**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

Факультет прикладної математики та інформатики
(повне найменування назва факультету)

Кафедра програмування
(повна назва кафедри)

# ДИПЛОМНА РОБОТА

«Засоби аналізу емоційного забарвлення тексту»

Виконав(ла): студент(ка) групи ПМІ-41
спеціальності 122 – комп'ютерні науки
(шифр і назва спеціальності)

_____
(підпис) (прізвище та ініціали)

Керівник _____
(підпис) (прізвище та ініціали)

Рецензент _____
(підпис) (прізвище та ініціали)

2023

## ЗМІСТ

# ВСТУП

Класифікація емоційного забарвлення повідомлень дуже важлива для аналізу відношення суспільства до продуктів та послуг компаній з метою визначення їх переваг та недоліків, аналізу реакцій відносно подій у світі, новини та телепередач задля збільшення рейтингу перегляду каналу тощо. Така класифікація забезпечує більш тісну взаємодію з користувачами та допомагає приймати рішення щодо подальшого розвитку у напрямах, які користуються більшим попитом.

Текст не має жодних власних емоцій, проте він може викликати емоції у особи, яка його читає, а також може відобразити або виразити емоційний стан людини, яка його написала. Класичні методи оцінки тональності тексту працюють на певному наборі заданих правил, за якими визначається належність тексту до певного класу емоцій. Використання наборів правил для оцінювання потребує детального аналізу великої кількості даних для додавання нових або корегування старих правил , відповідно до змін у суспільстві.

Альтернативний метод для визначення емоцій у тексті є використання технік та методів машинного навчання . У машинному навчанні проблема аналізу тональності тексту вирішується не за допомогою певного набору запрограмованих правил, а завдяки створенню певної моделі, яка може аналізувати та оцінювати дані для класифікування емоції. Частиною

машинного навчання є глибоке навчання, яке, в свою чергу, є частиною штучного інтелекту. Використовуючи глибоке навчання для вивчення даних, нейронні мережі будуть корисними інструментом, який може вирішити поставлене завдання.

Використання нейронних мереж для аналізу даних підвищує ефективність та швидкість обробки даних і не потребує трудомістких дій для корректної роботи.

Таким чином, метою дослідження є розробка інформаційної системи оцінювання тональності тексту, опис принципів її роботи та практична реалізація.

**Об’єкт дослідження:** методи розпізнавання тональності тексту.

**Предмет дослідження:** використання методик векторизації для оцінки тональності тексту на основі векторних представлень слів.

У процесі виконання поставленої задачі були виконані наступні дії:

- аналіз існуючих методів, підходів та рішень у галузі обробки природної мові та класифікації тексту за емоційним забарвленням;
- розглянуто варіанти підвищення ефективності існуючих рішень щодо аналізу тональності текстів;
- розроблено програмний продукт для розв’язання поставленої задачі;
- проведено аналіз варіантів покращень роботи системи;

**Практичне значення дослідження**

Результати даної роботи можна використовувати для аналізу емоційної реакції користувачів на певні новини, теми, визначення ставлення аудиторії певних сайтів до певних новин, персоналізації новинного сектору.

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1.Аналіз тональності текстів

Тональність – це ставлення автора висловлювання до того, про що йде мова в тексті (об'єкту реального світу, події, процесу або їх властивостей), виражене в тексті. Емоційна складова, виражена на рівні лексеми або комунікативного фрагмента, називається лексичної тональністю (або лексичним сентиментом). Тональність всього тексту в цілому можна визначити як функцію (в найпростішому випадку суму) лексичних тональностей складових його одиниць (речень) і правил їх поєднання.

Аналіз тональності текстів – це поле для обробки природних мов (Natural language processing, NLP), яке будує системи, які намагаються ідентифікувати і витягти думки в тексті. Зазвичай, крім ідентифікації думки, ці системи витягують атрибути виразу, наприклад:

– полярність: оратор висловлює позитивну чи негативну думку;

– тема: про те, про що говорять;

– власник думки: особа або організація, яка висловлює свою думку.

В даний час аналіз настроїв є цікавим напрямом, що розвивається, оскільки має багато практичних застосувань. Оскільки публічно та приватно доступна інформація по Інтернету постійно зростає, велика кількість текстів, що

висловлюють свої думки, доступна на сайтах рецензування, форумах, блогах і соціальних медіа.

За допомогою систем аналізу настроїв ця неструктурована інформація може автоматично трансформуватися у структуровані дані громадської думки про продукти, послуги, бренди, політику або будь-яку тему, щодо якої люди можуть висловлювати свої думки. Ці дані можуть бути дуже корисними для комерційних додатків, таких як маркетинговий аналіз, зв'язки з громадськістю, огляди продуктів, підрахунок чистого промоутера, відгуки про продукцію та обслуговування клієнтів.

Текстову інформацію можна розділити на два основні типи: факти та думки. Факти є об'єктивними виразами про щось. Думки - це, як правило, суб'єктивні вирази, що описують почуття, оцінки та почуття людей до теми чи теми.

Аналіз сентиментів, як і багато інших задач NLP, може бути змодельований як проблема класифікації, де необхідно вирішити двіі підпроблеми:

– класифікація речення як суб'єктивна або об'єктивна, відома як класифікація суб'єктності;

– класифікація речення як вираження позитивного, негативного або нейтрального думки, відома як класифікація полярності.

На думку, об'єкт, про який говорить текст, може бути об'єктом, його складовими, його аспектами, його атрибутами або його особливостями. Це також може бути продукт, послуга, особа, організація, подія або тема. Наприклад: "Термін служби акумулятора цієї камери надто короткий" – висловлюється негативна думка про особливість (час автономної роботи) суб'єкта (камери).

Існує два види думок: прямий і порівняльний. Прямі висновки дають висновок про суб'єкт безпосередньо, наприклад: "Якість зображення камери А погана.". Ця пряма думка висловлює негативну думку про камеру А.

У порівняльних висновках думка виражається шляхом порівняння суб'єкта, про який йдеться мова в тексті, з іншим, часто таким, що належить до такого ж классу, наприклад: "Якість зображення камери А краще, ніж у камери В.". Як правило, порівняльні думки виражають подібності або відмінності між двома або більше суб'єктами, використовуючи порівняльну або чудову форму прикметника або прислівника. У попередньому прикладі існує позитивна думка про камеру А і, навпаки, негативна думка про камеру Б.

Явна думка по темі - це думка, явно виражена в суб'єктивному реченні. Наступне речення виражає явну позитивну думку: "Якість голосу цього телефону дивовижна.".

Імпліцитна думка з цього питання є думкою, що передбачається в об'єктивному реченні. Наступне речення виражає неявну негативну думку: "Наушник зламався через два дні". У неявних думках ми могли б включати метафори, які можуть бути найскладнішим типом думок для аналізу, оскільки вони містять багато семантичної інформації.

## 1.2 Типи аналізу тональностей

В сучасних системах автоматичного визначення емоційної оцінки тексту найчастіше використовується одномірний емотивний простір: позитив чи негатив (добре або погано). Однак відомі успішні випадки використання і багатовимірних просторів.

Основним завданням в аналізі тональності є класифікація полярності текста, тобто визначення, чи є виражена в тексті думка позитивною, негативною або нейтральною. Більш розгорнута класифікація тональності виражається, наприклад, такими емоційними станами, як «злий», «сумний» і «щасливий».

Полярність документа можна визначати за бінарною шкалою. У цьому випадку для визначення полярності документа використовується два класи оцінок: позитивна чи негативна.

Одним з недоліків цього підходу є те, що емоційну складову документа не завжди можна однозначно визначити, тобто документ може містити ознаки як і позитивної, так і негативної. Ранні роботи в цій області включають в себе праці Терні і Панга, які застосовують різні методи розпізнавання полярності оглядів товару і відгуків про фільми відповідно.

Можна класифікувати полярність документа по багатосмуговій шкалі, що була винайдена Пангом і Снайдером. Ними було розширене основне завдання класифікації кіновідгуків від оцінки «позитивна або негативна» в бік прогнозування рейтингу по 3-х або 4-бальній шкалі. У той же час Снайдер провів поглиблений аналіз оглядів ресторанів, пророкуючи рейтинги їх різних властивостей, таких як їжа і атмосфера (за 5-бальною шкалою).

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

Інший дослідницький напрямок - це ідентифікація суб'єктивності/об'єктивності. Це завдання зазвичай визначається як віднесення даного тексту в один з двох класів суб'єктивний й/або об'єктивний. Ця проблема іноді може бути більш складною, ніж класифікація полярності: суб'єктивність слів і фраз може залежати від контексту, а об'єктивний документ може містити в собі суб'єктивні пропозиції (наприклад, новинна стаття, цитує думки людей). Більше того, як згадував Су, результати більшою мірою залежать від визначення суб'єктивності, вживаних в рамках анотації текстів. Якби це не було, Панг показав, що видалення об'єктивних пропозицій з документа перед класифікацією полярності допомогло підвищити точність результатів.

Модель більш докладного аналізу називається аналізом на основі функції/аспекту. Ця модель посилається на ухвалу думок або настроїв,

виражених різними функціями або аспектами сутностей, наприклад, у стільникового телефону, цифрової камери або банку. Властивість/аспект – це атрибут або компонент сутності, досліджуваної на тональність, наприклад, екран мобільного телефону або ж якість зйомки камери. Ця проблема вимагає вирішення ряду завдань, наприклад, ідентифікація актуальних сутностей, витяг їх функцій аспектів. Більш докладні дискусії на цей рахунок можуть бути знайдені в довіднику з NLP, у главі «Аналіз тональності та суб'єктивності».

Існує багато видів аналізу настроїв, а інструменти аналізу тональностей варіюються від систем, які зосереджуються на полярності (позитивні, негативні, нейтральні) до систем, які виявляють почуття і емоції (сердиті, щасливі, сумні тощо) або визначають наміри (наприклад, зацікавлені, не зацікавлений).

Інколи ви можете бути більш точними щодо рівня полярності думки, тому замість того, щоб просто говорити про позитивні, нейтральні або негативні думки, можна розглянути такі категорії:

– дуже позитивні;

– позитивні;

– нейтральні;

– негативні;

– дуже негативні.

Це зазвичай називають дрібнозернистим аналізом. Це може бути нанесено на 5-зірковий рейтинг в огляді, наприклад: дуже позитивний = 5 зірок і дуже негативний = 1 зірка.

Деякі системи також надають різні відтінки полярності, визначаючи, чи позитивні або негативні настрої асоціюються з певним почуттям, таким як гнів, смуток (тобто негативні почуття) або щастя, любов та ентузіазм (тобто позитивні почуття).

Виявлення емоцій спрямоване на оцінку таких емоцій, як щастя, розчарування, гнів, смуток і тому подібне. Багато систем виявлення емоцій

вдаються до лексиконів (тобто списків слів і емоцій, які вони передають) або складних алгоритмів машинного навчання.

Як правило, при аналізі настроїв суб'єктів, вас може зацікавити не тільки те, що люди говорять з позитивною, нейтральною або негативною полярністю про продукт, а й про його особливості. Ось про що йдеться в аспект-аналізі. У нашому попередньому прикладі: "Термін служби акумулятора цієї камери надто короткий".

Цей приклад висловлює негативну думку про камеру, а точніше, про час автономної роботи, що є особливістю камери.

Багатомовний аналіз настроїв може бути складним завданням. Як правило, необхідна велика кількість попередньої обробки, і попередня обробка використовує ряд ресурсів. Більшість цих ресурсів доступні в Інтернеті (наприклад, лексикони сентиментів), але потрібно створити багато інших (наприклад, перекладені корпуси або алгоритми виявлення шуму). Використання доступних ресурсів вимагає багато досвіду кодування і може зайняти багато часу для реалізації.

## 1.3 Огляд відомих рішень

### 1.3.1. Brand24

Brand24 – сервіс, який працює з більш ніж 30000 брендів по всьому світу. Його користувачами є багато відомих компаній, серед яких Panasonic, IKEA, H&M, Raiffeisen Bank.

Сервіс надає мільйони одиниць даних з більш ніж 25-ти соціальних мереж, новинних порталів, блогів, форумів і т. д. З його допомогою можна знайти згадки про конкурентів в зазначених джерелах і визначити ставлення клієнтів до компанії.

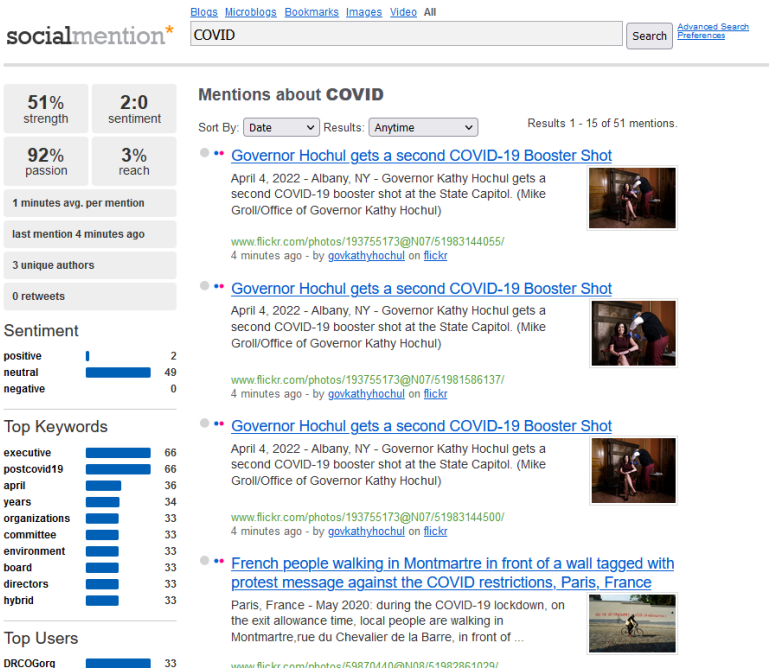
Звіт про отриманих даних містить графіки кількостей згадок і їх емоційної оцінки, сторінки сайтів з знайденими згадками. Всі результати відсортовані за різними видами джерел (Facebook, Twitter, відео, фото, блог, форум тощо).


### 1.3.2. Tweet Sentiment Visualization

Social Mention – це платформа пошуку та аналізу інформації у всесвітній мережі, яка спеціалізується на відстежуванні відгуків користувачів щодо певного предмету, бренду, новини тощо. Система дозволяє дуже легко відстежувати та оцінювати відгуки користувачів у режимі реального часу. Social Mention проводить моніторинг великої кількості соціальних мереж та інтернет- ресурсів, в тому числі Twitter, Facebook, YouTube тощо. За ключовим словом (новиною, назвою продукту, назвою бренду тощо) можливо отримати інформацію щодо кількості позитивних, негативних та нейтральних згадувань їх пропорційне відношення один до одного, а також кількість унікальних публікацій, їх пересилань та статистичні дані щодо частоти пошуку за згадуваним ключовим словом.


### 1.3.3. Social Mention

Social Mention – це платформа пошуку та аналізу інформації у всесвітній мережі, яка спеціалізується на відстежуванні відгуків користувачів щодо певного предмету, бренду, новини тощо. Система дозволяє дуже легко відстежувати та оцінювати відгуки користувачів у режимі реального часу. Social Mention проводить моніторинг великої кількості соціальних мереж та інтернет-ресурсів, в тому числі Twitter, Facebook, YouTube тощо. За ключовим словом (новиною, назвою продукту, назвою бренду тощо) можливо отримати інформацію щодо кількості позитивних, негативних та нейтральних згадувань їх пропорційне відношення один до одного, а також кількість унікальних публікацій, їх пересилань та статистичні дані щодо частоти пошуку за згадуваним ключовим словом.




# РОЗДІЛ 2. ОГЛЯД ЗАСТОСОВУВАНИХ ТЕХНОЛОГІЙ

Світ машинного навчання розвивається настільки стрімко, що методи, повсюдно застосовувані лише кілька років тому, сьогодні вже вважаються

застарілими. Багато компаній досі використовують такі застарілі методи і навіть вимагають їх знання від шукачів вакансій.

Будь-яка модель аналізу тональності (навіть найсучасніша) складається з трьох кроків: очищення та попередня обробка тексту, векторизація та моделювання.

## 2.1. Очищення та обробка тексту

Сирий текст завжди містить зайві символи, які заважають його інтерпретувати: табуляцію, переклади рядка тощо. Найчастіше видаляють і більшість розділових знаків: якщо ми збираємося завантажувати в нашу модель текст за однією пропозицією, нам нібито достатньо розділити текст на пропозиції. Якщо ж ми завантажуємо текст за абзацами або відразу весь, ніякі розділові знаки взагалі не потрібні. Як аналіз тональності тексту може ігнорувати, наприклад, знаки оклику? Таких питань чомусь ніхто не ставить. На цьому етапі часто проводиться видалення так званих "стоп-слів" - прийменників, які нібито не можуть впливати на тональність тексту і лише заважають його аналізувати. При цьому навіть в англійській мові "get up" - зовсім не те саме, що "get out", і видалення "зайвих" коротких слів може призвести до втрати критично важливої інформації.

Очищення здійснюється за допомогою регулярних виразів (regex). Ми видаляємо розділові знаки, цифри та теги HTML:

```
import re

REPLACE_NO_SPACE = re.compile("(\.)|(\;)|(\:)|(\!)|(\')|(\?)|(\,)|(\")|(\()|(\))|(\[)|(\])|(\d+)")
REPLACE_WITH_SPACE = re.compile("(<br\s*/><br\s*/>)|(\-)|(\/)|(\n)")
NO_SPACE = ""
SPACE = " "

def preprocess_reviews(reviews):

    reviews = [REPLACE_NO_SPACE.sub(NO_SPACE, line.lower()) for line in reviews]
    reviews = [REPLACE_WITH_SPACE.sub(SPACE, line) for line in reviews]

    return reviews

reviews_train_clean = preprocess_reviews(reviews_train)
```

**Векторизація.** Моделі машинного навчання не можуть працювати з текстовими даними, тому тексти потрібно перетворити на числа. Як правило, в результаті виходять одновимірні масиви чисел – вектори. Від того, як ви отримуєте ці вектори, багато в чому залежить максимальна точність моделі. Практично всі застарілі підходи кодують слова без урахування контексту, тобто слово "bank" у виразах "river bank" та "bank of Canada" кодується однаково. Методики векторизації включають:

**TF-IDF**. Використовується для оцінки важливості слів у контексті документа, що є частиною колекції документів чи корпусу. Вага (значимість) слова пропорційна кількості вживань цього слова у документі, і обернено пропорційна частоті вживання слова у інших документах колекції.

Показник TF-IDF використовується в задачах аналізу текстів та інформаційного пошуку. Його можна застосовувати як один з критеріїв релевантності документа до пошукового запиту, а також при розрахунку міри спорідненості документів при кластеризації.

**Стеммінг**. Це обрізання слів, щоб привести їх до єдиних загальних форм. Наприклад, слова "люблю", "любив", "любить", "любити" можна привести до єдиної форми "люб", і при цьому не втратити інформацію. З іншого боку, слова "улюблений" і "улюбленець" несуть зовсім різне емоційне забарвлення, і тут обрізання до єдиної форми неодмінно призведе до втрати інформації.

**Лематизацію**. Кожне слово наводиться для його словникової основи, званої лемою. Очевидно, що при правильно складеному словнику лематизація здатна домогтися набагато кращої уніфікації слів, ніж стеммінг, з меншим ризиком втрати інформації. Наприклад, слово "йшла" має в якості леми дієслово "йти", і жодна обрізка слова не дозволяє її отримати. Зрозуміло, для лемматизації потрібен величезний словник, у якому мають бути всі слова аналізованої мови.

**CountVectorizer**. Один із найпростіших способів векторизації тексту – це збір усіх можливих слів і створення для тексту єдиного вектора, в якому для кожного слова буде вказано кількість його входжень у текст (звичайний режим) або наявність у тексті (двійковий режим). Очевидно, що при цьому ми втрачаємо всю інформацію про порядок слідування слів, але, як ми побачимо нижче, інформації, що залишилася, достатньо для отримання досить високої точності передбачення.

**N-грами**. Якщо кодувати не тільки наявність слів, а ще й наявність послідовностей із двох і трьох слів (біграм та триграм), це дозволить частково компенсувати відсутність інформації про порядок їхнього слідування. Словник при цьому виходить просто величезний, але оскільки більшість коду кожного тексту буде заповнена нулями, його можна представити у вигляді розрідженої матриці, з якою добре працює Python.

**Моделювання.** Можна використовувати будь-яку модель машинного навчання, починаючи з найпростішої лінійної регресії. Оскільки наші дані сильно розріджені, застосуємо лінійний метод опорних векторів, який чудово справляється з такими даними:

```
from sklearn.metrics import accuracy_score
from sklearn.svm import LinearSVC

for c in [0.001, 0.005, 0.01, 0.05, 0.1]:
    svm = LinearSVC(C=c)
    svm.fit(X_train, y_train)
    print ("Accuracy for C=%s: %s" % (c, accuracy_score(y_val, svm.predict(X_val))))

final = LinearSVC(C=0.01)
final.fit(X, target)
print ("Final Accuracy: %s" % accuracy_score(target, final.predict(X_test)))
```

2.2. Що таке BERT, і чи можна до нього підготуватися?

BERT – це технологія обробки запитів користувачів і формування результатів видачі, в основі якої лежить використання машинного навчання.

Мета – давати користувачам максимально точну і коректну відповідь на неоднозначні (багатослівні, діалогові) пошукові запити.

Модель BERT (Bidirectional Encoder Representations from Transformers – “двонаправлені презентації кодувальника для трансформерів”) була представлена світові у статті, опублікованій дослідниками з Google AI Language. Вона викликала неабиякий ажіотаж у співтоваристві машинного навчання, представивши найпередовіші на сьогоднішній день результати для цілого ряду різних NLP (Natural Language Processing — “обробка природної мови”) завдань, включаючи формування відповідей на питання (SQuAD v1.1), формування міркувань на природному. мові (MNLI) та безліч інших категорій.

Ключовим технічним нововведенням BERT є застосування двонаправленого навчання трансформерів (популярної нині моделі з механізмом уваги) до мовного моделювання. Цей підхід йде в розріз з попередніми роботами, які розглядали текстову послідовність або лише зліва направо, або поєднували навчання зліва направо та праворуч наліво. Результати цієї роботи показують, що мовна модель з двонаправленим навчанням здатна досягти глибшого розуміння мовного контексту та потоку, ніж односпрямовані мовні моделі. У статті дослідники докладно описують нову техніку під назвою MLM (Masked Language Model - "масковане мовне моделювання"), яка дозволяє проводити двонаправлене навчання у моделях, для яких раніше це було неможливо.

### 2.2.1.У чому особливість алгоритму BERT?

BERT покликаний не просто підбирати результати видачі методом пошуку слів, які ввів користувач. Ця технологія буде «зчитувати» запит як звичайна людина і вловлювати прихований сенс. Моделі BERT можуть враховувати повний контекст запиту, розглядаючи слова, які йдуть до і після нього, що особливо корисно для розуміння мети пошукових запитів.

Запровадження моделей BERT відбувається поступово, по країнам. Коли технологію запровадять у пошук у всьому світі, представники пошуковика не коментують.

Але всіх уже цікавлять питання:

1. Як це вплине на видачу?
2. Як підготуватися до впровадження BERT?

Для користувачів це означає, що шанси отримати максимально релевантні результати незалежно від виду запиту збільшуються. Особливо це стосується:

- низькочастотних запитів із «довгим хвостом»;
- результатів пошуку за допомогою голосу;
- запитів, коли користувач сам не до кінця розуміє, про що запитує, і вводить не точні запити, а навколотематичні питання або просто набір слів, які, на його думку, пов'язані з темою, що його цікавить.

Підготовка до BERT... Підготовка не має на увазі впровадження чогось нового, незвіданого. Власникам сайтів і SEO-фахівцям варто продовжувати приділяти особливу увагу:

- Експертному контенту на сайтах. Не загальні фрази з теми, а детальний опис продукту чи послуги. Ще один аргумент: краще раз заплатити за якісний текст (якщо немає можливості написати самостійно), ніж шукати варіанти дешевого SEO-копірайтингу, який просто не працюватиме.
- Докладній відповіді на інтент користувача. Не просто текст, що відповідає одному цільовому пошуковому запиту (або групі дуже близьких), а максимально детальну відповідь на всі навколотематичні підказки. Важливо працювати не тільки з частотними запитами.
- Оптимізації під максимально широкий пул запитів. Це важливо було ще вчора, оскільки це не просто завдання підготовки до BERT, а механізм залучення на сайт максимально широкої цільової аудиторії.

### 2.2.2. Передісторія

Дослідники в галузі комп'ютерного зору неодноразово демонстрували користь трансферного навчання (попереднього навчання моделі нейронної мережі на добре відомому завданні, як наприклад ImageNet, з подальшим донавчанням) з використанням вже навченої нейронної мережі як основа для нової моделі з конкретною спрямованістю. В останні роки дослідники дійшли висновку, що подібна техніка може бути надзвичайно корисною і в багатьох завданнях обробки природної мови.

Ще один підхід, проілюстрований у статті ELMo, який також дуже популярний у NLP-завданнях – це навчання на основі ознак. У цьому підході попередньо навчена нейронна мережа створює векторні уявлення слів (word embeddings), які потім використовуються як ознаки у NLP-моделях.

### 2.2.3. Як працює BERT

BERT використовує трансформер - механізм "уваги", який вивчає контекстуальні відносини між словами (або підслів) у тексті. У своїй оригінальній формі трансформер включає два окремих механізми — кодувальник, який зчитує введений текст, і декодер, який видає прогноз для завдання. Оскільки метою BERT є створення мовної моделі, їй необхідний тільки кодувальник.

На наведеному нижче графіку представлена загальна структура кодувальника трансформера. Вхід є послідовністю токенів, які спочатку вбудовуються у вектори, а потім обробляються в нейронній мережі. Вихід є

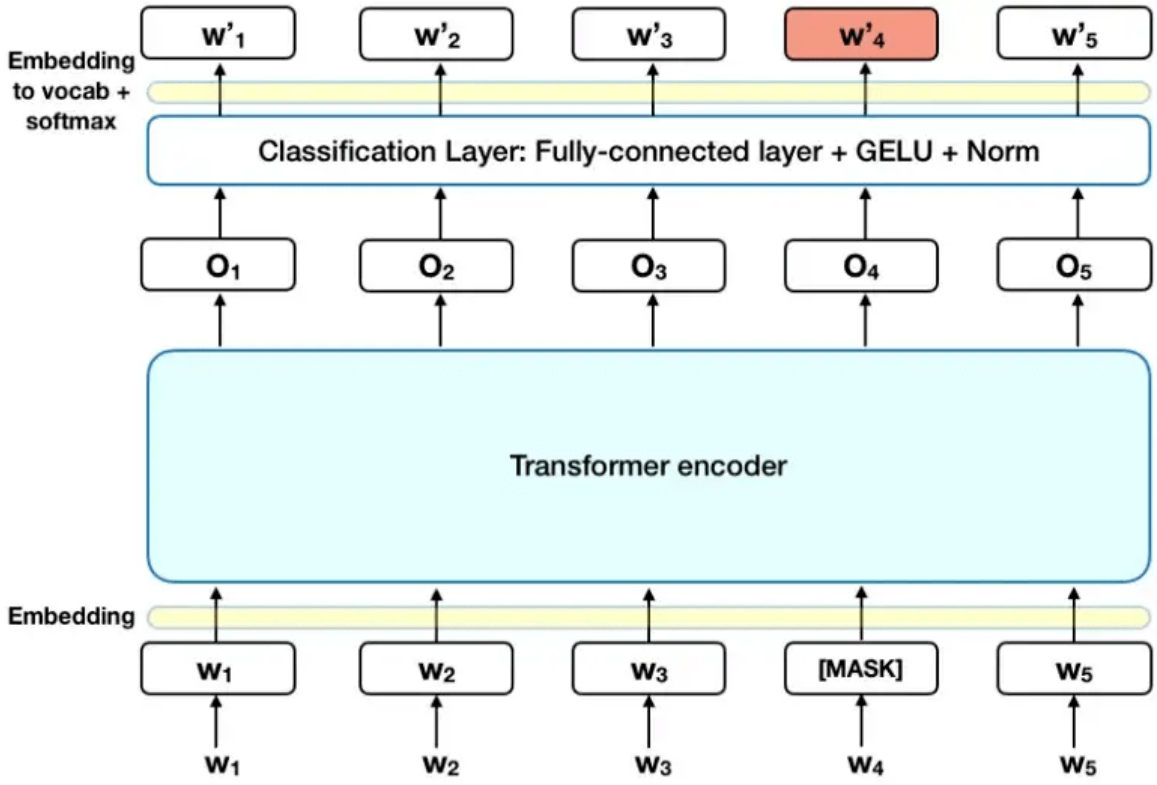
послідовністю векторів розміру H, в якій кожен вектор відповідає вхідному токену з тим же індексом.

Під час навчання мовних моделей виникає проблема визначення мети прогнозування. Багато моделей пророкують наступне слово в послідовності (наприклад, "Дитина прийшла додому з ___") - це спрямований підхід, який за своєю суттю обмежує контекстне навчання. Щоб подолати цю проблему, BERT використовує дві стратегії навчання:

Масковане мовне моделювання (MLM). Перед введенням послідовності слів у BERT 15% слів у кожній послідовності замінюється токеном [MASK]. Потім модель намагається передбачити вихідне значення замаскованих слів на основі контексту, який надається іншими, не замаскованими словами в послідовності. З технічної точки зору, передбачення вихідних слів вимагає:

1. Додавання шару класифікації поверх вихідних даних кодувальника.
2. Примноження вихідних векторів на матрицю векторних уявлень словника (embedding matrix), що перетворює їх у розмірність словника.
3. Розрахунок ймовірності кожного слова у словнику за допомогою softmax.

Функція втрат BERT враховує лише прогнози замаскованих значень та ігнорує прогнози не замаскованих слів. Як наслідок, модель сходиться повільніше, ніж спрямовані моделі, що компенсується більшою поінформованістю про контекст.

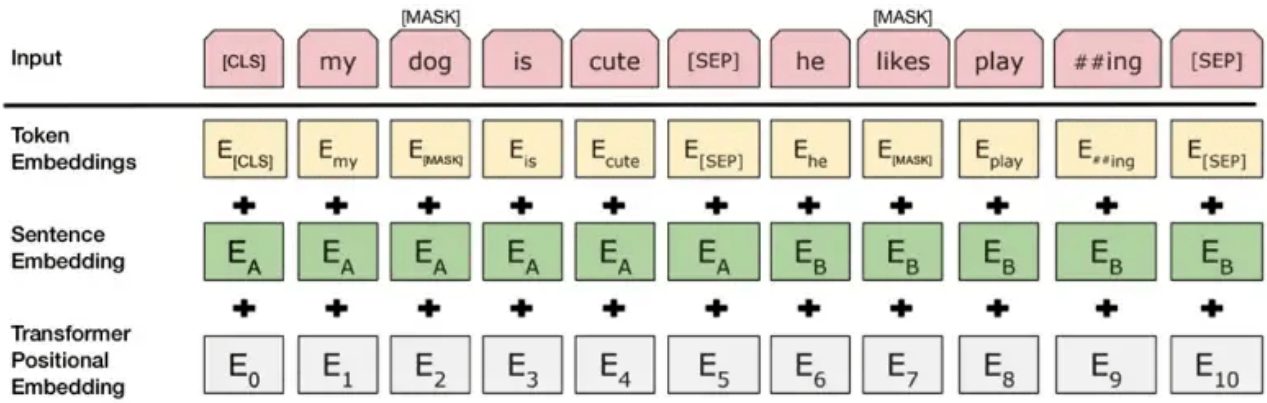



Прогнозування наступної пропозиції (NSP). В рамках процесу навчання BERT модель як вхідні дані отримує пари фраз, на яких вона вчиться передбачати, чи є друга фраза в парі наступної після першої у вихідному тексті. Під час навчання 50% вхідних даних є пари, в яких друга фраза дійсно є наступною фразою у вихідному тексті, а в інших 50% як друга фраза вибирається випадкова фраза з того ж тексту. Передбачається, що випадкова фраза не пов'язана за змістом з першою фразою.

Щоб допомогти моделі розрізнити дві фрази у процесі навчання, перед входом у модель вхідні дані обробляються так:

1. На початок першої фрази вставляється токен [CLS]. Наприкінці кожної фрази вставляється токен [SEP].
2. До кожного токену додається ембеддинг (векторне уявлення) фрази, що позначає Фразу А або Фразу В. Ембеддінг фраз за своєю концепцією аналогічні ембеддінг токенів зі словником з двох елементів.

3. До кожного токену додається ембеддинг позиційний, щоб вказати його положення в послідовності. Концепція та реалізація позиційного ембеддингу добре розкриті у статті, присвяченій трансформеру.




Щоб передбачити, чи справді друга фраза пов'язана з першою, виконуються такі кроки:

1. Вся вхідна послідовність відбувається через модель-трансформер.
2. Вихід токена [CLS] перетворюється на вектор розмірності 2×1 за допомогою простого шару класифікації (навчені матриці ваги і зсувів).
3. Обчислення ймовірності IsNextSequence за допомогою softmax.

У процесі навчання BERT-моделі, MLM та NSP навчаються разом із метою мінімізувати комбіновану функцію втрат двох стратегій.

### 2.2.4. Як використовувати BERT (донавчання)

Використовувати BERT для конкретного завдання щодо просто:

BERT можна використовувати для різних мовних завдань, додаючи невеликий для коригування базової моделі:

1. Завдання класифікації, такі як, наприклад, аналіз тональності, виконуються аналогічно класифікації наступної фрази, додаючи шар класифікації поверх вихідних даних трансформера для токена [CLS].
2. В задачах формирования ответов на вопросы (например, SQuAD v1.1) программа получает вопрос относительно текстовой последовательности и должна отметить ответ в этой последовательности. Используя BERT, модель вопрос/ответ можно обучить, изучая два дополнительных вектора, которые отмечают начало и конец ответа.
3. При распознавании именованных объектов (NER - Named Entity Recognition) программа получает текстовую последовательность и должна помечать различные типы объектов (человек, организация, дата и т. д.), которые появляются в тексте. Используя BERT, NER-модель можно обучить, пропуская выходной вектор каждого токена через классификационный слой, который прогнозирует NER-метку.

## 2.3.Трансформер (модель машинного навчання)

Трансфо́рмер (англ. Transformer) — це модель глибинного навчання, яка переймає механізм уваги, роздільно зважуючи важливість кожної частини даних входу. Її використовують переважно в області обробки природної мови (ОПМ) та в комп'ютерному баченні (КБ).

Як і рекурентні нейронні мережі (РНМ), трансформери призначено для обробки послідовних даних входу, таких як природна мова, для таких задач як переклад та реферування тексту. Проте, на відміну від РНМ, трансформери оброблюють дані не обов'язково послідовно. Радше, механізм уваги забезпечує контекст для будь-якого положення в послідовності входу. Наприклад, якщо дані входу є реченням природної мови, то трансформерові не потрібно обробляти його початок, перш ніж взятися за обробку його кінця. Він, радше, визначає

контекст, який надає значення кожному слову в цій послідовності. Ця властивість уможливлює набагато більше розпаралелювання, ніж РНМ, і відтак знижує тривалості тренування.

З моменту свого дебюту 2017 року трансформери все частіше стають обраною моделлю для задач ОПМ, замінивши моделі РНМ, такі як довга короткочасна пам'ять (ДКЧП). Додаткове розпаралелювання тренування уможливлює тренування на більших наборах даних, ніж це було колись можливим. Це призвело до розробки попередньо натренованих систем, таких як BERT (англ. Bidirectional Encoder Representations from Transformers) та GPT (англ. Generative Pre-trained Transformer), які було треновано на великих мовних наборах даних, таких як корпуси Вікіпедії та Common Crawl, і які можливо тонко налаштовувати для конкретних мовних задач.

До трансформерів більшість сучасних системи ОПМ покладалися на вентильні РНМ, такі як ДКЧП та вентильні рекурентні вузли (ВРВ), з доданими механізмами уваги (англ. attention mechanisms). Трансформери будують на цих технологіях уваги без застосування структури РНМ, підкреслюючи той факт, що механізми уваги самі по собі можуть відповідати продуктивності РНМ з увагою.

### 2.3.1. Послідовна обробка

Вентильні РНМ обробляють лексеми послідовно, підтримуючи вектор стану, що містить подання даних, побачене після кожної лексеми. Щоб обробити n-ту лексему, ця модель поєднує стан, що подає речення по лексему $n - 1$, з інформацією про цю нову лексему, щоби створити новий стан, що подає речення по лексему n. Теоретично, інформація від однієї лексеми може поширюватися послідовністю як завгодно далеко, якщо в кожній точці стан продовжує кодувати контекстну інформацію про цю лексему. На практиці цей механізм має вади: проблема зникання градієнту залишає стан моделі в кінці довгого речення без точної, добутної інформації про попередні лексеми.

**Увага.** Цю проблему було розв'язано механізмами уваги. Механізми уваги дозволяють моделі робити висновки зі стану в будь-якій ранішій точці протягом послідовності. Шар уваги має доступ до всіх попередніх станів, і зважує їх відповідно до навченої міри або доречності, забезпечуючи доречну інформацію про віддалені лексеми.

Яскравим прикладом цінності уваги є мовний переклад, у якому для визначення значення слова в реченні є важливим контекст. У системі перекладу з англійської українською перше слово україномовного виходу найімовірніше сильно залежить від перших декількох слів англомовного входу. Проте в класичній моделі ДКЧП, щоби виробити перше слово україномовного виходу, моделі дається лише вектор стану крайнього англомовного слова. Теоретично цей вектор може кодувати інформацію про все англомовне речення, даючи моделі все необхідне знання. На практиці ДКЧП часто зберігає цю інформацію погано. Для подолання цієї проблеми може бути додано механізм уваги: декодувальникові надають доступ до векторів стану кожного вхідного англомовного слова, а не лише крайнього, і він може навчатися ваг уваги, що диктують, як багато уваги приділяти кожному з векторів стану англомовного входу.

При додаванні до РНМ механізми уваги покращують продуктивність. Розробка архітектури Трансформера показала, що механізми уваги були потужними самі по собі, і що послідовна рекурентна обробка даних не була необхідною для досягання виграшу в продуктивності РНМ з увагою. Трансформери використовують механізм уваги без РНМ, обробляючи всі лексеми одночасно, й обчислюючи ваги уваги між ними в послідовних шарах.

### 2.3.1. Архітектура

Подібно до раніших моделей, трансформер використовує кодувально-декодувальну архітектуру. Кодувальник складається з кодувальних шарів, що

оброблюють вхід ітеративно, шар за шаром, тоді як декодувальник складається з декодувальних шарів, які роблять те саме з виходом кодувальника.

Функцією кожного з кодувальних шарів є породжувати кодування, що містять інформацію про те, які частини входів є релевантними одна до одної. Він передає свої кодування наступному кодувальному шарові як входи. Кожен декодувальний шар робить протилежне, беручи всі ці кодування й використовуючи вбудовану до них контекстну інформацію, щоби породжувати послідовність виходу.Для цього кожен кодувальний та декодувальний шар використовує механізм уваги.

Для кожного входу увага зважує доречність кожного іншого входу та робить висновки з них, щоби виробляти вихід.Кожен декодувальний шар має додатковий механізм уваги, що дістає інформацію з виходів попередніх декодувальників перед тим, як цей декодувальник дістає інформацію з кодувань.

Як кодувальні, так і декодувальні шари мають нейронну мережу прямого поширення для додаткової обробки виходів, та містять за́лишкові з'єднання й кроки шарового унормовування (англ. layer normalization).

**Кодувальник.** Кожен кодувальник (англ. encoder) складається з двох головних складових: механізму самоуваги (англ. self-attention mechanism), та нейронної мережі прямого поширення. Механізм самоуваги приймає кодування входу з попереднього кодувальника, та зважує їхню релевантність одне одному, щоби породити кодування виходу. Нейронна мережа прямого поширення здійснює подальшу обробку кожного кодування виходу окремо. Ці кодування виходу відтак передають наступному кодувальникові як його вхід, так само як і декодувальникам.

Перший кодувальник отримує як вхід не кодування, а інформацію про положення та вкладення послідовності входу. Інформація про положення є необхідною трансформерові, щоби використовувати порядок послідовності, оскільки жодна інша частина трансформера його не використовує.

**Декодувальник.** Кожен декодувальник (англ. decoder) складається з трьох головних складових: механізму самоуваги, механізму уваги над кодуваннями, та нейронної мережі прямого поширення. Декодувальник працює подібно до кодувальника, крім вставленого додаткового механізму уваги, що натомість дістає релевантну інформацію з кодувань, породжених кодувальниками.

Подібно до першого кодувальника, перший декодувальник бере як свій вхід не кодування, а інформацію про положення та вкладення послідовності виходу. Трансформер мусить не використовувати для передбачування виходу поточний або майбутній вихід, тож послідовність виходу мусить бути частково приховано, щоби запобігти цьому зворотному потокові інформації. За крайнім декодувальником йде завершальне лінійне перетворення та шар softmax, щоби виробляти ймовірності виходу над словником.

**Тренування.** Трансформери, як правило, підлягають напівавтоматичному навчанню, що включає спонтанне попереднє тренування, з наступним керованим тонким налаштуванням. Попереднє тренування, як правило, виконують на більшому наборі даних, ніж тонке налаштування, через обмежену доступність мічених тренувальних даних. До задач попереднього тренування та тонкого налаштування зазвичай належать:

- моделювання мов
- передбачування наступного речення
- відповідання на питання
- розуміння прочитаного
- аналіз тональності
- перефразовування

**Застосування.** Трансформер досяг великого успіху в обробці природної мови (ОПМ), наприклад, в задачах машинного перекладу та передбачування часових рядів. Багато попередньо натренованих моделей, такі як GPT-2, GPT-3,

ChatGPT (GPT-3.5), BERT, XLNet та RoBERTa, демонструють здатність трансформерів виконувати широкий спектр таких пов'язаних з ОПМ задач, і мають потенціал знаходити застосування в реальному світі. До них можуть належати:

- машинний переклад
- реферування документів
- породжування документів
- розпізнавання іменованих сутностей (РІС)
- аналіз біологічних послідовностей
- розуміння відео.

2020 року було показано, що архітектуру Трансформер, а точніше GPT-2, може бути налаштовано для гри в шахи. Трансформери було застосовано до обробки зображень з результатами, конкурентноздатними у порівнянні зі згортковими нейронними мережами.

**Втілення.** Трансформерову модель було втілено в стандартних бібліотеках глибинного навчання, таких як TensorFlow та PyTorch.

Transformers є бібліотекою, що виробляє Hugging Face, яка пропонує архітектури на основі трансформерів та попередньо натреновані моделі.

## 2.4.TensorFlow

TensorFlow - це відкрите програмне забезпечення для машинного навчання та глибинного навчання, розроблене компанією Google Brain Team та випущене в 2015 році. Його призначення полягає у створенні, тренуванні та застосуванні

різноманітних моделей машинного навчання. TensorFlow забезпечує широкий спектр можливостей для створення моделей, від класифікації текстів до розпізнавання образів та глибинного навчання. Він дозволяє користувачам побудувати складні моделі з великою кількістю параметрів та ефективно тренувати їх на великих наборах даних. TensorFlow забезпечує інтерфейси для різноманітних мов програмування, таких як Python, C++, Java та інші, що робить його доступним для широкого кола розробників.

### 2.4.1. Історія розвитку TensorFlow

TensorFlow був розроблений командою Google Brain в 2015 році. Оригінальною метою була створення внутрішньої бібліотеки машинного навчання для вирішення проблем з обробкою великих об'ємів даних в Google. TensorFlow був випущений як відкрите програмне забезпечення під ліцензією Apache 2.0, що дозволило розробникам з усього світу використовувати його в своїх проектах. З часом TensorFlow став однією з найпопулярніших бібліотек машинного навчання та нейромереж, і зараз використовується в різних галузях, від наукових досліджень до розробки комерційних додатків.

У світі існує безліч відкритих фреймворків для машинного навчання, таких як PyTorch, Keras, Theano та інші. Однак, TensorFlow здобув домінуючу позицію завдяки своїм перевагам.

Перш за все, TensorFlow має велику підтримку від Google, що забезпечує стабільність та довгострокову підтримку. Крім того, TensorFlow має широкий функціонал, що дозволяє створювати складні моделі машинного навчання та робити розподілений обрахунок на різних пристроях, таких як графічні процесори та хмарні платформи.

Нарешті, TensorFlow має велику спільноту користувачів та розробників, що активно розвивається та підтримується. Це дозволяє швидко отримувати

відповіді на запитання та знаходити відповідні бібліотеки та рішення для різних задач машинного навчання.

Всі ці фактори забезпечили TensorFlow позицію лідера на ринку фреймворків машинного навчання.

### 2.4.2. Основні можливості TensorFlow

TensorFlow - це потужний фреймворк машинного навчання з відкритим вихідним кодом, який забезпечує велику кількість можливостей для розробки інноваційних моделей машинного навчання. Основною можливістю TensorFlow є побудова графів обчислень.

У TensorFlow граф обчислень - це граф з вузлами, що представляють математичні операції, а ребра - це дані, що передаються між вузлами. Це дозволяє ефективно використовувати ресурси та оптимізувати обчислення на графі.

Крім того, TensorFlow забезпечує можливість автоматичної диференціації, що дозволяє легко створювати складні моделі машинного навчання. Завдяки цьому можливо автоматично виводити градієнти функцій відносно вхідних даних та параметрів моделі.

Також TensorFlow має вбудовані інструменти для візуалізації графів обчислень, що дозволяє легко розуміти структуру моделі машинного навчання та відстежувати процес навчання.

Ще однією важливою можливістю TensorFlow є можливість обчислювати моделі на різних пристроях, таких як графічні процесори (GPU), центральні процесори (CPU) та Tensor Processing Units (TPU). Це дозволяє забезпечити ефективність обчислень та швидкість розробки моделей машинного навчання.

Отже, TensorFlow - це потужний інструмент для розробки моделей машинного навчання з великою кількістю можливостей для побудови графів

обчислень, автоматичної диференціації, візуалізації графів та обчислення на різних пристроях.

### 2.4.2. Використання TensorFlow в машинному навчанні

TensorFlow став одним з найпопулярніших фреймворків для машинного навчання завдяки своїм потужним можливостям і зручному інтерфейсу. Він дозволяє легко будувати і оптимізувати складні моделі машинного навчання за допомогою графів обчислень.

Зокрема, TensorFlow дозволяє використовувати глибокі нейронні мережі для розв'язання складних завдань, таких як класифікація зображень, розпізнавання мови або робота з великими обсягами даних. Крім того, фреймворк надає можливість автоматичної диференціації, що спрощує процес оптимізації моделей.

Також TensorFlow підтримує візуалізацію графів обчислень, що дозволяє бачити структуру моделі і відстежувати процес її навчання. Крім того, фреймворк підтримує обчислення на різних пристроях, включаючи графічні процесори, що забезпечує більшу швидкість роботи.

У загальному, TensorFlow дозволяє зручно і ефективно використовувати складні алгоритми машинного навчання для розв'язання різноманітних завдань в різних галузях, від комп'ютерного зору до обробки природних мов.

TensorFlow є одним з найпопулярніших інструментів для машинного навчання, який використовують в різних областях. Нижче наведені деякі приклади використання TensorFlow в машинному навчанні.

1. Класифікація зображень: TensorFlow може використовуватись для класифікації зображень, наприклад, для розпізнавання об'єктів на зображеннях. За допомогою TensorFlow можна побудувати нейронну мережу, яка буде класифікувати зображення з високою точністю.

2. Сегментація зображень: TensorFlow також може використовуватись для сегментації зображень, тобто розділення зображення на окремі сегменти або об'єкти. Це можна використовувати, наприклад, для розпізнавання меж об'єктів на зображеннях.
3. Глибоке навчання: TensorFlow є одним з основних інструментів для роботи з глибоким навчанням. Він може використовуватись для навчання глибоких нейронних мереж, які можуть виконувати складні завдання, такі як розпізнавання мови або обробка природних мов.
4. Генерація зображень: TensorFlow можна використовувати для генерації нових зображень за допомогою глибоких нейронних мереж. Наприклад, він може бути використаний для створення нових зображень людей або тварин.
5. Обробка мовного тексту: TensorFlow можна використовувати для розпізнавання та обробки природних мов, таких як аналіз тональності тексту або автоматичний переклад.
6. Рекомендації: TensorFlow може використовуватись для створення систем рекомендацій, наприклад, для рекомендацій покупок або рекомендацій фільмів.

Ще одним прикладом використання TensorFlow є навчання рекурентних нейронних мереж (RNN) для обробки послідовностей даних, таких як мовлення та текст. RNN можуть використовуватись для машинного перекладу, заснованого на послідовному аналізі слів або фраз. TensorFlow також може використовуватись для навчання засобів управління, таких як нейронні мережі, що керують роботами або автономними автомобілями.

Загалом, TensorFlow є потужним інструментом для машинного навчання та інших областей, які вимагають обробки великих обсягів даних. Він має багато корисних функцій і може використовуватись на різних платформах.

### 2.4.3. TensorFlow в інших областях

TensorFlow може бути використаний не тільки в області машинного навчання, але й в інших областях. Наприклад, в обробці природних мов, TensorFlow може бути використаний для завдань, таких як машинний переклад, генерація тексту, розпізнавання мови та аналізу настроїв тексту.

У сфері обробки звуку, TensorFlow може бути використаний для задач, таких як розпізнавання мови, звукові ефекти та генерація музики. TensorFlow має спеціальні інструменти для обробки аудіоданих, такі як обрізка, масштабування та перетворення звукових сигналів.

TensorFlow також може бути використаний для обробки даних, таких як обробка зображень, розпізнавання облич, рекомендації та аналіз даних. Зокрема, TensorFlow має спеціальні інструменти для роботи з табличними даними та іншими форматами даних, які дозволяють використовувати його для завдань, пов'язаних з обробкою даних.

Загалом, TensorFlow є дуже гнучким фреймворком, який може бути використаний в багатьох різних областях, пов'язаних з обробкою даних та машинним навчанням. Це робить його одним з найбільш популярних інструментів у своєму класі.

### 2.4.4. Застосування

Загальною характеристикою TensorFlow є його відкритість і гнучкість. TensorFlow надає можливість побудови графів обчислень, автоматичної диференціації, візуалізації графів, обчислення на різних пристроях та багато іншого.

TensorFlow знаходить широке використання в різних областях машинного навчання, таких як класифікація, сегментація зображень, глибоке навчання, обробка природних мов, обробка звуку та інше. TensorFlow використовується у

різних проектах, від великих корпорацій до невеликих стартапів, що свідчить про його потужність та гнучкість.

TensorFlow продовжує активно розвиватися, включаючи розширення функціоналу, підтримку нових архітектур та пристроїв, поліпшення продуктивності та т.д. Однією з основних перспектив розвитку TensorFlow є збільшення швидкості навчання моделей та оптимізація використання ресурсів. Також, TensorFlow може бути використаний в багатьох областях, таких як медицина, фінанси, автомобільна промисловість, реклама, ігрова індустрія та інші, що дає можливість його широкого використання в різних галузях. З огляду на постійний розвиток машинного навчання та інших технологій, використання TensorFlow та інших фреймворків машинного навчання стане ще більш важливим в майбутньому.

## 2.5. PyTouch

PyTorch - це відкрите фреймворк машинного навчання, який базується на мові програмування Python. Він розроблений з метою спростити створення та навчання нейронних мереж та інших моделей глибокого навчання. PyTorch надає зручні інструменти для обробки даних, побудови моделей, розрахунку градієнтів та оптимізації параметрів моделі. Його основними перевагами є динамічний обчислювальний граф, що дозволяє зручно виконувати операції з тензорами та автоматично обчислювати градієнти, а також багата колекція наборів даних та популярних алгоритмів для завдань машинного навчання. PyTorch є одним з найпопулярніших інструментів для дослідження та розробки моделей глибокого навчання завдяки своїй зручній інтерфейсу та активній спільноті користувачів.

### 2.5.1. Історія розвитку PyTouch

PyTorch розвивається відкритою командою дослідників та розробників з Facebook AI Research (FAIR). Перша версія фреймворку була випущена в січні

2017 року і відразу здобула популярність серед науково-дослідницької спільноти та спеціалістів з глибокого навчання.

PyTorch був розроблений як наступник фреймворку Torch, який був популярним серед дослідників з машинного навчання. Однак, відмінність PyTorch полягає у його інтеграції з Python, що робить його більш доступним та зручним для розробників.

У 2018 році PyTorch отримав значні покращення з випуском версії 1.0. Введення динамічного обчислювального графа було основним оновленням, що дозволило розробникам використовувати ізоляцію графу під час виконання, що дозволяє більшу гнучкість та зручність при розробці моделей. Крім того, було додано підтримку для виконання на графічних процесорах (GPU) та розподілених обчислень.

У наступні роки PyTorch продовжував активно розвиватись та отримувати нові функції та покращення. Версія 1.6, випущена в 2020 році, додала підтримку для статичного графу, що спрощує процес оптимізації та ефективного виконання моделей.

PyTorch також активно сприяє розвитку екосистеми інструментів та бібліотек. У 2018 році була запущена бібліотека torchvision, яка надає набір утиліт для роботи з комп'ютерним зором та обробки зображень. Крім того, PyTorch підтримує інтеграцію з популярними фреймворками для автоматичного диференціювання, такими як Autograd, і забезпечує інтерфейс для виконання обчислень на різних пристроях, включаючи мобільні пристрої та мікроконтролери.

З часом PyTorch став одним з найпопулярніших інструментів для дослідження та розробки моделей глибокого навчання. Його активна спільнота розробників та дослідників продовжує вносити нові ідеї та розширювати можливості фреймворку. PyTorch використовується в академічних

дослідженнях, промислових проектах та конкурсах з машинного навчання, що підтверджує його значну популярність та вплив на галузь штучного інтелекту.

### 2.5.2. Основні можливості PyTouch

PyTorch надає розширений набір можливостей для розробки моделей глибокого навчання. Основні можливості PyTorch включають:

1. Тензорний обчислення: PyTorch надає потужні інструменти для роботи з тензорами, що дозволяють виконувати операції лінійної алгебри, маніпуляції та трансформації даних, а також розрахунок градієнтів. Тензори в PyTorch сумісні з NumPy, що спрощує обмін даними з іншими бібліотеками та фреймворками.
2. Автоматичне диференціювання: PyTorch автоматично обчислює градієнти для параметрів моделі за допомогою вбудованої системи автоматичного диференціювання. Це дозволяє використовувати методи оптимізації, такі як стохастичний градієнтний спуск (SGD), для тренування моделей.
3. Модельний API: PyTorch має інтуїтивно зрозумілий та зручний API для побудови моделей глибокого навчання. Він дозволяє легко створювати та налаштовувати складні нейронні мережі, використовуючи різні типи шарів, активаційні функції та регуляризацію.
4. Вбудовані набори даних та завдань: PyTorch надає багато наборів даних для розробки моделей глибокого навчання, таких як torchvision для комп'ютерного зору, torchaudio для обробки аудіо та інші. Це спрощує завантаження та підготовку даних для тренування та тестування моделей.
5. Розподілене обчислення: PyTorch підтримує розподілене обчислення, що дозволяє розподіляти навантаження між багатьма пристроями, такими як графічні процесори (GPU) або множина серверів. Це дозволяє прискорити обчислення та тренування моделей на великих обсягах даних.

6. Інтеграція з іншими бібліотеками: PyTorch може взаємодіяти з іншими популярними бібліотеками для наукових обчислень та машинного навчання, такими як NumPy, SciPy та scikit-learn. Це дозволяє використовувати різноманітні інструменти для обробки та аналізу даних в контексті моделей глибокого навчання.
7. Статистика та візуалізація: PyTorch надає інструменти для візуалізації та аналізу результатів тренування моделей. Інтеграція з бібліотеками, такими як Matplotlib та TensorBoard, дозволяє відстежувати метрики, відображати графіки та зображення, що допомагає зрозуміти та налаштувати моделі.

Ці можливості роблять PyTorch потужним інструментом для розробки, тренування та впровадження моделей глибокого навчання в різних сферах, включаючи комп'ютерний зір, обробку природної мови, рекомендаційні системи та багато інших.

### 2.5.3. Використання PyTorch в машинному навчані

PyTorch є одним з найпопулярніших фреймворків для машинного навчання і використовується в широкому спектрі задач та доменів. Деякі з основних використань PyTorch в машинному навчанні включають:

1. Розробка та тренування нейронних мереж: PyTorch надає потужні інструменти для розробки та настройки нейронних мереж. Розробники можуть використовувати різні типи шарів, активаційні функції та оптимізатори для створення та тренування моделей з високою точністю.

2. Комп'ютерний зір: PyTorch має багато вбудованих функцій та наборів даних для роботи з комп'ютерним зором. Він широко використовується для завдань розпізнавання образів, семантичного сегментації, об'єктного виявлення та генерації зображень.

3. Обробка природної мови (NLP): PyTorch надає інструменти для обробки текстових даних та розробки моделей для завдань NLP, таких як машинний переклад, виявлення емоцій, сентимент-аналіз та генерація тексту. Він підтримує роботу зі збірками слів, ембеддингами слів та рекурентними нейронними мережами.

4. Рекомендаційні системи: PyTorch може бути використаний для розробки рекомендаційних систем, які допомагають прогнозувати та рекомендувати продукти, фільми, музику тощо. За допомогою нейронних мереж, розробники можуть створити моделі, які аналізують поведінку користувачів та знаходять особисті рекомендації.

5. Поодинокі задачі: PyTorch може бути використаний для вирішення інших типів задач машинного навчання, таких як кластеризація, класифікація, генерація тексту, регресія та багато інших. Завдяки гнучкості та розширюваності PyTorch, розробники можуть адаптувати фреймворк до своїх конкретних потреб.

Це лише кілька прикладів використання PyTorch в машинному навчанні, але фреймворк може бути застосований в різних галузях та задачах. Його популярність серед дослідників та розробників продовжує зростати завдяки його простоті використання, потужним інструментам та активній спільноті.

### 2.5.4. Можливості використання PyTouch у аналізі емоційного забарвлення текстів

PyTorch має декілька можливостей, які можуть бути використані в аналізі емоційного забарвлення текстів. Деякі з цих можливостей включають:

1. Будівництво моделей емоційного аналізу: За допомогою PyTorch можна побудувати нейронні мережі, які аналізують емоційне забарвлення текстів. Можна використовувати рекурентні нейронні мережі (RNN), зокрема LSTM або

GRU, для розпізнавання емоцій в текстах. Потім модель може бути навчена на підготовленому наборі даних з етикетованими емоціями.

2. Попереднє навчання ембедінгів слів: PyTorch надає можливість попереднього навчання ембедінгів слів на великому обсязі текстових даних. Це може поліпшити якість моделі емоційного аналізу, оскільки модель буде мати краще уявлення про семантику слів та їх відношення.

3. Постановка задачі багатокласової класифікації: Аналіз емоцій може бути сформульований як задача багатокласової класифікації, де кожен текст класифікується за певними емоціями (наприклад, щастя, сум, гнів і т.д.). PyTorch надає зручний інтерфейс для побудови моделей класифікації з використанням різних архітектур, наприклад, згорткових нейронних мереж (CNN) або трансформерів.

4. Генерація векторних представлень текстів: PyTorch також надає інструменти для генерації векторних представлень текстів, таких як BERT або GPT. Ці моделі можуть бути використані для відповідної обробки тексту, а потім передані до моделі емоційного аналізу для отримання більш точних результатів.

5. Построєння архітектури з використанням attention механізму: Attention механізм може бути використаний для підвищення якості моделі емоційного аналізу, дозволяючи моделі зосередитись на важливих частинах тексту, що впливають на емоційне забарвлення. PyTorch надає різні модулі та функції для побудови архітектур з використанням attention механізму.

Використання PyTorch в аналізі емоційного забарвлення текстів дозволяє розробникам ефективно будувати та навчати моделі для виявлення та класифікації емоцій в текстах.

## РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ

Програмний застосунок оцінки емоцій в тексті коментарів є потужним інструментом, який дозволяє автоматично аналізувати та оцінювати емоційну семантику текстових коментарів. Завдяки поєднанню машинного навчання та обробки природної мови, цей застосунок допомагає розуміти тон та настрій, що виражаються у коментарях, надаючи корисні інсайти для аналізуваної аудиторії.

З ростом популярності соціальних медіа та форумів великі обсяги текстових коментарів з'явилися як ніколи раніше. Аналізувати цю величезну кількість даних вручну стало неможливою задачею для людей. Тут на допомогу приходять програмні рішення, які використовують алгоритми машинного навчання для автоматичної обробки тексту та виявлення емоційних відтінків.

Програмний застосунок оцінки емоцій в тексті коментарів базується на використанні попередньо навчених моделей, зокрема моделі BERT (Bidirectional Encoder Representations from Transformers). BERT є однією з найефективніших моделей для розуміння природної мови і відтворення контексту та залежностей між словами. Використовуючи цю модель та навчену на великій кількості даних, програма здатна автоматично класифікувати коментарі на позитивні та негативні, визначати ймовірності емоцій та надавати оцінку емоційного відтінку кожного коментаря.

Цей застосунок має широкі можливості застосування. Він може бути корисним для аналітиків, маркетологів, модераторів соціальних мереж та всіх, хто зацікавлений в аналізі та сприйнятті емоційного контексту коментарів. За його допомогою можна виявити та відстежувати настрій груп користувачів, оцінювати ефективність маркетингових кампаній, виявляти популярні теми або проблеми, а також реагувати на негативний контент в реальному часі.

У цій розробленій програмі використовуються передові технології, такі як бібліотеки PyQt5, sqlite3, torch та transformers, що забезпечують потужну функціональність та ефективність обробки тексту. Завдяки цим інструментам програмний застосунок оцінки емоцій в тексті коментарів надає користувачам зручний інтерфейс, швидку обробку даних та точні результати.

Загальне застосування програмного застосунку оцінки емоцій в тексті коментарів відкриває шлях до нових можливостей в аналізі емоційної складової текстового контенту. Завдяки автоматичній обробці та оцінці емоцій коментарів, користувачі можуть отримати цінні уявлення про реакції та настрій аудиторії, що є важливим фактором в сучасному інформаційному світі.

Програма використовує наступні бібліотеки та фреймворки:

PyQt5: PyQt5 є Python-інтерфейсом для бібліотеки Qt, яка розробляється компанією The Qt Company. Qt є потужним інструментарієм для розробки крос-платформенних програм з графічним інтерфейсом. PyQt5 створено Riverbank Computing Limited і представляє собою обгортку навколо Qt для використання в мові програмування Python. Вона надає багатофункціональні можливості для розробки GUI, включаючи створення вікон, діалогових вікон, кнопок, полів введення та інших елементів інтерфейсу.

sqlite3: Бібліотека sqlite3 є частиною стандартної бібліотеки Python. Вона надає зручний спосіб взаємодії з базами даних SQLite, які зберігають дані в локальному файлі. SQLite є легковажним, серверним інструментом управління базами даних, який не вимагає окремого сервера. Використання бібліотеки sqlite3 дозволяє зчитувати дані з бази даних та виконувати запити для отримання необхідної інформації.

torch: Бібліотека torch є фреймворком для машинного навчання і глибокого навчання, який спеціалізується на обчисленнях на графічних процесорах. Вона надає потужні інструменти для створення та тренування нейронних мереж. Torch був створений для забезпечення зручного і ефективного інтерфейсу для розробки алгоритмів машинного навчання. Вона має багато модулів і функцій для роботи з даними, обчисленнями, нейронними мережами та іншими аспектами машинного навчання.

transformers: Бібліотека transformers є набором інструментів для роботи з нейронними мережами з архітектурою трансформера. Вона розроблена

компанією Hugging Face і є однією з найпопулярніших бібліотек для обробки природної мови (NLP). Transformers надає доступ до попередньо навчених моделей, включаючи моделі трансформера BERT, які досягають високої точності в завданнях класифікації тексту. Вона також містить інструменти для токенізації тексту, роботи з послідовністю токенів та іншими завданнями NLP.

Опис програми:

Ця програма реалізує аналіз емоцій коментарів за допомогою попередньо навченої моделі BERT (distilbert-base-uncased-finetuned-sst-2-english) і токенізатора. Вона дозволяє користувачеві ввести коментар для аналізу та отримати результати оцінки емоцій для даного коментаря.

Програма використовує наступні бібліотеки та фреймворки:

- PyQt5: для реалізації графічного інтерфейсу користувача.
- sqlite3: для зчитування коментарів з бази даних SQLite.
- torch: для роботи з моделлю BERT та обчислення оцінок емоцій.
- transformers: для завантаження попередньо навченої моделі BERT та токенізатора.

Клас CommentSelectionWindow відповідає за вікно вибору коментаря, в якому користувач може вибрати один з завантажених коментарів.

Клас MainWindow є головним вікном програми і містить графічний інтерфейс, який складається з поля для введення коментаря, поля для відображення результату оцінки емоцій та бічного меню. Він також має методи для оновлення коментарів з бази даних, вибору коментаря, аналізу емоцій окремого коментаря та аналізу емоцій всіх коментарів.

Для аналізу емоцій окремого коментаря використовується функція analyze_emotions, яка перетворює коментар на послідовність токенів, кодує його за допомогою токенізатора і передає його до моделі BERT для отримання оцінок емоцій. Результат оцінки відображається в графічному інтерфейсі.

Функція analyze_all_comments аналізує емоції всіх коментарів, які були завантажені з бази даних. Вона обробляє кожен коментар окремо, використовуючи ті ж кроки, що і в функції analyze_emotions. Результати оцінок для 5 найкращих та 5 найгірших коментарів відображаються в графічному інтерфейсі.

Ця програма дозволяє користувачеві швидко аналізувати емоції коментарів і отримувати результати в зручному форматі.

У програмному коді, пов'язаному з програмним застосунком оцінки емоцій в тексті коментарів, ми можемо знайти різні функції та класи, кожен з яких виконує певні завдання для забезпечення функціональності програми. Ось опис деяких з них:

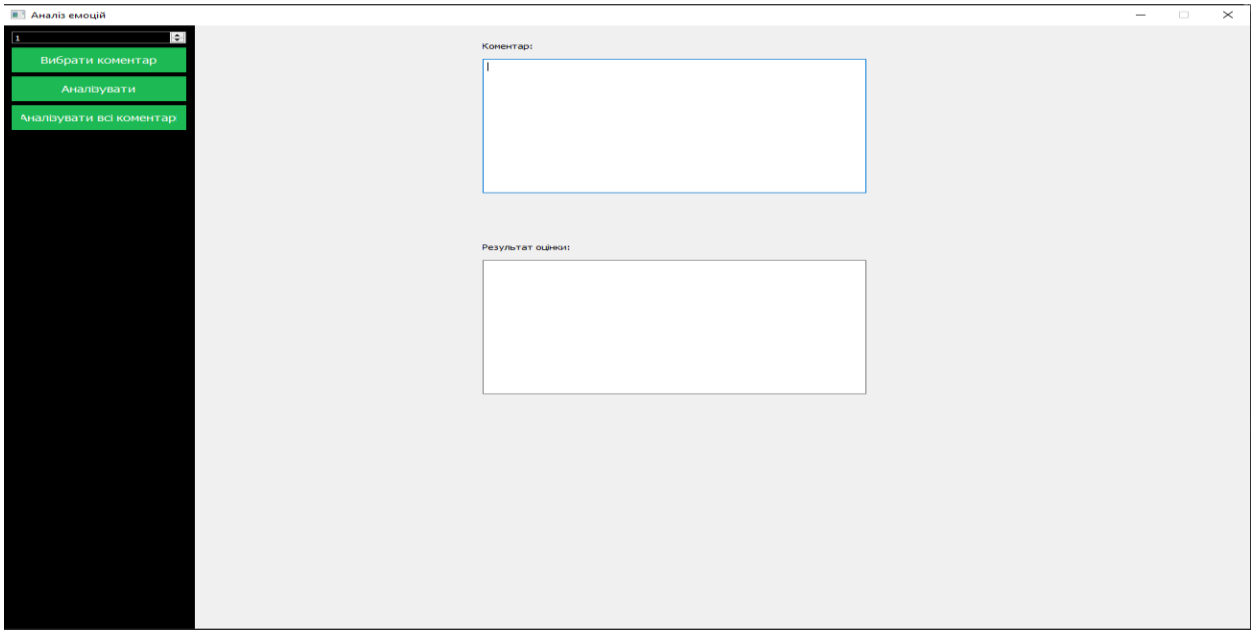
1. Клас EmotionAnalyzer: Цей клас є основним модулем, який відповідає за аналіз емоцій в тексті коментарів. Він містить методи та функції для обробки тексту, використовуючи попередньо навчену модель BERT та токенізатор.
2. Клас EmotionAnalyzer приймає на вхід коментарі та повертає результати аналізу емоцій, такі як класифікація на позитивний або негативний, або вірогідність різних емоцій.
3. Клас DatabaseHandler: Цей клас відповідає за зчитування коментарів з бази даних SQLite. Він має методи для з'єднання з базою даних, виконання запитів і отримання коментарів. Клас DatabaseHandler забезпечує зручний спосіб отримання даних з бази даних для подальшої обробки.
4. Функція GUI_MainWindow: Ця функція відповідає за графічний інтерфейс користувача програми. Вона використовує бібліотеку PyQt5 для створення вікна програми, кнопок, текстових полів тощо. Функція GUI_MainWindow встановлює зв'язок між графічним інтерфейсом та рештою функціоналу програми.
5. Функція preprocess_text: Ця функція використовується для попередньої обробки тексту перед подальшою обробкою. Вона може включати такі

операції, як видалення зайвих символів, нормалізація тексту, лематизація тощо. Функція preprocess_text готує текст для подальшого аналізу емоцій.

6. Клас ModelLoader: Цей клас відповідає за завантаження попередньо навченої моделі BERT та токенізатора за допомогою бібліотеки transformers. Він має методи для завантаження моделі та токенізатора з відповідних файлів. Клас ModelLoader забезпечує доступ до моделі BERT для аналізу емоцій.

## 3.1. Графічний інтерфейс програми

Головне вікно програми є графічним інтерфейсом користувача, яке надає зручний спосіб взаємодії з програмою оцінки емоцій в тексті коментарів. Основна мета цього вікна - надати користувачеві можливість ввести текст коментарів, а також отримати результати аналізу емоцій для цих коментарів.




Давайте детальніше розглянемо складові цього головного вікна:

1. Поле введення коментаря: В цьому текстовому полі користувач може ввести текст коментаря, для якого він бажає отримати аналіз емоцій. Воно знаходиться у верхній частині вікна і має розміщення та розміри, які забезпечують зручне введення тексту.
2. Кнопка "Аналізувати": Після введення коментаря користувач може натиснути цю кнопку для запуску процесу аналізу емоцій. Після натискання кнопки програма виконає обробку тексту коментаря та покаже результати аналізу.
3. Результати оцінки емоцій: Це текстове поле, яке відображає результати аналізу емоцій для введеного коментаря. Тут буде вказано, чи є коментар позитивним або негативним, а також надана вірогідність різних емоцій. Результати відображаються після натискання кнопки "Аналізувати" або відповідного події.
4. Бічне меню: Головне вікно може містити бічне меню з додатковими опціями та функціями. Наприклад, це може бути вибір кількості коментарів для аналізу, кнопка для аналізу всіх коментарів або кнопка для вибору коментарів з бази даних. У коді, який ви надали, бічне меню містить такі елементи:
    - Поле введення кількості коментарів (QSpinBox): Це поле дозволяє користувачеві встановити кількість коментарів, які будуть проаналізовані. Користувач може вибрати мінімальне та максимальне значення для кількості коментарів.
    - Кнопка "Вибрати коментар" (QPushButton): При натисканні цієї кнопки програма викликає метод select_comment батьківського вікна. Цей метод може відкрити інше вікно для вибору коментарів з бази даних або здійснити інші дії, пов'язані з вибором коментаря.
    - Кнопка "Аналізувати" (QPushButton): При натисканні цієї кнопки програма викликає метод analyze_emotions батьківського вікна. Цей метод запускає процес аналізу емоцій для введеного коментаря.

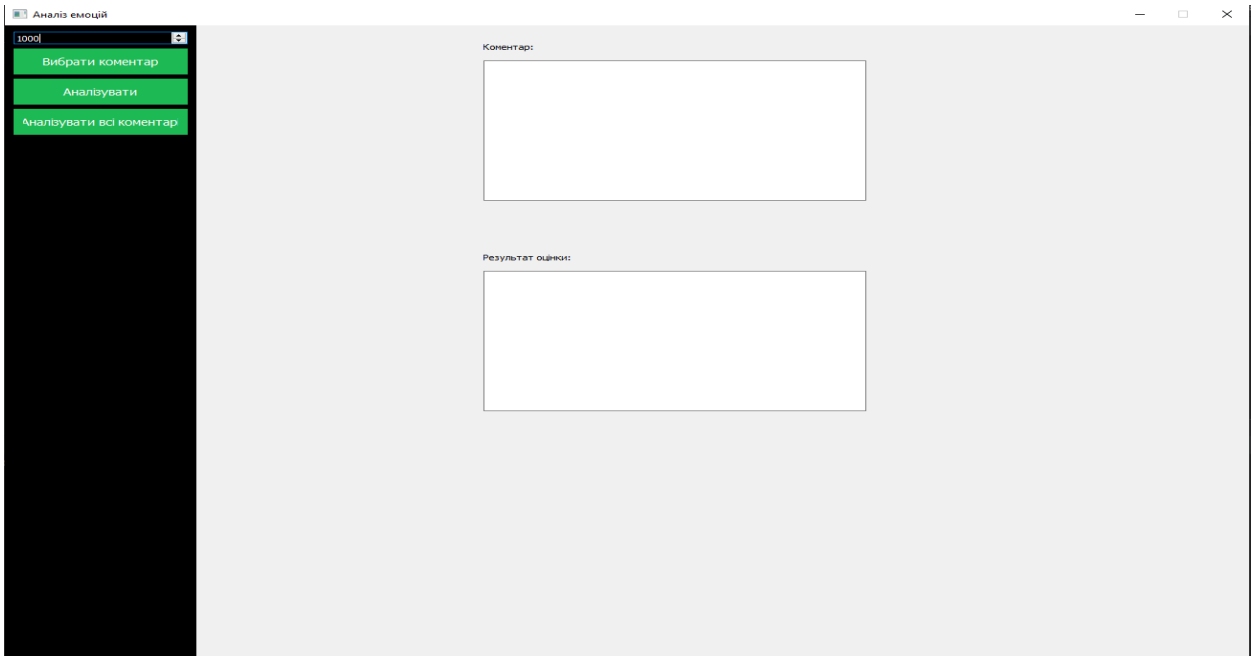
- Кнопка "Аналізувати всі коментарі" (QPushButton): При натисканні цієї кнопки програма викликає метод analyze_all_comments батьківського вікна. Цей метод запускає процес аналізу емоцій для всіх коментарів в базі даних.

Бічне меню забезпечує додатковий функціонал та можливості для користувача, які полегшують взаємодію з програмою. Він доповнює основний функціонал головного вікна і дозволяє користувачеві здійснювати різні дії, пов'язані з аналізом коментарів.

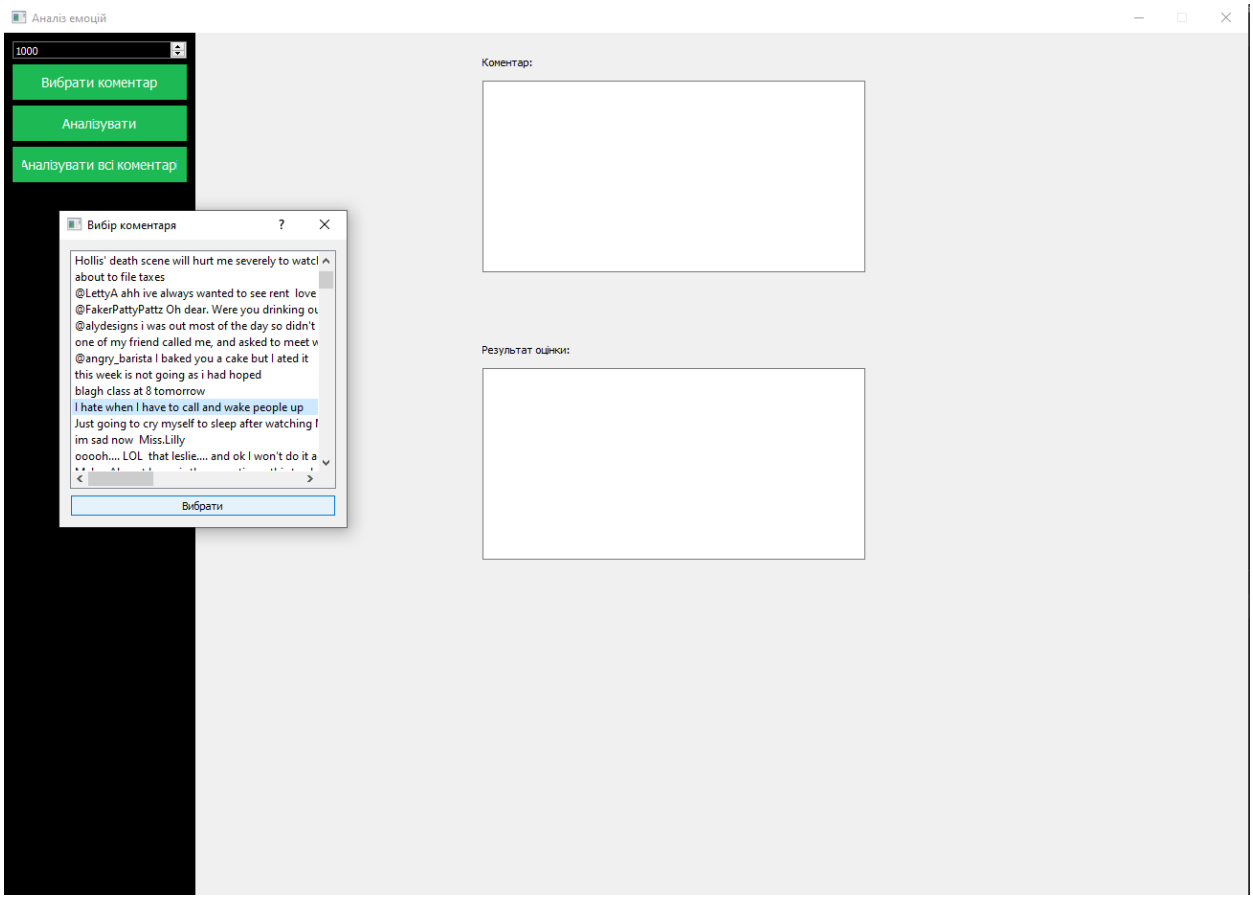
## 3.2. Результат роботи програми

Для того, щоб здійснити оцінку коментаря за допомогою програми, користувач повинен виконати наступні дії:

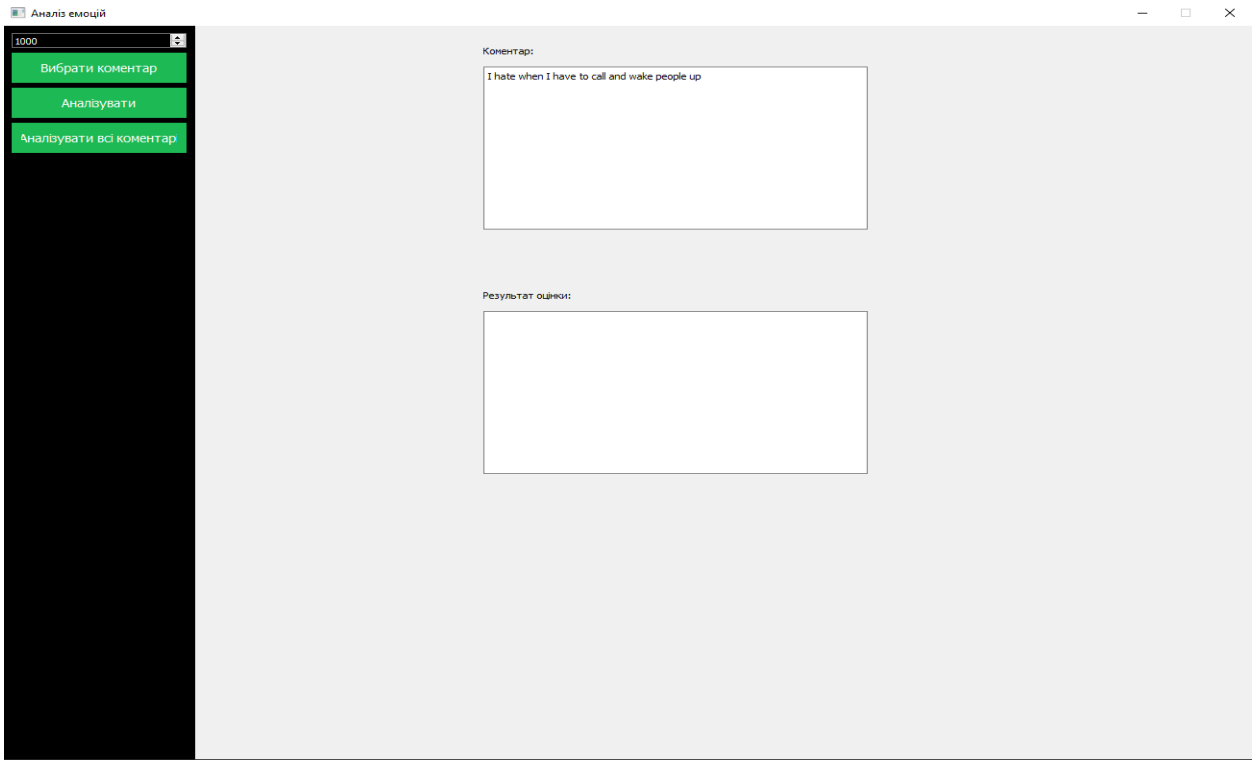
1. Відкрити головне вікно програми. Головне вікно є графічним інтерфейсом користувача і надає зручний спосіб взаємодії з програмою оцінки емоцій в тексті коментарів.
2. В полі "Кількість коментарів" користувач вводить бажану кількість коментарів з бази даних, які він бажає оцінити.

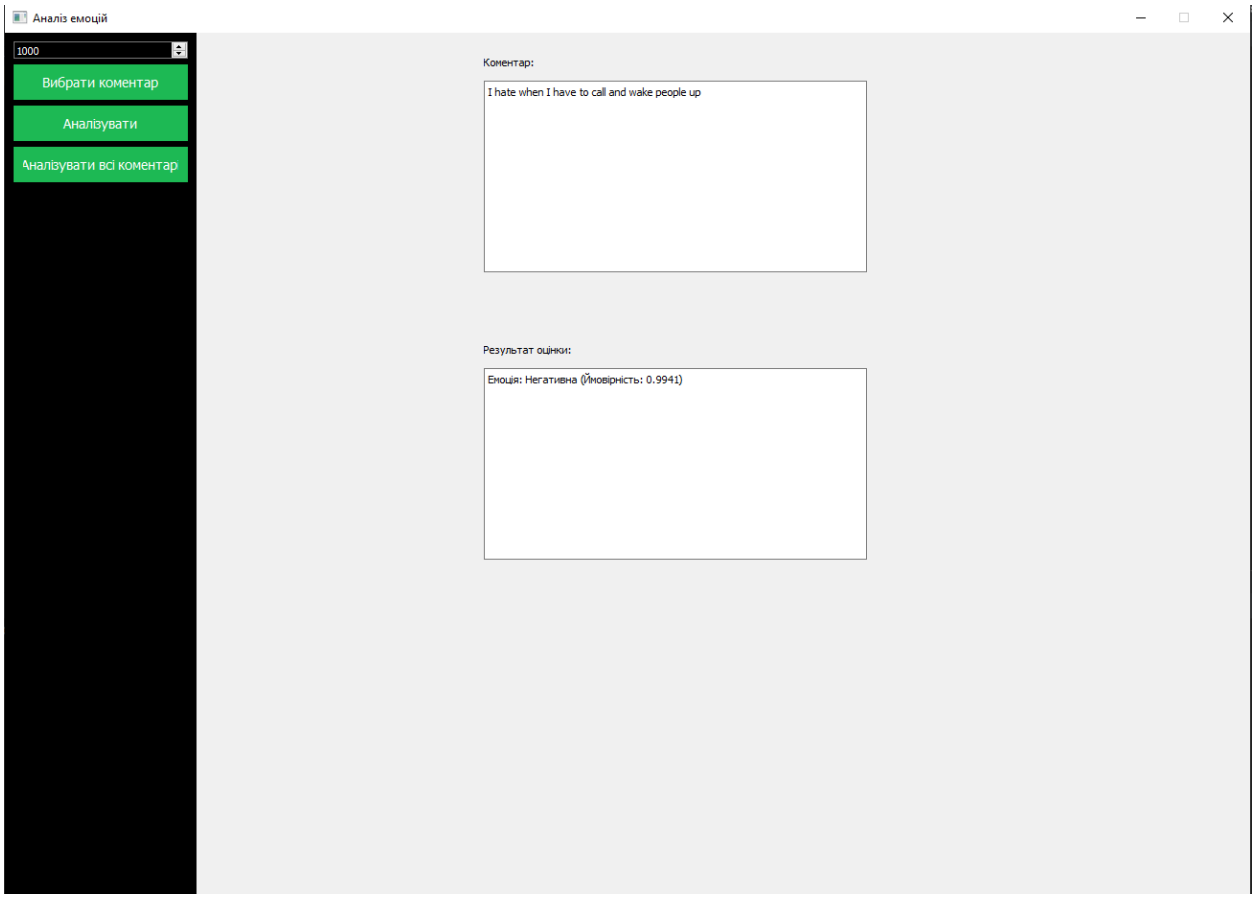


3. Після введення кількості коментарів користувач натискає кнопку "Вибрати коментарі". Ця кнопка знаходиться на бічному меню головного вікна і відкриває вікно для вибору коментарів з бази даних.
4. У вікні вибору коментарів користувач обирає бажані коментарі, зазвичай, за допомогою прапорців або позначок. Після вибору коментарів користувач натискає кнопку "Вибрати" або подібну, щоб підтвердити свій вибір.




5. Після вибору коментарів користувач повинен натиснути кнопку "Аналізувати". Ця кнопка знаходиться на головному вікні програми і запускає процес аналізу емоцій для вибраних коментарів з бази даних та отримує результат оцінки в полі «Результат оцінки»

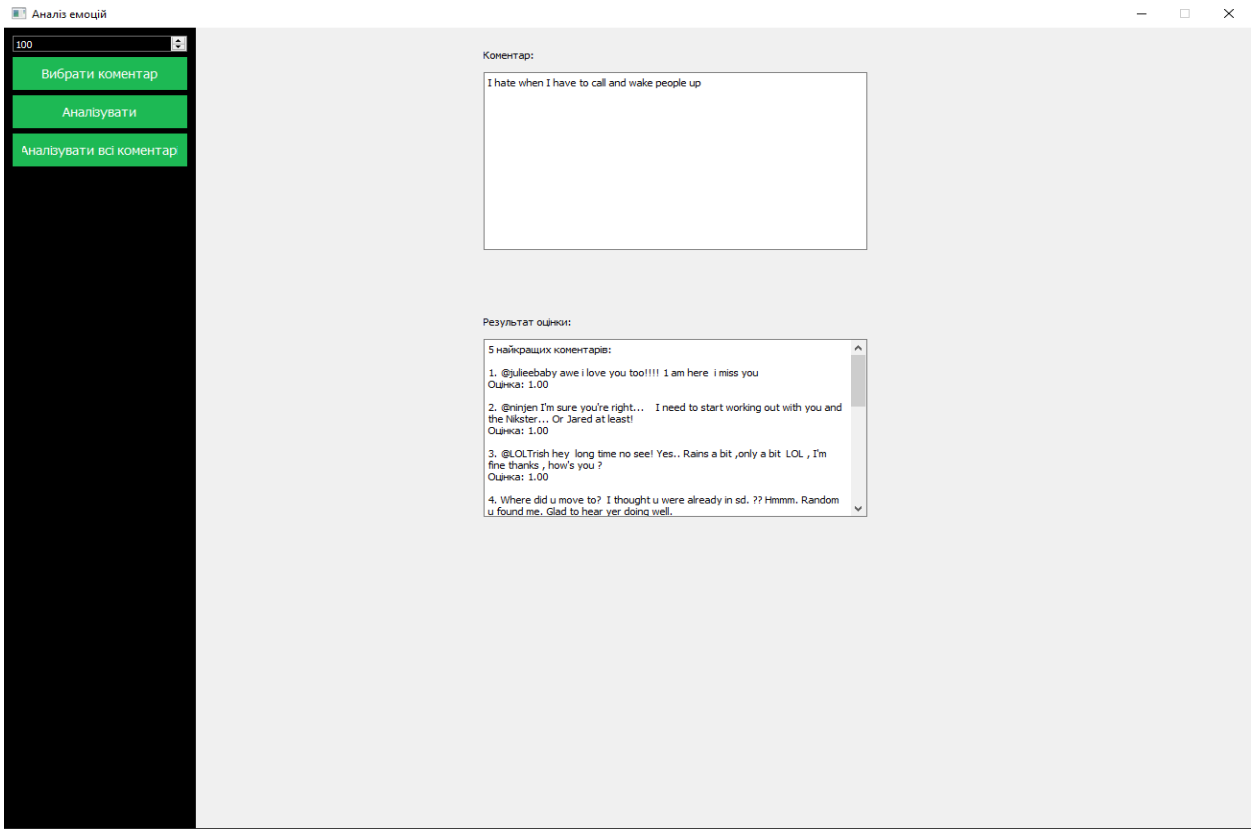



6. Якщо користувач бажає проаналізувати 5 найкращих і 5 найгірших коментарів, він повинен ввести кількість коментарів з бази даних, які

будуть враховані у цьому аналізі. Після введення кількості коментарів користувач натискає кнопку "Аналізувати всі коментарі".

7. Після натискання кнопки "Аналізувати всі коментарі" програма починає обробку тексту коментарів та проводить аналіз емоцій для кожного коментаря окремо. Результати оцінки емоцій для кожного коментаря відображаються в текстовому полі на головному вікні або зберігаються в базі даних для подальшого використання.




Отже, послідовні дії користувача для оцінки коментарів з бази даних будуть наступними: відкриття головного вікна програми, введення кількості коментарів, натискання кнопки "Вибрати коментарі" для вибору коментарів, натискання кнопки "Аналізувати" для аналізу вибраних коментарів, або введення кількості коментарів для проаналізування 5 найкращих і 5 найгірших коментарів, та натискання кнопки "Аналізувати всі коментарі".

**Висновок до розділу**

В цьому програмному застосунку для оцінки емоцій в текстових коментарях ми використовували модель BERT (Bidirectional Encoder Representations from Transformers). BERT є однією з найпотужніших моделей для розуміння природної мови та аналізу тексту. Вона базується на архітектурі трансформерів, що дозволяє моделі ефективно моделювати контекст та взаємозв'язки між словами у тексті.

Модель BERT має велику кількість попередньо навчених параметрів, які дозволяють їй розуміти семантику та емоційний зміст тексту. Вона здатна класифікувати текст за емоційними категоріями, такими як позитивний, негативний а також визначати ймовірність різних емоцій.

Застосування моделі BERT дозволяє програмному застосунку точно та надійно аналізувати емоційний відтінок коментарів і надавати користувачеві цінні відомості про емоційний контекст тексту. Вона є одним з ключових елементів, що забезпечують високу якість та точність результатів аналізу емоцій у програмі.

Головне вікно програми є графічним інтерфейсом користувача, яке надає зручний спосіб взаємодії з програмою оцінки емоцій в тексті коментарів. Основна мета цього вікна - надати користувачеві можливість ввести текст коментарів, а також отримати результати аналізу емоцій для цих коментарів.