НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри
_____ Тарасенко В.П.
(підпис) (ініціали, прізвище)
"___" червня 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напряму підготовки ***6.050102 «Комп'ютерна інженерія»***
на тему: Програмні засоби сентимент аналізу повідомлень в мережі Інтернет

Виконав: студент IV курсу, групи КВ-53
Бондур Ярослав Андрійович
(підпис)

Керівник_____
_____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М.

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____
_____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки 6.050102 «Комп’ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Тарасенко В.П.
(підпис) (ініціали, прізвище)
«___» червня 2019 р.

ЗАВДАННЯ
на дипломний проект студенту

Бондур Ярослав Андрійович

1. Тема проекту «Програмні засоби аналізу звукових сигналів з використанням нейронних мереж»,
   керівник роботи Замятін Денис Станіславович, канд. тех. наук, доцент
   затверджені наказом по університету від «22» травня 2019 р. №1330-С
2. Термін подання студентом роботи: “___”_____2019 р.
3. Вихідні дані до роботи: див. Технічне завдання.
4. Зміст пояснювальної записки: див. Пояснювальна записка
5. Перелік графічного матеріалу:
   - Презентація
   - Схема алгоритму розробленої програми.
   - Структурна схема класів для виділення ознак.
   - Структурна схема класів класифікаторів.
   - Схема алгоритму нормалізації повідомлення.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., доц.,к.т.н.		

7. Дата видачі завдання: “___”_____2019 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів
1.	Вивчення літератури за тематикою роботи	15.04.2019
2.	Розроблення та узгодження технічного завдання	30.04.2019
3.	Аналіз існуючих рішень	05.05.2019
4.	Підготовка матеріалів першого розділу дипломної роботи	10.05.2019
5.	Підготовка матеріалів другого розділу дипломної роботи	18.05.2019
6.	Підготовка графічної частини дипломної роботи	20.05.2019
7.	Оформлення документації дипломної роботи	25.05.2019
8.	Попередній огляд матеріалів диплому на кафедрі	30.05.2019

Студент _____ _____
(підпис) (ініціали, прізвище)

Керівник проекту _____ _____
(підпис) (ініціали, прізвище)

АНОТАЦІЯ

Об’єкт розробки – програмні засоби сентимент аналізу повідомлень в мережі Інтернет, які дозволяють дослідити та проаналізувати текст повідомлення на емоційну забарвленість і емоційну оцінку думок по відношенню до об’єктів, мова про які йде в тексті.

Програмні засоби дозволяють: виконувати сентимент аналіз повідомлень з соціальної мережі Twitter, за заданою темою користувачем, отримати статистику результатів аналізу та переглядати повідомлення, що були проаналізовані.

В ході розробки:

- проведено аналіз методів сентимент аналізу тексту
- розроблений метод нормалізації повідомлень, що підвищує якість сентимент аналізу
- розроблені засоби взаємодії з Twitter API
- розроблено користувацький інтерфейс

Використання цих програмних засобів полегшить користувачам пошук думок інших користувачів з приводу будь-яких об’єктів та тем обговорення.

Ключові слова:

СЕНТИМЕНТ АНАЛІЗ, АНАЛІЗ ТОНАЛЬНОСТІ, ТВІТТЕР, TWITTER.

SUMMARY

The object of development - software for sentimental analysis of messages on the Internet, which allows you to explore and analyze the text of the message on the emotional assessment of opinions in relation to objects.

The software allows: to perform a sentimental analysis of messages from the social network Twitter, according to the user's given topic, get the statistics of the analysis results and view the messages that were analyzed.

During development:

- has been analyzed the methods of sentiment analysis
- has been developed method for normalizing messages is developed that improves the quality of the analysis sentiment
- has been developed tools for interacting with the Twitter API
- has been developed an user interface

Using these software tools will make it easier for users to find the opinions of other users about any objects and topics discussed.

Keywords:

SENTIMENT ANALYSIS, TWITTER,

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
			Документація загальна			
			Новорозроблена			
	А4	ІАЛЦ. 045490.002 ТЗ	Програмні засоби	4		
			сентимент аналізу			
			повідомлень в мережі			
			Інтернет			
			Технічне завдання			
	А4	ІАЛЦ. 045490.003 ТП	Програмні засоби	2		
			сентимент аналізу			
			повідомлень в мережі			
			Інтернет			
			Відомість технічного			
			проекту			
	А4	ІАЛЦ. 045490.004 ПЗ	Програмні засоби	51		
			сентимент аналізу			
			повідомлень в мережі			
			Інтернет			
			Пояснювальна записка			
	А4	ІАЛЦ. 045490.005 Д1	Програмні засоби	1		
			сентимент аналізу			
			повідомлень в мережі			

Зм	Лист	№ докум.	Підп	Дата	ІАЛЦ. 045490.001 ОА		
Розроб.		Бондур Я.А.			Програмні засоби сентимент аналізу повідомлень в мережі Інтернет *Опис альбому*	Літ.	Лист 1 / Листів 2
Перев.		Замятін Д.С.					
Н. контр.		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФПМ, КВ-52	
Затв.		Тарасенко В.П.					

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
			Інтернет			
			Алгоритм розробленої			
			програми			
			Схема алгоритму			
	А4	ІАЛЦ. 045490.006 Д1	Програмні засоби	1		
			сентимент аналізу			
			повідомлень в мережі			
			Інтернет			
			Діаграма класів виділення			
			ознак			
			Схема структурна			
	А4	ІАЛЦ. 045490.007 Д1	Програмні засоби	1		
			сентимент аналізу			
			повідомлень в мережі			
			Інтернет			
			Діаграма класів			
			класифікаторів			
			Схема структурна			
	А4	ІАЛЦ. 045490.008 Е3	Програмні засоби	1		
			сентимент аналізу			
			повідомлень в мережі			
			Інтернет			
			Алгоритм нормалізації			
			повідомлення			
			Схема алгоритму			

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ 2

2. ПІДСТАВА ДЛЯ РОЗРОБКИ 2

3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ 2

4. ДЖЕРЕЛА РОЗРОБКИ 2

5. ТЕХНІЧНІ ВИМОГИ 2

5.1. Вимоги до програмного продукту, що розробляється 2

5.2. Вимоги до апаратного забезпечення 3

5.3. Вимоги до програмного та апаратного забезпечення користувача 3

6. ЕТАПИ РОЗРОБКИ 4

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Сервіс аналізу тональності тексту».

Галузь застосування: підприємства, що потребують аналіз тональності тексту для підвищення якості продукту.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи ступеня «бакалавр комп'ютерної інженерії», затверджене кафедрою системного програмування і спеціалізованих комп’ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення сервісу аналізу тональності тексту за заданою конфігурацією.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- сумісність з будь якою операційною системою, що має будь-який браузер для перегляду веб-сторінок;
- можливість аналізу тональності тексту англійською мовою;
- можливість гнучкого налаштування запиту для аналізу тональності;

- можливість гнучкого розширення API сервісу;
- можливість компіляції та запуску веб-сервісу;
- Наявність користувацького інтерфейсу.

5.2. Вимоги до апаратного забезпечення

- Процесор: 2-х ядерний, Intel, AMD;
- Оперативна пам’ять: 2 Гб;
- Наявність доступу до мережі Internet.

5.3. Вимоги до програмного та апаратного забезпечення користувача

- Операційна система Windows, Unix-подібні системи;
- Встанослений браузер для перегляду веб-сторінок;
- Наявність доступу до мережі Internet.

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту	15.04.2019
2.	Розроблення та узгодження технічного завдання	30.04.2019
3.	Аналіз існуючих рішень	05.05.2019
4.	Підготовка матеріалів першого розділу дипломного проекту	15.05.2019
5.	Підготовка матеріалів другого розділу дипломного проекту	20.05.2019
6.	Підготовка графічної частини дипломного проекту	25.05.2019
7.	Оформлення документації дипломного проекту	27.05.2019
8.	Попередній огляд матеріалів диплому на кафедрі	30.05.2019

№ п/п	Формат	Позначення	Найменування	Кількість листів	Примітка
			Документація загальна		
			Новорозроблена		
1	А4	ІАЛЦ.045490.004 ПЗ	Програмні засоби сентимент	51	
			аналізу повідомлень в мережі		
			Інтернет		
			Пояснювальна записка		
			проекту		
2	А4	ІАЛЦ.045490.005 Д1	Програмні засоби сентимент	1	
			аналізу повідомлень в мережі		
			Інтернет		
			Алгоритм розробленої програми		
			Схема алгоритму		
3	А4	ІАЛЦ.045490.006 Д2	Програмні засоби сентимент	1	
			аналізу повідомлень в мережі		
			Інтернет		
			Діаграма класів виділення		
			ознак		
			Схема структурна		
4	А4	ІАЛЦ.045490.007 Д3	Програмні засоби сентимент	1	
			аналізу повідомлень в мережі		
			Інтернет		
			Діаграма класів класифікаторів		
			Схема структурна		
5	А4	ІАЛЦ.045490.008 Д4	Програмні засоби сентимент	1	
			аналізу повідомлень в мережі		
			Інтернет		
			Алгоритм нормалізації		
			повідомлення		
			Схема алгоритму		

Зм.	Арк	№ докум	Підпис	Дата
Розроб.		Бондур Я.А.		
Перевір.		Замятін Д.С.		
Н. контр.		Клятченко Я.М.		
Затв.		Тарасенко В.П.		

ІАЛЦ.045490.003 ВП

Програмні засоби сентимент аналізу повідомлень в мережі Інтернет.
Відомість проекту

Літ.	*Арк.*	*Аркушів*
	1	1

«КПІ ім. Ігоря Сікорського», ФПМ, КВ-53

№ п/п	Формат	Позначення	Найменування	Кількість листів	Примітка
			Документація загальна		
			Новорозроблена		
1	А4	ІАЛЦ.045490.004 ПЗ	Програмні засоби сентимент	50	
			аналізу повідомлень в мережі		
			Інтернет		
			Пояснювальна записка		
			проекту		
2	А4	ІАЛЦ.045490.005 Д1	Програмні засоби сентимент	1	
			аналізу повідомлень в мережі		
			Інтернет		
			Алгоритм розробленої програми		
			Схема алгоритму		
3	А4	ІАЛЦ.045490.006 Д2	Програмні засоби сентимент	1	
			аналізу повідомлень в мережі		
			Інтернет		
			Діаграма класів виділення		
			ознак		
			Схема структурна		
4	А4	ІАЛЦ.045490.007 Д3	Програмні засоби сентимент	1	
			аналізу повідомлень в мережі		
			Інтернет		
			Діаграма класів класифікаторів		
			Схема структурна		
5	А4	ІАЛЦ.045490.008 Д4	Програмні засоби сентимент	1	
			аналізу повідомлень в мережі		
			Інтернет		
			Алгоритм нормалізації		
			повідомлення		
			Схема алгоритму		

Зм.	Арк	№ докум	Підпис	Дата	***ІАЛЦ.045490.003 ВП***		
Розроб.		Бондур Я.А.			Програмні засоби моделювання апаратного кешування даних.	*Літ.*	*Арк.* 1 / *Аркушів* 1
Перевір.		Замятін Д.С.					
Н. контр.		Клятченко Я.М.				«КПІ ім. Ігоря Сікорського», ФПМ, КВ-53	
Затв.		Тарасенко В.П.					

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ 4
ВСТУП 5
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ 7
1.1 Конкретизація задачі 7
1.2 Аналіз існуючих інструментів сентимент 9
1.3 Обґрунтування теми дипломного проекту 13
1.4 Обґрунтування вибору середовища розробки 14
2. ОСОБЛИВОСТІ ЗАДАЧІ ДЛЯ ДАНИХ З МІКРОБЛОГІВ 16
2.1 Основні характеристики данних 16
2.2 Особливості текстів 17
2.2.1 Обробка графічних символів 17
2.2.2 Обробка метаданих 19
2.2.3 Скорочення, подовження приголосних і пунктуація 21
2.2.4 Аналіз графів слів 22
2.2.5 Використання онтології 22
2.2.6 Розширення моделі темами 23
2.3 Використання особливостей текстів для попередньої обробки 23
3. МОДЕЛЬ КЛАСИФІКАТОРА 25
3.1 Аналіз основних методів сентимент аназізу тесту 25
3.1.1 Наївний байєсівський класифікатор 25
3.1.2 Класифікація методом опорних векторів 25
3.1.3 Метод максимальної ентропії 26
3.1.4 Порівняння методів на данних з Twitter 27

3.2 Перехід до байєсівського рішення 30
3.2.1 Опис класифікатора 30
3.2.2 Навчання і передбачення 31
3.2.3 Проблеми рішення 32
3.2.4 Перехід до байєсівського рішення 33
3.2.5 Використання n-грам для вимірювання ознак 35
4. РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ 38
4.1 Онтології для заміни невідомих слів 38
4.2 Алгоритми підготовки даних, передбачення і навчання 39
4.3 Кількісна оцінка методу 40
4.4 Програмна реалізація 42
4.4.1 Структура пакетів 42
4.4.2 Пакет My_Classifiers 43
4.4.3 Веб-додаток 46
ВИСНОВОК 49
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 50

ДОДАТКИ

Додаток 1. Копії графічних матеріалів

ІАЛЦ.045490.005 Д1. Алгоритм розробленої програми. Схема алгоритму.

ІАЛЦ.045490.006 Д2. Діаграма класів виділення ознак. Структурна схема.

ІАЛЦ.045490.007 Д3. Діаграма класів класифікаторів. Структурна схема.

ІАЛЦ.045490.008 Д4. Алгоритм нормалізації повідомлення. Схема алгоритму.

Додаток 2. Презентація

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ASCII - American Standard Code for Information Interchange

CSS - Cascading Style Sheets

PPM – Prediction by Partial Matching

SO – Semantic orientation

SVM – Support vector machine

ВСТУП

Аналіз тональності - один із напрямів галузі обробки текстів на природних мовах. Задачу можна визначити як обчислювальне виявлення суб'єктивності в текстах і відношення авторів цих текстів до деяких об'єктів.

Думки інших людей впливали на наш процес прийняття рішень ще до поширення інтернету. Однак, якщо раніше було можливим дізнатися думку лише у обмеженого числа знайомих, то в останнє десятиліття, у зв'язку з ростом популярності мережі інтернет, все більшого значення набувають відгуки, залишені користувачами в інтернет-магазинах, блогах, соціальних мережах, а також спеціалізованих ресурсах.

Згідно з опитуванням, проведеним компанією Dimensional Research [1], 88% опитаних вважають, що читання позитивних або негативних відгуків в інтернеті впливає на їх рішення при купівлі товарів. Також відгуки про якість обслуговування безпосередньо впливають на продажу: продажу інтернет-магазинів з рейтингом "5 зірок" на "Hotline" на 55% більше магазинів з рейтингом "4 зірки".

Все частіше користувачі залишають відгуки не на спеціалізованих сайтах відгуків, а в соціальних мережах. Відгуки в соціальних мережах залишає на 10% більше число покупців, ніж на спеціалізованих сайтах-агрегаторах [1]. Так як соціальні мережі містять не тільки відгуки, а обсяг повідомлень занадто великий для ручної обробки, актуальною є задача автоматичного пошуку і класифікації відгуків.

"Твіттер" (Twitter) - одна з найпопулярніших соціальних мереж. Число активних користувачів перевищує 200 мільйонів, і вони залишають більше 400 мільйонів повідомлень в день [2]. Обмеження на довжину повідомлення (140 символів), великий набір використовуваної лексики,

сленгу і граматичних помилок роблять автоматичний пошук і аналіз думок нетривіальним завданням.

Таким чином, постає завдання розмістити відповідно до емоційних забарвленнь безліч твітів, що мають відношення до конкретного об'єкту, заданим словом або словосполученням, тобто знайдених по пошуковому запиту, з використанням особливостей саме цієї соціальної мережі. Таке завдання вирішують і для великих текстів за допомогою лінгвістичного словникового підходу і обчислювально, методами машинного навчання. Мета даної роботи - дослідити проблему для Twitter і запропонувати варіант її вирішення з використанням апарату імовірнісних моделей. Для досягнення цієї мети можна сформулювати наступні кроки:

- проаналізувати особливості завдання для мікроблогів;
- порівняти базові методи навчання з учителем для даних з мікроблогів і вибрати кращий за параметрами точності, повноти результатів і часу навчання;
- запропонувати, обґрунтувати та реалізувати новий метод на основі обраного;
- оцінити результати роботи нового методу.
- Розробити програмні засоби для практичного використання автоматичного розпізнавання тональності в соціальній мережі Twitter

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1 Конкретизація задачі

Завдання аналізу емоційного забарвлення текстів зводиться до задачі класифікації. У нашому випадку є набір твітів, кожен з яких потрібно віднести до однієї з трьох категорій: позитивні, нейтральні або негативні.

Іноді класифікація відбувається в два етапи і на обох етапах є бінарної. На першому відокремлюються суб'єктивні повідомлення від об'єктивних. Об'єктивними в цьому випадку називаються якраз ті, які не несуть емоційного забарвлення і є нейтральними в варіанті з трьома класами. Другий етап ділить суб'єктивні тексти на позитивні і негативні. У випадку з Twitter, де майже всі повідомлення суб'єктивні, а критерії нейтральності можна сформулювати тільки в сенсі «не позитивно» і «не негативно», будемо для простоти розглядати поділ на два класи.

Твіт - це рядок, що складається з не більше ніж 280 символів. Він може містити спеціальні слова, що починаються з певних знаків: відразу після «@» пишеться ім'я користувача, з яким пов'язано повідомлення або до якої воно звернено, а після «#» знаходиться так званий хештег - слово, яке явно вказує на зв'язок твіта з об'єктом, який цим словом позначається. Всі твіти створюються користувачами, тому можуть містити помилки, скорочення, особливу пунктуацію і інші способи вираження думки в короткому тексті. У кожного повідомлення в Twitter є час, коли воно опубліковано, і автор. Якщо один твіт є відповіддю на інший, то у першого є посилання на другий, тобто на «батьківський». Ретвіти також містять дані про початкове розміщення. Поставлена задача вирішується обчислювально, за допомогою технік машинного навчання. Перша згадка завдання аналізу тональності

відноситься до 2002 року. Тоді були розглянуті стандартні рішення методом навчання без вчителя [3] і методом навчання з учителем [4]. В обох статтях досліджувалися відгуки на спеціалізованому ресурсі: метою була задача визначення чи рекомендує користувач, який залишив відгук, те, про що він написав.

Таблиця 1 – Приклад роботи системи аналізу тональності

Повідомлення	**Тональність, визначена аналізатором**	**Реальна тональність повідомлення**
Мій рідний Київ - місто сили, волі, нескореності та свободи українського духу! З 1537 Днем Народження мій Київ	Позитивна	Позитивна
Київська дитяча залізниця після свого оновлення розпочала 66-й сезон перевезень	Нейтральна	Нейтральна
31 травня у трьох кінотеатрах столиці стартує безкоштовний кінофестиваль для дітей і підлітків «Чілдрен Кінофест»	Позитивна	Позитивна
Кожна п’ята школа Печерського району перейшла на сучасне мультипрофільне харчування	Нейтральна	Нейтральна
Київ - місто парканів. А хотілось би - парків.	Нейтральна	Негативна

1.2 Аналіз існуючих інструментів сентимент аналізу повідомлень

Для аналізу тональності тесту на сьогодні є багато програм та бібліотек, основні з яких є:

- SentiStrenght;
- Sentiment Analyzer;
- Компонент аналізу тональності тексту в складі систем «Аналітичний кур'єр» і «X-files»;
- Компонент аналізу тональності в складі системи RCO Fact Extractor;

SentiStrength – система, розроблена M. Thelwall, K. Buckley, G. Paltoglou і D. Cai. Спочатку, дана система була розроблена для аналізу коротких неструктурованих неформальних текстів англійською мовою. Однак, вона може бути налаштована для роботи з текстами на ряді інших мов, в тому числі і для текстів українською мовою [5]. Приклад користувацького інтерфейсу представлений на рис. 1.

Рисунок 1 – Користувацький інтерфейс системи SentiStrenght

Результат видається у вигляді двох оцінок - оцінка позитивної складової тексту (за шкалою від +1 до +5) і оцінка негативної складової (за шкалою від -1 до -5). Крім того, існує можливість надання оцінок в іншому вигляді:

- Бінарна оцінка (позитивний / негативний текст) ;
- Тернарна оцінка (позитивний / негативний / нейтральний);
- Оцінка за єдиною шкалою від -4 до +4;

Алгоритм заснований на пошуку максимального значення тональності в тексті для кожної шкали (тобто пошук слова з максимальною негативною оцінкою і слова з максимальною позитивною оцінкою). При роботі алгоритму враховується найпростіша взаємодія слів (наприклад, слова-підсилювачі підсилюють значення тональності для слова, на яке вони діють - «дуже злий» матиме більш негативну оцінку, ніж просто «злий») і ідіоматичні вирази [6].

Переваги:

- Підтримка багатьох мов;
- Підтримується розробниками;
- Можливість регулювання видачі результату аналізу;
- Безкоштовна програма;

Недоліки:

- Реалізований алгоритм не враховує специфіку мови тексту;

Sentiment Analyzer – це безкоштовний сервіс, що дозволяє провести аналіз тональності тексту, написаного англійською мовою. Показники настроїв коливаються від -100 до +100, де -100 вказує на негативний або серйозний тон, а +100 - на позитивний або захоплений тон.

Система була натренована колекцією більше чим 8000 зразків тексту, і найкраще працює з американською англійською мовою після 1990 року [7]. Приклад користувацького інтерфейсу системи представлений на рис. 2.

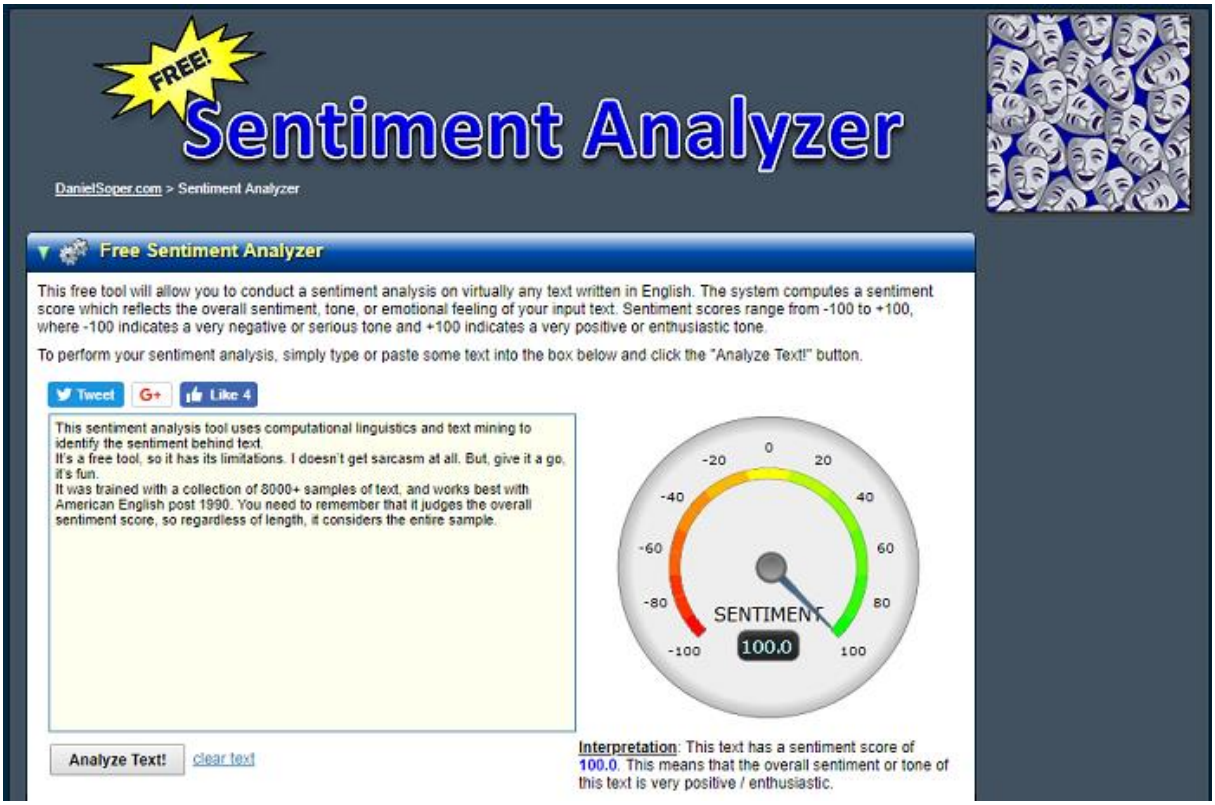


Рисунок 2 – Користувацький інтерфейс системи Sentiment Analyzer

Переваги:

- Безкоштовна програма;

Недоліки:

- Реалізований алгоритм не враховує специфіку мови тексту;
- Не підтримується розробниками;
- Закритий вихідний код;

Компонент аналізу тональності тексту в складі систем «Аналітичний кур'єр» і «X-files» – розроблений компанією «Ай-Теко». Компонент визначення тональності тексту реалізує метод, заснований на словниках і правилах [8].

Дана система видає користувачеві масив розмічених речень. У реченнях розмічаються об'єкти тональності (при наявності таких) і ланцюжок слів,

що несе в собі тональність по відношенню до них. Крім того, на підставі знайдених ланцюжків слів підраховується загальна тональність для кожного речення. Для підрахунку загальної тональності використовується ряд спеціальних правил. Наприклад (для пропозиції «Лікар Сміт вилікував хворого на грип»), є правило, яке говорить, що поєднання позитивного дієслова «вилікувати» з негативним ланцюжком (в даному випадок «хворий на грип») приписує позитив словам, що підлягає дієслову (в нашому прикладі - «лікар Сміт»). Тональність оцінюється по тернарній шкалі (позитивний / негативний / нейтральний). [8]

Система працює в декілька етапів:

- Попередня обробка тексту, виділення і класифікація знайдених слів;
- Об'єднання знайдених слів в пов'язані один з одним ланцюжки;
- Виділення об'єктів тональності;

Переваги:

- Підтримується розробниками;
- Простий у використанні;

Недоліки:

- Платна програма з закритим вихідним кодом;
- Відсутність кількісної оцінки тексту

Компонент аналізу тональності в складі системи RCO Fact Extractor – система, розроблена компанією RCO. Для аналізу тональності тексту система використовує підхід, заснований на правилах. Дана система враховує синтаксичну структуру тексту і взаємодію різних типів слів [9].

Робота компонента відбувається в п'ять етапів:

- Розпізнавання всіх згадувань про об'єкт у всіх формах, включаючи повні, короткі і інші форми згадок;

- Відсів і переклад текстових конструкцій, в яких відображаються всі події та ознаки, пов'язані з цільовим об'єктом;
- Виділення і класифікація тих позицій, в яких явно виражається тональність, і тих пропозицій, які описують емоційні ситуації;
- Для кожної пропозиції ухвалення рішення про тональності «позитив-негатив» з урахуванням тих місць, які займають в її змісті емоційні, тональні і нейтральні слова, засоби вираження заперечення;
- Оцінка загальної тональності тексту на основі тональностей всіх речень, що входять в нього;

Для своєї роботи компонент використовує модулі синтаксичного аналізу тексту і ототожнення найменувань, розроблені також в компанії RCO [9].

Переваги:

- Підтримується розробниками;
- Простий у використанні;
- Зручний інтерфейс;

Недоліки:

- Платна програма з закритим вихідним кодом;
- Відсутність кількісної оцінки тексту

1.3 Обґрунтування теми дипломного проекту

Темою дипломної роботи було обрано створення програмних засобів сентимент аналізу повідомлень в мережі інтернет з такими перевагами:

- Безкоштовний продукт з відкритим кодом;
- Зручний та простий інтерфейс;
- Наявність кількісної оцінки тексту
- Сентимент аналіз повідомлень, що орієнтований на дані з мікроблогів

Дані засоби можуть використовуватись як самостійний інструмент для аналізу думок людей, що користуються соціальною мережею Twitter, з приводу сутностей заданими користувачем.

1.4 Обґрунтування вибору середовища розробки

Для створення засобів аналізу тональності в мережу Інтернет було обрано мову Python, з використанням Django - високорівневий відкритий Python-фреймворк (програмний каркас) для розробки веб-систем. а для інтерфейсу користувача CSS фреймворк Materialize - це орієнтована на CSS і Javascript середовище, засноване на матеріалах Google Material Design. Він повністю заснований на керівних принципах Google Material Design і служить в якості шаблону для більш ефективного і зручного використання дизайну матеріалів. У ній є деякі визначені плагіни, такі як Scrollspy, Scrollfire, lightbox, paralax та інші. Переваги мови Python:

- Інтерпретатор Python реалізований практично на всіх платформах та операційних системах;
- Наявність великої кількості модулів, що підключаються до програми, які забезпечують різноманітні додаткові можливості;
- Простота та гнучкість мови;

Такий підхід дозволить наростити можливості програмних засобів в майбутньому.

Висновки

В цьому розділі розібрані існуючі рішення сентимент аналізу повідомлень в мережі Інтернет, їх переваги та недоліки і базуючись на цьому розроблена модель програмних засобів, які представлені в даній

дипломній роботі. Вони повинні увібрати в себе всі переваги існуючих рішень та відкинути їх недоліки.

2. ОСОБЛИВОСТІ ЗАДАЧІ ДЛЯ ДАНИХ З МІКРОБЛОГІВ

2.1 Основні характеристики даних

Мікроблоги - це, в першу чергу, сервіси для спрощення публікації і сприйняття призначених для користувача даних. Зазвичай повідомлення в мікроблогах складаються з одного або декількох речень, а для Twitter є суворе обмеження на довжину повідомлення - 280 символів. У 280 символів користувачам платформи необхідно вмістити контекст, своє ставлення до теми і, можливо, посилання на фотографію, Інтернет-ресурс або інший медіа об'єкт. Часто контекст відновлюється з навколишнього світу, тобто користувач пише про те, що хвилює Інтернет в цей момент, і люди, володіючи цією інформацією, зіставляють висловлювання з реальними подіями. У комп'ютера так просто бути в курсі обговорюваних тем не виходить, тому на відновлення контексту розраховувати не доводиться.

Платформи для ведення мікроблогів також є соціальними мережами, де користувачі можуть взаємодіяти один з одним. У Twitter, наприклад, крім соціальних графів можна спостерігати графи, в які вибудовуються самі повідомлення: користувачі можуть відповідати на твіти, а також розміщувати у себе твіти інших користувачів, в термінології платформи це називається «ретвітів». Інформація про соціальні взаємодіях може уточнювати результати класифікації, наприклад, є інтуїтивне припущення, що відповідь на негативно забарвлене повідомлення теж потрапить в клас негативних.

2.2 Особливості текстів

2.2.1 Обробка графічних символів

Для вираження емоцій в тексті користувачі ставлять смайли. Смайл - це набір символів, який ілюструє вираз обличчя автора, а точніше його настрій. Всі смайли можна поділити на східні та західні по географії їх використання, останні приведені в табл. 2 з мітками, які відповідають їхньому емоційному забарвленню. У випадку з короткими текстами немає більш простого способу виразити своє ставлення до теми, ніж поставити смайл, але не всі користувачі так роблять, тому розмічати повідомлення з їх допомогою в загальному випадку не вийде. Приклади твітів де використуються смайли для більшого вираження емоційного забарвлення показані на рис. 3, 4 та 5. Є й більш складні конструкції з дужок, двокрапок і інших символів, але вони використовуються не так часто і зазвичай означають вже не просто відношення, а якісь дії або об'єкти, тобто емоційного забарвлення не несуть.

Рисунок 3 – Твіт з використанням позитивних смайлів, що показує його емоційне забарвлення

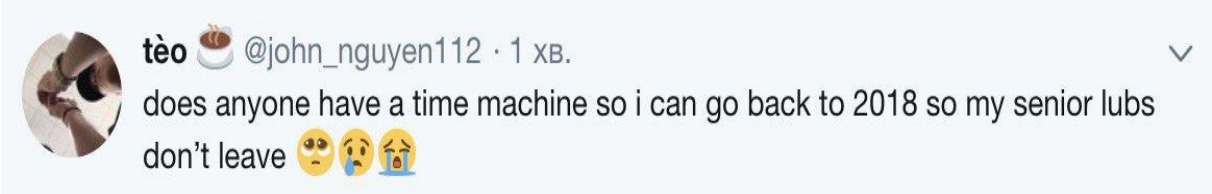


Рисунок 4 – Твіт з використанням сумних смайлів

Рисунок 5 – Твіт з використанням стурбованих смайлів, що показує його емоційне забарвлення

Таблиця 2 – Емоційне забарвлення смайлів

Смайл	Мітка	Смайл	Мітка	Смайл	Мітка	Смайл	Мітка
:-)	+	:)	+	:o)	+	:]	+
:c)	+	:>	+	=)	+	8)	+
:}	+	:^)	+	:>)	+	xD	+
8-D	+	8D	+	>:[	-	:D	+
XD	+	=-D	+	D:	-	:(	-
B^D	+	:-))	+	v.v	-	D;	-
:-c	-	:c	-	:-/	-	;]	+
:-[	-	:[	-	=\	-	:/	-
:@	-	>:(	-	:|	-	<3	+

Крім ASCII смайлів є ще й графічні - це картинки, які вставляються в текст. У сучасних веб-сервісах і мобільних додатках використовується графічна мова Emoji для запису слів, емоцій і дій. На рис. 6 зображені деякі відомі графічні смайли, які використовуються в соціальній мережі Facebook. Зазвичай для кожного з них є ASCII аналог, причому не один. Набираючи повідомлення на клавіатурі комп'ютера або ноутбука, зручніше поставити двокрапку з дужкою, але смартфони та планшети надають всі зручності для вставки усміхнених картинок: поряд з російською та англійською клавіатурою, наприклад, на них можна підключити і клавіатуру графічного мови Emoji.

Так як смайли є свого роду розміткою повідомлень самими користувачами, їх необхідно використовувати при аналізі емоційного забарвлення. У цій роботі буде розглянуто застосування символьних і графічних посмішок для збору корпусу твітів і для попередньої обробки даних безпосередньо перед класифікацією.

Рисунок 6 – Деякі графічні смайли, що використовуються в Facebook

2.2.2 Обробка метаданих

Ще одна особливість спілкування в мікроблогах - хештеги. Користувач позначає в своєму повідомленні слово, ставлячи перед ним «#», тим самим показуючи зв'язок об'єкта, що позначається цим словом, і всього повідомлення. Платформи для мікроблогів пропонують можливість шукати за хештегом, вибирати з них популярні і стежити за потоками актуальної інформації. Багато хештегів використовується протягом короткого періоду часу, але потім саме по ним можна знайти інформацію, яка колись була актуальною і знадобилася через кілька місяців. Наприклад, організатори заходів намагаються придумувати унікальний хештег, розміщувати його на інформаційних стендах, щоб учасники стежили за

твітами один одного і поширювали інформацію по всьому Інтернету. Можна сказати, що це оповідна функція хештегів, точніше, тих з них, які вказують на об'єкт, вони можуть допомогти здійснювати пошук повідомлень на певну тему.

Іншу функцію цих спеціальних слів-асоціацій можна назвати описовою. Саме такі хештеги можна використовувати у визначенні емоційного забарвлення текстів. В роботі [10] запропонований спосіб классіфіцікаціі хештегів по їх емоційному забарвленню. Автори пропонують читачам подивитися на 20 найпопулярніших хештегів з кожної групи. Для наочності в табл. 3 наведені перші три для кожної емоції.

Використання хештегів безпосередньо для оцінки емоційного забарвлення можна вважати приблизно таким самим, як і у смайлів, але лише тоді, коли слово однозначно відноситься або до позитивних, або до негативних. В іншому випадку вони або стають звичайними словами: без символу «#» і беруть участь в класифікації нарівні з іншими, або уточнюють ймовірність повідомлення потрапити в той чи інший клас за допомогою підрахунку умовних ймовірностей, де умовою і є хештег.

Таблиця 3 – Найпопулярніші хештеги для п'яти почуттів: прихильності, гніву, страху, подяки і смутку

Прихильність	**Гнів**	**Страх**	**Подяка**	**Смуток**
#yourthebest	#godie	#wimp	#thanking	#catlady
#hyc	#donttalktome	#freakedout	#thankful	#buttrue
#alwaysandforever	#getoutofmylife	#sinistet	#thankyou	#lonelytweet

2.2.3 Скорочення, подовження голосних і пунктуація

Тексти в мікроблогах містять не тільки уточнюючу інформацію, а й частково зайву. Її потрібно навчитися використовувати, так як специфіка повідомлень не дозволяє хоч щось викидати.

Обмеження в 280 символів змушує людей скорочувати слова, причому як за допомогою загальновідомих абревіатур, наприклад, «КПІ» - це Київський Політехнічний Інститут, так і за допомогою жаргонних конструкцій: «h8» - це насправді hate. На прикладі останнього видно, що позбутися від цього слова було б марнотратно, але навряд чи «h8» внесло б внесок в імовірність повідомлення потрапити в клас негативних, як і слово «hate». Отже, скорочення потрібно вміти розшифровувати.

Коли користувачам здається, що довжина повідомлення не така вже й маленька, вони використовують подовження голосних - ще один спосіб висловлювати стурбованість темою твіта. Автор примножує голосну в слові, зображуючи її тривале звучання, наприклад, крик. Так «nooooooo» буде, швидше за все, означати категоричну незгоду, а «so cuuuute» - зворушення. Таким чином, кожне таке слово щось означає, але класифікатор може про це не знати, значить, потрібно розповідати класифікатору іншими способами, що це важливе слово і яке з відомих є його менш емоційним аналогом.

Авторська пунктуація може розповісти про емоційне забарвлення повідомлення не менше, ніж смайли. Наприклад, в нейтральних твітах вкрай рідко зустрічаються знаки оклику. Однак, однозначно класифікують особливості пунктуації не так сильно: наявність знаків оклику вказує на наявність емоційного забарвлення, при цьому не можна без додаткового аналізу сказати, яке саме; поєднання «?!», швидше за все, буде означати здивування, тобто класифікується як негативне; крапки зазвичай говорять про нейтральність.

2.2.4 Аналіз графів слів

Ідея, запропонована в статті [11], ґрунтується на побудові графа для отримання інформації про класи. Для позитивних і негативних слів будуються два графа відповідно. Їх структури відновлюються з навчальної вибірки. Для присвоювання мітки новому прикладу пропонується використовувати ще і словник синонімів: ймовірність слова опинитися в класі враховує кількість влучень його самого і всіх його синонімів в цей клас.

2.2.5 Використання онтологій

У статті [12] пропонується для кожної конкретної теми будувати онтології (схеми областей знань), які уточнюють запити, звужуючи всі знайдені в пошуку твіти до тих, в яких дійсно мова йде про цей об'єкт. Тема запиту замінюється на пару «корінь онтології» і «властивість», наприклад, якщо вихідний запит - «смартфон», то це і є тема онтології, а властивостями можуть бути «android», «iphone» і «акумулятор». У цьому випадку замість однієї спроби пошуку буде вже три, але з більш релевантною видачею: «смартфон android», «смартфон iphone» і «смартфон акумулятор». За допомогою комерційної програми з закритим вихідним кодом OpenDover14 авторами статті проводиться подальший аналіз тональності отриманих результатів видачі.

Автори статті [13] пропонують використовувати онтології в момент підготовки знайдених даних до розмітки. Автори пропонують три варіанти: ставити категорію з попереднього рівня поруч зі словом в повідомленні, для якого ця категорія знайдена; замінювати слово на більш загальну категорію; розглядати в прикладах розподіл слів як умовний розподіл від категорій. Стаття описує останній спосіб застосування

онтологій. Ймовірність влучення прикладу в клас обчислюється таким чином: враховуються не тільки розподіл слів, а й додаткові ознаки. Точніше, від величин додаткових ознак залежить розподіл слів в повідомленні. Запропонований метод є уточненням наївного байєсівського класифікатора за допомогою онтологій.

2.2.6 Розширення моделі темами

Інший погляд на уточнення моделі описаний в статті [14]. Тут пропонується розширювати стандартну модель наївного байєсівського класифікатора умовним розподілом слів, залежних від теми. Перед навчанням моделі вважається, що є якась кількість тем, за якими потрібно розділити слова з навчальної вибірки. Наприклад, ймовірність зустріти слово «Шекспір» у тій же темі, що і «Пушкін», більше, ніж в тій же темі, що і «телефон».

2.3 Використання особливостей текстів для попередньої обробки

Смайли, хештеги, скорочення, подовження голосних і пунктуація - це те, про що класифікатор ще не знає, тобто перед подачею йому повідомлення необхідно перетворити це повідомлення так, щоб усі ці особливості не вибивалися і перетворилися в звичайні слова.

Смайли, перераховані в табл. 4, замінюються в тексті на відповідну їм мітку. Це робиться для того, щоб в навчальній вибірці слово «+» зустрілося більше разів серед позитивних твітів, тим самим, ймовірність потрапити в клас позитивних «+» дасть більший внесок, ніж просто «:)». Так само, заміною на «+» і «-», обробляється пунктуація.

До смайлів, заміненними на мітки, додається заміна деяких

однозначних хештегів, що класифікуються. Відбувається це з тієї ж причини, що і зі смайлами. Якщо заміна хештегу не відбулася, то вважається, що він повинен стати звичайним словом, тобто «#» з початку видаляється спеціальною функцією, і далі робота відбувається вже без урахування того, що це хештег.

Якщо в повідомленні зустрічається невідоме слово, його варто перевірити на наявність в словнику скорочень. У даній роботі використовується словник «No slang», до якого програма звертається під час підготовки даних до подачі класифікатором. Запит до словника відбувається в онлайн-режимі, і для обробки скорочень потрібно підключення до мережі Інтернет.

Повторення голосних прибирати зовсім не потрібно: достатньо скоротити кількість повторюваних голосних до двох, тобто «nooooooo» заміниться на «noo». У цьому випадку слово «noo» може зустрітися в навчальній вибірці, на відміну від «nooooooo», де саме сім, а не вісім або дев'ять букв «o». Таким чином слово «no» вже не те ж саме, що «noo», але всі, скільки завгодно довгі продовження голосних «o» зведуться до одного і того ж емоційного «noo», яке дасть кожному з таких слів з продовженнями однаковий привід потрапити в клас «-».

3. МОДЕЛЬ КЛАСИФІКАТОРА

3.1 Аналіз основних методів сентимент аналізу тексту

3.1.1 Наївний байєсівський класифікатор

Наївний байєсівський класифікатор [15] працює з умовними ймовірностями, наївно припускаючи, що слова в реченні незалежні. Цей простий класифікатор добре показує себе у вирішенні задачі класифікації текстів [16]. Спершу необхідно обрати закон, за яким розподілені дані будуть спрогнозовані. Потім по розмічених прикладах обчислюються параметри цього розподілу, які в подальшому використовуються для розмітки. Припустимо, що дані розподілені за законом Бернуллі. В такому випадку клас c*, До якого належить невідоме повідомлення t, обчислюється за формулою:

$$c^* = \arg\max_c \frac{P(c)\sum_{i=1}^{m} P(x \mid c)^{x_i(t)}}{P(t)} \quad (1)$$

Тут x - це характеристики, за якими оцінюються повідомлення, і всього їх m, а $x_i(t)$ - величини, які показують, як i-та характеристика представлена в повідомленні t, c - мітка класу з множини всіх міток C. $P(c)$ і $P(x \mid c)$ - параметри моделі, знайдені при навчанні класифікатора.

3.1.2 Класифікація методом опорних векторів

Метод опорних векторів – SVM [17] працює за принципом поділу простору на підпростори, відповідні класам. Тут теж вибираються ознаки, за якими вимірюються приклади і відповідно до вимірів перетворюються в числові вектори. Далі робота йде вже з цими векторами і простором, в якому вони розташовуються. На етапі навчання, завдання методу - перетворити простір за допомогою оператора ядра так, щоб знайшлися

такі гіперплощини, які поділяють приклади з різних класів навчальної вибірки. Прогноз відбувається згідно з тим, в яку частину простору щодо знайдених гіперплощин потрапляє вектор, відповідний новому прикладу.

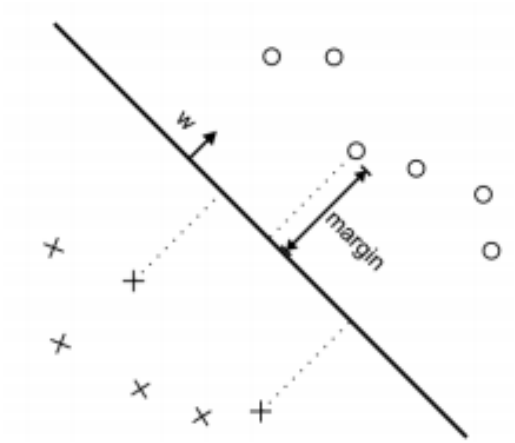


Рисунок 7 – Двійкова класифікація SVM. Margin - відстань від гіперплозщини до кожного з класів. W - вектор нового прикладу

Ілюстрація поділу на два класи за допомогою лінійного ядра зображена на рис. 1. Тут показано, як будується рівновіддалені від обох множин гіперплощини і як новий вектор потрапляє в одну з них в залежності від розташування щодо цієї гіперплощини.

3.1.3 Метод максимальної ентропії

Наступною розглянемо класифікацію за допомогою методу максимальної ентропії [18]. У випадку з розбивкою на два класи це використання логістичної регресії для пошуку розподілу даних по класах. На відміну від наївного байєсівського класифікатора цей метод не передбачає незалежності ознак. Це означає, що можна використовувати для передбачення ознаки різної природи, наприклад, вимірювати n-грами і словосполучення в повідомленні одночасно. Суть цього методу в тому,

що треба вибрати найбільш підходящу модель, що задовольняє всім природним обмеженням. Модель описується настурною формулою

$$P(c \mid t, \lambda) = \frac{\exp\left(\sum_{i=1}^{N} \lambda_i x_i(c,t)\right)}{\sum_{c \in \beta} \exp\left(\sum_{i=1}^{N} \lambda_i x_i(c,t)\right)} \quad (2)$$

Тут c - мітка класу, t – повідомлення. що розглядається, x_i (c, t) - спільна представленість i-ої ознаки в класі c і в прикладі t, N - кількість ознак, λ - вектор ваг для всіх ознак: чим більше вага, тим більше значимість цієї ознаки для класифікатора. На етапі навчання за допомогою методів оптимізації обчислюється саме вектор ваг ознак. При прогнозі класу для нового прикладу знову потрібно знайти таке c^* з множини міток, що розглянута величина P $(c \mid t, \lambda)$ буде максимальною.

3.1.4 Порівняння роботи методів на даних з Twitter

Порівняння проводиться на даних, зібраних з Twitter самостійно. Тут ми беремо текст в сирому вигляді, без додаткової обробки, і передаємо алгоритму. У табл. 1, 2 і 3 представлені результати роботи наївного байєсівського класифікатора, класифікатора на основі методу опорних векторів і класифікатора за принципом максимальної ентропії відповідно. Алгоритми навчалися на 1000000 розмічених прикладів і передбачали результати для 400 нових.

Щоб оцінити якість класифікації, зазвичай використовують $F1$-score - гармонійне середнє двох інших: precision (точність) і recall (повнота). Мовою ймовірностей можна визначити ці величини наступним чином: precision - це ймовірність того, що випадково обраний твіт потрапив в той клас, якому він належить насправді; recall - це ймовірність того, що

випадково обраний твіт з класу при класифікації в нього і потрапить. Покажемо, що означають ці величини більш формально. Нехай зафіксований клас c і є множиною всіх твітів $\mathcal{T}$, що класифікуються, яке ділиться на дві множини: $\mathcal{T}c$ - ті, що насправді відносяться до класу c, і $\mathcal{T} \setminus \mathcal{T}c$ - ті, у яких повинні стояти інші позначки. За результатами експерименту визначаються наступні величини:

$T\ P$ - кількість твітів з $\mathcal{T}$, яким алгоритм поставив мітку c;

$F\ P$ - кількість твітів з $\mathcal{T} \setminus \mathcal{T}c$, яким алгоритм поставив мітку c;

$T\ N$ - кількість твітів з $\mathcal{T} \setminus \mathcal{T}c$, яким алгоритм поставив точно не c;

$F\ N$ - кількість твітів з $\mathcal{T}c$, яким алгоритм поставив точно не c.

Для перевірки роботи методів використовуються реалізації з бібліотеки Scikit-learn [19] для мови Python. Для наївного байєсівського класифікатора передбачається, що дані розподілені за законом Бернуллі. В якості характеристик беруться всі слова, які зустрілися в навчальній вибірці, і кожен твіт перетворюється в вектор з цілих чисел, де на місці i-ого слова ставиться 0, якщо слово зустрілося в повідомленні, і 1, якщо ні.

Таблиця 4 – Класифікація наївним байєсівським класифікатором. Час навчання - 1 секунда

Мітка класу	**Precision**	**Recall**	**F_1-score**	**Кількість**
-1.0	0.82	0.75	0.78	204
1.0	0.74	0.82	0.78	196
avg/total	0.79	0.78	0.78	400

Таблиця 5 – Класифікація методом опоних векторів. Час навчання - 750 секунд

Мітка класу	**Precision**	**Recall**	**F_1-score**	**Кількість**
-1.0	0.86	0.73	0.79	204
1.0	0.74	0.87	0.80	196
avg/total	0.80	0.80	0.79	400

Таблиця 6 – Класифікація методом максимальної ентропії. Час навчання - 437 секунд

Мітка класу	**Precision**	**Recall**	**F_1-score**	**Кількість**
-1.0	0.82	0.75	0.78	204
1.0	0.74	0.82	0.78	196
avg/total	0.79	0.78	0.78	400

Як вже було сказано, всього в тестовій вибірці було 400 повідомлень, з яких 196 були помічені «+», а 204 - «-». З табл. 4, 5 і 6 видно, що всі методи показали приблизно однакову точність і повноту роботи, але час навчання при цьому у наївного байєсівського класифікатора відрізняється на порядок від двох інших. Для постановки задачі, коли прогнозування відбувається для даних з мікроблогів, час навчання при рівних показниках $F1$ - score є вирішальним, так як змінюються теми, про які пишуть в Інтернеті, а значить змінюється лексика і способи вираження ставлення до них - класифікатору потрібно підлаштовуватися під ці обставини, постійно перенавчатися.

Як підсумок порівняння, за основу варто взяти наївний байєсівський класифікатор. Варто сказати, що для моделювання даних можна було без

додаткових зусиль вибрати і закон поліноміального розподілу. У випадку з твітами це означає, що в векторі, в який цей твіт перекладається, кожному слову з навчальної вибірки зіставляється кількість разів, скільки воно зустрілося в повідомленні. Коли тексти короткі, у випадку з поліноміальною моделлю вектори в більшості результів з 0 і 1, тому окремо його можна не розглядати.

3.2 Перехід до байєсівського рішеня

3.2.1 Опис класифікатора

Спершу розглянемо простий випадок перетворення твітів в числовий вектор. Для цього побудуємо словник на основі навчальних даних: кожне слово - це окрема ознака, яка вимірюється в конкретному повідомленні. Нехай в тренувальній вибірці було D унікальних слів, тоді числовий вектор, відповідний твіту, виглядає наступним чином: $x = \{x1, ..., xD\}$. Тут кожна компонента x_j показує, як представлено j-те слово в повідомленні x. Будемо вважати, що дані розподілені за законом Бернуллі, тоді представленість слова визначається його наявністю або відсутністю, тобто $x_j = 1$, якщо j-те слово зустрілося в повідомленні, і 0 в іншому випадку. Завдання - зіставити кожному такому вектору найбільш подібну мітку класу, яку будемо позначати y. Множину всіх класів позначимо $\mathcal{C}$. Виходить, що для кожного класу $c \in \mathcal{C}$ необхідно порахувати функцію подібності p (x | $y = c$). Наївний байєсівський класифікатор дійсно наївний в тому сенсі, що він передбачає незалежність ознак. Це дозволяє вважати шукану ймовірність, як множину ймовірностей для кожної ознаки:

$$p(x \mid y = c, \theta) = \prod_{j=1}^{D} p(x_j \mid y = c, \theta_{jc}) \quad (3)$$

Тут θ – це матриця-параметр розподілу Бернуллі, який шукається на

етапі навчання класифікатора, тобто за відомими парами x, y. Насправді елемент цієї матриці θjc – це ймовірність того, що j-те слово зустрінеться в прикладах з класу c.

3.2.2 Навчання і передбачення

На етапі навчання класифікатора потрібно знайти, як уже було сказано, параметри розподілу, що моделює дані, і апріорні ймовірності класів π. Спершу позначимо тренувальні дані $\mathcal{D}$, тоді $x_i \in \mathcal{D}$ – це приклад з навчальної вибірки, а x_{i1}, …, x_iD – кількісні характеристики ознак для прикладу x, і y_i – мітка класу для нього.

Подивимося на i-ий приклад, тоді обчислення ймовірності пари x_i, y_i за умови параметрів θ відбувається за формулою:

$$p(x_i, y_i \mid \theta) = p(y_i \mid \pi)\prod_{j=1}^{D} p(x_{ij} \mid \theta_j) = \prod_{c\in\beta} \pi_c^{(y_i=c)} \prod_{j=1}^{D}\prod_{c\in\beta} p(x_{ij} \mid \theta_{jc})^{(y_i=c)} \qquad (4)$$

Переходячи до всього набору даних виходить, що потрібно максимізувати наступну величину

$$p(D \mid \theta) = \prod_{c\in\beta} \pi_c \prod_{j=1}^{D}\prod_{c\in\beta}\prod_{i:y_i=c} p(x_{ij} \mid \theta_{jc}) \qquad (5)$$

Тепер необхідно знайти відповідні цьому максимуму параметри π і θij. Виконавши всі необхідні обчислення, отримаємо, що

$$\hat{\pi}_c = \frac{N_c}{N} \qquad (6)$$

$$\hat{\theta}_{jc} = \frac{N_{jc}}{N_c} \qquad (7)$$

В цих формулах N - кількість прикладів в навчальній вибірці, Nc- кількість елементів в класі c, Njc - кількість разів, скільки j-те слово

зустрілося в прикладах з класу c. Навчання моделі відбувається за $O(ND)$ кроків.

Для передбачення класу y для нового прикладу x необхідно максимізувати такий вираз по всіх класах $c \in C$

$$\hat{\pi}_c \prod_{j=1}^{D} \left(\hat{\theta}_{jc}\right)^{(x_j=1)} (1-\hat{\theta}_{jc})^{(x_j=0)} \quad (8)$$

Клас, для якого він вийшов максимальним, і вважається найбільш правдоподібним для даного прикладу.

3.2.3 Опис задачі з нульвою ймовірністю для байєсівського класифікатора

Основна проблема описаного підходу полягає в тому, що не обов'язково всі слова (ознаки) зустрінуться у всіх класах. Наприклад, якщо j-те слово не представлено в класі c, то $Njc = 0$ і ймовірність того, що повідомлення яке розглядається потрапить в клас c теж стане нульовою. Від цієї проблеми може позбавити перехід до байєсівського підходу, коли параметри стають випадковими величинами зі своїми розподілами. Ще для усунення цієї проблеми використовується згладжування Лапласа [20], коли і до чисельника, і до знаменника додається по 1. Це окремий випадок байєсівського підходу.

Інша проблема, яку необхідно вирішувати, - це неможливість розширення словника: якщо слова не було в навчальній вибірці, то класифікатор знову стикається з нульовими можливостями, але тепер для всіх класів.

3.2.4 Вирішення задачі нульової ймовірності для байєсівського класифікатора

Щоб позбутися від проблеми з нульовими можливостями «чесним» способом, перейдемо до наївного байєсівського класифікатора. Наївний байєсівський класифікатор – простий ймовірнісний класифікатор, що базується на використані теореми Баєса зі строгими (наївними) припущеннями про незалежність одного з класів. В цьому випадку параметри замість точкових величин представляються випадковими величинами з апріорними розподілами. У моделі класифікатора параметри - це ймовірності, тобто їх значення обмежені на відрізку [0, 1]. Так розподіл для кожного з параметрів має задавати випадкову величину, обмежену на відрізку. Для моделювання параметра π будемо використовувати розподіл Діріхле Dir (α0), який встановлює закон розподілу багатовимірної випадкової величини, де всі компоненти - величини з відрізка [0, 1] і підсумовуються в 1. Вважаємо також, що кожен з параметрів θjc має Бета-розподіл на відрізку [0, 1] Beta (β0, β1). Всі параметри вважаються незалежними, тому завжди апріорна ймовірність параметрів обчислюється за формулою:

$$p(\theta) = p(\pi)\prod_{j=1}^{D}\prod_{c\in\beta} p(\theta_{jc}) \qquad (9)$$

Завдання тепер отримати апостеріорну ймовірність на підставі даних $\mathcal{D}$ з навчальної вибірки і з'ясувати, чи потрапить вона в той же клас, що і завжди апріорна, тобто вираз для неї матиме вигляд Dir (...) ·Beta (...) · ... ·Beta (...). По теоремі Байеса шукану ймовірність можна порахувати за наступною формулою

$$p(\theta \mid D) = \frac{p(\theta)\cdot p(D \mid \theta)}{p(D)} \qquad (10)$$

Знаменник виразу не залежить від параметрів, тому розглянемо окремо

чисельник. Якщо розписати щільності розподілів і згрупувати отриманий множник, результатом буде скорочена версія яка наведена нижче. M тут деяка постійна.

$$p(\theta)\cdot p(D\mid\theta)=M\cdot\prod_{c\in\beta}\pi_c^{a_{0c}-1}\cdot\prod_{j=1}^{D}\prod_{c\in\beta}\theta_{jc}^{\beta_0}\left(1-\theta_{jc}\right)^{\beta_1}\cdot\prod_{x\in\vartheta}\prod_{c\in\beta}\pi_c\prod_{j=1}^{D}\theta_{jc}^{x_{jc}} \tag{11}$$

$$=M\cdot\prod_{c\in\vartheta}\pi_c^{a_{0c}+N_c}\cdot\prod_{c\in\vartheta}\prod_{j=1}^{D}\theta_{jc}^{\beta_0+N_c-N_{jc}}(1-\theta_{jc})^{\beta_1+N_{jc}} \tag{12}$$

У виразі записано не що інше, як добуток щільності розподілу Діріхле і Бета-розподілів. Так ймовірність p (θ | $\mathcal{D}$) залишилася в тому ж класі, що і p (θ).

$$p(\pi\mid D)=Dir(N_1+a_{01},\dots,N_c+a_{0c},\dots) \tag{13}$$

$$p(\theta_{jc}\mid D)=Beta(N_c-N_{jc}+\beta_0,N_{jc}+\beta_1) \tag{14}$$

Позначимо отримані параметри як α_0^*, β_0^* і β_1^* відповідно.

Залишається зрозуміти, як за допомогою отриманих даних можна передбачити клас для нових прикладів. Розглянемо приклад x. Метою є знаходження для нього значення мітки y. Для цього потрібно максимізувати p ($y = c$ | x, $\mathcal{D}$) за всіма $c \in \mathcal{C}$.

$$p(y=c\mid x,D)=\frac{p(y=c\mid D)\cdot p(x\mid y=c,D)}{p(x\mid D)} \tag{15}$$

Знову будемо дивитися тільки на чисельник. Перший його множник насправді є математичним очікуванням πc по π зі щільністю розподілу Діріхле. Коротка версія виведення приведена нижче, в (18), (19) і (20). Тут для перетворення використовується техніка маргіналізації, а перехід до (20) випливає з визначення математичного очікування.

$$p(y=c\mid D)=\int_{\pi\in\Pi}p(y=c,\pi\mid D)d\pi \tag{16}$$

$$= \int_{\pi \in \Pi} p(\pi \mid D)\, p(y = c, \pi \mid D) d\pi \tag{17}$$

$$= \mathop{\mathrm{E}}_{\pi \sim Dir(a_0^*)} \pi_c \tag{18}$$

Аналогічно першому розбираємо другий множник. Знову скористаємося технікою маргіналізації для запису (19). Тепер розкриваємо ймовірність в добуток ймовірностей - так можна робити, тому що величини x_i і θ_{*c} незалежні, і отримуємо формулу (21). Далі переписуємо інтеграл в позначеннях математичного очікування і, згідно попередніх результатів, приходимо до того, що завдяки незалежності параметрів отримали добуток математичних очікувань випадкових величин θ_{jc} зі щільністю Бета – розподілення.

$$p(x \mid y = c, D) = \int_{\theta \in \Theta} p(x, \theta_{*c} \mid y = c, D) d\theta_{*c} \tag{19}$$

$$= \int_{\theta \in \Theta} p(x \mid y = c, D, \theta_{*c})\, p(\theta_{*c} \mid y = c, D) d\theta_{*c} \tag{20}$$

$$= \mathop{\mathrm{E}}_{p(\theta_{*c} \mid y=c, D)} p(x \mid y = c, D, \theta_{*C}) \tag{21}$$

$$= \prod_{j=1}^{D} \mathop{\mathrm{E}}_{\theta_{*C} \sim Beta(\beta_0^*, \beta_1^*)} \theta_{jc}^{x_{ij}} \tag{22}$$

Так як параметри обчислені на етапі навчання, а формула математичних очікувань для відповідних розподілів відома, залишається тільки підставити їх в вираз (15), максимум якого ми шукаємо.

3.2.5 Використання n-грам для вимірювання ознак

Альтернативний погляд на підрахунок ймовірностей повідомлень - це розбиття повідомлення на n-грами, тобто n-літерні сполучення. Така зміна дозволить штучно послабити вимогу про незалежність ознак. Припустимо,

що вихідне повідомлення розбивається на триграми. Три тут обрано, так як обчислювальна кількість всіх можливих триграм ще не така велика, але вже інформативно перевизначає x_i, тепер вони представляють триграми. Нехай триграм в повідомленні всього L, тоді вектор x = $x1$, ..., xL описує представленість триграм в повідомленні. Імовірність того, що повідомлення потрапить в клас c, в цьому випадку підраховується за такою формулою

$$p(x, y = c) = \prod_{j=2}^{L} p(x_j \mid x_{j-1}, y = c) \qquad (23)$$

Модель класифікатора трохи зміниться, а новими параметрами стануть якраз p (x_j | $x_{j\text{-}1}$, y = c), які і буде шукати алгоритм на етапі навчання.

Перейдемо до попередніх позначок і з'ясуємо, як зміняться формули підрахунку параметрів і шуканої величини. Вектор x знову показує представленість кожної триграми з навчальної вибірки в розглянутому прикладі: 1 стоїть на місці присутньої триграми і 0 на місці відсутньої. Введемо також функцію, яка допоможе розуміти послідовність триграм в прикладі: n (x_j) показує, яка за рахунком зустрілася j-та триграмма в прикладі. Якщо не зустрілася, то нехай значення цієї функції буде -∞. Для спрощення вважаємо, що кожна триграмма зустрілася тільки один раз. У формулі перерахунку параметра π (13) нічого не зміниться, так як на апріорні ймовірності класу зміна не вплине. Нехай D - це тепер кількість усіх триграм, які зустрілися в навчальній вибірці. Додатково позначимо $Njkc$ - кількість пар з триграми з номером k і наступної прямо за нею триграми з номером j в класі c. Так як θjkc - ймовірність зустріти триграму з номером k відразу за триграмою з номером j в класі c, тоді перерахунок параметра θjkc буде здійснюватися за формулою, аналогічною (14), але з урахуванням зв'язку триграм.

$$p(\theta_{jkc} \mid D) = Beta(N_c - N_{jkc} + \beta_0, N_{jkc} + \beta_1) \quad (24)$$

Нові параметри знову позначимо β_0^* і β_1^* для зручності. У формулі (15) знову будемо дивитися тільки на чисельник. Перший множник не зміниться, а другий буде обчислюватися так

$$p(x \mid y = c, \vartheta) = \prod_{j=1}^{D} \prod_{k=1}^{D} \mathop{\mathrm{E}}_{\theta_{**c} \sim Beta(\beta_0^*, \beta_1^*)} \theta_{jkc}^{(n(x_j)=n(x_k)+1)} \quad (25)$$

Причому в результаті кожна θjkc представлена не більше, ніж один раз, так як в прикладі за триграмою може слідувати тільки одна інша.

4. РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ

4.1 Алгоритми підготовки даних, передбачення і навчання

Спосіб змістовного усунення проблеми з невідомими словами в аналізованих прикладах, запропонований в роботі, - це використання онтологій. Онтологією називається деяка схема області знань, зазвичай вона являє собою спрямований ациклічний граф, вершини якого - поняття. Чим вище в графі знаходиться вершина, тим ширше відповідне їй поняття. Таким чином, якщо в повідомленні зустрілося слово, яке невідоме класифікатору, його можна замінити на більш загальне поняття відповідно до онтології.

База знань Вікіпедії складається користувачами і на даний момент містить 4515000 статей тільки англійською мовою. Структура організації статей у Вікіпедії схожа на те, що ми шукаємо, - це граф категорій. На основі категорій Вікіпедії вже складена онтологія проектом DBpedia [21] - вона і використовується в роботі.

Дані викачуються в форматі Turtle. У нас є два файли: перший - відображення статей в підмножині категорій, другий - ліс всіх категорій. Граф категорій незв'язний, більшість з них не беруть участь ні в якій з ієрархій. Згідно з цими особливостями потрібно вирішити задачу зберігання онтології і пошуку потрібної категорії.

В першу чергу, варто сказати, що зберігається відображення категорій в номери вершин в графі, так як зберігати цілі рядки для такої кількості даних неекономно. Ці відображення кладуться в бор, в сусідньому борі запам'ятовуються відображення статей в категорії. Коли з'являється невідоме слово, шукається стаття, де шукане слово - префікс, так як назви статей складаються не завжди з одного слова. Вибираємо з усіх категорій, до яких стосується ця стаття, ту, що в ієрархії знаходиться нижче всього. У

разі рівного розподілу рівня вибираємо будь-яку.

4.2 Алгоритми підготовки даних, передбачення і навчання

Описані в ході даної роботи ідеї були об'єднані в метод класифікації і реалізовані на мові Python. Деякі моменти не були роз'яснені раніше, тому їм приділяється трохи більше уваги.

Окремо розглянемо три частини: підготовка даних, навчання класифікатора, передачення класу.

Підготовка данних:

- заміна HTML сутностей на слово «URL», згадок користувача на «USER» і чисел на «42»;
- форматування повідомлення в нижній регістр;
- заміна смайлів, хештегів і деяких розділових знаків на «+» або «-» (розділ 2.3);
- заміна довгого повторення голосних на поєднання з двох букв (розділ 2.3);
- розбиття на слова з урахуванням розділових знаків за допомогою Penn Treebank Tokenizer з NLTK [22];
- заміна скорочень на їх розшифровки (розділ 2.3);
- заміна невідомих класифікатором слів на узагальнення кожного з них;
- приведення всіх слів до початкової форми за допомогою алгоритму Snowball Stemmer [15].

Навчання класифікатора:

- підготовка даних і збереження множини відомих слів;

- перетворення отриманих рядків на пари триграм, що перекриваються, тепер одна пара триграм - це ознака, за якою вимірюються повідомлення;
- кожен приклад з навчальної вибірки перетворюється в числовий вектор: на місці відповідної пари триграм з набору ставиться 1, якщо вона є в прикладі, і 0, якщо немає;
- пошук апостеріорних розподілів параметрів відповідно за формулами (13) і (24).

Передбачення класу для нового прикладу:

- підготовка прикладу відповідно до описаного вище в «Підготовка даних»;
- перетворення отриманого рядка на пари триграм;
- перетворення отриманого набору в числовий вектор: на місці відповідної пари триграм з набору ставиться 1, якщо вона є в прикладі, і 0, якщо немає;
- обчислення ймовірності прикладу виявитися в кожному з класів («1» або «-1») відповідно до виразу (15) і обчисленню множників з нього по (18) і (25);
- вибір класу, який дав найбільшу ймовірність попадання в нього.

4.3 Кількісна оцінка методу

Твіттер надає API для вилучення та пошуку повідомлень, в тому числі по пошуковому запиту. Відповіддю на запит до Twitter є набір сутностей, кожна з яких зберігає інформацію про повідомленні: його id, текст, ім'я користувача, час публікації, а також, якщо воно є відповіддю або ретвітом

іншого, то вказується id «батьківського» твіту. У такому вигляді не відновити ланцюжка твітів: щоб знайти діалог між користувачами, потрібно обійти всі існуючі твіти та знайти з них ті, які посилаються на певні повідомлення, - так можна знайти всі відповіді на нього або його ретвіти. Швидше за все, якщо знайдений певний твіт про об'єкт, то відповіді на нього будуть про цей же об'єкт, тобто обчислювати відповіді необхідно.

Пропонується робити це таким чином. Нехай є твіт T, його id T_{id}, ім'я користувача, який його опублікував T_{user}, і час публікації T_{time}. Помітна особливість відповідей в Twitter, як говорилося раніше, - це згадки, тобто відповідь на твіт користувача з ім'ям username будуть починатися з рядка «@username». Тоді, щоб знайти всі відповіді на твіт T будемо шукати не по безлічі всіх можливих повідомлень, а за всіма повідомленнями, опублікованими пізніше T_{time} по пошуковому запиту «@ T_{user}». Серед них вже можна буде виділити повідомлення, які посилаються на твіт з номером T_{id} - це і будуть всі відповіді на T.

Так збираються дані по слову-запиту для розмітки емоційного забарвлення актуальних повідомлень по цій темі. Цей метод використовувався для розширення тестової вибірки, використовуваної для аналізу алгоритмів в ході всієї роботи.

Для навчання класифікатора використовувалися 1000000 розмічених повідомлень. Класифікатори порівнювалися на 400 тестових прикладах, 204 з яких негативні, 196 - позитивні.

Таблиця 7 – Класифікація наївним байєсівським класифікатором

Мітка класу	**Precision**	**Recall**	**F_1-score**	**Кількість**
-1.0	0.82	0.75	0.78	204
1.0	0.74	0.82	0.78	196
avg/total	0.79	0.78	0.78	400

Таблиця 8 – Класифікація методом опорних векторів з триграмами та онтологіями

Мітка класу	**Precision**	**Recall**	**F_1-score**	**Кількість**
-1.0	0.88	0.74	0.80	204
1.0	0.75	0.88	0.81	196
avg/total	0.82	0.81	0.81	400

У табл. 7 і 8 наведено порівняння базового класифікатора і його зміненої версії, заснованої на Байєсова підході і використанні триграм і онтологій. Новий класифікатор дав поліпшення передбачення на 3%, при цьому метод не втратив можливості навчатись інкрементально, тобто його можна уточнювати в онлайн-режимі.

4.4 Програмна реалізація

4.4.1 Структура пакетів

Весь проект реалізований з використанням мови програмування Python. Програмна реалізація складається з трьох частин:

1. Пакет My_Classifiers містить класи та функції для роботи з алгоритмами класифікації, для вибору ознак, перехресної перевірки, а також для роботи з корпусами даних.

2. Пакет Twitter_API_Wrapper містить клас-обгортку для доступу до Twitter API, використовує бібліотеку tweepy.
3. Веб-додаток Web_App служить для пошуку думок в соціальній мережі Twitter.

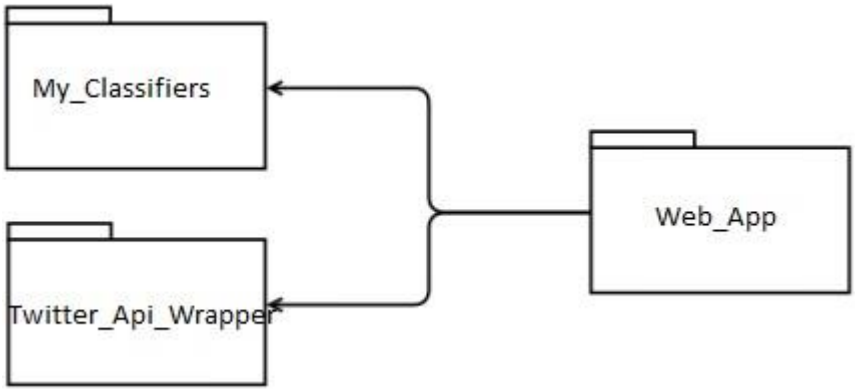


Рисунок 8 – Діаграма пакетів

4.4.2 Пакет My_Classifiers

Всі класифікатори успадковуються від базового класу ClassifierBase. Він має посилання на екземпляри підкласів PreprocessorBase (передобробники) і FeatureExtractorBase (виділення і вибірка ознак). Це зручно, так як для збереження класифікатора на диск (і для подальшого завантаження) досить серіалізовати один об'єкт (сериализація здійснюється за допомогою стандартної бібліотеки pickle).

У класі ClassifierBase визначені наступні методи:

learn(labels, documents) - процедура навчання. Приймає на вхід масив документів навчальної вибірки і масив відповідних документів класів (класами можуть бути як рядки, так і числа). Визначається в спадкоємців.

classify_one(document) - функція класифікації одного документа. Приймає на вхід документ (рядок), повертає клас, передбачений відповідно

до моделі. Повертає клас з максимальним значенням функції get_conditional_probability (document, class).

classify_batch(documents) - функція класифікації безлічі документів. Приймає на вхід масив рядків, повертає масив класів.

get_conditional_probability(document, class) - повертає ймовірність того, що документ належить класу відповідно до параметрів моделі.

get_encoded_labels(labels) - кодує класи навчальної вибірки в діапазон числами від 0 до n-1 (n - число уникальних класів).

У класі ClassifierBase визначені наступні поля:

preprocessor - передобробник, що нормалізує документ.

feature_extractor - визначник ознак, що перетворює рядок в вектор ознак.

class_index - асоціативний масив. Ключ - оригінальний клас, значення - індекс від 0 до n-1.

classes - масив унікальних класів.

Класифікатори працюють з векторним поданням документа. Підкласи класу FeatureExtractorBase перетворюють текстовий документ в вектор ознак. У ньому визначені наступні методи:

learn(documents, labels) - процедура навчання. Визначається в нащадках.

extract(document) - функція вилучення ознак. Приймає на вхід рядок, повертає вектор ознак.

get_feature_count() - повертає розмір вектора ознак.

Вибір ознак здійснюється за допомогою підкласу FeatureSelectorBase, який є декоратором FeatureExtractorBase. Таким чином, дуже просто додати або прибрати відбір ознак (класифікатор не знає, з чим він працює).

Витяг n-грам здійснюється за допомогою підкласу

NgrammFeatureExtractor. У конструкторі йому передається масив необхідних n (можна використовувати 1-грами, 2 грами і т.д. разом або окремо).

Підкласи NgrammFeatureExtractorBoolean і NgrammFeatureExtractorCount відрізняються способом визначення ваги у ознак.

Передобробники виконують нормалізацію тексту.

У всіх підкласах PreprocessorBase визначений метод preprocess, який отримує на вхід рядок, застосовує до неї нормалізує, і повертає нормалізований рядок.

Для можливості гнучкої зміни порядку нормалізації всі атомарні дії реалізовані окремими класами, які об'єднуються за допомогою компонувальника CombinedPreprocessor.

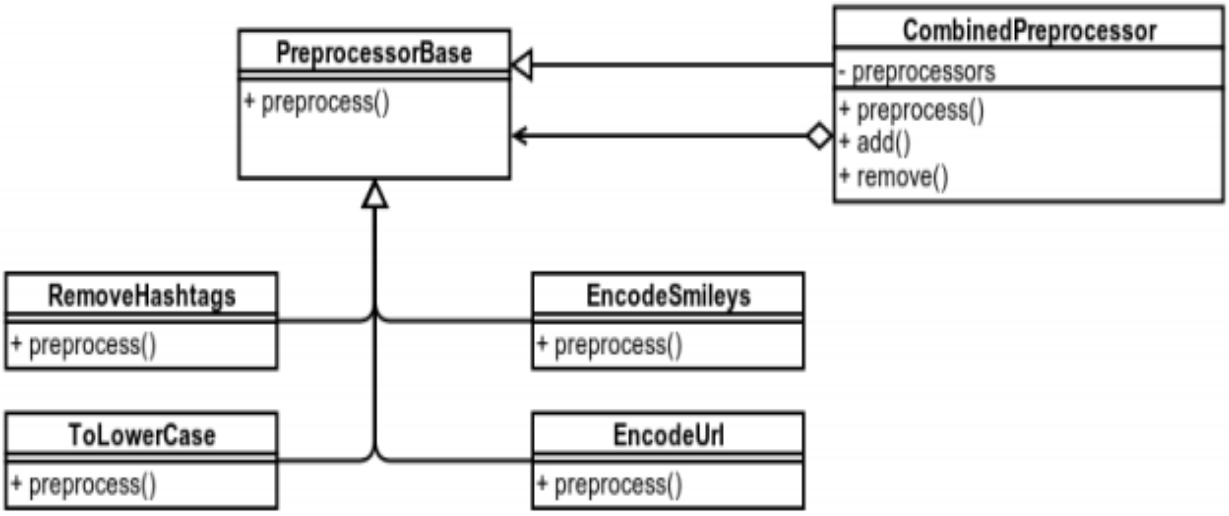


Рисунок 9 – Діаграма класів для передобробників

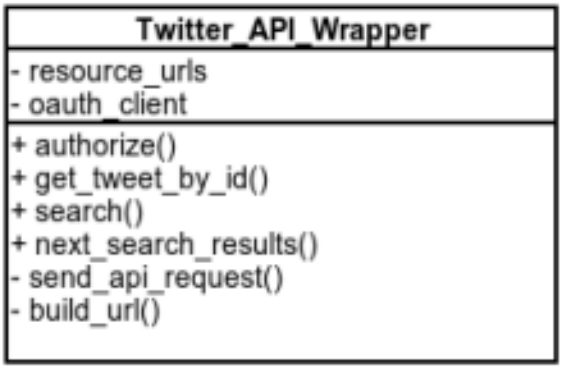


Рисунок 10 – Діаграма класів клієнта для Twitter

4.4.3 Веб-додаток

Веб-додаток було розроблено за допомогою веб-фреймворку Django, інтерфейс розроблений з допомогою css-фреймворка Materialize.

На головній сторінці користувач може ввести назву сутності, що цікавить його.

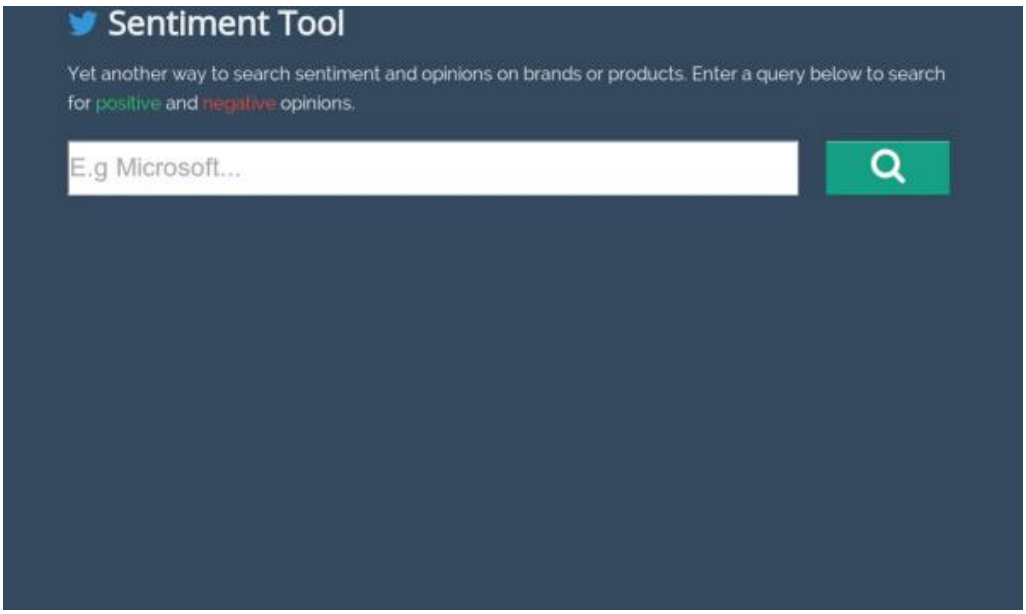


Рисунок 11 – Головна сторінка веб-додатку

Після того, як користувач натиснув на кнопку пошуку, відбувається ajax-запит до сервера. Сервер повертає вже класифіковані повідомлення в форматі JSON. За замовчуванням показуються останні 400 повідомлень. Повідомлення, що мають негативну тональність підсвічені червоним, позитивну - зеленим, нейтральну - темно-синім. Приклад на рис. 12.

Рисунок 12 – Вкладка повідомлень

На вкладці "Statistics", рис. 13, відображається кількість повідомлень кожної категорії, а також кругова діаграма для того, щоб швидко оцінити загальну думку про об'єкт пошуку.

На вкладці "Summary", рис. 14, відображаються дві хмари слів - позитивна (популярні слова, що вживають в позитивному контексті) і негативна Розмір слова пропорційний логарифму частоти вживання.

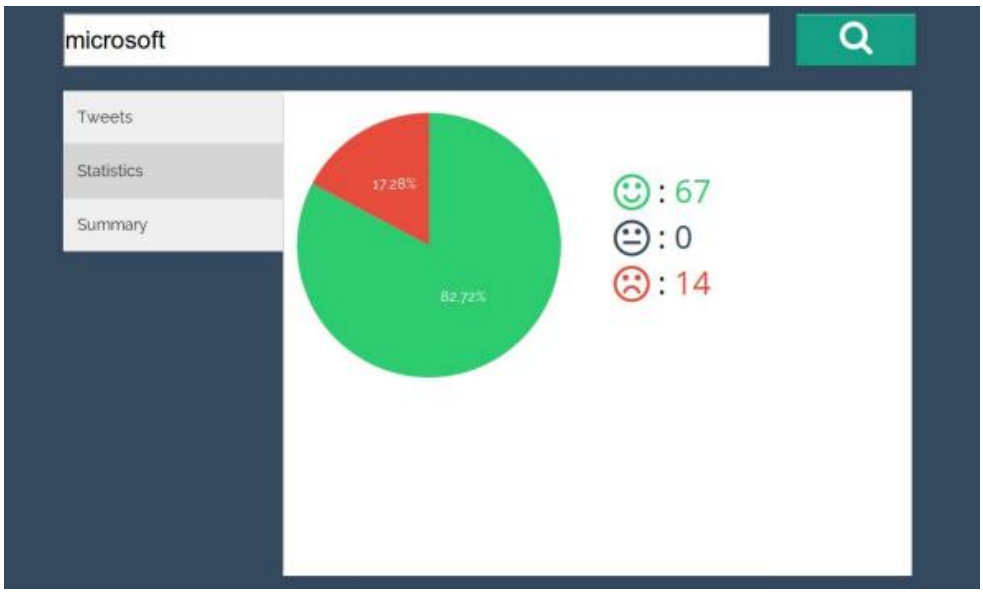


Рисунок 13 – Вкладка статистики

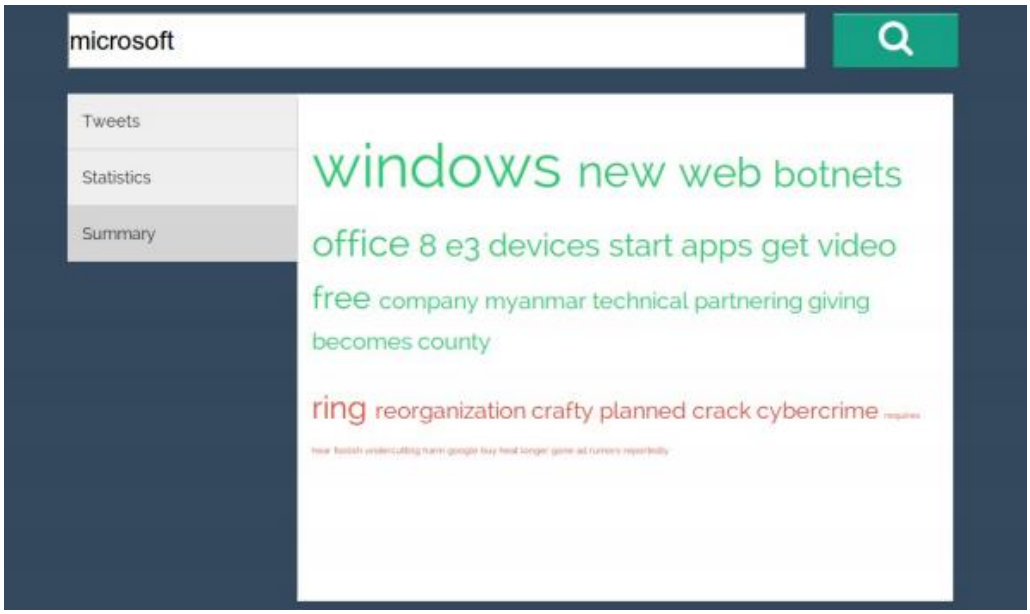


Рисунок 14 – Вкладка підсумку

ВИСНОВОК

Результатом роботи дипломного проекту є реалізація веб-сервісу для аналізу тональності повідомлень з соціальної мережі Twitter, за допомогою мови Python.

Створені програмні засоби можуть бути використані як самостійний продукт або ж як модуль для проектів покращення роботи з соціальними мережами.

В якості продовження роботи можна сформулювати і вирішити задачу класифікації повідомлень з мікроблогів на суб'єктивні і об'єктивні і такою класифікацією доповнити вже отриманий метод. Окремим завданням також можуть бути способи обробки зібраної інформації і можливість автоматично робити з неї висновки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The Impact Of Customer Service On Customer Lifetime Value [Електронний ресурс]. – Режим доступу : http://www.zendesk.com/resources/customer-service-and-lifetime-customer-value, 2019.

2. Аудиторія Twitter зросла на 60 мільйонів чоловік за рік [Електронний ресурс]. – Режим доступу : http://lenta.ru/news/ 2013/03/21/twohundred, 2019.

3. Peter D Turney. “Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews”. B: Proceedings of the 40th annual meeting on association for computational linguistics. [Текст] / Association for Computational Linguistics. 2012, с. 417-424.

4. Bo Pang, Lillian Lee, Shivakumar Vaithyanathan. “Thumbs up?: sentiment classification using machine learning techniques”. B: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. [Текст] / Association for Computational Linguistics. 2012, с. 79-86.

5. SentiStrength [Електронний ресурс]. – Режим доступу : http://sentistrength.wlv.ac.uk/#About, 2012.

6. Thelwall M., Buckley K., Paltoglou G., Cai D., Kappas A. Sentiment strength detection in short informal text . [Текст] / Journal of the American Society for Information Science and Technology. 2010.

7. Sentiment Analyzer [Електронний ресурс]. – Режим доступу : https://www.danielsoper.com/sentimentanalysis/default.aspx, 2013.

8. Система виведення знань з текстів «Аналітичний кур’єр» [Електронний ресурс]. – Режим доступу : http://www.i-teco.ru/ac.html, 2014.

9. RCO Fact Extractor SDK [Електронний ресурс]. – Режим доступу : http://www.rco.ru/product.asp?ob_no=5047, 2017

10. Ashequl Qadir, Ellen Riloff. “Bootstrapped Learning of Emotion Hashtags# hashtags4you”. [Текст] / B: WASSA 2013 (2013), с. 2.

11. Francesco Colace, Massimo De Santo, Luca Greco. “A Probabilistic

Approach to Tweets' Sentiment Classification". [Текст] / В: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (2013), с. 37-42.

12. Efstratios Kontopoulos "Ontology-based sentiment analysis of twitter posts". [Текст] / В: Expert Systems with Applications 40.10 (2013), с. 4065-4074.

13. Hassan Saif, Yulan He, Harith Alani. "Semantic sentiment analysis of twitter". [Текст] / В: The 11th International Semantic Web Conference. 2012.

14. Asli Celikyilmaz. "Probabilistic model-based sentiment analysis of twitter messages". [Текст] / В: Spoken Language Technology Workshop. 2010, с. 79-84.

15. Martin F Porter. Snowball. ". [Текст] / A language for stemming algorithms. 2011.

16. Christopher D Manning, Hinrich Schütze. [Текст] / Foundations of statistical natural language processing. MIT press, 2009.

17. Simon Tong, Daphne Koller. "Support vector machine active learning with applications to text classification". [Текст] / В: The Journal of Machine Learning Research 2 (2012), с. 45-66.

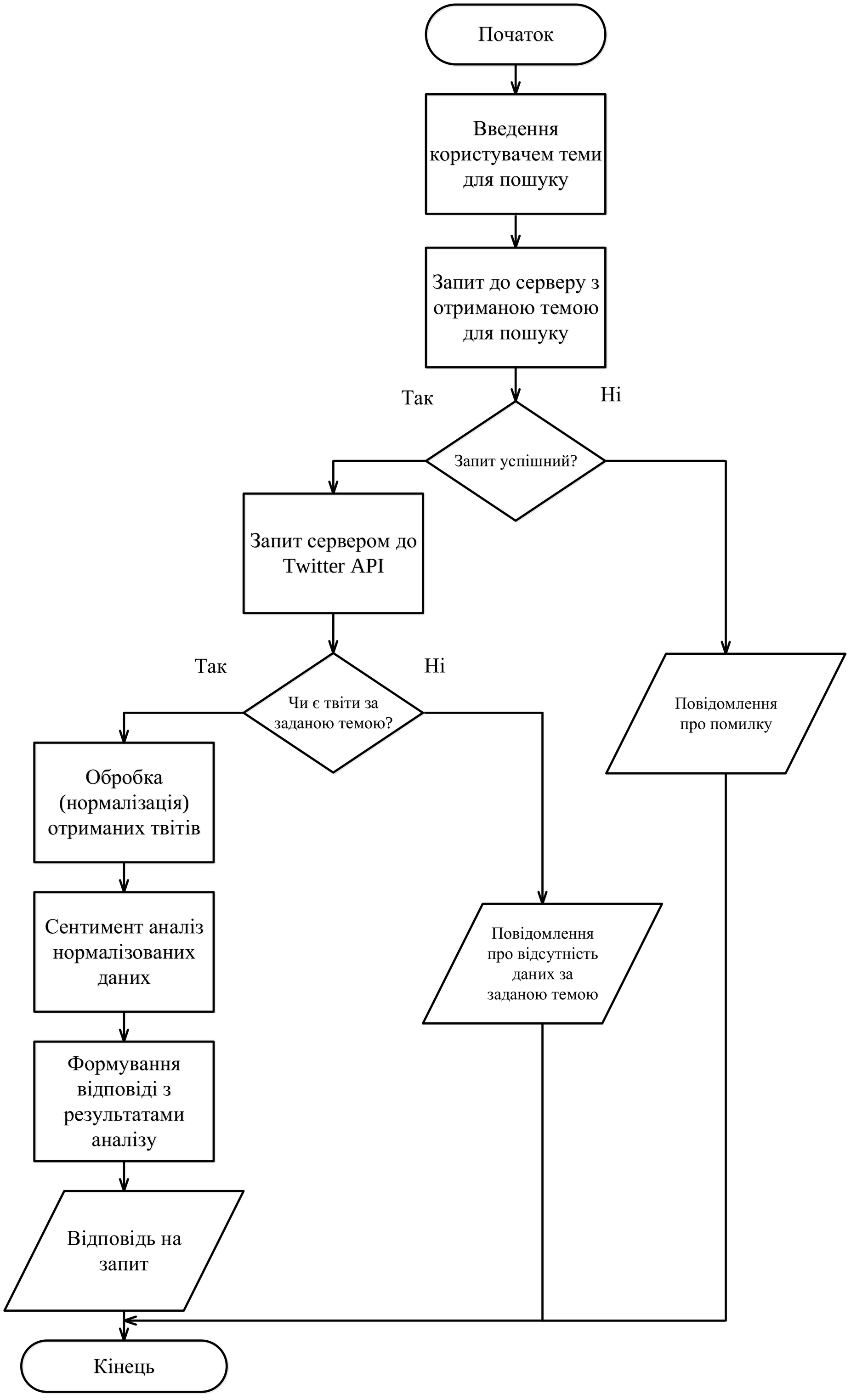
18. Kamal Nigam, John Lafferty, Andrew McCallum. "Using maximum entropy for text classification". [Текст] / В: IJCAI99 workshop on machine learning for information filtering. 2009, с. 61-67.

19. F. Pedregosa "Scikit-learn: Machine Learning in Python". [Текст] / В: Journal of Machine Learning Research 12 2011, с. 2825-2830.

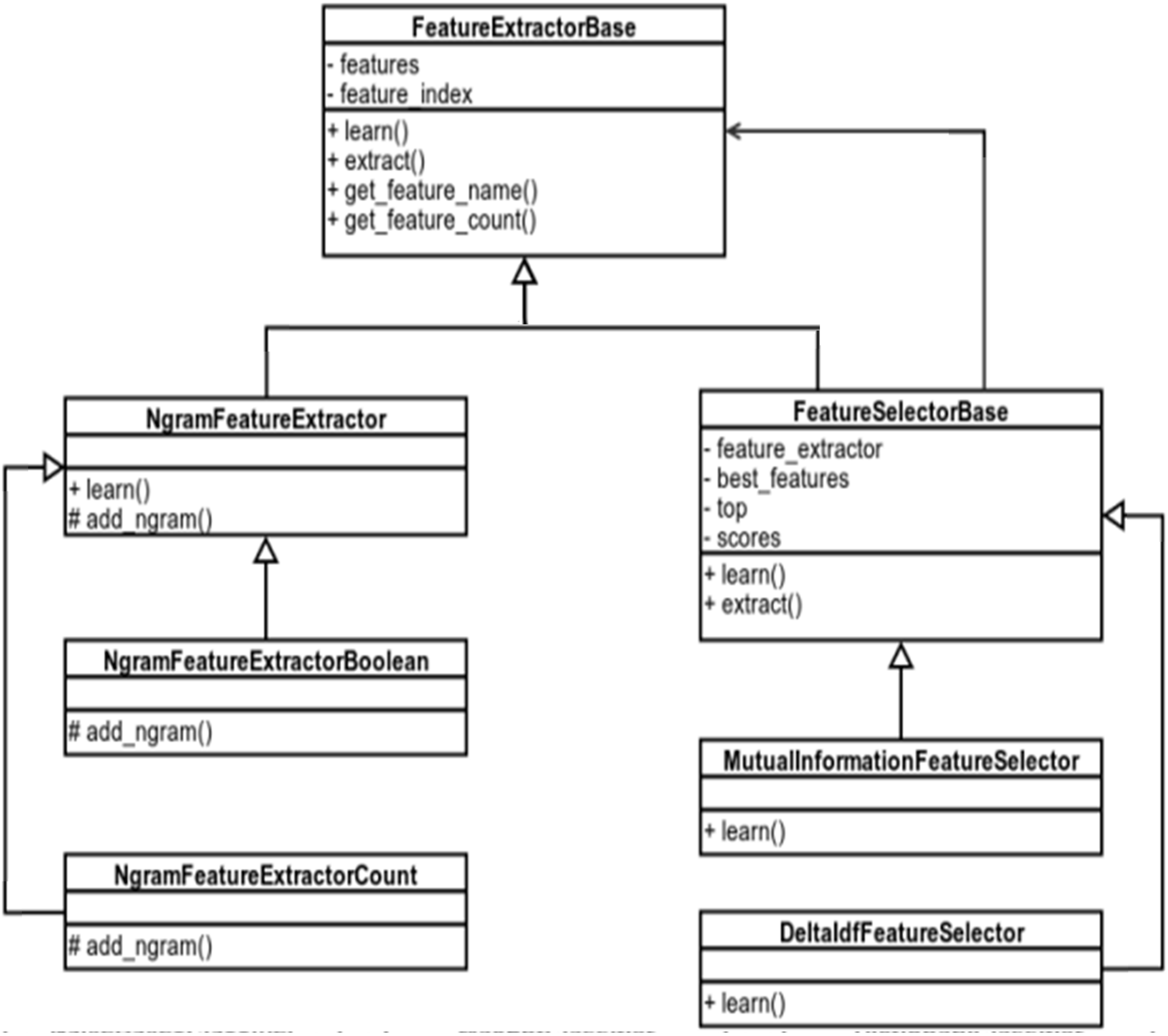
20. David A Field. "Laplacian smoothing and Delaunay triangulations". [Текст] / В: Communications in applied numerical methods 4.6, 2008, с. 709-712.

21. Jens Lehmann "DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia". [Текст] / В: Semantic Web Journal, 2014.

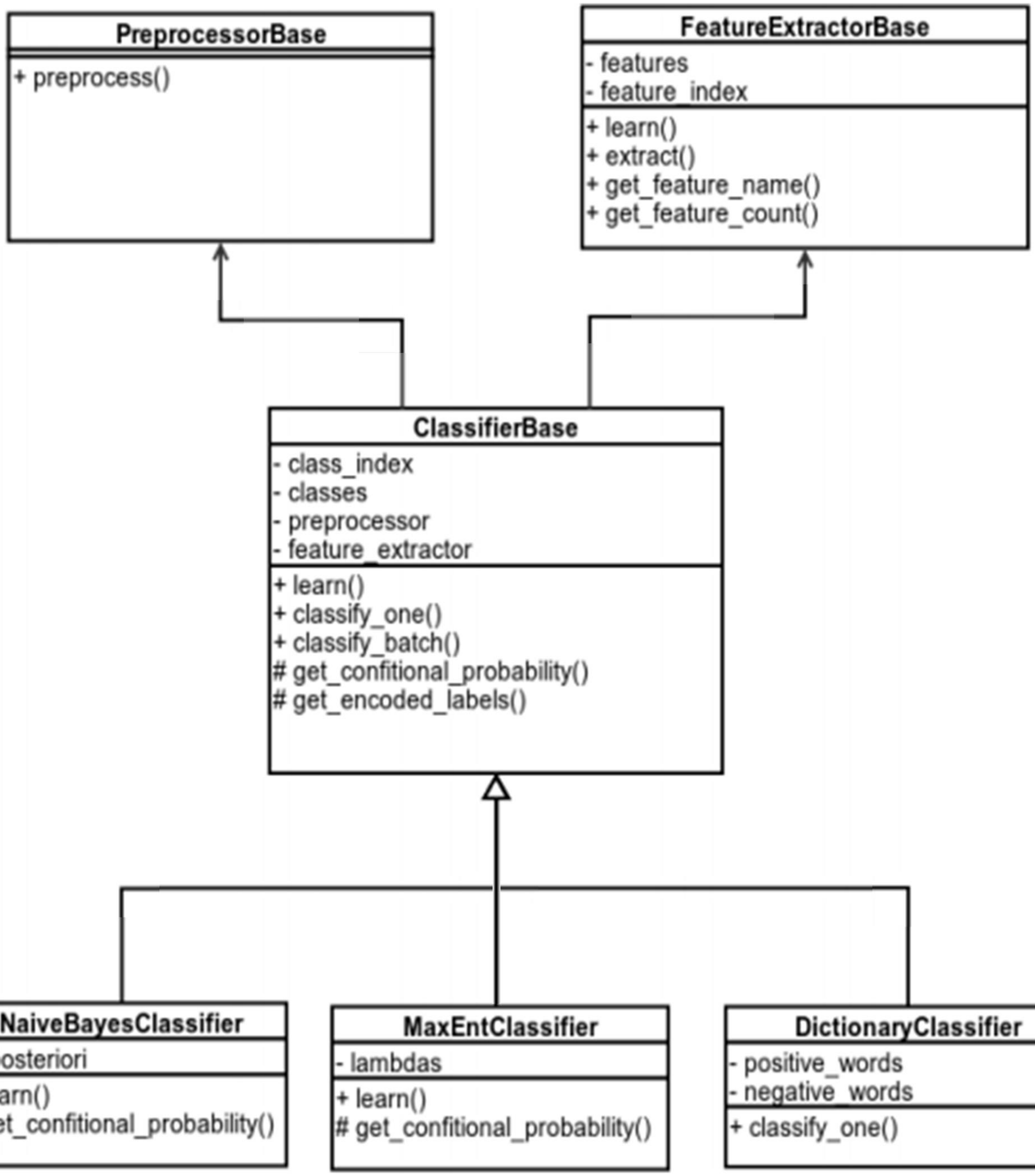
22. Steven Bird. "NLTK: the natural language toolkit". [Текст] / В: Proceedings of the COLING/ACL on Interactive presentation sessions. Association for Computational Linguistics. 2009, с. 69- 72.



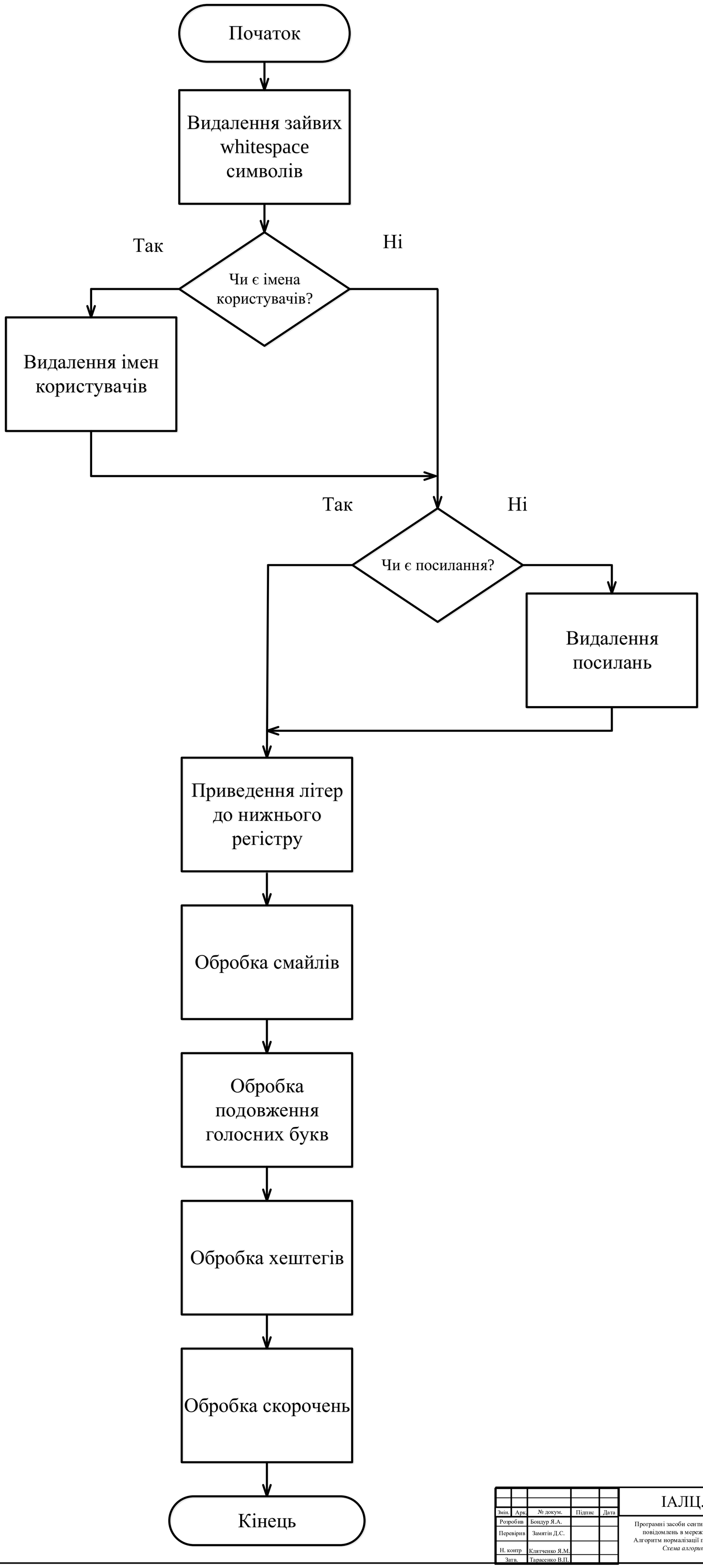
					ІАЛЦ.045490.005 Д1	
Змін.	Арк.	№ докум.	Підпис	Дата		
Розробив		Бондур Я.А.			Програмні засоби сентимент аналізу повідомлень в мережі Інтернет Алгоритм розробленої програми *Схема алгоритму*	Літ. / Аркуш: 1 / Аркушів: 1
Перевірив		Замятін Д.С.				
Н. контр		Клятченко Я.М.				«КПІ ім. Ігоря Сікорського», ФІПМ, КВ-53
Затв.		Тарасенко В.П.				



					ІАЛЦ.045490.006 Д2			
Змін.	Арк.	№ докум.	Підпис	Дата				
Розробила		Бондур Я.А.			Програмні засоби сентимент аналізу повідомлень в мережі Інтернет Діаграма класів для виділення ознак *Структурна схема*	Літ.	Аркуш	Аркушів
Перевірив		Замятін Д.С,					1	1
Н. контр		Клятченко Я.М.				«КПІ ім. Ігоря Сікорського», ФІПМ, КВ-53		
Затв.		Тарасенко В.П.						



					ІАЛЦ.045490.007 Д3		
Змін.	Арк.	№ докум.	Підпис	Дата			
Розробила		Бондур Я.А.			Програмні засоби сентимент аналізу повідомлень в мережі Інтернет Діаграма класів класифікаторів *Структурна схема*	Літ.	Аркуш 1 / Аркушів 1
Перевірив		Замятін Д.С.					
Н. контр		Клятченко Я.М.				«КПІ ім. Ігоря Сікорського», ФПМ, КВ-53	
Затв.		Тарасенко В.П.					



Змін.	Арк.	№ докум.	Підпис	Дата
Розробив		Бондур Я.А.		
Перевірив		Замятін Д.С.		
Н. контр		Клятченко Я.М.		
Затв.		Тарасенко В.П.		

ІАЛЦ.045490.008 Д4

Програмні засоби сентимент аналізу повідомлень в мережі Інтернет
Алгоритм нормалізації повідомлення
Схема алгоритму

Літ.	Аркуш	Аркушів
	1	1

«КПІ ім. Ігоря Сікорського», ФІПМ, КВ-53