Форма № Н-6.01

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

ФІТКІ, кафедра програмного забезпечення

(повна назва кафедри, циклової комісії)

Диплом

на тему Методи та програмні засоби розпізнавання мовних забарвлень

Студента . 6 . курсу . 1ПІ-18м . групи

Напряму підготовки . 121 - ПІ .

. Денисюк Є.Г.

(прізвище та ініціали)

керівник МКР викладач кафедри ПІ,

. к.т.н. доц. Хошаба О. М.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала

_____

Кількість балів: _____Оцінка:

ECTS _____

м. Вінниця – 2019 рік

ЗМІСТ

ВСТУП

Обґрунтування вибору теми дослідження. Обробка природної мови (Natural Language Processing, NLP) - напрямок штучного інтелекту і комп'ютерної лінгвістики, відноситься до взаємодії між комп'ютерами і природними мовами.

Методи обробки природної мови необхідні для того, щоб комп'ютери розуміли мову людей та могли на ній спілкуватися, допомагати нам орієнтуватися в навколишньому світі інформації, підбирати контекстну рекламу.

Приклади технологій NLP: пошукові сервіси, голосове керування, машинний переклад та інше [1].

Одне із завдань обробки природної мови – визначення тональності (емоційного забарвлення) текстів, яке з кожним роком набирає популярність [2].

Тональність тексту - це визначення того, які емоції автор передає в своєму повідомленні: позитивні, негативні, нейтральні [3].

Думки людей завжди впливали на прийняття рішень. І зараз, в епоху інтернету, люди щодня перевіряють пошту, заходять в соціальні мережі, на улюблені сайти, залишають коментарі.

При виборі товару не обмежуються думкою тільки друзів і сім'ї, тому що є багато відгуків про продукцію на просторах інтернету.

Компаніям сьогодні немає необхідності наймати консультантів, щоб знати думки споживачів про їх продукти та послуги, чи про товари, послуги конкурентів, тому що призначений для користувача контент в мережі може вже сьогодні дати їм таку інформацію.

Якщо про продукт компанії або послуги багато позитивних відгуків, то швидше за все споживач вибере саме його (її). Але для цього потрібно переглянути сотні, мільйони відгуків.

Наприклад, в соціальній мережі Twitter користувачі в день відправляють в середньому 500 000 000 твітів (повідомлень).

[4]. Треба багато часу на вивчення такого великого обсягу інформації.

А як казав Натан Ротшильд: «Хто володіє інформацією, той володіє світом »[5].

Більшість готових рішень для визначення емоційного забарвлення текстів представлені на англійській мові. Тому необхідно поліпшення і розробка ефективних алгоритмів визначення тональності текстів для російської мови.

Все вищевикладене робить актуальним завдання розпізнавання мовних забарвлень.

Мета та завдання дослідження. Метою роботи є підвищення достовірності оцінювання забарвлення мови за рахунок розробки нового методу та алгоритмів.

Основними задачами дослідження є:

- провести аналіз методів і засобів визначення мовного забарвлення, для визначення напрямків підвищення їх достовірності;
- запропонувати новий метод підвищення достовірності оцінювання мовного забарвлення;
- розробити модифікований алгоритм та програмний компонент на основі запропонованого методу;
- провести експериментальне дослідження розробленого засобу розпізнавання.

Об’єкт дослідження – процес розпізнавання мовного забарвлення.

Предмет дослідження – методи та засоби для визначення мовного забарвлення.

Методи дослідження. У процесі досліджень використовувались: теорія чисел та чисельних методів, теорія статистики.

Наукова новизна отриманих результатів.

1. Уперше запропоновано метод визначення лексичних одиниць-маркерів, особливість якого полягає у визначенні найбільш значимих слів, які впливають на тональність тексту, що дає можливість підвищити достовірність системи до 5% .
2. Вперше запропоновано метод визначення мовного забарвлення, особливість якого полягає у використанні стоп-слів та машиного навчання, що дозволяє підвищити достовірність системи.

Практична цінність отриманих результатів. Практична цінність одержаних результатів полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби для аналізу мовного забарвлення текста.

Розроблена на основі запропонованого методу і алгоритмів програмна система може бути використана як самостійний інструмент для резюмування думок щодо аспектів досліджуваної сутності на основі колекції документів, що належать конкретній предметній області. Також розроблена система може бути інтегрована в інформаційні сервіси з метою автоматичного сортування позитивних і негативних статей.

Зв’язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення

Особистий внесок здобувача. Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто.

Апробація матеріалів магістерської кваліфікаційної роботи. Основні положення магістерської кваліфікаційної роботи доповідалися та обговорювалися на конференції ВНТУ 2018.

Структура та обсяг роботи. Магістерська кваліфікаційна роботи складається зі вступу, чотирьох розділів, висновків, списку літератури, що містить 45 найменувань, 3 додатків. Робота містить 15 ілюстрацій, 3 таблиці.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Сфера досліджень, яка фокусується на взаємодії між природною мовою та комп'ютерами, називається обробкою природної мови (Natural Language Processing, NLP).

NLP використовується для аналізу текстів, дозволяючи машинам зрозуміти, що каже людина. Це складне завдання в інформатиці. Зрозуміти природну мову - означає необхідність зрозуміти не тільки окремі слова, а й їх зв'язок, сенс, емоційну забарвленість.

Дослідження в сфері обробки природної мови почалися в 1950-х роках. Автоматичний переклад з російської мови на англійську був протестований в 1954 році.

Це була спільна розробка Джорджтаунського університету і компанії «IBM». В ході експерименту було продемонстровано автоматичний переклад понад 60 пропозицій. Презентація позитивно вплинула на розвиток машинного перекладу в наступні 12 років [6].

У 1960-х роках з'явилася програма, яка розуміла природну мову - SHRDLU. Користувач спілкувався з програмою SHRDLU на розмовній англійській мові. За його наказом SHRDLU носила прості об'єкти, такі як кубики, кулі, конуси [7].

До 1980-х років більшість систем NLP функціонувала на складних рукописних правилах, але в кінці 1980-х відбулася революція в обробці природної мови з впровадженням алгоритмів машинного навчання. У 21 столітті дослідження в області NLP швидко розвиваються. Комп'ютери по всьому світу стають все більш розумнішими, наприклад, IBM Watson [8].

IBM Watson - суперкомп'ютер, розроблений фірмою IBM, має систему штучного інтелекту, яка здатна приймати питання і відповідати на них. У лютому 2011 система була перевірена в телепередачі «Jeopardy!».

Його суперниками були Бред Раттер – володар найбільшого виграшу в програмі, і Кен Дженнінгс, який вигравав на шоу 75 раз поспіль. Watson,

використовуючи сучасні методи NLP, переглянувши свою величезну базу даних, без підключення до інтернету, дав найбільш ймовірну відповідь і переміг [9].

Зараз обробка природної мови включає в себе такі методи як [10]:

- автоматичне реферування - витяг найбільш важливих відомостей з одного або декількох документів і генерація на їх основі лаконічних звітів [11];
- аналіз тональності;
- витяг термінології – додаткове завдання вилучення інформації, метою якого є автоматичне вилучення термінів з тексту документів [12];
- розпізнавання іменованих сутностей - визначення місця розташування іменованих сутностей в тексті, таких як імена людей, організацій, міст і так далі [13];
- розмітка частин мови - визначення частини мови і граматичних характеристик слів в тексті [14];
- витяг відносин - виявлення і класифікація семантичних відносин в текстах [15]. Приклади відносин: (автор, книга), (організація, головний офіс);
- стемінг - процес знаходження основи слова для заданого вихідного слова. Основа слова може відрізнятися від морфологічного кореня слова [16].

1.1 Аналіз тональності текстів

Аналіз тональності тексту (сентимент-аналіз, емоційне забарвлення, sentiment analysis, opinion mining) - це область комп'ютерної лінгвістики, яка займається вивченням думок в текстових документах [3].

Текстову інформацію в мережі інтернету можна розділити на два типи:

1. Факт - подія, яка відбулася або є насправді [17].
2. Думка - судження, що виражає чуттєве ставлення до предмета або події – суб’єктивну оцінку, погляд.

Думки поділяються на два типи [18]:

- прості - точка зору автора про який-небудь об'єкт;
- порівняння - процес кількісного або якісного порівняння різних властивостей (схожості, відмінності, переваг і недоліків) двох або більше об'єктів з метою формування певної оцінки або думки для одного з них.

Говорячи про емоційне забарвлення тексту, слід виділити три параметри [2]:

- суб'єкт тональності (автора тексту);
- тональна оцінка (позитивна, нейтральна, негативна);
- об'єкт тональності (тема, про яку можна почути думку).

Текст вважається нейтральним, якщо він не містить в собі емоційної складової [19].

Приклади:

- Повідомлення: «Приємно, коли країна, яка проводить Паралімпіаду, є твоєю Батьківщиною. Всі активно нас підтримують, і це неймовірно!»

  Автор: Михалина Лисова.

  Тема: «Олімпійські ігри».

  Тональність: «позитивна».

- Повідомлення: «Купа ляпів, тупий сюжет, картонні персонажі – і це тільки невелика частина цього фільму. Сходив у кіно, витратив даремно час і гроші. Загалом, повний нуль!»

  Автор: Владлен Ісаков.

  Тема: «Фільми».

  Тональність: «негативна».

- Повідомлення: «Фільм дуже природньо показує життя підлітків, їх відносини з батьками.»

  Автор: Валентина Инкин.

  Тема: «Фільми».

  Тональність: «нейтральна».

1.2 Проблеми автоматичного визначення емоційної забарвленості текстів

Проблеми, які виникають при використанні автоматичних алгоритмів визначення емоційного забарвлення текстів:

- Російська мова - флективна мова з вільним порядком слів. Такі особливості необхідно враховувати на етапі попередньої обробки, наприклад, робити лематизацію тексту – приведення слів до початкової форми. Це призводить до ускладнення алгоритму.
- Значення тональності текстів залежить від предметної області. Наприклад, слово «несподіваний» може мати позитивне забарвлення, якщо відноситься до якого-небудь фільму, але в області клієнтського обслуговування це не так. Для правильної класифікації при використанні методів навчання з учителем досить, щоб навчальні та тестові вибірки мали однакову предметну область.
- Вживання смайликів відмінного від тексту забарвлення. Наприклад, «Це якийсь жах!) Кошмар!)».
- Текст може мати позитивне і негативне забарвлення в одному реченні. Наприклад, «Я люблю каву, але кава три в одному мені не подобається.». «Я люблю каву» - позитивне забарвлення, «кава три в одному мені не подобається» - негативне.
- Складно правильно класифікувати текст, в якому використовується сарказм. Не всі люди його можуть розпізнати.
- Дослідження, проведені в роботі [21] показують, що 93% слів в тексті вживаються менш, ніж 10 раз. Дослідження проводилось на корпусі рецензій фільмів з IMDB. Такі дані пояснюються частим використанням сленгу.

1.3 Підходи до класифікації тональності

Існує кілька підходів до класифікації тональності текстів [2]:

- підходи, засновані на правилах орфографії, семантики, морфології та словниках;
- машинне навчання з вчителем;
- машинне навчання без вчителя.

Підхід, заснований на словниках використовує в своїй основі тональні словники. Тональні словники - список слів зі значенням тональності. Приклад таблиці тонального словника приведений у таблиці 1.1.

Таблиця 1.1

Приклад тонального словника

Слово	Тональність
смуток	-1
небо	0
радість	1
щасливий	1
сумний	-1

Для отримання підсумкової тональність необхідно взяти середнє арифметичне або обчислити суму значень тональності всіх слів тексту [22].

Головна перевага цього алгоритму - простота використання. Недоліки - для кожної предметної області потрібен окремий словник і створення такого словника може бути досить важким завданням.

Підхід, заснований на правилах, полягає в застосуванні правил, які складаються експертами на основі аналізу предметної області [23].

Приклад такого правила: якщо текст містить один або кілька позитивних прикметників з набору {«веселий», «смішний», «добрий». .. } І не містить прикметників {«поганий», «нудний», «страшний». . . }, тоді текст відноситься до позитивного класу тональності.

Розглянемо ще один приклад: слово «сліпучий» в більшій частині зустрічається як позитивна характеристика, але в реченні: «Сніг на сонці був настільки сліпучим, що я вже нічого навколо не бачив» - є негативною характеристикою, тому що вживається в значенні «Засліплювало». З чого випливає, що для хорошої роботи системи необхідно скласти велику кількість правил, це вимагає значних трудових витрат.

Найбільш поширеним підходом є машинне навчання з учителем. Спочатку на заздалегідь розмічених текстах навчається машинний класифікатор, а потім

отримана модель використовується при аналізі нових текстів. Короткий алгоритм [24]:

- спочатку навчання машинного класифікатора відбувається на заздалегідь зібраній колекції документів (текстів);
- кожен документ розглядається у вигляді вектора ознак, за яким він буде досліджуватися;
- визначається тональність для кожного документа;
- вибирається алгоритм класифікації і підхід для навчання класифікатора;
- отримана модель використовується для визначення емоційного забарвлення документів нової колекції.

Машинне навчання без вчителя спрямоване на виявлення внутрішніх взаємозв'язків, взаємозалежності, закономірностей, що існують між об'єктами. Для тренування алгоритму використовується навчальна вибірка, що складається з текстів, класи яких заздалегідь невідомі (або відомі, але ця інформація не використовується алгоритмом). Це і є головна перевага цього підходу.

Але він значно поступається в точності машинному навчанню з учителем, тому що неможливо передбачити, що буде на виході [25].

1.4 Методи навчання без учителя

Оскільки емоційно забарвлені слова (або слова-сентименти) є домінуючим фактором для аналізу тональності, не важко уявити, що емоційно забарвлені слова та вирази можуть бути використані в методах для аналізу тональності з відсутністю вчителя. Подібні методи були описані в роботі Turney. Такий метод виконує класифікацію, що основана на певних синтаксичних структурах, які як правило використовуються людьми для висловлення думок. Такі синтаксичні структури, або шаблони, як правило, базуються на певних частинах мови (Part-of-speech, POS) та їх тегах. Алгоритм для такої класифікації складається з наступних кроків:

Крок 1. Два послідовних слова вилучаються, якщо їх POS теги відповідають одному з шаблонів. Наприклад, шаблон означає, що 2 послідовних слова

вилучаються, якщо перше слово – прислівник, а друге слово – прикметник. Третє слово, що не вилучається, це – іменник. Наприклад, для такого речення «Це піаніно має такий чудовий звук», «чудовий звук» вилучається, адже ця фраза відповідає шаблону, де перше слово – це прикметник, а друге слово – іменник. Причина, по якій використовуються ці шаблони, це те, що слова JJ, RB, RBR та RBS як правило висловлюють певну думку. Іменники та дієслова виступають у ролі контексту, тому що саме в залежності від контексту JJ, RB, RBR та RBS можуть висловлювати різні думки, позитивні чи негативні. Так, наприклад, прикметник (JJ) «непередбачуваний» буде висловлювати негативну думку, якщо це відгук про машини («непередбачуване рульове управління»), або позитивну думку, якщо мова йде про відгуки до фільмів («непередбачуваний сюжет»).

Крок 2. Необхідно оцінити орієнтацію емоційного забарвлення (або сентименту) всіх вилучених фраз, використовуючи метрику точної взаємної інформації (Pointwise mutual information, PMI) за допомогою формули 1.1:

$$PMI(term1,term2) = \log2\left(\frac{Pr(term1 \wedge term2)}{Pr(term1)Pr(term2)}\right) \quad (1.1)$$

де

PMI - ступінь статистичної залежності між двома термінами.

Pr ($term1 \wedge term2$) – це фактична ймовірність спільної появи term1 та term2.

Pr ($term1$)Pr ($term2$) – спільна ймовірність виникнення двох доданків, якщо вони є статистично незалежними.

Емоційне забарвлення фрази (позитивне чи негативне) вираховується з огляду на його зв'язок з позитивним опорним словом "відмінно" і негативним опорним словом "погано" за допомогою формули 1.2:

$$SO(\text{фраза}) = PMI(\text{фраза, відмінно}) - PMI(\text{фраза, погано}) \quad (1.2)$$

Ймовірності обчислюються на основі запитів до пошукової системи, і збираючи кількість влучень (hits). Для кожного запиту, пошуковий механізм видає

набір релевантних документів, що і є кількістю влучень. Таким чином, виконуючи пошук по двом термам разом та окремо, ймовірності в формулі можуть бути прораховані. В роботі Tuney [24], був використаний пошуковий двигун AltaVista, тому що він має оператор ПОРЯД, щоб обмежити пошук документами, які містять слова в межах десяти слів один від одного в довільному порядку. Нехай hits – це отримана кількість влучень. Тоді рівняння (1.2) можна переписати наступним чином:

$$SO(\text{фраза}) = \log 2 \left(\frac{hits(фраза\ ПОРЯД\ відмінно)\, hits(погано)}{hits(фраза\ ПОРЯД\ погано)\, hits(відмінно)} \right) \qquad (1.3)$$

Крок 3. Маючи певний огляд, алгоритм обчислює середнє значення SO всіх фраз в огляді і класифікує огляд, як позитивний результат, якщо середній SO є позитивним і як негативний – в іншому випадку. Підсумкова класифікація по точності оглядів з різних областей становить діапазон від 64% для автомобільних оглядів до 46% для оглядів фільмів.

1.5 Методи, засновані на словниках

Іншим методом є метод, оснований на словниках. Цей метод використовує словник емоційно забарвлених слів та фраз. Кожному слову чи фразі привласнена полярність та сила, і включає в себе інтенсифікацію і заперечення, щоб обчислити емоційне забарвлення кожного документу. Найбільшою проблемою методів, заснованих на словниках і правилах, є важкість процесу складання словника. Для одержання методу, що класифікує документ з високою точністю, терміни словника повинні мати вірну вагу, адекватну предметній області документа. Наприклад, слово «непередбачуваний» по відношенню до сюжету фільму є позитивною характеристикою, але негативною по відношенню до, наприклад, політика. Тому даний метод вимагає значних затрат часу людини, через те, що для хорошої роботи системи необхідно скласти велику кількість правил. Часто можлива

автоматизація складення словників, проте зазвичай лише у дуже вузькій предметній області.

У простому вигляді тональний словник представляє з себе список слів зі значенням тональності для кожного слова. Приклад з бази ANEW, перекладений на українську мову, наведений в таблиці 1.2.

Таблиця 1.2

Приклад тонального словника по шкалі 1-9

Слово	Тональність (1–9)
happy	8.11
good	7.37
dull	2.85
angry	2.75
sad	1.51

1.6 Напівавтоматичне навчання

Напівавтоматичне навчання (semi-supervised learning) - це клас методів навчання, які використовують для тренувань як розмічені так нерозмічені дані, при чому як правило, кількість розмічених даних менше за кількість нерозмічених даних. Напівавтоматичне навчання залежить від навчання без учителя (без будь-яких помічених навчальних даних) та навчання з учителем (з повністю наголошеними навчальними даними). Багато дослідників, які навчаються в машинобудуванні, встановили, що немітовані дані, коли вони використовуються разом із невеликою кількістю мічених даних, можуть значно покращити точність навчання. Для отримання розмічених даних для вивчення проблеми часто потрібен кваліфікований агент-людина (наприклад, для транскрибування аудіо сегмента) або фізичний експеримент (наприклад, визначення тривимірної структури білка або визначення того, чи існує масло в певному місці). Таким чином, витрати, пов'язані з процесом маркування, можуть стати занадто високими, а отримання немаркованих даних є

Відносно недорогим. У подібних ситуаціях напівавтоматичне навчання може мати велике практичне значення. В аналізі тональності напівавтоматичне навчання не дуже розповсюджене, найбільш відомим методом напівавтоматичного навчання в аналізі тональності є метод, описаний у роботі Хе Юлан та Жоу Деу. Метод полягає у використанні як словникових методів, так і класичного класифікатора, наприклад SVM. Словникові методи використовуються для навчання класифікатора на великій нерозміченій вибірці, і таким чином поступово класифікатор навчається особливостям предметної області.

1.7 Алгоритми для автоматичної класифікації тональності текстів

Для класифікації текстів за допомогою машинного навчання з учителем існують кілька відомих алгоритмів:

- наївний байєсовський класифікатор;
- метод k найближчих сусідів;
- метод опорних векторів;
- метод логістичної регресії.

1.7.1 Наївний байєсовський класифікатор

Наївний байєсовський алгоритм - це алгоритм класифікації, заснований на теоремі Байєса з припущенням про незалежність ознак. Він передбачає, що наявність якої-небудь ознаки в класі не пов'язано з наявністю будь-якої іншої ознаки.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \qquad (1.4)$$

де

P (c | d) - ймовірність, що документ d належить класу c;

P (d | c) - ймовірність зустріти документ d серед всіх документів класу c;

P (c) - безумовна ймовірність зустріти документ класу c в корпусі документів;

P (d) - безумовна ймовірність документа d в корпусі документів.

Теорема дозволяє розрахувати ймовірність того, що саме ця причина привела до спостережувальної події. Для того, щоб визначити найбільш ймовірний клас, до якого належить документ, потрібно скористатися оцінкою апостеріорного максимуму.

Іншими словами, потрібно розрахувати ймовірність для всіх класів і вибрати той клас, який володіє максимальною вірогідністю, яка обчислюється за формулою 1.5.

$$C_{map} = \arg max_{сэс}[P(d \mid c) \cdot P(c)] \quad (1.5)$$

Згідно класифікатора Байєса, документ - набір слів, ймовірності яких умовно не залежать одне від одного. В результаті чого, умовна ймовірність документа апроксимується утворенням умовних ймовірностей всіх слів, що входять в документ, як показано у формулі 1.6.

$$P(d \mid c) \approx P(w_1 \mid c) \cdot P(w_2 \mid c) \cdot \ldots \cdot P(w_n \mid c) = \prod^{n}_{i=1} (P\, w_i \mid c) \quad (1.6)$$

Підставивши отриманий вираз в формулу (1.5), отримаємо:

$$C_{map} = \arg max_{сэс}[P(c) \cdot \prod^{n}_{i=1} (P\, w_i \mid c)] \quad (1.6)$$

При порівняно великому обсязі документа необхідно множити велику кількість маленьких чисел. Тому, щоб уникнути арифметичного переповнення, знизу можна скористатися властивістю логарифма множення log ab = log a + log b. Так як логарифм є монотонною функцією, то його застосування до обох частин вираження змінить тільки чисельне значення, але не параметри, при яких досягається максимум.

Формула (1.6) з використанням логарифма:

$$C_{map} = \arg max_{c \in C} [P(c) \cdot \sum_{i=1}^{n} \log P(w_i|c)] \quad (1.7)$$

Ймовірність класу визначається за формулою 1.8:

$$P(c) = \frac{Dc}{D} \quad (1.8)$$

де

D_c - кількість документів, що належать класу с,

D – загальнf кількість документів у вибірці.

Ймовірність слова в класі визначається формулою 1.9:

$$P(w_i|c) = \frac{w_{ic}}{\sum_{i'v} w_{i'c}} \quad (1.9)$$

де

w_{ic} - кількість разів коли і-е слово зустрічається в документах класу с,

V - кількість слів у всіх документах класу с.

Але, якщо зустрілося слово, якого немає в документах класу, то P (w_1| c) буде дорівнювати нулю. Вирішити дану проблему можна домовившись, що будь-яке слово в класі ми зустрічаємо мінімум один раз. Ось чому класифікатор Байєсова називається наївним.

Отримуємо підсумкову формулу 1.10 [26]:

$$P(c) \cdot \sum_{i=1}^{n} \log \frac{w_{ic}}{|V| + \sum_{i' \in V} W_{i'c}} \quad (1.10)$$
$$c_{map} = argmax_{c \in C}$$

Основні переваги даного класифікатора:

- низька обчислювальна складність;
- застосовується на великих наборах даних.

1.7.2 Метод опорних векторів

Метод опорних векторів відноситься до сімейства лінійних класифікаторів. Завданням лінійної класифікації є пошук гіперплощини в просторі ознак, що розділяє всі об'єкти на класи.

Основна ідея методу опорних векторів полягає в виявленні поділяючої гіперплощини, максимально віддаленої від найближчих до неї крапок в просторі ознак.

Якщо класи не можна відразу лінійно розділити, то алгоритм буде додавати новий вимір в прагненні подальшого поділу. Він буде продовжувати цей процес зображений на рисунку 1.1, поки не буде здатний розділити ознаки на два окремих класи.

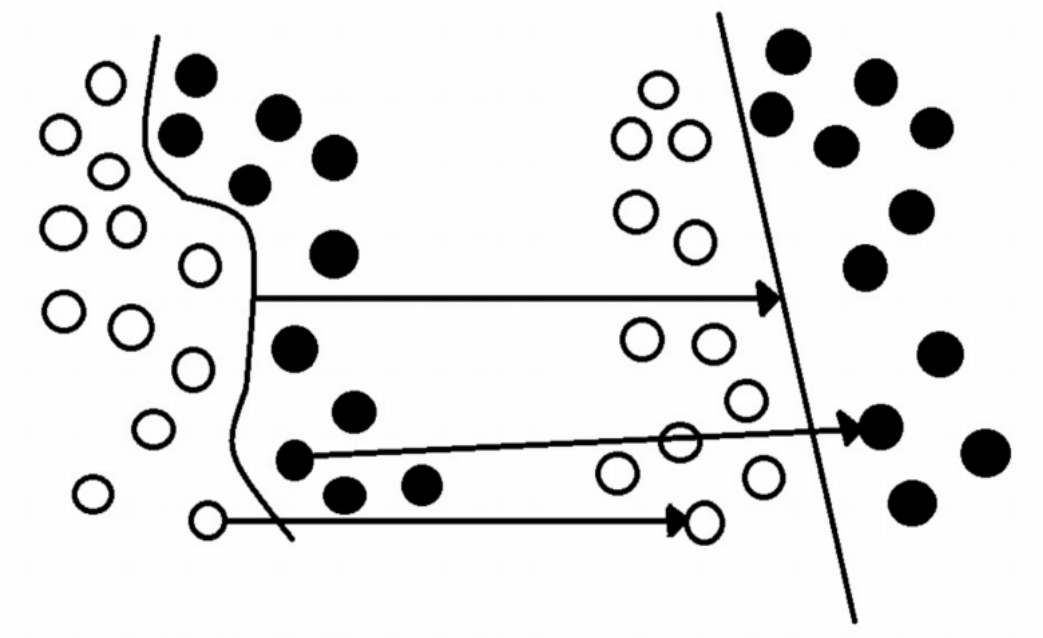

Рисунок 1.1 - Приклад поділу на класи методом опорних векторів

Після знаходження алгоритмом гіперплощини, що забезпечує максимальний рівень відмінності між класами, він може приступити до класифікації нових даних, які класифікуються в залежності від того, по який бік від гіперплощини вони виявляються.

Однак у даного методу істотний недолік: при збільшенні кількості вимірів, вектор ознак збільшується в геометричній прогресії [27; 28].

1.7.3 Метод k найближчих сусідів

Метод k найближчих сусідів зображений на рисунку 1.2 - є найпростішим алгоритмом класифікації. Основна ідея: об'єкт належить до того класу, до якого належить більшість найближчих його сусідів.

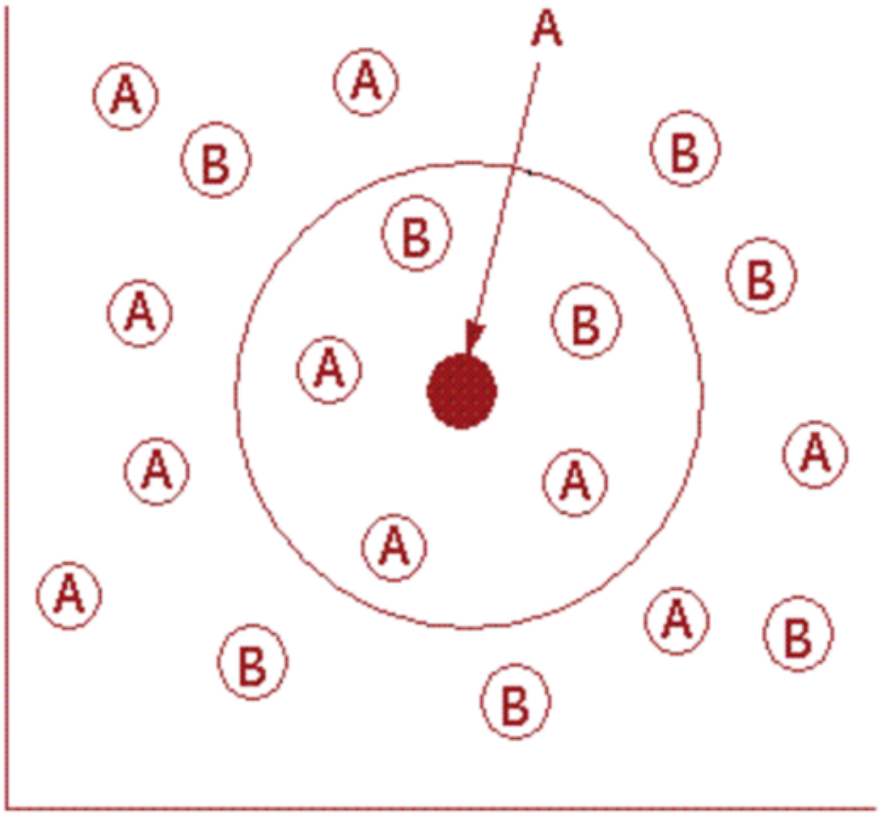

Рисунок 1.2 - Метод k найближчих сусідів

k - кількість сусідніх об'єктів, які порівнюються з класифікаційним об'єктом. Якщо k = 5, то об'єкт буде порівнюватися з 5 сусідами.

Алгоритм:

1. Підготувати розмічену (навчальну) вибірку об'єктів, тобто вказати до якого класу належить кожен об'єкт.

2. Обчислити відстань до кожного з об'єктів навчальної вибірки. В більшості алгоритмів відстань розраховується за допомогою формули Евклідової відстані 1.11:

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{m} (x_i^{(l)} - x_j^{(l)})} \quad (1.11)$$

де

$x_i = (x_i^{(1)}, .., x_i^{m})$ – вектор m ознак i-ого об’єкту,

$x_j = (x_j^{(1)}, \ldots x_j^{m})$ – вектор m ознак j-ого об’єкту.

3. Відібрати k об'єктів навчальної вибірки, відстань до яких мінімальна.
4. Клас класифікуємого об'єкту - це клас, який найбільш часто зустрічається серед k найближчих сусідів.

Якщо значення k буде малим, то може виявитися, що єдиним найближчим об'єктом буде об'єкт з неправильним певним класом, який дасть невірне рішення, такі випадки називають «викид».

Якщо k матиме велике значення (N), то тоді «переможе» найпопулярніший клас. В такому випадку, відстань до класифікуємого об'єкту не має значення.

Компромісом вважають, коли k = $\sqrt{N}$. При впевненості, що усі об'єкти вибірки правильно класифіковані, можна брати k меншим [29].

Недоліки алгоритму [30]:

- якщо серед навчальних об'єктів є об'єкт невірно класифікований, то і найближчі до нього об'єкти, також будуть класифіковані невірно;
- алгоритм повністю залежить від того, наскільки вдало вибрані ознаки.

1.7.4 Метод логістичної регресії

Логістична регресія застосовується для передбачення ймовірності виникнення деякої події за значеннями безлічі ознак. Ознаки можуть бути як числовими, так і категоріальними. Наприклад, ймовірність, того, що у людини

станеться серцевий напад в певний період часу, може бути передбачена в залежності від віку людини, статі та індексу маси тіла. На рисунку 1.3 зображено приклад такої функції.

Рисунок 1.3 - Логічна функція з z на горизонтальній осі і f (z) на вертикальній осі

Логістична функція обраховується за допомогою формули 1.12:

$$f(x)=\frac{1}{1+e^{-z}} \tag{1.12}$$

Де змінна z включає в себе набір факторів ризику, які обраховуються формулою 1.13, а f (x) – ймовірність конкретного результату, при заданому наборі ризиків.

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + .. + \beta_k x_k \quad (1.13)$$

де

β_0 - «точка перетину», тобто величина z, при нульових значеннях факторів ризику;

β_1, β_2, .., β_k - «коефіцієнт регресії» для факторів ризику x_1, x_2,..,x_k.

Якщо коефіцієнт регресії позитивний, то це означає, що збільшується ймовірність аналізованого результату. Якщо коефіцієнт негативний, то ймовірність аналізованого результату зменшується.

Великий коефіцієнт регресії означає, що даний фактор значно впливає на сукупний ризик. Якщо коефіцієнт регресії близький до нуля, то цей фактор має невеликий вплив на ймовірність результату [31].

1.8 Огляд готових рішень для аналізу тональності російських текстів

1.8.1 SentiStrength

SentiStrength - програмний продукт, створений для оцінки рівня емоцій в коротких веб-контекстах, здатний працювати з тисячами текстів в секунду. Виконує аналіз тональності для 14 мов: російської, англійської, німецької та інших. В основі системи лежить підхід на правилах і підхід машинного навчання з учителем.

Результат аналізу представляється у вигляді трьох оцінок - позитивної (за шкалою від +2 до +5), негативної (за шкалою від -2 до -5), нейтральної (-1, 1).

Крім того, SentiStrength надає докладне пояснення отриманої оцінки [32].

Приклад аналізу тексту: «Данный фильм заставил меня разочароваться в современном искусстве».

Як видно з рисунка 1.4, SentiStrength оцінив даний текст як нейтральний (result = 1), хоча він відноситься до негативного класу тональності.

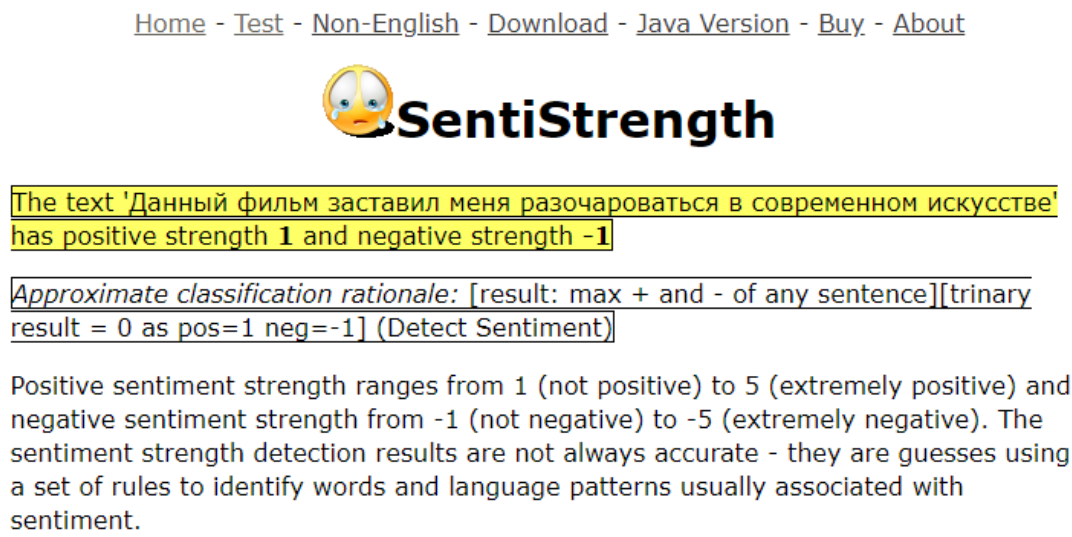

Рисунок 1.4 - Результат аналізу тональності в SentiStrength

Недоліки: дана система не враховує специфіку російської мови, наприклад, морфологію. Для ефективності результатів аналізу необхідно визначити початкову форму слова або зберігати всі словоформи в базі даних.

1.8.2 WordNet-Affect

WordNet-Affect - ресурс (словник), який містить слова, які описують емоції. В основі WordNet-Affect лежить підхід на словниках.

Для даного ресурсу дієслова, іменники, прикметники, прислівники, які несуть деяку емоційну складову, були вручну розмічені за допомогою спеціальних позначок. Ці емоційні позначки характеризують різні стани, що виражають настрої або ситуації, які викликають емоції.

Для визначення тональності текстів в WordNet-Affect є 4 позначки: нейтральна, позитивна, неоднозначна, негативна.

Позитивна позначка забарвлення включає в себе такі мітки як радість # 1 або захоплення # 1.

Негативна включає в себе негативні мітки, наприклад, гнів # 4 або печаль #4.

Більш докладний опис кожної мітки дано на офіційному сайті WordNetAffect [33].

Згідно роботи [34] SentiStrength показує більшу ефективність в аналізі тональності англійських текстів з соціальної мережі Twitter, ніж WordNet-Affect. Тестування для російської мови також зазнало невдачі.

Однак, можна зробити висновок, що комбінація двох підходів (підхід на правилах і машинне навчання з учителем) ефективніше, ніж підхід на словниках.

В роботі [35] за результатами використання машинного навчання з учителем для класифікації тональності показано, що такий підхід значно перевершує підхід на словниках.

Тому в даній роботі вирішено використовувати комбінацію декількох підходів: машинне навчання з учителем і підхід на словниках.

Додавання підходу на словниках допоможе поліпшити алгоритм. Словники міститимуть певну кількість слів, які явно вказують на

приналежність до конкретного класу тональності, а не весь словник російської мови.

Згідно роботи [36] найбільш ефективними при визначенні тональності коротких тестових повідомлень себе показали: наївний байєсовський класифікатор і метод опорних векторів.

В роботі [37], в ході попереднього тестування, було виявлено, що метод k найближчих сусідів показує істотно гірші результати, ніж інші.

Так як наївний байєсовський класифікатор має низьку обчислювальну складність і має високу ефективність, то було вирішено використовувати його в якості алгоритму для класифікації текстів.

1.9 Висновок

Був проведений аналіз літератури в області визначення емоційного забарвлення текстів. За підсумками аналізу був обраний комбінований підхід - машинне навчання з учителем і підхід на словниках, а в якості алгоритму - наївний байєсовський класифікатор так як він володіє рядом переваг:

- низька обчислювальна складність;
- застосовується на великих наборах даних.

2. РОЗРОБКА МОДИФІКОВАНОГО АЛГОРИТМА ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТІВ

У попередньому розділі в якості підходів для аналізу тональності було вибрано машинне навчання з учителем і підхід на словниках, так само був обраний алгоритм класифікації (наївний байєсовський класифікатор).

Тому розробка ефективного алгоритму повинна включати наступні етапи:

- вибір векторної моделі;
- вибір навчальної вибірки;
- попередня обробка текстів;
- вибір заходів для оцінки ефективності алгоритму;
- тестування ефективності алгоритму.

2.1 Векторна модель

В рамках завдань обробки тексту популярно представлення документів у вигляді n-грамм, де n-грамма - послідовність слів довжиною n (n> 0) [38].

Якщо n = 1, то така послідовність складається з одного слова і називається уніграмма. У ній вектор ознак являє собою набір всіх слів з тексту. Якщо n дорівнює двом - біграмма, тобто термін утворює пари слів з тексту, і так далі. Наприклад, пропозиція «Катя любить ходити по крамницям» можна представити у вигляді уніграмм: {« Катя »,« любить », «ходити», «по», «крамницям»} або біграмм: { «Катя любить», «любить ходити», «Ходити по", "по крамницям»}.

Найчастіше використання біграмм дає точніші результати. Наприклад: словосполучення «не люблю» якщо розглянути, як незалежні слова (уніграмми), отримаємо нейтральну тональність, так як «не» - негативне забарвлення, «люблю» - позитивне. При використанні біграмм, «не люблю» носить негативне забарвлення.

Також можна комбінувати уніграмми і біграми: {«Катя», «любить», «любить ходити», «по крамницям»}.

З ростом, n-грамми все точніше відображають оригінальний текст, тому цінність її застосування значно зменшується. У завданнях класифікації текстів мало коли використовується n > 3.

2.2 Навчальна вибірка

Навчальна вибірка представляє собою заздалегідь розмічену (з позначенням емоційного забарвлення) колекцію документів, на основі якої буде складено тональний словник.

В якості навчальної вибірки був обраний корпус російськомовних текстів, зібраних з соціальної мережі Twitter і розподілених на три класи: позитивний, негативний і нейтральний [39]. Приклад текстів зображено на рисунку 2.1.

Кожен текст в корпусі має наступні атрибути:

- текст твіта;
- клас, до якого належить текст (позитивний, негативний).

18831	Хотя казалось бы - мой журнал непопулярен, а неадекваши прутъ и прутъ.А расскажите про руки? приходилось. за хамство и оскорблен	0
18832	ru . Захожу значит туда, вбиваю свои данные, там имя по-моему и фамилию, а и ещё дату рождения если память меня не подводит. Ну та	-1
18833	не плевать на Штаты, совершенно. Я хочу лишь, чтобы нефть и газ в России скорее закончились, чтобы уже можно было начинать норма.	0
18834	Попросил посчитать, если не будет никаких изменений, без учета роста цен ,минимальной зарплаты/пенсии и без учета инфляции (пред	-1
18835	Сейчас четвёртый отказ. А третий кто поставил? Официально, если в паспорте шенгенская виза, страна въезда не должна играть роль, на	-1
18836	и вам всего доброго :) он 204 см ростом.много копать надо! а у нас вообще есть кто в политике не с указанной вами целью? а я - проти	-1
18837	Причём они удивительно хорошо встраиваются в нынешнюю кланово-феодальную систему власти, и оказываются в тех же администрац	-1
18838	Мой комментарий в частности был ответом на вопрос из вашего поста жертва и агрессор могут жить в одном человеке ... Кого будем обе	-1
18839	Лимонова и его сторонников устраивает наша выборная система и её результаты? Нет? Тоже европейской системы хочется? Нет? А како	-1
18840	тоже не все ставлю, но и категорически их не отвергаю. Это хороший показатель, значит все хорошо. Привки, да это палки о двух конца	0
18841	а литургия? это теперь называется помолился ?не , я тоже понимаю,что это объезд территории без свиты. у него нет домового храма,чт	0
18842	Придет время , когда и ваше видимое благополучие начнет медленно исчезать (как уже было !) , а вместе с тем и спесь , сквозящая с ка	0
18843	аким образом, только потери физлиц могут составить до 3,5 миллиардов евро. Дочки российских банков и российские компании, зареги	0
18844	т ее переизбытка надо порой порой освобождаться.За неимением причин защищать свое имущество, на которое никто не покушался, м	0
18845	- Ну, и про самоисплнимые договоры тож)))) Да я много чего могу рассказать. — Теперь, говорю, понятно, отчего в прошлом году сош	0
18846	Их надо гуманно лечить.По либеральному, по европейски!Эвтаназией :) Тогда и вас туда же. Потому что вы сейчас троллите. а вот и тро	0
18847	Как и во время господства в Китае маньчжурской династии такое тоже могло быть... В этот период они, например, захватили ту террито	0
18848	очень правильно и грамотно поступали: заставляли священников самим вскрывать раку,снимать облачения и покровы. доставать оттуда	0
18849	Доброго,хорошего утра!) Желто-зеленые!!!Желто-зеленые!!!Желто-зеленые!!!Желто-зеленые!!!Желто-зеленые!!!Желто-зеленые!!!	1
18850	ердюков им воровать не давал, его классически и подставили, но не подрассчитали, что подстава чиновника такого уровня приведет к вз	0
18851	ама не курю, если что.) считаю это нарушением прав человека)(Их раздражает курево, а меня раздражают они. И чем они лучше меня	-1

Рисунок 2.1 - Приклад текстів

2.3 Попередня обробка текстів

Повідомлення в соціальних мережах містять інформацію, що не є необхідною для аналізу тональності:

- Хештег - являє собою слово або об'єднання слів, яким передує символ #, наприклад, # спорт, # Пітер. Використовується для полегшення пошуку повідомлень по темі або змісту.
- Посилання - універсальний покажчик ресурсу.
- Ім'я користувача - унікальне позначення користувача в соціальної мережі. У Twitter ім'я користувача починається з символу @, наприклад, @username.

Використовуючи регулярні вирази, відповідні дані були видалені з текстів. Також були вилучені знаки пунктуації та усі великі букви приведені до рядкових.

2.3.1 Стоп-слова

Стоп-слова - слова, які не несуть смислового навантаження. До них відносять: союзи, займенники, прийменники, частки, вигуки, цифри. В разі аналізу тональності тексту, такі слова не тільки не несуть смислового навантаження, але і засмічують словник.

У задачах класифікації видалення таких слів у багатьох випадках дозволяє поліпшити якість алгоритму [40]. Також було вирішено приводити слова до словникової форми.

Словникова форма слова - одна з словоформ, яка зазвичай збігається з початковою формою слова, наприклад, в російській мові словарними формами іменників є форми І.П. однини, прикметників - І.П. однини чоловічого роду, дієслів - інфінітив.

Це не тільки підвищує якість алгоритму, а й зменшує обсяг словника.

Для визначення частин мови і початкової форми слова використовувалася бібліотека mystem-scala.

Mystem-scala виконує морфологічний аналіз тексту російською мовою. Доступна для Java, Scala і інших мов на платформі JVM (Java Virtual Machine). Застосовується як для некомерційного, так і для комерційного використання.

Приклад використання mystem-scala для речення: «У тихому болоті чорти водяться».

Таблиця 2.1

Результат морфологічного аналізу пропозиції

Слово	Частина мови	Морфологічні ознаки	Початкова форма
В	PR		в
тихому	A		тихий
болоті	S	M, abl, sg	Болото
чорти	S	M, nom, pl	чорт
водяться	V	Ipf, pl, indic, 3p	водитися

Розшифровка граммем, представлених в таблиці 2.1:

- PR - прийменник,
- A - прикметник,
- S - іменник,
- V - дієслово,
- m - чоловічий рід,
- abl - місцевий відмінок,
- sg - однина,
- nom - називний відмінок,
- pl - множина,
- ipf - недосконалий вид,
- indic - дійсний спосіб,
- 3p - 3-я особа дієслова.

Розшифровка всіх граммем представлена на офіційному сайті Mystem [41].

2.3.2 Слова з максимальною частотою зустрічання

У даній роботі був складений частотний словник, в який спочатку увійшли 100 слів з максимальною частотою зустрічальності в російській мові згідно «частотних словників сучасної російської мови» [42], який був побудований на основі «Національного корпусу російської мови» [43].

Але так як більшість слів в складеному словнику виявилася займенниками, прийменниками, союзами, які видаляються за допомогою mystem-scala, то словник був скорочений до 29 слів. Приклад таких слів зображено у таблиці 2.2.

Таблиця 2.2

Частота використання слів

Слово	Частота використання
Быть	12160.7
год	3727.5
Мочь	2912.3
Человек	2723.0
Говорить	2396.6
Еще	2323.9
Уже	2179.2
Время	2015.7
Сказать	1755.0
Знать	1713.8
Справа	1412.1
Жизнь	1389.8
День	1258.4
новый	1217.5

2.4 Обробка смайликів

Смайлик (смайл) - спосіб вираження емоцій у дописах. В тому числі, вони допомагають позбутися неоднозначності [44].

Був складений словник смайликів, в який увійшли (рисунок 2.3):

- найпопулярніші текстові смайли [45];
- смайли з максимальною частотою в навчальній вибірці.

Для кожного з них був проставлений тип емоційного забарвлення (позитивний, негативний).

2.5 Словник негативних слів

У соціальних мережах іноді вживають смайли з відмінною забарвленістю від тексту, що призводить до неоднозначності визначення тональності.

Наприклад, «Це якийсь жах!) Кошмар!)». В даному прикладі текст має явно негативне забарвлення, а смайлики - позитивну. Тому було вирішено створити словник негативних слів, в якому всі слова приведені до словникової форми.

Тоді алгоритм буде наступний:

- якщо текст негативний і містить слова з негативного словника, то незважаючи на смайлики, він буде віднесений до негативного класу тональності;
- якщо текст негативний і не містить слів з негативного словника, але включає позитивні смайлики, то він буде віднесений до позитивного класу тональності.

На рисунку 2.4 наведено приклад слів з негативного словника.

Смайл	Тональность	Смайл	Тональность
:-)	положительный	:o)	положительный
:-D	положительный	;)	положительный
;-)	положительный	;v)	положительный
xD	положительный	:D	положительный
;-P	положительный	:^D	положительный
:-p	положительный	:/	негативный
8-)	положительный	:-b	положительный
B-)	положительный	=^*	положительный
:-(	негативный	:-x	положительный
;-]	положительный	8-]	положительный
3(	негативный	>:-(	негативный
:'(	негативный	>:-[	негативный
:_(	негативный	:-0	негативный
:((	негативный	:-o	негативный
:o	негативный	:,-(	негативный
3-)	положительный	:'-(	негативный
O:)	положительный	:,-)	положительный
;o	негативный	:<	негативный
<3	положительный	:-||	негативный
:-*	положительный	D;	негативный

Рисунок 2.3 - Приклад смайликів в словнику

В ході дослідження було з'ясовано: якщо наївний байєсовський класифікатор відносить текст з імовірністю більшою або рівною 0.8 до позитивного класу тональності, то можна не враховувати до якого класу тональності належать смайлики, так як текст буде явно позитивний.

Наприклад: «Сьогодні чудовий день! (». В даному реченні користувач швидше за все помилився при вказівці смайлика; якщо наївний байєсовський класифікатор відносить текст з ймовірністю меншою або рівній 0.45 до позитивного або негативного класу, і текст не містить позитивних або негативних смайликів, то він має нейтральне забарвлення.

Наприклад, твіт: «привеет. Як справи?» Не містить смайликів і слів з явним емоційним забарвленням, тому його можна віднести до нейтрального класу тональності.

Приклад слів із негативного словника:

- безысходность;
- отстой;
- ужас;
- ужасный;
- глупость;
- гадость;
- подлость;
- гнев;
- тупой;
- идиот.

2.6 Оцінка ефективності алгоритму

При оцінці більшості алгоритмів вилучення інформації використовуються такі терміни як точність, повнота і доля правильних відповідей *Accuracy*.

Точність в межах класу - частка текстів, що дійсно належать даному класу, щодо всіх текстів, які класифікатор відніс до цього класу.

Повнота системи - частка знайдених класифікатором текстів, що належать класу, щодо всіх текстів цього класу в тестовій вибірці.

В результаті класифікації текстів рецензій тестової вибірки, до класу позитивних рецензій правильно віднесені TP текстів, неправильно – FP, до класу негативних рецензій правильно були віднесені TN текстів, неправильно – FN. Іншими словами:

- TP – істинно-позитивне рішення;
- TN – істинно-негативне рішення;
- FP – хибно-позитивне рішення;
- FN – хибно-негативне рішення.

$$\text{Точність} = \frac{TP}{TP+FP} \tag{2.1}$$

$$\text{Повнота} = \frac{TP}{TP+FN} \tag{2.2}$$

$$Accuracy = \frac{TP+TN}{P+N} \tag{2.2}$$

2.7 Модифікований алгоритм визначення тональності текстів

У даній роботі емоційне забарвлення текстів ділиться на класи: позитивна, негативна, нейтральна. Провести оцінку ефективності алгоритму можна тільки в тому випадку, якщо кількість класів тональності на вході і виході буде однаковою.

Текст розбивається на уніграмми і біграми в разі заперечення, наприклад, «не люблю».

Оскільки в якості алгоритму був обраний наївний байєсовський класифікатор, то для визначення тональності необхідно враховувати наступні параметри: кількість позитивних і негативних повідомлень в корпусі, кількість позитивних і негативних слів (словосполучень) в корпусі. Нижче представлені діаграми алгоритмів: «Підготовка навчальної вибірки» (рисунок 2.4) і «Визначення тональності тексту і заповнення словника» (Рисунок 2.5, Рисунок 2.6).

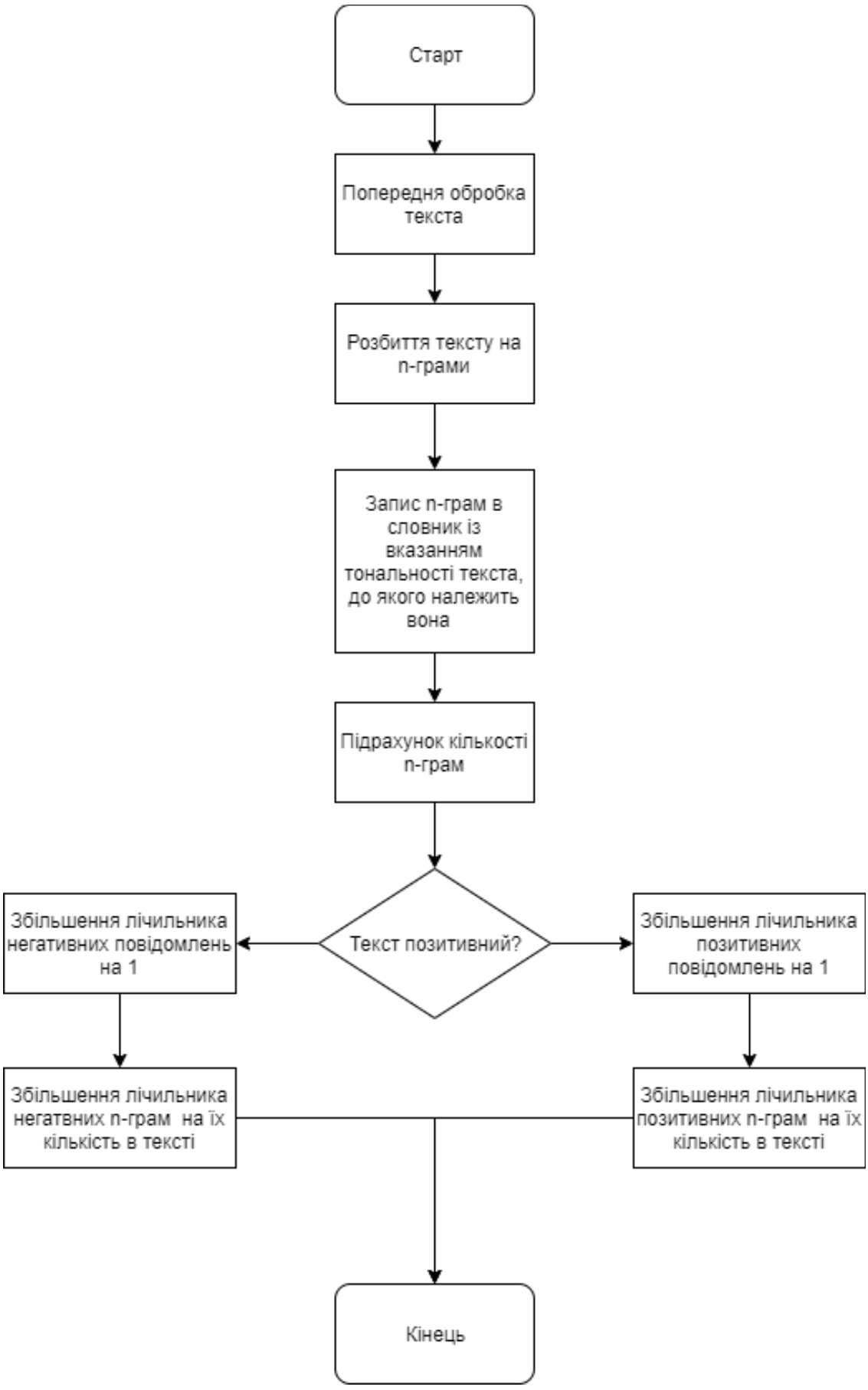

Рисунок 2.4 - Підготовка навчальної вибірки (для кожного тексту)

Навчання класифікатора передбачає заповнення словника, який включає в себе: n-грами, тональність n-грами, скільки конкретна n-грама зустрічалася в текстах кожного класу тональності. Для цього необхідно попередньо обробити кожен текст: видалити посилання, хештеги, імена користувачів, розділові знаки, привести слова до початкової формі і т.д.

Другий етап - розбиття тексту на n-грами та запис їх в словник з урахуванням тональності, до якої належить текст.

Третій етап - підрахунок кількості n-грам в тексті.

Четвертий - збільшення лічильника кількості n-грам відповідно до кількості, підрахованої в третьому етапі з урахуванням класу тональності, до якої належить текст, і в тому ж класі тональності збільшення лічильника кількості текстів на одиницю.

Діаграма «Визначення тональності тексту і заповнення словника» (Рисунок 2.5, 2.6) майже ідентична діаграмі «Підготовки навчальної вибірки». В першу ще додається етап визначення тональності тексту.

2.8 Тестування ефективності модифікацій алгоритму

Для тестування модифікованого алгоритму тональності текстів був використаний метод крос-валідації (ковзний контроль). Процедура крос-валідації включає наступні етапи:

- Визначаються дві підвибірки: навчальна і тестова.
- Навчаємо класифікатор тональності за допомогою навчальної вибірки даних і виконуємо тестування на тестовій.
- Визначимо середнє значення оцінок ефективності алгоритму на тестовій підвибірці за допомогою визначення точності і повноти.

Для тестування було взято 100 позитивних і 100 негативних текстів для тестування із навчальної вибірки.

Оцінено ефективність алгоритму, в якому попередня обробка текстів включала: видалення хештегів, посилань і імен користувачів, знаків пунктуації; усі великі літери приведені до рядкових.

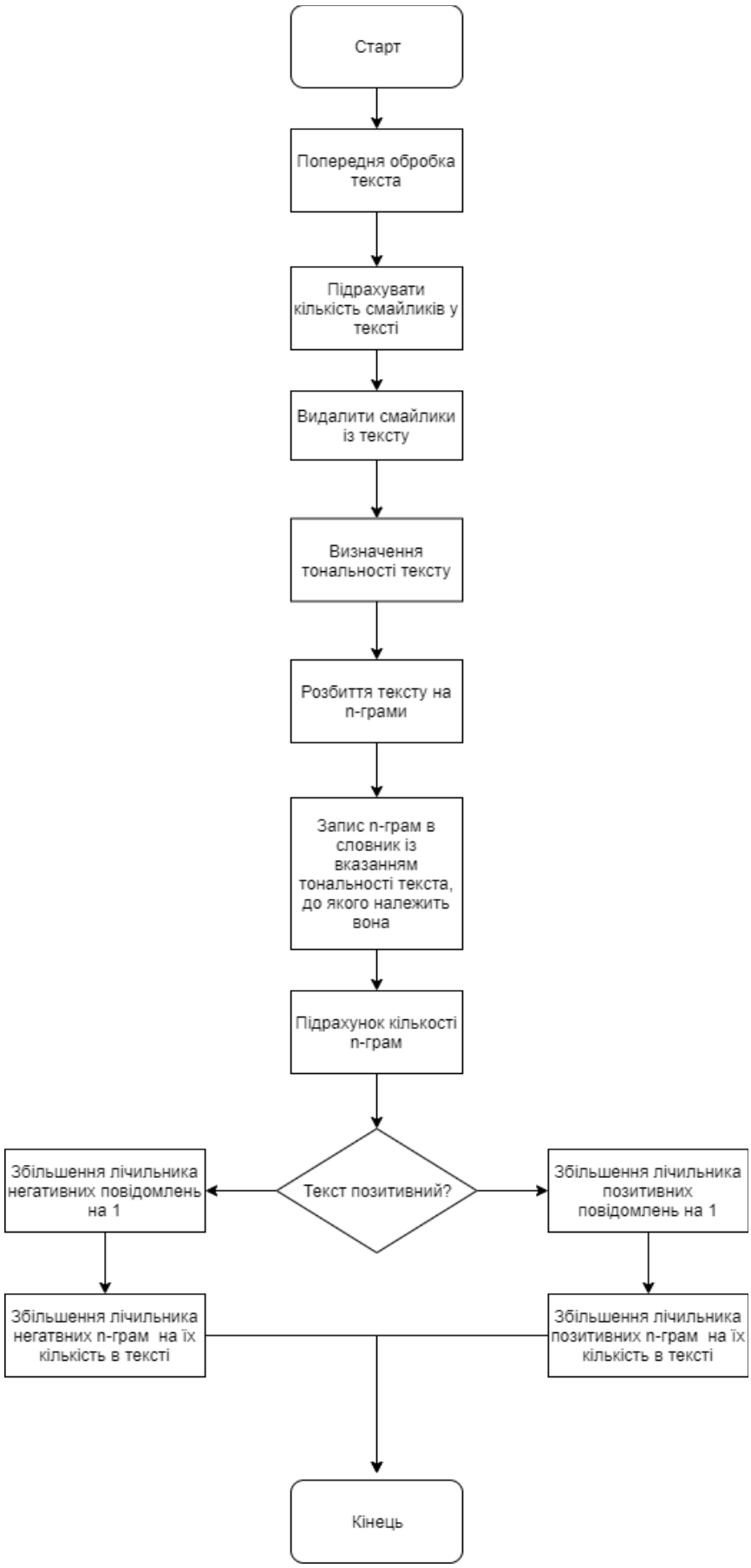

Рисунок 2.5 - Визначення тональності тексту і заповнення словника

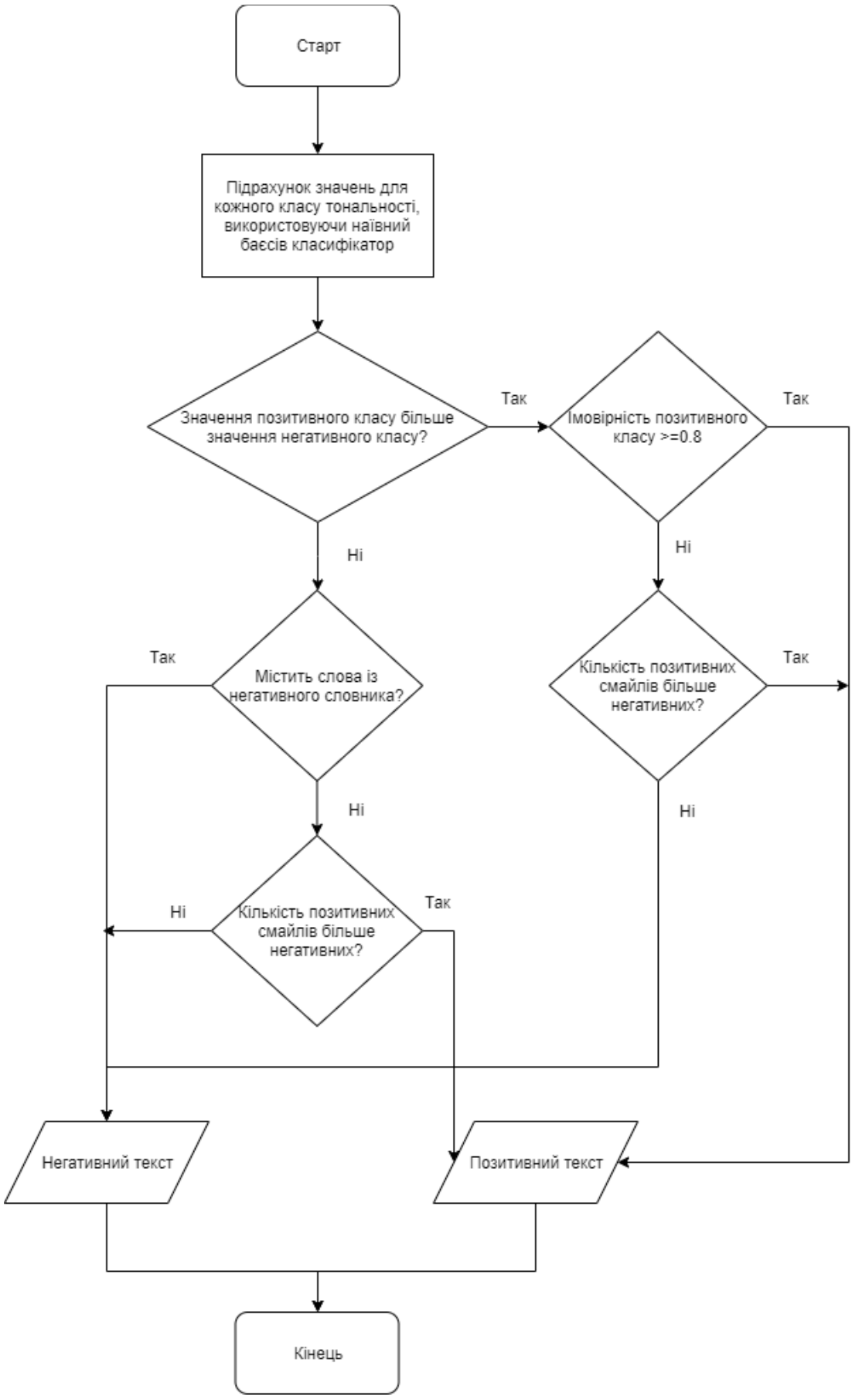

Рисунок 2.6 - Визначення тональності тексту

Результати тестової вибірки відображено у таблиці 2.3.

Таблиця 2.3

Оцінка ефективності модифікованого алгоритму тональності

Міри N-грами	Точність, %	Повнота, %
Уніграми	81,7	84,3
Біграми і уніграми	82,3	85,5

Наївний байєсовський класифікатор показав дуже хороші результати. Найбільш ефективним виявилося застосування поєднання уніграмм і біграмм. Але так як застосування лише уніграмм незначно поступається, то можна зробити висновок, що обидва варіанти реалізації ефективні.

Спробуємо поліпшити результати, додавши до попередньої обробки текстів приведення слів до словникової форми і видалення стоп-слів, які не впливають на емоційне забарвлення. Нижче представлений графік співвідношення стоп слів до решти слів з тестової виборки (рисунок 2.7)

Рисунок 2.7 - Графік співвідношення стоп-слів до решти слів з тестової виборки

Після видалення стоп-слів було отримано результати показані у таблиці 2.4.

Таблиця 2.4

Оцінка ефективності модифікованого алгоритму тональності

Міри N-грами	Точність, %	Повнота, %
Уніграми	83,3	86,7
Біграми і уніграми	84,9	88,4

З результатів оцінки ефективності алгоритму в таблиці 2.2 і 2.3 видно, що видалення стоп-слів покращує результати класифікатора.

Для покращення роботи треба додати до попередньої обробки текстів видалення слів з максимальною частотою зустрічання замість видалення стоп-слів.

Таблиця 2.5

Оцінка ефективності модифікованого алгоритму тональності

Міри N-грами	Точність, %	Повнота, %
Уніграми	82,1	85,9
Біграми і уніграми	83,3	87,5

Як видно з таблиці 2.5 видалення тільки слів, з максимальною частотою зустрічання, показало гірші результати, ніж видалення стоп-слів. Тому було вирішено в попередній обробці текстів видаляти стоп-слова.

Результати кінцевого алгоритму представлені в таблиці 2.6.

Таблиця 2.6

Оцінка ефективності модифікованого алгоритму тональності

Міри N-грами	Точність, %	Повнота, %
Уніграми	83.8	87.1
Біграми і уніграми	85	89,1

За результатами тестування модифікованого алгоритму можна зробити наступні висновки:

- застосування поєднання біграмм і уніграмм ефективніше ніж застосування одних уніграмм;
- в попередню обробку текстів має увійти видалення слів з максимальною частотою зустрічальності, приведення всіх слів до словникової форми і видалення стоп-слів.

2.9 Висновок

В даному розділі були розглянуті способи поліпшення алгоритму тональності текстів на основі наявного байєсовського класифікатора.

Зроблені оцінки ефективності декількох варіантів алгоритму, по результатам яких був розроблений модифікований алгоритм тональності текстів.

3. РОЗРОБКА ПРОГРАМИ ДЛЯ АНАЛІЗУ МОВНИХ ЗАБАРВЛЕНЬ

3.1 Функціональні вимоги

Функціональні вимоги до системи складаються із наступних пунктів:

- Можливість відкриття текстового файлу для аналізу тексту
- Відображення мовного забарвлення даного тексту
- Зберігання історії перевірок текстів впродовж сесії

Користувач може використовувати текст будь-якої довжини для визначення його забарвлення. Данна можливість матиме найбільшу перевагу при маштабних текстах, які людина буде перевіряти довгий час.

Крім того, у головному вікні програми відображається історія аналізу текстів користувачем впродовж його сеансу роботи.

3.2 Обґрунтування вибору платформи та мови програмування

Система включає в себе наступні підсистеми:

- візуалізація і робота з користувачем;
- збереження даних сеансу;
- обробка даних.

В якості мов та інструментів були обрані java, python, tkinter, scikit-learn. Для програми, що розроблюється було обрано простий і зручний інтерфейс, що дозволить користувачеві відразу взаємодіяти із програмою на інтуїтивному рівні. Саме тому для цього було обрано мову python, яка відзначається простотою і швидкодією.

Суттєвим чинником при виборі мови python у якості основної мови для обробки даних та машинного навчання була наявність суттєвих наробків у сфері машинного навчання саме на цій мові, наприклад програмні бібліотеки scikit-learn, що була використані мною у магістерській роботі. Також, python був обраний через свою популярність і відносно високу швидкодію.

Бібліотека scikit-learn - найпоширеніший вибір для вирішення завдань класичного машинного навчання. Вона надає широкий вибір алгоритмів навчання з учителем і без вчителя. Навчання з учителем передбачає наявність розміченого датасета, в якому відомо значення цільового показника. У той час як навчання без учителя не припускав наявності розмітки в датасета - потрібно навчитися отримувати корисну інформацію з довільних даних. Одне з основних переваг бібліотеки полягає в тому, що вона працює на основі декількох поширених математичних бібліотек, і легко інтегрує їх один з одним. Ще однією перевагою є широка спільнота і докладна документація. Scikit-learn широко використовується для промислових систем, в яких застосовуються алгоритми класичного машинного навчання, для досліджень, а так само для новачків, які тільки робить перші кроки в області машинного навчання.

Python — це інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою, що була розроблена в 1990 році Гвідо ван Россумом. Реалізовані структури даних високого рівня та динамічна семантика, динамічне зв’язування, автоматичне управління пам’яттю збільшують її зручність для швидкої розробки програм, а також для поєднання існуючих компонентів. Python підтримує модулі та пакети модулів, що сприяє повторному використанню коду.

3.3 Обробка попередніх даних

Для реалізації цієї програми необхідно провести попередню обробку отриманих даних, яка полягає у визначенні тональності тексту.

Попередня обробка тексту включає:

- видалення хештегів і посилань рисунок 3.1;

```
tweet = tweet.replaceAll(
    "((http|https|ftp):\\S+)|" +
    "(@[A-Za-z-\\_]|"+
    "(\\d+{A-Za-z-]*)|" +
    "(#[A-Za-z-\\_]+)|","")
);
```

Рисунок 3.1 - Видалення хештегів, посилань, імен користувачів

- підрахунок смайликів і їх видалення (рисунок 3.2);

```
while(resultQuery.next()) {
    emoticon = resultQuery.getString(1).trim();
    int markOfEmoticon = resultQuery.getInt(2);
    if(tweet.contains(emoticon)){
        countEmoticons = StringUtils.countMatches(tweet, emoticon);
        mark= mark +(countEmoticons*markOfEmoticon);
        tweet = tweet.replace(resultQuery.getString(1).trim(),"");
        }
        }
        statement.close();

        Pattern openingBracket = Pattern.compile("\\(");
        Matcher matcherOpeningBracket = openingBracket.matcher(tweet);
        while(matcherOpeningBracket.find()) countOpeningBracket++;

        Pattern openingBracket = Pattern.compile("\\)");
        Matcher matcherOpeningBracket = openingBracket.matcher(tweet);
        while(matcherClosingBracket.find()) countClosingBracket++;
    if(countOpeningBracket>countClosingBracket || countOpeningBracket<
    countClosingBracket)
        tweet = tweet.replace("(","");
        mark = mark + (-1)*countOpeningBracket;
        tweet = tweet.replace(")","");
        mark = mark + *countClosingBracket;
        }
```

Рисунок 3.2 - Підрахунок і видалення смайликів

- видалення знаків пунктуації, замість «ё» записуємо «е», наводимо текст до нижнього регістру (рисунок3.3);

```
String tweet = text.replaceAll("\\p{Punct}"," ").
    replaceAll("[\\.\\-\\-\\*\\'\\+]+"," ").
    replace("ë", "e").
    replaceAll("\\s{2,}"," ").toLowerCase();
```

Рисунок 3.3 - Видалення знаків пунктуації, приведення тексту до нижнього регістру

- видаляємо слова, які не впливають на емоційне забарвлення тексту за допомогою бібліотеки Mystem-scala.

```
Pattern pattern =Pattern.compile("(^ADVPRO.*)|(^ANUM.*)|(^APRO.*)|(^CONJ.*)|+
    (^NUM.*)|(^PART.*)|(^PR.*)|(^SPRO.*)|(^S\\.persn.*)|(^S\\.patrn.*)");

ArrayList<String> arrayWords = new ArrayList<~>();

Iterable<info> result=null;
try{
result= JavaConversions.asJavaIterable(mysteamAnalyzer.analyze(Request.
                    apply(tweet)).info().toIterable()
    );
}catch (MySteamApplicationException e){
e.printStackTrace();
}
```

Рисунок 3.4 - Видалення слів, які не впливають на емоційне забарвлення тексту

Більш докладний алгоритм визначення тональності текстів описаний в розділі 2.

3.4 Візуалізація і робота з клієнтом

Як інструмент візуалізації і роботи з клієнтами був обраний спеціальний набір компонентів tkinter.

Tkinter - це багато-платформна графічна бібліотека інтерфейсів на основі засобів Tk (широко розповсюджена у світі GNU/Linux та інших UNIX подібних систем, портована в тому числі і на Microsoft Windows, Apple Mac OS), поширювана з відкритими вихідними текстами, написана Стіном Лумхольтом

(Steen Lumholt) і Гвідо ван Россумом. Входить в стандартну бібліотеку Python. [51].

Під графічним інтерфейсом користувача (GUI) маються на увазі всі ті вікна, кнопки, текстові поля для введення, скролери, списки, радиокнопки, прапорці та ін., Які ви бачите на екрані, відкриваючи ту чи іншу програму. Через них ви взаємодієте з програмою і керуєте нею. Всі ці елементи інтерфейсу разом будемо називати віджетами (widgets).

Додатки з графічним інтерфейсом користувача подієво-орієнтовані. Тобто та чи інша частина програмного коду починає виконуватися лише тоді, коли трапляється та чи інша подія.

Подієво-орієнтоване програмування базується на об'єктно-орієнтованому і структурному. Навіть якщо ми не створюємо власних класів та об'єктів, то все-одно ними користуємося. Всі віджети - об'єкти, породжені вбудованими класами.

Події бувають різними. Спрацював часовий чинник, хтось клікнув мишкою або натиснув Enter, почав вводити текст, перемкнув радіокнопки, прокрутив сторінку вниз і т. д. Коли трапляється щось подібне, то, якщо був створений відповідний обробник, відбувається спрацьовування певної частини програми, що приводить до якого-небудь результату.

Щоб написати GUI-програму, треба виконати наступне:

- Створити головне вікно.
- Створити віджети і конфігурувати їх властивостей (опцій).
- Визначити події, тобто те, на що буде реагувати програма.
- Визначити обробники подій, тобто те, як буде реагувати програма.
- Розташувати віджети в головному вікні.
- Запустити цикл обробки подій.

Для створення графічного додатку потрібно виконати наступні команди, зображені на рисунку 3.5.

```
root = tk.Tk()
root.title("Определение тональности")
frame = tk.Frame(root)
button_file = tk.Button(
    frame, text="Открыть файл", command=UploadAction)
button_action = tk.Button(
    frame, text="Определить тональность", command=Calculation)
button_clear = tk.Button(
    frame, text="Очистить", command=clear)
frame.pack()
label = tk.Label(frame, text="Тональность(-1, 0, 1): ")
button_file.grid(row=1, column=1)
button_action.grid(row=1, column=2)
button_clear.grid(row=1, column=3)
label.grid(row=2, column=2)
root.mainloop()
```

Рисунок 3.5 - Видалення слів, які не впливають на емоційне забарвлення тексту

3.5 Тренування класифікатора

Нехай у нас є рядок тексту O. Крім того, є класи C, до одного з яких ми повинні віднести рядок. Нам необхідно знайти такий клас c, при якому його ймовірність для цього рядка була б максимальна. Математично це записується за допомогою формули 3.1:

$$c = \arg\max_{C} P(C|O) \tag{3.1}$$

Обчислити P (C | O) складно. Але можна скористатися теоремою Байеса і перейти до непрямих можливостей:

$$P(C|O) = \frac{P(O|C)P(C)}{P(O)} \tag{3.2}$$

Так як ми шукаємо максимум від функції, то знаменник нас не цікавить (він вданому випадку константа). Крім того, потрібно поглянути на рядок O. Зазвичай, немає сенсу працювати з усією рядком. Набагато ефективніше виділити з неї певні ознаки (features). Таким чином формула набуде вигляду наступного вигляду 3.3:

(3.3)

$$P(C|o_1o_2...o_n) = \frac{P(o_1o_2...o_n|C)P(C)}{P(o_1o_2...o_n)}$$

Але це знову складно. Тут включаємо «наївне» припущення про те, що змінні O залежать тільки від класу C, і не залежать один від одного. Це сильно спрощення, але найчастіше це працює. Чисельник набуде вигляду:

$$P(C)P(o_1|C)P(o_2|Co_1)...P(o_n|Co_1o_2...o_n) = P(C)P(o_1|C)P(o_2|C)..P(o_n|C) = P(C)\prod_i(o_i|C)$$

(3.3)

Фінальна формула набуде вигляду показаного на формулі 3.4:

$$c = \arg\max_{c\in C} P(c|o_1o_2...o_n) = \arg\max_{c\in C} P(c)\prod P(o_i|c)$$

(3.3)

Тобто все що потрібно зробити, це обчислити ймовірності P (C) і P (O | C). Обчислення цих параметрів і називається тренуванням класифікатора.

Програмно це реалізовано при використанні бібліотеки sklearn наступним чином, що показано на рисунку 3.6:

```
# открытие файла
df = pd.read_excel("doc_comment_summary.xlsx",
                   header=None, names=["text", "value"])

# удаления не подходящий значений
df.iloc[:, 1] = df.iloc[:, 1].apply(correct_values)
df.iloc[:, 0] = df.iloc[:, 0].apply(correct_text)
df = df[df.value < 3]
df = df[df.text != 100]

# разделение данных на данные для тренироваки и данные для тестирования
X_train, X_test, y_train, y_test = train_test_split(
    df.iloc[:, 0], df.iloc[:, 1], test_size=0.25, random_state=0)
# разные варианты токенизации текста: (tokenizer=tokenize_ru, ngram_range=(1,3))
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(X_train)  # перевод
X_test = vectorizer.transform(X_test)
nb = MultinomialNB()

# тренировка модели
model = nb.fit(X_train, y_train)
```

Рисунок 3.6 - Графік співвідношення стоп-слів до решти слів з тестової виборки

Так як тренування моделі займає тривалий час, було вирішено зберегти готову модель, для подальшого її імпорту у готову програму за допомогою наступного методу показаного на рисунку 3.7.

```
# запись готовой модели и векторизации
pickle.dump(model, open("model.sav", 'wb'))
pickle.dump(vectorizer.transform, open("vectorizer.sav", 'wb'))
```

Рисунок 3.7 - Графік співвідношення стоп-слів до решти слів з тестової виборки.

3.6 Огляд додатку

На рисунку 3.8 представлена головна сторінка додатка, на якій користувач буде працювати із програмою.

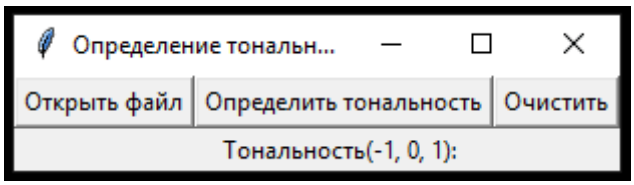

Рисунок 3.8 - Головна сторінка додатка

Користувач може відкрити файл натиснувши відповідну кнопку і вибрати текстовий документ із попередньо записаним текстом для аналізу. Далі необхідно натиснути кнопку «Определить тональность» після чого програма визначить мовне забарвлення вашого тексту.

Впродовж сеансу програма запам’ятовує ваші перевірки і відображає статистику внизу під клавішами.

3.7 Висновок

В даному розділі в повній мірі реалізовано додаток аналізу тональності текстів. Також описана архітектура додатка і приведений список використовуваних технологій.

4. ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть Арсенюк І. Р. та Богач І. В.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 5.1.

Таблиця 4.1

Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Експерт 1	2. Експерт 2
	Бали, виставлені експертами:	
1	4	4
2	3	3
3	3	4
4	4	3
5	3	3
6	4	4
7	3	3
8	4	4
9	3	3
10	4	3
11	3	4
12	3	4
Сума балів	$СБ_1 = 42$	$СБ_2 = 42$

Продовження таблиці 4.1

Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{1}^{3} СБ_i}{2} = 42$

Отже, з отриманих даних таблиці 5.1 видно, що нова розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.

Для розробки нового програмного продукту необхідні такі витрати. Основна заробітна плата для розробників визначається за формулою (5.1):

$$З_о = \frac{М}{Т_р} \cdot t, \qquad (4.1)$$

де М- місячний посадовий оклад конкретного розробника;

$Т_р$ - кількість робочих днів у місяці, $Т_р$ = 21 день;

t - число днів роботи розробника, t = 40 днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 5.2.

Таблиця 4.2

Розрахунки основної заробітної плати

Працівник	Оклад М, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	5500	261,90	8	2095,2

Продовження таблиці 4.2

Інженер-програміст	3500	166,66	40	6666,4
Всього:				8761,6

Розрахуємо додаткову заробітну плату:

$$З_{дод} = 0,1 \cdot 8761,6 = 876,16 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5… 40% від суми їхньої основної та додаткової заробітної плати:

$$Нзп = (Зо + Зр) \cdot \frac{\beta}{100}, \qquad (4.2)$$

$$Нзп = (8761,6+876,16) \cdot \frac{36,3}{100} = 3498,50 (грн.).$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$А = \frac{Ц \cdot Н_а}{100} \cdot \frac{Т}{12}, \qquad (4.3)$$

де Ц – балансова вартість обладнання, грн;

$Н_а$ – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

Т – Термін використання (Т=3 міс.).

Таблиця 4.3

Розрахунок амортизаційних відрахувань

Найменування програм	Балансова вартість,	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань,грн
Персональний	9000	25	3	562,5
Всього:				562,5

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_{1}^{n} H_i \cdot Ц_i \cdot K_i, \qquad (4.3)$$

де n – кількість комплектуючих;
Ні - кількість комплектуючих і-го виду;
Ці – покупна ціна комплектуючих і-го виду, грн;
Кі – коефіцієнт транспортних витрат (приймемо Кі = 1,1).

Таблиця 4.4

Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	шт.	200	1	200
Пачка паперу	уп.	120	1	120
Ручка	шт.	5	1	5
Всього з урахуванням транспортних витрат				357,5

Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п \quad ; \qquad (4.5)$$

де В – вартість 1кВт-години електроенергії (В=1,7 грн/кВт);
П – установлена потужність комп'ютера (П=0,6кВт);
Ф – фактична кількість годин роботи комп'ютера (Ф=175 год.);
K_n – коефіцієнт використання потужності ($K_n < 1$, $K_n = 0{,}9$).

$$В_е = 1{,}7 \cdot 0{,}6 \cdot 175 \cdot 0{,}9 = 160{,}65 \text{ (грн.)}$$

Розрахуємо інші витрати $В_{ін}$.

Інші витрати $І_в$ можна прийняти як (100…300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$B_{ін} = (1..3) \cdot (З_о + З_р). \qquad (4.6)$$

Отже, розрахуємо інші витрати:

$$B_{ін} = 1 * (8761,6 + 876,16) = 9637,76 \quad \text{(грн).}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$B = З_о + З_д + Н_{зп} + А + К + В_е + І_в \qquad (4.7)$$

В = 8761,6+876,16+3498,5+562,5+357,5+160,65+9637,76= 23854,67 (грн.)

Розрахуємо загальну вартість наукової роботи $B_{заг}$ за формулою:

$$B_{заг} = \frac{B_{ін}}{\alpha} \qquad (4.8)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$B_{заг} = \frac{23854,67}{1} = 23854,67$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{заг}}{\beta} \qquad (4.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{23854,67}{0,9} = 26505,18 \quad \text{(грн.)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності .

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta П_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta П_i = \sum_{1}^{n} \begin{matrix} \Delta N \\ ¿\Delta П_я \cdot N + П_я ¿ \\ ¿ \end{matrix} \qquad (4.10)$$

де $\Delta П_я$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$П_я$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технології зменшаться на 20 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 20 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 180 користувачів, протягом другого року – на 160 користувачів, протягом третього року – 140 користувачів. Реалізація інформаційної технології до впровадження результатів наукової розробки складала 500 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 200 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $\Delta П_1$ протягом першого року складатиме:

$$\Delta П_1 = 20 \cdot 500 + (200 + 20) \cdot 180 = 49600\, грн.$$

Протягом другого року:

$$\Delta П_2 = 20 \cdot 500 + (200 + 20) \cdot (180 + 160) = 84800\, грн.$$

Протягом третього року:

$$\Delta П_3 = 20 \cdot 500 + (200 + 20) \cdot (180 + 160 + 140) = 115600\, грн.$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{абс}$ вкладених інвестицій розраховується за формулою:

$$E_{абс} = (ПП - PV), \qquad (4.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3,4.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 5.1.

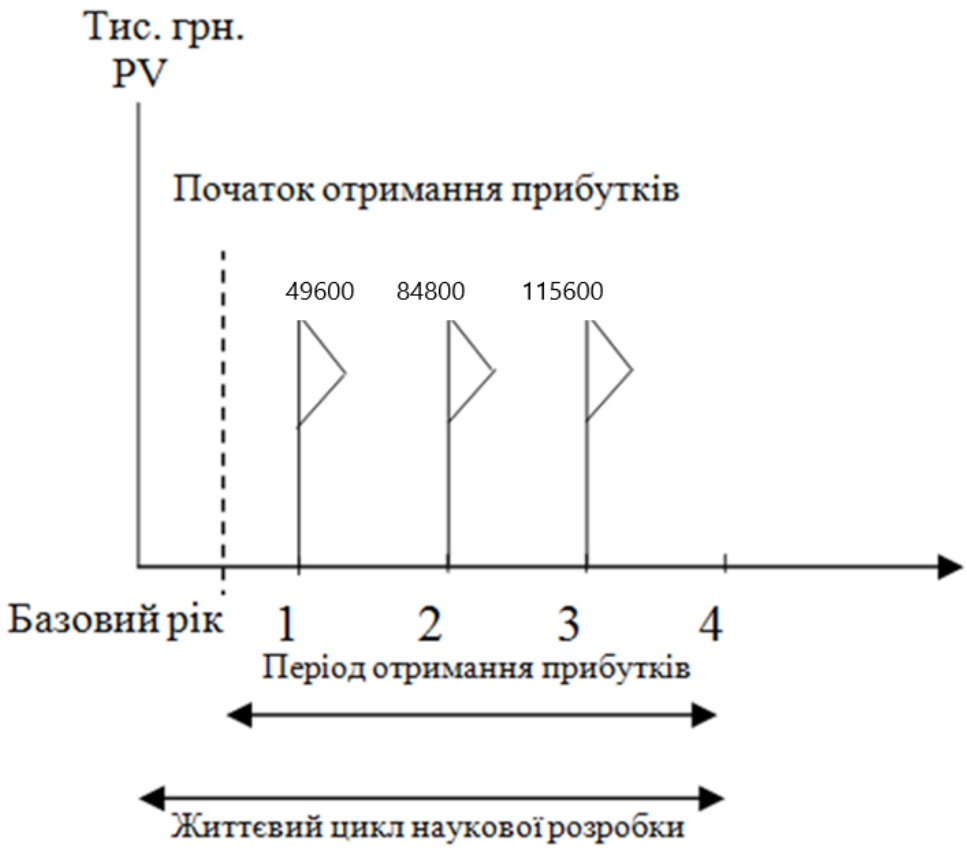

Рисунок 5.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою:

$$ПП = \sum_{1}^{m} \frac{\Delta П_i}{1+\tau^t} \qquad (4.12)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{26505,18}{1+0,1^0} + \frac{}{1+0,1^2} + \frac{}{1+0,1^3} + \frac{}{1+0,1^4} \quad \text{(грн.)}$$

Тоді розрахуємо $E_{абс}$:

$$E_{абс} = 210164,75 - 26505,18 = 183659,57 \text{ грн.}$$

Оскільки $E_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$ за формулою:

$$E_в = \sqrt[T]{1 + \frac{E_{абс}}{PV}} - 1 \qquad (4.13)$$

де
$E_{абс}$ – абсолютна ефективність вкладених інвестицій, грн;
PV – теперішня вартість інвестицій PV = ЗВ, грн;
$T_ж$ – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_в = \sqrt[3]{1 + \frac{183659,57}{26505,18}} - 1 = 0,99 \text{ або } 90\ \%$$

Далі, розраховану величина $E_в$ порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{мін}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою:

$$\tau = d + f \ , \qquad (4.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні d = 0,2;
f – показник, що характеризує ризикованість вкладень, величина f = 0,1.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки $E_в$ = 99% > $\tau_{мін}$ = 0,3 = 30%, то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_в} \qquad (4.15)$$

$$T_{ок} = \frac{1}{0{,}99} = 1{,}01 \; року$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

4.5 Висновок

При складанні бізнес-плану по комерціалізації результатів НДР даної ВКР, були вироблені практичні навички проведення розрахунку собівартості на прикладі певної розробки.

ВИСНОВОК

Метою випускної кваліфікаційної роботи була розробка додатку для аналізу мовного забарвлення текстів.

Для досягнення зазначеної мети були вирішені наступні завдання:

- проаналізовано існуючі підходи і методи класифікації тональності текстів;
- розроблений алгоритм для автоматичного визначення тональності тексту;
- розроблено додаток для аналізу тональності текстів;
- складено бізнес-план по комерціалізації результатів науково-дослідної роботи.

На підставі проведеної роботи були зроблені наступні висновки:

1. Машинне навчання з учителем ефективніше, ніж підхід на правилах, словниках або машинне навчання без учителя. Як показали дослідження для найкращої ефективності необхідно комбінувати кілька підходів. У даній роботі використовувалися машинне навчання з учителем і підхід на словниках.
2. Найбільш ефективними при визначенні тональності коротких тестових повідомлень себе показали: наївний байєсовський класифікатор і метод опорних векторів. Однак у методу опорних векторів є істотний недолік: при збільшенні кількості вимірювань, вектор ознак збільшується в геометричній прогресії. Тому найкращим класифікатором для аналізу тональності є наївний байєсовський класифікатор, який має низьку обчислювальну складність і високу ефективність.
3. Повідомлення можуть містити інформацію, що не потрібна для аналізу тональності текстів: посилання, імена користувачів, стоп-слова, слова з максимальною частотою зустрічальності. Також для поліпшення якості алгоритму і зменшення словника необхідно приводити слова до словникової форми.
4. При аналізі тональності текстів з соціальних мереж необхідно враховувати смайлики, так як вони допомагають позбутися неоднозначності.

5. Всі попередні пункти (1 - 4) були враховані при розробці алгоритму визначення тональності російських текстів, який показав високу ефективність (85% текстів були класифіковані правильно).
6. Як джерело текстів була обрана соціальна мережа Twitter. Для зберігання повідомлень з Twitter.
7. Розроблено додаток для визначення тональності текстів на мові програмування python, яка дозволяє користувачеві дізнатися результат тональності текстів.

У наш час соціальні мережі є популярним засобом комунікації. У них люди висловлюють свої думки про продукти, послуги, політику і т.д., що робить соціальні мережі цікавими для соціологічних і маркетингових досліджень. Тому тема класифікації текстів є актуальною.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AI - Natural Language Processing // TutorialsPoint [Електронний ресурс]. URL: https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_natural_language_processing.htm (дата звернення: 11.10.2019).
2. Навчаємо комп'ютер почуттям (sentiment analysis по-російськи) // Хабрахабр [Електронний ресурс]. URL: https://habrahabr.ru/post/149605/ (Дата звернення: 11.10.2019).
3. Understanding Sentiment Analysis: What It Is & Why It's Used //brandwatch [Електронний ресурс]. URL:https://www.brandwatch.com/blog/understanding-sentiment-analysis/ (дата звернення: 11.10.2019).
4. Number of tweets per day? // David Sayce [Електронний ресурс]. URL: http://www.dsayce.com/social-media/tweets-day/ (дата звернення: 11.10.2019).
5. Хто володіє інформацією, той володіє світом. // пліткарі [Електронний ресурс]. URL: http://www.spletnik.ru/blogs/govoryat_chto/75404_kto-vladeetinformatciey-tot-vladeet-mirom (дата звернення: 12.10.2019).
6. The First Public Demonstration of Machine Translation Occurs //HistoryOfInformation.com [Електронний ресурс]. URL: http://www.historyofinformation.com/expanded.php?id=852 (дата звернення: 13.10.2019).
7. SHRDLU // Stanford.Edu [Електронний ресурс]. URL:http://hci.stanford.edu/winograd/shrdlu/ (дата звернення: 13.10.2019).
8. The History of Natural Language Processing // ETHW [Електронний ресурс]. URL: http://ethw.org/The_History_of_Natural_Language_Processing (дата звернення: 13.10.2019).
9. IBM Watson Vanquishes Human Jeopardy Foes // PCWorld [Електронний ресурс].

URL:http://www.pcworld.com/article/219893/ibm_watson_vanquishes_human_jeopardy_foes.html (дата звернення: 13.10.2019).

10. Introduction to Natural Language Processing (NLP) 2016 // Algorithmia [Електронний ресурс]. URL: http://blog.algorithmia.com/introductionnatural-language-processing-nlp/ (дата звернення: 14.10.2019).

11. Автоматичне реферування // Bourabai [Електронний ресурс]. URL: http://bourabai.ru/dbt/internetica/autorefer.htm (дата звернення: 14.10.2019).

12. Terminology extraction // Wikipedia [Електронний ресурс]. URL: https://en.wikipedia.org/wiki/Terminology_extraction (дата звернення: 15.10.2019).

13. Named-entity recognition // Wikipedia [Електронний ресурс]. URL: https://en.wikipedia.org/wiki/Named-entity_recognition (дата звернення: 15.10.2019).

14. Part-of-speech tagging // Apertium [Електронний ресурс]. URL: http://wiki.apertium.org/wiki/Part-of-speech_tagging (дата звернення: 15.10.2019).

15. Relationship extraction // DeepDive [Електронний ресурс]. URL: http://deepdive.stanford.edu/relation_extraction (дата звернення: 15.10.2019).

16. стемінг // Textis [Електронний ресурс]. URL:http://textis.ru/stemming/ (дата звернення: 15.10.2019).

17. ФАКТ // Національна філософська енциклопедія [Електронний ресурс]. URL: http://terme.ru/termin/fakt.html (дата звернення: 16.10.2019).

18. Вороніна І.Є., Гончаров В.А. Аналіз емоційного забарвлення повідомлень в соціальних мережах (на прикладі мережі «ВКонтакте») // ВІСНИК ВДУ. - 2015 [Електронний ресурс]. URL: http://www.vestnik.vsu.ru/pdf/analiz/2015/04/2015-04-21.pdf (дата звернення: 16.10.2019).

19. Sentiment Analysis // Lexalytics [Електронний ресурс]. URL:https://www.lexalytics.com/technology/sentiment (дата звернення: 16.10.2019).

20. Tsur O., Davidov D., Rappoport A. ICWSM - A Great Catchy Name: SemiSupervised Recognition of Sarcastic Sentences in Online Product Reviews /Назва конференції: Proceedings of the Fourth International Conference on Weblogs and Social Media, Вашингтон, 23-26 травня 2010, С. 162-168.

21. Saif H., He Y., Alani H. Alleviating Data Sparsity for Twitter Sentiment Analysis // CEUR-WS.org [Електронний ресурс]. URL: http://ceurws.org/Vol-838/paper_01.pdf (дата звернення: 17.10.2019).

22. Пазельская А., Соловйов А. Метод визначення емоцій у текстах російською мовою // dialog-21 [Електронний ресурс]. URL: http://www.dialog-21.ru/digests/dialog2011/materials/ru/pdf/50.pdf (дата звернення: 18.10.2019).

23. Kan D. Rule-based approach to sentiment analysis at ROMIP 2011 //dialog-21 [Електронний ресурс]. URL: http://www.dialog21.ru/media/1393/138.pdf (дата звернення: 18.10.2019).

24. Supervised and Unsupervised Machine Learning Algorithms // Machine Learning Mastery [Електронний ресурс]. URL: http://machinelearningmastery.com/supervised-and-unsupervised-machinelearning-algorithms/ (дата звернення: 20.10.2019).

25. Dayan P. Unsupervised Learning // Gatsby Computational Neoroscience Unit [Електронний ресурс]. URL: http://www.gatsby.ucl.ac.uk/~dayan/papers/dun99b.pdf (дата звернення: 20.10.2019

26. Наївний байєсовський класифікатор // bazhenov.me [Електронний ресурс]. URL: http://bazhenov.me/blog/2012/06/11/naive-bayes.html (дата звернення: 21.10.2019).

27. Аналіз тональності висловлювань в Twitter // nauchkor [Електронний ресурс]. URL: http://nauchkor.ru/pubs/analiz-tonalnosti-vyskazyvaniy-vtwitter-587d36485f1be77c40d58b7b (дата звернення: 22.10.2019).

28. К. В. Воронцов Лекції по методу опорних векторів // ccas [Електронний ресурс]. URL: http://www.ccas.ru/voron/download/SVM.pdf (дата звернення: 22.10.2019).

29. Метод k найближчих сусідів // Вікіпедія [Електронний ресурс]. URL: https://ru.wikipedia.org/wiki/Метод_k_ближайших_соседей (дата звернення: 23.10.2019).

30. Метод k-найближчих сусідів // BaseGroup Labs [Електронний ресурс]. URL: https://basegroup.ru/community/glossary/nearest-neighbor (дата

31. звернення: 23.10.2019).

32. Логіт-аналіз // MachineLearning.ru [Електронний ресурс]. URL: http://www.machinelearning.ru/wiki/index.php?title=Логит-анализ (дата звернення: 23.10.2019).

33. SentiStrength Download // Softpedia [Електронний ресурс]. URL: http://www.softpedia.com/get/Others/Home-Education/SentiStrength.shtml (Дата звернення: 27.10.2019).

34. WordNet Domains // wndomains.fbk [Електронний ресурс]. URL: http://wndomains.fbk.eu/wnaffect.html (дата звернення: 28.05.2017).

35. Gonçalves P. та ін. Comparing and Combining Sentiment Analysis Methods [Електронний ресурс]. URL:

36. http://homepages.dcc.ufmg.br/~fabricio/download/cosn127-goncalves.pdf (Дата звернення: 28.10.2019).

37. Opinion mining and sentiment analysis / Pang B., Lee L. Нью-Йорк: ACM, 2013, С. 79-86.

38. Sentiment inTwitter Events / Thelwall M., Buckley K., Paltoglou G. // J. Am. Soc. Inf. Sci. Technol., 2011. вип. № 2. С. 406-418.

39. Котельников Є. В., Клековкіна М. В. Автоматичний аналіз тональності текстів на основі методів машинного навчання // dialog21 [Електронний ресурс]. URL: http://www.dialog21.ru/media/1380/105.pdf (дата звернення: 06.11.2019).

40. Syntactic N-grams as machine learning features for natural language processing / Sidorov G. та ін. // Expert Syst., 2014. вип.№ 3. С. 853-860.

41. Російськомовний корпус коротких текстів // study.mokoron [Електронний ресурс]. URL: http://study.mokoron.com/ (дата звернення: 07.11.2019).

42. Калімолдаев М., Пак А. Нейромережевий метод семантичного імовірнісного виведення в завданні поліпшення релевантності результатів пошукового запиту // problem-info [Електронний ресурс]. URL:

43. http://www.problem-info.sscc.ru/2014-3/8.pdf (дата звернення:

44. 08.11.2019).

45. MyStem - Технології Яндекса // Яндекс [Електронний ресурс]. URL: https://tech.yandex.ru/mystem/ (дата звернення: 09.11.2019).

46. Ляшевская О. Н., Шаров С. А. Новий частотний словник російської лексики // Словники на основі національного корпусу російської мови [Електронний ресурс]. URL: http://dict.ruslang.ru/freq.php (дата звернення: 11.11.2019).

47. Національний корпус російської мови // Національний корпус російської мови [Електронний ресурс]. URL: http://www.ruscorpora.ru/ (Дата звернення: 11.11.2019).

48. Using Emoticons to reduce Dependency in Machine Learning Techniques for Sentiment Classification / Read J. Страсбург: Association for Computational Linguistics, 2005. С. 43-48.

49. Список текстових усмішок і акронімів // It's my life [Електронний ресурс]. URL: http://ancher.ru/content/06-04-2009/Spisok-tiekstovykhsmailov-i-akronimov (дата звернення: 11.11.2019).

50. Оцінка класифікатора (точність, повнота, F-міра) // bazhenov.me [Електронний ресурс]. URL: http: //bazhenov.me/blog/2012/07/21/classification-performanceevaluation.html (дата звернення: 15.11.2019).

51. Spring Social // Spring по-русски! [Електронний ресурс]. URL: http://spring-projects.ru/projects/spring-social/ (дата звернення: 20.11.2019).

52. Twitter Developer Documentation // Developers [Електронний ресурс]. URL: https://dev.twitter.com/rest/public (дата звернення: 20.11.2019).

53. Thread Pools // Jenkov.com [Електронний ресурс]. URL: http://tutorials.jenkov.com/java-concurrency/thread-pools.html (дата звернення: 22.11.2019).

54. Глава 1 - Основи // The Little MongoDB Book [Електронний ресурс]. URL: http://jsman.ru/mongo-book/Glava-1-Osnovy.html (дата звернення: 21.11.2019).

55. Why Spring Boot? // DZone [Електронний ресурс]. URL:

56. https://dzone.com/articles/why-springboot (дата звернення: 22.11.2019).

57. Spring Framework // Spring [Електронний ресурс]. URL: http://springprojects.ru/projects/spring-framework/ (дата звернення: 22.11.2019).

58. Chart.js // Chart.js [Електронний ресурс]. URL: http://www.chartjs.org/docs/latest/ (дата звернення: 23.11.2019).