Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

# Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему «Метод та засіб захисту від деструктивних інформаційних впливів»

08-20.МКР.012.00.000 ПЗ

Виконав: студент 2 курсу, групи 1БС-18м
Спеціальність 125 Кібербезпека
ОПП Безпека інформаційних і комунікаційних систем

_____ Олійник Є. А.

Керівник: к. т. н., доц. каф. ЗІ

_____ Дудатьєв А. В.

Рецензент: к. т. н., доц. кафедри ОТ

_____ Крупельницький Л. В.

Вінниця - 2019 року

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Освітньо-кваліфікаційний рівень магістр
Спеціальність 125 Кібербезпека
ОПП Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ
**Завідувач кафедри ЗІ, д. т. н., проф.**
**_____ В. А. Лужецький**
**_____ _____ 2019 року**

**З А В Д А Н Н Я**

**НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Олійнику Євгену Анатолійовичу

1. Тема роботи: «Метод та засіб захисту від деструктивних інформаційних впливів» керівник роботи: Дудатьєв Андрій Веніамінович, к. т. н., доц. каф. ЗІ, затверджена наказом ректора ВНТУ № 254 від 02.10.2019 р.

2. Строк подання студентом роботи _____ 2019 р.

3. Вихідні дані до роботи:
   - методи аналізу тональності тексту;
   - аналіз за вмістом сторінки з новиною;
   - мови програмування Python, javascript.

4. Зміст розрахунково-пояснювальної: Вступ. Аналіз інформаційних джерел. Розробка моделі для аналізу тональності тексту. Розробка програмного засобу. Економічна частина. Висновки. Перелік інформаційних джерел. Додатки.

5. Перелік графічного матеріалу.
Таблиця налізу існуючих засобів класифікації тональності тексту(плакат, А4). Схеми методів класифікації тональності тексту (плакат, А4). Схема роботи програмного засобу (плакат, А4). Таблиця з порівнянням ефективності методів класифікації тональності тексту (плакат, А4). Діаграма популярності браузерів в Україні (плакат, А4). Інтерфейс програмного засобу (плакат, А4).

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада Консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| 1 | Дудатьєв А. В., к. т. н., доц. каф. ЗІ | | |
| 2 | Дудатьєв А. В., к. т. н., доц. каф. ЗІ | | |
| 3 | Дудатьєв А. В., к. т. н., доц. каф. ЗІ | | |
| 4 | Мацкевічус С. С., ст. викл. каф. ЕПВМ | | |

7. Дата видачі завдання _____ 2019 року

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів магістерської кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Аналіз завдання. Вступ | 01.09.2019 – 04.09.2019 | |
| 2 | Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи | 05.09.2019 – 15.09.2019 | |
| 3 | Науково-технічне обґрунтування | 16.09.2019 – 22.09.2019 | |
| 4 | Розробка технічного завдання | 23.09.2019 – 29.09.2019 | |
| 5 | Розробка рішень | 30.09.2019 – 12.10.2019 | |
| 6 | Практична реалізація, моделювання, експериментування, результати | 14.10.2019 – 10.11.2019 | |
| 7 | Розробка розділу економічного обґрунтування доцільності розробки | 11.11.2019 – 17.11.2019 | |
| 8 | Аналіз виконання ТЗ, висновки | 18.11.2019 – 24.11.2019 | |
| 9 | Оформлення пояснювальної записки | 25.11.2019 – 30.11.2019 | |
| 10 | Попередній захист та доопрацювання МКР | 28.11.2019 – 01.12.2019 | |
| 11 | Перевірка магістерської роботи на наявність плагіату | 02.12.2019 – 10.12.2019 | |
| 12 | Представлення МКР до захисту, рецензування | 11.12.2019 – 14.12.2019 | |
| 13 | Захист МКР | 16.12.2019 – 18.12.2019 | |

Студент _____ Олійник Є. А.

Керівник роботи _____ Дудатьєв А. В.

# АНОТАЦІЯ

У магістерській кваліфікаційній роботі розглянуто та проаналізовано основні методи деструктивних інформаційних впливів та методи захисту від них. Проаналізовано існуючі програмні засоби для аналізу тональності тексту. Проведено аналіз відомих класифікаторів тональності тексту та визначено критерії для оцінки ефективності. Проведено порівняння ефективності існуючих методів класифікації тональності тексту та запропоновано модифікований метод. Розроблено структурну модель та реалізовано запропоний методу класифікації тексту. Проведено тестування ефективності розробленого методу та доведено його ефективність. На основі запропонованого методу розроблено програмний засіб для аналізу тональності тексту веб-сайтів. Наведено результати роботи програмного засобу.

## ABSTRACT

In the master's qualification work the basic methods of destructive information influences and methods of protection against them are considered and analyzed. Existing software for analyzing the tone of the text is analyzed. An analysis of the known classifiers of the tone of the text and the criteria for evaluating the effectiveness. A comparison of the effectiveness of existing methods of classification of the tone of the text is performed and a modified method is proposed. A structural model has been developed and a proposed text classification method implemented. The efficiency of the developed method is tested and its effectiveness is proved. Based on the proposed method, a software tool has been developed to analyze the tone of the text of websites. The results of the software are given.

**ЗМІСТ**

# ВСТУП

Через розвиток різних інтернет-сервісів, в яких можна висловити своє ставлення до чого-небудь, за останні роки відбулося стрімке збільшення кількості суб'єктивного контенту: повідомлення в блогах і соціальних мережах, думки щодо послуг або товарів, ресторанів, політичних подій або підприємств - контенту, що створюється і спільно використовуваного користувачами інтернету. В Україні згідно соціологічного опитування Київського міжнародного інституту соціології 27% українців читають інформацію на українських інтернет-сайтах, 24% дізнаються з соцмереж[1].

Суб'єктивний контент це величезна кількість інформації без чіткої структури, яку можна використовувати. Але кількість відгуків та оглядів, які публікуються в соцмережах, може досягати десятки тисяч, їх обробка людиною є неможливою. Також тональність у новинах має серйозний вплив на погляди людини.

Аналіз тональності тексту відноситься до обробки природної мови та комп'ютерної лінгвістики. Аналіз тональності тексту дозволяє отримати з тексту думку автора яку він закладав у інформацію. Ставлення може висловлювати судження, думку або оцінку автора, його емоційний стан. На основі даної інформації можна отримати дані, опираючись на які, уникати інформаційних впливів.

Дослідження аналізу тональності проводяться рідко попри зростаючі потреби в суспільстві щодо аналізу думок. Також дослідники часто стикаються з проблемами при розробці методів автоматизованого аналізу тексту.

Аналіз новинних сайтів може бути корисним для політтехнологів, працівників медіа та звичайних користувачів. Це робить актуальним завдання захисту від деструктивних інформаційних впливів.

Об’єкт дослідження – процес захисту від деструктивних інформаційних впливів.

Предмет дослідження – Моделі аналізу тональності текстової інформації.

Метою магістерської кваліфікаційної роботи є підвищення захищеності від деструктивних інформаційних впливів.

Для досягнення мети необхідно розв’язати такі задачі:

– дослідити методи деструктивного впливу;

– дослідити методи аналізу тональності тексту;

– здійснити огляд проблем аналізу тональності тексту та методи їх вирішення;

– здійснити огляд існуючих засобів для оцінки тональності тексту;

– дослідити методи машинного навчання для класифікації даних;

– проаналізувати переваги та недоліки основних методів машинного навчання, на основі отриманих даних розробити новий метод для класифікації тексту та довести його ефективність;

– на основі розробленого методу розробити програмний засіб який здійснює аналіз тональності тексту.

– здійснити економічне обґрунтування та доведено доцільність його розробки.

Наукова новизна полягає у вдосконаленні методу класифікації тональності тексту шляхом розробки класифікатора із більшою точністю визначення тональності тексту порівнюючи з існуючими методами.

Практична цінність полягає у розробці програмного забезпечення для аналізу тональності тексту з веб-сайтів, що дозволяє виявити потенційний деструктивний інформаційний впив.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Методи реалізації деструктивних інформаційних впливів

Інформаційний вплив може бути реалізований з використанням багатьох механізмів.

*Приєднання до внутрішнього світу аудиторії*. Для першого контакту – заявити, що відправник повідомлення входить з читачами в певну спільність (по соціальній, культурній ознаці і т.д.). Ознакою маніпуляції в цьому випадку - використання не точних слів і метафор. Уточнення ідеалів та інтересів, яких дотримується «відправник повідомлення», одразу включає психологічний захист людей, які не поділяють цієї позиції, та спонукає до уявного діалогу, чим різко ускладнюється маніпуляцію.[2]

*Психічні автоматизми*. Психічні автоматизми - це стандартні схеми поведінки, що звичні для більшості людей. Кожна людина використовує «автоматизми» в сприйнятті. Користь стереотипів для людини важлива, щоб сприймати і оцінювати повідомлення швидко, не думаючи; зловмисник може використати їх як "фільтри", через що інформація сприймається за дійсне.[2]

*Стереотипи.* Головний "матеріал", яким користується маніпулятор, є соціальні стереотипи. Зазвичай стереотипи складаються з емоційного ставлення людини до якихось об'єктів і явищ, тому при вироблення стереотипів мова йде не лише про інформацію також про соціально-психологічний процес. Через це стереотипи можуть бути виявлені і використані як цілі для маніпуляції.[2]

*Метафора*. Найбільш важливий момент в історії поняття - перехід від прямого значення до переносного, тобто народження метафори. Стереотипи - це готові штампи мислення, але штампи естетично привабливі. Думка яку виражають поетично грає велику роль об’єднанні людей та програмування їхньої поведінки. Відомо, що людина, щоб діяти в своїх інтересах (а не в інтересах маніпулятора), повинен реалістично визначити три речі: нинішній

стан, бажане для нього майбутній стан, шлях переходу від нинішнього стану до майбутнього. Спокуса заощадити інтелектуальні зусилля змушує людину замість вивчення і осмислення всіх цих трьох речей вдаватися до асоціацій і аналогій: називати ці речі якоюсь метафорою, яка відсилає до інших, вже вивченим станам. Найчастіше, ілюзорна і сама упевненість в тому, що ті стани, через які він пояснює собі нинішнє, йому відомі або зрозумілі.[2]

*Мотиваційне забезпечення*. Разом з тим, як би не вдалим був сам по собі автоматизм, на який метил маніпулятор, ніяка структура не буде працювати без мотиваційної підтримки, ніяка дія - внутрішнє або зовнішнє - не почнеться, якщо до того немає внутрішньої роздільною санкції. Звідки ж така санкція береться, якщо такий маніпулятивний вплив суперечить інтересам адресата? Відбувається це в результаті того, що маніпулятивний вплив, втім, як і всякий вплив, серед мішеней обов'язково має і мотивують структури (кінцева мотиваційна спрямованість маніпулятивного впливу вже відзначалася вище). Спряженість зовнішнього впливу і внутрішніх процесів в мотиваційній частині така, що важко розрізнити прийом і його ефект.[2]

*Перцептивні маріонетки*. У всі часи і у всіх психотехнічних системах - релігіях, психотерапевтичних школах, педагогічній практиці, мистецтві оперування образами складало ключові елементи використовуваних технологій: обрядів, ритуалів, методів, прийомів. Механізм такого впливу заснований на асоціації між образом і релевантною йому потребою, устремлінням або мотиваційної установкою.[2]

Отже, перцептивно-орієнтований маніпулятивний вплив використовує наступні засоби:

• основний діючий агент - образ;

• способи спонукання - пряма актуалізація мотиву, спокушання, провокація, пробудження інтересу;

• мішені - бажання, інтереси,

• автоматизми - міжмодальні асоціації, релевантність образу мотиву, наміченому як мішень впливу.

Визначивши методи здійснення деструктивного інформаційного впливу, було виявленно, що всі методи взаємодіють на сентиментальному рівні. Саме тому для того, щоб реалізувати захист потрібно здійснювати сентиментальний аналіз інформації.

## 1.2 Сентиментальний аналіз в класифікації тексту

Сентиментальний аналіз виконується на декількох логічних рівнях [3]. На рівні документа завдання полягає в тому, щоб класифікувати тональність документа: позитивна вона чи негативна, іноді ще розглядають нейтральний випадок. На рівні пропозиції завданням є класифікація настрою окремого пропозиції: позитивна або негативна. Тут також можливий розгляд 3-го випадку - нейтрального настрою. І, нарешті, рівень аспектів або сутностей. Завдання: класифікувати емоційне забарвлення окремих пропозицій або фраз, для певних сутностей або аспектів.

Метою аналізу настроїв на рівні документа є передбачення загальної тональності, вираженої в документі. У документах, на які застосовується даний тип аналізу, як правило, автор оцінює тільки один об'єкт, наприклад, огляди продуктів або послуг, готелів, книг, фільмів і т.п.

Завдання передбачення тональності на рівні документа може бути розглянута як стандартна проблема класифікації тексту. Тоді її можна вирішити використовуючи методи машинного навчання, такі як логістична регресія, наївний байєсівський класифікатор і ін. Є кілька припущень, пов'язаних з підходом до класифікації тексту. По-перше, передбачається. Що весь текст присвячений єдиній меті, а саме об'єкту, що є предметом огляду. По-друге, автор вважається власником, власником думки [4].

Формально задача визначення тональності на рівні документа визначається наступним чином. Існує множина документів $D$ з думками авторів, класифікація визначає, яку думку, позитивну або негативну, про об'єкт містить кожен документ $d \in D$. Документ $d$ є відгуком про об'єкт $o$, який може бути продуктом, людиною, організацією, річчю і т. д., $o$ зображується ієрархічним деревом компонентів і під компонентів(рис.1.1). Кожен вузол являється компонентом і асоційований з ним набір атрибутів [5].

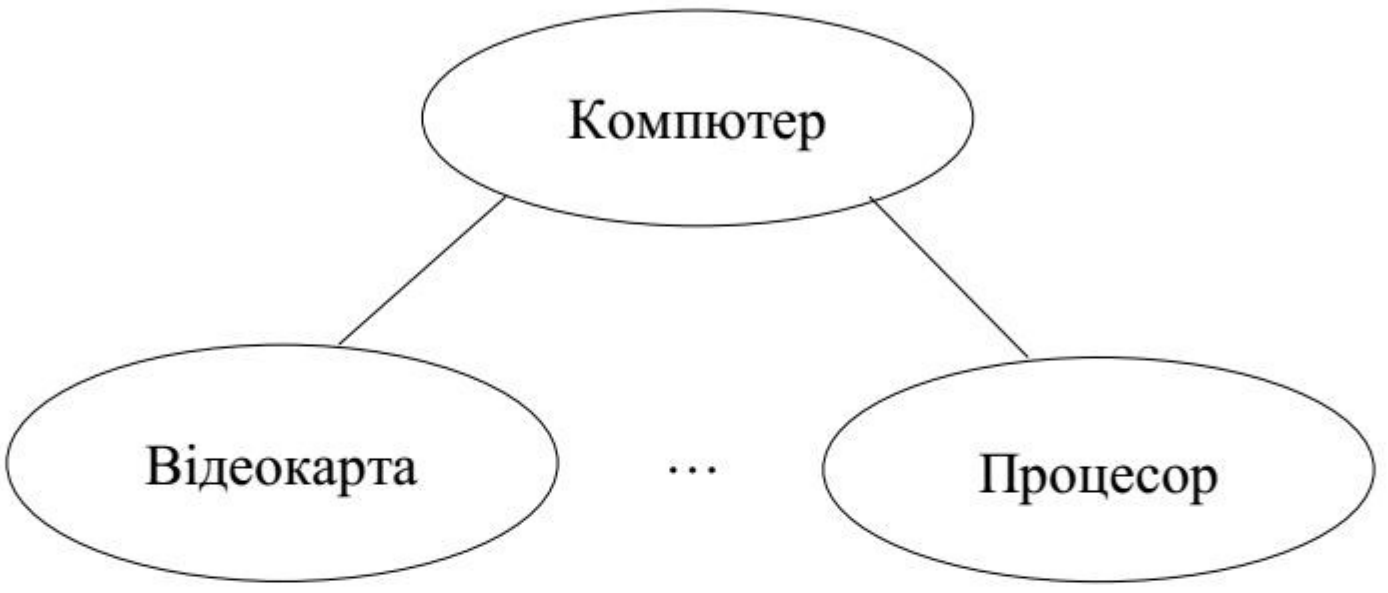



Рисунок 1.1 – Ієрархічне представлення об’єкту

Думка може бути висловлено про будь-який вузол або атрибут вузла. Для позначення компонентів і їх атрибутів використовується термін аспект або ознака.

## 1.3 Підходи до класифікації тональності тексту

Існує кілька підходів до аналізу тональності тексту, розглянемо їх нижче[6].

Класифікація. Для визначення тональності використовується один з алгоритмів (класифікаторів), навчений на характеристиках вмісту розмічених документів, тобто тих, для яких вказано «правильну відповідь» - тип тональності, такий спосіб називають навчанням з учителем. При цьому документи подаються у вигляді векторів за обраною метрикою, а їх

характеристики або особливості являють собою координати векторів. Підхід автоматизований, але залежний від навчальних даних, їх якості.

*Метод тональних словників* полягає в обчисленні полярності документа, з використанням значень тональності слів. У найпростішому це словник із значеннями їх важливості[7]. При використанні даного підходу, кожному слову привласнюється його значення з словника, потім обраним способом, наприклад, значенням середнього арифметичного за всіма отриманими ваг тональностей, обчислюється загальна тональність тексту. Спосіб простий, але не універсальний, тому що немає гарантії, що в ньому виявляться абсолютно все слова.

*Підхід, заснований на правилах*, шукає думки в тексті і класифікує їх, базуючись на кількості негативних і позитивних слів[8]. В ньому розглядаються різні правила класифікації, такі як тональні словники, слова-заперечення, що підсилюють значення слова, ідіоми, змішані думки і т.д. Самі правила будуються на основі шаблонів виділених з тексту. Це можуть бути слова і фрази. Цей метод є найбільш точним і часто використовується в комерційних системах, але складний в реалізації, тому, що потрібно якомога більше правил для точної класифікації: в українській мові порядок слів у реченні не фіксований, необхідно врахувати безліч варіантів.

*Кластеризація або кластерний аналіз* відноситься до навчання без вчителя, є завданням угруповання набору об'єктів таким чином, щоб об'єкти в одній групі (кластерах) були найбільш схожі один з одним, ніж об'єкти в інших групах (кластерах)[9]. Подібність об'єктів визначається за допомогою різних метрик. підхід повністю автоматизований, не вимагає навчальної вибірки, але має низьку точністю, вимагає ручного коректування метрик подібності при поганих результатах.

Який з даних вище підходів застосовувати, залежить від потрібної точності аналізу, наявності навчальних даних, тимчасових рамок розробки та цілей. Як було сказано раніше, метод, заснований на застосуванні правил, є

найбільш точним, проте, в основному, аналіз тональності використовується в бізнес-розвідці, метою його є думки інтернет-користувачів. Прочитавши пару повідомлень з тих же соціальних мереж, можна зрозуміти, що не всі користувачі пишуть без орфографічних помилок, не кажучи вже про граматичні. І якщо орфографію ще можна виправити, написавши програмне рішення, то з граматикою все складніше. Це значно знижує ефективність підходу з правилами. Деякі готові програмні рішення, об'єднують декілька методів, заявляючи про високі показники метрик. Головною проблемою всіх підходів, за винятком кластеризації, яка має дуже погану точністю, є прихильність до мови: граматика правил, слова в шаблонах і словниках, навчальних вибірках класифікаторів.

## 1.4 Проблеми аналізу тональності тексту

Крім специфічних для конкретного методу складнощів, є проблеми властиві їм всім які негативно впливають на результат класифікації, опишемо основні[10].

*Сарказм і глузування*. По сарказму і глузувань складно присвоїти вірну тональність тексту, оскільки сарказм висловлює негативну думку, хоча слова, самі по собі, можу носити позитивне забарвлення.

*Висловлювання з нейтральним змістом слів*. Якщо автор ніяк явно не повідомляє про своє емоційне відношення, описуючи, наприклад певні події, стає незрозуміло, як розглядати його висловлювання. Повідомлення може бути в цілому, як негативного, так і позитивного характеру, але за словами зрозуміти це неможливо, таке речення буде віднесено до нейтрального класу.

*Різні настрої щодо різних цілей*. Автор в своєму висловлюванні може озвучити думку стосовно кількох предметів, людей, якості обслуговування і т.д. При цьому висловивши відношення різної тональності до цих об'єктів. Складно зрозуміти, до якого класу повинно бути віднесено таку думку. Якщо проводити

більш детальний аналіз, видаючи думку по кожному об'єкту, а не в цілому по тексту, проблема буде вирішена.

*Розбіжність емоційного стану автора і думки в тексті*. Реальний емоційний стан автора може відрізнятися від тональності тексту. Наприклад, сам текст оповідає про невдачі кого-небудь, про те, яка ця людина погана і т.д. Тобто наповнений словами, що виражає негативний сенс, але сам автор радіє промахам, нещастю цієї людини. Такий текст може бути несправедливо віднесений до негативного класу при поверхневому аналізі тональності документа.

Тлумачення слів поза контекстом (двозначність). Часто розгляд слів поза контекстом дає невірну оцінку, оскільки вони можуть мати інший сенс. Це особливо актуально у відгуках з вживанням сленгів. наприклад, слово «жесть» має кілька значень. Воно може позначати тонку листову сталь, незвичайну ситуацію або подію, що викликали бурхливу емоційну реакцію, яка може бути як негативною, так і позитивною.

Розглянуті вище проблеми ускладнюють задачу аналізу тональності, але є цілком розв'язуються, вимагаючи, часом, розгляд вхідних даних на більш глибокому логічному рівні, з використанням додаткових методів обробки природної мови.

## 1.5 Огляд існуючих систем для визначення тональності текстів

Незважаючи на те, що сентимент-аналіз є завданням актуальним і перспективною, існує не так багато відомих програмних рішень для даної проблеми.

Програмний засіб Sentimental - Аналіз тональності для платформи Node.js з використанням словника AFINN-111, де словами привласнені значення тональності: негативні слова мають негативні ваги, позитивні - позитивний[11]. Програма видає рейтинг позитивності, негативності або аналіз - об'єднує ці два

рейтинги в сукупну оцінку. документація для даного продукту відсутній зовсім. Є лише код на GitHub з коротким поясненням установки і використання. Підтримує російську і англійську мови. Система знаходиться у вільному доступі.

Модуль Eureka Engine - Рішення модульного типу, включає в себе: ідентифікацію мови, визначення тональності, автокласифікацію текстів за категоріями, виділення власних назв, нормалізацію слів і морфоаналізатор, скористатися нею можна через REST (Representational State Transfer) API або розгорнути систему всередині закритої мережі компанії[12]. Модуль визначення тональності здатний вирішити два завдання: визначення тональності щодо заданого користувачем об'єкта і автоматично певного системою об'єкта на базі комплексної інформації про нього, зрозуміти три тональності повідомлення: нейтральну, негативну і позитивну. Особливістю модуля є оцінка сили емоційності, тобто він не тільки відносить повідомлення до одного з трьох класів, а й надає кількісне співвідношення негативного і позитивного ставлення до нього. В Як метод оцінки тональності використовується машинне навчання, підтримується тільки російську мову. Система має детальну документацію, код закритий, є платною.

Програмний засіб ISPRAS API: Texterra - Надає більше 25 інструментів, заснованих на обробці природної мови, аналізі соціальних мереж і використанні бази знань[13]. Реалізований інструмент для сентимо-аналізу дозволяє аналізувати думку автора тексту щодо предметної області та кількох аспектів, відносити текст до одного з трьох класів (Позитивний, негативний і нейтральний). В якості методу аналізу тональності використовується машинне навчання. Надає SDK для мов Python і Ruby, але в альфа версіях. На інших мовах програмування пропонується використовувати HTTP REST API, для сентимент-аналізу це - Texterra REST. Підтримується російську та англійську мови, надана дуже детальна документація з прикладами. Система є безкоштовною для дослідних цілей і платна для комерційних, код закритий.

Програмний засіб Pattern - веб-модуль інтелектуального аналізу для мови програмування Python[14]. Містить 6 плагінів для роботи з web (Pattern.web), базами даних (pattern.db), текстом (pattern.text), пошуку по шаблоном (pattern.search), кластеризації та класифікації (pattern.vector), аналізу і візуалізації графів (pattern.graph). дослідження настроїв входить в pattern.text. Методом аналізу є виділення шаблонів, використання правил, регулярних виразів. Сам модуль оцінює текст, надаючи кількісне співвідношення негативного і позитивного класів в ньому в діапазоні [-1,0; 1,0]. Непогана документація, але зовсім немає прикладів по темі роботи. Бібліотека поширюється вільно, код відкритий, підтримує англійська, іспанська, німецька, французька, італійська та нідерландський мови.

Бібліотека TextBlob - є бібліотекою мови Python, надає простий API для вирішення завдань NLP, таких як, переклад, класифікація, аналіз настроїв, маркування частин мови, витяг іменованих сутностей[15]. Є дві реалізації аналізу тональності тексту PatternAnalyzer (Базується на бібліотеці Python Pattern) і NaiveBayesAnalyzer (бібліотека NLTK, наївний байесовский алгоритм класифікації), натренований на рецензіях до фільмів. Методом аналізу є машинне навчання або використання правил. Крім можливостей, успадкованих від Pattern, програма оцінює суб'єктивізм і об'єктивізм, де значення 0 показує високу об'єктивність, а 1.0 - суб'єктивність. передбачена можливість токенізації (виділення з тексту фрагментів для подальшого аналізу).Є хороша, детальна документація. При використанні PatternAnalyzer підтримуються всі, реалізовані в Pattern, мови, при застосуванні NaiveBayesAnalyzer допускається робота тільки з англійським. Бібліотека поширюється вільно, код відкритий.

Програмний засіб SentiStrength - система використовує словниковий підхід, розроблена для оцінки коротких текстів, навіть на неформальному мовою. визначає клас для документа, написаного англійською, фінською, німецькою, португальською, французькою, італійською, валлійському, грецькою, шведською, польському або арабською[16]. Оцінка тексту видається

в діапазоні [1; 5] для позитивної складової і [-5; -1] для негативної. Існують і інший вид надання оцінок: бінарна (позитивний, негативний клас), тернарного (позитивний, нейтральний, негативний клас) і оцінка по єдиної шкалою [-4; 4]. Програма є платною, відсутня документація і код, платформи .NET, Java.

Програмний засіб «Аналітичний кур'єр» - Даний продукт розроблений компанією «Айтек», сентимент-аналіз є лише одним з багатьох його можливостей, оцінку щодо об'єкта він виробляє на рівні оцінного висловлювання, найчастіше пропозиції. Видає результати у вигляді тверджень, забарвлених в червоний або зелений колір, в залежності від їх тональності; виділяє об'єкт, до якого вони належать, оцінює і позитивну, і негативну складову за шкалою від 0 до 5[17]. Система застосовує лексико-синтаксичні правила до результатів аналізу тексту, поданого на вхід, використовується велика кількість складних оціночних конструкцій. При цьому використовуються як структурні, так і лінгвістичні характеристики тексту: приналежність до частини мови, наявність заперечення і його позиція, відстань між об'єктом оцінки і маркером оцінки і т.д. Підтримуються російською та англійською мовами. Програма є платною, створена на платформі .NET. Відкритої документації немає, код відсутній, однак представлені ролики з демонстрацією роботи кожного інструменту.

Веб-сервіс BrandSpotter - який охоплює понад 90 млн джерел, таких як соціальні мережі, блоги, сотні онлайн ЗМІ, тисячі форумів, мікроблогів[18]. Система автоматично визначає тональність на основі тексту повідомлення по відношенню до об'єкта (продукту, бренду і т.д.), використовуючи лінгвістичний модуль. Вона може бути позитивною, негативною або нейтральної. Плюсом системи є комбінований підхід до сентимент-аналізу: використовується машинне навчання, правила і словники тональностей разом. Крім того, розробники також запевняють, що даний модуль постійно перенавчатися, виправляються помилки с урахуванням специфіки бізнес-сегмента. BrandSpotter

працює з російською, українською, казахською, вірменською мовами. Система не поставляється в вільному доступі, документації немає.

Програмний засіб DictaScope - здійснює інформаційний пошуку та аналіз тексту[19]. У нього входять: DictaScope Syntax (синтаксичний аналізатор), DictaScope Phrase Modifier (обробка словосполучень), DictaScope Tokenizer (витяг об'єктів), DictaScope Structure (аналіз структури документа). Саме ж рішення для визначення тональності тексту носить назву «Моніторинг соціальних медіа», воно розроблено для спостереження за думками про послуги, товари, персонах, може бути використано для виявлення ризиків. Відповідно до опису, оцінювання відгуків відбувається в 3 етапи:

1) з соціальних медіа збирається всіляка Інформація про об'єкт, який потребує оцінка;

2) здійснюється глибокий лінгвістичний аналіз тексту щодо нашого об'єкта моніторингу і належних до нього характеристик;

3) За цим зібраним даними відбувається оцінювання з трьома можливими значеннями: позитивно, негативно, нейтрально. При цьому величезним мінусом є повна відсутність опису процесу оцінювання, що використовуються при цьому технологій. Немає ніякої документації, а вже тим більше і коду.

Веб-сервіс IBM Watson: Tone Analyzer - створений для аналізу емоцій і настроїв усього, що люди пишуть онлайн, тобто рецензій, твітів, повідомлень в соціальних мережах[20]. самі розробники рекомендують використовувати його компаніям для збору статистики думок про цієї компанії. Сервіс використовує лінгвістичний аналіз для виявлення емоційних тональностей в тексті, може аналізувати настрій як на рівні документа, так і на рівні пропозиції, відносячи його до якогось з трьох класів тональності. Взаємодія з ним здійснюється через Tone Analyzer API explorer. Доступний аналіз англійською та французькою мовами. Є відмінна документація, яка дає відповіді на всілякі питання, немає відкритого коду. Система надає обмежений функціонал в вільному доступі, повний, звичайно ж, за оплату.

Програмний засіб NetOwl Extractor - пропонує розширені можливості аналізу настроїв, використовуючи витяг об'єктів і сутностей разом з іншими технологіями NLP[21]. NetOwl не обмежується визначенням позитивної або негативною тональністю на рівні документа або пропозиції, оскільки такий підхід не може розпізнати множинні, іноді конфліктуючі почуття, що існують в рамках одного документа або пропозиції, також, як і сфокусуватися на предметі, про який висловлено думку. Він застосовує аналіз на рівні аспектів і сутностей, визначає ставлення до різним типам об'єктів, таким як люди, бренди і т.п. На рівні аспектів здійснюється більш тонкий аналіз, орієнтований на ознаки сутностей, наприклад, нова політика країни, ціна продукту, кампанію кандидата в президенти. Працює з англійською та арабською мовами, на платформах Linux і .NET. За даним продуктом немає ніякої документації, він є повністю закритим і платним. Невідомий навіть метод аналізу. Імовірно це комбінація машинного навчання і правил.

Веб-сервіс ConveyAPI - розроблений компанією CONVERSEON INC[22]. Як і попередні продукти, виробляє оцінку настроїв в соціальних мережах, але тільки англійською мовою. При цьому сказано лише, що це відбувається через однойменний API, використовуються методи машинного навчання, натреновані на сеті Converseon, що забезпечує найвищу точність, повноту і релевантність. При цьому використовується якась передова, відзначена нагородами семантична технологія, що очищає повідомлення користувачів від сарказму, сленгу і помилок. Код закритий, немає документації, неясно навіть, на які класи в результаті будуть класифіковані повідомлення та як - просто віднесені до класу або буде чисельно вказана полярність. Відповіді на ці питання можна з'ясувати тільки після заявки виробнику.

Веб-сервіс Repustate - використовується машинне навчання та правила[23]. Repustate враховує також і емоційно забарвлені фіксовані мовні звороти, використовуючи розроблені цією ж компанією сентимент-моделі різних мов. Можливий аналіз:

- За темами, вказавши об'єкт, отримуємо всі оцінки настроїв з тексту по ньому, тобто аналіз на рівні аспектів;

- По шматочках документа;

- По кожному документу з колекції.

Сама оцінка тональності ділиться на позитивну і негативну. Доступний аналіз повідомлень на арабською, китайською, німецькою, датською, англійською, французькою, італійською, польською, тайською, в'єтнамською, турецькому, корейському, японською, іспанською та російською мовами. Дана докладна документація з описом кожного методу використовуваного REST API. Крім роботи через згаданий програмний інтерфейс програми, можливо розгортання системи всередині мережі компанії. Відкритого коду, звичайно ж, не побачити. Система повністю платна, але є тріал-версія. Отже більшість програмних рішень не поширюється вільно, дуже мало з дійсно повною документацією.

## 1.6 Попередня обробка документів для подальшого аналізу

Завдання визначення тональності тексту зводиться до задачі класифікації тексту на 2 (бінарна класифікація, позитивний і негативний) або 3 (додається нейтральний) класи. В загальному випадку класифікацією називається розділ машинного навчання, вирішальний наступне завдання. є множина екземплярів, яка розбита на класи. Задано підмножину екземплярів з певними класами, навчальна вибірка. У інших екземплярів класи невідомі. Потрібно побудувати алгоритм, здатний передбачити за наявними даними навчальної вибірки клас екземпляра, у якого він невідомий[24].

Тексти, написані на природній мові, часто містять слова, що не несуть ніякої корисної для класифікації інформації, їх називають стоп словами. Для української мови це можуть бути  сполучники, прийменники, для англійської -

артиклі, також можна віднести сюди і розділові знаки. Для поліпшення якості класифікації такі слова видаляють.

Після, або перед видаленням стоп-слів, здійснюється приведення кожного слова до нормальної форми. Це робиться для того, щоб одне і те ж слово, що зустрічається в тексті кілька разів, але записане в різній формі, трактувалося як одне і теж, було вірно зважено при застосуванні будь-якої метрики. Для цього застосовується або стемінг, або лематизація. Стемінг просто відкидає закінчення слів, керуючись одним з алгоритмів, лематизація ж приводить слово до словникової форми. Стемінг працює швидше лематизації. Однак усічення слова до основи може дати однаковий результат для абсолютно незв'язаних слів. Лематизація працює краще, але не враховує контекст, що може привести до неоднозначності при визначенні частини мови.

Крім перерахованих вище способів можна використовувати приведення слів до єдиного регістру і видалення помилок. Далі розглянемо підходи до індексування документів.

## 1.7 Векторизація тексту для визначення приналежності до класу

Для здійснення автоматичної обробки тексту необхідно представити його в зрозумілій для комп'ютера формі. подання тексту документа у вигляді вектора ознак називається індексуванням документа. Що вважати ознакою, залежить від підходу, це може бути, наприклад, просто слово. Після цього до тексту можливо застосовувати методи машинного навчання, які визначають належність тексту до класу за ознаками. в т.ч., тексти зі схожими ознаками виявляться в одному класі. Найбільш популярними методами для векторизації є Bag-of-words і Word2vec, проте перед їх застосуванням потрібна попередня обробка тексту.

### 1.7.1 Метод векторизації Bag-of-word

У класичній моделі Bag-of-words для представлення тексту $D_j$ створюється вектор довжини $n$, рівній унікальній кількості слів у всьому корпусі документів $D$. Кожна координата вектора визначає кількість входжень певного слова в будь-який документ. В результаті буде отримана розріджена матриця $C$, представлена на рис.1.1, де рядки являють собою слово, стовпці - векторизовані документи, а $c_{ij}$ - входження слова $w_i$ в документ $D_{jx}$[25].

$$C = \begin{array}{c} \\ w_1 \\ w_2 \\ \cdots \\ w_n \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & \cdots & D_m \end{array} \\ \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_m \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nm} \end{pmatrix} \end{array}$$

Рис. 1.1 – Матриця входження слів

Bag-of-words розбиває стійкі словосполучення(фразеологізми), зміст яких визначають всі слова у сполученні, передаючи зміст таких фразеологізмів невірно. Дану проблему можна усунути, використовуючи Bag-of-N-grams. N-грамма - це послідовність з N слів. Наприклад, з фразеологізму «Знаходитись на межі» будуть отримані наступні біграми: «Знаходитись на», «на межі».

### 1.7.2 Метод векторизації Word2vec

Word2vec - технологія (набір алгоритмів), розроблена компанією Google[26] і використовується для векторизації слів на основі однієї з двох нейромережевих архітектур CBOW і Skip-gram. При цьому передбачається, що слова, які використовуються в подібних контекстах, означають схожі речі, тобто є семантично близькими. отримані векторні уявлення слів мають розмірністю порядку десятків і сотень, при це семантично близькі слова є близькими векторами по косинусной міру, що обчислюється за формулою:

$$similarity = cos(\varphi) = \frac{x \cdot y}{\mathrm{V}x\mathrm{V}\mathrm{V}y\mathrm{V}} = \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n}(x_i)^2} \times \sqrt{\sum_{i=1}^{n}(y_i)^2}} \in [-1,1] \quad (1.1)$$

де $x$ і $y$ порівнювані вектори, $x_i$ і $y_i$ – -е координати цих векторів. Continuous-bag-of-word (CBOW) - безперервний мішок слів. Ця модель передбачає слово при даному контексті. Розглянемо спрощену модель двошарової нейронної мережі, припустивши, що контекст полягає лише з одного слова.

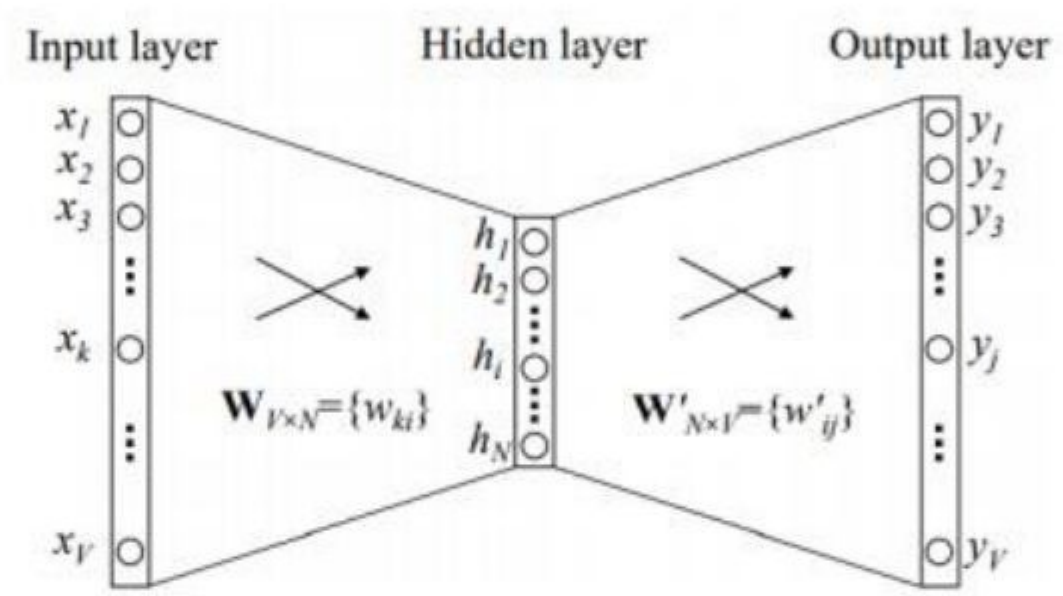



Рисунок 1.2 – Спрощена модель двохшарової нейронної мережі

Нехай розмір словника $V$, довжина вектора передбачає слова $N$. Тоді кількість нейронів на вхідному (input) і вихідному (output) рівні має кількість $V$, на прихованому рівні їх $N$ штук. Обидва шари повнозв'язні. На вхідні мережі подається вектор $x = \{x_1,\ldots, x_V\}^T$ для заданного контекстного слова, відповідного номеру $k$ в словнику, коефіцієнт $x_k = 1$, залишкові дорівнюють 0.

Ваги між вхідним і прихованими шарами може бути представлена у вигляді матриці $W_V \times N = \{w_{k\mathrm{i}}\}$. Кожен рядок матриці відповідає слову в слові. На прихованому шарі вхідної інформаційної елементі є сумма всіх вхідних нейронів, помножених на відповідну їм вагу, вона може бути представлена формулою:

$$h^T = x^T W = W_k = V_{wl}^T \quad (1.2)$$

де $h$ - вектор входів нейронів на прихованому рівні. По суті проводиться копіювання n-го рядка матриці $W$ в $h$. $v_{wl}$ - векторне подання вхідного слова $w_l$. Функція активації на прихованому рівні лінійна. Ваги між прихованим шаром і вихідним шарами розташовані в іншій ваговій матриці $W'_{N \times V} = \{w'_{ij}\}$, з їх допомогою можна порахувати оцінку $u_j$ для n-го слова словника, що є вхідним сигналом j-го нейрону вихідного шару.

### 1.7.3 Метод векторизації TF-IDF

Класифікація тексту відбувається по векторах ознак, розрахованим на основі обраної метрики. Вище був показаний класичний bag-of-words, враховує лише кількість входжень конкретного слова в документ (term frequency, TF (w, d)). Високі показники метрики можуть дати не емоційно забарвлені слова, а просто результат найчастіше вживаних[27].

Вона також може бути не інформативна при класифікації, оскільки, в тих ж рецензіях до фільмів, такі слова як «фільм», «мені», «подивився» можуть зустрічатися приблизно з однаковою частотою як в позитивних, так і в негативних відгуках.

Популярною метрикою оцінки важливості слова в контексті документа є TF-IDF (Term Frequency-Inverse Document Frequency). метою її є зменшення значення метрики для часто вживаних слів в корпусі документів і збільшення для слів, часто зустрічаються в конкретному документі. Частота терміна, що оцінює важливість слова в документі, визначається як зворотна частота народження терміна в документі, зменшує вагу широко вживаних слів, за формулою а сама міра TF-IDF обчислюється за формулою:

$$tf(t,d) = \frac{c_t}{\sum_n c_n} \quad (1.3)$$

де $c_t$ - число входжень терма $t$ в документ $d, \sum_n c_n$ - загальна кількість слів в даному документі.

$$idf(t,D) = log \quad (1.4)$$

де $|\ D\ |$ - число документів у колекції, $\{d_i \in D | t \in d_i\} \vee$ - число документів у корпусі $D$, в яких зустрічається терм $t$.

$$tf - idf(t,d,D) = tf(t,d) \times idf(t,D) \quad (1.5)$$

Оцінка ефективності перерахованих метрик в завданню аналізу тональності тексту буде представлена в розділі 3 даної роботи.

## 1.8 Метрики якості класифікації

Confusion matrix (Матриця неточностей) - одна з найбільш інтуїтивно зрозумілих і простих метрик для визначення правильності та точності побудованої моделі. У табл.1.1 представлена матриця що у випадку двох класів, показує, скільки раз вірно і невірно алгоритм визначив клас екземпляра[28].

Таблиця 1.1 – Матриця неточностей для бінарної класифікації

| Клас | Істинний | |
|---|---|---|
| Передбачений | C = 1 | C = 0 |
| C = 1 | True positive(TP) | False positive(FP) |
| C = 0 | False negative(FN) | True negative(TN) |

Кожен рядок матриці представляє екземпляри передбачених класів, стовпці - справжніх класів. Є кілька термінів, пов'язаних з матрицею неточностей:

1. True positive (TP) - істинно позитивними рішеннями важаються ті, в яких істинним класом є 1 (True), і передбачений результат був також 1 (True);

2. True negative (TN) - істинно негативні рішення, це такі випадки, коли справжній клас екземпляра 0 (False), передбачений - 0 (False);

3. False positive (FP) - помилково позитивні рішення виникають тоді, коли істинним класом є 0 (False), а передбачений був 1 (True);

4. False negative (FN) - помилково негативне рішення. Виникає у випадках, коли справжній клас екземпляра є 1 (True), а передбачений 0 (False).

Для оцінки якості роботи класифікатора використовуються різні числові метрики, розглянемо найпопулярніші в термінах матриці неточностей.

Accuracy - в задачі класифікації вимірюється як відношення числа всіх правильно зроблених прогнозів до всім прогнозам, зробленим моделлю. Обчислюється по всіх класах, за формулою:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1.6)$$

Достовірність дозволяє добре оцінити якість класифікації, якщо класи в оцінюваної вибірці майже збалансовані. Але якщо це не так, і, наприклад, 90% примірників вибірки належить до першого класу, а 10%, відповідно, до другого, достовірність є марною, даючи більше високу оцінку достовірності просто при віднесенні всіх примірників до більшого за обсягом класу.

Для 3х і більше класів запис цієї метрики задана формулою:

$$Accuracy = \frac{P}{N} \quad (1.7)$$

де $P$ - кількість примірників, за якими було прийнято вірне рішення (всі класи), $N$ - загальна кількість примірників у вибірці.

Висновки за розділом. В даному розділі розглянуто, методи деструктивного впливу, та визначено, що для цього найкраще підходить тональний аналіз тексту. Проаналізовано які основні методи використовуються для вирішення задачі аналізу тональності та показано, які основні недоліки присутні кожному методу.

Також, обрано критерії, за якими буде в подальшому оцінюватись ефективність модифікованого методу. В якості таких метрик обрано точність та повноту, адже саме вони відображають правильність вирішення задачі текстової класифікації.

# 2 МОДИФІКАЦІЯ МЕТОДУ АНАЛІЗУ ТОНАЛЬНОСТІ ТЕКСТУ

Основною метою даного розділу є виявлення найкращого методу аналізу наведених у попередньому розд та його модифікація для підвищення ефективності. Для оцінки ефективності роботи методу, використовуються метрики ефективності, що були наведені в попередньому розділі. Загалом, як було показано раніше, методи аналізу тональності поділяються на такі групи – методи навчання з учителем, semi-supervised методи та методи навчання без учителя (словникові).

Важливим кроком для методів машинного навчання з учителем є формування вектору ознак, тобто формату, в якому документ буде представлений для класифікатора. Як правило, в якості таких ознак використовуються наявність певних термінів та їх частота, що включає в себе уніграми, біграми чи їх комбінацію. Як було показано іншими авторами, що проводили порівняння ефективності різних методів векторного представлення максимальну ефективність має представлення ‘feature presence’, тобто наявність певних термінів, а не їх частота чи більш складні формули.

Методи, засновані на словниках для класифікації текстів використовують тональний словник. Для його створення, може використовуватися один з трьох методів: складання вручну, методи на основі корпусів та методи на основі словників.

## 2.1 Методи машинного навчання

Задача аналізу тональності є, по свої суті, задачею текстової класифікації. Традиційна класифікація тексту, як правило, класифікує текст за різними темами, наприклад політика, наука, спорт на інші. В такому типі класифікації, ключовими компонентами є певні слова, що відносяться до певної теми. Але в задачі аналізу тональності, емоційно забарвлені слова, або слова сентименту,

що визначають позитивну чи негативну думку є більш важливими, наприклад добре, прекрасно, погано, жахливо.

Навчання з учителем – способ машинного навчання, в ході якого система навчається за прикладами «стимул-реакція». З погляду кібернетики, є одним з видів кібернетичного експерименту. Між входами і еталонними виходами (стимул-реакція) може існувати деяка залежність, але вона невідома. За відомою сукупністю прецедентів – пар «стимул-реакція», що називається навчальною вибіркою, потрібно відновити залежність (побудувати модель, входом якої буде стимул, а виходом – реакція, що буде придатна для прогнозування). Для вимірювання точності відповідей, так само як і в навчанні на прикладах, може вводитися функціонал якості.

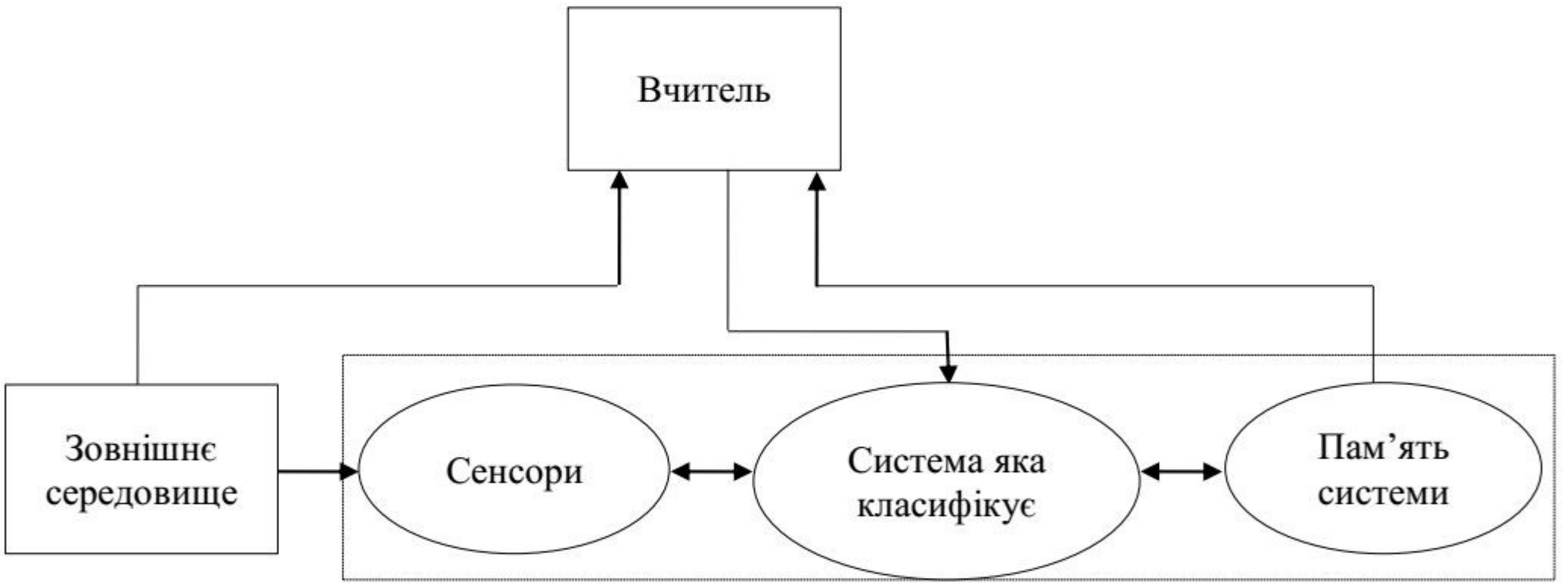



Рисунок 2.1 – Алгоритм навчання з вчителем

Для аналізу тональності тексту необхідно вирішити задачу класифікації, що являє собою системний розподіл досліджуваних предметів чи явищ за видами або типами, за якими-небудь істотними ознаками, та, можливо, розташування їх у певному порядку, що відбиває ступінь цієї схожості. Задача класифікації – формалізована задача, яка полягає у віднесення містить множиний об'єктів до певних класів. Нехай задана кінцева множина об'єктів та класів до яких відноситься кожен з поданих об’єктів (вибірка). Нехай також

невідомо до якого класу належать інші надані об’єкти. Метою задачі є створення алгоритму, що буде здатний до класифікації довільного об'єкту з вихідної множини.

У машинному навчанні завдання класифікації вирішується, як правило, за допомогою методів штучної нейронної мережі.

### 2.1.1 Наївний класифікатор Баєса

Наївний класифікатор Баєса(НБК) – простий імовірнісний класифікатор, заснований на застосуванні Теореми Баєса зі строгими (наївними) припущеннями про незалежність[29]. Перевагою цього підходу є те, що вимоги до розміру вибірки скорочуються від експоненційних до лінійних. Недоліком – те, що модель може бути точною лише у випадку, коли виконується припущення про незалежність, інакше, строго кажучи, обчислені ймовірності вже не є точними. Однак часто результати роботи класифікатора продовжують корелювати з істинною приналежністю образів до класів навіть за умови істотної залежності. Імовірнісна модель для класифікатора – це умовна модель

$$p(C \vee F_1, \dots, F_n) \tag{1.8}$$

від залежної змінної класу $C$ з малою кількістю результатів чи класів, залежна від кількох змінних $F_1$ $F_n$ . Проблема полягає в тому, що коли кількість властивостей $n$ у рівнянні дуже велика чи коли властивість може приймати велику кількість значень, коли будувати таку модель на імовірнісних таблицях стає неможливо. Тому ми переформулюємо модель, щоб зробити її легко оброблюваною. Використовуючи теорему Баєса, запишемо:

$$p(C|F_1, \dots, F_n) = \frac{p(C)p(F_1,\dots,F_n \vee C)}{p(F_1,\dots,F_n)} \tag{1.9}$$

На практиці інтерес викликає лише чисельник цього дробу, оскільки знаменник не залежить від $C$ і значення властивостей $F_i$ дані, тому знаменник – константа. Чисельник еквівалентний сумісній імовірності моделі

$$p(C, F_1, \dots, F_n) \tag{1.10}$$

яка може бути переписана наступним чином, використовуючи повторно додатки визначень умовної імовірності:

$$\begin{aligned} & p(C, F_1, \dots, F_n) = \\ & p(C)p(F_1, \dots, F_n|C) = \\ & \quad p(C)p(F_1|C)p(F_2, \dots, F_n|C, F_1) = \\ & p(C)p(F_1|C)p \end{aligned} \tag{1.11}$$

і так далі. Тепер можна використовувати «наївні» припущення умовної незалежності: припустимо, що кожна властивість $F_i$ умовно незалежна від властивості $Fj$ при $j \neq i$. Це означає:

$$p\left(F_i\middle|C, F_j\right) = p(F_i \vee C) \tag{1.12}$$

таким чином, сумісна модель може бути виражена як:

$$p(C|F_1, \dots, F_n) = p(C)p(F_1|C)p(F_2|C) \dots = p(C)\prod_{i=1}^{n} p(F_i \vee C) \tag{1.13}$$

Це означає, що з припущення про незалежність, умовний розподіл по класовій змінній C може бути виражено так: де Z – це масштабний множник,

$$p(C|F_1, \dots, F_n) = \frac{1}{Z}p(C)\prod_{i=1}^{n} p(F_i \vee C) \tag{1.14}$$

де Z – це масштабний множник, що залежить тільки від $F1, \dots, Fn$, тобто константа, якщо значення змінних відомі. До переваг цього класифікатора можна віднести наступні:

– простота реалізації;

– досить швидкий процес навчання;

– незважаючи на те, що припущення про незалежність класифікаційних ознак не є вірним в природній мові (значення слова залежать від контексту), НБК часто показує хороші результати при текстовій класифікації.

– До недоліків цього методу можна віднести наступні:

– значення, що повертаються при класифікації, не можна трактувати, як імовірнісні. Таким чином не можна відповісти на питання, з якою часткою впевненості вийшов результуючий клас;

– так як в природній мові слова не є незалежними, НБК не є найоптимальнішім методом.

### 2.1.2 Класифікатор AdaBoost

AdaBoost (скор. Adaptive Boosting) є ансамблевим класифікатором, складається з декількох слабких алгоритмів, посилюючи результат їх роботи, формуючи сильний класифікатор. одиночний алгоритм може давати слабкі результати, але якщо об'єднати кілька класифікаторів з вибором тренувального сету на кожній ітерації, отримавши вірні ваги в кінці роботи, будемо мати хорошу підсумкову точність класифікації [30].

AdaBoost тренує алгоритми ітеративно, вибираючи новий тренувальний набір даних на підставі точності, отриманої на попередньому етапі. Приклади, що були класифіковані невірно, набувають більшої ваги, підвищуючи тим шанс потрапити до вибірки іншому класифікатором на наступній ітерації.

Класифікатори також отримують вагу, в залежності від досягнутої ними точності. Якщо точність менше 50% вага негативний, якщо дорівнює 50%, то нульовий, інакше позитивний.

Формула результуючого класифікатора являє собою лінійну комбінацію базових класифікаторів і виглядає наступним чином:

$$H(x) = sign \tag{1.15}$$

де $T$ - число відібраних класифікаторів із загального числа на підставі якогось критерію, $h_t\ (x)$ - базовий класифікатор, $\alpha_t$ - ваговий коефіцієнт цього класифікатора залежить від допустимої ним помилки.

### 2.1.3 Класифікатор із застосуванням логістичної регресії

Логістична регресія здійснює оцінку з урахуванням деяких попередніх даних ймовірність настання будь якої події для конкретного об'єкта[31]. Логістична регресія працює з бінарними даними: подія відбувається (1) чи ні (0). Така логістична регресія називається біноміальної. Існує узагальнення на кілька класів, що носить назву поліноміальний логістична регресія.

Функція логістичної регресії має s-образну криву, обмежену між 0 і 1 по осі ординат, яка показує імовірність того, що подія відбулася. Для обчислення ймовірності такий класифікатор використовує поняття коефіцієнта або відношення шансів. Воно визначається як відношення ймовірності $P\ (y = 1 \mid x)$ того, що подія для об'єкта $x$ відбудеться, до того, що цього не станеться:

$$Odds = \frac{P(y=1 \vee x)}{1-P(y=1 \vee x)} \tag{1.16}$$

Взяття натурального логарифма відношення шансів називають логістичним:

$$Logit\big(P(x)\big) = ln\left(\frac{P(y=1 \vee x)}{1-P(y=1 \vee x)}\right) \quad (1.17)$$

Передбачення настання події може бути сформульовано через лінійну регресію як:

$$P(x) = \beta_0 + \beta_1 x \quad (1.18)$$

де $\beta_0$, $\beta_1$ - коефіцієнти регресії, де $P(x) \in P\ (y = 1 \mid x)$. але ймовірність при цьому приймає значення $P\ (x) \in (-\infty; \infty)$, для її обмеження в потрібних межах формули прирівнюються, перетворюються і виходить остаточне рівняння логістичної кривої:

$$P(y = 1|x) = f(z) = \frac{1}{1+e^{-z}} \quad (1.19)$$

де $z = \beta_0 + \beta_1 x$. У випадку з n незалежними змінними-ознаками $x_i$ і фіктивним $x_0 = 0$, маємо вектор ознак $xx = (1, x_1, ..., xn)$, параметри $\beta = (\beta_0, ..., \beta_n)$ і $z = \beta^T x$. Навчання класифікатора полягає в пошуку коефіцієнтів логістичної регресії, для цього використовується метод максимальної правдоподібності, де відбувається їх підбір і максимізація функції правдоподібності $L$. Для простоти зазвичай використовують максимізацію натурального логарифма цієї функції, на навчальній вибірці.

$$\beta = arg^{max}_{\beta} lnL(\beta) = arg^{max}_{\beta}\left[\sum_{i=1}^{m} logP\{y = y^{(i)} \vee x = x^{(i)}\}\right] \quad (1.20)$$

де $P\ \{y \mid x\} = f\ (\beta^T x)\ y\ (1 - f\ (\beta^T x))^{1-y}$, $y \in \{0,1\}$, $x \in R^n$ – функція розподілу $y$ при даному $y$, $m$ - кількість елементів в навчальній вибірці.

### 2.1.4 Класифікатор із застосуванням дерева прийняття рішень

Дерево прийняття рішень зазвичай застосовують в ситуаціях, коли є багато випадків, кожен з яких описується кінцевим набором атрибутів. У вузлах дерева, які не є листям, знаходяться атрибути, за якими йде поділ випадків-класів при спуску по дереву. Завдання класифікації з використання дерева рішень полягає і в тому, щоб пройтися по ньому зверху вниз і визначити, в який із листя потрапить клас об'єкта[32].

При побудові дерева рішень важливими питаннями є вибір критерію атрибута розбиття, відсікання гілок і зупинку навчання. Обраний атрибут повинен розбивати множину об'єктів так, щоб утворені підмножини належали одному класу або були до нього максимально наближені. Для вибору найбільш підходящого атрибута використовуються різні критерії. Відсікання або заміна піддеревом здійснюється для тих гілок, видалення яких не приведе до зростання помилки. Зупинка побудови дерева задається або його глибиною, або кількістю прикладів (samples), які повинні міститися одержаних вузлах.

### 2.1.5 Класифікатор із застосуванням методу опорних векторів

Метод опорних векторів (англ. Support vector machines) шукає гіперплощину, яка найкраще розділить дану вибірку на два класи. Допускається багатокласова класифікація стратегіями one-vs-all і one-vs-one. Дана вибірка елементів $x_i \in R_n$ і зіставлені їм класи $y_i \in \{-1,1\}$. Об'єкти вибірки представляються точками. Опорні вектори - це точки даних, розташовані ближче всього до гіперплощини, при їх видаленні зміниться становище гіперплощини [33]. Їх вважають критичними елементами набору даних. У простій задачі бінарної класифікації, з лінійно розділеною вибіркою, гіперплощина можна представлятись у вигляді лінії, що розділяє вибірку на два

класи. Чим далі дані лежать від гіперплощини, тим вірніше вони класифіковані. Кращою гіперплощиною вважається та, відстань $\frac{1}{|w|\vee}$ від якої до кожного класу максимальна, тут $w$ - нормальний вектор до роділяючої гіперплощини, яка може бути записана як множина точок $x$, задовольняючих рівняння:

$$wx - b = 0 \tag{1.21}$$

де $b$ - допоміжний параметр. Якщо навчальна вибірка лінійно розділена, можна вибрати дві паралельні гіперплощини так, щоб вони розділили цю множину на два класи. Область між ними називається зазором, маржею або різницею. Ці площини описуються рівняннями:

$$wx - b = 0 \tag{1.22}$$
$$wx - b = 0$$

Мінімізуючи відстань $\frac{1}{|w|\vee}$ і виключаючи попадання в зазор, отримуєм задачу мінімізації:

$$\{|w| \vee v^2 \to miny_i(wx_i - b) \geq 1, \qquad 1 \leq i \leq n \tag{1.23}$$

Таке завдання вважають еквівалентним пошуку сідлової точки функції Лангранжа, зводять до спеціального типу - завдання квадратичного програмування, де присутні лише подвійні змінні $\lambda_i$. Вирішивши це завдання, можна висловити $w$ і $b$ відповідно:

$$\begin{matrix} w = \sum_{i=1}^{n} \lambda_i x_i y_i, \\ b = wx_i - y_i, \end{matrix} \qquad \lambda_i \geq 0 \tag{1.24}$$

Остаточно класифікатор записується як:

$$a(x) = sign(\sum_{i=1}^{n} \lambda_i y_i x - b) \quad (1.25)$$

Якщо вибірка лінійно нероздільна, відбувається відображення векторів в простір більш високої розмірності, при цьому, в наведеній вище формулі відбувається заміна скалярного твори на одну з функцій нелінійного ядра $K$ ($x_i$, $x$). Після чого також будується найкраща розділяє гіперплоскость.

## 2.2 Тестування ефективності методів класифікації

Кожен корпус тексту( проходить обробку, якщо обрана відповідна опція, за замовчуванням це так. Регулярні вирази записані таким чином, щоб очистити текст від небуквених символів, видалити html-теги, посилання. При цьому зберігаються деякі «смайлики» - послідовність знаків, що виражає якусь емоцію. далі при допомоги процесу стемінгу для заздалегідь підготовлених стоп-слів, а також самого тексту кожне слово замінюється його основою. З отриманого «чистого» тексту видаляються стоп-слова, що не несуть ніякої емоційного забарвлення. Описаний вище процес був названий «Нормалізацією тексту». Оскільки одиночний корпус тексту може бути досить великим, не кажучи вже про об'єднання всіх текстів, використання регулярних виразів і нормалізація проводилися в багатопотоковому режимі, був задіяний процесор Intel Core i7-4770K. Було використано 8 віртуальних ядер, виконання нормалізації на об'єднаному наборі даних обсягом 849 763 рядків зайняло 15 хв.

У даній роботі для навчання використовується вибірка lang-ua – навчальні дані розмічених новинних статей, дана вибірка містить 500 тисяч екземплярів з новинних статей зібраний із таких періодичних видань:

- Українська правда
- ТОВ УНІАН

- Громадська організація Hromadske.TV
- ПАТ телерадіокомпанія Люкс, Zaxid.net
- ТОВ Кореспондент
- Експрес
- «ZN.UA»
- Приватне підприємство "Україна молода"
- ПАТ «Видавництво« Високий замок »
- Газета Бровари (Ви маєте право знати)
- Український тиждень

Дана вибірка була розбита 80% для навчання та 20% для тестування. Для більшої об'єктивності порівняння модифікованого методу із стандартними було проведено два додаткових тестування з вибіркою IMDB – вибірка оглядів до фільмів та twitter_comments – вибірка коментарів у твітері. Навчання та тестування моделей для класифікації тональності тексту здійснювались у середовищі Matlab.

Під час порівняння використовуються наступні поняття:

Accuracy – міра яка показує на скільки точно класифікатор визначив тестову вибірку.

Recall – міра яка показує частку з тестового набору яку класифікатор визначив як позитивну.

Precision – міра яка показує часту серед всіх позитивних об'єктів тестового набору класифікатор визначив як позитивну.

Результати класифікації за допомогою моделі НБК наведені в таблиці 2.1. Наївний байесівський класифікатор - простий імовірнісний класифікатор, заснований на застосуванні теореми Байеса зі строгими (наївними) припущеннями про незалежність.

Залежно від точної природи ймовірнісної моделі, наївні байєсовські класифікатори можуть навчатися дуже ефективно. У багатьох практичних

додатках для оцінки параметрів для наївних байєсівських моделей використовують метод максимальної правдоподібності.

Незважаючи на наївний вигляд і, безсумнівно, дуже спрощені умови, наївні байєсовські класифікатори часто працюють набагато краще з багатьма складними текстами.

Перевагою наївного байєсівського класифікатора є мала кількість даних необхідних для навчання, оцінки параметрів і класифікації.

Таблиця 2.1 – Результати роботи наївного байєсівського класифікатора

| НБК | Precision | Recall | Accuracy |
|---|---|---|---|
| Lang-ua | 0.87 | 0.91 | 0.85 |
| IMDB | 0.93 | 0.89 | 0.86 |
| twitter_comments | 0.90 | 0.88 | 0.85 |

Результати класифікації за допомогою моделі AdaBoost наведені в таблиці 2.2. Ключова ідея AdaBoost полягає у використанні зважених версій тих же даних замість їх випадкових підвибірок. Один і той же навчальний (тренувальний) набір багаторазово використовується, тобто цей набір може бути дуже великим, як було потрібно більш ранніми методами підвищення.

Таблиця 2.2 – Результати роботи моделі AdaBoost

| AdaBoost | Precision | Recall | Accuracy |
|---|---|---|---|
| Lang-ua | 0.84 | 0.91 | 0.84 |
| IMDB | 0.93 | 0.88 | 0.87 |
| twitter_comments | 0.89 | 0.91 | 0.85 |

Результати класифікації за допомогою методу логістичної регресії наведені в табл 2.3. логістична регресія застосовується для прогнозування ймовірності виникнення деякої події за значеннями множини ознак. Для цього

вводиться так звана залежна змінна, приймаюча лише одне з двох значень - як правило, це числа 0 (подія не відбулося) і 1 (подія відбулася), і множини незалежних змінних (також званих ознаками, предикторами або регресорів) - речових , на основі значень яких потрібно обчислити ймовірність прийняття того чи іншого значення залежної змінної.

Таблиця 2.3 – Результати роботи моделі на основі логістичної регресії

| Логістична регресія | Precision | Recall | Accuracy |
|---|---|---|---|
| Lang-ua | 0.90 | 0.88 | 0.85 |
| IMDB | 0.89 | 0.93 | 0.87 |
| twitter_comments | 0.91 | 0.93 | 0.89 |

Результати класифікації за допомогою дерева рішень на попередньо оброблених даних наведені в таблиці 2.4. Перевагою даного методу є те, що він не параметричний (не використовуються якісь ймовірні параметри, які збільшують невизначеність результату), немає необхідності в побудові припущень для застосування алгоритму. Метод легко бореться з «викидами» - аномальними значеннями, просто поміщаючи їх в окремий вузол, очищаючи від таких «шумів» інші дані. Недоліками цього методу є відсутність стабільності і те, що він не підходить для побудови дерев з досить складною структурою.

Таблиця 2.4 – Результати роботи дерева прийняття рішень

| Дерево рішень | Precision | Recall | Accuracy |
|---|---|---|---|
| Lang-ua | 0.88 | 0.86 | 0.82 |
| IMDB | 0.83 | 0.93 | 0.82 |
| twitter_comments | 0.87 | 0.95 | 0.84 |

Результати класифікації за допомогою звичайного словникового методу наведена в таблиці 2.5. Існують тезауруси, розмічені силами людей з урахуванням емоційного забарвлення слів, що містяться в них. Такі словники дозволяють визначати тональність тексту без застосування алгоритмів машинного навчання. Тональність тексту визначається як сума тональностей слів, що містяться в розмічених словниках.

Основною проблемою методів, заснованих на словниках є трудомісткість побудови словника: окремого для кожного нового мови і кожної нової тематики.

Таблиця 2.5 – Результати роботи звичайного словникового методу

| Словниковий метод | Precision | Recall | Accuracy |
|---|---|---|---|
| Lang-ua | 0.57 | 0.67 | 0.72 |
| IMDB | 0.60 | 0.61 | 0.68 |
| twitter_comments | 0.59 | 0.68 | 0.68 |

Результати класифікації за допомогою методу опорних векторів наведена в таблиці 2.6. Метод опорних векторів (Support Vector Machine, SVM) - метод, в якому основою є побудова (оптимальної) розділяючої гіперплощини.

Таблиця 2.6 – Результати роботи методу опорних векторів

| SVM | Precision | Recall | Accuracy |
|---|---|---|---|
| Lang-ua | 0.87 | 0.90 | 0.86 |
| IMDB | 0.87 | 0.86 | 0.87 |
| twitter_comments | 0.91 | 0.88 | 0.88 |

За результатами проведеного тестування було виявлено, що на якість класифікації мала вплив довжина даних, чим вона довша тим менша точність.

Найгірші результати в класифікації текстів були отримані із словниковим методом, що швидше за все викликано малим розміром словника. Однак використання методів НБК та логічної регресії дали найвищий показник точності.

## 2.3 Реалізація модифікованого методу

У попередньому підрозділу було проведено тестування існуючих методів класифікації тексту. В результаті тестування найнижчі показники точності були під час класифікації словниковим методом(таб.2.5). Всі інші результати тесту були зведені в одну таблицю(таб.2.7). Згідно з результатами тестування найкращий результат класифікації був отриманий за допомогою методу НБК та методу логістичної регресії.

Таблиця 2.7 – Порівняння ефективності існуючих методів класифікації

| | Precision | Recall | Accuracy |
|---|---|---|---|
| **НБК** | | | |
| Lang-ua | 0.87 | 0.91 | 0.85 |
| IMDB | 0.93 | 0.89 | 0.86 |
| twitter_comments | 0.90 | 0.88 | 0.85 |
| **Логістична регресія** | | | |
| Lang-ua | 0.90 | 0.88 | 0.85 |
| IMDB | 0.89 | 0.93 | 0.87 |
| twitter_comments | 0.91 | 0.93 | 0.86 |
| **AdaBoost** | | | |
| Lang-ua | 0.84 | 0.91 | 0.84 |
| IMDB | 0.93 | 0.88 | 0.85 |
| twitter_comments | 0.89 | 0.91 | 0.85 |
| **Дерево рішень** | | | |
| Lang-ua | 0.88 | 0.86 | 0.82 |
| IMDB | 0.83 | 0.93 | 0.82 |
| twitter_comments | 0.87 | 0.95 | 0.84 |
| **SVM** | | | |

| | | | |
|---|---|---|---|
| Lang-ua | 0.87 | 0.90 | 0.84 |
| IMDB | 0.87 | 0.86 | 0.86 |
| twitter_comments | 0.91 | 0.88 | 0.85 |

Серед методу НБК та методу логістичної регресії було вирішено модифікувати метод НБК оскільки його реалізація простіша і потребує меншу кількість навчальних даних. Даний метод має як позитивні так і негативні сторони.

Позитивні сторони:

– Класифікація, в тому числі багатокласова, виконується легко і швидко.

– Коли допущення про незалежність виконується, НБК перевершує інші алгоритми і при цьому вимагає менший обсяг навчальних даних.

– НБК краще працює з класифікованими ознаками, ніж з безперервними. Для безперервних ознак передбачається нормальний розподіл, що є досить сильним допущенням.

Негативні сторони:

– Якщо в тестовому наборі даних є певне значення ознаки, яке не зустрічалося в навчальному наборі даних, тоді модель присвоїть нульову ймовірність цього значення і не зможе зробити прогноз. Дану проблему можна вирішити за допомогою згладжування. Ще одним обмеженням НБК є припущення про незалежність ознак. В реальності набори повністю незалежних ознак зустрічаються вкрай рідко.

Загальний алгоритм методу НБК зображений на рисунку 2.2.

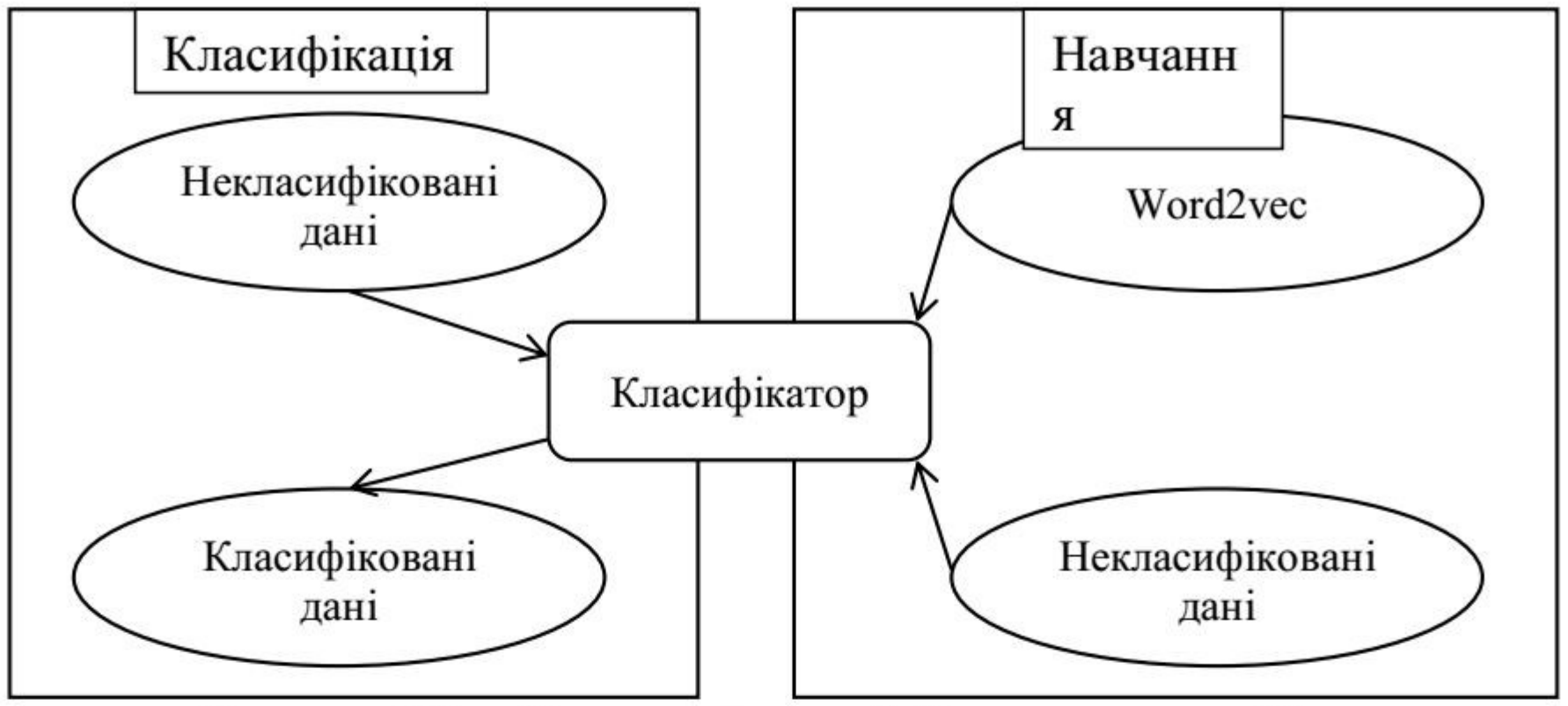



Рисунок 2.2 – Схема наївного байєсового класифікатора

Запропонований метод полягає в наступному: на першому етапі як і зазвичай буде здійснюватись класифікація даних за допомогою словникового методу, а потім класифікований текст буде використаний для навчання іншого класифікатора. Загальна схема запропонованого алгоритму зображена на рисунку 2.3.

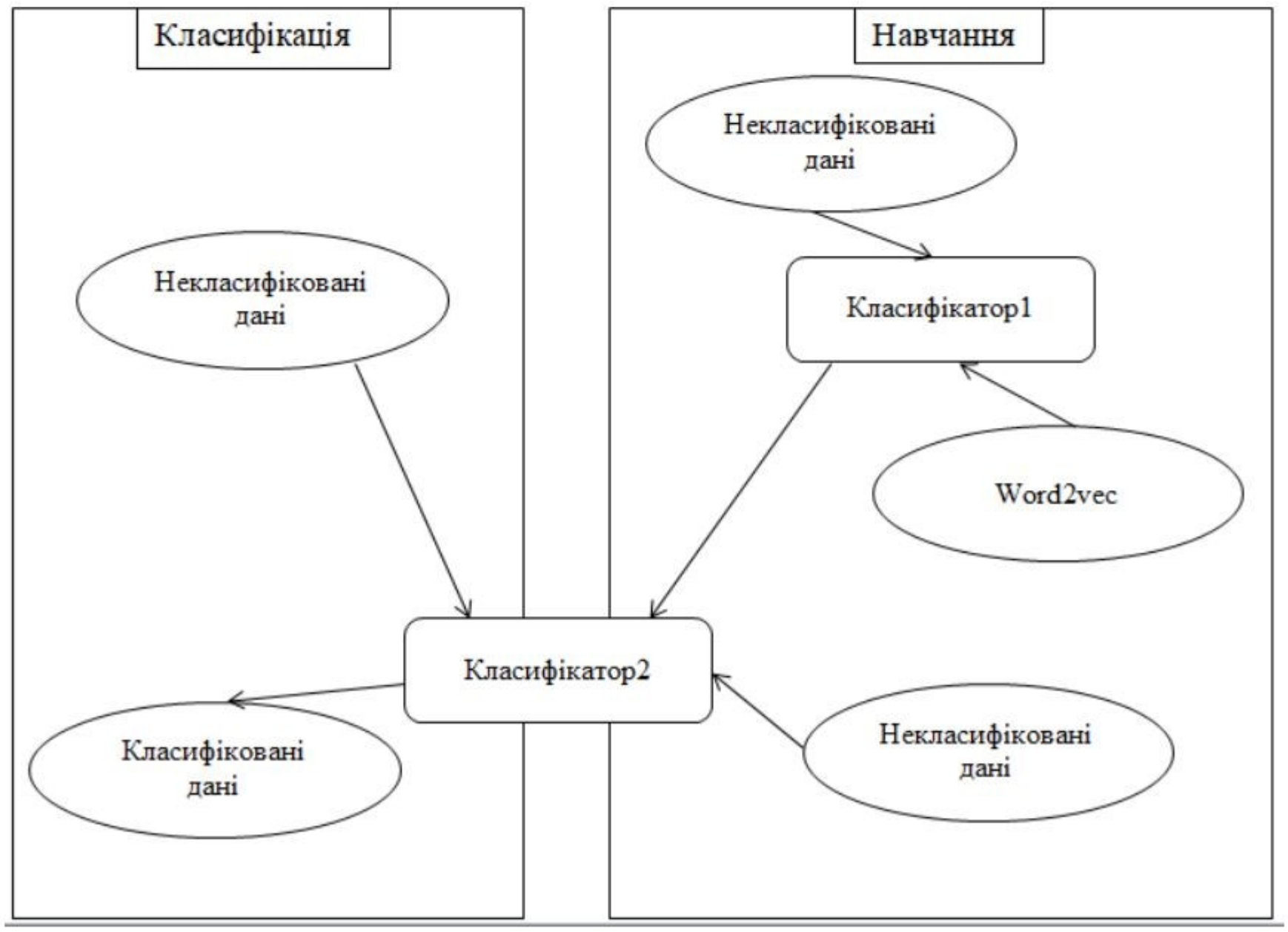



Рисунок 2.3 – Схема запропонованого методу класифікації

В якості можливих варіантів покращення методу пропонується подавати дані у вигляді біграм, з використанням уніграм із використанням двох класифікаторів з уніграмами та використання двох класифікаторів НБК з використанням біграм. Результати тестування усіх 3 запропонованих модифікацій знаходяться у таблиці 2.8.

Таблиця 2.8 – Ефективність роботи запропонованих модифікацій

| Модифікація методу | Precision | Recall | Accuracy |
|---|---|---|---|
| НБК з біграмами | 0.87 | 0.91 | 0.86 |
| НБК+НБК | 0.88 | 0.90 | 0.88 |
| НБК з біграмами+НБК з біграмами | 0.89 | 0.91 | 0.89 |

Результатом тестування визначено, що метод НБК з біграмами має результати які не відрізняються від звичайного методу НБК. Метод НБК+НБК дає кращий результат точності (0,88) порівнювано з стандартним методом(0,85). Третій варіант застосування для навчання і класифікації двох класифікаторів НБК з використанням біграм має найвищу ефективність серед трьох запропонованих методів(0,89), але в порівнянні з методом НБК+НБК різниця складає лише 1,14%. Тому потрібно визначити не тільки точність але й швидкість класифікації для визначення який із методів дасть найкращий результат(таб.2.9).

Таблиця 2.9 – Час роботи модифікацій методу за умови 500 тис екземплярів

| Модифікація методу | Швидкість навчання(500 тис екземплярів) | Швидкість класифікації (500 тис екземплярів) |
|---|---|---|
| НБК з біграмами | 579 | 542 |
| НБК+НБК | 898 | 397 |
| НБК з біграмами+НБК з біграмами | 1151 | 545 |

При порівнюванні методу НБК+НБК та НБК з біграмами+ НБК з біграмами швидкість навчання відрізняється на 28% і швидкість класифікації на 27%. В результаті для розробки програмного засобу став метод НБК+НБК.

Висновки за розділом. В даному розділі було проаналізовано ефективність стандартних методів класифікації тональності, запропоновано декілька варіантів модифікації обраного алгоритму. Проведено їх порівняння відповідно до обраних критеріїв, та доведено ефективність запропонованого методу.

# 3 РОЗРОБКА ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАСОБУ

## 3.1 Обгрунтування вибору платформи та мови програмування

В якості мови та інструментів було обрано NodeJS, python,Flask, Keras, pandas.

Програмний засіб буде розроблятись у вигляді веб-додатку для цього буде доцільно використати для python для розробки модулів сентиментального аналізу тексту, а для розробки інтерфейсу застосувати NodeJS.

Python - високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує структурние, об'єктно-орієнтоване, функціональнеі аспектно-орієнтоване програмування. Основні архітектурні риси - динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується розбиття програм на модулі, які, в свою чергу, можуть об'єднуватися в пакети.

Node або Node.js - програмна платформа, що перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення. Node.js додає можливість JavaScript взаємодіяти з пристроями введення-виведення через свій API (написаний на C ++), підключати інші зовнішні бібліотеки, написані на різних мовах, забезпечуючи виклики до них з JavaScript-коду. Node.js застосовується переважно на сервері, виконуючи роль веб-сервера, але є можливість розробляти на Node.js і десктопні віконні додатки і навіть програмувати мікроконтролери.

Pandas - програмна бібліотека мовою Python для обробки і аналізу даних. Робота pandas з даними будується поверх бібліотеки NumPy, що є інструментом

нижчого рівня. Надає спеціальні структури даних і операції для маніпулювання числовими таблицями і тимчасовими рядками. Назва бібліотеки походить від економетричного терміна «панельні дані», використовуваного для опису багатовимірних структурованих наборів інформації.

Keras - відкрита нейромережева бібліотека, написана на мові Python. Виrкористовується для роботи над фреймворками Deeplearning4j, TensorFlow і Theano. Орієнтована на оперативну роботу з вимірюванням глибокого навчання, при цьому спроектована так, щоб бути компактною, модульною та має можливість розширення. Ця бібліотека використовує досить широко використовувані будівельні блоки нейронних вимірювачів, як шари, цільові та передавальні функції, оптимізатори та без додаткових інструментів для спрощення роботи з зображеннями та текстами.

Flask - фреймворк для створення веб-додаткових пристроїв для програмування Python, який використовує інструменти для роботи Werkzeug, а також шаблонізатор Jinja2. Відноситься до категорії так званих мікрофреймворкісв - достовірних каркасів веб-додатків, надають лише основні функції.

## 3.2 Аналіз вимог користувача до програмного продукту

В сучасномі світі користувачі мережі інтернет зустрічають безліч новин як шукаючи їх власноруч так і зустрічаючи новинні заголовки в контекстній рекламі, переходячи на новинний сайт. Часто користувачі не хочуть читати наприклад негативні новини, саме для цього і потрібний класифікатор. Розробка програмного засобу у вигляді окремого додатку є не ефективним оскільки користувачі зазвичай не хочуть переключатись між кількома вікнами. Тому буде доцільно розробити засіб у вигляді додатку до браузера.

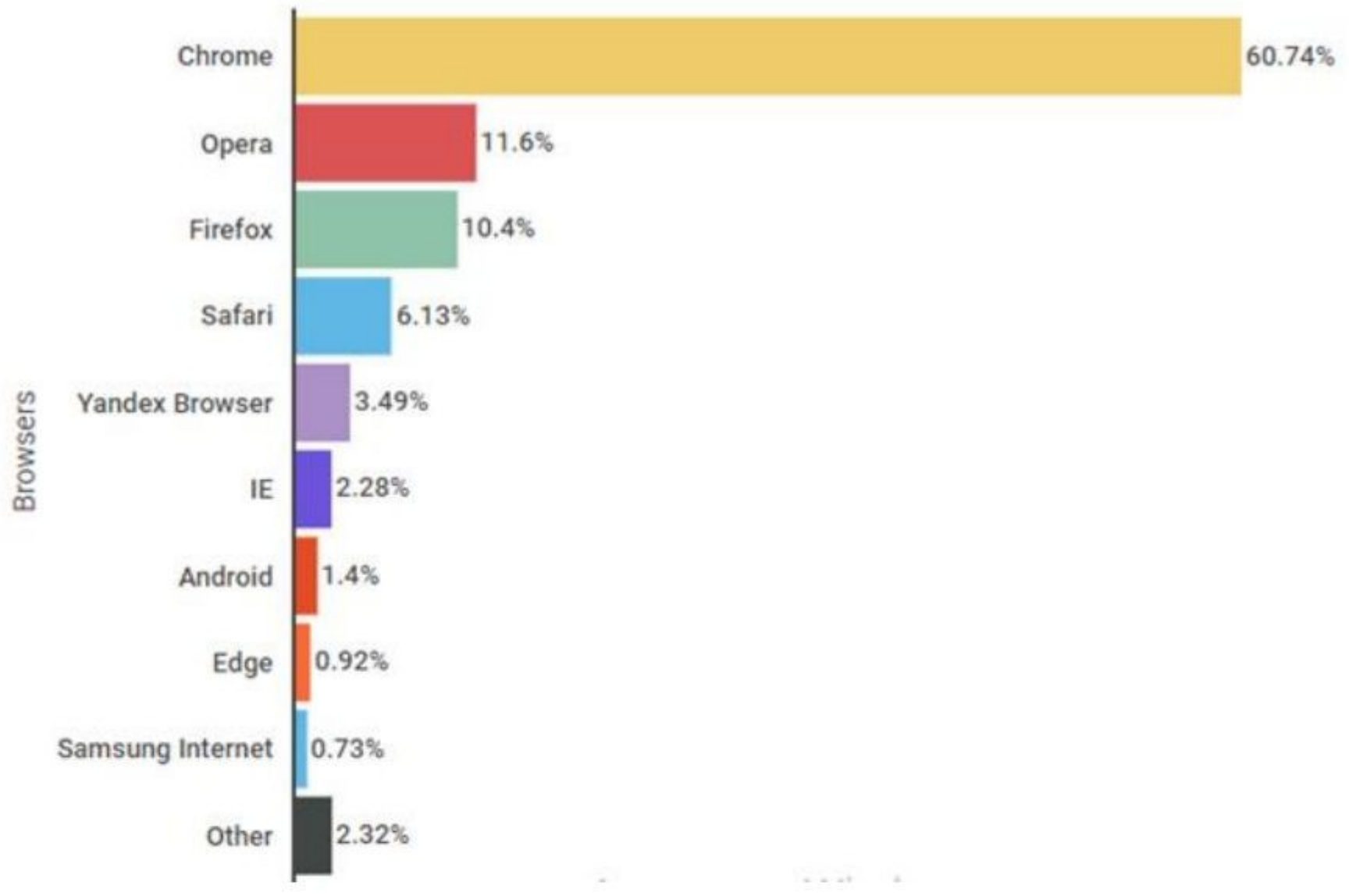



Рисунок 3.1 – Популярність браузерів в Україні

Оскільки найбільш популярний браузер Chrome тому розширення буде розроблятись для нього.

## 3.3 Розробка та тестування програмного засобу

Загальний принцип роботи програмного засобу виглядає наступним чином:

користувач заходить на сайт із новиною;

функція Background.js отримує текстовий вміст із сторінки;

за допомогою Porter-stemmer.js отриманий текст приводиться до нормальної форми;

Background-load-objects.js здійснює пошук та видалення стоп-слів у тексті;

кінцевий оброблений текст класифікується за допомогою моделі яка відкривається у service.py.

Архітектура взаємодії елементів розробленої системи зображена на рисунку 3.2:

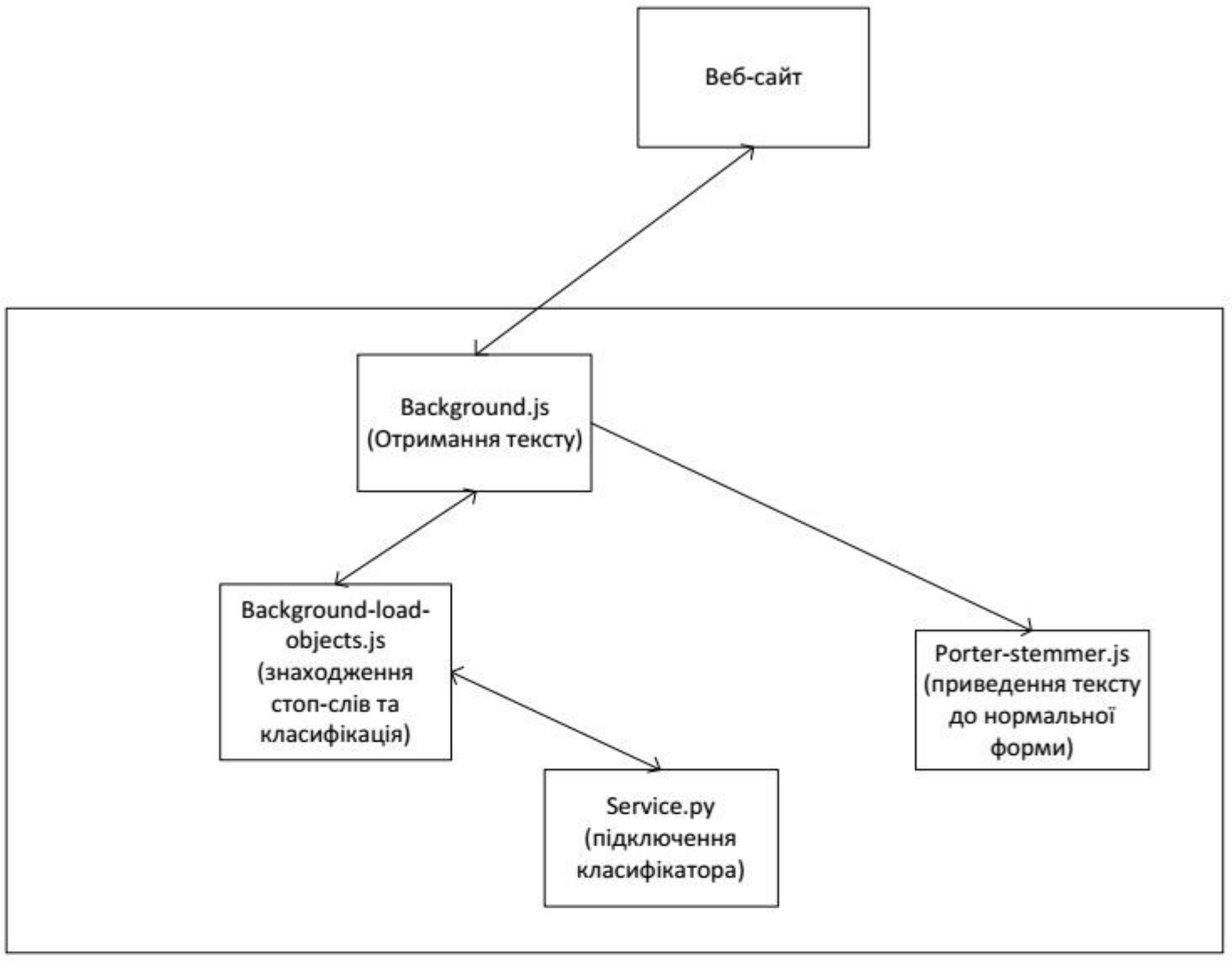



Рисунок 3.2 – Архітектура системи

Загальний вигляд структури файлів проекту зображено на рисунку 3.3

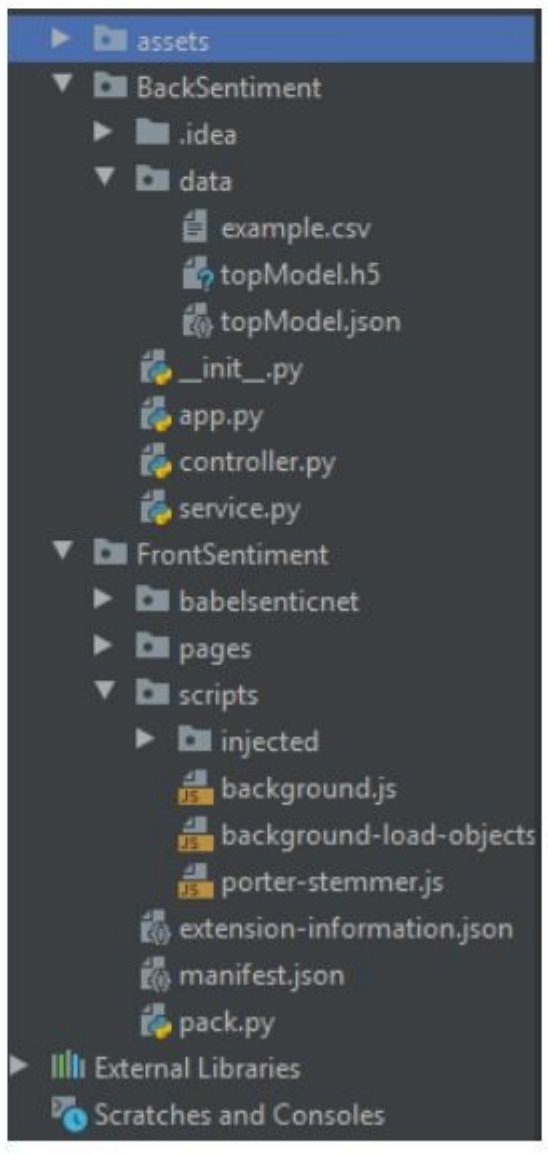



Рисунок 3.3 – Структура файлів проекту

Загальна схема роботи розробленого програмного засобу зображена на рисунку 3.2:


Рисунок 3.4 – Блок-схема роботи програмного засобу

Для більш точного аналізу тексту з веб-сайту при його витягуванні потрібно видалити зайві частини тексту які не мають впливу на тональність. За допомогою ряду правил відсікає суфікси та закінчення слів. Для цього був розроблений стемер на основі відомого стеммера Портера який працює на правилах англійської мови. Було проаналізовано правила побудови слів в українській мові та виділено необхідні частини слів, що збільшує швидкість класифікації тексту.

```
class UkrainianStemmer():
    def __init__(self, word):
        self.word = word
        self.vowel = r'аеиоуюяіїє'
        self.perfectiveground = r'(ив|ивши|ившись|ыв|ывши|ывшись((?<=[ая])(в|вш
и|вшись)))$'
        self.reflexive = r'(с[яьи])$'
        self.adjective = r'(ими|ій|ий|а|е|ова|ове|ів|є|їй|єє|еє|я|ім|ем|им|ім|и
х|іх|ою|йми|іми|у|ю|ого|ому|ої)$'
        self.participle = r'(ий|ого|ому|им|ім|а|ій|у|ою|ій|і|их|йми|их)$'
        self.verb = r'(сь|ся|ив|ать|ять|у|ю|ав|али|учи|ячи|вши|ши|е|ме|ати|яти|
є)$'
        self.noun = r'(а|ев|ов|е|ями|ами|еи|и|ей|ой|ий|й|иям|ям|ием|ем|ам|ом|о|
у|ах|иях|ях|ы|ь|ию|ью|ю|ия|ья|я|і|ові|ї|ею|єю|ою|є|еві|ем|єм|ів|їв|ю)$'
        self.rvre = r'[аеиоуюяіїє]'
        self.derivational = r'[^аеиоуюяіїє][аеиоуюяіїє]+[^аеиоуюяіїє]+[аеиоуюяі
їє].*(?<=о)сть?$'
        self.RV = ''
```

Навчена модель для класифікації зберігається у файлі topModel.h5. Дані з моделі для класифікації зчитуються у модулі файлу service.py :

```
json_file = open("./data/topModel.json", "r")
loaded_model_json = json_file.read()
json_file.close()
loaded_model = keras.models.model_from_json(loaded_model_json)
loaded_model.load_weights("./data/topModel.h5")
```

Оскільки користувачу важливо знати одразу, яка тональність новини яку він переглядає тому дане розширення самостійно проводить аналіз при відкриванні сторінки :

```
else if (request.type == "autoParagraphScan" || request.type == "autoPara
graphScanData") {
                    if (request.type == "autoParagraphScanData") {
                        requestNumber = request.requests;
                } else {
                    collectedScanData.push(getSentimentAverage(request.content,
"rightClick", null));
                    if (collectedScanData.length == requestNumber) {
```

```
                    chrome.tabs.query({active: true, currentWindow: true}, (ta
bs) => {
                        chrome.tabs.sendMessage(tabs[0].id, {type: "COLOR-PG-
DATA", data: collectedScanData}, null);
                        collectedScanData = [];
                    });
```

Також користувач може дізнатись детальніше про певний уривок тексту окремо.

```
storage.get(['COLOR_SELECTION'], (result) => {
    if (result.COLOR_SELECTION) {
        chrome.tabs.query({active: true, currentWindow: true},(tabs) => {
            chrome.tabs.sendMessage(tabs[0].id, {type: "COLOR-
PH", data: res, content: request.title}, null);
        });
```

Вигляд інтерфейсу зображений на рисунку 3.5 .

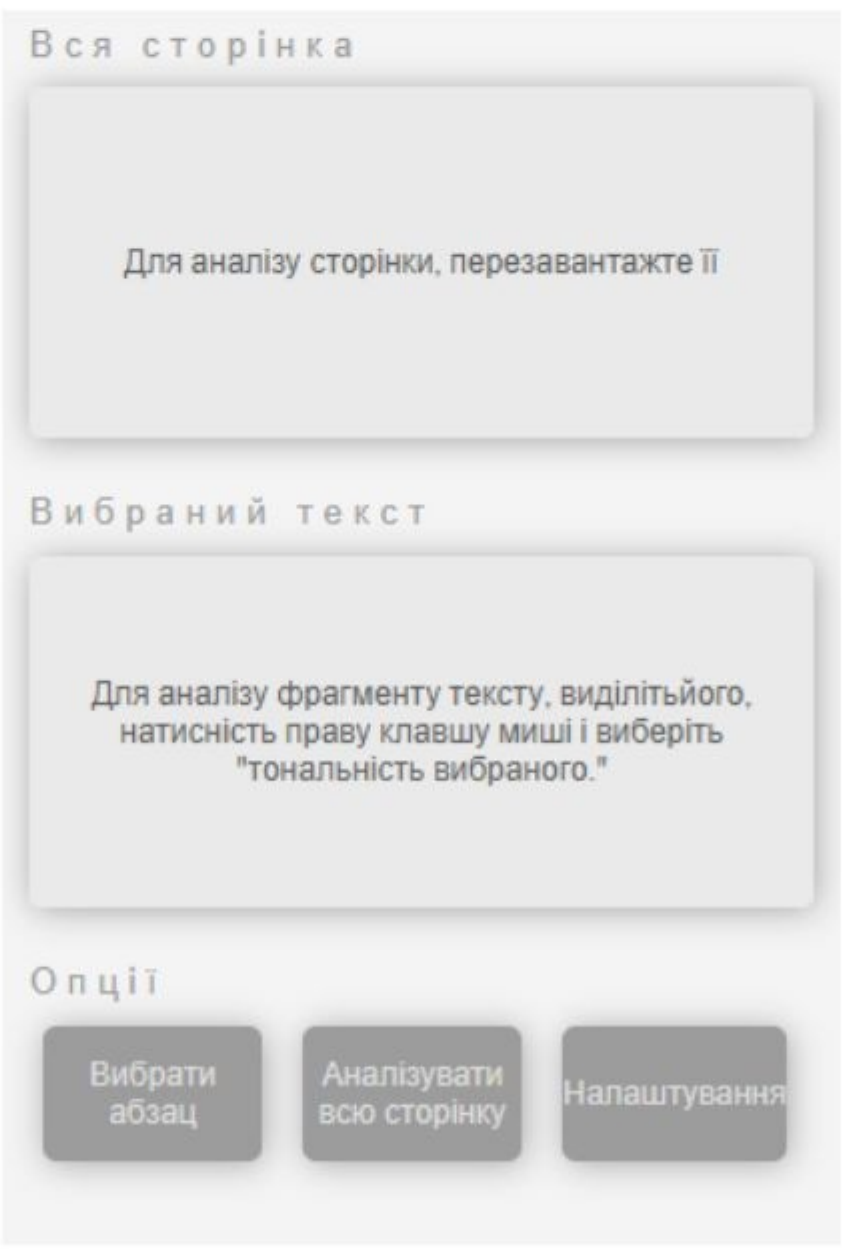



Рисунок 3.5 – Вигляд інтерфейсу програмного засобу

Інтерфейс складається з наступних елементів:

- поле з результатом аналізу усієї сторінки;
- поле з результатом вибраної частини тексту;
- кнопка для аналізу частини тексту;
- кнопка для аналізу усієї сторінки;
- кнопка налаштувань.

Також було створено модуль для  налаштування деяких параметрів:

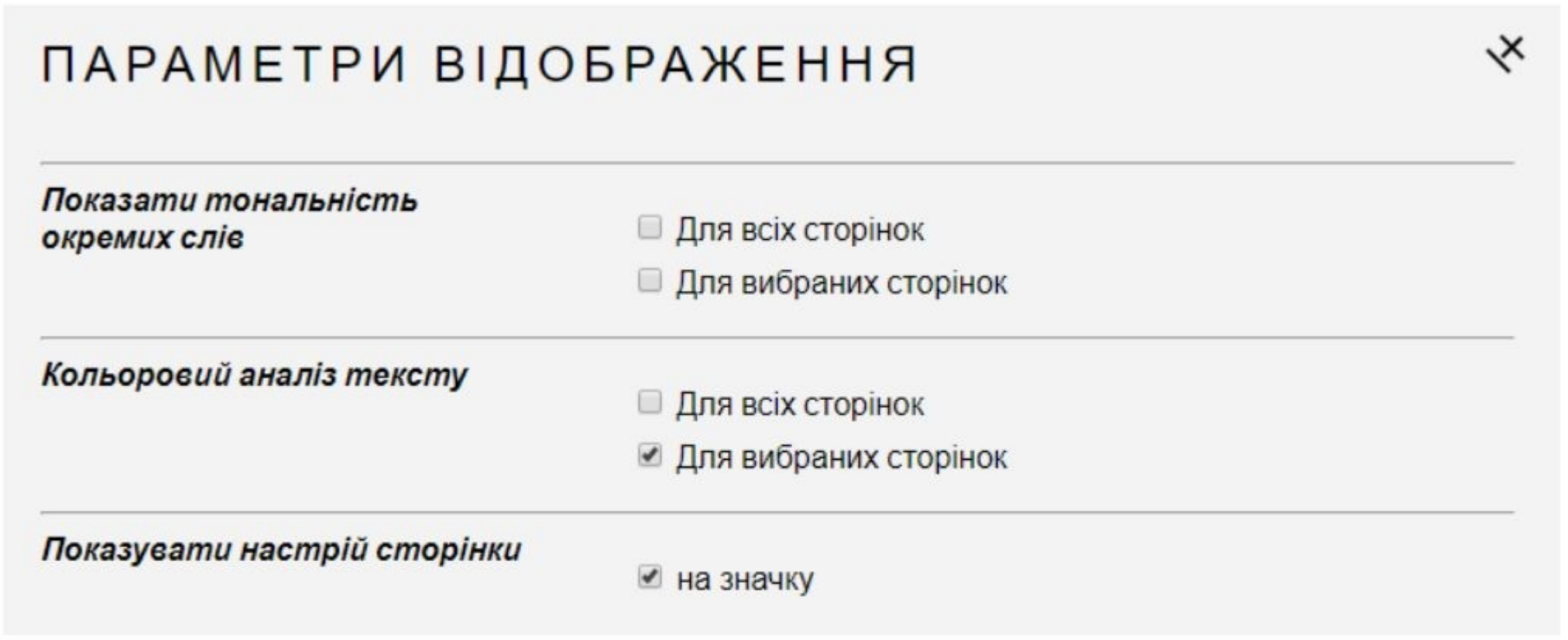



Рисунок 3.6 – Вікно з налаштуваннями програми

У вікні налаштувань користувачеві доступні наступні параметри:

Пункт «показувати тональність окремих слів» дає можливість отримувати тональність кожного слова яке вдалось класифікувати на сторінці або у вибраному фрагменті тексту.

Кольоровий аналіз тексту дає можливість аналізувати виділений текст. Якщо вибрати пункт «для всіх сторінок», автоматично буде виділятись весь текст для аналізу. При виборі пункту «для вибраних сторінок» аналіз буде здійснюватись лише на фрагменті тексту який виділяє сам користувач на сторінці.

Пункт «Показувати настрій сторінки» дає можливість відображення оцінки тональності на панелі розширень.

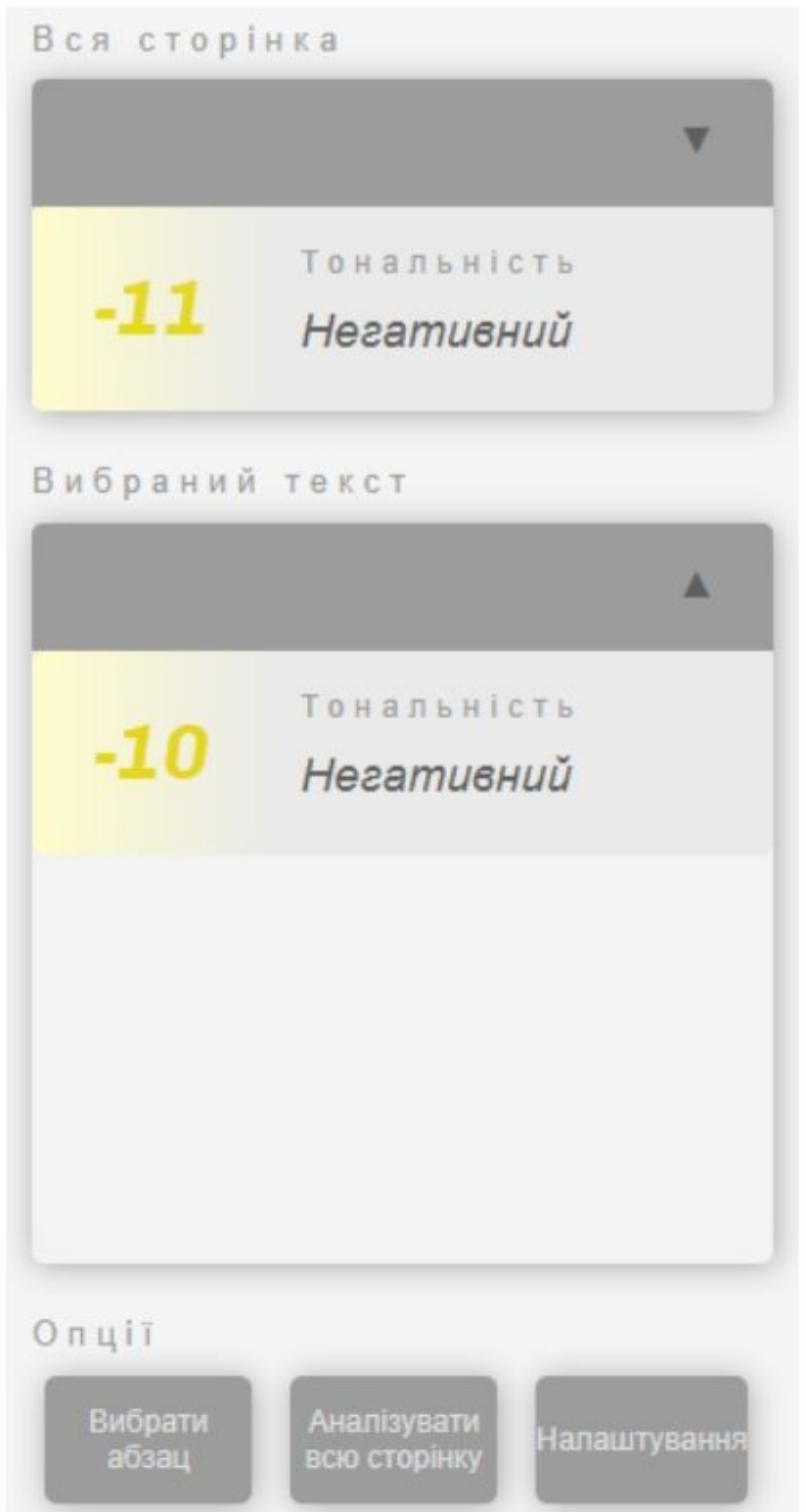



Рисунок 3.7 – Результат тестування

Як можна побачити на рис. 3.7 результат аналізу тональності усієї сторінки на якій йдеться про маніпуляції в засобах масової інформації -11, тобто це що свідчить про негативну тональність тексту. При аналізі наступного фрагменту тексту «Держрегулювання ЗМІ може зіткнутися з низкою ускладнень. Не завжди можна знайти власника або редактора сайту. До того ж, особливо в українських реаліях, втручання держави може становити загрозу свободі слова» було отримано оцінку -10 що є негативним результатом і опираючись на текст отримана оцінка є коректною.

Для початку роботи користувачеві достатньо встановити розширення у браузер після чого він буде автоматично аналізувати вміст всіх сторінок на сайті.

Висновки за розділом. В результаті розробки було використано модель для класифікації тексту НБК+НБК, що була розроблена та протестована в попередньому розділі. Створено програмний засіб для визначення тональності тексту на веб-сторінці у вигляді розширення для браузера. Створена можливість аналізувати як фрагмент тексту так і всю сторінку цілком. Додана можливість для користувача налаштувати програмний засіб.

В результаті тестування засобу було визначено що програма працює без помилок, інформація з сторінки отримується та аналізується коректно.

Розроблений програмний засіб дає оцінку тональності щодо негативного чи позитивного ключа, в якому здійснюється висловлювання. Отримана оцінка може свідчити про імовірне здійснення деструктивного впливу. Результати моделювання дають можливість запропонувати управлінські рішення щодо мінімізації наслідків ймовірних деструктивних інформаційних впливів.

# 4 АНАЛІЗ КОМЕРЦІЙНОГО ПОТЕНЦІАЛУ РОЗРОБКИ

## 4.1 Визначення рівня комерційного потенціалу розробки методу та засобу захисту від деструктивних інформаційних впливів

4.2

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки методу та засобу захисту від деструктивних інформаційних впливів. В результаті оцінювання можна буде зробити висновок щодо напрямів (особливостей) організації подальшого її впровадження з врахуванням встановленого рейтингу.

Для проведення технологічного аудиту залучимо 3-х незалежних експертів. У нашому випадку такими експертами будуть керівник магістерської роботи та провідні викладачі випускової та споріднених кафедр.

Оцінювання комерційного потенціалу розробки методу та засобу захисту від деструктивних інформаційних впливів будемо здійснювати за 12-ю критеріями згідно рекомендацій.

Результати оцінювання комерційного потенціалу розробки методу та засобу захисту від деструктивних інформаційних впливів заносимо до таблиці 1.1.

Таблиця 4.1 - Результати оцінювання комерційного успіху розробки методу та засобу захисту від деструктивних інформаційних впливів

| Критерії | Експерти | | |
|---|---|---|---|
| | к.т.н. доц. Дудатьєв А. В. | к.т.н., ст. викл Лукічов В. В. | к.т.н., доцент Войтович О.П. |
| | Бали, виставлені експертами | | |
| 1 | 1 | 2 | 2 |
| 2 | 2 | 3 | 2 |
| 3 | 3 | 4 | 3 |
| 4 | 2 | 3 | 3 |

| | | | |
|---|---|---|---|
| 5 | 1 | 2 | 1 |
| 6 | 2 | 3 | 3 |
| 7 | 3 | 4 | 3 |
| 8 | 2 | 3 | 2 |
| 9 | 1 | 2 | 2 |
| 10 | 2 | 2 | 2 |
| 11 | 3 | 2 | 3 |
| 12 | 1 | 2 | 1 |
| Сума балів | 23 | 32 | 27 |
| Середньоарифметична сума балів, СБ | 27 | | |

За даними таблиці 4.1 робимо висновок щодо рівня комерційного потенціалу розробки методу та засобу захисту від деструктивних інформаційних впливів. При цьому користуємося рекомендаціями, наведеними в таблиці 4.2.

Таблиця 4.2 – Рівні комерційного потенціалу розробки

| Середньоарифметична сума балів, розрахована на основі висновків експертів | Рівень комерційного потенціалу розробки |
|---|---|
| 0 – 10 | Низький |
| 11 – 20 | Нижче середнього |
| 21 – 30 | Середній |
| 31 – 40 | Вище середнього |
| 41 – 50 | Високий |

Таким чином, робимо висновок, щодо рівня комерційного потенціалу нашої розробки методу та засобу захисту від деструктивних інформаційних впливів – середній.

## 1 Визначення рівня якості розробки методу та засобу захисту від деструктивних інформаційних впливів

Оцінювання рівня якості розробки методу та засобу захисту від деструктивних інформаційних впливів проводиться з метою порівняльного аналізу і визначення найбільш ефективного, з технічної точки зору, варіанта інженерного рішення.

Рівень якості – це кількісна характеристика міри придатності певного виду продукції для задоволення конкретного попиту на неї при порівнянні з відповідними базовими показниками за фіксованих умов споживання.

Абсолютний рівень якості розробки методу та засобу захисту від деструктивних інформаційних впливів знаходимо обчисленням вибраних для її вимірювання показників, не порівнюючи їх із відповідними показниками аналогічних виробів. Для цього необхідно визначити зміст основних функцій, які повинні реалізовувати розробка, вимоги замовника до неї, а також умови, які характеризують експлуатацію, визначають основні параметри, які будуть використані для розрахунку коефіцієнта технічного рівня виробу. Система параметрів, прийнята до розрахунків, повинна достатньо повно характеризувати споживчі властивості інноваційного товару (його призначення, надійність, економічне використання ресурсів, стандартизація тощо).

Далі визначаємо величину параметрів якості в балах та встановлюємо граничні його значення (кращі, гірші, середні). Всі ці дані для кожного параметра заносимо в табл. 4.3.

Таблиця 4.3 – Основні параметри методу та засобу захисту від деструктивних інформаційних впливів

| Параметри | Абсолютне значення параметра | | | Коефіцієнт вагомості параметра |
|---|---|---|---|---|
| | Краще +5…+4 | Середнє +3 | Гірше +1…+2 | |

| | | | | |
|---|---|---|---|---|
| Швидкодія | | 3 | | 0,2 |
| Точність аналізу | | 3 | | 0,2 |
| Доступність | 4 | | | 0,4 |
| Імовірність похибки | | 3 | | 0,2 |

Із врахуванням коефіцієнтів вагомості відповідних параметрів можна визначити абсолютний рівень якості інноваційного рішення за формулою:

$$К_{я.а.} = \sum_{i=1}^{n} Рні \cdot аі, \quad (4.1)$$

де Рні – числове значення і-го параметра інноваційного рішення, n – кількість параметрів інноваційного рішення, що прийняті для оцінювання, аі – коефіцієнт вагомості відповідного параметра (сума коефіцієнтів вагомості всіх параметрів повинна дорівнювати 1).

Отже, абсолютний рівень якості методу та засобу захисту від деструктивних інформаційних впливів становитиме – 3,4 бали.

Одночасно визначаємо відносний рівень якості методу та засобу захисту від деструктивних інформаційних впливів, що виробляється (проектується), порівнюючи її показники з абсолютними показниками якості найліпших вітчизняних та зарубіжних аналогів (товарів-конкурентів) (табл. 4.4).

Таблиця 4.4 – Основні параметри методу та засобу захисту від деструктивних інформаційних впливів та товару-конкурента

| Параметри | Варіанти | | Відносний показник якості | Коефіцієнт вагомості параметра |
|---|---|---|---|---|
| | Базовий (конкурент) | Новий | | |
| Швидкодія | 1 | 2 | 2 | 0,2 |

| | | | | |
|---|---|---|---|---|
| Точність аналізу | 80% | 90% | 1,125 | 0,2 |
| Доступність | 1 | 3 | 3 | 0,4 |
| Імовірність похибки | 10-30% | 10-20% | 1,5 | 0,2 |

Відносний рівень якості методу та засобу завадостійкого розподілу секрету визначаємо за формулою:

$$К_{я.в.} = \sum_{i=1}^{n} qi \cdot ai, \quad (4.2)$$

За розрахунками відносний рівень якості методу та засобу захисту від деструктивних інформаційних впливів становитиме – 2,125. Це означає, що наша розробка краща за якістю на 72% від товару-аналога.

2 Визначення конкурентоспроможності розробки методу та засобу захисту від деструктивних інформаційних впливів

У найширшому розумінні конкурентоспроможність товару – це можливість його успішного продажу на певному ринку і в певний проміжок часу. Водночас конкурентоспроможною можна вважати лише однорідну продукцію з технічними параметрами і техніко-економічними показниками, що ідентичні аналогічним показникам уже проданого товару. Для того, щоб високоякісний товар був одночасно і конкурентоспроможним, він має відповідати критеріям оцінювання споживачів конкретного ринку в конкретний час.

Дані для розрахунку загального показника конкурентоспроможності розробки необхідно занести до таблиці 4.5.

Таблиця 4.5 – Нормативні, технічні та економічні параметри методу та засобу захисту від деструктивних інформаційних впливів і товару-конкурента

| Параметри | Варіанти | | Відносний показник якості | Коефіцієнт вагомості параметра |
|---|---|---|---|---|
| | Базовий (конкурент) | Новий | | |
| Швидкодія | 1 | 2 | 2 | 0,2 |
| Точність аналізу | 80% | 90% | 1,125 | 0,2 |
| Доступність | 1 | 3 | 3 | 0,4 |
| Імовірність похибки | 10-30% | 10-20% | 1,5 | 0,2 |
| Ціна за продукт, тис. грн. | 18000 | 4500 | 0,25 | - |

Загальний показник конкурентоспроможності розробки (К) з урахуванням вищезазначених груп показників визначаємо за формулою:

$$К = \frac{Іт.п.}{Іе.п.} = \frac{2,125}{0,25} = 8,5, \quad (4.3)$$

де Іт.п. – індекс технічних параметрів (відносний рівень якості інноваційного рішення); Іе.п. – індекс економічних параметрів.

$$Іе.п. = \frac{РНеі}{РБеі} = \frac{4500}{18000} = 0,25, \quad (4.4)$$

де РНеі, РБеі – економічні параметри (ціна придбання та споживання товару) відповідно нового та базового товарів.

Згідно розрахунків загальний показник конкурентоспроможності – 2,07 . Це означає, що наша розробка методу та засобу захисту від деструктивних інформаційних впливів більш конкурентна у 8,5 разів від товару-аналога.

## 4.2 Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи

1 Розрахунок витрат, що стосуються виконавців розробки методу та засобу захисту від деструктивних інформаційних впливів

Основна заробітна плата кожного із розробників (дослідників) Зо, якщо вони працюють в наукових установах бюджетної сфери:

$$Зо = \frac{М}{Тр} \cdot t, \qquad (4.5)$$

де М – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.

У 2019 році величини окладів (разом з встановленими доплатами і надбавками) рекомендується брати в межах (5000…10000) грн. за місяць; Тр – число робочих днів в місяці; приблизно Тр = (21…23) дні; t – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 4.6.

Таблиця 2.1 – Заробітна плата розробників

| Посада | Місячний осадовий оклад, грн. | Оплата за робочий день, грн. | Число днів роботи | Витрати на заробітну плату, грн. |
|---|---|---|---|---|
| Керівник | 20000 | 909 | 10 | 9090 |
| Інженер-програміст | 15000 | 682 | 10 | 6820 |
| Всього: | | | | 15910 |

Основна заробітна плата робітників Зр, якщо вони беруть участь у виконанні даного етапу роботи і виконують роботи за робочими професіями у

випадку, коли вони працюють в наукових установах бюджетної сфери, розраховується за формулою:

$$Зр = \sum_{i=1}^{n} ti \cdot Ci, \quad (4.6)$$

де ti – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; Ci – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. Сі визначається за формулою:

$$Ci = \frac{Мм \cdot Кі}{Тр \cdot Тзм}, \quad (4.7)$$

де Мм – розмір мінімальної заробітної плати за місяць, грн.; в 2019 році мінімальна заробітна плата становить – 4173 грн., Кі – тарифний коефіцієнт робітника відповідного розряду, Тр – число робочих днів в місяці; приблизно Тр = 21…23 дні; Тзм – тривалість зміни, зазвичай Тзм = 8 годин.

Таблиця 4.7 – Заробітна плата робітників

| Найменування робіт | Трудомісткість, н-год. | Розряд роботи | Погодинна тарифна ставка | Тариф. коеф. | Величина, грн. |
|---|---|---|---|---|---|
| Налагоджувальні | 8 | 5 | 34,4 | 1,45 | 275,2 |
| Складальні | 2 | 3 | 30,1 | 1,27 | 60,2 |
| Всього | | | | | 335,4 |

Додаткова заробітна плата Зд всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10…12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Зд = 0{,}1 \cdot (Зр + Зо) = 0{,}1 \cdot (15910 + 335{,}4) = 1624{,}5 \text{ грн.} \quad (4.8)$$

Нарахування на заробітну плату Нзп розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

де Зо – основна заробітна плата розробників, грн.; Зр – основна заробітна плата робітників, грн.; Зд – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

$$Нзп = 0{,}22 \cdot (Зр + Зо + Зд) = 0{,}22 \cdot (15910 + 335{,}4 + 1624{,}5) =$$
$$= 3932 \text{ грн.} \qquad (4.9)$$

Амортизація обладнання, комп'ютерів та приміщень А, які використовувались під час (чи для) виконання даного етапу роботи.

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

У спрощеному вигляді амортизаційні відрахування А в цілому бути розраховані за формулою:

$$А = \frac{Ц \cdot На}{100} \cdot \frac{Т}{12}, \qquad (4.10)$$

де Ц – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; На – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що На = (10...25)%; Т – термін, використання обладнання, приміщень тощо, місяці.

Таблиця 4.8 - Амортизаційні відрахування

| Найменування | Ціна, грн. | Норма амортизації, % | Термін використання, м. | Сума амортизації |
|---|---|---|---|---|
| ПК | 15000 | 20 | 2 | 500 |

| | | | | |
|---|---|---|---|---|
| Інше обладнання | 3000 | 10 | 2 | 50 |
| Всього | 550 | | | |

Витрати на силову електроенергію Ве, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

$$Ве = В \cdot П \cdot Ф \cdot Кп, грн. \quad (4.11)$$

В – вартість 1 кВт-год. електроенергії, в 2019 р. В ≈ 8,45 грн./кВт; П – установлена потужність обладнання, кВт; Ф – фактична кількість годин роботи обладнання, годин, Кп – коефіцієнт використання потужності; Кп < 1.

Потужність обладнання складає – 0,5 кВт.

Кількість годин роботи складає – 350 годин.

Коефіцієнт викор. потужності -0,9.

Ве=1331 грн.

Інші витрати Він охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо.

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

$$Ів = 1 \cdot (Зо + Зр) = 1 \cdot (15910 + 335{,}4) = 16245 \text{ грн.} \quad (4.12)$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи – В.

$$В = 39928 \text{ грн.}$$

1 Розрахунок собівартості розробки методу та засобу захисту від деструктивних інформаційних впливів

Витрати на силову електроенергію Ве, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

$$Ве = В \cdot П \cdot Ф \cdot Кп, грн$$

В – вартість 1 кВт-год. електроенергії, в 2019 р. В ≈ 8,45 грн./кВт; П – установлена потужність обладнання, кВт; Ф – фактична кількість годин роботи обладнання, годин, Кп – коефіцієнт використання потужності; Кп < 1.

Потужність обладнання складає – 0,5 кВт.

Кількість годин роботи складає – 350 годин.

Коефіцієнт викор. потужності -0,9.

Ве=1331 грн.

Основна заробітна плата робітників Зр, якщо вони беруть участь у виконанні даного етапу роботи і виконують роботи за робочими професіями у випадку, коли вони працюють в наукових установах бюджетної сфери, розраховується за формулою:

$$Зр = \sum_{i=1}^{n} ti \cdot Ci, \quad (4.13)$$

де ti – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; Сі – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. Сі визначається за формулою:

$$Ci = \frac{Мм \cdot Ki}{Тр \cdot Тзм}, \quad (4.14)$$

де Мм – розмір мінімальної заробітної плати за місяць, грн.; в 2019 році мінімальна заробітна плата становить – 4173 грн., Кі – тарифний коефіцієнт робітника відповідного розряду, Тр – число робочих днів в місяці; приблизно Тр = 21…23 дні; Тзм – тривалість зміни, зазвичай Тзм = 8 годин.

Таблиця 4.9 – Заробітна плата робітників

| Найменування робіт | Трудомісткість, н-год. | Розряд роботи | Погодинна тарифна ставка | Тариф. коеф. | Величина, грн. |
|---|---|---|---|---|---|
| Налагоджувальні | 8 | 5 | 34,4 | 1,45 | 275,2 |
| Складальні | 2 | 3 | 30,1 | 1,27 | 60,2 |
| Всього | | | | | 335,4 |

Додаткова заробітна плата Зд всіх робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10…12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Зд = 0{,}1 \cdot (Зо) = 0{,}1 \cdot (335{,}4) = 33{,}54 \text{ грн.} \quad (4.15)$$

Нарахування на заробітну плату Нзп розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

де Зо – основна заробітна плата розробників, грн.; Зр – основна заробітна плата робітників, грн.; Зд – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

$$Нзп = 0{,}22 \cdot (Зо + Зд) = 0{,}22 \cdot (335{,}4 + 33{,}54) =$$

$$= 81 \text{ грн.} \quad (4.16)$$

«Загальновиробничі витрати» належать витрати: пов'язані з управлінням виробництвом (утримання працівників апарату управління виробництвом, оплата службових відряджень персоналу цехів, витрати на інформаційне забезпечення управління тощо); на повне відновлення та капітальний ремонт основних фондів загальновиробничого призначення; витрати некапітального характеру, пов'язані з удосконаленням технологій та організацією виробництва, поліпшенням якості продукції; на утримання, обслуговування, поточний ремонт виробничих приміщень; на контроль за виробничими процесами та кістю продукції.

Крім того, загальновиробничі витрати з розрахунку на одиницю продукції можна розрахувати за нормативами відносно до основної заробітної плати основних робітників, які виготовляють продукцію:

$$ЗВВ = Нв \cdot Зо, \quad (4.17)$$

Норматив загальновиробничих витрат для програмних продуктів становить 230-270%.

$$ЗВВ = 2{,}3 \cdot 335{,}4 = 771{,}42 \text{ грн},$$

Сума попередніх витрат утворює виробничу собівартість розробки:

$$S_в = 2552 \text{ грн}.$$

## 4.3 Розрахунок мінімальної ціни та чистого прибутку від реалізації розробки методу та засобу захисту від деструктивних інформаційних впливів

Ціна – це грошовий вираз вартості товару (продукції, послуги). Вона завжди коливається навколо ціни виробництва (перетвореної форми вартості

одиниці товару, що дорівнює сумі витрат виробництва й середнього прибутку) та відображає рівень суспільне необхідних витрат праці.

Виходячи з того, що розробки, як правило, приймаються та впроваджуються за завданням замовника, або коли результатом розробки є продукція, що підлягає державному регулюванню, то нижню межу ціни реалізації розробки можна розрахувати за формулою:

$$Ц = S_В \cdot \left(1 + \frac{Р}{100}\right) \cdot \left(1 + \frac{\omega}{100}\right), \quad (4.18)$$

де Sв – виробнича собівартість інноваційного рішення, грн.; Р – норматив рентабельності узгоджений із замовником або встановлений державою, (Р=30…60%); w – ставка податку на додану вартість, % (в 2019 році w=20%).

$$Ц = 2552 \cdot \left(1 + \frac{60}{100}\right) \cdot \left(1 + \frac{20}{100}\right) = 4900 \text{ грн.}$$

Чистий прибуток від реалізації розробки можна розрахувати за формулою:

$$П = \left(Ц - \frac{(Ц - МР) \cdot f}{100} - S_В - \frac{q \cdot S_В}{100}\right) \cdot \left(1 - \frac{h}{100}\right) \cdot РП, \quad (4.19)$$

де Ц – ціна розробки, грн.; МР – вартість матеріальних та інших ресурсів, що були придбані виробником для виготовлення розробки (МР=(0,1…0,2) Цр), грн.; f – зустрічна ставка податку на додану вартість, %; Sв – виробнича собівартість розробки, грн.; q – норматив, який визначає величину адміністративних витрат, витрат на збут та інші операційні витрати, % (рекомендовано q=5…10%); h – ставка податку на прибуток, %, РП – прогнозований попит продажів:

$$П = 18788 \text{ грн.}$$

## 4.4 Розрахунок терміну окупності коштів, вкладених в наукову розробку методу та засобу захисту від деструктивних інформаційних впливів

Термін окупності вкладених у реалізацію наукового проекту інвестицій Ток можна розрахувати за формулою:

$$\text{Ток} = \frac{\text{В}}{\text{П}} = \frac{39928}{18788} = 2{,}1 \text{ роки.} \quad (4.20)$$

Оскільки Ток < 3 років, то фінансування даної наукової розробки методу та засобу захисту від деструктивних інформаційних впливів є доцільним.

Висновки за розділом. Здійснено економічне обґрунтування та доведено доцільність розробки методу та засобу захисту від деструктивних інформаційних впливів. У ході дослідження виявленно, що чистий прибуток сягає 18788 грн, а період окупності становить 2,1 роки.

# ВИСНОВКИ

В даній роботі було вирішено задачі побудови методу та засобу захисту від деструктивних інформаційних впливів. Було досягнено наступних цілей:

– досліджено методи деструктивного впливу та визначено необхідність використання сентиментального аналізу тексту;

– досліджено методи аналізу тональності тексту;

– здійснено огляд проблем які виникають під час аналізу тональності тексту та можливі методи їх вирішення;

– здійснено огляд існуючих засобів для оцінки тональності тексту;

– досліджено основні методи машинного навчання;

– проаналізовано переваги та недоліки основних методів машинного навчання, опираючись на отримані дані було запропоновано новий метод для аналізу тональності тексту, проведено порівняння ефективності запропонованого методу з існуючими та доведено його ефективність;

– на основі розробленого методу був створений програмний засіб який здійснює аналіз тональності тексту з обраної користувачем сторінки веб-сайту або фрагменту тексту.

– здійснено економічне обґрунтування та доведено доцільність його розробки.

На основі виконаної роботи можна зробити висновок, що на сьогоднішній день існує велика кількість засобів для аналізу тональності тексту, але всі вони вузькоспеціалізовані і не можуть застосовуватись для будь якої мови. Для кожної мови потрібно створювати навчальні вибірки і навчати нові моделі для класифікації. Існуючі методи класифікації дають не найкращий результат і потребують постійного покращення, один із можливих методів покращення був успішно створений і реалізований у виконаній роботі.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДЖЕРЕЛА ІНФОРМАЦІЇ, МЕДІАГРАМОТНІСТЬ І РОСІЙСЬКА ПРОПАГАНДА [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://www.kiis.com.ua/materials/pr/20192505_ruspropaganda/DM-KMIS_Report.pdf

2. Дудатьєв А. В., Войтович О. П. Інформаційна безпека соціотехнічних систем: Модель інформаційного впливу / А. В. Дудатьєв, О. П. Войтович // Інформаційні технології та комп'ютерна інженерія. – 2017. – № 38. – С. 16 - 21.

3. Opinion Mining on the Web by Extracting Subject-Aspect-Evaluation Relations [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Nozomi Kobayashi, Ryu Iida, Kentaro Inui, Yuji Matsumoto – 2006 – Режим доступу: https://www.aaai.org/Papers/Symposia/Spring/2006/SS-06-03/SS06-03-018.pdf

4. Vyrva N. Sentiment Analysis in Social Media [Електронний ресурс]: [Веб-сайт]. – Електронні дані. –Режим доступу: https://brage.bibsys.no/xmlui/bitstream/handle/11250/2394497/16-00685-4%20Masteravhandling%20-0sentiment%20analysis%20in%20social%20media%20-%20Natalia%20Vyrva.pdf%20266514_1_1.pdf?sequence=1

5. Liu B. Sentiment analysis and subjectivity // Handbook of natural language processing [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – pp. 627 − 666 14. Horton J. Chapter 11– 2006 – Режим доступу: http://slideplayer.com/slide/6165591/

6. Horton J. Chapter 11: Opinion Mining [Електронний ресурс]: [Веб-сайт]. – Електронні дані. - Режим доступу: http://slideplayer.com/slide/6165591

7. Collomb A., Costea C., Joyeux D., Hasan O., Brunie L. A Study and Comparison of Sentiment Analysis Methods for Reputation Evaluation: [Електронний ресурс]. − (http://liris.cnrs.fr/Documents/Liris-6508.pdf)

8. Пак А. Обучаем компьютер чувствам: [Електронний ресурс]. − (https://habr.com/post/149605/).

9. Rani S., Kumar P. Rule Based Sentiment Analysis System // Elsevier Publications. – 2014. – pp. 1 – 79. 9. Merrell R., Diaz D. Clustering Analyses Methods: Strategies and Algorithms // Reviews in Theoretical Science. – 2016. – V. 4. – no. 2. – pp. 153-158.

10. Saif M. Challenges in Sentiment Analysis //A Practical Guide to Sentiment Analysis. – 2015. – pp. 61 – 83

11. Roth K. SentiMental: [Електронний ресурс]. – (https://github.com/thinkroth/Sentimental).

12. PalitrumLab. Eureka Engine: [Електронний ресурс]. – (http://eurekaengine.ru/ru/).

13. ISPRAS Information Systems. ISPRAS API: [Електронний ресурс]. – (https://api.ispras.ru/).

14. CLiPS Research Center. Pattern: [Електронний ресурс]. – (https://www.clips.uantwerpen.be/pattern).

15. Loria S. TextBlob: Simplified Text Processing: [Електронний ресурс]. – (https://textblob.readthedocs.io/en/dev/).

16. CyberEmotions. SentiStrength: [Електронний ресурс]. – (http://sentistrength.wlv.ac.uk/).

17. YouScan. BrandSpotter – система мониторинга социальных медиа: [Електронний ресурс]. – (https://brandspotter.ru/technologies).

18. Dictum. DataScope: [Електронний ресурс]. – (http://www.dictum.ru/).

19. IBM. IBM Watson: Tone Analyzer: [Електронний ресурс]. – (https://www.ibm.com/watson/services/tone-analyzer/).

20. Go A, Bhayani R, Huang L. Sentiment140: [Електронний ресурс]. – (http://help.sentiment140.com/).

21. SRA, International, Inc. NetOwl Extractor: [Електронний ресурс]. – (https://www.netowl.com/sentiment-analysis/). Проверено 7.05.2018. 30.

22. Converseon, Inc. ConveyAPI: [Електронний ресурс]. – (https://converseon.com/solutions/).

23. Repustate Inc. Repustate: [Електронний ресурс]. – (https://www.repustate.com/).

24. Нормализация и лемматизация текста с использованием тезауруса: [Електронний ресурс]. – (http://www.solarix.ru/for_developers/docs/text-normalization.shtml)

25. Bergsma S., Jung D., Lau R., Wang Y. Machine Learning Approaches to Sentiment Classification. CMPUT 551: Course Project. – Edmonton: University of Alberta, 2005. – 35 p

26. Rong X. Word2vec Parameter Learning Explained: [Електронний ресурс]. – (https://arxiv.org/pdf/1411.2738.pdf)

27. Pang B., Lee L., Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques // Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). – 2002. – pp. 79 – 86.

28. Sunasra M. Performance Metrics for Classification problems in Machine Learning- Part I: [Електронний ресурс]. – (https://medium.com/greyatom/performance-metrics-for-classification-problems-inmachine-learning-part-i-b085d432082b).

29. Баженов Д. Наивный байесовский классификатор: [Електронний ресурс]. – (http://bazhenov.me/blog/2012/06/11/naive-bayes).

30. Patel Savan. Chapter 6: Adaboost Classifier: [Електронний ресурс]. – (https://medium.com/machine-learning-101/https-medium-com-savanpatel-chapter-6-adaboost-classifier-b945f330af06)

31. Brannick M. Logistic Regression: [Електронний ресурс]. – (http://faculty.cas.usf.edu/mbrannick/regression/Logistic.html)

32. Акобир Ш. Деревья решений – общие принципы работы: [Електронний ресурс]. – (https://basegroup.ru/community/articles/description).

33. Bambrick N. Support Vector Machines: A Simple Explanation: [Електронний ресурс]. – (https://www.kdnuggets.com/2016/07/support-vectormachines-simple-explanation.html).

# ДОДАТКИ

Додаток А

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра захисту інформації

ЗАТВЕРДЖУЮ

Зав. Кафедри ЗІ, д. т. н., проф

_____ В. А.Лужецький

_____ 2019 р.

**ТЕХНІЧНЕ ЗАВДАННЯ**

на магістерську кваліфікаційну роботу

«Метод та засіб захисту від деструктивних інформаційних впливів»

08-20.МКР 012.00.000 ТЗ

Керівник магістерської кваліфікаційної роботи

к. т. н. доц. кафедри ЗІ

_____Дудатьєв А. В.

«___»_____2019 р.

Вінниця 2019

## 1 Назва та область використання

Метод та засіб захисту від деструктивних інформаційних впливів. Програмний засіб використовується для аналізу тональності тексту, що допомагає визначити імовірний деструктивний вплив. Програмний засіб може використовуватись як для домашнього користування, так і для виявлення деструктивних впливів спеціалізованими службами.

## 2 Основа для розробки

Робота проводиться на підставі наказу ректора ВНТУ № 254 від 02.10.2019 р.

## 3 Мета та призначення розробки

Покращення існуючого методу класифікації тональності тексту та створення програмного засобу з розробленим методом класифікації імовірних деструктивних впливів у текстах для їхнього подальшого аналізу.

## 4 Джерела розробки

4.1 ДЖЕРЕЛА ІНФОРМАЦІЇ, МЕДІАГРАМОТНІСТЬ І РОСІЙСЬКА ПРОПАГАНДА [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://www.kiis.com.ua/materials/pr/20192505_ruspropaganda/DM-KMIS_Report.pdf

4.2 Баженов Д. Наивный байесовский классификатор: [Електронний ресурс]. − (http://bazhenov.me/blog/2012/06/11/naive-bayes).

4.3 Opinion Mining on the Web by Extracting Subject-Aspect-Evaluation Relations [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Nozomi Kobayashi, Ryu Iida, Kentaro Inui, Yuji Matsumoto – 2006 – Режим доступу: https://www.aaai.org/Papers/Symposia/Spring/2006/SS-06-03/SS06-03-018.pdf

4.4 Vyrva N. Sentiment Analysis in Social Media [Електронний ресурс]: [Веб-сайт]. – Електронні дані. –Режим доступу: https://brage.bibsys.no/xmlui/bitstream/handle/11250/2394497/16-00685 4%20Masteravhandling%20-

0sentiment%20analysis%20in%20social%20media%20-%20Natalia%20Vyrva.pdf%20266514_1_1.pdf?sequence=1

4.5 Liu B. Sentiment analysis and subjectivity // Handbook of natural language processing [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – pp. 627 − 666 14. Horton J. Chapter 11– 2006 – Режим доступу: http://slideplayer.com/slide/6165591/

## 5 Вимоги до програмного засобу

5.1 Параметри розроблюваного програмного засобу:

– операційна система - Windows;
– мова програмування – Python, javascript;
– середовище розробки – visual studio code;
– підключення до мережі Інтернет.

5.2 Програмний засіб повинен виконувати такі дії:

– відображати тональність тесту;
– аналізувати сторінку повністю;
– аналізувати фрагмент тексту з сторінки.

## 6 Вимоги до супровідної документації

Графічна і текстова документація повинна відповідати діючим стандартам України.

## 7 Стадії та етапи розробки

| № з/п | Назва етапів магістерської кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Аналіз завдання. Вступ | 01.09.2019 – 04.09.2019 | Чорновик вступу |
| 2 | Розробка технічного завдання | 23.09.2019 – 29.09.2019 | Проект технічного завдання |
| 3 | Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи | 05.09.2019 – 15.09.2019 | Чорновик першого розділу. Схеми та алгоритми |
| 4 | Практична реалізація, моделювання, експериментування, результати | 14.10.2019 – 10.11.2019 | Програмний засіб. Розділ пояснювальної записки |
| 5 | Аналіз виконання ТЗ, висновки | 18.11.2019 – 24.11.2019 | Висновки, інструкції |
| 6 | Оформлення пояснювальної записки | 25.11.2019 – 30.11.2019 | Пояснювальна записка, графічний матеріал |

## 8 Порядок контролю та прийому

До прийому магістерської кваліфікаційної роботи подається:

- заключний звіт (пояснювальна записка);
- ілюстративний матеріал;
- програмний засіб.

Початок розробки 01.09.2019
Крайній термін виконання
магістерської кваліфікаційної роботи 14.12.2019

Розробив студент групи 1БС-18м_____Олійник Є. А.

# Додаток Б
# Лістинг програмного засобу

## Background.js

```
let resOverall;
let resSelection;
let selection;
let requestNumber = 0;
let collectedScanData = [];

const storage = chrome.storage.sync;

chrome.runtime.onInstalled.addListener((details) => {
    // first time install
    if (details.reason == "install") {
        storage.set({DEFAULT_AFINN_WEIGHT: 0.4});
        storage.set({DEFAULT_SENTIC_WEIGHT: 0.6});
        storage.set({AFINN_WEIGHT: 0.4});
        storage.set({SENTIC_WEIGHT: 0.6});
        storage.set({SHOW_SENTIMENT_FOR_PAGES: false});
        storage.set({SHOW_SENTIMENT_FOR_SELECTION: false});
        storage.set({SHOW_SENTIMENT_ON_ICON: true});
        storage.set({COLOR_PAGES: false});
        storage.set({COLOR_SELECTION: true});
    } else {
    }
    let id = chrome.contextMenus.create({
        "title": "Get sentiment of selection",
        "contexts": ["selection"],
        "id": "context" + "selection"
    });
});

let getSentimentAverage = (data, type, callback) => {
    let afinnData = analyzeTextSentimentAFINN111(data, type);
    let senticData = analyzeTextSentimentSenticNet5(data, type);
    let res = senticData;
    afinnData.sentiment = mapValueToRange(afinnData.sentiment, -0.7, 0.7, -
100, 100);

    let afinnWt, senticWt;
    // get weights from storage
    storage.get(['AFINN_WEIGHT'], (result) => {
        afinnWt = result.AFINN_WEIGHT;
        storage.get(['SENTIC_WEIGHT'], (result) => {
            senticWt = result.SENTIC_WEIGHT;
```

```
            let weightedMean = (afinnData.sentiment * afinnWt + senticData.sentimen
tMapped * senticWt) / (1);
            res.sentimentMapped = Math.round(weightedMean);
            res.descriptorSentiment = getValueDescriptor(weightedMean, "sentiment")
            res.wordsPositive = res.wordsPositive.concat(afinnData.wordsPositive);
            res.wordsNegative = res.wordsNegative.concat(afinnData.wordsNegative);
            res.wordsMostPositive = res.wordsMostPositive.concat(afinnData.wordsMos
tPositive);
            res.wordsMostNegative = res.wordsMostNegative.concat(afinnData.wordsMos
tNegative);
            res.OGafinn = afinnData.sentiment;
            res.OGsentic = senticData.sentimentMapped;
            if (callback != null) callback(res);
        });
    });
    return res;
};

let handleSelectedTextHelper = () => {
    // clean the text of specials, convert to lower, and split into words
    let cleaned = selection.toLowerCase().replace(/\n/g, ' ').replace(/[^\w\s-
]/g, '').split(' ');
    // analyze
    resSelection = getSentimentAverage(cleaned, "rightClick", (res) => {
        chrome.extension.onConnect.addListener((port) => {
            port.postMessage({data: res, title: selection});
        })
    storage.get(['COLOR_SELECTION'], (result) => {
        if (result.COLOR_SELECTION) {
            chrome.tabs.query({active: true, currentWindow: true},(tabs) => {
                chrome.tabs.sendMessage(tabs[0].id, {type: "COLOR-
PH", data: res, content: selection}, null);
            });
        }
    });
    storage.get(['SHOW_SENTIMENT_FOR_SELECTION'], (result) => {
        if (result.SHOW_SENTIMENT_FOR_SELECTION) {
            chrome.tabs.query({active: true, currentWindow: true},(tabs) => {
                chrome.tabs.sendMessage(tabs[0].id, {type: "COLOR-WD-
PH", data: res, content: selection}, null);
            });
        }
    });
});
};

let handleSelectedText = (info) => {
    if (info == null) {
        chrome.tabs.query({active: true, currentWindow: true},(tabs) => {
```

```
            chrome.tabs.sendMessage(tabs[0].id, {type: "GET-
SELECTION"}, (response) =>{
                selection = response.selection;
                if (selection != null && selection != undefined){
                    // check for null just in case chrome allows you to use key com
mands without selected text
                    handleSelectedTextHelper();
                }

            });
        } else {
            selection = info.selectionText;
            handleSelectedTextHelper();
        }
};

chrome.commands.onCommand.addListener((command) => {
    if (command == "analyze-selected-text") {
        handleSelectedText(null);
    }
});

chrome.contextMenus.onClicked.addListener((info, tab) => {
    handleSelectedText(info);
});
chrome.runtime.onMessage.addListener((request, sender, sendResponse) => {

    if (request.type == "pageScan") {
        // message contains a page scan
        resOverall = getSentimentAverage(request.pageContents, "pageScan", (res) =>
 {

            storage.get(['SHOW_SENTIMENT_ON_ICON'], (result) => {
                if (result.SHOW_SENTIMENT_ON_ICON) {
                    chrome.browserAction.setBadgeText({text: String(res.sentimentMa
pped)});
                    let rgbColor = HSLToRGB(res.sentimentColor, 1, 0.4);
                    // change the background color of the number
                    chrome.browserAction.setBadgeBackgroundColor({color: [rgbColor[
0], rgbColor[1], rgbColor[2], 255] });
                }
            });

            // send result to popup
            chrome.extension.onConnect.addListener((port) => {
                port.postMessage({data: res, title: request.pageName});
            })
            // check if the user wants to color the page's individual sentiments
```

```
            storage.get(['SHOW_SENTIMENT_FOR_PAGES'], (result) => {
                if (result.SHOW_SENTIMENT_FOR_PAGES) {
                    // send result to content script
                    chrome.tabs.query({active: true, currentWindow: true},(tabs) =>
 {
                        chrome.tabs.sendMessage(tabs[0].id, {type: "COLOR-WD-
PG", data: res}, null);
                    });
                }
            });
            // check if the user wants to color the all the page's paragraphs
            storage.get(['COLOR_PAGES'], (result) => {
                if (result.COLOR_PAGES) {
                    chrome.tabs.query({active: true, currentWindow: true},(tabs) =>
 {
                        chrome.tabs.sendMessage(tabs[0].id, {type: "COLOR-
PG", data: res}, null);
                    });
                }
            });

        });
    }
    else if (request.type == "selectionClick") {
        resSelection = getSentimentAverage(request.clickContents, "rightClick", (re
s) => {
            chrome.extension.onConnect.addListener((port) => {
                port.postMessage({data: res, title: request.title});
            });
        storage.get(['COLOR_SELECTION'], (result) => {
            if (result.COLOR_SELECTION) {
                chrome.tabs.query({active: true, currentWindow: true},(tabs) => {
                    chrome.tabs.sendMessage(tabs[0].id, {type: "COLOR-
PH", data: res, content: request.title}, null);
                });
            }
        });
        storage.get(['SHOW_SENTIMENT_FOR_SELECTION'], (result) => {
            if (result.SHOW_SENTIMENT_FOR_SELECTION) {
                chrome.tabs.query({active: true, currentWindow: true},(tabs) => {
                    chrome.tabs.sendMessage(tabs[0].id, {type: "COLOR-WD-
PH", data: res, content: request.title}, null);
                });
            }
        });
    });
    }
    else if (request.type == "autoParagraphScan" || request.type == "autoParagraphS
canData") {
```

```
        // message contains an auto paragraph scan or scan data
        if (request.type == "autoParagraphScanData") {
            // get the number of incoming scan requests to expect
            requestNumber = request.requests;
        } else {
            // we got a scan request, calculate and put in finished array
            collectedScanData.push(getSentimentAverage(request.content, "rightClick
", null));
            // if we've scanned all requests,
            if (collectedScanData.length == requestNumber) {
                // send the arrray of finished scans to the content script
                chrome.tabs.query({active: true, currentWindow: true}, (tabs) => {
                    chrome.tabs.sendMessage(tabs[0].id, {type: "COLOR-PG-
DATA", data: collectedScanData}, null);
                    // clear the scanned data array
                    collectedScanData = [];
                });
            }
        }
    }
});

chrome.extension.onConnect.addListener((port) => {
    port.onMessage.addListener((msg) => {
        if (msg == "select_paragraphs") {
            // send message to content script to make all paragraphs clickable
            chrome.tabs.query({active: true, currentWindow: true},(tabs) => {
                chrome.tabs.sendMessage(tabs[0].id, {type: "SELECT-PH"}, null);
            });
            // force the popup window to close
            port.postMessage("CLOSE_POPUP");
        }
        else if (msg == "analyze_page") {
            // reload the current page to tigger sentiment analysis
            chrome.tabs.getSelected(null, (tab) => {
                let code = 'window.location.reload();';
                chrome.tabs.executeScript(tab.id, {code: code});
            });
            // force the popup window to close
                port.postMessage("CLOSE_POPUP");
            }
        });
})


function analyzeTextSentimentAFINN111(text, type) {
    // variables for processing text
    let index = text.length,
    score = 0, positive = [], negative = [],
```

```
mostPos = [], mostNeg = [];
// loop through words in the text
while (index--) {
    // get word and its sentiment value from the dictionary
    let obj = text[index], objSentimentVal = afinnData[obj];
    // if there is a dictionary value,
    if (objSentimentVal != null) {
        if (index > 0) {
            // need to negate?
            if (negs[text[index - 1]]) objSentimentVal = -objSentimentVal;
        }
        // add the word to either positive or negative array
        if (objSentimentVal > 0) positive.push(obj);
        else if (objSentimentVal < 0) negative.push(obj);
        // if word is extreme, add it to mostPos or mostNeg array
        if (objSentimentVal <= -4 ) mostNeg.push(obj);
        else if (objSentimentVal >= 4) mostPos.push(obj);
        // add the sentiment to the overall score
        score += objSentimentVal;
    }
}
// return JSON object with data
return {
    scanType: type,
    sentiment: text.length > 0 ? score / text.length : 0,
        wordsPositive: positive,
        wordsNegative: negative,
        wordsMostPostitive: mostPos,
        wordsMostNegative: mostNeg,
        words: text
    };
}


function analyzeTextSentimentSenticNet5(text, type) {
    // variables for processing text
    let polarityScore = 0, attentionScore = 0, pleasantnessScore = 0,
    positive = [''], negative = [''], mostPos = [''], mostNeg = [''],
    exciting = [], dry = [], pleasant = [], unpleasant = [],
    unknown = [], ordered = [];

    // loop through words in the text
    for (let index = 0; index < text.length; index++) {
        // get word and its sentiment value from the dictionary
        let obj = text[index], objSentimentVal = null;

        // make all possible word combinations and look up in senticnet5 dictionary
        // if no combinations possible, singularize the word and try again
        (function () {
```

```
            // make word combinations depending on current index (so we don't
            // have an out of bounds exception on the last indices)
            let chooseCombination = () => {
                if (index <= text.length -
4) objSentimentVal = makeWordCombinations(4, index, text);
                else if (index <= text.length -
3) objSentimentVal = makeWordCombinations(3, index, text);
                else if (index <= text.length -
2) objSentimentVal = makeWordCombinations(2, index, text);
                else objSentimentVal = makeWordCombinations(1, index, text);
            }
            chooseCombination();

            if (objSentimentVal == undefined || objSentimentVal == null) {
                text[index] = singularize(obj);
                chooseCombination();

                if (objSentimentVal != null && objSentimentVal != undefined) {
               }
               else {

                    text[index] = stemmer(obj);
                    // retry making word combinations
                    chooseCombination();

                        if (objSentimentVal != null && objSentimentVal != undefined) {

                        }
                    }
                    text[index] = obj;
                }
            }());
            if (obj == "of" || obj == "the" || obj == "to") objSentimentVal = null;

            ordered.push(objSentimentVal);

            if (objSentimentVal !== null && objSentimentVal !== undefined) {

                let polarity_value = parseFloat(objSentimentVal.val[2]),
                attention_value = parseFloat(objSentimentVal.val[1]),
                pleasantness_value = parseFloat(objSentimentVal.val[0]);
                if (index > 0) {
                    // if the previous word is in the list of negators, negate the curr
ent word
                    if (negs[text[index - 1]]) {
                        polarity_value = -polarity_value;
                        attention_value = -attention_value;
                        pleasantness_value = -pleasantness_value;
                    }
```

```
}
// add the word to either positive or negative array
if (polarity_value > 0) positive.push(obj);
else if (polarity_value < 0) negative.push(obj);
// if word is extreme, add it to mostPos or mostNeg array
if (polarity_value <= -0.8 ) mostNeg.push(obj);
else if (polarity_value >= 0.8) mostPos.push(obj);
// add the word to either exciting or dry array
if (attention_value > 0) exciting.push(obj);
else if (attention_value < 0) dry.push(obj);
                // add the word to either pleasant or unpleasant array
                if (pleasantness_value > 0) pleasant.push(obj);
                else if (pleasantness_value < 0) unpleasant.push(obj);
                // add the sentiment to the overall score
                polarityScore += polarity_value;
                attentionScore += attention_value;
                pleasantnessScore += pleasantness_value;
            }
            else {
                // don't know the word
                unknown.push(obj);
            }
        }
        // calculate the sentiment values from the scores, scale them, and round to nea
rest integer
        let sentimentRes = Math.round(mapValueToRange(text.length > 0 ? polarityScore /
 text.length : 0, -0.65, 0.75, -100, 100));
        let attentionRes = Math.round(mapValueToRange(text.length > 0 ? attentionScore
/ text.length : 0, -0.65, 0.75, -100, 100));
        let pleasantnessRes = Math.round(mapValueToRange(text.length > 0 ? pleasantness
Score / text.length : 0, -0.65, 0.75, -100, 100));
        // return JSON object with data
        return {
            scanType: type,
            sentiment: sentimentRes,
            sentimentMapped: sentimentRes,
            attentionMapped: attentionRes,
            pleasantnessMapped: pleasantnessRes,
            descriptorSentiment: getValueDescriptor(sentimentRes, "sentiment"),
            descriptorAttention: getValueDescriptor(attentionRes, "attention"),
            descriptorPleasantness: getValueDescriptor(pleasantnessRes, "pleasantness")
,
            sentimentColor: mapValueToColor(text.length > 0 ? polarityScore / text.leng
th : 0, -0.65, 0.75),
            attentionColor: mapValueToColor(text.length > 0 ? attentionScore / text.len
gth : 0, -0.65, 0.75),
            pleasantnessColor: mapValueToColor(text.length > 0 ? pleasantnessScore / te
xt.length : 0, -0.65, 0.75),
            wordsPositive: positive,
```

```
        wordsNegative: negative,
        wordsMostPositive: mostPos,
        wordsMostNegative: mostNeg,
        wordsPleasant: pleasant,
        wordsUnpleasant: unpleasant,
        wordsExciting: exciting,
        wordsDry: dry,
        wordsUnknown: unknown,
        wordsOrdered: ordered,
        words: text
    };
}

function makeWordCombinations(numberOfWords, index, text) {
    // loop through all possible combinations of words
    for (let i = numberOfWords - 1; i >= 0; i--) {
        // return the largest possible combination of words
        let res = null;
        let data = null;
        switch(i) {
            case 3:
            res = [text[index], text[index + 1], text[index + 2], text[index + 3]].
join("_");
            data = sentic5Data[res];
            if (data != null) return {word: res, val: data, color: mapValueToColor(
data[2], -1, 1)}; break;
            case 2:
            res = [text[index], text[index + 1], text[index + 2]].join("_");
            data = sentic5Data[res];
            if (data != null) return {word: res, val: data, color: mapValueToColor(
data[2], -1, 1)}; break;
            case 1:
            res = [text[index], text[index + 1]].join("_");
            data = sentic5Data[res];
            if (data != null) return {word: res, val: data, color: mapValueToColor(
data[2], -1, 1)}; break;
            case 0:
            res = text[index];
            data = sentic5Data[res];
            if (data != null) return {word: res, val: data, color: mapValueToColor(
data[2], -1, 1)}; break;
        }
    }
    return null;
}

function getValueDescriptor(value, scoreType) {

    let outcomes = [];
```

```
// select the correct array based off the provided scoreType
if (scoreType == "sentiment") outcomes = sentimentDescriptors;
else if (scoreType == "attention") outcomes = attentionDescriptors;
else if (scoreType == "pleasantness") outcomes = pleasantnessDescriptors;
else {}// scoreType was not valid
// return string based on value
if (value >= 70) return outcomes[6];
else if (value >= 30) return outcomes[5];
else if (value >= 10) return outcomes[4];
else if (value <= -70) return outcomes[0];
    else if (value <= -30) return outcomes[1];
    else if (value <= -10) return outcomes[2];
    else return outcomes[3];
}

/**
*    map a sentiment value to a specific color (HSL color space)
*    @param {float} value
*    @param {float} rangeStart, rangeEnd - current range value is in-
 rangeStart <= value <= rangeEnd
*    @return {float} - HSL hue value in [0, 120]
*/
function mapValueToColor(value, rangeStart, rangeEnd) {
    // change value to map between 0 and 1
    value = mapValueToRange(value, rangeStart, rangeEnd, 0, 1);
    // max red and green saturations from HSV
    let maxRed = 0, maxGreen = 120;
    // interpolate between red and green with value
    return value * maxGreen + (1 - value) * maxRed;
}

function mapValueToRange(value, startOld, endOld, startNew, endNew) {
    let res = startNew + ((endNew - startNew) / (endOld - startOld)) * (value -
 startOld);
    if (res > endNew) return endNew;
    else if (res < startNew) return startNew;
    return res;
}
function pluralize(inputWord) {
    // if singular and plural are the same, return
    if (uncountable.indexOf(inputWord.toLowerCase()) >= 0) return inputWord;
    // check if word is irregular, if so, replace with regex and return
    for (word in irregular){
        let pattern = new RegExp(word+'$', 'i');
        if (pattern.test(inputWord)) return inputWord.replace(pattern, irregular[wo
rd]);
    }
    // check if word is regular, if so, replace with regex and return
    for(reg in plural){
```

```
        var pattern = new RegExp(reg, 'i');
        if (pattern.test(inputWord)) return inputWord.replace(pattern, plural[reg])
;
    }
    return inputWord;
}

function singularize(inputWord) {
    if (uncountable.indexOf(inputWord.toLowerCase()) >= 0)
      return inputWord;
    for (word in irregular) {
        let pattern = new RegExp(irregular[word] + '$', 'i');
        if (pattern.test(inputWord)) return inputWord.replace(pattern, word);
    }
    for (reg in singular) {
        let pattern = new RegExp(reg, 'i');
        if (pattern.test(inputWord)) return inputWord.replace(pattern, singular[reg
]);
    }
    return inputWord;
}
function HSLToRGB(hue, saturation, lightness) {

  if (hue == undefined ){
    return [0, 0, 0];
}
let chroma = (1 - Math.abs((2 * lightness) - 1)) * saturation;
let huePrime = Math.floor(hue / 60);
let secondComponent = chroma * (1 - Math.abs(((hue / 60) % 2) - 1));
let red, green, blue;

if (huePrime === 0 ){
    red = chroma;
    green = secondComponent;
    blue = 0;
} else if (huePrime === 1 ){
    red = secondComponent;
    green = chroma;
    blue = 0;
} else if (huePrime === 2 ){
    red = 0;
    green = chroma;
    blue = secondComponent;
} else if (huePrime === 3 ){
    red = 0;
    green = secondComponent;
    blue = chroma;
} else if (huePrime === 4 ){
    red = secondComponent;
```

```
    green = 0;
    blue = chroma;
} else if (huePrime === 5 ){
    red = chroma;
    green = 0;
    blue = secondComponent;
}

let lightnessAdjustment = lightness - (chroma / 2);
red += lightnessAdjustment;
    green += lightnessAdjustment;
    blue += lightnessAdjustment;

    return [Math.round(red * 255), Math.round(green * 255), Math.round(blue * 255)]
;
}
```

### Service.py

```
import keras.models
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

import pandas as pd
import numpy as np
import re

def get_sentiment(record):
    fullData = pd.read_csv("./data/example.csv")

    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(fullData["text"])

    X = tokenizer.texts_to_sequences(fullData['text'].values)
    X = pad_sequences(X)

    json_file = open("./data/topModel.json", "r")
    loaded_model_json = json_file.read()
    json_file.close()
    loaded_model = keras.models.model_from_json(loaded_model_json)
    loaded_model.load_weights("./data/topModel.h5")
    loaded_model.compile(loss="binary_crossentropy", optimizer="adam", metrics=["ac
curacy"])
    if(np.argmax(sentiment) == 0):
       return 'negative'
    elif(np.argmax(sentiment) == 1):
        return 'positive'
```

### porter-stemmer.js

```
import re


class UkrainianStemmer():
    def __init__(self, word):
        self.word = word
        self.vowel = r'аеиоуюяіїє'  # http://uk.wikipedia.org/wiki/Голосний_звук
        self.perfectiveground = r'(ив|ивши|ившись|ыв|ывши|ывшись((?<=[ая])(в|вши|вш
ись)))$'
        # http://uk.wikipedia.org/wiki/Рефлексивне_дієслово
        self.reflexive = r'(с[яьи])$'
        # http://uk.wikipedia.org/wiki/Прикметник + http://wapedia.mobi/uk/Прикметн
ик
        self.adjective = r'(ими|ій|ий|а|е|ова|ове|ів|є|їй|єе|еє|я|ім|ем|им|ім|их|іх
|ою|йми|іми|у|ю|ого|ому|ої)$'
        # http://uk.wikipedia.org/wiki/Дієприкметник
        self.participle = r'(ий|ого|ому|им|ім|а|ій|у|ою|ій|і|их|йми|их)$'
        # http://uk.wikipedia.org/wiki/Дієслово
        self.verb = r'(сь|ся|ив|ать|ять|у|ю|ав|али|учи|ячи|вши|ши|е|ме|ати|яти|є)$'
        # http://uk.wikipedia.org/wiki/Іменник
        self.noun = r'(а|ев|ов|е|ями|ами|еи|и|ей|ой|ий|й|иям|ям|ием|ем|ам|ом|о|у|ах
|иях|ях|ы|ь|ию|ью|ю|ия|ья|я|і|ові|ї|ею|єю|ою|є|еві|ем|єм|ів|їв|ю)$'
        self.rvre = r'[аеиоуюяіїє]'
        self.derivational = r'[^аеиоуюяіїє][аеиоуюяіїє]+[^аеиоуюяіїє]+[аеиоуюяіїє].
*(?<=о)сть?$'
        self.RV = ''

    def ukstemmer_search_preprocess(self, word):
        word = word.lower()
        word = word.replace("'", "")
        word = word.replace("ё", "е")
        word = word.replace("ъ", "ї")
        return word

    def s(self, st, reg, to):
        orig = st
        self.RV = re.sub(reg, to, st)
        return (orig != self.RV)

    def stem_word(self):
word = self.ukstemmer_search_preprocess(self.word)
if not re.search('[аеиоуюяіїє]', word):
    stem = word
else:
    p = re.search(self.rvre, word)
    start = word[0:p.span()[1]]
    self.RV = word[p.span()[1]:]

    # Step 1
```

```
if not self.s(self.RV, self.perfectiveground, ''):

    self.s(self.RV, self.reflexive, '')
    if self.s(self.RV, self.adjective, ''):
        self.s(self.RV, self.participle, '')
    else:
        if not self.s(self.RV, self.verb, ''):
            self.s(self.RV, self.noun, '')
# Step 2
self.s(self.RV, 'и$', '')

            # Step 3
            if re.search(self.derivational, self.RV):
                self.s(self.RV, 'ость$', '')

            # Step 4
            if self.s(self.RV, 'ь$', ''):
                self.s(self.RV, 'ейше?$', '')
                self.s(self.RV, 'нн$', u'н')

            stem = start + self.RV
        return stem


if __name__ == '__main__':
    stemObj = UkrainianStemmer('Рефлексивного')
    print(stemObj.stem_word())
    stemObj = UkrainianStemmer('Тямущий')
    print(stemObj.stem_word())
```

### content.js

```
let getPageContents = () => {
    let elementsContent = [], words = [];
    Array.from(document.getElementsByTagName("p")).forEach((element) => {
        let content = String(element.innerText)
        let style = window.getComputedStyle(element);
        if (!bannedWords.includes(content) && style.display !== 'none' && element.o
ffsetHeight > 0) elementsContent.push(content);
    });
    elementsContent.forEach((element) => {
        words = words.concat(element.toLowerCase().replace(/\n/g, ' ').replace(/[^\
w\s]/g, '').split(' '));
    });
    return words;
}
let bannedWords = ["Advertisement", "ADVERTISEMENT", "AD", "ad", "ADVERTISING", "Ad
vertising", "Advert", "ADVERT"];
// get the page's text content
let res = getPageContents();
```

```
// send text content to background script
chrome.runtime.sendMessage({pageContents: res, type: "pageScan", pageName: document
.title}, null);

let resData = [];
let elements = 0;
let elementsScanned = 0;

chrome.extension.onMessage.addListener((msg, sender, sendResponse) => {
    let elementsContent = [], words = [];

    if (msg.type == "SELECT-PH") {
        // function to make a page's paragraphs clickable to scan for sentiment
        let loadParagraphs = () => {
            Array.from(document.getElementsByTagName("p")).forEach((element) => {
                if (element.classList.contains("sentitude-scanned")) {
                    let newElement = element.cloneNode(true);
                    newElement.style.cursor = "default";
                    newElement.title = "";
                    element.parentNode.replaceChild(newElement, element);
                } else {
                    element.style.cursor = "pointer";
                    element.title = "Click to analyze sentiment";
                    element.addEventListener('click', () => {
                        element.classList.add("sentitude-scanned");
                        // get clicked element's content and clean
                        let res = element.innerText.toLowerCase().replace(/\n/g, '
').replace(/[^\w\s-]/g, '').split(' ');
                        // send contents to background script
                        chrome.runtime.sendMessage({clickContents: res, type: "sele
ctionClick", title: element.innerText}, null);
                        let newElement = element.cloneNode(true);
                        newElement.style.cursor = "default";
                        newElement.title = "";
                        element.parentNode.replaceChild(newElement, element);
                    });
                }
            });
        }
        loadParagraphs();
        // infinitely reload the paragraphs to accomodate for dynamically-
loaded content
        setInterval(() => {loadParagraphs()}, 5000);
    }
    else if (msg.type == "COLOR-PH") {
        // style parent paragraph with border and background color according
        // to sentiment value
        let elt = window.getSelection().anchorNode.parentElement;
```

```
        if (elt.tagName == undefined) {
            Array.from(document.getElementsByClassName("sentitude-
scanned")).forEach((element) => {
                if (element.innerText == msg.content) {
                    elt = element;
                }
            });
        }
        if (elt.tagName.toLowerCase() != "p") {
            elt = elt.parentElement;
        }
        elt.classList.add("sentitude-scanned");
        elt.style.border = "1px solid " + "hsl(" + msg.data.sentimentColor + ", 100
%, 50%)";
        elt.style.borderRadius = "4px";
        elt.style.backgroundColor = "hsla(" + msg.data.sentimentColor + ", 100%, 50
%, 0.3)";
        elt.classList.add("sentitude-tooltip");
        let spanWithResults = document.createElement("SPAN");
        spanWithResults.classList.add("sentitude-tooltiptext");
        spanWithResults.innerHTML = "Sentiment: " + msg.data.descriptorSentiment +
"</br>" +
        "Pleasantness: " + msg.data.descriptorPleasantness + "</br>" +
        "Attention Value: " + msg.data.descriptorAttention;

        elt.appendChild(spanWithResults);
        if (window.getSelection) window.getSelection().removeAllRanges();
        else if (document.selection) document.selection.empty();
    }
    else if (msg.type == "COLOR-WD-PG") {
        let position = 0;
        Array.from(document.getElementsByTagName("p")).forEach((element) => {
            let content = String(element.innerText);
            let contentSplit = [], wordsInCombination = 0;
            // style of element
            let style = window.getComputedStyle(element);
            if (!bannedWords.includes(content) && style.display !== 'none' && eleme
nt.offsetHeight > 0) {
                contentSplit = contentSplit.concat(content.replace(/\n/g, ' ').spli
t(' '));

                contentSplit.forEach((word, i) => {
                    // if we don't need to skip a word
                    if (wordsInCombination <= 0){
                        // if the word was matched in the scanned data
                        if (msg.data.wordsOrdered[position] !== null) {
                            // get number of words in combination-
 words have been joined with '_'
```

```
                                wordsInCombination = (msg.data.wordsOrdered[position].w
ord.match(/_/g)||[]).length;
ation,
elow
                                let fullString = word;
                                for (let j = i + 1; j <= i + wordsInCombination; j++) {
                                    fullString += (" " + contentSplit[j]);
                                    contentSplit[j] = "";
                                }
                                contentSplit[i] = "<span style=\"background-color: " +
                                                  "hsla(" + msg.data.wordsOrdered[pos
ition].color +
                                                  ", 100%, 50%, 0.3); border-
radius: 5px;\" title=\"Matched: \'" +
                                                  msg.data.wordsOrdered[position].wor
d +
                                                  "\'&#13;Sentiment: " + Math.round(m
sg.data.wordsOrdered[position].val[2] * 100) +
                                                  "\">" + fullString + "</span>";
                            }
                            // if the word wasn't matched, leave it alone
                        } else {
                            // we needed to skip a word, decrement
                            wordsInCombination--;
                        }
                        // move to next scanned word
                        position++;
                    });
                    contentSplit = contentSplit.join(' ');
                    element.innerHTML = contentSplit;
                }
            });
        }
        else if (msg.type == "COLOR-WD-PH") {
            let element, content, oldContent;

            if (window.getSelection().anchorNode == null) {
                Array.from(document.getElementsByClassName("sentitude-
tooltiptext")).forEach((elt) => {
                    oldContent = elt.innerHTML;
                    let words = elt.parentNode.innerText.replace(elt.innerText, "");
                    if (msg.content == words){
                        element = elt.parentNode;
                        content = words;
                    }
                });
            } else {
                Array.from(document.getElementsByTagName("p")).forEach((elt) => {
                    // if the selection matches the paragraph,
```

```
        if (window.getSelection() == elt.innerText){
            element = elt;
            content = elt.innerText;
        }
    });
}

let position = 0;
// if the selection is a whole paragraph,
if (element != null) {
            let contentSplit = [], wordsInCombination = 0;
            // style of element
            let style = window.getComputedStyle(element);
            // ensure the element is visible and not and advertisement
            if (!bannedWords.includes(content) && style.display !== 'none' && eleme
nt.offsetHeight > 0) {
                // split into individual words
                contentSplit = contentSplit.concat(content.replace(/\n/g, ' ').spli
t(' '));

                // loop through split words
                contentSplit.forEach((word, i) => {
                    // if we don't need to skip a word
                    if (wordsInCombination <= 0){
                        // if the word was matched in the scanned data
                        if (msg.data.wordsOrdered[position] !== null) {
 words have been joined with '_'
                            wordsInCombination = (msg.data.wordsOrdered[position].w
ord.match(/_/g)||[]).length;
ation,
elow
                            let fullString = word;
                            for (let j = i + 1; j <= i + wordsInCombination; j++) {
                                fullString += (" " + contentSplit[j]);
                                contentSplit[j] = "";
                            }
                            contentSplit[i] = "<span style=\"background-color: " +
                                              "hsla(" + msg.data.wordsOrdered[pos
ition].color +
                                              ", 100%, 50%, 0.3); border-
radius: 5px;\" title=\"Matched: \'" +
                                              msg.data.wordsOrdered[position].wor
d +
                                              "\'&#13;Sentiment: " + Math.round(m
sg.data.wordsOrdered[position].val[2] * 100) +
                                              "\">" + fullString + "</span>";
                        }
                    } else {
```

```
            wordsInCombination--;
        }
        position++;
    });
    contentSplit = contentSplit.join(' ');
    element.innerHTML = contentSplit;
    let newSpan = document.createElement("SPAN");
    newSpan.innerHTML = oldContent;
    newSpan.classList.add("sentitude-tooltiptext");
    element.appendChild(newSpan);
                }
            }
        }
        else if (msg.type == "COLOR-PG") {
            Array.from(document.getElementsByTagName("p")).forEach((element) => {
                let content = String(element.innerText)
                let style = window.getComputedStyle(element);
                if (!bannedWords.includes(content) && style.display !== 'none' && eleme
nt.offsetHeight > 0) elementsContent.push(content);
            });
xpect
            chrome.runtime.sendMessage({type: "autoParagraphScanData", requests: elemen
tsContent.length});
            elementsContent.forEach((elt) => {
                // clean the text of specials, convert to lower, and split into words
                let words = elt.toLowerCase().replace(/\n/g, ' ').replace(/[^\w\s]/g, '
').split(' ');
                // send contents to background script for scan
                chrome.runtime.sendMessage({content: words, type: "autoParagraphScan",
title: elt.innerText}, null);
            });
        }

        else if (msg.type == "COLOR-PG-DATA") {
ragraphs
            resData = msg.data;
            let i = 0;
            Array.from(document.getElementsByTagName("p")).forEach((elt) => {
                let content = String(elt.innerText)
                let style = window.getComputedStyle(elt);
                if (!bannedWords.includes(content) && style.display !== 'none' && elt.o
ffsetHeight > 0){
                    elt.classList.add("sentitude-scanned");
                    elt.style.border = "1px solid " + "hsl(" + resData[i].sentimentColo
r + ", 100%, 50%)";
                    elt.style.borderRadius = "4px";
                    elt.style.backgroundColor = "hsla(" + resData[i].sentimentColor + "
, 100%, 50%, 0.3)";
                    elt.classList.add("sentitude-tooltip");
```

```
                let spanWithResults = document.createElement("SPAN");
                spanWithResults.classList.add("sentitude-tooltiptext");
                spanWithResults.innerHTML = "Sentiment: " + resData[i].descriptorSe
ntiment + "</br>" +
                "Pleasantness: " + resData[i].descriptorPleasantness + "</br>" +
                "Attention Value: " + resData[i].descriptorAttention;
                elt.appendChild(spanWithResults);
ans
                i++;
            }
        });
    }
    else if (msg.type == "GET-SELECTION") {
        sendResponse({selection: window.getSelection().toString()});
    }
});
```

# ІЛЮСТРАТИВНА ЧАСТИНА

## Програми та сервіси для аналізу тональності

| Назва | Метод | Підтримувані мови | Платформа |
|---|---|---|---|
| Платне програмне забезпечення | | | |
| ISPRAS API: Texterra | Машинне навчання | Російська, англійська | Веб-сервіс, Python, Ruby |
| Eureka Engine | Машинне навчання | Російська | Веб-сервіс |
| SentiStrength | Словниковий | Російська, англійська, фінська, німецька, португальська, французька, італійська, валійська, грецька, шведська, польська, арабська | .NET, Java |
| BrandSpotter | Машинне навчання, правила, словниковий | Українська, російська,казахська | Веб-сервіс |
| DictaScope | Правила | російська | FreeBSD, .NET |
| IBM Watson: Tone Analyzer | Правила | Англійська, французька | Веб-сервіс |
| NetOwl | Не вказано | Англійська, арабська | Linux, .NET |
| ConveyAPI | Машинне навчння | Англійська | Веб-сервіс |
| Repustat | Машинне навчання, правила, словниковий | Арабська, китайська, німецька, датська, англійська, французька, італійська, польська, тайська, японська, корейська, іспанська | Веб-сервіс |
| Вільне програмне забезпечення | | | |
| Sentimental | Словниковий | Російська, англійська | Node.js |
| Pattern | Правила | Англійська, іспанська, німецька, французька, італійська, нідерландська | Python |
| TextBlob | Правила, машинне навчання | Англійська, іспанська, німецька, французька, італійська, нідерландська | Python |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | 08-20.МКР.012.00.000 І41 |
| Змн. | Арк. | № докум. | Підпис | Дата | |
| Розроб. | | Олійник Е. А.. | | | Метод та засіб захисту від деструктивних інформаційних впливів. Таблиця аналізу існуючих засобів класифікації тональності тексту |
| Перевір. | | Дудатьєв А. В. | | | |
| Реценз. | | Крупельницьки | | | |
| Н. Контр. | | Дудатьєв А. В. | | | |
| Затверд. | | Лужецький В. А. | | | |

| Літ. | Маса | Масштаб |
|---|---|---|
| | | |

ВНТУ, гр. 1 БС-18

## Схеми навчання методів класифікації тональності тексту

Схема роботи наївного байєсового класифікатора

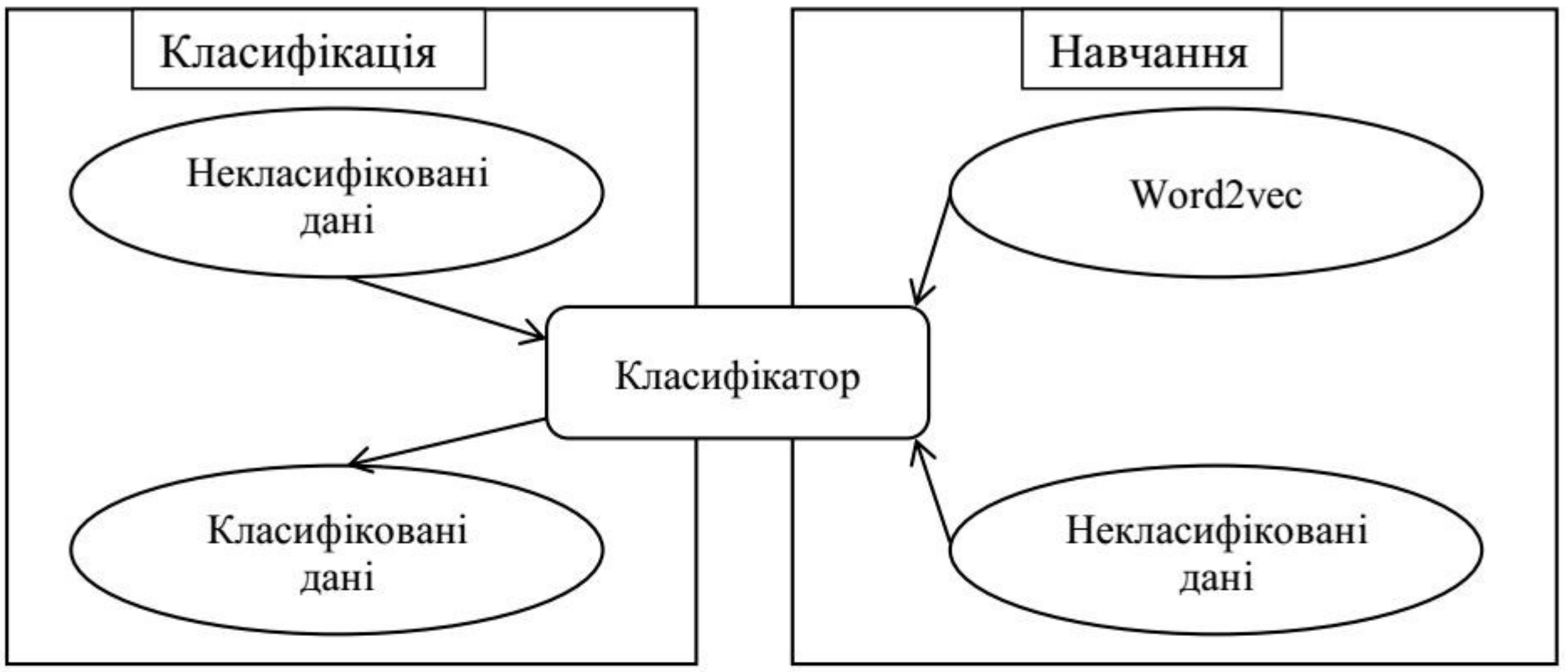
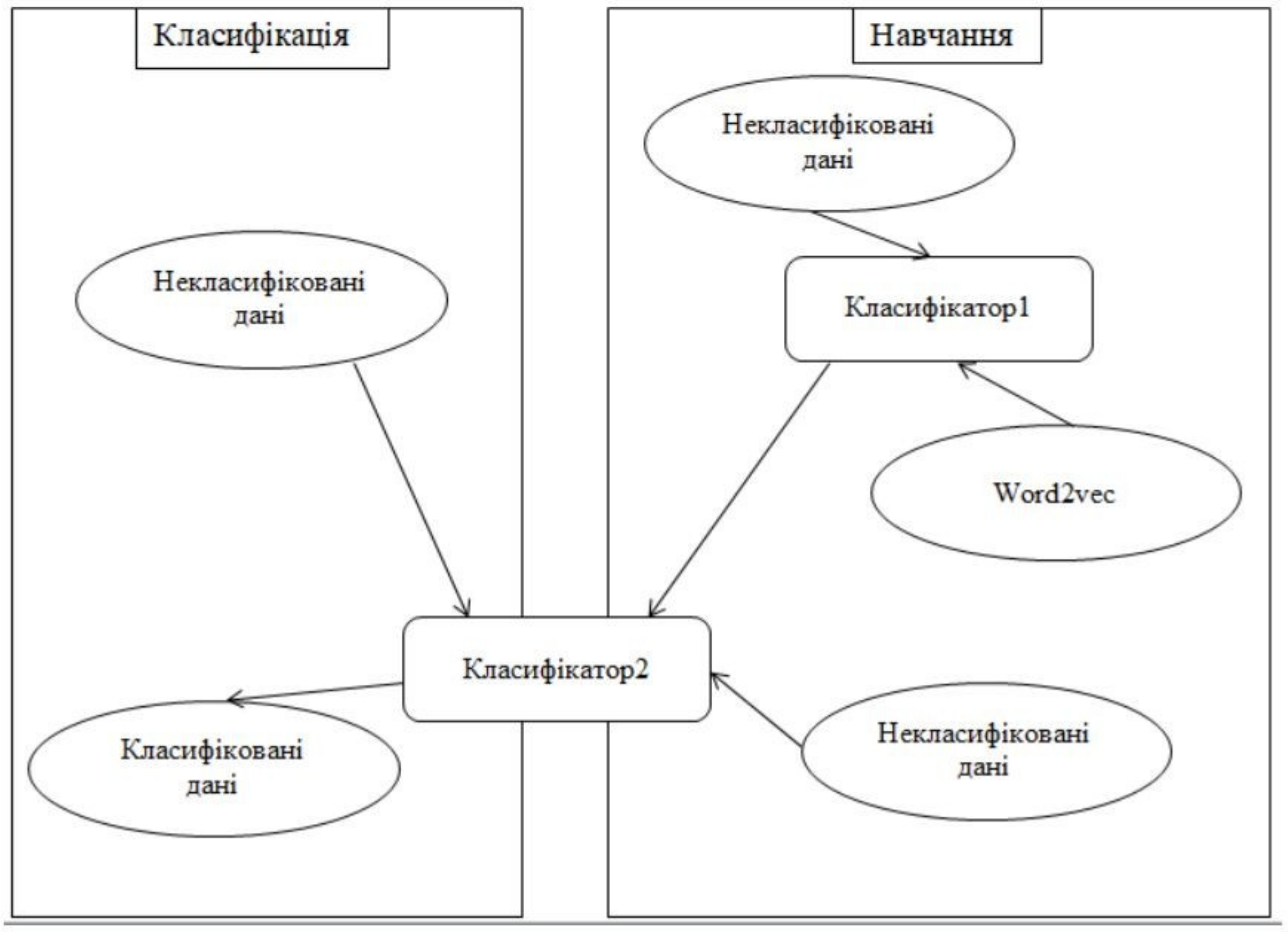



Схема покращенного методу класифікації



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | 08-20.МКР.012.00.000 І42 | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | | |
| Розроб. | | Олійник Е.А. | | | Метод та засіб захисту від деструктивних інформаційних впливів. Схеми навчання класифікаторів тональності тексту | Літ. | Маса / Масштаб |
| Перевір. | | Дудатьєв А. В. | | | | | |
| Реценз. | | Крупельницьки | | | | ВНТУ, гр. 1 БС-18 | |
| Н. Контр. | | Дудатьєв А. В. | | | | | |
| Затверд. | | ЛужецькийВ.А. | | | | | |

# Схема роботи програмного засобу

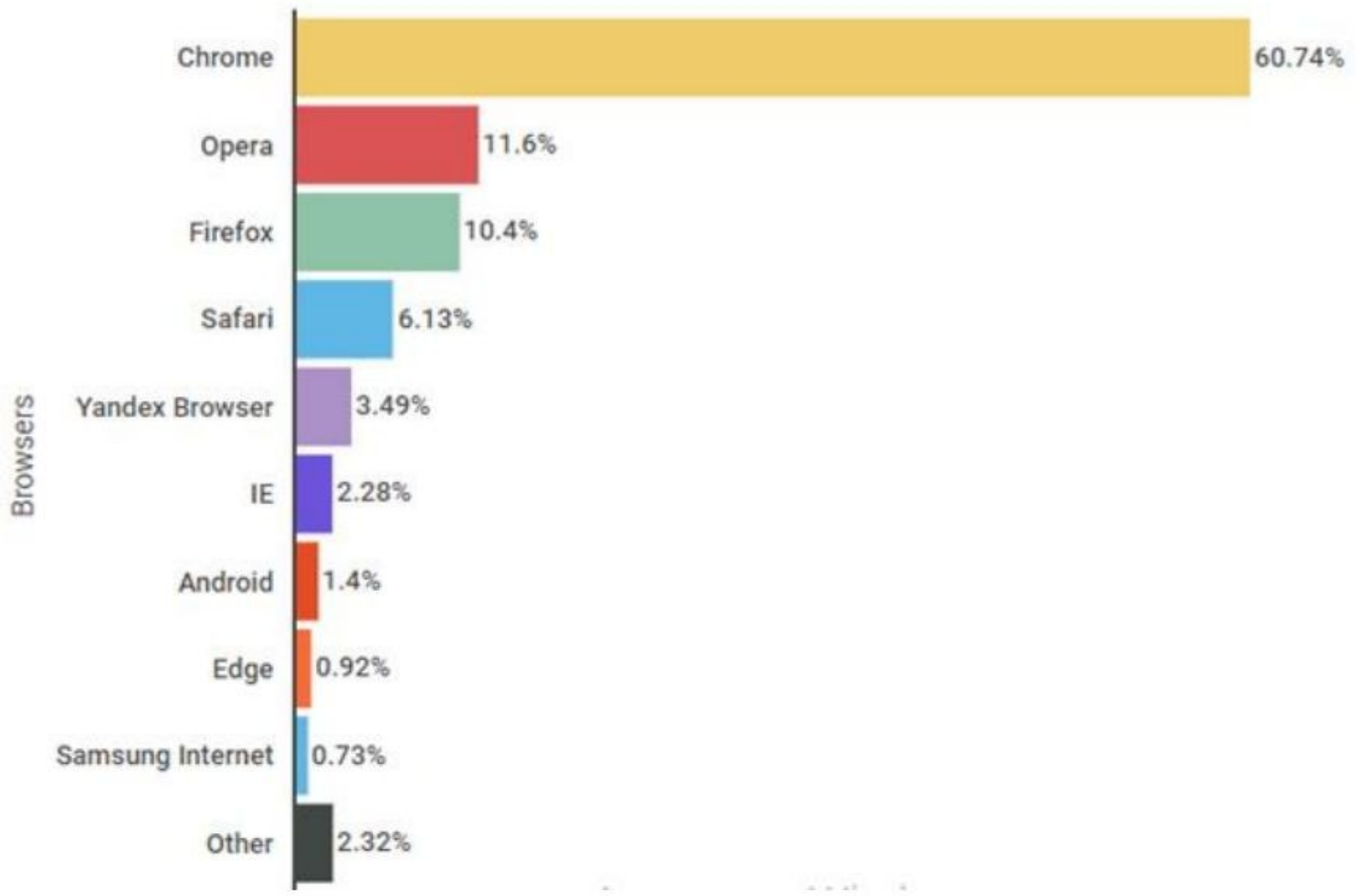
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | 08-20.МКР.012.00.000 ІЧЗ | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | | | |
| Розроб. | | Олійник Е.А. | | | Метод та засіб захисту від деструктивних інформаційних впливів. Схема роботи програмного засобу | Літ. | Маса | Масштаб |
| Перевір. | | Дудатьєв А. В. | | | | | | |
| Реценз. | | Крупельницький | | | | ВНТУ, гр. 1 БС-18 | | |
| Н. Контр. | | Дудатьєв А. В. | | | | | | |
| Затверд. | | ЛужецькийВ.А. | | | | | | |

**Порівняння ефективності методів класифікації тональності тексту**

| | Precision | Recall | Accuracy |
|---|---|---|---|
| **НБК** | | | |
| Lang-ua | 0.87 | 0.91 | 0.85 |
| IMDB | 0.93 | 0.89 | 0.86 |
| twitter_comments | 0.90 | 0.88 | 0.85 |
| **Логістична регресія** | | | |
| Lang-ua | 0.90 | 0.88 | 0.85 |
| IMDB | 0.89 | 0.93 | 0.87 |
| twitter_comments | 0.91 | 0.93 | 0.86 |
| **AdaBoost** | | | |
| Lang-ua | 0.84 | 0.91 | 0.84 |
| IMDB | 0.93 | 0.88 | 0.85 |
| twitter_comments | 0.89 | 0.91 | 0.85 |
| **Дерево рішень** | | | |
| Lang-ua | 0.88 | 0.86 | 0.82 |
| IMDB | 0.83 | 0.93 | 0.82 |
| twitter_comments | 0.87 | 0.95 | 0.84 |
| **SVM** | | | |
| Lang-ua | 0.87 | 0.90 | 0.84 |
| IMDB | 0.87 | 0.86 | 0.86 |
| twitter_comments | 0.91 | 0.88 | 0.85 |

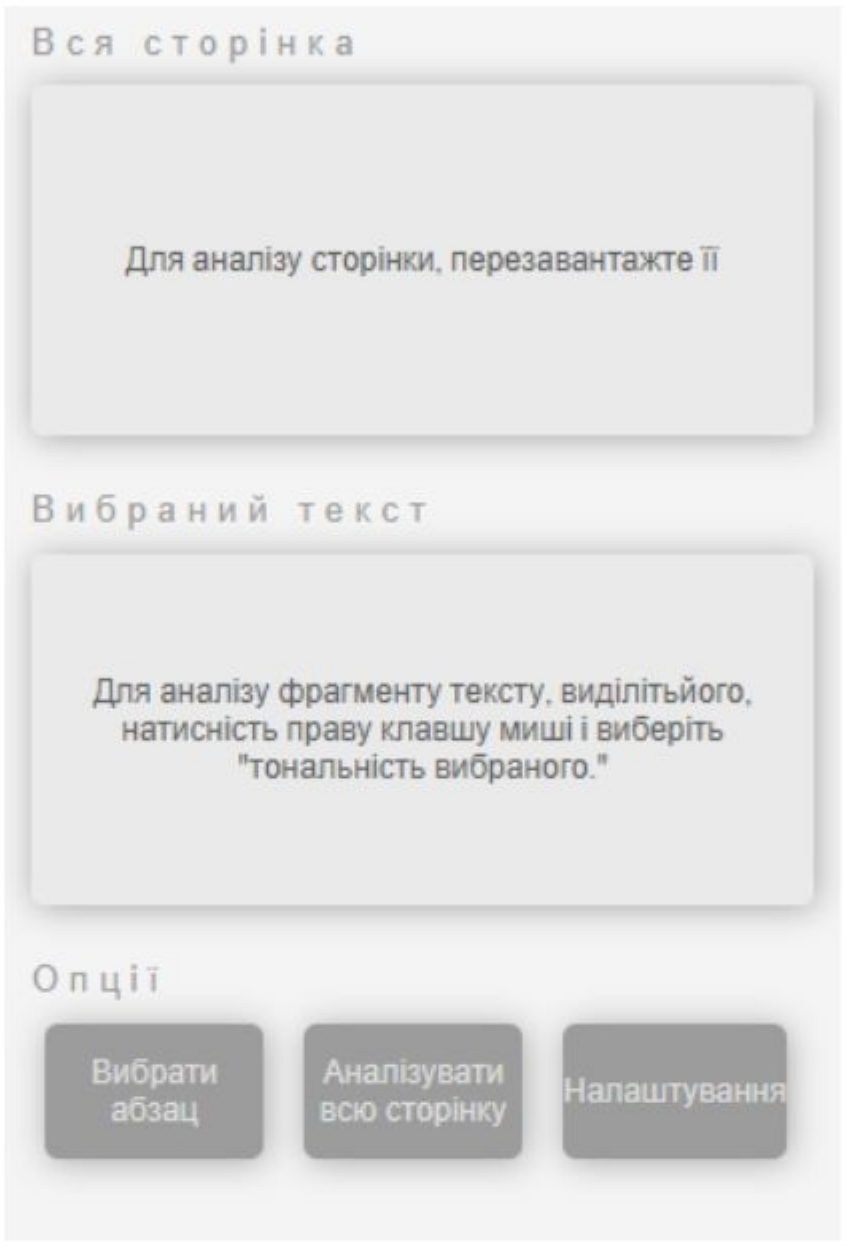
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | 08-20.МКР.012.00.000 І44 | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | | |
| Розроб. | | Олійник Е. А. | | | Метод та засіб захисту від деструктивних інформаційних впливів. Таблиця з порівнянням ефективності методів класифікації тональності тексту | Літ. | Маса / Масштаб |
| Перевір. | | Дудатьєв А. В. | | | | | |
| Реценз. | | Крупельницький | | | | ВНТУ, гр. 1 БС-18 | |
| Н. Контр. | | Дудатьєв А. В. | | | | | |
| Затверд. | | Лужецький В. А. | | | | | |

## Діаграма популярності браузерів в Україні



| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | 08-20.МКР.012.00.000 145 | |
| Змн. | Арк. | № докум. | Підпис | Дата | | |
| Розроб. | | Олійник Є. А. | | | Метод та засіб захисту від деструктивних інформаційних впливів. Діаграма популярності браузерів в Україні | Літ. / Маса / Масштаб |
| Перевір. | | Дудатьєв А. В. | | | | |
| Реценз. | | Крупельницький | | | | ВНТУ, гр. 1 БС-18 |
| Н. Контр. | | Дудатьєв А. В. | | | | |
| Затверд. | | Лужецький В. А. | | | | |

**Інтерфейс програмного засобу**




| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | 08-20.МКР.012.00.000 І46 | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | | | |
| Розроб. | | Олійник Е. А. | | | Метод та засіб захисту від деструктивних інформаційних впливів. Інтерфейс програмного засобу | Літ. | Маса | Масштаб |
| Перевір. | | Дудатьєв А. В. | | | | | | |
| Реценз. | | Крупельницьки | | | | ВНТУ, гр. 1 БС-18 | | |
| Н. Контр. | | Дудатьєв А. В. | | | | | | |
| Затверд. | | Лужецький В. А. | | | | | | |