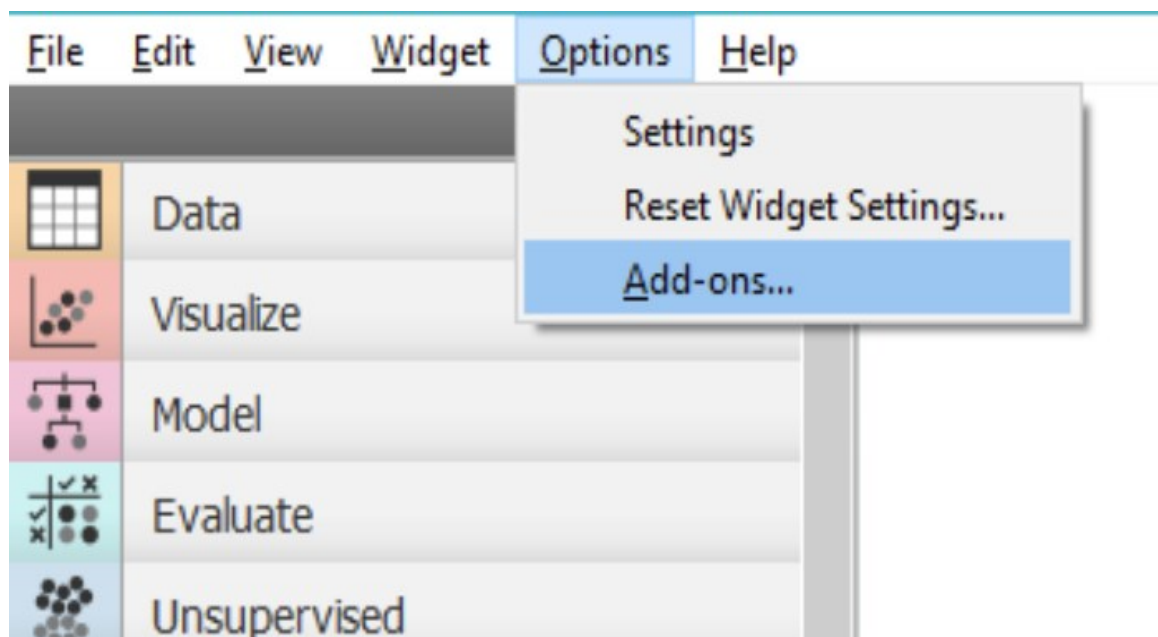


Рис.2.3. Підключення модуля Orange3 Text

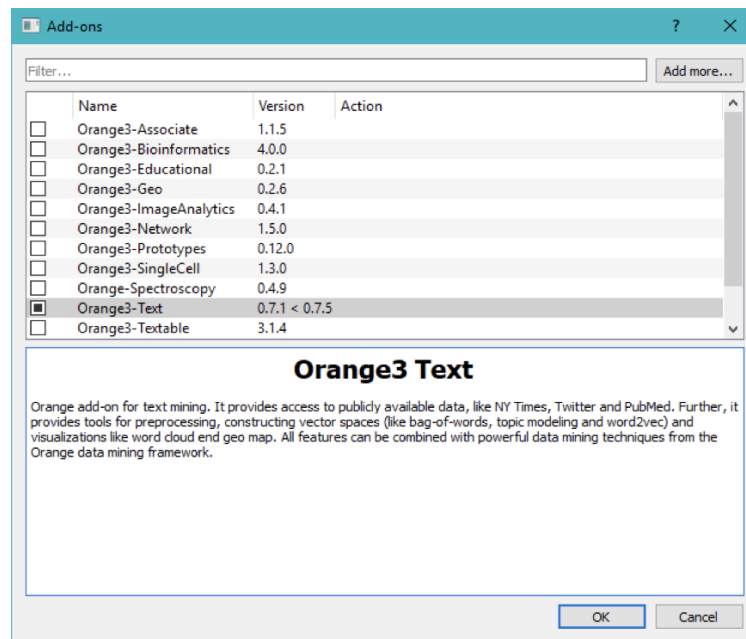


Рис.2.4. Підключення модуля Orange3 Text

Модуль Orange3 Text надає доступ до загальнодоступних ресурсів, таких як NY Times, Twitter, PubMed. Окрім цього, він надає інструменти для попередньої обробки, побудови векторних просторів та візуалізацій даних, такі як карта геоміток та хмари з слів. Усі функції можна поєднувати з іншими методами Data Mining із системи Orange.

На наступному кроці необхідно додати віджети у модель. Першим буде віджет Twitter.

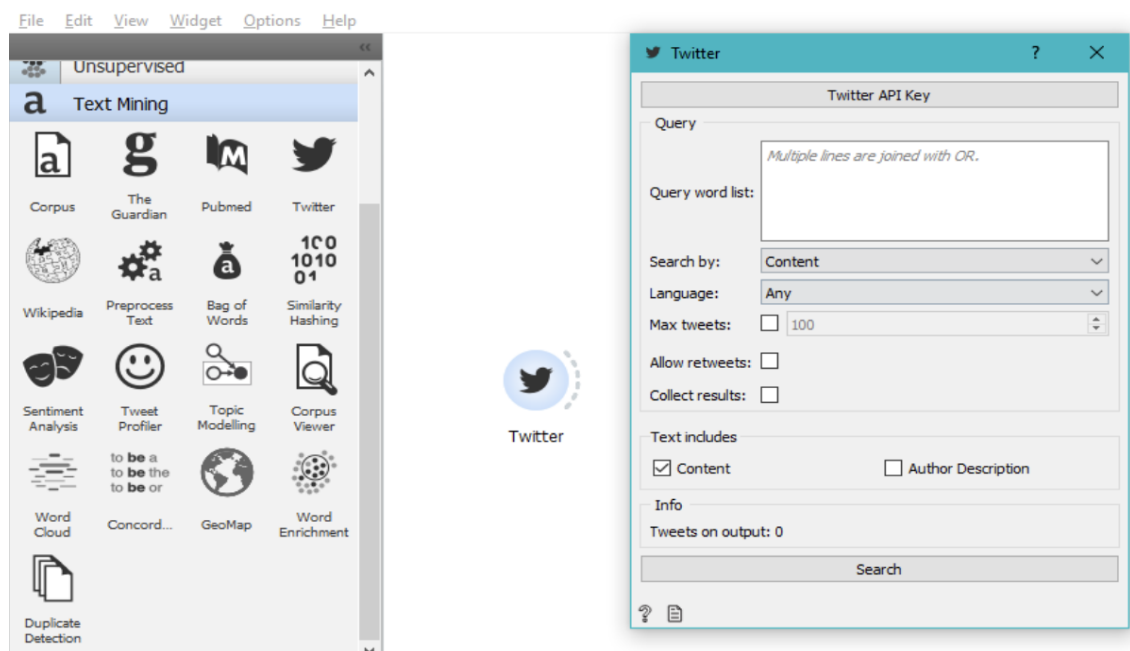


Рис.2.5. Віджет Twitter

Віджет Twitter дає можливість робити різноманітні запити пошуку твітів за допомогою Twitter API. Для його роботи необхідно підключити Twitter API з використанням отриманих в особистому кабінеті розробника Twitter ключа та паролю.

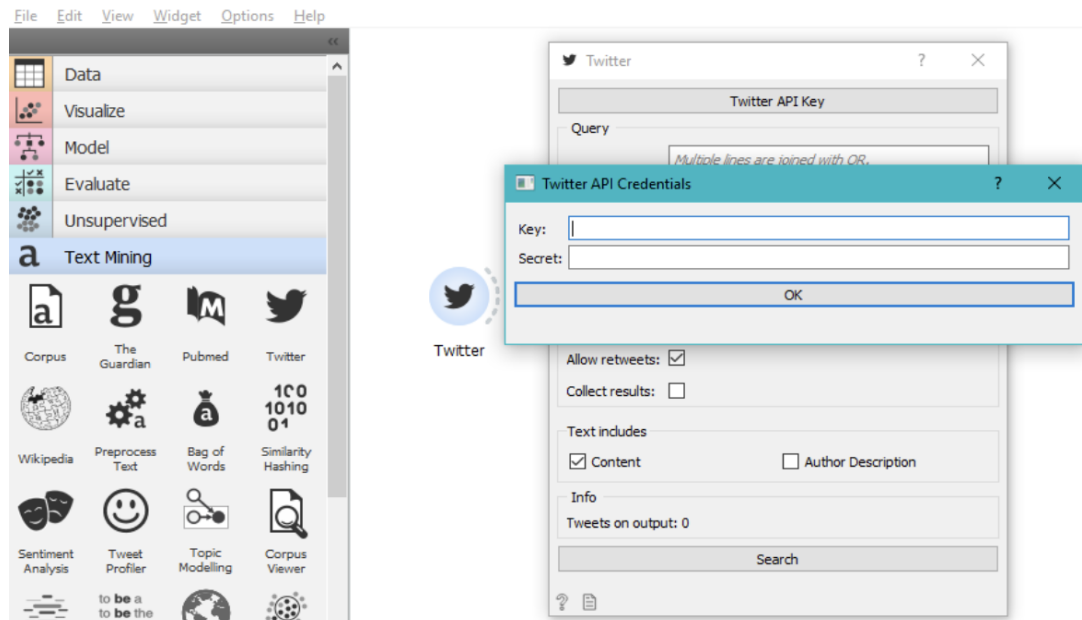


Рис.2.7. Вікно Twitter API Credentials

Наступний крок - це введення ключових слів для пошуку твітів. Пошук здійснюється шляхом натискання кнопки Search для пошуку інформації в Twitter.

Оскільки соціальна мережа Twitter міжнародна, то більшість твітів написано англійською мовою. На рис. 2.6. наведено пошук 100 твітів англійською мовою на тему "машинне навчання".

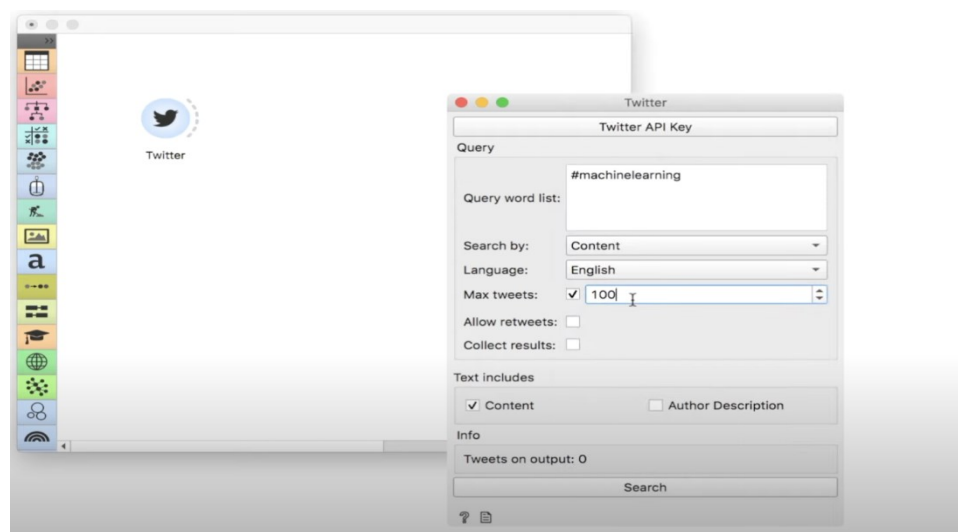


Рис.2.6. Пошук твітів на тему "машинне навчання"

2.5. Проблеми використання Twitter API.

Розглянемо проблеми, які виникають при використанні Twitter API.

Отже, як ми бачимо, проблеми є навіть на одному з перших етапів - отриманні доступу. В п.2.1 ми дослідили що Twitter однозначно є неймовірно великим джерелом даних, яке можна використати за допомогою API.

Політика компанії Twitter змінюється, тому з'являються труднощі в отриманні доступу, що може завадити можливості використання Twitter API. Але ця проблема вирішується шляхом ознайомлення з "Політикою використання Twitter API". На даний момент існує рівень доступу "Academic Research". Він надається студентам та науковцям. Наразі, його достатньо для проведення даного дослідження.

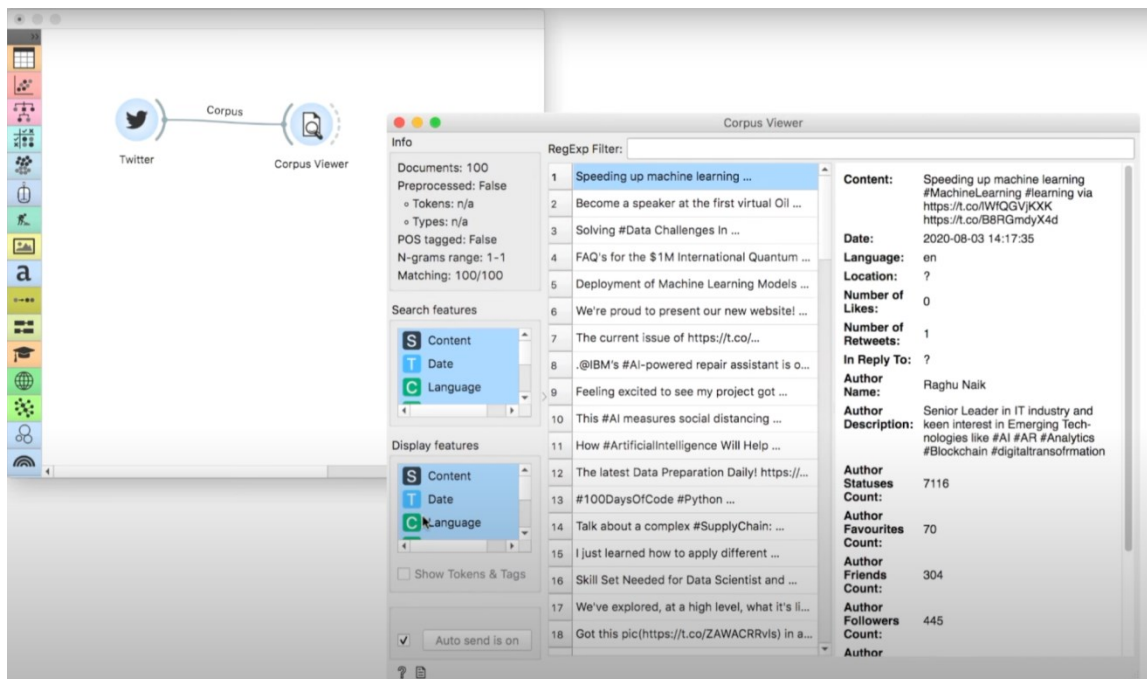
Щодо інших досліджень, вище наведені усі рівні доступу, тому якщо задачі потребують розширених можливостей, то можна обрати інший рівень доступу під час реєстрації. Якщо цілі дослідження відповідають політиці компанії, то доступ буде надано, а якщо ні, то ви будете змушені шукати інше джерело даних.

2.6. Попередня обробка документів.

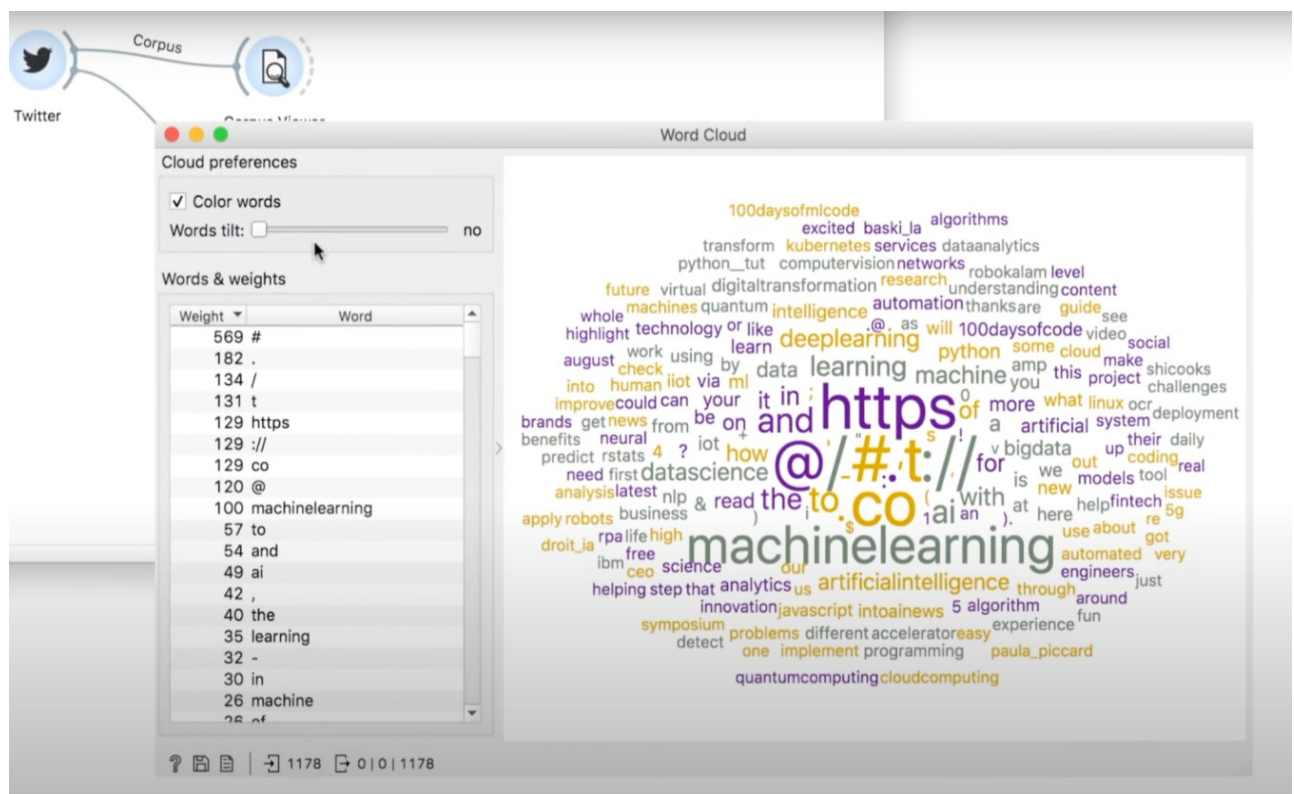
На цьому кроці виконуються найпростіші, але необхідні перетворення з документами для представлення їх у вигляді, з яким працюють методи Text Mining. Метою таких перетворень є видалення зайвих слів і надання тексту більш строгої форми.

В попередньому розділі ми розглянули різні програмні рішення для Text Mining. Кожен програмний продукт має функціонал що дає можливість підготувати текст до аналізу.

Вище наведено способи підключення до Twitter API та пошук твітів. Результати пошуку зручно переглянути за допомогою елементу Corpus Viewer.



Також ми можемо проаналізувати частоту зустрічі певних слів за допомогою елемента Word Cloud. Як бачимо, в хмарі відображується багато зайвих слів та знаків.



Зайві слова та знаки відфільтровані за допомогою елементу Preprocess Text.

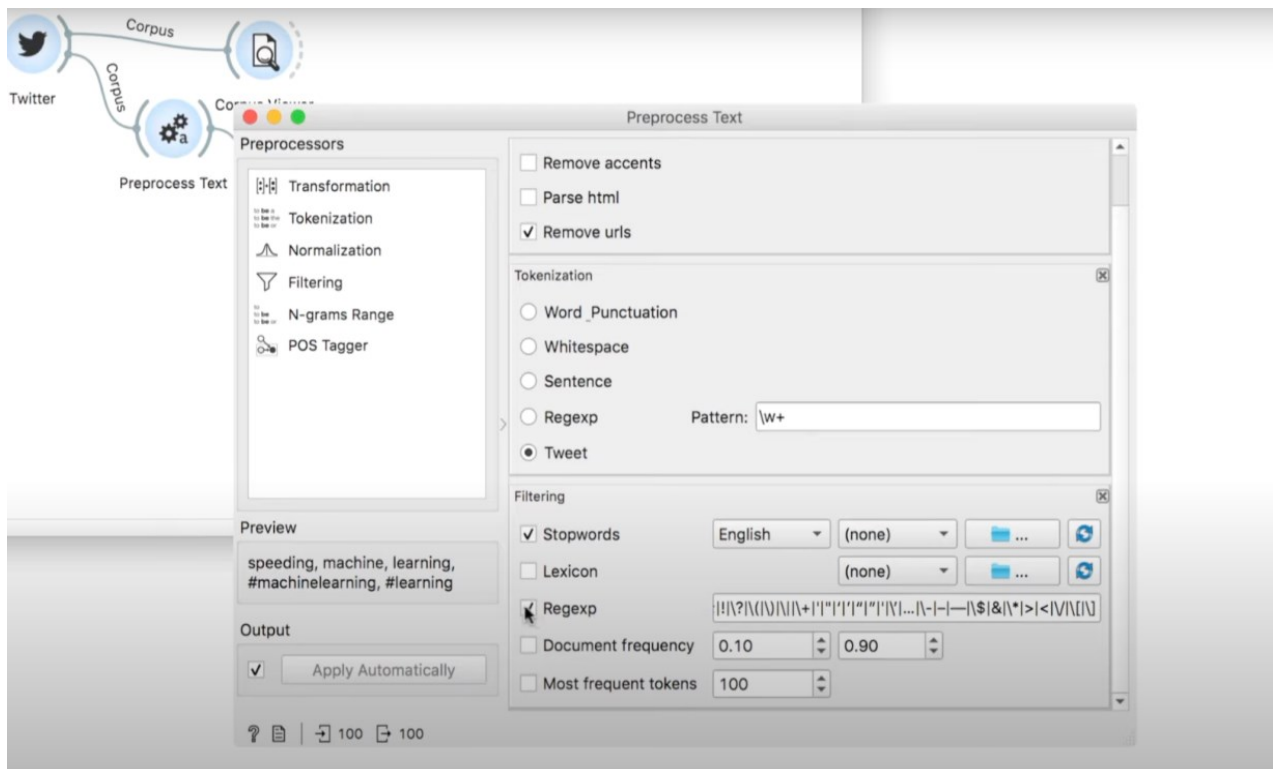


Рис.2.10. Фільтрація слів у твітах

Після фільтрації хмара слів відображається значно зрозуміліше. Чим більшим є слово у хмарі, тим частіше воно згадується в тексті твітів.

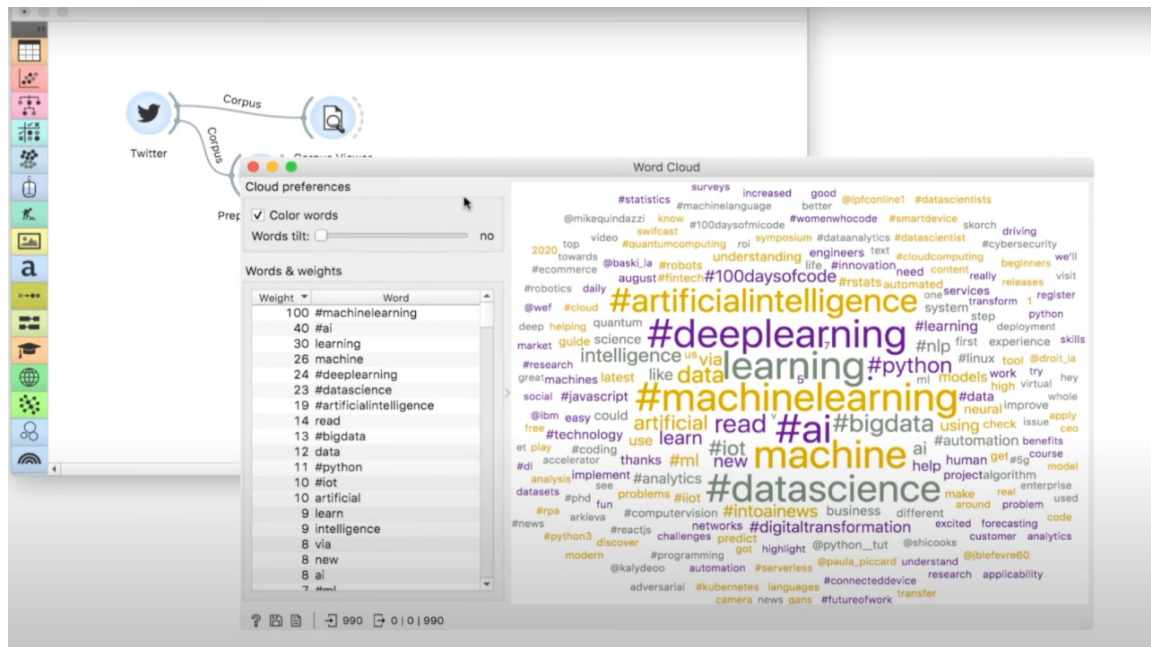


Рис.2.11. Відфільтрована хмара слів

2.7. Аналіз проблем попередньої обробки документів

Як бачимо, на прикладі ПП Orange, програмні продукти для Text Mining мають функціонал для попередньої обробки документів. Orange, дає можливості трансформувати та фільтрувати документи, що детальніше показано на Рис.3.1.

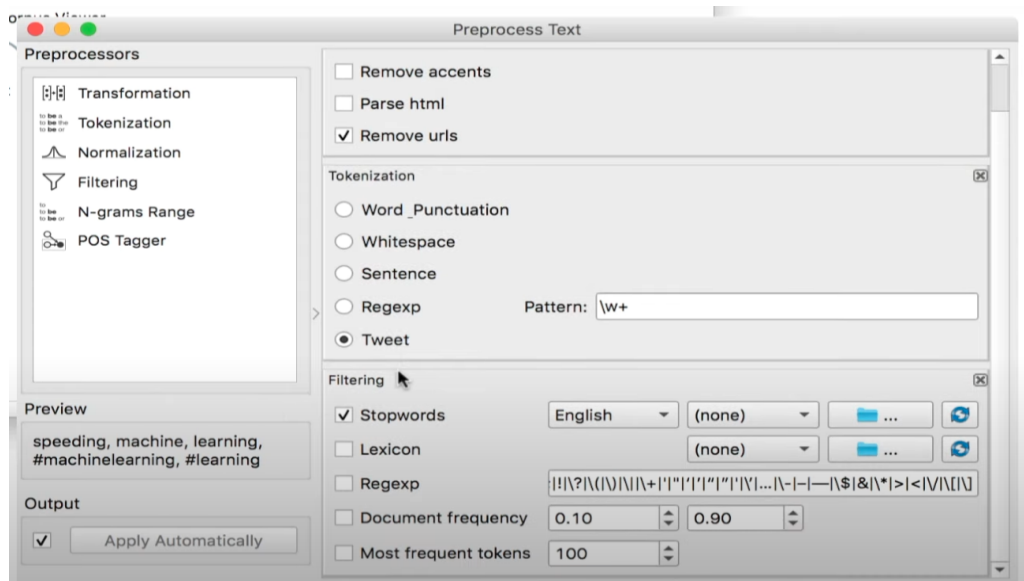


Рис.2.12. Обробка тексту

Нажаль, функціонал є обмеженим, тобто ми не можемо додати власні алгоритми обробки документів. Тому в деяких випадках необхідно буде використовувати сторонні програмні продукти чи мови програмування для підготовки документів до текстового аналізу.

2.8. Використання методів аналізу тексту та інтерпретація результатів.

Пропоную використати метод кластеризації для групування твітів по ключовим словам. Для цього використано інструмент Topic Modeling

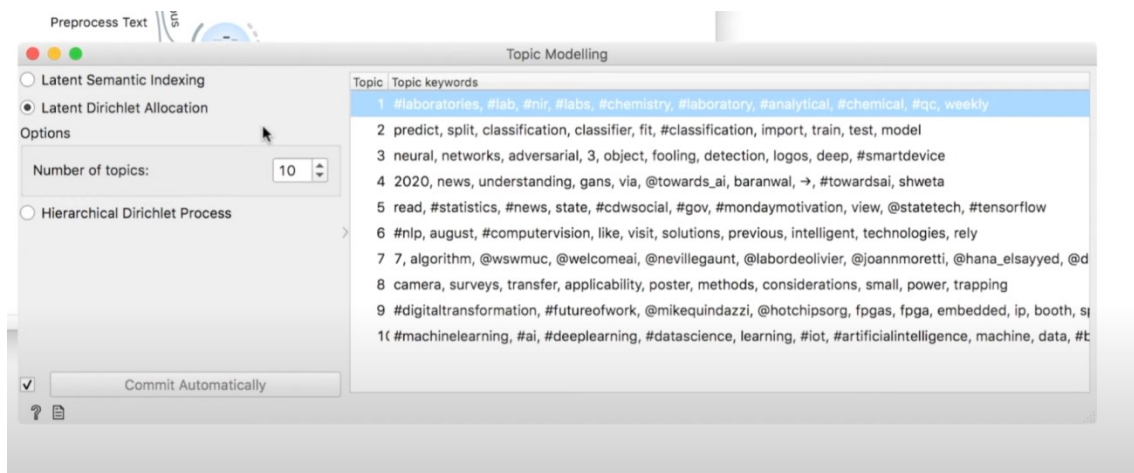


Рис.2.13. Сформовані групи твітів по ключовим словам

Далі ми можемо детальніше ознайомитись з результатами групування, наприклад, розглянемо твіти однієї з груп. Для цього використаємо інструмент Corpus Viewer.

RegExp Filter:	
1 Do you want to know how ...	Topic 1: 0
2 Hey all, I'm presenting a poste...	Topic 2: 0
3 The current issue of https://t.c...	Topic 3: 0
4 Latest from #containerjournal ...	Topic 4: 0
5 Top 8 #MachineLearning ...	Topic 5: 0
6 Going LIVE on Twitch now. Joi...	Topic 6: 0
7 Feeling excited to see my ...	Topic 7: 0
8 #MachineLearning can be a ...	Topic 8: 0.912533
9 The latest Data Preparation ...	Topic 9: 0
10 Read the latest 'Connecting th...	Topic 10: 0.0510954
11 Hey I am Bhagya Lakshm, For ...	Content: Do you want to know how @saraassicura rationalized #cybersecurity services and #net infrastructure with a bundle of @Cisco products and #Italtel services? Thanks to a highly innovative project? Curious? https://t.co/e5h66EFmgi #MachineLearning #cloud #datadriven https://t.co/Xl9G3isKcF
	Date: 2020-08-03 14:00:03
	Language: en
	Location: ?
	Number of Likes: 1
	Number of Retweets: 2
	In Reply To: ?
	Author Name: Italtel
	Author Account ufficiale Twitter per le comunicazioni Cor-

Рис.2.13. Перегляд результатів з 1 групи

Як бачимо до цієї групи було віднесено 11 твітів. Також ми бачимо що алгоритм розпізнав що цей твіт на 91% належить до 8 групи відповідно до ключових слів.

2.9. Проблеми на етапі інтерпретації результатів

Розглянувши приклад використання методу кластеризації для групування твітів за ключовими словами була досягнута поставлена ціль. Але зазвичай в дослідження на цьому не закінчуються. Далі маємо проаналізувати результати, та зробити певні висновки, можливо навіть потрібно якось відреагувати. Існуючі програмні продукти дозволяють нам проаналізувати результати, але далі ми мусимо вручну якось відреагувати то зробити певні дії. Тобто існуючі програмні продукти не дають можливості описати певні алгоритми та автоматично аналізувати та реагувати на результати текстового аналізу.

2.10. Розв'язок проблем використання програмних засобів для текстового аналізу

В попередніх пунктах розглянуті всі етапи процесу текстового аналізу на прикладі програмного продукту Orange, як бачимо у використанні програмних продуктів є певні проблеми. Загалом вони пов'язані з обмеженою функціональністю. Було розглянуто один з найкращих програмних продуктів і як бачимо, навіть він не задовольняє всіх потреб. Виявлені проблеми наведені у таблиці 2.

Таблиця 2. Проблеми при роботі з Text Mining

№	Етап	Проблеми
1	Пошук інформації та виявлення вихідних даних.	Труднощі отримання доступу до Twitter API. Відсутність інтеграції з Twitter API
2	Попередня обробка документів.	Обмежені можливості попередньої обробки та фільтрації.
3	Інтерпретація результатів.	Не має можливості автоматичного аналізу результатів та реагування

Пропоную розглянути шляхи вирішення кожної з виявлених проблем.

2.10.1 Труднощі отримання доступу до Twitter API.

Для отримання доступу наразі ми маємо створити акаунт розробника. Під час реєстрації ми маємо відповісти на питання стосовно цілей використання Twitter API. Перша за все потрібно проаналізувати цілі та методи які ми будемо використовувати, та перевірити чи відповідають вони політиці користування Twitter API. Якщо ваші цілі не відповідають політиці компанії, то ви будете змушені шукати інше джерело даних. Але якщо ми не порушуємо жодних правил то зможемо отримати доступ. На жаль, ми не можемо повпливати на політику компанії Twitter.

2.10.2 Відсутність інтеграції з Twitter API

Нажаль, не всі програмні продукти мають інтеграції з Twitter API, тому пропоную розробити власну програму для отримання, обробки та збереження твітів за певною темою з використанням мови програмування JavaScript. Пропоную використати бібліотеку Twitter, вона надає можливість під'єднатися та взаємодіяти з Twitter API. В першій версії для економії часу пропоную програму консольну, тобто буде працювати за допомогою викликів команд в терміналі. В такому випадку існують певні обмеження. Для її запуску необхідно встановити на комп'ютер Node - це середовище, що дає можливість запускати програми написані мовою програмування javascript. В подальших версіях, можна розробити графічний інтерфейс. Перевагою консольної програми є можливості використання її як частини інших програмних продуктів, оскільки, за допомогою мов програмування ми зможемо викликати дану команду в рамках інших розробок. Звичайно головною метою є, використання цієї консольної команди для отримання даних для подальшого текстового аналізу в тих випадках коли програмний додаток не має інтеграції з Twitter API. Окрім цього, ми зможемо будь-яким чином використати її в рамках будь-якого стороннього проекту для отримання даних з Twitter. Код програми наведено в Додатку А.

2.10.3. Обмежені можливості попередньої обробки текстів

Програмні продукти для текстового аналізу мають обмежений список інструментів для попередньої обробки даних. На щастя, є інструменти, які є універсальними, та за допомогою яких ми можемо як завгодно обробляти та підготувати тексти до аналізу. Таким інструментом є мови програмування JavaScript. Ця мова має вбудовані функції для роботи з текстовими даними, окрім цього існує неймовірно велика кількість сторонніх бібліотек які ми можемо встановити за допомогою NPM (Node Package Manager).

2.10.4. Автоматична інтерпретація результатів текстового аналізу та реагування.

Безумовно, перед початком текстового аналізу ми ставимо певні цілі, та після проведення аналізу зазвичай аналізуємо результати та певним чином інтерпретуємо їх. Існуючі програмні продукти зазвичай мають лише методи що полегшують інтерпретацію результатів за рахунок графічного відображення. Однак, аналіз можна автоматизувати за допомогою мови програмування JavaScript. Зазвичай результатом текстового аналізу є певні текстові або числові дані. Пропоную використати мову програмування для аналізу результатів, оскільки це дає можливість автоматично реагувати на результат. Наприклад, відправляти повідомлення на пошту за допомогою служб розсилок, або реалізувати власний алгоритм графічного відображення результатів, або ж провести будь які інші дії залежно від результату.

2.11 Дослідження способів використання JavaScript для Text Mining

Вище розглянуто приклад використання програмного продукту для текстового аналізу. Як бачимо, більшість проблем з якими ми зіштовхуємось в процесі текстового аналізу - це функціональна обмеженість програмних продуктів. На щастя, є інструменти, які є універсальними та надають там більш широкі можливості. Таким інструментом є мова програмування Javascript. На даний момент ця мова програмування є доволі популярною, окрім цього вона не є дуже складною для вивчення. Безумовно, завдяки графічним інтерфейсам, існуючі програмні продукти для текстового аналізу є більш простими у використанні. Однак, вони є обмеженими тому великих досліджень вони не підходять. Зазвичай текстовий аналіз є лише частиною певного дослідження. В такому випадку результати текстового аналізу вручну переносять, наприклад, в інший програмний продукт де проводять подальші дослідження. Мови програмування дають можливість використовувати текстовий аналіз в рамках проектів чи певних розробок. Наприклад якщо ми є розробником додатку, то можна використати

технологію Text Mining в рамках цього додатку. Пропоную розглянути інструменти та бібліотеки які необхідні для організації процесу Text Mining, засобами мови програмування.

Мова програмування JavaScript має велику кількість бібліотек, в тому числі і бібліотеки для текстового аналізу. Для під'єднання до Twitter API, пропоную використовувати бібліотеку Twitter. Для збереження отриманих документів в файли - бібліотеку FS. Бібліотека Text-Miner, включає в себе реалізовані методи текстового аналізу, які можна імпортувати та використовувати. Щодо інтерпретації результатів, то використання бібліотек залежить від задач для яких було використано текстовий аналіз. Наприклад, якщо досліджується частота згадувань певної події, то в такому випадку можна відображати користувачу певну таблицю зі статистикою.

2.12. Висновок до розділу 2.

В розділі 2 розглянуто приклад використання програмних продуктів для вирішення задач текстового аналізу.

Виявлено проблеми використання програмних продуктів для роботи з Text Mining та запропоновано способи їх вирішення.

Обґрунтовано доцільність використання мов програмування для вирішення задач текстового аналізу.

Таким чином, використання мов програмування для вирішення задач текстового аналізу є набагато універсальнішим рішенням ніж використання програмних продуктів. Даний підхід дозволяє використовувати текстовий аналіз технології Text Mining для розв'язання певних задач під час розробки інших комплексних програмних продуктів.

В наступному розділі розглянуто приклад такого використання.

РОЗДІЛ 3. ВИКОРИСТАННЯ JAVASCRIPT ДЛЯ TEXT MINING

3.1. Постановка задачі

Наразі в 2022-2023 роках відбувається дуже багато змін в світовій політиці, технологіях тощо. В інформаційному просторі розповсюджується дуже багато думок від різних людей, в тому числі в соціальній мережі Твіттер, люди висловлюють свою думку стосовно різних подій. Для багатьох людей, не залежно від того чим вони займаються, цікаво та важливо знати думки інших людей стосовно різноманітних подій. Пропоную розглянути можливість реалізації Telegram-бота який на основі ключових слів, за допомогою метода аналізу тональності формує звіт стосовно реакції людей на подію описану в ключових словах.

3.2. Розробка Телеграм бота

Створений Телеграм-бот має дві частини, клієнтську та серверну. Клієнтською частиною виступає інтерфейс месенджеру Telegram.

Серверну частину було розроблено окремо, за допомогою мови програмування Javascript (текст програмного коду наведено в Додатку А). Взаємодія клієнтської та серверної частини відбувається по протоколу HTTPS.

Для початкової версії бота використано BotFather. Це бот, який дає можливість створювати власних ботів. Для створення власного бота достатньо лише відправити декілька команд в BotFather. Після чого автоматично створюється новий бот з заданою назвою.

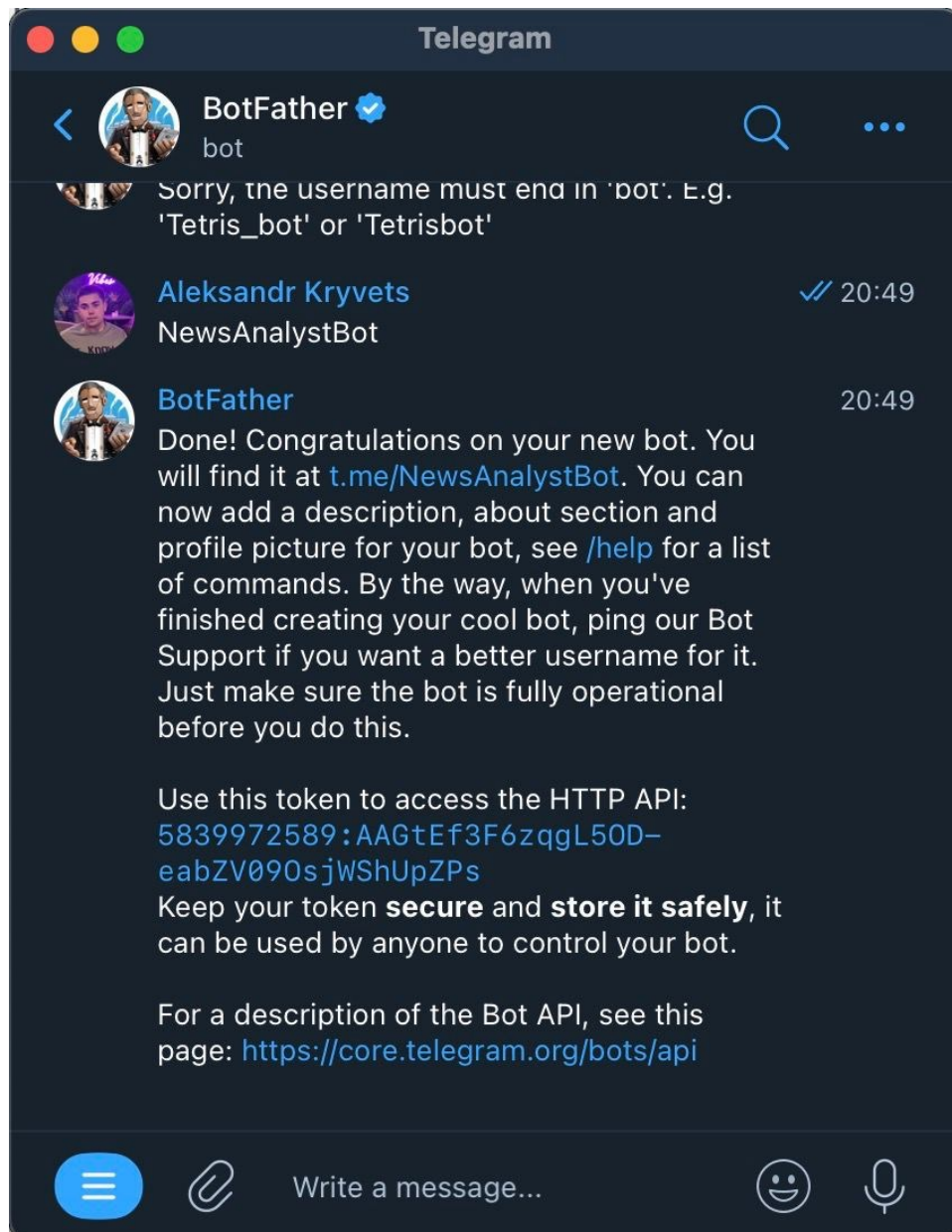


Рис.3.10. Створення нового бота, за допомогою BotFather

Після створення бота надається ключ, який дає можливість взаємодіяти з ботом за допомогою серверу.

Щоб розпочати роботу з ботом на отримати головне меню, необхідно відправити команду `/start`, або натиснути на кнопку Start.

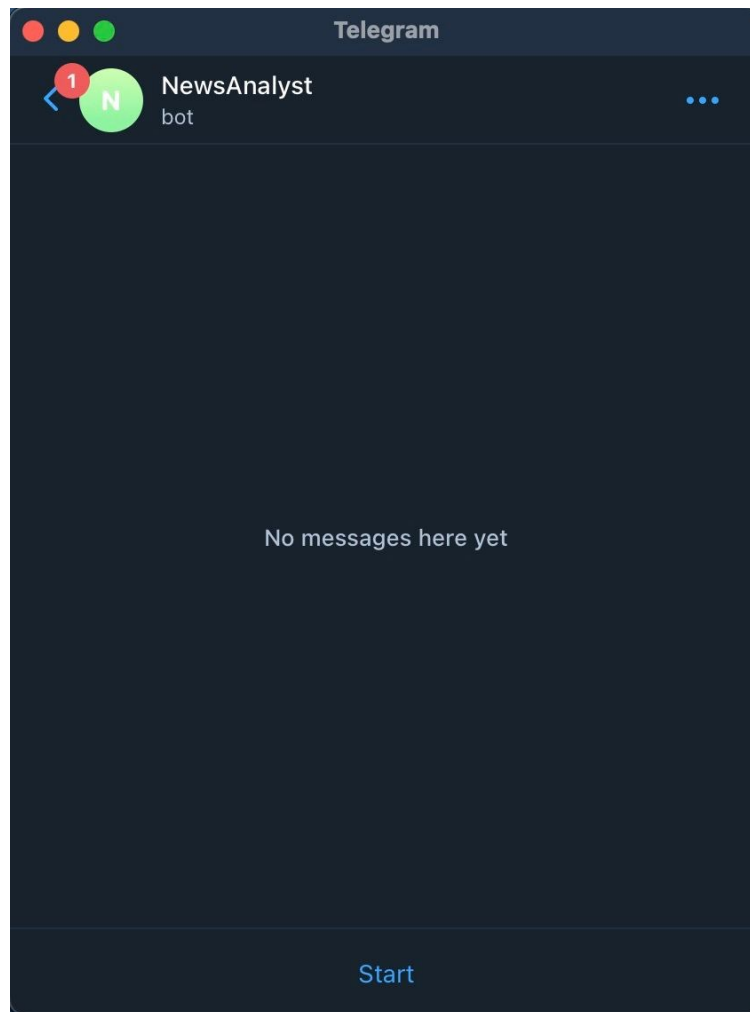


Рис.3.11. Початок роботи з ботом

Для написання серверної частини було використано мову програмування Javascript, а також бібліотеку Telegraf. Дана бібліотека дає можливість підключитися до телеграм-боту за допомогою ключа, та реалізувати підписки на певні команди. Таким чином, було реалізовано головне меню для взаємодії з ботом, за такими пунктами:

- Налаштувати слова для пошуку - для налаштування ключових слів які будуть використовуватися для пошуку;
- Отримати статистику - для отримання статистики емоційного забарвлення твітів відповідно заданих ключових слів.

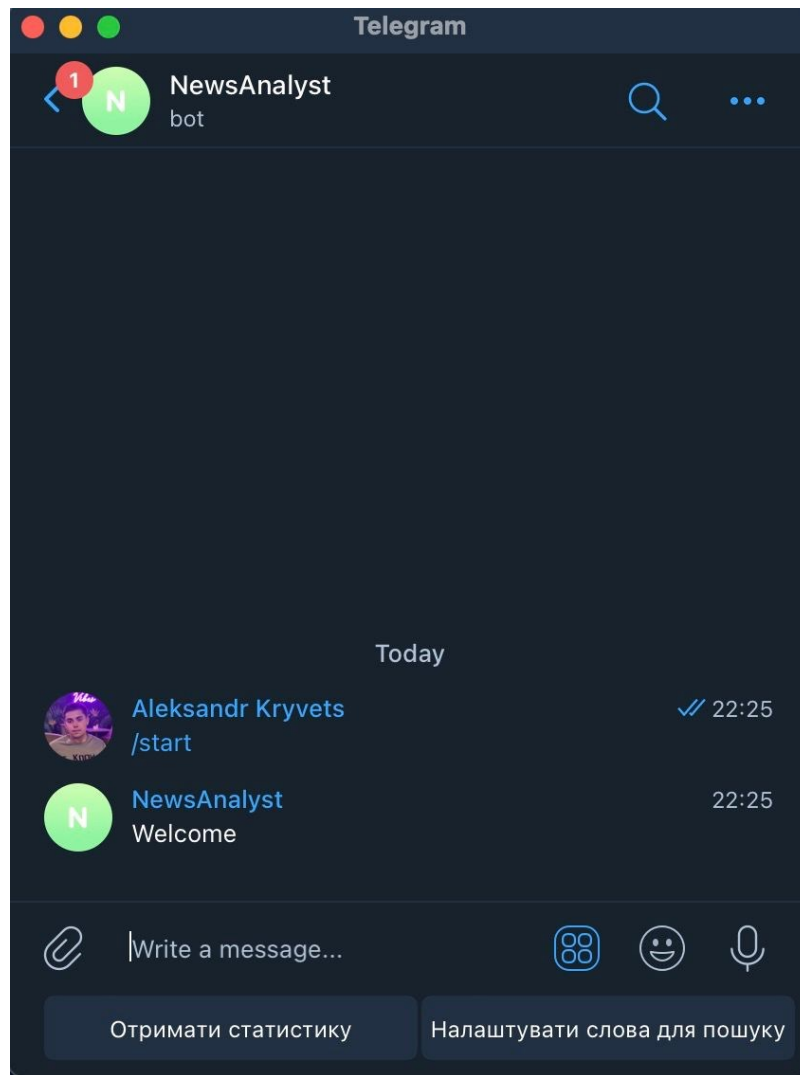


Рис.3.12. Головне меню бота

Ключові слова зберігаються в змінній відповідно індексу користувача, тому після перезапуску серверу, вся історія зникає.

Після натискання на кнопку Отримати статистику відбувається запит до Twitter API, та отримуються твіти відповідно заданих ключових слів, це реалізовано за допомогою бібліотеки Twitter.

Отримані твіти сортуються стосовно емоційного забарвлення за допомогою бібліотеки Text-Miner. Після чого, за допомогою вбудованих можливостей мови програмування рахується статистика: кількість проаналізованих твітів, відсоток позитивних твітів, відсоток негативних твітів, відсоток нейтральних твітів.

Після чого сформований звіт відправляється користувачу.

Створений бот дає можливість аналізувати реакцію населення планети на будь які події на основі даних отриманих з Твітер.

Останнім часом, актуальним є обговорення криптовалюти Bitcoin, пропоную, за допомогою створеного боту, провести аналіз емоційного забарвлення твітів на цю тему. В даному випадку ключовими словами для пошуку є: Bitcoin, Cryptocurrency. Для налаштування ключових слів натискаємо на відповідний пункт меню «Налаштування слів для пошуку», та вводимо ключові слова, через «,», після чого відправляємо їх.



Рис.3.13. Налаштування ключових слів

Після налаштування ключових слів, обираємо пункт меню «Отримати статистику». Через певний проміжок часу, сервер надіслав звіт (Рис. 3.14).

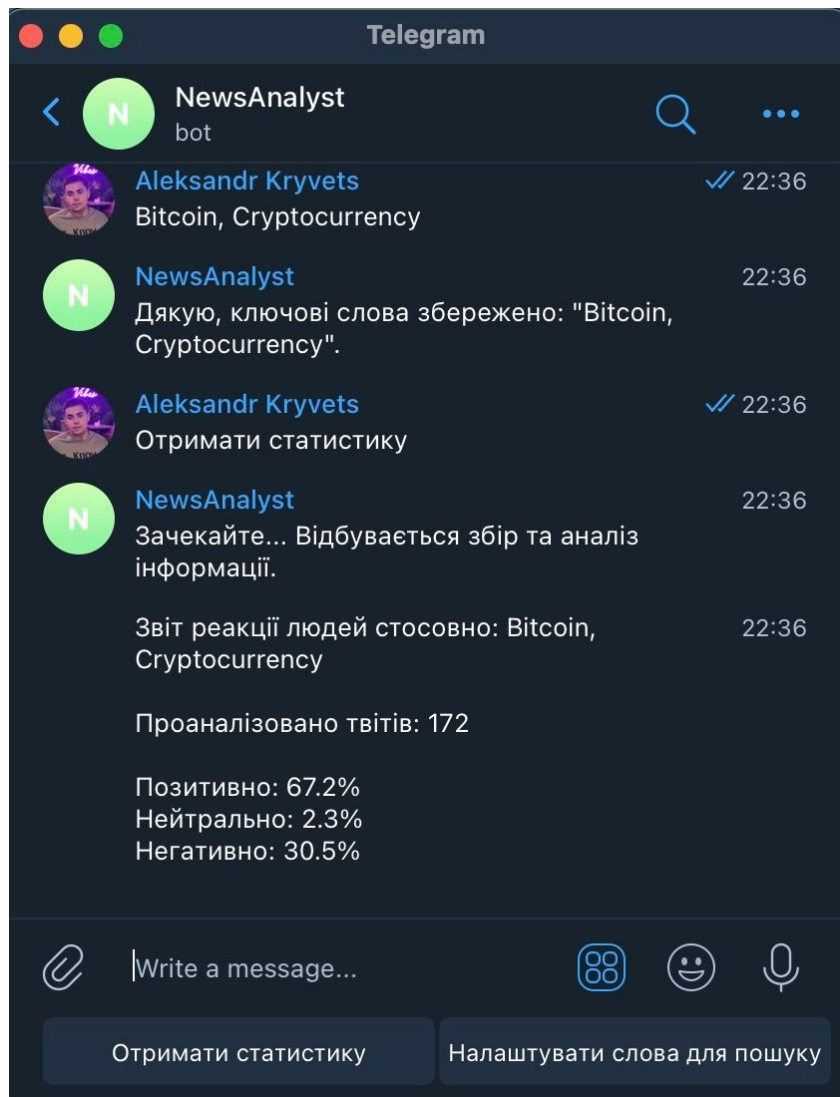


Рис.3.14. Отриманий звіт реакцій

Отриманий звіт включає в себе інформацію про кількість проаналізованих твітів, відсоток позитивних, відсоток негативних та відсоток нейтральних твітів.

Таким чином, *можна зробити висновок* що аудиторія Twitter в більшості випадків згадує Bitcoin в позитивній формі, у своїх твітах.

3.3. Висновки до розділу 3

В даному розділі розглянуто приклад використання мови програмування для аналізу твітів. На основі месенджера Telegram, розроблено бот, який використовує метод аналізу тональності тексту технології Text Mining. Таким чином, використання мови програмування зробило значно універсальнішим застосування методів текстового аналізу технології Text Mining.

ВИСНОВКИ

В магістерській роботі досліджено технологію Text Mining та можливості її використання для статистичного аналізу та прогнозування на основі даних отриманих з соціальної мережі Twitter. Досліджено варіанти отримання даних з Twitter та способи їх використання.

Досліджено програмні продукти для роботи з методами Text Mining та для подальшого дослідження обрано ПП Orange.

За його допомогою, реалізовано приклад використання методів технології Text Mining для статистичного аналізу тексту на основі даних отриманих з Twitter.

Виявлено та описано проблеми використання Twitter API, програмних продуктів для роботи з технологією Text Mining, та запропоновані способи їх вирішення.

Досліджено можливості та обґрунтовано цілі використання мов програмування для розв'язку задач Text Mining.

Реалізовано Telegram-бот, який на основі пошуку коротких текстових повідомлень соціальної мережі Twitter здійснює аналіз їх емоціонального забарвлення і формує статистичний звіт.

Розробка чат-боту для месенджера Telegram здійснена з використанням методу аналізу тональності тексту технології Text Mining та мови програмування JavaScript.

Результати, наведені у дослідженні можуть бути використані при розробці додатків для інтелектуального аналізу тексту отриманого з мережі Інтернет, соціальних мереж, інших джерел.

ДОДАТКИ

ДОДАТОК А «ФРАГМЕНТИ КОДУ ПРОГРАМ»

1. Консольний додаток для отримання даних з Twitter:

```
const Twitter = require('twitter');
const fs = require('fs');
const TWITTER_CONSUMER_KEY = 'YOUR_CONSUMER_KEY';
const TWITTER_CONSUMER_SECRET = 'YOUR_CONSUMER_SECRET';
const TWITTER_ACCESS_TOKEN_KEY = 'YOUR_ACCESS_TOKEN_KEY';
const TWITTER_ACCESS_TOKEN_SECRET = 'YOUR_ACCESS_TOKEN_SECRET';
const SEARCH_QUERY = '#javascript';

const client = new Twitter({
  consumer_key: TWITTER_CONSUMER_KEY,
  consumer_secret: TWITTER_CONSUMER_SECRET,
  access_token_key: TWITTER_ACCESS_TOKEN_KEY,
  access_token_secret: TWITTER_ACCESS_TOKEN_SECRET
});

function writeTweetsToCSV(tweets) {
  const fields = ['created_at', 'text', 'user.screen_name'];
  const opts = { fields };

  try {
    const parser = new Parser(opts);
    const csv = parser.parse(tweets);

    fs.writeFileSync('./tweets.csv', csv);
    console.log('Successfully saved tweets to tweets.csv');
  } catch (err) {
    console.error(err);
  }
}

client.get('search/tweets', { q: SEARCH_QUERY }, (error, tweets, response) => {
```

```

if (error) {
  console.error(error);
} else {
  console.log(`Found ${tweets.statuses.length} tweets matching the search query`);
  writeTweetsToCSV(tweets.statuses);
}
});

```

2. Налаштування залежностей для Телеграм боту

```

{
  "name": "telegram-bot",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "node index.js",
    "dev": "nodemon index.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "dependencies": {
    "node-telegram-bot-api": "^0.61.0",
    "fs": "^0.0.1-security",
    "twitter": "^1.7.1",
    "text-miner": "^2.0.0"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "nodemon": "^2.0.20"
  }
}

```

3. Код серверної частини що опрацьовує команди користувача

```

const TelegramBot = require('node-telegram-bot-api');
const { parse } = require('twitter-parser');
const TextMiner = require('text-miner');

```

```

const formatTweets = require('tweets-formater');

const model = {};

const token = 'TELEGRAM_TOKEN';

const bot = new TelegramBot(token, { polling: true });

const SET_KEYWORDS_MESSAGE = 'Налаштувати слова для пошуку';
const GET_STATISTIC_MESSAGE = 'Отримати статистику';

bot.onText(/\/start/, (msg) => {
  bot.sendMessage(msg.chat.id, "Welcome", {
    "reply_markup": {
      "keyboard": [[
        GET_STATISTIC_MESSAGE,
        SET_KEYWORDS_MESSAGE
      ]]
    }
  });
});

function updateChat (chatId, values) {
  model[chatId] = {
    ...(model[chatId] ?? {}),
    ...values
  }

  return model
}

function setKeywords (chatId, words) {
  const keywords = words.split(',');

  updateChat(chatId, { keywords });
}

```

```

    bot.sendMessage(chatId, `Дякую, ключові слова збережено: "${words}".`);
  }

function setLastCommand (chatId, command) {
  model[chatId] = {
    ...(model[chatId] ?? {}),
    lastCommand: command
  }
}

function getCurrentChatModel (chatId) {
  return model[chatId];
}

async function getStatistic (chatId, currentModel) {
  try {
    if (currentModel?.keywords) {
      bot.sendMessage(
        chatId,
        `Зачекайте... Відбувається збір та аналіз інформації за такими ключовими
словами ${currentModel?.keywords}.`
      );

      const tweets = parse(currentModel?.keywords);
      const analyzer = new TextMiner.Corpora(formatTweets(tweets));

      const response = analyzer.sort({ method: 'tonality' });
      const { positive, negative, neutral } = response;

      const count = tweets.length;
      const positivePercent = positive.length * 100 / count;
      const neutralPercent = neutral.length * 100 / count;
      const negativePercent = negative.length * 100 / count;
    }
  }
}

```

```

    bot.sendMessage(
      chatId,
      `Звіт реакції людей стосовно: ${currentModel?.keywords} \n \n` +
      `Проаналізовано твітів: ${count} \n \n` +
      `Позитивно: ${positivePercent}% \n ` +
      `Нейтрально: ${neutralPercent}% \n ` +
      `Негативно: ${negativePercent}% \n `
    );
  } else {
    bot.sendMessage(
      chatId,
      `Будь ласка, спочатку налаштуйте ключові слова.`
    );
  }
} catch (e) {
  bot.sendMessage(
    chatId,
    `Вибачте, сталася помилка! Спробуйте повторити ваш запит пізніше.`
  );
}
}

bot.on('message', (message) => {
  try {
    const chatId = message.chat.id;
    const text = message.text;
    const currentModel = getCurrentChatModel(chatId)

    setLastCommand(chatId, text);

    switch (text) {
      case SET_KEYWORDS_MESSAGE: {
        bot.sendMessage(chatId, 'Будь ласка, введіть слова по яким буде здійснюватися
пошук (через ",").');
        break;

```

```

    }
    case GET_STATISTIC_MESSAGE: {
        getStatistic(chatId, currentModel)
        break;
    }
}

switch (currentModel?.lastCommand) {
    case SET_KEYWORDS_MESSAGE: {
        setKeywords(chatId, text)
        break;
    }
}
} catch (e) {
    console.log(e)
}
});

```