Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра _____Програмної інженерії_____
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

_____другий (магістерський)_____
(рівень вищої освіти)

Дослідження методів семантичного аналізу для автоматизації обробки тексту
(тема)

Виконала: студент 2 курсу, групи ПЗСм-18-1_____

спеціальності 121- Інженерія програмного забезпечення____
(код і повна назва спеціальності)

Освітньо-професійної програми
_Програмне забезпечення систем_____

_____Тур Д. В._____
(прізвище, ініціали)

Керівник _____д. т. н. проф. Четвериков Г. Г._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____ З.В.Дудар

2019 р.

Харківський національний університет радіоелектроніки

Факультет_Комп'ютерних наук_____

Кафедра_Програмної інженерії_____

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність _121-Інженерія програмного забезпечення_____

(код і повна назва)

освітньо-професійна програма _Програмне забезпечення систем_____

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____»_____ 2019 р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові _Туру Дмитру Володимировичу_____

(прізвище, ім'я, по батькові)

1. Тема роботи _ Дослідження методів семантичного аналізу для автоматизації обробки тексту_____

затверджена наказом по університету від "___" _____2019 р № _____

2. Термін подання студентом роботи до екзаменаційної комісії _____2019 р.

3. Вихідні дані до роботи *алгоритми векторизації, методи векторизації, методи класифікації, пояснювальна записка. Використовувати ОС Windows*

4. Перелік питань, що потрібно опрацювати в роботі *аналіз проблемної галузі і постановка задачі, огляд застосування методів семантичного аналізу для автоатичної обробки тексту, огляд методів попередньої обробки та векторизації текстів, огляд методів класифікації для аналізу тональності*

5. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім’я, по батькові)	Позначка консультанта про виконання розділу	
		Підпис	дата
Спецчастина	д.т.н. проф. Четвериков Г. Г.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	25 вересня 2019р.	
2.	Огляд існуючих методів	10 жовтня 2019р.	
3.	Проведення дослідження методів семантичного аналізу для семантичного пошуку	7 листопада 2019р.	
4.	Підготовка пояснювальної записки	26 листопада 2019р.	
5.	Спецчастина		
6.	Підготовка презентації та доповіді		
7.	Попередній захист		
8.	Нормоконтроль, рецензування		
9.	Занесення диплома в електронний архів		
10.	Допуск до захисту у зав. Кафедри		

Дата видачі завдання _____ _____2019 р.

Студент _____
(підпис)

Керівник роботи _____ __ д.т.н. проф. Четвериков Г. Г._____

РЕФЕРАТ / ABSTRACT

Звіт з професійної практики: 87 с., 20 рис., 17 формул, 2 додатки, 30 джерел.

АВТОМАТИЧНА ОБРОБКА ТЕКСТУ, СЕМАНТИЧНИЙ АНАЛІЗ, АНАЛІЗ ТОНАЛЬНОСТІ, КЛАСИФІКАЦІЯ, НЕЙРОННА МЕРЕЖА, ВЕКТОРИЗАЦІЯ, PYTHON.

Об’єкт дослідження – методи семантичного аналізу.

Метою роботи дослідження є дослідження методів семантичного аналізу, зокрема, аналізу тональності тексту.

Методи розробки базуються на використанні мови програмування Python та її бібліотек: NLTK, TensorFlow та Pandas для застосування математичного апарату та аналізу текстових даних, а також інструмент Jupyter Notebook. У якості алгоритмів будуть використані векторизація, наївний байєсівський класифікатор та нейронна мережа.

Результат дослідження – виконано аналіз предметної галузі та аналогів, розглянуто методи векторизації та класифікації текстів, проведено ряд експериментів з метою виявлення оптимального підходу для аналізу тональності для різних датасетів, розроблено прототип програмної системи для аналізу тональності текстів.

AUTOMATIC TEXT PROCESSING, SEMANTIC ANALYSIS, SENTIMENT ANALYSIS, CLASSIFICATION, NEURAL NETWORK, VECTORIZATION, PYTHON.

The object of research is semantic analysis methods.

The aim is the research of semantic analysis methods and, specifically, text sentiment analysis.

Methods of developing technology based on using of Python programing language and its libraries: NLTK, TensorFlow and Pandas for the and data analysis. There also will be used vectorization, naive Bayes classifiers and neural network.

Результат дослідження – виконано аналіз предметної галузі та аналогів, розглянуто методи векторизації та класифікації текстів, проведено ряд експериментів з метою виявлення оптимального підходу для аналізу тональності для різних датасетів, розроблено прототип програмної системи для аналізу тональності текстів.

The result of the research is the analysis of the subject domain and the analogues, review of the vectorization and text classification methods, run a series of experiments with the goal to define the best solutions for different datasets and implement the software prototype for sentiment text analysis.

ЗМІСТ

ВСТУП

У зв’язку із постійним збільшенням електронної інформації, усе більш актуальним стає питання її автоматичної обробки. Звісно, обробка тексту також до них відноситься. Вона дозволяє автоматизувати такі процеси як оцифрування друкованих текстів, розпізнання людської мови, генерація тексту, пошук за текстовими ресурсами, вилучення із них змісту та багато іншого.

Одним із таких напрямків є семантичний аналіз, за допомогою якого можна зрозуміти зміст тексту. Зазвичай вважається, що семантичний аналіз є наступним кроком після лексичного, синтаксичного та деяких інших аналізів, але сьогодні можливе розуміння змісту без цих попередніх етапів, так званий латентний підхід [1].

Семантичний аналіз може бути застосований для різних напрямків обробки тексту, наприклад, для створення діалогових систем, анотування, адаптації літературних творів і так далі. Деякі з них допомагають зробити життя пересічних громадян краще, інші направлені на аналітику у сфері бізнесу. Серед останніх – аналіз емоційного забарвлення. Цей напрям є доволі молодим та перспективним. З його допомогою можна визначити емоційне забарвлення тексту. Таким чином можна не тільки визначати відношення споживачів до бренду або виборців до певного кандидату, але і робити деякі прогнози та аналізувати тенденції у суспільстві. Особливо актуальне застосування такого аналізу для текстів, що згенеровані користувачами, такі як відгуки, коментарі, твіти, повідомлення та інше. Кожен із цих типів даних має свої особливості, які можуть бути враховані під час аналізу.

Мета роботи – підбір оптимальних методів аналізу тональності для різних типів даних та датасетів різного розміру.

Об’єкт дослідження – використання методів семантичного аналізу для автоматизації обробки текстів.

Предмет дослідження – процес автоматичного визначення емоційного забарвлення текстів.

Для аналізу предметної галузі було використано такі теоретичні методи дослідження як аналіз та синтез. Для проведення практичної частини дослідження були застосовані такі емпіричні методи як експеримент, вимірювання та порівняння.

У результаті дослідження було виявлено, що лінійні класифікатори менше страждають від невеликої навчальної вибірки, аніж нейронні мережі. Нейронні мережі гарно працюють лише на великих датасетах. Окрім цього, було виявлено, що при використанні нейронної мережі на великих датасетах має сенс використовувати попередньо навчений Word2Vec, оскільки це збільшує точність класифікації. Ще одним цікавим спостереженням було те, що нейронна мережа класифікувала відгуки із більшою точністю, ніж твіти, при однаковій кількості документів. Вірогідно, для кращого навчання сумарна кількість тексту важливіша, ніж сумарна кількість документів у вибірці. Також було виявлено, що застосування стемінгу у поєднанні із Word2Vec не призводить до покращення результатів класифікації, тому його застосування не виправдане.

Результати, що отримані у результаті проведеного дослідження можуть бути використані для підбору оптимального методів аналізу тональності в залежності від вхідних даних системами, які являють собою сервіси для визначення емоційного забарвлення або вбудовуються у систему, що автоматично аналізує згенеровані користувачами тексти. Також отримані закономірності можуть допомоги у задачах вилучення корисних даних.

Методи, що у результаті дослідження були визначені найкращими, були застосовані для розробки системи аналізу тональності, яка визначає емоційне забарвлення твітів та відгуків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Семантичний аналіз та його застосування в обробці тексту

У зв’язку із появою Інтернету надзвичайно зросла кількість інформації. Змінився і формат доступу людей до даних. Так, наприклад, тепер їм не потрібно йти до бібліотеки або у певне бюро, щоб зайти необхідну інформацію. Чимало процесів автоматизувалися і робота із інформацію у тому числі. Але це має і негативні наслідки. Зокрема, людина просто не здатна самостійно опрацювати незліченні дані. У зв’язку комп’ютеризацією процесів збільшуються вимоги до швидкості роботи працівників, від яких вимагають усе більше, вважаючи, що тепер практично усю роботу можна зробити дуже швидко. На жаль, незважаючи на автоматизацію, можливості сприйняття людини обмежені, пошук необхідної інформації та її обробка, враховуючи кількість даних, поглинає дуже багато часу та зусиль. Значимість даних також зростає, для бізнесу надзвичайно важливі статистичні відомості, відношення споживачів до бренду та інше. Однак, комп’ютерні технології також покращуються, що робить життя людини простіше. Активно розвивається і галузь комп’ютерної лінгвістики та автоматична обробка природних мов.

Комп’ютерна лінгвістика – це галузь, яка виникла на перетині лінгвістики, математики, інформатики та штучного інтелекту [1].

Лінгвістика включає наступні області:

- фонологія (вивчає звуки та їх сполуки);
- морфологія (вивчає форму та будову слів);
- лексикографія (займається описом лексики природних мов, граматичні та семантичні властивості окремих слів та методи створення словників);
- синтаксис (вивчає структуру речень, порядок слідування слів та загальні властивості речень);
- семантика (займається змістом слів, речень та інших одиниць мови).

У моделюванні природної мови зазвичай виділяють кілька рівнів: рівень речень (синтаксичний рівень), рівень слів (морфологічний), рівень фонем (фонологічний). Однак, питання про кількість рівнів все ще відкрите. Можуть також розглядатися підрівні. Інколи виділяють також лексичний (лексеми) рівень. Ще одним рівнем, що активно обговорюється, є семантичний. Аргументом його виділення є те, що люди зазвичай запам’ятовують зміст твердження, а не його конкретну мовну форму.

При розробці систем автоматичної обробки природної мови часто використовуються модулі, кожний із яких виконує свою певну функцію. Часто використовуються наступні модулі:

- графематичний аналіз (або сегментація), який виділяє токени (словоформи та речення), у цьому модулі відбувається перехід від символів до слів;
- морфологічний аналіз, що виконує перехід від словоформ до їх словникових форм чи основ;
- синтаксичний аналіз, який виявляє синтаксичні зв’язки слів та структуру речень з точки зору граматики;
- семантичний аналіз, що виявляє зміст фраз.

Що стосується методів обробки семантики текстових даних, то їх умовно можна поділити на такі види: ті, що відносяться до «старої школи» та методи сучасного підходу.

«Стара школа» передбачає явний семантичний аналіз, а також використовує онтології. Онтологія – це формалізоване представлення певної області знань за допомогою концепцій. У випадку із обробкою текстові інформації, онтології – це співставлення слів та словосполучень із поняттями, які вони можуть виражати у тексті [2].

Такий підхід також потребує застосування доволі складних технік для побудови дерев синтаксичного розбору та вирішення питання омонімії на рівні речень, що є доволі складним завданням.

Таким чином, використання методів, в основі яких лежать онтології та численні рукописні правила, потребує чимало зусиль, часу та ресурсів, оскільки у рамках отримання точних результатів такого семантичного аналізу потрібно вирішувати відразу багато інших завдань. Особливо трудомісткими є написання правил, які важко застосовувати для інших мов. Не зважаючи на недоліки, цей підхід має і переваги. Зокрема, його результати легко інтерпретувати та розуміти, де виникають помилки. Ще однією перевагою «старої школи» є можливість роботи із короткими текстами та реченнями [3].

Альтернативою такого класичного підходу є сучасний підхід, який базується на латентному семантичному аналізі та застосовує методи машинного навчання [4]. Зараз він є доволі розповсюдженим та продовжує набирати популярність. Його особливостями є доволі точні результати роботи на великих обсягах даних (що, власне, і характерне для машинного навчання). Також латентний семантичний аналіз працює із відносно невеликою кількістю параметрів, на відміну від «старої школи», де використання численних правил є необхідністю [5]. На сьогоднішній день є чимало готових інструментів для роботи за допомогою такого підходу. Недоліками цього підходу є складність інтерпретації результатів та відносно погана робота на невеликих обсягах тексту.

Окрім цього, для представлення семантики можуть бути використані такі формалізми як формули предикатів, які виражають властивості, стани, відносини та дії, а також семантичні мережі (графи, у яких вершини – поняття, а ребра – відносини між ними).

Семантичний аналіз є важливою складовою для багатьох напрямків автоматичної обробки тексту.

Мабуть, одним із найвідоміших прикладів застосування семантичного аналізу є машинний переклад. Власне кажучи, дана задача і була одним із перших напрямків, який викликав інтерес до автоматичної обробки текстів. Для машинного перекладу зазвичай використовується чимало інших видів аналізу (наприклад, синтаксичний), а також у для його реалізації часто застосовуються ще й засоби генерації тексту, де, до речі, семантика також використовується. Цікавим варіантом

роботи є використання паралельних корпусів, коли співставляються два тексти, написані різними мовами; такий метод часто допомагає виконувати не просто дослівний переклад, а враховувати контекст. На сьогоднішній день існує не так багато систем автоматичного перекладу, які дають дійсно гарні результати.

Ще один популярний варіант використання семантичного аналізу – інформаційний пошук. Використання семантичного пошуку дозволяє шукати за змістом. Більшість пошукових систем хоч і здійснюють доволі якісний пошук даних, здебільшого спираються на інші фактори, такі як ключові слова, метадані, популярність, структура сторінок та інші.

Важливе значення такий аналіз має і для вилучення із тексту фактів та сутностей. Особливо корисно це може бути у випадку агрегування новин або, наприклад, при розробці фільтрації.

Даний вид аналізу також використовується для автоматичного створення анотацій. Для його реалізації можливе застосування двох підходів. Перший полягає у аналізі змісту документу та вилученні речень, що мають найбільшу вагу з точки зору семантики. Другий варіант – більш складний, оскільки передбачає автоматичну генерацію тексту, відповідно до виявленого змісту.

Актуальним питанням, для якого необхідний семантичний аналіз, є адаптація документів та творів. Звісно, адаптація літературних творів є більш популярною. Це актуально у випадку спрощення текстів для дітей (особливо важко читати неадаптовані тексти дітям емігрантів), не носіїв мови, а також для заміни застарілих слів (такі слова часто підвищують складність читання для більшості людей). Однак, адаптація може бути використана не тільки для літератури, але й, наприклад, для юридичних документів або інших документів, де використовується доволі специфічна лексика, оскільки людям, що не мають відповідної освіти зазвичай буває важко зрозуміти подібний текст, що містить специфічну для певної предметної галузі термінології.

Ще одним напрямком, де застосовується семантичний аналіз, є фільтрація спаму. Хоча варто відмітити, що сучасні системи досить непогано справляються із цією задачею.

Корисним цей аналіз є і для діалоговим систем. Прикладом таких систем можуть слугувати чат-боти. Вони є доволі популярними серед онлайн-сервісів, оскільки дають змогу користувачам швидше отримати необхідну інформацію. Однак, деякі подібні системи створені не стільки для надання необхідної інформації, скільки для імітації спілкування із людиною. Звісно, що у випадку, коли потрібно надати інформацію відповідно до запиту, аналізу за ключовими словами може виявитися замало, оскільки інколи людина не знає слова, яким може вдало описати те, про що вона хоче отримати інформацію. У таких ситуаціях розуміння змісту, а отже і семантичний аналіз, буде у нагоді. Варто зазначити, що системи подібного виду можуть як просто пропонувати знайдені результати чи відповідати завчасно підготовлені відповіді, так і генерувати текст. Зрозуміло, що другий варіант є більш складним і також потребує врахування семантики.

Важливе значення семантичний аналіз також має для перевірки на плагіат. Причини зрозумілі: люди можуть використовувати інші фрази та міняти слова та речення місцями, але це не впливатиме на зміст. У цьому разі врахування змісту є просто необхідним.

1.2 Аналіз тональності, його застосування та особливості імплементації

Доволі цікавим напрямком автоматичної обробки тексту, що активно розвивається в останнє десятиріччя, є визначення емоційного забарвлення тексту [6]. Зазвичай цей вид аналізу використовується на текстах, що згенеровані користувачами. Даний напрям є досить актуальним у зв’язку із розвитком соціальних мереж, сервісів із відгуками користувачів та рекомендаціями щодо товарів та послуг. Аналіз текстів соціальних мереж часто використовується для соціальних досліджень та політичних вподобань, а також для виявлення суджень, що містять агресію, заклик до конфліктів, фейкові новин та інше. Моніторинг позитивних та негативних відгуків цікавить чимало компаній для яких важливо

слідкувати за репутацією та вчасно реагувати на негативні відгуки. Важливим аналіз тональності може бути і для виявлення трендів у новинах, звітах, публікаціях.

Особливістю відгуків, що згенеровані користувачами, є те, що вони зазвичай містять судження одного автора лише про одну сутність, але можуть містити відомості щодо її різних аспектів. Варто зазначити, що короткі тексти вимагають доволі точного аналізу. Тексти із новинами характеризуються можливістю вміщувати багато авторів та сутностей із різними оцінками. У такому разі складність аналіз дуже зростає. При аналізі потрібно враховувати і те, що текст, який пишуть користувачі, може відрізнятися від літературної мови. Люди можуть використовувати різну лексику в залежності від віку, місця проживання та інших факторів.

Аналіз тональності може бути різних видів. Найпростіший – бінарний, коли є усього два класи, тобто відгук може бути або негативним, або позитивним [7]. Втім, не завжди автор дає такий відгук, який можна класифікувати подібним чином, іноді він не несе конкретного емоційного забарвлення. У такому випадку класів буде вже три: позитивний, негативний та нейтральний. Один із підходів полягає у впровадженні так званої сірої зони. Тобто коли аналізатор не може точно визначити, позитивний відгук чи негативний, програма відносить його до цієї сірої зони. Даний метод дозволяє збільшити точність аналізу на полярних значеннях (точно позитивний чи точно негативний). У таких випадках необхідно підбирати критерії класифікації аби модель давала оптимальні результати. Та буває і так, що навіть трьох класів може бути недостатньо, адже відгук може бути скоріше позитивним або скоріше негативним, тоді класів буде вже п’ять.

При аналізі емоційного забарвлення варто мати на увазі, що відгук може бути і позитивний, і негативний одночасно. Таке часто трапляється, коли людина пише спочатку, що їй сподобалося, а наступною частиною – що викликало незадоволення або що можна було б покращити. Дослідники також можуть вилучати, до якої саме сутності відноситься та чи інша оцінка, щоб зробити аналіз більш точним.

Наступним рівнем аналізу тональності є аналіз за аспектами, коли визначається, до якого саме параметру сутності відноситься оцінка.

Складність при оцінки емоційного забарвлення тексту також може становити залежність від контексту, тобто певні слова, що не містять конкретного забарвлення можуть мати позитивне чи негативне забарвлення у конкретному контексті. Часто це залежить від предметної галузі, до якої відноситься судження. Часом буває і так, що одне слово може мати різне забарвлення для різних аспектів сутності навіть у межах однієї предметної галузі.

Важливе значення при визначенні тональності також мають так звані модифікатори полярності такі як «занадто», «дуже», «більш», «менш», частка «не» та інші. Для їх врахування потрібно розробити спеціальні моделі. Доволі поширеним є приписування певних коефіцієнтів до модифікаторів, які допомагають зрозуміти відтінок слів, до яких відноситься модифікатор.

Ще один фактор, що впливає на оцінку, – показник ірреальності контексту. Це може трапитися, коли людина висловлює тільки свої здогадки, надії, побажання. Тобто фактично судження не є оцінкою, тому оптимальніше знижувати вплив таких висловлювань на тональність. Зазвичай такі моменти характеризуються спеціальними мовними конструкціями.

Порівняння також ускладнюють аналіз відношення автора. Річ у тому, що вони зазвичай містять додаткові сутності чи факти, тому доволі складно виявити, які саме оцінки відносяться до цих сутностей.

Існує два основних виду аналізу емоційного забарвлення: інженерно-лінгвістичний (базується на правилах та словниках) та підхід на базі машинного навчання. Останні у свою чергу діляться на навчання з учителем (коли система спершу «навчається» на розмічених текстах) та без. Є і такі методи, які комбінують машинне навчання та використання словників із оціночною лексикою.

Що стосується інженерно-лінгвістичного методу, він використовує спеціальні словники із оціночними словами та виразами, а також застосовує лінгвістичні правила з урахуванням контексту. Тональність речень визначають базуючись на тональності його слів, а тональність множини речень у свою чергу

визначає тональність усього документу. Найбільш популярними правилами є використання слів модифікаторів (більш, дуже, не та інші), складання оцінок слів тексту та встановлення негативної оцінки до словосполучення, у якому використано хоча б одне негативне судження.

Методи машинного навчання використовують розмічені вручну тексти та певні алгоритми класифікації [8]. Окрім цього, для успішного використання даного методу необхідно визначити ознаки, а також обрати спосіб підрахунку ваги ознак (наприклад, це може бути проста булева форма або варіант, коли враховується частотність). Перед тим, як починати безпосередню класифікацію, із тексту видаляються стоп-слова (із ними потрібно бути обережними, оскільки такі стоп-слова як частка «не» можуть суттєво впливати на результат визначення відношення автора), а також виконується лематизація або стемінг.

Таким чином, загальний алгоритм навчання з учителем можна визначити як наступний:

- збір колекції документів (або відгуків) для навчання класифікатору;
- представлення кожного документи у вигляді вектору ознак;
- розмітка документів (зазначення вірної відповіді: позитивна чи негативна тональність);
- вибір алгоритму для класифікації та навчання класифікатора;
- використання отриманої моделі.

Поширеною та популярною моделлю для представлення тексту у більш зручній для обробки формі є метод «мішка слів». Його ідея полягає у представленні тексту у вигляді матриці, де кожний рядок – окремий документ або текст, а стовпець – деяке слово. Перетин рядків та стовпців містить кількість входжень слів у певний документ.

Для створення вектору ознак часто використовуються ознаки із певними вагами, хоча це і не є обов’язковим для усіх класифікаторів. Але для таких підходів, як метод опорних векторів, ваги можуть суттєво покращити результати. Одним із найбільш популярних варіантів присвоєння ваг є метод TF-IDF, принцип якого полягає у тому, що вага слова пропорційна частоті використання цього слова у

документі і обернено пропорційна частоті використання слова у всіх документах колекції. Тим не менше, навіть незважаючи на популярність та доволі точну роботу, даний підхід не дуже підходить для аналізу тональності, на відміну від випадку інформаційного пошуку. Це пояснюється тим, що для виявлення емоційного забарвлення найбільш частотні слова не настільки важливі, як для пошуку. З цієї причини для аналізу тональності зазвичай використовуються бінарні ваги, коли вага може бути або 1, або 0, в залежності від того, є ознака у тексті або ні.

Серед популярних моделей класифікації варто згадати метод наївної байєсівської класифікації та SVM (метод опорних векторів) [9].

Суть методу SVM полягає у розподіленні елементів на класи за допомогою деякої гіперплощини. Якщо розмірність простору усього два (а стільки й потрібно для роботи із двома класами), то гіперплощиною буде пряма. У рамках класифікації необхідно знайти пряму, відстань від якої до кожного класу – максимальна. Ті вектори, які розміщені найближче до гіперплощини, називають опорними векторами. Вважається, що даний метод не дуже підходить для аналізу тональності, оскільки має не дуже високу точність.

Наївний байєсівський класифікатор базується на умовній вірогідності належності деякого документу до певного класу. Цей метод показує доволі точні результати, незважаючи на серйозне допущення: усі ознаки документа незалежні один від одного. У дійсності таке твердження є здебільшого невірним, але даний алгоритм широкого використовується через його простоту реалізації та інтерпретації. Ще одним допущенням є те, що положення терміну у реченні не важливе [10]. Тож, для знаходження найбільш імовірного класу потрібно підрахувати умовні вірогідності документу для кожного класу та обрати той клас, який має найбільшу вірогідність.

Переваги цього методу полягають у швидкості роботи, відносній простоті програмної реалізації і легкості інтерпретації результатів роботи. Втім якість класифікації часто залишається доволі низькою, також даний алгоритм не варто застосувати у разі наявності залежності ознак.

Для класифікації текстів може бути використаний метод дерев рішень. Його ідея полягає у тому, що існує деякий ациклічний граф, по якому відбувається класифікація об’єктів, які описані деяким набором ознак. Кожен вузол дерева містить умовне розгалуження по одній із цих ознак. Під час класифікації алгоритм поетапно переходить від одного вузла до іншого в залежності від значення ознак. Класифікація завершується, коли алгоритм досягає одного із листів дерева. Значення листа відповідає класу, до якого належить об’єкт.

Зазвичай використовують класифікацію по бінарному дереву, тобто просто перевіряють, наявна певна ознака чи ні.

Цей метод також характеризується відносною простотою реалізації та легкістю інтерпретації результатів, хоча серед його недоліків варто відмітити нестійкість алгоритму до викидів у початкових даних та необхідність великого обсягу даних для отримання точних результатів роботи.

Ще одним методом класифікації є метод логістичної регресії (лінійний метод), що передрікає вірогідність деякої події по значенням множини ознак. Відповідно до алгоритму вводиться змінна y, яка приймає значення 1 (подія відбулася) чи 0 (подія не відбулася) та множина незалежних змінних (ознак) за допомогою яких визначається вірогідність прийняття залежною змінною певного значення. У випадку класифікації документів (текстів) роль залежної змінної виконує клас, а незалежних змінних – множина документів (тексів).

До переваг логістичної регресії варто віднести якість його результатів, можливість інкрементного навчання та відносну простоту реалізації. Серед недоліків – складність інтерпретації параметрів алгоритму та нестійкість до викидів у вхідних даних.

Усе більш поширеним стає використання нейронних мереж для визначення емоційного забарвлення [11]. У простому вигляді ідея полягає у тому, що при аналізі тексту створюється словник, у якому слова матимуть порядковий номер. Далі текст, що аналізується, пропускається через цей словник, створюється вектор із нулів та одиниць, що ставиться у відповідність словнику: нуль – коли слово із словника не було використано у тексті, і одиниця, коли слово зустрілося у тексті.

Потім ці дані відправляються вхідному шару нейронної мережі і далі наступним шарам, кількість нейронів у яких щоразу пропорційно зменшується і кожен нейрон поточного шару пов’язаний із кожним нейроном попереднього. Зрештою, на виході буде усього один шар, який складається із двох нейронів, сума яких буде дорівнювати одиниці. Таким чином, можна сказати, що з певною вірогідністю судження буде позитивним чи негативним.

Машинне навчання зазвичай дає кращі результати, аніж інші методи аналізу тональності, особливо, коли використовується доволі велика вибірка текстів для навчання.

Тим не менше, методи машинного навчання мають і свої недоліки, серед яких наступні:

– такі алгоритми дуже чутливі до предметної галузі, зазвичай, при застосуванні алгоритму на іншій галузі необхідне деяке перенавчання;
– результати важко інтерпретувати, оскільки їх складно пояснити, і через це виявлення та виправлення помилок ускладняється;
– ручна розмітка текстів для навчання – тривалий та трудомісткий процес [12].

Машинне навчання без учителя є автоматичним та не потребує вибірки для навчання, але, на жаль, на сьогоднішній день має низьку точність.

На останок потрібно зазначити деякі практичні особливості виявлення тональності тексту. Наприклад, слід бути обережним із стемінгом та лематизацією [13], оскільки деякі специфічні форми слів є носіями інформації щодо емоційного забарвлення, тому інколи така попередня обробка може навіть погіршити результати аналізу.

Варте уваги і те, що деякі слова можуть бути із запереченням (наприклад, часткою «не»). З цієї причини використання методу «мішка слів» може погіршити визначення, якого саме слова стосувалося заперечення. Щоб вирішити цю проблему можна поєднувати частку «не» із словом поруч. Звісно, це простий варіант і він не завжди буде працювати добре, оскільки не завжди заперечення дійсно відноситься саме до найближчого слова.

Ще один аспект – це вид тексту, що використовується для аналізу. Зазвичай для визначення тональності використовуються тексти користувачів, тож скоріше за все буде використано розмовний стиль, також може бути використано сленг, а сам текст із великою вірогідністю міститиме помилки та друкарські помилки. У цьому разі може мати сенс використати метод n-грам, що допомагає виправляти деякі помилки, базуючись на даних щодо вірогідності.

Пунктуація також є важливою для визначення тональності. Наприклад, знак оклику майже завжди означає, що фрагмент має сильно виражене емоційне забарвлення, тому йому можна надати більшої ваги. Окрім того, користувач може використовувати різноманітні смайли, що також несуть у собі інформацію щодо емоцій, як негативних, так і позитивних, тому їх також варто враховувати при аналізі.

Одним із найпопулярніших інструментів для реалізаціях семантичного аналізу і виявлення емоційного забарвлення у тому числі є бібліотека Word2Vec. Вона являє собою ряд моделей за допомогою яких виконується робота над словами. Реалізовані ці системи як двошарові нейронні мережі, які допомагають відтворювати контекст. У якості вхідних даних використовується великий корпус тексту, який перетворюється у багатовимірний векторний простір. Кожне унікальне слово відповідає певному вектору у просторі. Особливість полягає у тому, що слова, які мають спільний контекст у корпусі розташовуються ближче один до одного у векторному просторі. Такий підхід перевершує попередні моделі-аналоги, у тому числі й латентний семантичний аналіз.

Word2Vec допускає можливість використання двох архітектур: «мішка слів» та скіп-грам. У першому випадку поточне слово передбачається із урахуванням оточуючих контекстних слів без урахування порядку (це одне із основних припущень даного підходу). У другій архітектурі поточне слово використовується для передбачення оточуючих контекстних слів. Зазвичай архітектура краще підходить для слів, що рідко зустрічаються, хоча працює цей підхід повільніше.

Дана бібліотека також допускає можливість використання параметрів для покращення результатів навчання. До таких параметрів можна віднести наступні:

- алгоритм навчання (ієрархічний softmax для слів, які рідко зустрічаються, та негативна вибірка для тих, що зустрічаються часто);
- підвибірка (можна налаштовувати в залежності від частотності слів);
- кількість вимірів, які впливають на якість вбудови слів (зазвичай у діапазоні від 100 до 1000);
- контексте вікно, яке визначає скільки слів до та після поточного будуть враховані у якості контекстних слів (зазвичай 10 для скіп-грам та 5 для «мішка слів»).

Дана бібліотека реалізована для кількох мов програмування, у тому числі для Java та Python.

1.3 Аналіз аналогів та наукових публікації

Одним з продуктів, який займається аналізом емоційного забарвлення тексту, є OpenAmplify [14]. Ця система є одним із лідерів в обробці природних мов. Вона дозволяє виділити теми, бренди, людей, емоції, атрибути, наміри, обмеження у часі, які містяться у тексті. Також за допомогою даної платформи можна визначити стиль автору та навіть аналізувати авторство. Працює продукт тільки з англійською мовою.

Система має чотири модулі:

- лінгвістичний, що складається із терміналу POS, присвоює словам теги, що відповідають частинам мови, chunking та парсинг;
- модуль A, визначає тему кожного речення;
- модуль B, який відповідає за ідентифікацію домену;
- модуль полярності, який займається аналізом тональності.

Останній модуль складається із лексики та чисельних правил. За замовчуванням модуль присвоює кожному слову речення значення полярності: 1 – позитивний, 0 – нейтральний, -1 – негативний. Далі речення розбивається на

фрази і на основі оцінки окремих слів оцінюється тональність фрази і так далі. Таким чином, буде отримана оцінка твердження між -1 до +1.

Результат роботи на текстами англійською мовою може бути поміщений у зручних та практичних структурах даних, таких як XML та JSON.

Однією із найвідоміших платформ автоматичної обробки тексту, що включає у себе й аналіз тональності, є SAS Visual Text Analytics (див. рис. 1.1).

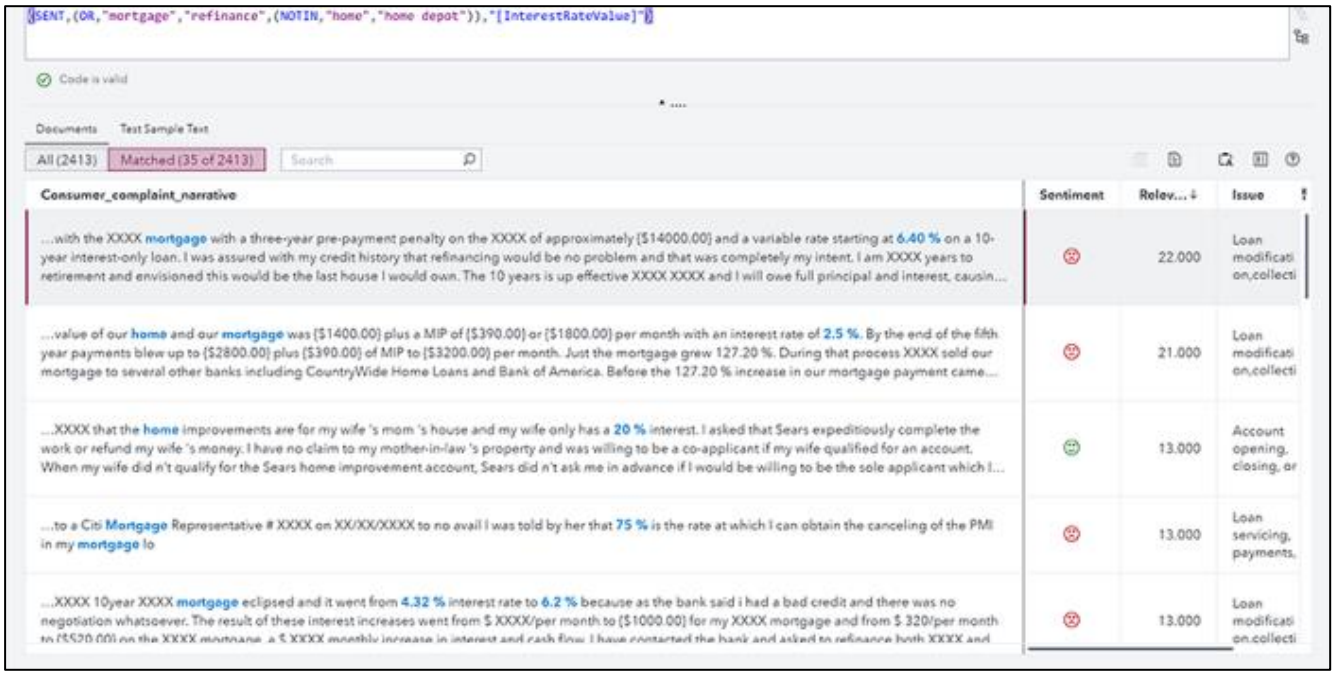

Рисунок 1.1 – Робота аналізатору тональності у SAS Visual Text Analytics

Програмне забезпечення комбінує машинне навчання та використання лінгвістичних правил. До можливостей цієї системи можна віднести наступні функції:

- підготовку та візуалізацію даних (зчитування, очистка та перетворення
- даних із різних видів сховищ, бере до уваги такі фактори, як локалізація та інтернаціоналізація, можливість візуалізації виділення сутностей, фактів та їх зав’язків за допомогою мережевих діаграм або термінальної карти);
- парсинг (розподілене накопичення тексту, токенізація, лематизація, аналіз орфографії, яка відбувається шляхом порівняння із набором

схожих слів, визначення частин мови у відповідності до їх контексту, виявлення меж речень, виявлення теми, знаходження синтаксичних залежностей);

- аналіз трендів (використовується два методи машинного навчання singular value decomposition та латентне розподілення Діріхле для групування документів по спільним темам, ведеться підрахунок релеватності, наскільки документ відповідає темі, і якщо він більше порогового значення, документ отримує бінарний флаг належності теми);
- витягування інформації (за допомогою розпізнання об’єктів, виявлення зв’язків та опорних посилань неструктуровані дані перетворюються у структруровані, виявлення загальних об’єктів);
- гібридні підходи до моделювання (токенізація, лематизація, виявлення орфографічних помилок, використання переліку стоп-слів, виявлення ключових понять за допомогою лінгвістичних правил, класифікація документів за допомогою нейронних мереж);
- аналіз емоційного забарвлення (ідентифікує та аналізує терміни, фрази або символи, які означають вираження емоцій, візуально відображає тональність, використання нейронних мереж для класифікації тональності);
- гнучке розгортання (програмний код оптимізовано для роботи із великими даними, можливість інтеграції у різні інфраструктури);
- підтримка 33 мов.

Аналіз тональності у даній системі використовує три класи: позитивний, негативний та нейтральний. Він реалізований за допомогою статистичних та лінгвістичних правил. Система працює із такими неструктурованими даними, як веб-сайти, соціальні медіа ресурси та інші ресурси, що містять текст. Можливе як виявлення тональності усього документа, так і окремо аспектів. Результати аналізу виводяться у звіти у вигляді діаграм. Можна також переглядати дані у динаміці в залежності від часу.

Інший продукт-аналог – Lithium (див. рис. 1.2). Дана система складається із трьох застосунків:

– Community Platform, який дозволяє моніторити розмови, що містять інформацію про бренд, під час дзвінків із клієнтами;
– Social Media Monitoring, що надає змогу моніторити повідомлення в Інтернеті, виявляти ставлення споживачів, знаходити спільноти, де бренд обговорюється найбільш активно, а також вилучати ідеї щодо покращення продукту;
– Customer Intelligence Center, який збирає дані про соціальну поведінку споживачів та формує її у профілі для бізнес-корпорацій, за допомогою яких можна підтримувати інтерес людей до бренду.

Рисунок 1.2 – Приклад роботи системи Lithium

Дана платформа використовує спеціальні словники, які відображають мовні моделі, вірогідності та зв’язки між об’єктами для роботи із сутностями [15]. Для

виявлених сутностей також розраховується їх важливість. Окрім цього, для аналізу використовуються окремі словники для виявлення тем та їх ієрархій, після чого текст відповідно хешується рекомендованими темами.

Аналіз тексту складається із наступних кроків:

- визначення мови;
- нормалізація тексту (очищення від символів, що не відображаються, заміщення не ASCII пунктуації);
- виділення речень за допомогою Java Text API;
- токенізація;
- виділення сутностей (використовуються словники та метод n-грам);
- дискримінація структури та зв'язків (у якості математичних методів для аналізу використовуються дерева рішень та лінійна регресія);
- визначення теми за допомогою векторизації та словників;
- хешування текстів рекомендованими темами;
- аналіз тональності, робота якого здійснюється за допомогою словників, що містять переліки слів, які класифікуються як позитивні чи негативні; тональність розраховується як різниця позитивної та негативної ваги, розділена на суму логарифму від загальної кількості слів;
- додавання до сутностей метаданих.

Sentimetrix – це ще одна система, призначення якої – автоматична обробка тексту в умовах «великих даних». До її функцій можна віднести автоматичний аналіз тональності, механізм пошуку емоцій (ідентифікує інтенсивність різних емоцій), аналітика соціальних даних (базується на аналізі новин та соціальних медіа) та аналітика соціальних медіа (виявляє активних користувачів, тренди, негативні відгуки щодо бренду та інше).

За семантичний аналіз у даній системі відповідає платформа SentiGrade. Вона має веб-інтерфейс та може візуалізувати результати. Платформа підтримує роботу на кількох мовах. Тональність може бути класифікована як позитивна, негативна чи нейтральна [16].

Однією із особливостей Sentimetrix є виконання аналізу емоційного забарвлення по відношенню до сутностей.

Приклад відображення статистичної інформації у системі наведений на рисунку 1.3.

Рисунок 1.3 – Приклад роботи Sentimetrix

Робота систематичного аналізатору можлива за допомогою наступних компонентів:

- компонент попередньої обробки тексту, який визначає мову, сегментує текст на параграфи та фрази, виконує токенізацію та нормалізацію та робить аналіз лексики (лематизація, виявлення стоп-слів);
- компонент виявлення іменованих сутностей (виявлення початкових слів-кандидатів, злиття та застосування правил класифікації відповідно до контексту, фінальна класифікація)
- компонент семантичного аналізу фрагментів, робота якого складається з таких етапів як виявлення локальних сигнальних слів, виявлення

модифікаторів (слів, що підсилюють емоцію, ознак заперечення) та підрахунок кінцевого рахунку для фрагменту, що аналізується.

У публікації «Sentiment Analysis of Twitter Data» висвітлюються результати дослідження аналізу тональності твітів [17]. Особливостю запропонованого аналізу є виконання семантичного аналізу окремих фраз, тональність яких формує тональність цілого твіту. Аналіз фраз відбувається за допомогою синтаксичного аналізу. Такий підхід дозволяє більш точно оцінювати тональність.

Загалом, аналіз тональності твітів широко використовується у різноманітних дослідженнях. Так, наприклад, у «Twitter as a Corpus for Sentiment Analysis and Opinion Mining» [18] на датасеті твітів у рамках дослідження було проведено будо проведено класифікацію текстів за допомогою наївного Байєсівського класифікатору, методів TreeTagger, POS-tags та n-грам. У результаті було отримано модель, що здатна виявляти позитивну, негативну та нейтральну тональність текстів. Особливістю моделі є можливість роботи із даними, що написані різними мовами.

2.4 Постановка задачі

У результаті аналізу предметної галузі було виявлено, що серед можливих варіантів використання семантичного аналізу доволі перспективним та новим напрямком є аналіз тональності, тому було прийнято рішення проводити дослідження у цьому напрямі.

Після аналізу існуючих підходів було вирішено зупинитися на більш сучасному підході, тобто на машинному навчанні, оскільки для нього не потрібні складні онтології та власноруч написані правила, що значно ускладнює реалізацію [19].

Більшість існуючих комерційних аналогів пропонує аналіз тональності у якості однієї із послуг. Відповідно до розглянутих публікацій, такі системи

пропонують один і той же алгоритм роботи для різних типів інформації. Одними із найбільш популярних об’єктів для визначення емоційного забарвлення є твіти та відгуки. Варто зазначити, що ці типи даних мають певні відмінності. Наприклад, твіти зазвичай виражають одну емоцію, тоді як відгуки часто можуть виражати і негативну, і позитивну, і нейтральну оцінку одночасно у різних частинах відгуку. До того ж, часто у кінці відгуку користувачі пишуть підсумок, який може потребувати особливої уваги при аналізі. Таким чином, відгуки є більш складним для аналізу типом даних.

Було прийнято рішення дослідити роботу різних методів для аналізу тональності для двох типів даних: твітів та відгуків. Гіпотеза полягає у тому, що для твітів буде достатньо «мішка слів». При обробці відгуків скоріше за все результати зможуть бути покращені при використанні більшого числа параметрів та врахування структури відгуків, скоріше за все у цьому випадку знадобиться біль складне векторне представлення, наприклад, Word2Vec, який може враховувати контекст. У рамках дослідження планується провести наступну роботу:

- дослідити роботу нейронних мереж на твітах та відгуках з точки зору точності результатів роботи алгоритму;
- дослідити роботу методу байєсівської класифікації як лінійного методу на твітах та відгуках та оцінити точність аналізу;
- оцінити якісь результатів при використанні «мішка слів» та Word2Vec для обох типів даних;
- дослідити особливості структури відгуків та спробувати врахувати це при аналізі тональності;
- зробити висновки щодо доцільності використання лінійних методів та нейронний мереж, а також методу та налаштувань векторизації при аналізі емоційного забарвлення твітів та відгуків.

При роботі планується використовувати готові розмічені датасети у форматі csv. Дослідження буде реалізовано мовою програмування Python 3 [20], буде використано інструмент для аналітичних звітів Jupyter Notebook, а також бібліотеки nltk, Pandas та Tensorflow.

2 ПРОЕКТУВАННЯ СИСТЕМИ ДЛЯ МОДЕЛЮВАННЯ

2.1 Моделювання роботи системи

На рисунку 2.1 зображено схему, що відображає порядок виконання аналізу емоційного забарвлення із його основними кроками, а також вхідними та вихідними даними:

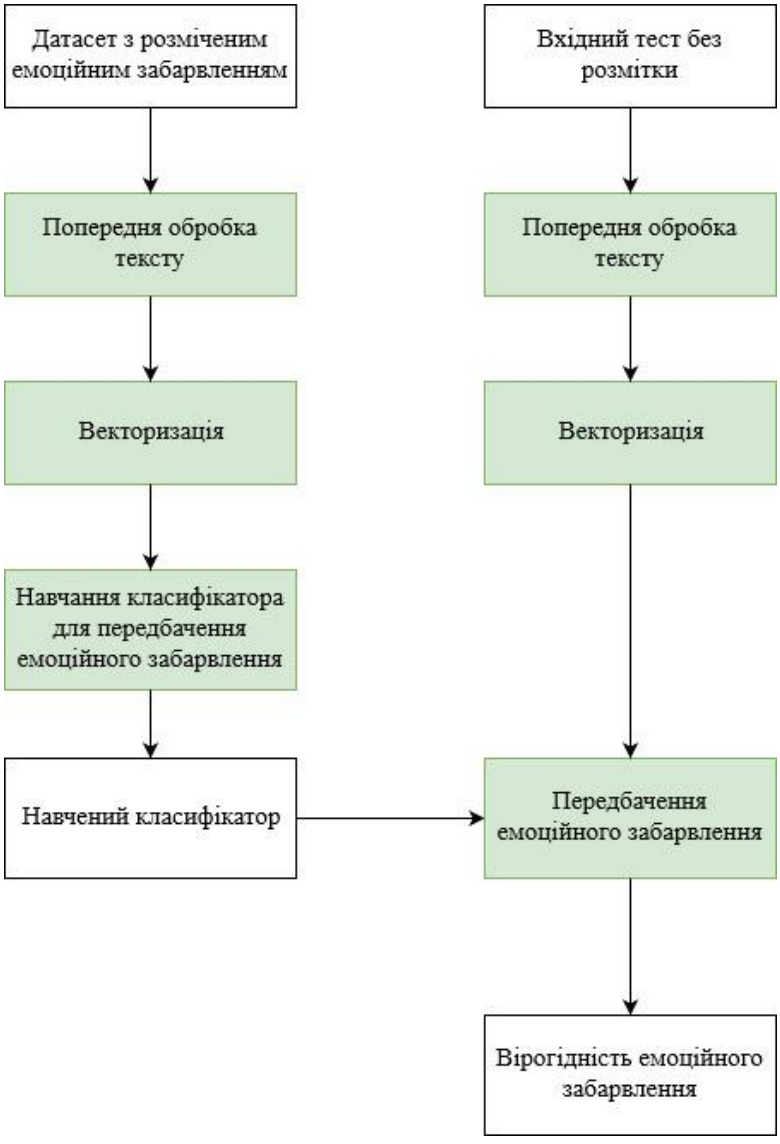

Рисунок 2.1 – Порядок виконання аналізу емоційного забарвлення тексту

Роботу системи загалом можна поділити на дві частини: навчання на розміченому тексті та виявлення тональності на робочих текстах.

На етапі навчання на вхід подається датасет із текстом, який вже розмічений відповідно до його емоційного забарвлення. Перш за все такий текст проходить попередню обробку аби зробити його придатним для подальшого аналізу. У цю обробку входить вилучення стоп-слів та стемінг.

Далі йде крок векторизації. У рамках дослідження планується виконати два варіанти векторизації: «мішок слів» та Word2Vec. Очікується, що для відгуків Word2Vec буде працювати краще, а для твітів буде достатньо «мішка слів».

«Мішок слів» передбачає просте виявлення унікальних слів та підрахунок їх кількості.

Word2Vec використовує більш складний підхід, зокрема враховує контекст. Реалізувати даний підхід допомагає багатовимірний векторний простір, де слова зі схожим контекстом розташовуються ближче одне до одного.

Після векторизації буде проведено навчання класифікатору. Планується дослідити два підходи, щоб зрозуміти, який з них краще використовувати для відгуків, а який – для твітів.

Перший підхід полягає у застосуванні нейронної мережі. Схема нейрону наведена на рисунку 2.2.

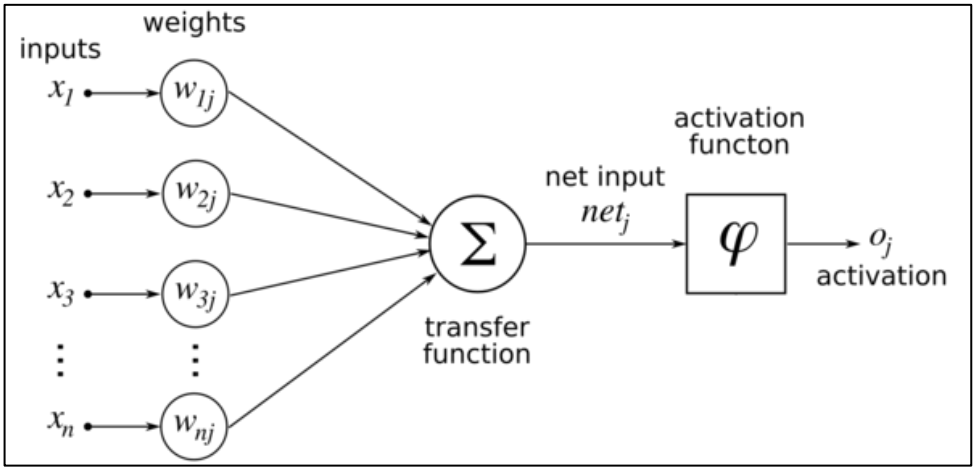

Рисунок 2.2 – Загальна схема нейрону

Нейрон складається з синапсів (що пов'язують входи нейрона з ядром), ядра нейрона (яке здійснює обробку вхідних сигналів) і аксона, який пов'язує нейрон з

нейронами наступного шару. Кожен синапс має вагу, який визначає, наскільки відповідний вхід нейрона впливає на його стан.

Функція трансформації S та функція активації φ мають наступний вигляд:

$$S = \sum_{i=1}^{n} x_i w_i, \tag{1}$$

$$\varphi = \frac{1}{1 + e^{-ax}}, \tag{2}$$

де n – число входів нейрона;

x_i – значення і-го входу нейрона;

w_i – вага і-го синапсу.

Для вирішення даної задачі було обрано нейрону мережу зворотного поширення. Така мережа складається з декількох шарів нейронів, причому кожен нейрон шару i пов'язаний з кожним нейроном шару $i + 1$ (повнозв'язкова мережа).

У загальному випадку задача навчання НС зводиться до знаходження функціональної залежності $Y = F(X)$, де X - вхідний, а Y - вихідний вектори. У загальному випадку така задача, при обмеженому наборі вхідних даних, має безліч рішень. Для обмеження простору пошуку при навчанні ставиться завдання мінімізації цільової функції помилки нейронної мережі, яка знаходиться за методом найменших квадратів:

$$E(w) = \frac{1}{2}\sum_{j=1}^{p}(y_i - d_i)^2, \tag{3}$$

де p – число нейронів на вихідному шарі;

y_j – значення j-го виходу мережі;

d_j – цільове значення j-го виходу.

Навчання нейронної мережі виконується методом градієнтного спуску, тобто на кожній ітерації зміна ваги здійснюється за формулою:

$$\Delta w_{ij} = -h\frac{\partial E}{\partial w_{ij}}, \tag{4}$$

де h – параметр який визначає швидкість навчання.

Другий варіант класифікації передбачає використання лінійних методів. Очікується, що вони краще спрацюють для коротких твітів.

У якості лінійного методу було обрано наївний байєсівський класифікатор. В основі наявного Байєсівського класифікатора лежить теорема Байєса:

$$P\ (A|\ B\ ) = \frac{P(B|A)P(A)}{P(B)} \tag{5}$$

Для даної моделі, документ – це вектор: d = {w_1, w_2, ..., w_n}, де w_i – вага i-ого терміну, а n – розмір словника вибірки. Таким чином, відповідно до теореми Байєса, ймовірність класу c для документа d буде:

$$P\ (c\ |\ d\ ) = \frac{P(c)P(d\ |\ c)}{P(d)} \tag{6}$$

Таким чином, обчислюється умовна ймовірність для всіх класів.

Найбільш ймовірний клас c*, якому належить документ d той, при якому умовна ймовірність приналежності документа d класу з максимальна:

$$c^* = arg\ max_c\ P\ (w_1, w_2\ , \dots , w_n\ \ |\ c) * P\ (c) \tag{7}$$

Після того, як класифікатор буде навчений, можна буде почати опрацьовувати робочу частину вибірки. Тексти без розмітки також мають пройти

попередню обробку і векторизацію. Після цього до текстів має бути застосований навчений класифікатор для визначення вірогідності емоційного забарвлення.

Логістична регресія – це статистична модель, що використовується для передбачення ймовірності виникнення деякої події шляхом підгонки даних до логістичної кривої. Логістична функція має вигляд:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

Логістична регресія застосовується для передбачення ймовірності виникнення деякої події за значеннями множини ознак. Для цього вводиться так звана залежна змінна y, що приймає лише одне з двох значень – як правило, це числа 0 (подія не відбулося) і 1 (подія відбулася), і безліч незалежних змінних (також званих ознаками) – дійсне $x_1, x_2, \dots x_n$, на основі значень яких потрібно обчислити ймовірність прийняття того чи іншого значення залежної змінної.

Робиться припущення про те, що ймовірність настання події $y = 1$ дорівнює:

$$P_r(y = 1|x) = f(z), \tag{9}$$

де x – вектор-стовпець значень незалежних змінних $x_1, \dots, x_n$;

$f(z)$ – логістична функція;

z – вектор стовпець значень параметрів (коефіцієнтів регресії) $\theta_1, \dots, \theta_n$ у степені Т.

Формула логістичної функції:

$$f(z) = \frac{1}{1 + e^{-z}} \tag{10}$$

Для того, щоб підібрати параметри $\theta_1, \dots, \theta_n$, необхідно скласти навчальну вибірку, що складається з наборів значень незалежних змінних та відповідних їм

значень змінної y. Формально, це множина пар $\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(m)}, y^{(m)}\right)$, де $x^{(i)} \in R^n$ – вектор значень незалежних змінних, а $y^{(i)} \in \{0,1\}$ – відповідне йому значення у. Кожна пара називається навчальним прикладом.

Зазвичай використовується метод максимальної правдоподібності, відповідно до якого вибираються параметри θ, що максимізують значення функцій правдоподібності на навчальній виборці:

$$\hat{\theta} = argmax_\theta L(\theta) = argmax_\theta \prod_{i=1}^{m} P_r(y = y^{(i)} | x = x^{(i)}) \quad (11)$$

Максимізація функції правдоподібності еквівалентна максимізації її логарифма:

$$\log L(\theta) = \sum_{i=1}^{m} \log P_r(y = y^{(i)} | x = x^{(i)}) = \sum_{i=1}^{m} y^{(i)} \log f(\theta^T x^{(i)}) + (1 - y^{(i)}) \quad (12)$$

Для максимізації цієї функції може бути застосований, наприклад, метод градієнтного спуску. Він полягає у виконанні наступних ітерацій, починаючи з деякого початкового значення параметрів θ:

$$\theta \coloneqq \theta + a \nabla \log L(\theta) = \theta + a \sum_{i=1}^{m} \left(y^{(i)} - f\left(\theta^T x^{(i)}\right)\right) x^{(i)}, a > 0 \quad (13)$$

На практиці також застосовують метод Ньютона і стохастичний градієнтний спуск.

При роботі з моделями класифікацій необхідно мати спосіб оцінювати їх якість. Це дає змогу зрозуміти які зміни в алгоритмі роблять його краще, а які зменшують якість класифікації.

Основою перевірки є тестова вибірка в якій вказано відповідність між документами і їх класами. Маючи такі данні достатньо застосувати класифікатор

до документів і співвіднести його рішення зі відомим правильним рішенням. Але для того, щоб визначати як нова версія алгоритму справляється з роботою, необхідна чисельна метрика його якості. Одним з варіантів такої метрики є точність (accuracy) – частка документів за якими класифікатор прийняв правильне рішення:

$$Accuracy = \frac{P}{N}, \tag{14}$$

де P – кількість документів за якими класифікатор прийняв правильне рішення;

N – розмір навчальної вибірки.

Проте, ця метрика має одна особливість, яку необхідно враховувати. Вона привласнює всім документам однакову вагу, що може бути не коректним, якщо розподіл документів в навчальній вибірці сильно зміщений в бік одного або декількох класів. В цьому випадку у класифікатора є більше інформації по цих класах і відповідно в рамках цих класів він буде приймати більш адекватні рішення.

Один вихід з цієї ситуації полягає в тому щоб навчати класифікатор на спеціально підготовленому, збалансованому корпусі документів. Мінус цього рішення в тому що ви відбираєте у класифікатора інформацію про відносну частоті документів. Ця інформація при інших рівних може виявитися дуже корисною для прийняття правильного рішення.

Інший вихід полягає в зміні підходу до формальної оцінці якості. Точність (precision) і повнота (recall) є метриками які використовуються при оцінці більшості алгоритмів вилучення інформації. Іноді вони використовуються самі по собі, іноді в якості базису для похідних метрик.

Точність системи в межах класу – це частка документів які дійсно належать даному класу відносно всіх документів, які система віднесла до цього класу. Повнота системи – це частка знайдених класифікатором документів, що належать класу, відносно всіх документів цього класу в тестовій вибірці. Ці значення легко розрахувати використовуючи таблицю контингентності, яка складається для

кожного класу окремо. У таблиці міститься інформація скільки разів система прийняла вірне і скільки разів невірне рішення за документами заданого класу.

Точність і повнота визначаються наступним чином:

$$Precision = \frac{TP}{TP + FP}, \tag{15}$$

$$Recall = \frac{TP}{TP + FN}, \tag{16}$$

де TP – істино-позитивне рішення;

FP – хибно-позитивного рішення;

FN – хибно-негативне рішення.

Зрозуміло, що чим вище точність і повнота, тим краще. Але в реальному житті максимальна точність і повнота недосяжні одночасно і доводиться шукати баланс. Тому бажано мати певну метрику, яка об'єднувала б у собі інформацію щодо точності та повноти обраного алгоритму. У цьому випадку буде простіше приймати рішення про те, яку реалізацію використовувати. Саме такий метрикою є F-міра.

F-міра – це гармонійне середнє між точністю і повнотою. Вона наближається до нуля, якщо точність або наближається до нуля:

$$F = 2\frac{Precision * Recall}{Precision + Recall} \tag{17}$$

Дана формула надає однакову вагу точності і повноти, тому F-міра буде зменшуватись однаково при зменшенні і точності, і повноти. Можна розрахувати F-міру надавши різну вагу для точності і повноти, якщо ви потрібно віддати пріоритет одній з цих метрик при розробці алгоритму.

Таким чином, F-міра, Recall та Precision є головними метриками при оцінці адекватності моделі.

2.2 Інструменти та засоби реалізації

Виконання даного дослідження було обрано мову програмування Python . Це високорівнева мова, що проста для розробки та читання коду. Python є однією з найпопулярніших мов для машинного навчання та аналізу даних, для якої написано чимало фреймворків та бібліотек, що полегшують процес розробки [21].

Оскільки для навчання класифікатору буде використовуватися нейронна мережа, планується застосувати фреймворк TensorFlow. Він являє собою потужну бібліотеку для машинного навчання, яка виконує велику кількість рутинних завдань по побудові нейронних мереж замість розробника. TensorFlow також дозволяє відстежувати прогрес навчання за допомогою графіків.

Перед тим, як застосовувати перераховані вище інструменти, необхідно попередньо опрацювати датасети, аби привести їх до необхідного формату. Щоб зробити це із мінімальними зусиллями, буде використана ще одна бібліотека – NTLK (Natural Language Toolkit). Вона призначена для символічної та статистичної обробки тексту. З її допомогою можна реалізовувати роботу із морфологією, парсингом, тегуванням, токенізацією та класифікацією.

Як середовище розробки буде використано PyCharm, який дозволяє оптимізувати рутинну роботу із кодом. Окрім цього, початкові етапах роботи будуть виконуватися у Jupyter Notebook, який являє собою веб-додаток, що дозволяє працювати із «живим» кодом, візуалізаціями та текстом, формувати із цього документи, якими потім можна обмінюватися.

2.3 Інтерфейс користувача

У межах проектування було розроблено прототип інтерфейсу користувача, за допомогою якого він буде взаємодіяти із системою [22]. Інтерфейс системи

передбачає виконання як аналізу відгуків, так і аналізу твітів. За замовчуванням першою буде сторінка відгуків (див. рис. 2.3).

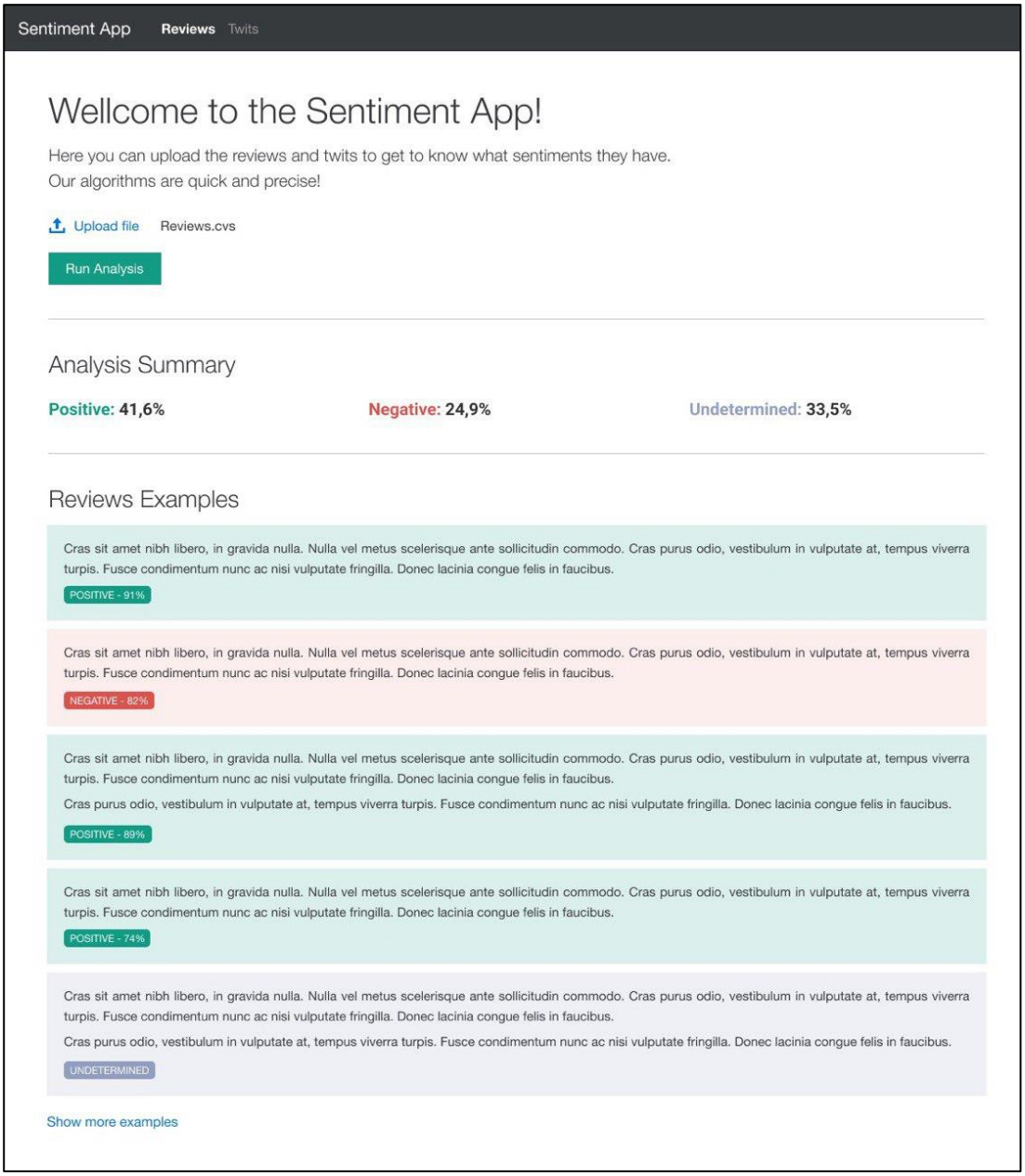

Рисунок 2.3 – Сторінка аналізу відгуків

У першу чергу користувач бачитиме привітання та короткий опис того, що він може аналізувати у системі. Одразу після цієї інформації на сторінці відгуків буде кнопка для завантаження файлів. Файли можна буде завантажувати у форматі csv. Після того, як файл буде завантажений, назва файлу з’явиться праворуч від кнопки. Далі користувач зможе натиснути на кнопку запуску аналізу.

Перехід на сторінку аналізу твітів (див. рис. 2.4) відбуватиметься у навігаційному меню.

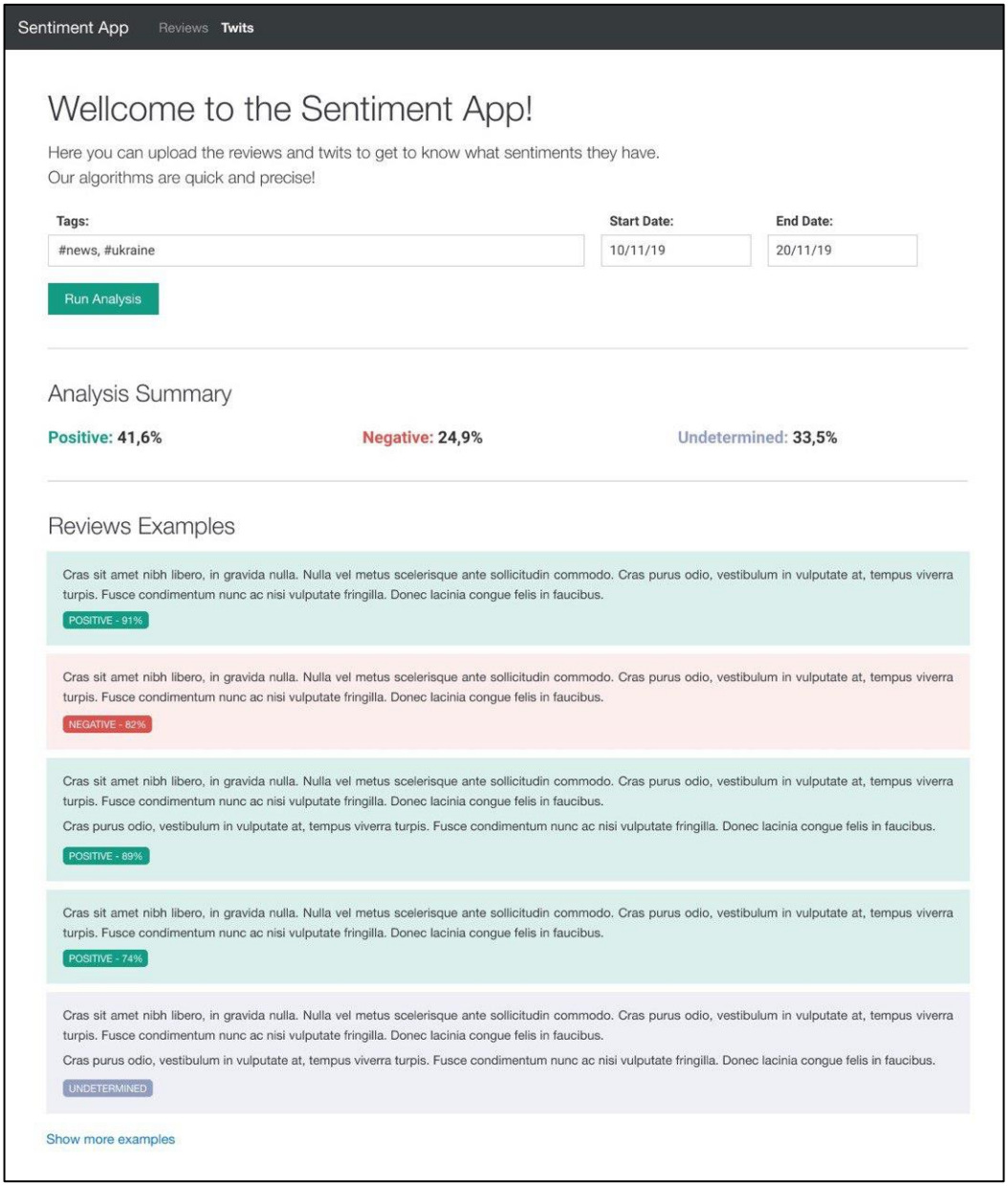

Рисунок 2.4 – Сторінка аналізу твітів

Очікується, що твіти будуть завантажуватися із Твіттеру, тому на відміну від сторінки відгуків, де дані завантажувалися вручну користувачем у вигляді файлу, на сторінці твітів будуть три поля для введення: поле для тегів (через кому будуть вводитися бажані теги, які будуть сприйматися пошуком, як об’єднані операцією логічного «І»), дата початку та кінця періоду, за який буде відбуватися пошук тегів.

Усі поля будуть обов’язковими. Після введення необхідних даних, користувач зможе натиснути кнопку початку аналізу.

Результати аналізу будуть представлені однаково як для відгуків, так і для твітів, та складатися із двох розділів. У першому розділі буде відображена загальна інформація: співвідношення позитивних, негативних та неоднозначних текстів у відсотках. Для спрощення сприйняття мітки «Positive», «Negative» та «Undetermined» будуть відповідних кольорів: зеленого, черевного та сірого [23]. У другому блоці будуть відображені приклади проаналізованих текстів, колір фону яких відповідатиме їх тональності. Враховуючи, що деякі користувачі можуть мати порушення сприйняття кольору, було прийнято рішення не спиратися тільки на фон, а додати ще кольорову мітку із текстом (найбільш імовірною тональністю). Окрім тональності, у цій мітці також відображатиметься і вірогідність тональності у відсотках.

У розділі із прикладами за замовчування буде виводитися лише п’ять текстів. Щоб побачити інші відгуки, користувач зможе натиснути на кнопку «Показати більше прикладів».

При підборі кольорів для системи, буде братися до уваги контрастність тексту для забезпечення легкого читання на моніторах будь-якої якості та при різних налаштуваннях.

Варто зазначити, що даний інтерфейс розроблений тільки для демонстрації роботи програмного забезпечення.

3 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

У межах дослідження було прийнято рішення проаналізувати роботу двох підходів векторизації («мішок слів» та Word2Vec) та двох підходів класифікації (лінійні моделі та рекурентні нейронні мережі, а саме нейронна мережа довгої короткочасної пам’яті). Дослідження проводилося на двох різних датасетах: твіти та відгуки – найбільш популярні типи даних, на яких зазвичай відбувається аналіз емоційного забарвлення. Датасет твітів складався із 1.6 мільйонів твітів різної тематики, а датасет відгуків – із 3 мільйонів відгуків щодо товарів інтернет-магазину Amazon.

3.1 Аналіз роботи «мішку слів» та лінійних моделей

Ряд перших досліджень базувався на використанні «мішка слів» для векторизації, а також кількох лінійних моделей: метод логістичної регресії, наївний Байєсівський класифікатор, Байєсівський класифікатор із розподіленням Бернулі та Байєсівський класифікатор із мультиноміальний розподіленням. Очікувалося, що «мішка слів» має бути достатньо для коротких текстів, наприклад, таких, як зазвичай зустрічаються у твітах. Також була висунута гіпотеза, що лінійні моделі краще спрацюють на невеликих обсягах даних для навчання.

При реалізації методу «мішка слів» був також застосований метод n-грам. Це дозволяє підраховувати не тільки окремі слова, а ще й у деяких випадках пари слів, свого роду словосполуки, які часто використовуються разом, а отже можуть відігравати велику роль при аналізі тональності текстів. При використанні цього методу можна вказувати та аналізувати словосполуки із різної кількості слів; для дослідження вилучалися словосполуки тільки із двох слів, оскільки вони є найбільш поширеними.

Перед перетворення тексту до формату «мішка слів» із текстів було вилучено стоп-слова (за виключенням слова «not», так як воно зазвичай виражає протилежну тональність).

3.1.1 Аналіз роботи методу логістичної регресії

Датасет було поділено на дві частини у відношенні 4 до 1. 80% текстів були використанні для навчання (навчаюча вибірка), а 20% було залишено для перевірки роботи моделі (контрольна вибірка). Під час дослідження було проведено навчання на датасетах різного розміру (10 000 та 10 000 текстів) як твітів, так і відгуків.

На рисунок 3.1 наведено приклад класифікації за методом логістичної регресії.

	feature	coef
38878	sad	-39.749050
31791	not happy	-34.701381
10009	depressed	-32.154355
5598	bummed	-30.570560
43008	starving	-30.226482
...	...	...
833	almost forgot	23.972461
7905	congratulations	24.417872
10811	don forget	25.422896
27304	make wish	28.890326
32145	nothing wrong	28.988137

Рисунок 3.1 – Приклад роботи методу логістичної регресії

Перший стовпець означає порядковий номер слова чи словосполуки та не несе важливої інформації. У стовпцях «feature» та «coef» можна побачити слова чи

словосполучення векторів та їх додатну або від'ємну вагу (що означають позитивне чи негативне емоційне забарвлення відповідно), які були визначені у результаті роботи логістичної регресії. Так, наприклад, слово «sad» має доволі сильне негативне забарвлення, тоді як «make wish» несе хоч і дещо менше за силою, але позитивне забарвлення.

Для того щоб більш точно та об'єктивно оцінити роботу класифікатору, були підраховані метрики якості моделі на навчальній вибірці у 100 000 твітів (див. рис. 3.2).

LogisticRegression:

	precision	recall	f1-score
positive	0.73	0.71	0.72
negative	0.72	0.74	0.73

Рисунок 3.2 – Метрики моделі логістичною регресії із навчаючою вибіркою 100 000 твітів

Значення точності, повноти та f-міри знаходяться у діапазонах 0.71-0.74 та є доволі високими. Далі було проведено аналогічний експеримент, але із навчаючою вибіркою у 10 разів меншою (див. рис. 3.3).

LogisticRegression:

	precision	recall	f1-score
positive	0.67	0.67	0.67
negative	0.67	0.67	0.67

Рисунок 3.3 – Метрики моделі логістичною регресії із навчаючою вибіркою 100 00 твітів

Із рисунку 3.3 видно, що усі три показники впали на 4-7% після зменшення розміру навчаючої вибірки.

Далі було проведено аналогічні експерименти, використовуючи датасет відгуків. При аналізі результатів виявилося, що показники якості моделі були підвищені на 13% на датасеті із 100 000 відгуків та приблизно на 10% на датасеті із 10 000 відгуків у порівнянні із твітами. Скоріше за все, це пов’язано із довжиною текстів. Деякого покращення результатів вдалося досягти при аналізі текстів без видалених стоп-слів – показники вдалося покращити ще на 2%. Це дало змогу висунути ще одну гіпотезу – залишення стоп-слів може покращити результати аналізу тональності.

На рисунку 3.4 зображено приклади відгуків та вірогідності їх відношення до позитивного чи негативного класу.

```
test_sample(logreg, "The product was good and easy to  use")
test_sample(logreg, "the whole experience was horrible and product is worst")
test_sample(logreg, "product is not good")

Sample estimated as POS: negative prob 0.011110, positive prob 0.988890
Sample estimated as NEG: negative prob 1.000000, positive prob 0.000000
Sample estimated as NEG: negative prob 0.999997, positive prob 0.000003
```

Рисунок 3.4 – Приклад оцінки вірогідності класів текстів за методом логістичної регресії

Як можна помітити, метод доволі добре класифікує однозначні короткі висловлювання, тож однозначно може бути використаний для систем аналізу емоційного забарвлення.

3.1.2 Аналіз роботи наївного Байєсівського класифікатору

Для дослідження роботи Байєсівського класифікатору було проведені такі ж експерименти, як і для випадку із логістичною регресією. Результати експерименту із наївним Байєсівським класифікатором наведені на рисунку 3.5.

Точність роботи цього класифікатору, навченого на датасеті із 100 000 твітів була практично такою ж, як і у логістичної регресії.

```
NLTK Naive bayes Accuracy : 0.741375
Most Informative Features
                 hurting = True              neg : pos    =     28.5 : 1.0
               cancelled = True              neg : pos    =     25.6 : 1.0
                   upset = True              neg : pos    =     22.0 : 1.0
              frustrated = True              neg : pos    =     19.8 : 1.0
                thankyou = True              pos : neg    =     18.9 : 1.0
```

Рисунок 3.5 – Результати роботи Байєсівського класифікатору (100 000 твітів)

Можна одразу помітити деякі слова із сильним емоційним забарвленням. Так, наприклад, слово «hurting» у 28 разів частіше має негативне емоційне забарвлення. Тоді як словосполучення «thankyou» у 19 разів частіше має позитивну тональність.

На рисунку 3.6 наведені результати класифікації із навчанням на 10 000 твітів.

```
NLTK Naive bayes Accuracy : 0.710625
Most Informative Features
                    sad = True              neg : pos    =     19.9 : 1.0
                   hurt = True              neg : pos    =     15.4 : 1.0
                    ugh = True              neg : pos    =     13.1 : 1.0
               computer = True              neg : pos    =     12.0 : 1.0
                 scared = True              neg : pos    =     10.6 : 1.0
```

Рисунок 3.6 - Результати роботи Байєсівського класифікатору (10 000 твітів)

Одразу помітно, що точність аналізу знизилася на 3% у порівнянням із навчальною вибіркою у 100 000 текстів, але вона все ж на 4% вища, аніж в аналогічному експерименті із логістичною регресією. Окрім цього, при навчанні на меншій вибірці класифікатор виділив інші слова із великою силою. Цікаво і те, що сила слів у півтора-два рази нижча, аніж у випадку із більшим датасетом.

Такі ж експерименти були проведені і для датасету з відгуками.

Точність результатів залишилася приблизно такою ж, як і у випадку із такою ж кількістю твітів, але сила емоційно забарвлених слів зросла у півтори рази (див. рис. 3.7).

```
NLTK Naive bayes Accuracy : 0.7488125
Most Informative Features
              uninspired = True              neg : pos    =     37.6 : 1.0
                  refund = True              neg : pos    =     27.1 : 1.0
              unoriginal = True              neg : pos    =     26.9 : 1.0
               miserably = True              neg : pos    =     26.3 : 1.0
                 unusable = True              neg : pos    =     25.5 : 1.0
```

Рисунок 3.7 – Результати роботи Байєсівського класифікатору (100 000 відгуків)

Точність результатів при навчальній вибірці у 10 000 відгуків становила 70%, а сила слів (перших п’яти) була хоч і нижчою, аніж у експериментів із вибіркою у 100 000 текстів, але вищою, ніж при вибірці із 100 000 твітів (15.5 – 36.6). Це дозволяє припустити, що значення емоційно забарвлених слів більше залежить від загального обсягу текстів у навчальній вибірці, ніж від кількості документів у ній.

Загалом, можна сказати, що аналіз із застосуванням наївного Байєсівського класифікатору працює дещо краще, ніж метод логістичної регресії.

3.1.3 Аналіз роботи Байєсівського класифікатору із мультиноміальний розподіленням

Байєсівський класифікатор із мультиноміальнім розподіленням вважаться покращенням звичайного наївного підходу, оскільки останній робить припущення, яке дуже рідко зустрічається на практиці: між параметрами немає зв’язку. У випадку із аналізом тексту, параметри – це слова, тож відповідно наївному підходу поява одного слова не залежить від появи іншого. Але це не так. Мультиноміальне

розподілення частково вирішує цю проблему, тож очікувалося, що він буде працювати краще.

На рисунку 3.8 зображені результати оцінки роботи моделі із використанням Байєсівського класифікатору із мультиноміальним розподілом.

Multinomial:

	precision	recall	f1-score
positive	0.76	0.76	0.76
negative	0.76	0.77	0.76
accuracy			0.76
macro avg	0.76	0.76	0.76
weighted avg	0.76	0.76	0.76

Рисунок 3.8 – Оцінка роботи моделі Байєсівського класифікатору із мультиноміальним розподілом (100 000 твітів)

Помітно, що Байєсівський класифікатор із мультиноміальним розподілом, навчений на 100 000 твітів має показники якості на 3% вищі, ніж логістична регресія. Приблизно на стільки ж краще працює модель, навчена на 10 000 твітів.

Показники якості аналізу відгуків (100 000 відгуків) лежать у межах 83 – 86, що практично не відрізняється від результатів роботи логістичної регресії. При навчанні на 10 000 відгуків показники якості даного методу також не мають суттєвих відмінностей. А ось залишення стоп-слів практично не змінило результати, на відміну від логістичної регресії.

3.1.4 Аналіз роботи Байєсівського класифікатору із розподілом Бернулі

Ще одним варіантом покращення наївного підходу є Байєсівський класифікатор із розподілом Бернулі. Результати досліджень показали, що суттєвого покращення у порівнянні із аналогічним методом із мультиноміальним розподілом досягти не вдається.

Цікавою особливістю, поміченою під час аналізу показників якості було те, що різниця у значеннях різних метрик якості були більш відмінні між собою, аніж у випадку попередніх методів (див. рис. 3.9). Більша різниця була і в оцінках позитивно так негативно класифікованих текстів.

Bernoulli:

	precision	recall	f1-score
positive	0.85	0.78	0.81
negative	0.79	0.86	0.83
accuracy			0.82
macro avg	0.82	0.82	0.82
weighted avg	0.82	0.82	0.82

Рисунок 3.9 – Метрики моделі Байєсівського класифікатору із розподілом Бернулі

Загалом, усі досліджені методи показали доволі точні результати. Приклад визначення емоційного забарвлення наведений на рисунку 3.10.

reviews.text	senti
I got one of these bracelets for my boyfriend....	neg
The music and reading is fun, but it is way to...	neg
The sound quality is pretty amazing considerin...	pos
This movie is GOOD. NOT Jason but looks and ac...	pos
The movie was awsome! I went to buy the soundt...	neg
...	...
I purchased this software to use with my young...	neg
For a high-powered big-city lawyer, Ellie was ...	neg
We love this mailbox! The style goes with our ...	pos
this game is cool.from the martial arts moves ...	pos
Having pulled out most of my remaining hair on...	pos

Рисунок 3.10 – Приклад класифікації текстів за тональністю

Також були спроби змінити кількість слів у n-грамах. При збільшенні кількості слів до 3, спостерігалося незначне покращення показників у випадку із

логістичною регресією та Байєсівським класифікатором із розподілом Бернулі у середньому на 2% – 3%.

3.2 Аналіз роботи Word2Vec та нейронна мережа довгої короткочасної пам’яті

Перед початком дослідження було висунуто гіпотезу, що нейронна мережа із Word2Vec має спрацювати краще на відгуках, оскільки вони можуть містити більше висловлювань, тональність яких відрізняється, а також мають більшу довжину, аніж твіти.

В основу даної гіпотези лягло те, що коли людина аналізує ставлення автору до певного об’єкту, вона аналізує як саме слово, так і його контекст. Контекст впливає на забарвлення. Бібліотека Word2Vec враховує контекст слова, розташовуючи слова зі схожим контекстом ближче до слова, що аналізується, таким чином це покращує «розуміння» змісту. Ще одним фактором, який може покращити розуміння змісту тексту є послідовність слів. І особливістю рекурентних нейронних мереж є те, що вони враховують її у тексті.

Для забезпечення роботи моделі датасет було так само, як і у випадку із лінійними моделями, розбито на дві частини: навчальну та контрольні вибірки у тому ж співвідношенні.

Спочатку навчання Word2Vec відбувалося на датасеті із 100 000 твітів. Для навчання було встановлено 32 епохи (проходи по колекції).

Для перевірки роботи навченого Word2Vec були перевірені близькі вектори для певних слів. Припущення, що близькі вектори мають схожий контекст, дозволило нашвидкуруч оцінити «навченість». Наприклад, до контексту слова «love» належали наступні слова (через тире вказана міра близькості):

- 'luv' – 0.5732780694961548;
- 'loves' – 0.5623787045478821;

- 'loved' – 0.5373271703720093;
- 'amazing' – 0.5026600360870361;
- 'adore' – 0.4942743480205536;
- 'looove' – 0.47235167026519775;
- 'awesome' – 0.4598265290260315;
- 'lovee' – 0.45823752880096436;
- 'loveee' – 0.4531649351119995;
- 'loooove' – 0.44260522723197937.

Варто зазначити, що «навчений» Word2Vec дозволив не лише виявити синонімічні слова, але і сленг та неправильне написання слова «love».

Дослідження роботи поєднання Word2Vec із нейронною мережею було розпочато з попередньої обробки тексту: були видалені зайві символи, проведено токенізацію, видалення стоп-слів та інше. Далі кожне слово було перетворене у вектор.

Цікаво, що застосування стемінгу не змінило результати. Вірогідно, це пов’язано із тим, що у випадку із обраним методом векторизації, частота слів не підраховується, як це відбувається у випадку із підходом «мішка слів». Таким чином, було зроблено висновок, що застосування стемінгу у поєднанні з Word2Vec не має сенсу.

Далі попередньо опрацьовані дані передаються на вхід рекурентної нейронної мережі. Проаналізувавши графік навчання нейронної мережі (див. рис. 3.10), було прийнято рішення зупинитися на восьми епохах, оскільки подальше навчання не збільшує точність.

У якості вихідних даних нейронна мережа надає число від 0 до 1. Для покращення точності класифікації було виділено так звану сіру зону, яка лежить у межах від 0.4 до 0.7. Тексти, оцінка яких потрапляє у цю зону, вважаються нейтральними. Оцінка, що вища за ці значення, інтерпретується як позитивне емоційне забарвлення, нижча – як негативне. Відповідно до першого досліду, висловлювання «I love the music» було інтерпретоване як позитивне із оцінкою

0.97, «I hate the rain» та «I don’t know what I’m doing» – як негативне із оцінками 0.01 та 0.27 відповідно.

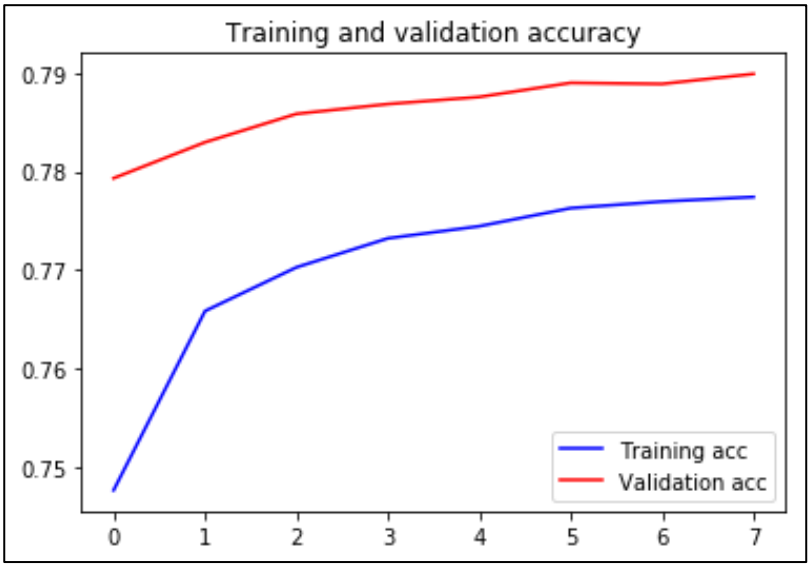

Рисунок 3.10 – Графік навчання нейронної мережі на датасеті із 100 000 твітів

Після навчання моделі та тестування її роботи були проаналізовані метрики її якості (див. рис. 3.11).

	precision	recall	f1-score
NEGATIVE	0.79	0.79	0.79
POSITIVE	0.79	0.80	0.79
micro avg	0.79	0.79	0.79
macro avg	0.79	0.79	0.79
weighted avg	0.79	0.79	0.79

Рисунок 3.11 – Оцінка якості роботи нейронної мережі, навченої на 100 000 твітах

Точність моделі становить 79% і це на 2% більше, аніж у випадку із лінійними моделями.

Наступним експериментом було навчання моделі на 10 000 твітів. Результати векторизації були набагато гіршими, аніж при 100 000 текстів. Так, тепер найближчими векторами слова «love» були наступні слова:

- 'hi' – 0.9325231909751892;
- 'thank' – 0.9258179664611816;
- 'thx' – 0.9088349342346191;
- 'hun' – 0.9048795104026794;
- 'kind' – 0.8882864713668823;
- 'tweets' – 0.8815054297447205;
- 'p' – 0.8811770081520081;
- 'follow' – 0.8776755332946777;
- 'funny' – 0.8744148015975952;
- 'followfriday' – 0.872011542320251.

Ці слова мало описують зміст слова «love». Графік навчання моделі також мав зовсім інший вигляд та нижчі показники точності (див. рис. 3.12).

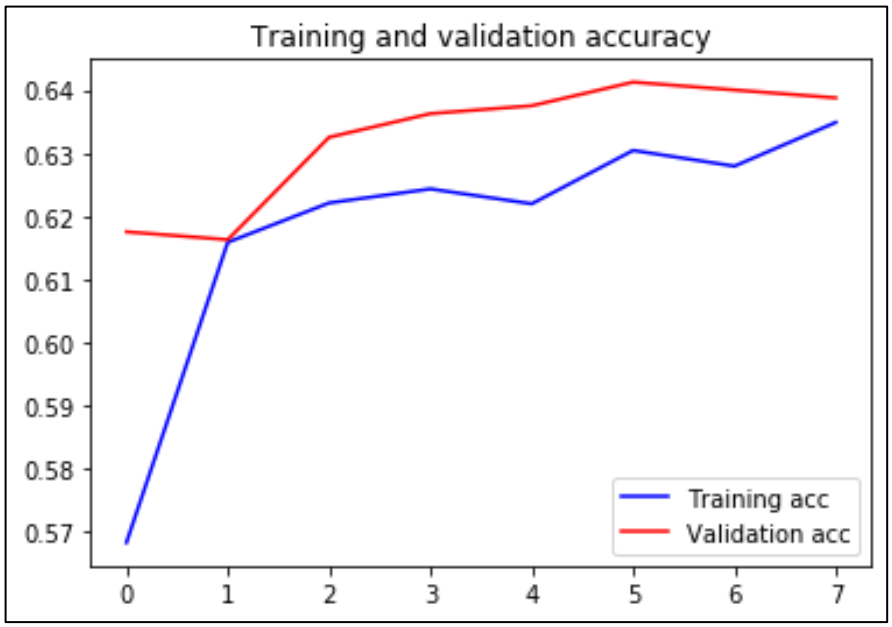

Рисунок 3.12 – Графік навчання нейронної мережі на датасеті із 10 000 твітів

Аналіз точності моделі показав, що вона суттєво знизилася у порівнянні із моделлю, навченою на у 10 разів більшому датасеті, та становила лише 62%.

Незадовільними виявилися і результати тестування роботи: тепер висловлювання «I love music» було інтерпретовано як нейтральне із оцінкою 0.67, так само, як і «I don’t know what I’m doing» (0.51), фраза «I hate the rain» хоч і була

інтерпретована як негативна, її оцінка була підвищена до 0.39, що дуже наближало її до нейтральної.

Таким чином, було виявлено, що моделі, які використовують нейронні мережі, є дуже чутливими до розміру датасету, на якому проводиться на навчання. Тож, коли мова заходить про невелику кількість даних, краще зробити вибір на користь лінійних класифікаторів.

У зв'язку із тим, що при навчанні моделі на невеликій вибірці даних проблеми починають виникати вже на етапі векторизації, було висунуто припущення, що використання вже навченого Word2Vec може покращити результат навчання нейронних мереж на малих датасетах.

У якості навченого Word2Vec було використано попередньо навчений Word2Vec від Google. Графік навчання при використанні цієї моделі значно покращився (див. рис. 3.13).

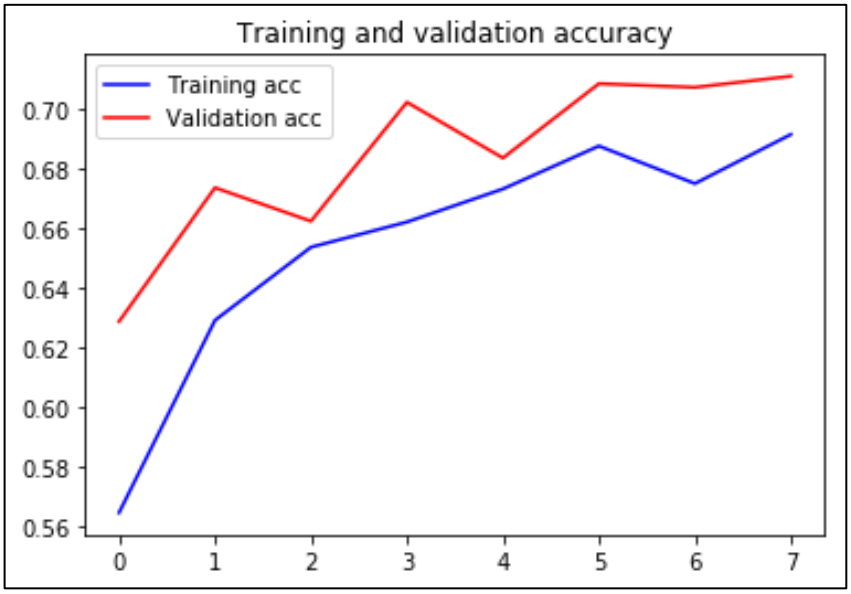

Рисунок 3.13 – Графік навчання нейронної мережі із використанням попередньо навченого Word2Vec (100 000 твітів)

Тестування роботи класифікатору також показало значно покращені результати у порівнянні із власноручним навчанням моделі. Текст «I love the music» було визначено як позитивний із оцінкою 0.72, «I hate the rain» - як негативний із оцінкою 0.26, а «I don't know what I'm doing» - нейтральним (0.48).

Аналіз якості роботи класифікатору також показав значні покращення показників – точність завдяки використанню попередньо навченого Word2Vec вдалося підвищити до 0.70, тобто на 8% (див. рис. 3.14).

Отож, можна зробити висновок, що на маленьких датасетах краще використовувати попередньо навчений Word2Vec.

	precision	recall	f1-score
NEGATIVE	0.69	0.71	0.70
POSITIVE	0.71	0.69	0.70
micro avg	0.70	0.70	0.70
macro avg	0.70	0.70	0.70
weighted avg	0.70	0.70	0.70

Рисунок 3.14 – Оцінка якості роботи нейронної мережі із використанням попередньо навченої моделі (10 000 твітів)

Попередньо навчений Word2Vec давав кращі результати при перевірці близьких слів. Для «love» були запропоновані наступні близькі слова:

– 'loved' – 0.6907792091369629;
– 'adore' – 0.6816873550415039;
– 'loves' – 0.6618633270263672;
– 'passion' – 0.6100709438323975;
– 'hate' – 0.600395679473877;
– 'loving' – 0.5886635780334473;
– 'Ilove' – 0.5702950954437256;
– 'affection' – 0.5664337873458862);
– 'undying_love' – 0.5547305345535278;
– 'absolutely_adore' – 0.5536839962005615.

Спираюсь на результати останнього експерименту, було вирішено перевірити якість роботи нейронної мережі із попередньо навченим Word2Vec на датасеті із 100 000 твітів.

Результати аналіз якості роботи моделі показали, що точність моделі падає у порівнянні із аналогічною нейронною мережею та Word2Vec, який було навчено самостійно (див. рис. 3.15). Відповідно до зібраних метрик точність впала приблизно на 1% і становила 78%.

	precision	recall	f1-score
NEGATIVE	0.79	0.77	0.78
POSITIVE	0.78	0.80	0.79
micro avg	0.78	0.78	0.78
macro avg	0.78	0.78	0.78
weighted avg	0.78	0.78	0.78

Рисунок 3.15 – Оцінка якості роботи нейронної мережі із використанням попередньо навченого Word2Vec (100 000 твітів)

Було зроблено висновок, що це пов’язано із тим, що різні домени використовують слова по-різному, завдяки чому навчання стає більш ефективним та дає кращі результати на подібних текстах.

Наступним кроком було вирішено перевірити роботу Word2Vec із рекурентною нейронною мережею для відгуків.

Спочатку було проведено аналіз тональності із 100 000 відгуками та самостійно навченою моделлю Word2Vec. Точність роботи класифікатору була вищою на 5%, аніж при аналогічному аналізі датасету із твітами. При тестуванні роботи висловлювання «I love the music» було інтерпретовано як позитивне із оцінкою 0.92, «I hate the rain» – як негативне (0.24), «I don’t know what I’m doing» – як нейтральне (0.49). Точність при цьому була доволі високою – 85%.

Підсумовуючи результати цього експерименту, можна сказати, що при однаковій кількості текстів на відгуках модель навчається дещо краще.

Далі було проведено аналогічний експеримент з датасетом у 10 000 відгуків. При тестуванні класифікатору модель інтерпретувала ті ж самі висловлювання

наступним чином: «I love the music» – позитивне (0.74), «I hate the rain» – негативне (0.16), «I don’t know what I’m doing» – нейтральне (0.35).

Із графік навчання нейронної мережі помітно, що значення точності є вищими, ніж у випадку із датасетом твітів такого ж розміру (див. рис. 3.16).

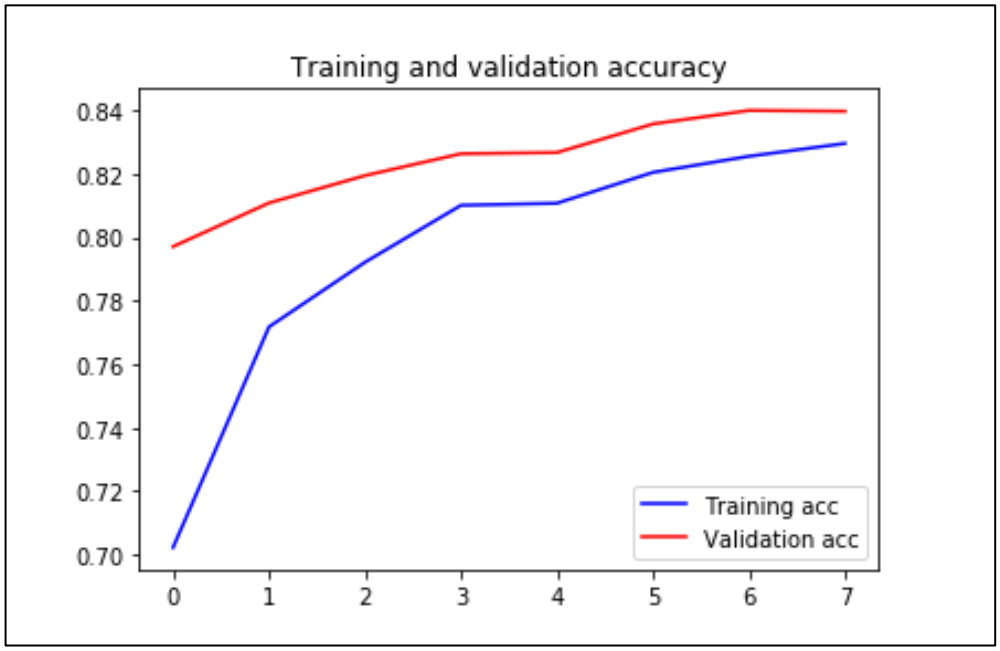

Рисунок 3.16 – Графік навчання нейронної мережі із попередньо навченим Word2Vec (10 000 відгуків)

Точність роботи класифікатору становила 75%, що на 13% більше аніж в аналогічному експерименті із тією ж кількістю твітів. Такі результати дозволяють припустити, що якість навчання більше залежить від загальної кількості слів у датасеті, ніж від кількості окремих текстів. Щоб перевірити це припущення було вирішено провести дослідження із датасетів відгуків та попередньо навченого Word2Vec. Якщо кількість слів має більше значення, ніж кількість текстів, суттєвих покращень точності не відбудеться, як це було у випадку із твітами.

Спочатку було проведено експеримент із датасетом у 10 000 відгуків. Дійсно, при використанні попередньо навченого Word2Vec точність становила 78%. Хоча це і вище на 3%, ніж із самостійно навченим Word2Vec, але це не таке суттєве покращення точності роботи, як у випадку із невеликим за кількістю текстів датасетом твітів.

Схожих результатів було досягнуто і при аналізі якості роботи нейронної мережі на датасеті із 100 000 відгуків. Точність аналізу трохи зросла та становила 86% (приблизно на 1%), тобто практично не змінилася. Тим не менше, цей показник на 16% вищий, аніж в аналогічному дослідженні із тією ж кількістю твітів.

3.3 Шляхи подальшого розвитку дослідження

Можна виділити кілька подальших шляхів розвитку як дослідження, так і самої системи. До варіантів розвитку дослідження можна віднести:

- аналіз тональності для окремих сутностей;
- дослідження роботи класифікаторів, які враховують різні модальності як, наприклад, заголовки, оцінка користувача (рейтинг) для відгуків;
- аналіз текстів різними мовами, що включає як автоматичне визначення мови, так і особливості аналізу емоційного забарвлення для кожної мови;
- врахування локації, із якої прийшов відгук, твіт, коментар, тощо.

Серед варіантів розвитку програмної системи є такі:

- розрахунок більшої кількості статистичної інформації та її презентація у вигляді інфографіки;
- аналіз трендів;
- аналіз зміни тенденцій у часі.

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Проектування архітектури та розробка back-end частини

Щоб задовольняти встановленим до прототипу програмної системи вимогам [24], для розробки прототипу програмної системи було обрано клієнт-серверну архітектуру [25]. Діаграма розгортання зображена на рисунку 4.1.

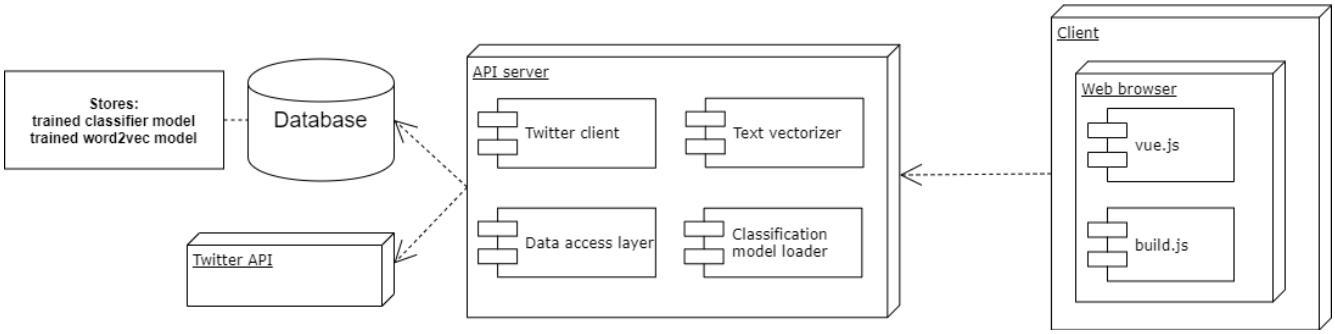

Рисунок 4.1 – Діаграма розгортання веб-застосунку

Рівень клієнту за допомогою інтерфейсу користувача приймає вхідні дані у вигляді датасету із твітів чи відгуків. Далі ці дані відправляються до рівня серверу. Іншою функцією цього рівня є виведення результатів аналізу емоційного забарвлення у зручний для сприйняття вигляд [26]. Для реалізації fron-end частини був обраний фреймворки Bootstrap та View.js.

Рівень серверу приймає дані від клієнту, виконує векторизацію отриманих даних, завантажує класифікатор, відповідно до отриманого типу даних за допомогою бібліотеки звертається до Twitter та до бази даних, у якій містяться навчений класифікатор та Word2Vec, а також передає проаналізовані дані на рівень клієнту. Серверна частина розроблена за допомогою мови програмування Python. Таке рішення можна пояснити там, що основна частина роботи лежить саме на безпосередньому аналізі даних [27]. Код відповідає стандартам для мови Python [28].

4.2 Розробка клієнтської частини

Для розробки інтерфейсу користувача було прийнято рішення використовувати фреймворк Bootstrap 4. Це сучасний фреймворк, який широко використовується та який добре зарекомендував себе. До його переваг слід віднести велику кількість вже розроблених компонентів інтерфейсу, використання сітки, яка дозволяє легко перебудовувати розмітку для різних екранів, наявність готових тем та коректну роботу у різних браузерах. Тож, використання Bootstrap 4 дозволяло пришвидшити розробку та зменшити кількість багів, пов’язаних із інтерфейсом, завдяки готовому протестованому рішенню.

Приклад розробленої сторінки наведений (сторінка аналізу відгуків) на рисунку 4.2.

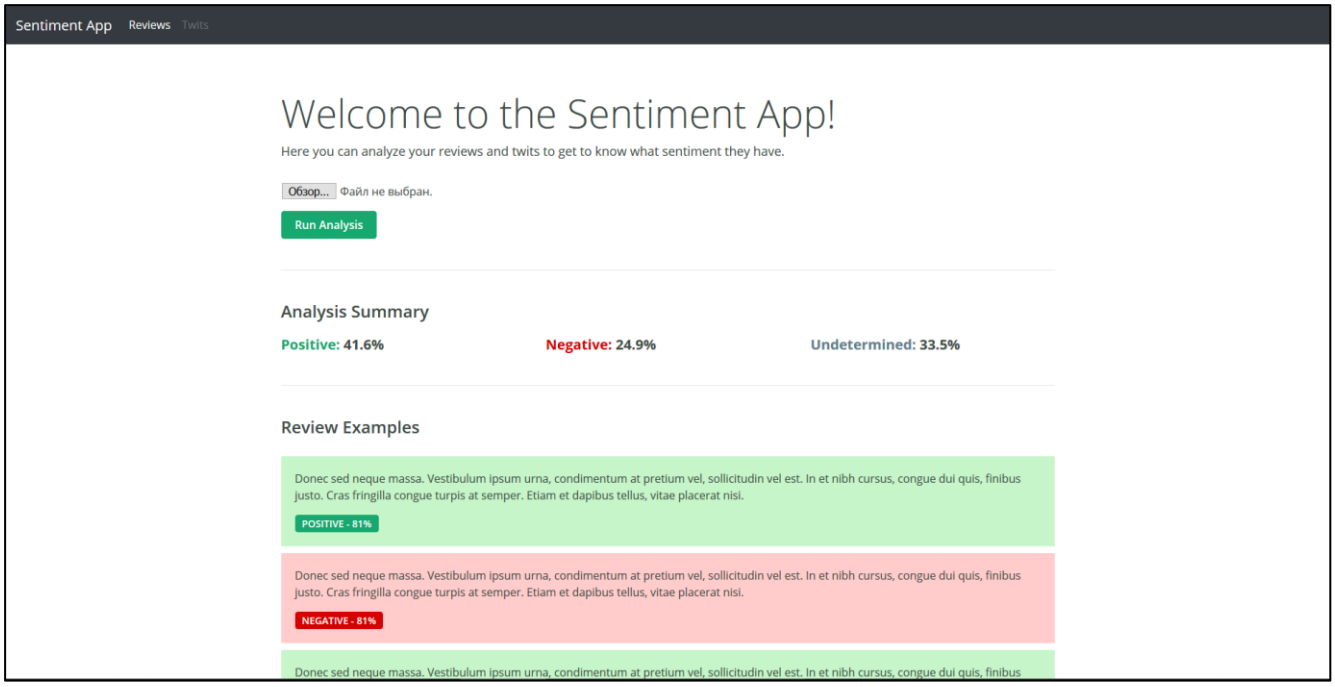

Рисунок 4.2 – Приклад розробленого інтерфейсу

Сторінки розроблені у відповідності до сітки у дванадцять колонок, тому при масштабуванні сторінки не виникає проблем (макет буде привильно відображатися на різних розмірів екранів персональних комп’ютерів).

Для відповідності створеним макетам, частина стилів була перекрита за допомогою CCS3 [29].

У якості сімейства шрифтів було обрано Open Sans. Даний вибір можна пояснити гарною підтримкою різними браузерам, відсутність ліцензійних обмежень на використання, а також велику кількість різних вагів (жирний, напівжирний, звичайний і так далі). Але головною перевагою даного сімейства є те, легкість читання.

Завантаження результатів відбувається за допомогою бібліотеки View.js – популярного фреймворку для роботи із UI елементами. Це забезпечує більш непомітну роботу для користувача, оскільки сторінка не буде перезавантажуватися повністю, а отже, контент не буде зникати.

ВИСНОВКИ

Під час проведення дослідження у межах атестаційної роботи було проведено аналіз використання методів семантичного аналізу для автоматичної обробки текстів. Після аналізу сфер застосування семантичного аналізу для подальшого дослідження було обрано відносно новий та актуальний напрямок – аналіз тональності.

При виконанні атестаційної роботи магістра були виконані наступні завдання:

- виконано аналіз предметної галузі;
- проведено огляд методів семантичного аналізу;
- розглянуто підходи та методи реалізації автоматичного аналізу тональності;
- проведено огляд способів попередньої обробки текстових даних;
- розглянуто методи векторизації тексту;
- порівняно роботу поєднання «мішка слів» та лінійних класифікаторів із Word2Vec та нейронної мережі довгої короткочасної пам’яті на датасетах твітів та відгуків із різною кількістю документів;
- порівняно роботу чотирьох лінійних моделей: логістичної регресії, наївного Байєсівського класифікатору, класифікаторів Байєса із поліноміальним розподілом та розподілом Бернулі;
- оцінено доцільність застосування стемінгу із Word2Vec;
- порівняно роботу попередньо навченого Word2Vec та навченого на власному датасеті;
- розроблено прототип системи аналізу тональності твітів та відгуків.

Прототип системи аналізу тональності був розроблений мовою програмування Python та із використанням фреймворку Bootstrap 4.

До подальших варіантів розвитку дослідження належить аспектний аналіз тональності, врахування метаданих та покращення візуалізації даних [30].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Большакова Е. Автоматическая обработка текстов на естественном языке и анализ данных: учеб. пособие / Е. И. Большакова, К. В. Воронцов, Н.Э. Ефремова, Э. С. Клышинский – М.: Изд-во НИУ ВШЭ, 2017. – 269 с.
2. Добров В. Онтологии для автоматической обработки текстов: описание понятий и лексических значений / Б. В. Добров, Н. В. Лукашевич – URL: http://www.dialog21.ru/digests/dialog2006/materials/html/Dobrov.htm (дата звернення: 19.09.19)
3. Рабчевский Е. Автоматическое построение онтологий на основе лексико-синтаксических шаблонов для информационного поиска / Е.А. Рабчевский. – Петрозаводск, 2009. – 107 с.
4. Бринк Х. Машинное обучение / Х. Бринк, Д. Ричардс, М. Феверолф. – СПб.: Питер, 2017. – 336 с.
5. Заболеева-Зотова А. Латентный семантический анализ: новые решения в Internet / А.В. Заболеева-Зотова. – М.: Информационные технологии, 2001. – 22 с.
6. Анализ тональности текста / Wikipedia, the free encyclopedia. – URL: https://ru.wikipedia.org/wiki/Анализ_тональнсти_текста (дата звернення: 24.09.19)
7. Гитис Л. Кластерный анализ в задачах классификации, оптимизации и прогнозирования / Л. Гитис. – М.: Издательство Московского государственного горного университета, 2001. – 104 с.
8. Дюран Б. Кластерный анализ / Б. Дюран. – М.: Отдельное издание, 2012. – 128 с.
9. Батура Т. Методы автоматической классификации текстов / Т. В. Батура. – Програмній продукты и системи. – 2017, №30 (1). – С. 85–99 – DOI: 10.15827/0236-235X.030.1.085-099 – URL: https://www.researchgate.net/

publication/315328102_Metody_avtomaticeskoj_klassifikacii_tekstov (дата звернення: 12.10.2019)

10. Barber D. Bayesian Reasoning and Machine Learning / D. Barber. – URL: http://web4.cs.ucl.ac.uk/staff/D.Barber/textbook/090310.pdf?roistat_visit=10865700 (дата звернення: 14.10.19)
11. Спицын В. Применение искусственных нейронных сетей для обработки информации / В. Г. Спицын, Ю. Р. Цой. – ТПУ Томск, 2007 – 32 с.
12. Nilsson J. Introduction to machine learning / J. Nilsson. – URL: http://robotics.stanford.edu/~nilsson/MLBOOK.pdf?roistat_visit=10865700 (дата зверення: 25.10.19)
13. Николаева И. Прикладная и компьютерная лингвистика / И. С. Николаева, О. В. Митрениной, Т. М. Ландо. — М.: URSS, 2016. — 320 с.
14. Kotelevskaya V. Evaluating and Improving an Automatic Sentiment Analysis System / V. Kotelevskaya. – URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.452.9996&rep=rep1&type=pdf (дата звернення: 10.10.2019)
15. Bhargave P. Lithium NLP: A System for Rich Information Extraction from Noisy User Generated Text on Social Media / P. Bhargava, N. Spasojevic, G. Hu – Proceedings of the 3rd Workshop on Noisy User-generated Text – 2017. – С. 131–139. – DOI: 10.18653/v1/W17-4417 – URL: https://www.aclweb.org/anthology/W17-4417.pdf (дата звернення: 10.10.19)
16. Gînscă A. Sentimatrix – Multilingual Sentiment Analysis Service / A. Gînscă , E. Boroș, A. Iftene, D. Trandabăț, M. Toader. – Association for Computational Linguistics. – 2011. – С. 189–195. – URL: https://www.aclweb.org/anthology/W11-1725.pdf (дата звернення: 11.10.19)
17. Agarwal A. Sentiment Analysis of Twitter Data / A. Agarwal, B. Xie, I. Vovsha, O. Rambow. – Proceedings of the Workshop on Languages in Social Media. – 2011. – С. 30-38. – URL: https://www.aclweb.org/anthology/W11-0705.pdf (дата звернення: 19.10.19)
18. Park. A. Twitter as a Corpus for Sentiment Analysis and Opinion Mining / A. Park, P. Paroubek. – Proceedings of the Seventh International Conference on Language

Resources and Evaluation (LREC'10). – 2010. – URL: https://lexitron.nectec.or.th/public/LREC-2010_Malta/pdf/385_Paper.pdf (дата звернення: 19.10.19)

19. Хайкин С. Нейронные сети. Полный курс / С. Хайкин. - М.: Вильямс, 2018. – 1104 с.
20. Рашка С. Python и машинное обучение / С. Рашка. - М.: ДМК Пресс, 2017. – 418 с.
21. Bird S. Natural Language Processing with Python / S. Bird, E. Klein, E. Loper. – O'Reilly Media, 2009 p. – 482 с.
22. Варфел Т. Прототипирование. Практическое руководство / Т. Варфел. – М. Манн, Иванов и Фербер, 2013 – 240 с.
23. Мандел Т. Разработка пользовательсого интерфейса. – Пер. с англ. – М.: ДМК Пресс, 2001. – 416 с.
24. Вигерс К. Разработка требований к программному обеспечению. / К. И. Вигерс. – М.: Русская редакция, 2004. – 576 с.
25. Фаулер М. Архитектура корпоративных программных приложений. – М.: Издательский дом "Вильямс", 2006. – 544 с.
26. Захаров А. Архитектура информационно-вычислительных сетей: методические указания / А. С. Захаров; Яросл. гос. ун-т им. П. Г. Демидова. – Ярославль: ЯрГУ, 2013. – 48 с.
27. Лутц М. Изучаем Python, 4-е издание / М. Лутц. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
28. Флаулер М. Рефакторинг: улучшение существующего кода. – Пер. с англ. – СПб: Символ-Плюс, 2003. – 432 с.
29. Робинс Д. HTML5, CSS3 и JavaScript. Исчерпывающее руковство / Д. Робинс. – Пер. с англ. – М.: Эксмо, 2014. – 528 с.
30. Бондаренко М. Концепції уніфікації інформаційно-інтелектуальних технологій в системах мовлення / М. Ф. Бондаренко, З. В. Коноплянко, Г. Г. Четвериков. – Бионика ителлекта. – 2011, №1(77). – С. 114-118.