Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Кафедра теоретичної кібернетики

**Кваліфікаційна робота**

**на здобуття ступеня бакалавра**

за спеціальністю 122 «Комп’ютерні науки»

на тему:

# Виявлення та дослідження сарказму в текстових повідомленнях

Студентки 4-го курсу
Прокопенко Юлії Сергіївни

_____
(підпис)

Науковий керівник:
професор, доктор фізико-математичних наук
Крак Юрій Васильович

_____
(підпис)

Робота заслухана на засіданні кафедри теоретичної кібернетики та рекомендована до захисту в ЕК, протокол № …… від ..... 2021р.

Завідувач кафедри Крак Ю.В.

Київ – 2021

# РЕФЕРАТ

Робота складається зі вступу, 3 розділів, висновків, списку використаних джерел (37 найменувань). Робота містить 14 рисунків. Загальний обсяг становить 51 сторінок, основний текст роботи викладено на 47 сторінках.

Ключові слова: ОБРОБКА ПРИРОДНОЇ МОВИ, РОЗПІЗНАВАННЯ САРКАЗМУ, КЛАСИФІКАЦІЯ ТЕКСТІВ, СОЦІАЛЬНІ МЕРЕЖІ, СЕНТИМЕНТ АНАЛІЗ, ПОПЕРЕДНЯ ОБРОБКА ТЕКСТІВ,

Об’єктом дослідження є повідомлення, коментарі та інші види текстових публікації в соціальних мережах.

У якості інструменту створення програмного сервісу було обрано PyCharm - інтегроване середовище розробки мовою Python. Також були використані наступні бібліотеки мови програмування Python: Keras, Scrapy, Flair, PyTorch.

Метою кваліфікаційної роботи є дослідження наявних методів визначення сарказму в текстових повідомленнях та розробка власної моделі визначення сарказму на обраному наборі даних.

Методи розроблення: аналіз методів формального представлення текстів, аналіз методів попередньої обробки текстових даних, аналіз існуючих методів визначення сарказму в текстах, аналіз існуючих методів визначення тональності текстів, порівняння доступних наборів розмічених даних, розробка різних архітектур текстових класифікаторів, візуалізація та аналіз отриманих результатів.

Результати роботи: виконано загальний огляд вже існуючих досліджень, проведено порівняльний аналіз доступних наборах розмічених даних, розв’язана поставлена задача, розроблені, натреновані та порівняні нейронні мережі на основі популярних архітектур трансформерів для задачі розпізнавання сарказму в текстових повідомленнях соціальної мережі.

Отримані нейронні мережі можна використовувати для розробки додатків з метою розпізнавання сарказму в соціальних мережах та для покращення роботи моделей сентимент аналізу в соціальних мережах.

# ЗМІСТ

# СКОРОЧЕННЯ

LSTM - long short-term memory

POS - part of speech

SVM - support vector machines

РНМ - рекурентна нейронна мережа

# ВСТУП

Емоції та почуття є невід’ємною і важливою частиною нашого життя. Вони часто визначають поведінку людини, складають мотиваційну основу її діяльності та виконують життєво необхідні функції. Інколи важливо розуміти ставлення людей до того чи іншого явища, вміти прогнозувати їхню реакцію на зміни, визначати суспільні думки та настрої. Для автоматизованого розв’язку цього завдання використовується один із напрямків комп'ютерної лінгвістики - сентимент-аналіз або аналіз тональності текстів.

Сьогодні, в епоху стрімкого розвитку соціальних мереж та інтернету, завдання комп’ютерного аналізу тональності тексту набуває все більшої популярності. Дослідження суб'єктивного образу об'єкта, що виникає природно або навмисно формується в інформаційному інтернет-просторі, є важливим аспектом для вдалого ведення бізнесу, визначення оцінки ефективності роботи компанії та інших видів маніпуляцій, що інформаційно обумовлені суспільною свідомістю.

Існує багато ефективних методів аналізу тональності текстів, але все ще є багато проблем вирішення яких може підвищити точність розпізнавання. Одна з таких проблем це наявність саркастичних висловлювань в текстових повідомленнях. Завдання автоматизації визначення сарказму складне, адже і самій людини інколи складно визначити чи містить певне текстове повідомлення сарказм.

**Мета й завдання роботи.** Метою кваліфікаційної роботи є дослідження наявних методів визначення сарказму в текстових повідомленнях та розробка власної моделі визначення сарказму на обраному наборі даних.

Досягнення поставленої мети передбачає розв’язання наступних завдань:

- аналіз методів формального представлення текстів
- аналіз методів попередньої обробки текстових даних
- аналіз існуючих методів визначення сарказму в текстах
- аналіз існуючих методів визначення тональності текстів
- порівняння доступних наборів розмічених даних
- розробка різних архітектур текстових класифікаторів
- візуалізація та аналіз отриманих результатів

**Об’єкт, методи й засоби дослідження та розробки.** Об’єктом дослідження є повідомлення, коментарі та інші види текстових публікації в соціальних мережах.

У якості інструменту створення програмного сервісу було обрано PyCharm - інтегроване середовище розробки мовою Python. Також були використані наступні бібліотеки мови програмування Python:

- Keras;
- Scrapy;
- Flair;
- PyTorch.

**Можливі сфери застосування**.

Розроблена модель може покращити роботу моделей сентимент-аналізу.

# 1. Сучасний стан задачі розпізнавання сарказму в текстових повідомленнях

## 1.1 Поняття сарказму та його особливості

### 1.1.1 Сарказм та іронія, як способи вираження емоцій в текстових повідомленнях

Людина може використовувати різні прийоми виразності та стилістичні фігури мови, для передачі свого емоційного стану. Вони надають образності та експресивності мовленню, підсилюють емоційне забарвлення висловлювань.

Іронія та сарказм - це одні з прийомів виразності, які є засобами висловлення комічного. Їх можна розглядати як особливий текстовий феномен, що формує двозначність речень. Іронія та сарказм є предметом дослідження в багатьох галузях науки таких як лінгвістика, філософія, психологія, нейропсихологія, неврологія, соціологія, комп’ютерна лінгвістика та інші.

Сарказм та іронія можуть вживатись з різних причин[1]., а саме:

- щоб викрити негативні явища та недоліки;
- висловити негативне оціночне судження автора;
- образити;
- для самозахисту;
- заради привернення уваги до себе;
- з метою інтелектуального домінування над співрозмовником

Іронія - це художній засіб виразності мови, що дозволяє висловити неявне глузування, насмішку, вживаючи слова чи словосполучення в сенсі, що протилежний істинному. Вона тісно пов’язана з непрямим вираженням почуттів, емоцій та оцінок. Іронія дозволяє приховати, замаскувати справжнє значення вислову, вживаючи слова в контексті, що суперечить їхньому буквальному сенсу. Об’єкт, що є ціллю іронії, висміюється,

ставиться під сумнів, сатирично викривається під виглядом схвалення та похвали.

Слово іронія походить від давньогрецького слова “eironeia”, що означанє “притвортство”, “лукавство”. Відомий філософ і філолог Лосєв О. Ф. стверджує, що спочатку цей термін вживався в значенні марнослів’я, пустої балаканини, глузливої насмішки[2]. За Платона іронія набуває більш негативного значення, приписує її людям підступним, лицемірним, здатним до обману. В Сократа термін набуває естетичного та більш позитивного значення і прирівнюється до “гри дуже тонкого і глибокого розуму”[2].

Відмінність іронії від обману полягає в тому, що іронія не просто приховує істину, а намагається виразити її в особливий, алегоричний спосіб.

У повсякденному житті ми часто використовуємо іронічні вислови:

- “Ну і кому потрібна така краса?”, - коли говоримо про неприглядну, малопривабливу річ;
- “Вже життя про це мріяв!”, - коли йдеться про річ чи пропозицію, яка нам взагалі непотрібна;
- “Тільки про це і думаю.”, - так відповідають, коли запитують про щось зовсім нецікаве.

Найвищим ступенем іронії вважається сарказм. Це ще одна стилістична фігура мовлення, якій властива гостра, уїдлива, відкрита насмішка. Саркастичні висловлювання мають негативне забарвлення і вказують на недоліки, вади людини, предмета або явища. Жорстокість, гострота, різкість викриття - основні відмінні риси сарказму[5]. Інколи його інтерпретують як жорстку, дошкульну іронію. Сарказм часто порівнюють із сатирою, якій притаманне осудливе висміювання несправедливості, нечесності та інших соціальних вад, проте він більш

гнівний, викривальний, містить високий ступінь ненависті та явної зневаги[3].

Термін походить від давньогрецького слова “sarkasmos”, що в перекладі означає “терзання” або “знущання”. Сарказм часто використовували античні оратори в своїх виступах проти опонентів. Цей прийом часто зустрічається в промовах відомих політичних діячів давньої Греції та Риму таких як Демосфен, Цицерон і Ювенал[4].

Лінгвіст Джон Хейман з коледжу Макалестер у Сент-Полі, штат Міннесота, і автор "Розмови дешево: сарказм, відчуження та еволюція мови" в своїй праці пише, що сарказм це сатирична за спрямованістю, зла іронія, що досягла трагедійного напруження, еволюціонувавши від легкої насмішки над людськими слабкостями до злої критики недоліків, від гіркого сміху до гнівного висміювання, що досягає трагічної ноти, від елементарно комічного до комічно страшного і жахливого, майже позбавленого сміху.  В зв’язку з цим цікавим є зауваження, що саркастичні люди сприймаються як злі, роздратовані і зневажливі. [6]

Сарказм та іронія мають багато спільного, тому частина авторів ототожнюють ці два поняття. І іронія і сарказм будуються на принципі використання слів не в прямому, а в переносному значенні, щоб висловити своє ставлення до об’єкта мовлення, вкладаючи викривальний прихований сенс. Іронія - це тонкий інструмент комічності. Вона подібна до жарту, коли протиставлення фактичного сенсу слів та істинного значення викликає сміх. Іронічне зауваження демонструє відношення до людини чи зображуваної ситуації, підкреслюючи безглуздість положення. Саркастичне висловлювання є відкритішим ніж іронічне, але не абсолютно, а відносно, що і відрізняє саркастичну оцінку від безапеляційного вироку. Сарказм же вбільшості не викликає сміх: він характеризує об’єкт мовлення з моральної точки зору, висловлюючи суб’єктивне неприйняття та осуд[7]. Сарказм використовується для

жорсткої критики при якій негативні особистісні якості людини або аморальність життєвих принципів набувають не просто карикатурної форми, але і викликають безкомпромісне суспільне засудження. [8]

Сарказм, на відміну від іронії, завжди має мішень для насмішок[9]. Наприклад речення - “Він винайшов ліки проти серцевої хвороби, але пізніше сам помер від неї”, - є іронічним, але не саркастичним. А ось - "Я не досягну успіху в Голлівуді, бо недостатньо погано пишу для нього", - це приклад сарказму, де ціллю є приниження американської кіноіндустрії.

### 1.1.2 Сарказм та іронія з психологічної точки зору

Іронія та сарказм це складні поняття не лише з лінгвістичної точки зору, а й з когнітивної(пізнавальної). Ще з дошкільного віку більшість людей починають розпізнавати та вживати іронічні висловлювання. Група вчених під керівництвом доктора Стефані Александра провели дослідження та зробили висновок [10], що діти здатні сприймати іронію з 4-5 років, але в такому віці вони не вбачають в ній нічого кумедного. Це відчуття приходить пізніше - у 8-9 років.

У людей, що мають травми головного мозку, може погіршуватись або і зовсім зникати здатність розпізнавати сарказм. Найтяжчі наслідки у тих, хто має пошкодження правої півкулі головного мозку, вважається, що саме вона відповідає за сприйняття образів та символів. [11]

### 1.1.3 Сарказм та іронія в соціальних мережах

За відносно невеликий період свого існування, Інтернет глибоко увійшов в наш світ. Він проник в різні сфери життя людини, а особливо - у сферу спілкування. Сьогодні ми маємо безліч варіантів взаємодії з іншими людьми в мережі Інтернет і одним з найпопулярніших є використання соціальних мереж. Люди використовують їх не лише для підтримання контактів з рідними та друзями, а й для поширення власних думок та ідей.

Соціальні мережі дозволяють використовувати різні типи інформації: відео, аудіо, зображення та тексти. Саме текстова інформація є однією з найбільш досліджуваних. Часто з текстових повідомлень можна дізнатись вік, стать автора та встановити його особистість[12].

Під час живого спілкування іронічні та саркастичні висловлювання можна виявити, звернувши увагу на жести, міміку, тон голосу та поведінку мовця. В текстових повідомленнях ці ознаки представляються у вигляді смайликів, стікерів, символів пунктуації та хештегами [13]. Хештег це - ключове слово або декілька слів, які використовується в соціальних мережах та полегшують пошук повідомлень по темі або змісту і починаються зі знака решітки(#).

Існує теорія про те, що люди частіше використовують сарказм в соціальних мережах ніж в житті. Обмін повідомленнями займає більше часу ніж спілкування віч-на-віч. Час, за який користувачі пишуть текстове повідомлення, використовується для обдумування більш складних і саркастичних висловів[14].

Якщо говорити про вплив сарказму на автоматичний сентимент-аналіз, то очевидно, що існує невідповідність між справжньою тональністю повідомлень і результатами аналізу, адже негативне враження користувачі висловлюють позитивними словами. Програма, яка не розпізнає сарказм, визначить такі повідомлення як нейтральні або навіть позитивні. Звичайно, це погіршує точність результатів аналізу тональності текстів.

### 1.2 Сучасні методи розпізнавання сарказму в текстах

Сучасні методи розпізнавання сарказму можна поділити на дві основні групи: методи, на основі контенту та на основі контексту.

У підходах, що базуються на вмісті(контенті) самих текстових повідомленнях, лексичних та мовних підказках, використовують

синтаксичні зразки для підготовки класифікаторів виявлення сарказму. Деякі дослідники[15, 16] використовують такі мовні особливості, як вставні слова, смайлики та лапки, інші зосереджують свою увагу на синтаксичних конструкціях та лексичних ознаках, пов'язаних із сарказмом. Використання позитивного висловлювання в негативному контексті є достовірною ознакою наявності саркастичних висловів[17]. Деякі автори досліджень описують особливості, такі як неявна та явна невідповідність контексту[18]. Вищезазначені методи використовують лише вхідний текст для виявлення сарказму, не звертаючи уваги на контекстну інформацію.

Контекстні підходи набули популярності з появою різних платформ соціальних мереж. Оскільки тексти з цих веб-сайтів схильні до граматичних помилок та широкого використання сленгу, контекстна інформація допомагає виправити цю ситуацію, тим самим підвищити точність ідентифікації сарказму. Відомі методи виявлення сарказму, які використовують настрої та емоційну інформацію з вхідного тексту як контекстну інформацію[19]. Дані про автора повідомлення а також історія його попередніх публікацій використовуються для відслідковування саркастичних тенденцій користувача[20, 21]. Всі ці праці доводять, що контекстна інформація(якщо вона доступна), допомагає поліпшити ефективність моделі, але не є визначальним і достатнім фактором для виявлення сарказму.

Існує багато робіт, в яких дослідники використовують методи на основі власноруч підібраних параметрів, наприклад розбір речення на частини мови, виявлення іменованих сутностей, обробка аудіо- та відеоінформації, дослідження параметрів стилометрії та особистих якостей автора публікації. Також часто використовуються методи глибокого навчання для виявлення значущих ознак.

## 1.3 Сучасний стан задачі аналізу тональності текстів та її зв’язок з задачею розпізнавання сарказму

### 1.3.1. Аналіз предметної області

Аналіз тональності текстів та сентимент-аналіз це галузь комп’ютерної лінгвістики, яка займається аналізом людських емоцій та думок, наявних в тексті. Поняття сентимент-аналізу має декілька схожих і близьких за значенням термінів: сентиментометрія (sentiment metrics), бренд моніторинг (brand monitoring), розвідка думок (opinion mining), “підслуховування” думок (opinion listening), аналіз суб’єктивності (subjectivity analysis), аналіз тональності тексту та інші[25].

Тональність — це висловлювання, що показує емоційне ставлення автора до певної події, явища, процесу чи об'єкта. Емоційна частина, висловлена на рівні лексеми або мовного звороту, називається лексичним сентиментом. Загальна тональність тексту може бути визначена як сума всіх лексичних тональностей всіх його елементів (речень) та правил за якими вони поєднуються. З точки зору сентимент-аналізу в тексті можна виділити три параметри: суб'єкт тональності (автора тексту), тональну оцінку (позитивну, негативну та нейтральну) і об'єкт тональності (те, відносно чого висловлено думку)[22].

Текстова інформація ділиться на два типи: факти та думки. Для аналізу тональності текстів потрібно визначити саме думки.

Думки поділяються на два типи:  прості думки та порівняння.

Проста думка це думка, яка відноситься до одного об'єкту. Прості думки можна описати формально. Простою думкою називають п’ятірку(E, Ai,  S, H, T)[23]:

- E - об’єкт або сутність(entity), про аспект якого  автор висловив думку;

- Ai - і-та властивість(feature), по відношенню до якої висловлено думку;
- S - тональність думки висловленої щодо властивості;
- H - автор або джерело думки;
- T - час висловлення думки(time)

Розглянемо коментар на рис.1, в якому автор висловив свою думку з приводу зміни дизайну одного з відомих сайтів прогнозу погоди. В цьому повідомленні: сайт прогнозу погоди є об'єктом, новий дизайн - аспект об'єкта, Олексій - автор висловленої думки, "2 червня, 2019 рік" - час відправки повідомлення, "жахливий" - частина повідомлення, з якої можна зробити висновок, що тональність є негативною.

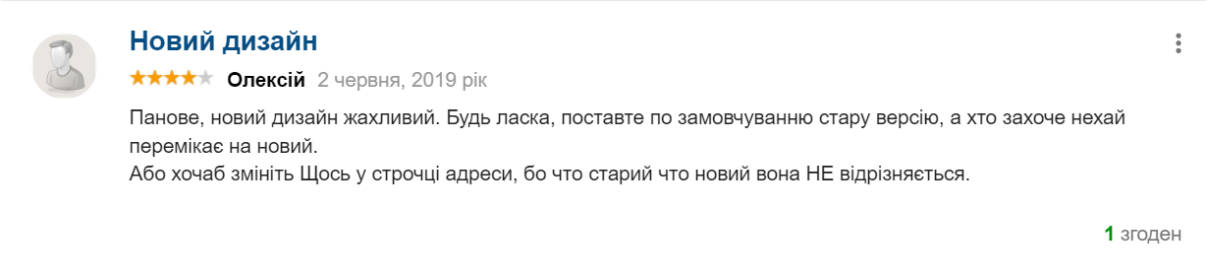


**Рис. 1 Приклад емоційно забарвленого тексту**

Другий тип думок - порівняння. В свою чергу порівняння можна розділити на три види [23]:

1. Порівняння характеристик об'єктів на користь одного (Non-equal Gradable)

2. Зрівняння характеристик різних об'єктів (Equative)

3. Перевага одного об'єкта перед іншими (Superlative)

Порівняння аспектів це означає що "аспект першого об'єкта перевершує в чомусь аспект другого об'єкта", наприклад: "AMOLED-екрани більш надійні ніж звичайні рідкокристалічні дисплеї.". Другий тип виражає схожість аспектів різних об'єкти на зразок: "I Windows, і Linux однаково зручні для розробки додатків під них". Прикладом третього типу може послужити пропозиція

“Фотоапарат від Canon виявився найдешевшим в тому магазині”. Думка другого типу визначається як кортеж (E1, E2, A, po, holder,time)[23]::

- E1 і E2 - множини порівнюваних об'єктів
- Ai - властивість, за якою порівнюються об’єкти
- po - множина  об'єктів, яким автор (holder) надав перевагу
- Time - час, коли думка була опублікована.


**Рис. 2 Приклад повідомлення, що містить думку другого типу.**

Наприклад, для повідомлення на рис.2 кортеж думки виглядає так:

(Sony Xperia, Samsung Galaxy, загальна, Sony Xperia, @wakeupwhy, 10 травня).

На відміну від простої, формальний опис порівнянь(думок другого типу) не містить оцінку емоцій автора.

Суб’єктивність  - ще одна важлива характеристика висловлювань, яка є надважливою в процесі сентимент аналізу. Об'єктивне речення містить фактичну інформацію, тоді як суб'єктивне виражає ставлення автора згідно з особистими поглядами, інтересами або смаками[22]. Прикладом об’єктивного твердження є наступне речення: “Футбольний клуб “Боруссія” був створений 1960 року в Дортмунді.”. Приклад суб’єктивного твердження: “Я фанат Боруссії” .

У більшості випадків об’єктивні речення не несуть емоційного навантаження. Дослідження показали, що видалення об’єктивних пропозицій з тексту дозволяє  підвищити точність оцінки емоційного забарвлення[23]. Висловлення, що містять думку, зазвичай є суб'єктивними, тому сентимент-аналіз текстів передбачає розпізнавання

суб'єктивного (на відміну від фактичного) матеріалу: почуттів, думок, настроїв, емоцій та інших видів інформаційної поведінки.

### 1.3.2 Дослідження методів класифікації тональності

Задача сентимент-аналізу, полягає в класифікації на два класи – позитивний та негативний, але можливий ще третій варіант - нейтральний. Оскільки повідомлення на форумах часто мають також і чисельну оцінку(зазвичай кількість зірок від 1 до 5). Для класифікації на "позитивний" та "негативний" класи часто використовують цю ж саму шкалу оцінок. Якщо відгук містить 4-5 зірок його позначають як позитивний, якщо 1 або 2 зірки, то негативний.

В задачі аналізу тональності, слова-сентименти, які є емоційно забарвленими, мають найбільше значення. Для визначення таких слів використовуються методи текстової класифікації, наприклад, класифікація, метод максимум ентропії, методом опорних векторів (SVM, support vector machines) або метод Байєсівського класифікатора.

Основним завданням сентимент-аналізу є визначення набору ознак (features), наприклад терміни та частота їх входжень. Ці ознаки є досить поширеними в традиційній текстовій класифікації. Інколи також враховується і порядок слів у реченнях.

Визначення до якої частини мови (POS, Part Of Speech) належить слово також може покращити результати сентимент-аналіз. На основі цієї інформації слова можна аналізувати по-різному. Наприклад, було виявлено, що прикметники несуть в собі велику частку інформації про тональність думки. Отже, можна розробити окремий спосіб аналізу прикметників як особливих ознак.

Емоційно забарвлені слова – це слова, які висловлюють позитивну чи негативну думку. Наприклад, слово чудовий часто використовують, щоб виразити позитивне враження, а от слово жахливо, зазвичай, висловлює

негативної думки. Більшість емоційно забарвлених слів це або прикметники, або прислівники, часом таку ж функцію можуть виконувать іменники (жах, дурниця).

Часто для зміни полярності речення використовуються заперечення. Наприклад, вираз “Мені не подобається обслуговування в цьому закладі” є негативним. Але в не завжди заперечення свідчать про негативне висловлювання , наприклад частка “не” у фразі “не тільки, але також” не змінює тональність на негативну.

### 1.3.3 Методи навчання з учителем

Сьогодні найбільш поширеними в дослідженнях методами є методи машинного навчання з учителем. Суть таких методів в тому, що на першому етапі машинний класифікатор навчається на заздалегідь розмічених текстах, а потім отриману модель використовується при аналізі нових документів. Приклад алгоритму:

1. спочатку проводиться збір колекцій текстів;
2. кожен документ представляється вектором ознак (аспектів);
3. для кожного документа вказуємо правильний тип тональності;
4. виконується навчання класифікатора;
5. отриману модель застосовують для класифікації тональності документів нової колекції.

#### 1.3.1.1 Наївний класифікатор Байєса

Наївний класифікатор Байєса – це простий імовірнісний класифікатор, в основі якого лежить теорема Байєса зі строгими припущеннями про незалежність. Класифікатор Байєса дає можливість визначити ймовірність того, що елемент спостереження належить до одного з наперед заданих класів. Мінусом цього методу (тому він і називається “наївним”) є те, що ми вважаємо, що слова зустрічаються незалежно, що в загальному випадку

не є вірним[24]. Однак він досить ефективний і має багато переваг - швидкість, простота, невибагливість до розмірів пам’яті, через що він такий поширений.

Основне завдання класифікації полягає в тому щоб зрозуміти до якого класу тональності належить документ, тому нам потрібна не сама ймовірність, а найбільш ймовірний клас. Байєсівський класифікатор застосовує оцінку апостеріорного максимуму, щоб визначити найбільш вірогідний клас[24]:

$C_{max} = arg\,max\,[P(c)\prod_{i=1}^{n} P(w_i|c)]$ , де

$P(c)$- безумовна ймовірність появи елемента класу у вхідній множині елементів;

$\prod_{i=1}^{n} P(w_i|c)$ - добуток ймовірностей, що і-тий елемент належить класу с.

Отже, потрібно обчислити ймовірність для всіх класів і обрати клас з максимальною вірогідністю.

1.3.1.2 Метод опорних векторів

Метод опорних векторів - це метод класифікації, який визначає класи за допомогою меж просторів. Цей метод належить до методів лінійних класифікаторів[24]. Вхідні вектори переводяться в простори вищих порядків і шукаються гіперплощини, які мають найбільші проміжки між ними.

Суть методу можна показати на прикладі(рис. 3): нехай маємо точки на площині, які входять в навчальну вибірки та які можна поділити на два класи. Розділити ці точки можна провівши лінію. Далі, всі нові точки, що

не належать тестовій вибірці, будуть класифікуватися за наступним правилом: точка, що лежить вище прямої відносимо до класу А, а точки, що потрапляють нижче прямої, відносимо до класу В.

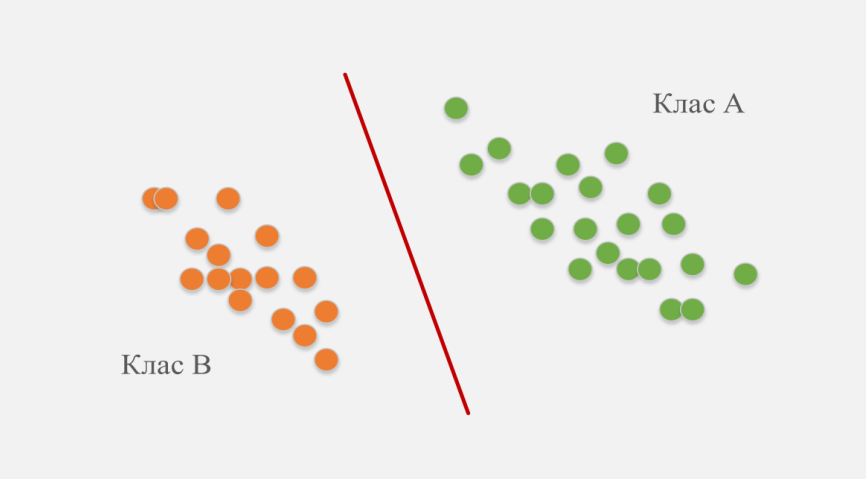


**Рисунок 3. Класифікація на два класи за допомогою розділяючої прямої**

Така пряма має назву розділяючої прямої[24]. Але в багатовимірних просторах така пряма вже не може розділяти класи за таким правилом. Тому замість прямих потрібно вводити гіперплощини.

1.3.1.3 LSTM

LSTM (Long short-term memory) - один з видів архітектури рекурентних нейронних мереж. Вона була запропонована в 1997 році Зеппом Хохрайтером і Юргеном Шмідхубер. LSTM-мережа при достатній кількості елементів мережі може виконати будь-яке обчислення, на яке здатний звичайний комп'ютер, за наявності відповідної матриці ваги[27].

Це нейронна мережа, яка на додаток до інших мережевих модулів містить рекурентні LSTM-модулі, що можуть пам'ятати значення протягом довгих проміжків часу, завдяки тому, що вони не використовують функцію активації всередині своїх рекурентних компонентів[29]. Тож, збережене значення не буде втрачатися з часом, і при використанні методу зворотного поширення помилки градієнт не зникає з часом при тренуванні мережі.

### 1.3.4 Методи навчання без учителя

Методи навчання для аналізу тональності з відсутністю вчителя проводять класифікацію, на основі певних синтаксичних структур, які зазвичай використовують люди для висловлення власних думок. Такі синтаксичні структури складаються з певних частин мови (Part-of-speech, POS) та їх тегах.

Слова-сентименти можуть використовуватись в методах для аналізу тональності без вчителя. Такий метод виконує класифікацію, що грунтується на певних синтаксичних структурах, які зазвичай використовують люди для висловлення думок. Ці синтаксичні структури базуються на певних частинах мови.

Алгоритм класифікації наступний:

Етап 1.

Два слова, що записані послідовно, видаляються, якщо їх POS теги відповідають одному з визначених шаблонів. Наприклад, два послідовних слова вилучаються, якщо перше слово це прислівник, а друге слово – іменник. Так, наприклад, для такого речення “У цьому ресторані такі смачні страви”, “смачні страви” вилучається, адже ця фраза відповідає шаблону. Іменники та дієслова несуть контекстну інформацію, тому що саме в залежності від контексту можуть висловлювати різні думки. Наприклад, прикметник "непередбачуваний" буде виражати негативну емоцію, якщо це відгук про машину (“непередбачуване рульове управління”), або позитивну емоцію, якщо ми говоримо про відгук до фільму (“непередбачуваний сюжет”).

Етап 2.

Оцінимо емоційне забарвлення всіх вилучених фраз, для цього використовується метрика точної взаємної інформації (PMI - Pointwise mutual information)[30]:

$$PMI(term_1, term_2) = log_2(\frac{Pr(term_1 \wedge term_2)}{Pr(term_1)\,Pr(term_2)})$$

PMI показує ступінь статистичної залежності між двома термами. Pr(*term*1 ∧ *term*2) – це фактична ймовірність одночасної появи першого та другого терму, а Pr (*term*1)Pr (*term*2) – спільна ймовірність виникнення двох термів, за умови, що вони є статистично незалежними. Тональність фрази оцінюється на основі його зв'язку з опорними словами "відмінно" та "погано"[30]:

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

Ймовірності обчислюються враховуючи кількість запитів до пошукової системи та підраховуючи кількість влучень. Кожен запит, отримує набір релевантних документів, це і є кількість влучень. Виконуючи пошук за двома термами разом та окремо, можна обрахувати ймовірності за формулою[30].

Етап 3.

Алгоритм обчислює середнє значення SO всіх фраз і визначає речення, як позитивне, якщо середній SO є позитивним або як негативне– в іншому випадку.

### 1.3.5 Методи, засновані на словниках

Поширеним є лінгвістичний метод на основі словників. Алгоритми, засновані на правилах, показують точніші результати, так як ці методи тісно пов'язані з семантикою слів, а от методи машинного навчання працюють зі статистикою і теорією ймовірності.

Але, лінгвістичний підхід має ряд серйозних недоліків. Він може надати відносно точні результати, будучи реалізованим граматично правильних текстів. Оскільки головним застосуванням аналізу тональності є бізнес-розвідка, стає зрозуміло, що отримати всі граматично правильні тексти, ми не можемо. Не можна сподіватися на відсутність орфографічних

та інших помилок в соціальних мережах. Також, підхід, заснований на правилах, сильно залежить від конкретної мови. Методи, що засновані на навчанні без учителя, зазвичай показують невисоку точність, хоча і не потребують початкової навчальної вибірки.

Ще одна проблема методів, заснованих на словниках і правилах, є складність процесу складання словника. Щоб класифікація документа мала високу точність, словникові терміни повинні мати правильну вагу, яка підходить предметній області документа.

### 1.3.6 Проблеми сентимент-аналізу

В процесі проведення сентимент аналізу, особливу увагу потрібно звернути на наступні особливості, які можуть ускладнювати процедуру аналізу тональності текстів:

- Тональність може залежати від предметної області тексту. Так, наприклад, слово “величезний” застосовне до опису телевізора, має позитивну тональність, але як характеристика, наприклад ноутбуку, це слово приймає дещо негативний відтінок.
- Використання сарказму. Сарказм це одне з почуттів, які важко відслідкувати автоматичного та вірно інтерпретувати. Приклад: "Яка чудова у них служба підтримки, через три дні перезвонили".
- Неоднозначність негативних відгуків. Наприклад, речення “Яка сукня, просто очманіти!” хоча і є позитивним, але може помилково бути інтерпретоване як негативне.
- Значення тональності залежить, від точки зору дослідника. Наприклад, фраза “продажі компанії Coca-Cola стрімко зросли” має позитивний забарвлення для компанії Coca-Cola і негативний для Pepsi.
- Омоніми — це слова, які однаково звучать та пишуться, але мають різне значення. Наприклад: варта (іменник) – варта

(прикметник жіночого роду), злий (прикметник) — злий (дієслово наказового способу).

- ➢ Використання смайликів. Схвалення або осуд часто передаються в повідомленнях за допомогою специфічних символів, різноманіття яких відображає різні емоційні стани. Смайлик став емоційним маркером. У зв'язку з цим виникає потреба в правильному кодуванні таких символів з урахуванням культурних, ментальних особливостей їх авторів.
- ➢ Помилки: випадкові і навмисні, так само як і безграмотні тексти, ускладнюють процес визначення об'єкта моніторингу та спотворюють значення аспектів, що призводить до невірного розуміння тональності повідомлення.

## 1.4 Огляд сучасних методів попередньої обробки текстів

Зазвичай, перед початком класифікації тональності, виконується попередня обробка даних. Цей етап такий не менш важливий, ніж етап безпосередньої класифікації, адже від того в якому стані будуть вхідні дані, буде залежати точність результатів. Загалом, мета цих процедур – якнайбільше зменшити розмірність задачі без втрати емоційної складової. Спершу проводиться видалення чисел, посилань на інтернет сторінки, власних імен, звертань та стоп-слів з тексту, що аналізується. Наступний крок це представлення документу у вигляді вектора ознак та проведення стемінгу.

### 1.4.1. Представлення документу у вигляді вектора ознак

Вектор ознак – це алгебраїчна модель подання текстів[23]:

Векторне представлення документу j ;

$l_j = (v_{1j}, v_{2j}, \dots, v_{nj})$, де

$v_{1j}$ – вага конкретного терміну у документі j, n – кількість термінів.

Найпопулярніший спосіб представлення документів у задачах аналізу природної мови – це у вигляді набору N-грам[31]. Наприклад, речення “Я обожнюю ванільне морозиво” можна представити у за допомогою набору уніграм (Я, обожнюю, ванільне, морозиво) або біграм (Я обожнюю, обожнюю ванільне, ванільне морозиво).

Як правило, найкращі результати можна отримати за допомогою уніграм та біграм, а от використання N-грам більш високих порядків призводить до втрати ефективності, оскільки розмір навчальною вибірка в більшості випадків є недостатнім. У більшості випадків є сенс оцінити результати із застосуванням уніграм та біграм (Я, люблю, ванільне, морозиво, Я люблю, люблю ванільне, ванільне морозиво).

Менш поширеним варіантом є представлення слів у вигляді символьних N-грам. Так, речення наведене вище може бути передане у вигляді 4-символьних N-грам: “я обо”, “обож”, “жнюю”, “нюю”, і т.д. Даний метод може здатися примітивнішим ніж попередні, адже набір символів фіксованої довжини є не дуже інформативним, але цей метод за певних умов може давати результати навіть кращі ніж N-грами слів. N-граммам символів у відповідність можна поставити морфеми слів, у випадку слова "люблю" корінь “люб” містить в собі основне значення. Символьні N-грами часто використовуються у двох випадках:

- якщо у тексті велика кількість орфографічних помилок – набір символів у тексті, що містить орфографічні помилки буде дуже схожим на набір символів у тексті без помилок;
- для мов, у яких слова змінюються за відмінками; слова в українській мові можуть мати багато різних закінчень, префіксів і так далі, але при цьому корінь слова залишається незмінним.

Часом можуть використовуватись і інші ознаки: частини мови, смайлики, стікери, знаки оклику, заперечення, вигуки і т.д.

Існує кілька базових функцій для розрахунку ваги вектора. Одним з найбільш часто застосовуваних методів оцінки ваги ознак в інформаційному пошуку є метод TF-IDF[32]. TF-IDF (term frequency-inverse document frequency) – міра, що дає оцінку важливості слова в контексті документу[32]. TF (частота слова) – це відношення кількості входження деякого слова до загальної кількості слів у тексті. Це відношення показує частоту, тож можна припустити, що і важливість слова в певному документі[32].

IDF (inverse document frequency – обернена частота документа) – обернена частота, з якою деяке слово вживається у певному документі[32]. Міра IDF робить меншою вагу часто вживаних слів. Міра TF-IDF складається з двох елементів: TF та IDF.

В сентимент-аналізі цей метод показує погані результати. На відміну від завдання пошуку, для сентимент-аналізу не так важливі слова, які повторюються в тексті багаторазово(тобто слова з високим показником TF).

Досліджено, що бінарна функція зважування векторів показує вище ефективність.[33] Частота появи деякого терміну менш важлива ніж факт наявності його у тексті. Бінарні вектори задаються послідовністю нулів і одиниць: якщо деякий термін із словника вибірки є в тексті – то вагу терміну ставимо 1, інакше – 0. Частотні вектори базуються на кількості входжень певного терміну в документ. Наприклад, речення “Я обожнюю ванільне морозиво” буде представлене наступним вектором (опускаються слова з вагою = 0):

{ " Я ": 1 , " обожнюю": 1 , " ванільне": 1 , " морозиво": 1 }

Деякі методи оцінюють важливість слів, обчислюючи ваги цих слів і показують набагато кращі результати при класифікації тональності, наприклад, метод дельта TF- IDF[32] .

Суть методу дельта TF-IDF зробити так, щоб слова, в яких тональність не нейтральна, отримали найвищу вагу, адже саме вони визначають тональність всього тексту. Для оцінки ваги слова t використовується наступна формула:

$$Vt,d = Ct,d * \log ( |N| * Pt\ P| * Nt ) \text{ , де:}$$

- $Vt,d$ – вага слова t у документі d;
- $Ct,d$ – кількість слів t в документі d;
- $|P|$ – кількість позитивних документів;
- $|N|$ – кількість негативних документів;
- $Pt$ – кількість позитивних документів, які мають слово t;
- $Nt$ – кількість негативних документів, які мають слово t.

Нехай, маємо набір даних з відгуками про деякий фільмів. Дослідимо такі слова: “відмінний”, “нудний”, “сценарій”. Другий множник log (...) більш важливий у формулі дельта TF-IDF. Для цих трьох слів він буде відрізнятися: слово “відмінний” частіше трапляється в позитивних (*Pt*) відгуках і майже відсутній у негативних (*Nt*), тому його вага буде позитивним числом, адже відношення *Pt*/*Nt* буде числом набагато більше 1; слово “нудний” навпаки трапляється в основному в негативних відгуках, тому відношення *Pt*/*Nt* буде менше одиниці, а отже логарифм від’ємний. В результаті вага слова буде від’ємним числом, але великим по модулю; слово “сценарій” зустрічається з однаковою ймовірністю і в позитивних і в негативних відгуках, тому відношення Pt/Nt буде дуже близьким до одиниці, і в підсумку логарифм буде майже нуль. Вага слова теж буде дорівнювати нулю. В результаті отримаємо, що вага позитивних слів буде великим додатнім числом, вага слів з негативною тональністю

буде від’ємним числом, вага нейтральних слів буде близькою до нуля. Такі оцінки допомагають підвищити точність класифікації тональності текстів.

Word2Vec - розробка компанії Google, яка описує слова як вектор чисел певної розмірності. Слова, що стоять на однакових позиціях, а отже повинні мати схожий сенс матимуть схоже векторне представлення. Це дозволяє спростити задачу аналізу тональності.

### 1.4.2 Стемінг та лематизація

У деяких дослідженнях перед безпосередньо аналізом тексту всі слова піддаються процедурі стемінгу - видалення закінченнь, - або лематизації - приведення до початкової форми. Така процедура здійснюється для того, щоб зменшити розмірність вхідних даних. Але, на практиці як правило це дає значного ефекту. Причина полягає в тому, що, при видаленні закінчень слів, ми також втрачаємо морфологічну інформацію, яка може допомогти в процесі сентимент-аналізу. Наприклад, слова “хочу” і “хотів” мають різне емоційне забарвлення. Перше слово швидше за все висловлює позитивну тональність, тому що автор може висловлювати надію, бажання то у другому випадку дієслово у минулому часі, що може свідчити про те, що автор жалкує за чимось.

## 1.5 Огляд сучасних моделей класифікації текстів

На сьогодні найкращі результати показують моделі на основі архітектури трансформерів, які застосовують механізм самоуваги для отримання правильної матриці ембендингів з слів тексту.

### 1.5.1 Використання архітектури трансформерів для задачі класифікації текстів

Рекурентні нейронні мережі, зокрема, long short-term memory та керовані рекурентні нейронні мережі, впевнено зарекомендували себе як сучасний підхід до вирішення задач трансдукції, таких як моделювання

мови та машинний переклад. З того часу численні зусилля спрямовувались на вдосконалення рекурентних мовних моделей та архітектур кодерів-декодерів. У 2017 році виходить стаття “Увага це все, що вам необхідно”(“Attention Is All You Need”) [34], у якій було запропоновано нову архітектуру нейронної мережі під назвою трансформер. Вона перевершила моделі засновані на рекурентних нейронних мережах в обчислювальній ефективності.

Як і рекурентні нейронні мережі(РНМ) ця архітектура показала високу ефективність в задачах аналізу природної мови. Та на відміну від РНМ вона може обробляти вхідні дані незалежно від порядку в цих даних. Тобто не обо’язково обробити першу частину речення перед тим, як переходити до другої. Це означає, що трансформери можна легко розпаралелити, тим самим пришвидшити час тренування моделей.

Архітектура трансформера складається з кодувальника та декодувальника. Кодувальник отримує на вхід векторизовану послідовність з позиційною інформацією. Декодувальник отримує на вхід частину цієї послідовності і вихід кодувальника. Кодувальник і декодувальник складаються з шарів. Шари кодувальника послідовно передають результат наступному шару в якості його входу. Шари декодувальник послідовно передають результат наступного шару разом з результатом кодувальника в якості його входу.

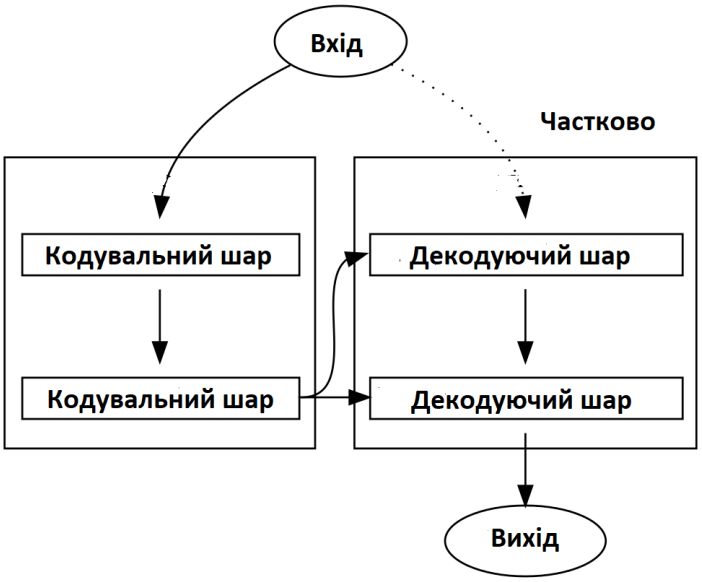


**Рис. 4 Кодувальні та декодувальні шари**

**Механізм самоуваги**

Як зрозуміло з назви статті особливу роль в цій архітектурі відіграє механізм самоуваги або уваги на себе (self-attention mechanism).

Увага енкодера звернуто на попередній шар енкодера, тобто як би на себе ж, але в минулому. Найперший шар не має попередніх, тому трансформер отримує “сирі” вектори слів і концентрує увагу на деяких з них. Новий варіант вхідного пропозиції, де якісь слова вже позначені як важливі, потрапляє в нейромережу з прямим зв'язком. Звідти перероблений вектор потрапляє на наступний шар кодера, і так повторюється стільки разів скільки шарів міститься в кодувальнику (зазвичай їх 6).

Наступний шар знову додає ваги якимось словами (але вже іншим, тому що у нього інші початкові налаштування, тобто ваги), результат передається далі.

У публікації про трансформер автори вводять три нових поняття: запит (query), ключ (key) і значення (value).

Є аналогія, яка допоможе краще зрозуміти, навіщо потрібні “запит”, “ключ” і “значення”. Уявіть, що ви шукаєте відео на Youtube. У базі даних Youtube зберігається відео - це значення V. Там є ключі K - це набір тегів

до відео. Одному ключу, тобто набору тегів, відповідає одне відео, тобто значення. Є запит Q - то, що ви пишете в пошуковому рядку. Запит Q зіставляється з усіма ключами (тегами) K в базі даних, знаходяться найближчі до запиту ключі, а користувач отримує значення V (тобто відео), зіставлені знайденим ключам.

Отже, маємо запит Q, шукаємо найближчі до нього ключі K і видаємо відповідні значення V.

Кожен механізм уваги параметризований матрицями ваг запитів $W_Q$ , матриця ваг ключів $W_K$, матриця ваг значень$W_V$. Для обчислення уваги вхідного вектора X до вектору Y, обчислюються вектора $Q = W_Q X$, $K = W_K X$, $V = W_V Y$. Ці вектори використовуються для обчислення результату уваги за формулою:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}) V$$

Результат, обчислений за цією формулою, також називають скалярною увагою. На кожному шарі уваги є три різних матриці ваг - матриці ваги “запиту”, “ключа” та “значення”. Вони заповнюються випадково при першому запуску, і визначаються під час навчання. На кожному шарі вони різні: так шари вчаться звертати увагу на різні речі і доповнюють один одного.

Метою механізму уваги є знаходження значень V, які найкраще підходять під запит Q конкретного слова. Кожен рядок матриці Q - “запит” одного слова, а кожен стовпець транспонованою матриці K - “ключ” одного слова. Скалярний добуток стовпця і рядка матриці - число, яке займає одну клітинку в результуючій матриці. Великий скалярний добуток будуть мати ті вектори, які приблизно спрямовані в одному напрямку. Проводиться масштабуванням (scaling) - отримана матриця (Q * K) ділиться на

квадратний корінь довжини одного ключа - d(k). Далі для нормалізації застосовується функція Softmax.

Останні крок це зважування кожного значення V, перемножуючи його на матрицю, отриману з попереднього кроку. Тоді в підсумковому векторі уваги Z можуть ужитися два або три елементи V, помножених за ступенем важливості. Вектор уваги Z і є результат роботи механізму скалярної уваги. На рисунку 5 графічно зображено роботу алгоритму:

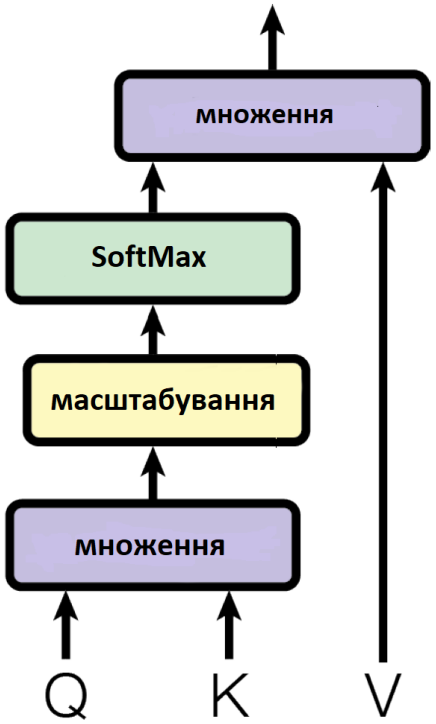


**Рис. 5 Алгоритм самоуваги**

**Кодувальник**

Кодувальник (encoder) складається з двох частин: механізму самоуваги та нейронної мережі з прямим зв'язком. Кодувальник отримує на вхід слова і видає певні ембеддінги, відповідні словами, які будуть використовуватися декодером. Від попереднього кодувальника механізм самоуваги отримує кодування входу , далі визначаються частини входів, які є релевантними по відношенню одна до одної, ця інформація буде

знаходитись в наборі кодувань виходу. Подальша обробка кожного кодування виходу здійснюється нейронною мережею з прямим зв'язком.

Вона потрібна, щоб моделювати складніші функції. Функції на шарі уваги декодера - лінійні. Безглуздо передавати результат від однієї лінійної функції до іншої по ланцюжку, додавання нових шарів не ускладнює модель і не наближає її до комплексної реальності. Ця проблема вирішується, якщо між лінійними шарами уваги додати шари з нелінійними функціями активації.

У трансформері нейромережа з прямим зв'язком - це матриця ваг, на яку треба помножити вектор, нелінійна функція активації, через яку йде результат, і ще одна матриця ваг, на яку знову множиться результат. Матриці ваг потрібно спочатку визначити в ході навчання, в ненавченої нейромережі там випадкові числа. Після кожного множення на матрицю додається певне число(зсув).

Функція активації може бути різною, але стандарнтно для трансформера обирається функція ReLU. Ця функція порівнює елемент вектора з нулем, і якщо елемент від’ємний замість нього пише нуль. Графік ReLU виглядає ось так:

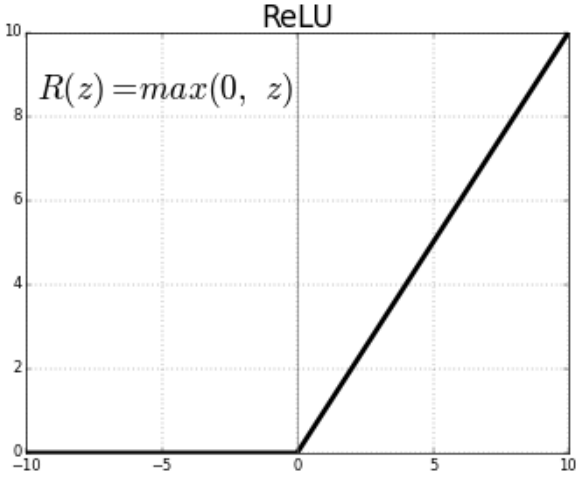


**Рис. 6 Графік функції ReLU**

Нейромережа з прямим зв'язком в трансформері - це множення на матрицю, “зсув”, функція ReLU, множення на другу матрицю, ще один “зсув”.

Після кожного проходження механізму уваги результат проходить через нелінійну функцію. Це відбувається кілька разів і ускладнює підсумкову модель.

Далі, ці кодування виходу переходять наступному кодувальникові та декодувальникам на вхід.

**Декодувальник**

Декодувальник (decoder) складається з трьох частин: механізму самоуваги, механізму уваги над кодуваннями, та нейронної мережі з прямим зв'язком. Принцип роботи декодувальника подібний до роботи кодувальника, але використовується ще один додатковий механізму уваги, що отримує релевантну інформацію з кодувань від кодувальниками.

На сьогодні є кілька популярних архітектур трансформерів, які чудово справляються з побудовою векторів ембендингів, та мають велику кількість попередньо навчених моделей.

**Архітектура BERT** - це двонаправлена мовна модель, яка має на меті вивчити контекстуальні відносини між словами за допомогою архітектури трансформатора. Вхідними даними є текст документу (або речення), або текстова пара. Перший параметр кожної послідовності - це спеціальний параметр класифікації [CLS], за яким слідує два тексти розділені певним символом. На додаток до вбудованих параметрів, BERT використовує позиційні параметри для представлення позиції токенів у послідовності. Для навчання BERT застосовує задачі маскованої мовної моделі та передбачення наступного речення. У першій задачі BERT випадково маскує 15% усіх токенів і вчиться їх передбачати.

**Архітектура RoBERTa**, проводить повторне навчання BERT за удосконаленою методологією. Використовується набагато більше даних, з

більшим обсягом груп тренування та довшим часом навчання. У RoBERTa змінилася задача передбачення наступного речення та використовується кодування пари байтів як алгоритм токенізації.

**Архітектура DistilBERT** - це легша і швидша версія BERT, в якій зменшили розмір моделі на 40% і зберегли 97% можливостей на завданнях розуміння мови. Процес дистиляції включає навчання моделі на основі взаємодії з BERT.

## 1.6 Огляд наявних розмічених датасетів для задачі розпізнавання сарказму

На даний момент у відкритому доступі є велика кількість розмічених датасетів, які можна застосовувати для навчання класифікатора саркастичних повідомлень.

З поміж всіх, були виділені наступні:

- SARC: Reddit Training Dataset [35];
- Sarcasm Corpus V2 [36];
- Headlines dataset [37].

Вони є найбільш якісними та перспективними для навчання сучасних моделей глибокого навчання.

### 1.6.1 SARC: Reddit Training Dataset

Reddit - це сайт, на якому користувачі можуть створювати публікації та обговорювати їх. Можна також відповідати на коментарі інших користувачів, внаслідок чого утворюється деревоподібна структура розмови, де кожен коментар має один батьківський.

Для того, щоб позначати коментарі, що містять саркастичні висловлювання, відвідувачі Reddit додають спеціальний маркер “/ s” в кінець саркастичних висловлювань. Такий метод анотації сарказму походить від фейкового html тегу <sarcasm></sarcasm>.

Набір даних Reddit це велика кількість коментарів з цього сайту, які були додані з січня 2009 року по квітень 2017 року. Ще однією особливістю є те, що датасет містить несаркастичні коментарі лише тих авторів, які використовували в своїх попередніх коментарях маркер для позначення сарказму, щоб бути впевненими, що вони знайомі з таким методом позначення. Це зроблено для того,, щоб уникнути коментарів з сарказмом, але без позначення. Кожен коментар має автора, дату додавання, лейбл, що показує чи містить він сарказм, власний ідентифікатор та ідентифікатор батьківського коментар. Дані зберігаються у форматі CSV. Файл містить 533 мільйони коментарів, з яких у близько 1.3 мільйона коментарів наявний сарказм.

### 1.6.2 Sarcasm Corpus V2

Соціальна мережа Твіттер дозволяє користувачам постити невеличкі текстові повідомлення(обмеження до 280 символів), що називаються “твітами”. Таким чином можна вести свою сторінку-блог, висловлюючи свої думки, погляди на певні теми. Також Твіттер дозволяє ретвітнутнути твіт, тобто поділитись дописом іншого автора у себе на сторінці, додавши власний коментар до нього.

Для Твіттера характерним є популярність використання функції хештегів. Хештег це - ключове слово або декілька слів, які використовується в соціальних мережах та полегшують пошук повідомлень по темі або змісту і починаються зі знака решітки(#). Для того, щоб показати іншим користувачам, що ваш допис містить сарказм використовуються хештеги #sarcasm, #sarcastic, #not. Саме базуючись на цих позначеннях було зібрано даний датасет.

В датасет додавались саркастичні чи несаркастичні твіти лише тоді, коли вони з’являються в діалозі (тобто починались із символу «@» - символ користувача) і мали принаймні два або більше попередніх твітів,

тобто наявний контекст розмови. Сумарно дана колекція містить, приблизно, 6 тисяч твітів, 3 тисячі з яких містять саркастичні вислови.

### 1.6.3 Headlines dataset

Набір даних Headlines для виявлення сарказму зібраний на двох веб-сайтах новин. “The Onion” - це американський веб-сайт на якому публікують саркастичні версії світових новин. Набір даних включає всі заголовки із категорій “Короткі новини”, “Новини у фотографіях” та реальні (несаркастичні) заголовки новин з американського інтернет-видання HuffPost. Цей набір даних має наступні переваги перед існуючими наборами даних з Twitter та Reddit:

- Оскільки заголовки новин пишуться професіоналами у офіційній формі, без орфографічних помилок та без вживання неформальних висловів.
- Крім того, оскільки основною ціллю видання “The Onion” є публікація саркастичних новин, набір даних має високу точність та набагато менший рівень шуму в порівнянні з наборами даних, що описані вище.
- На відміну від твітів, які є відповідями на інші твіти, отримані заголовки новин є незалежними.

## 2. Розробка моделі машинного навчання для виявлення сарказму в текстових повідомленнях

### 2.1 Постановка задачі та цілі моделі виявлення сарказму в в текстових повідомленнях

Завдання полягає у визначенні, чи є певний коментар, зроблений в соціальний мережі, є саркастичним. Для експерименту буде

використовуватися датасет SARC, розмічений на даних з соціальної мережі reddit.com.

Позначимо $U = \{u_1, ..., u_{Nu}\}$, як множину всіх користувачів для нових користувачів, $N_u$ - кількість користувачів, які приймають участь в обговореннях. Також, позначимо множину форумів обговорення, як $S = \{s_1, \dots s_{Nt}\}$, де $N_t$ - це кількість всіх форумів. Відповідно, необхідно класифікувати коментар $C_{ijk}$ *який є* k-тим по порядку, зроблений користувачем $u_i$ в форумі $s_j$ на саркастичний або не саркастичний.

## 2.2 Обґрунтування вибору платформи та мови програмування

Для розробки програмного забезпечення було обрано мову Python та бібліотеку обробки природної мови FlairNLP.

Python — це інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою, яка була розроблена в 1990 році Гвидо ван Россумом. На сьогодні, це одна з найпопулярніших мов програмування, в якій є велика кількість реалізованих бібліотек з відкритим кодом, для навчання моделей машинного навчання, дослідження та візуалізації даних.

FlairNLP - це бібліотека з відкритим програмним кодом, призначена для спрощення, та організації роботи над задачами обробки природних мов. Навчання відбувається за допомогою бібліотеки PyTorch, яка є однією з найпопулярніших бібліотек глибокого навчання. Бібліотека FlairNLP має велику кількість попередньо навчених моделей, які мають високі результати на популярних датасетах виділення іменованих сутностей та сентимент аналізу.

## 2.3 Розвідувальний аналіз обраного набору розмічених даних

Для обраного датасету SARC було проведено розвідувальний аналіз даних, результати якого представлені на графіках нижче.

Графік на рисунку 7 показує, що датасет збалансований, оскільки кількість саркастичних та несаркастичних коментарів однакова.

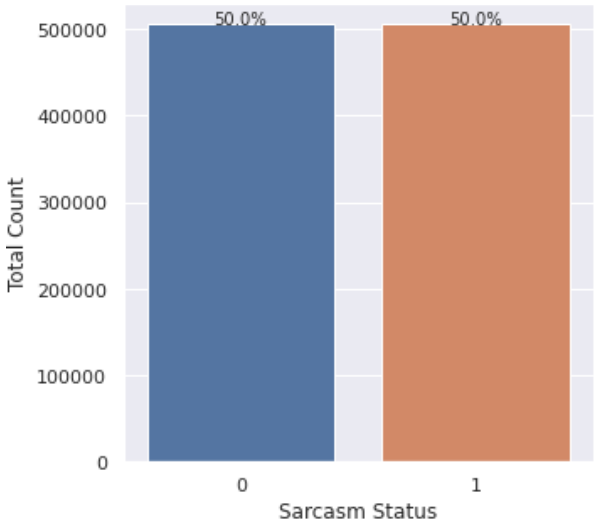


**Рис. 7 Кількість саркастичних та несаркастинчих коментарів**

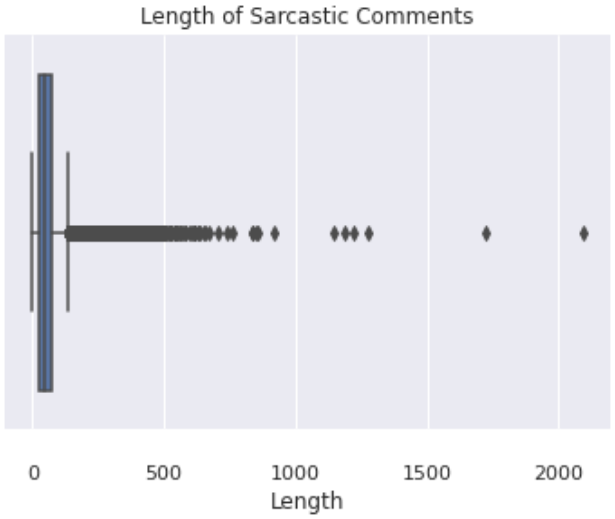


**Рис. 8 Довжина саркастичних коментарів**

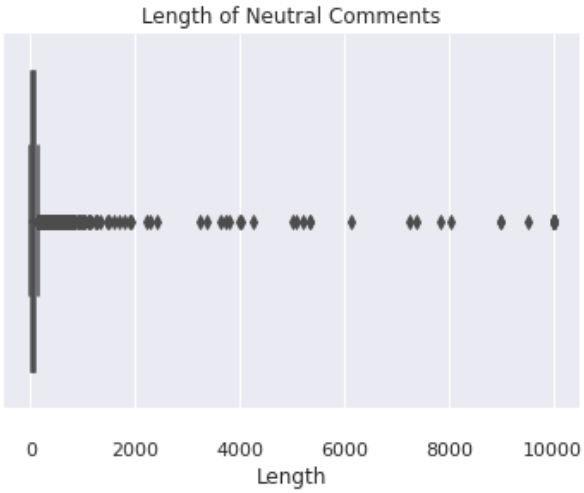


**Рис. 9 Довжина несаркастичних коментарів**

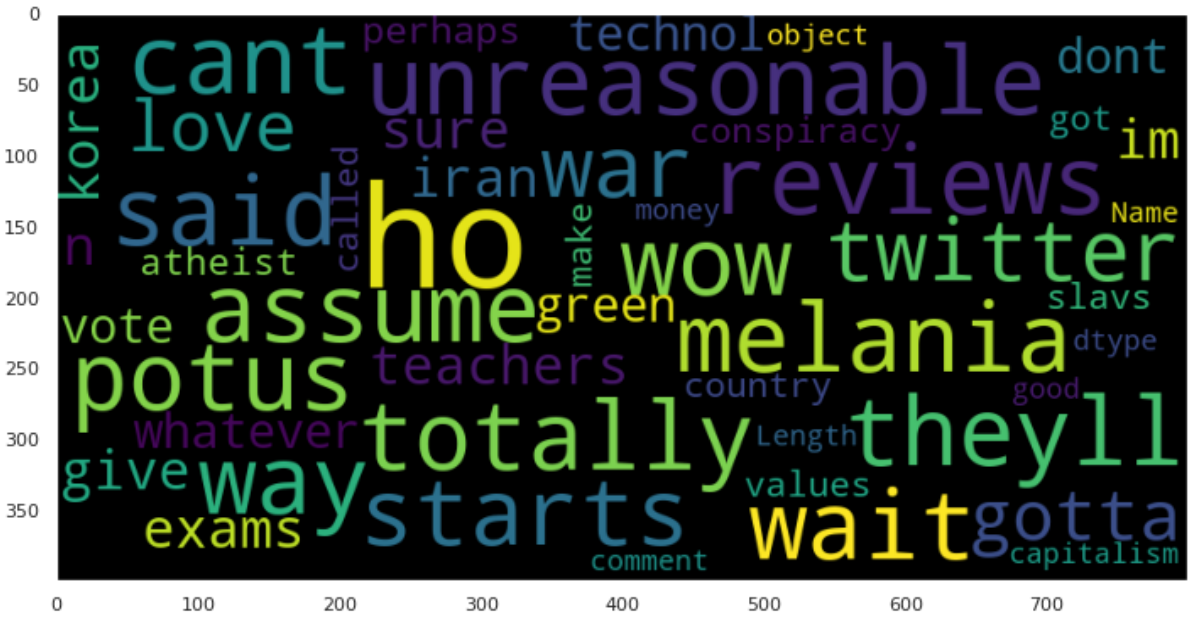


**Рис. 10 Найчастіше вживані слова в досліджуваному датасеті**

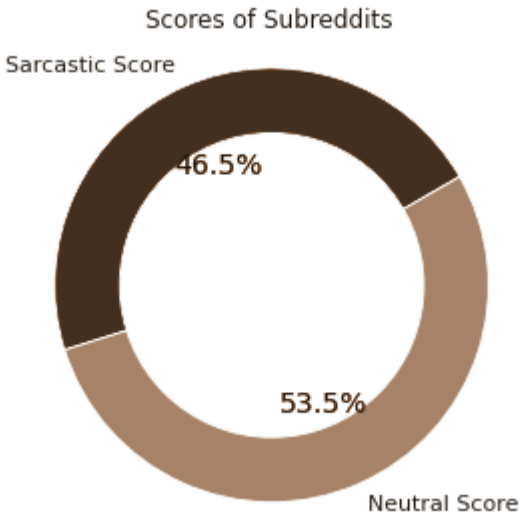


**Рис. 11 Популярність саркастичних та несаркастичних коментарів.**

На рисунку 11 показано, що саркастичні коментарі, як правило, менш популярні і мають нижчу кількість балів.

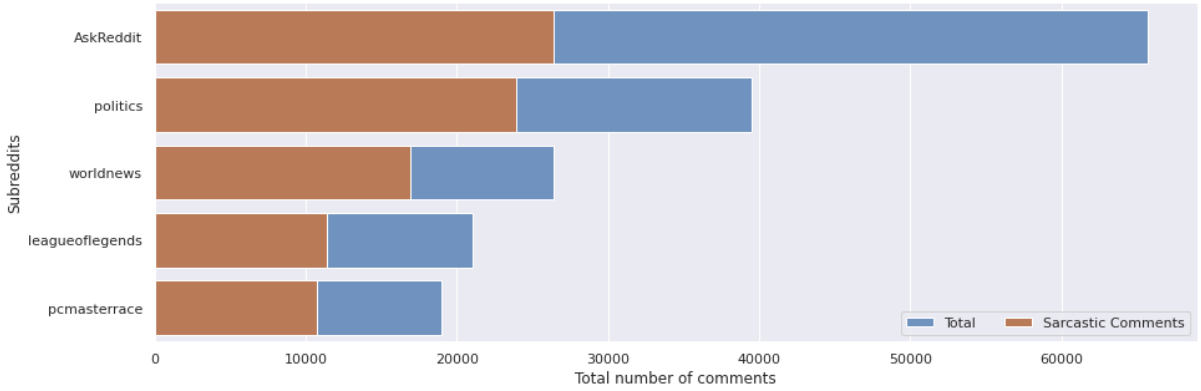


**Рис. 12 Топ 5 найпопулярніших топіків на сайті Reddit та співвідношення саркастичних коментарів до загальної кількості.**

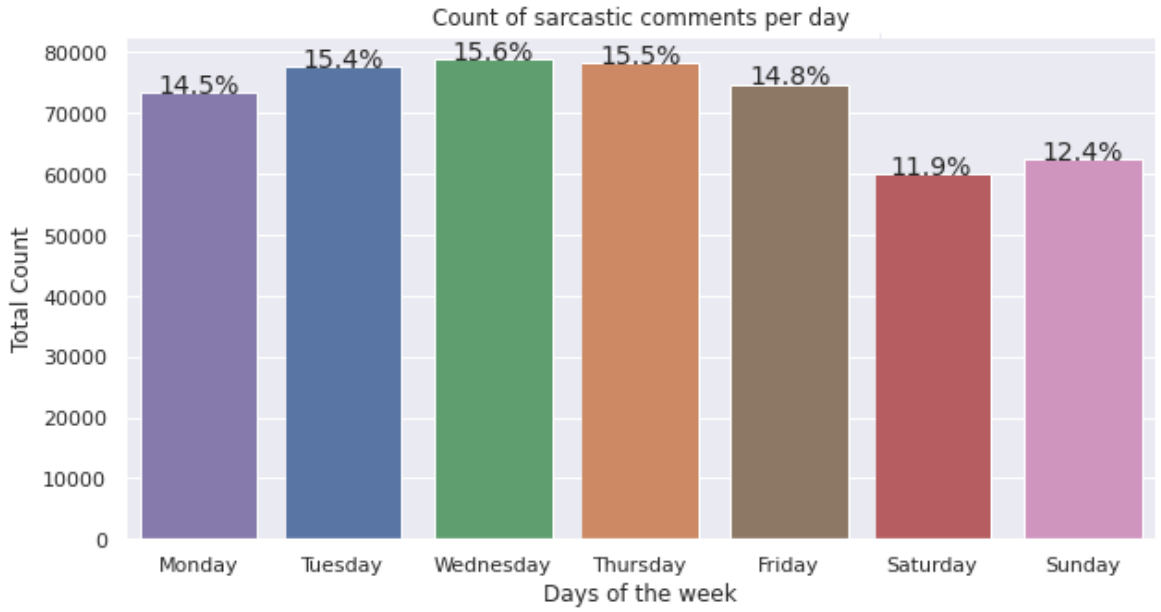


**Рис. 13 Кількість коментарів за певний день тижня**

Як бачимо з рисунка 12, найбільше саркастичних коментарів користувачі залишають середу та четвер, а найменше - на вихідних.

## 2.4 Архітектура запропонованої моделі

Тексти в соціальних мережах часто складаються з посилань на інші публікації, користувачів, гіперпосилання, смайликів та пунктуації. Для навчання аналізатора потрібно було очистити дані від подібної інформації, яку буде складно опрацювати моделі машинного навчання. Також були додані наступні етапи попередньої обробки даних:

- видалення стоп слів
- видалення пунктуації
- приведення тексту до нижнього регістру
- стамінг
- лематизація

З головної збалансованої навчальна вибірки датасету SARC було виділено валідаційний датасет. В розмірі 10 % від загальної кількості

розмічених текстів. В свою чергу тестова вибірка складає 15% від загальної кількості розмічених текстів.

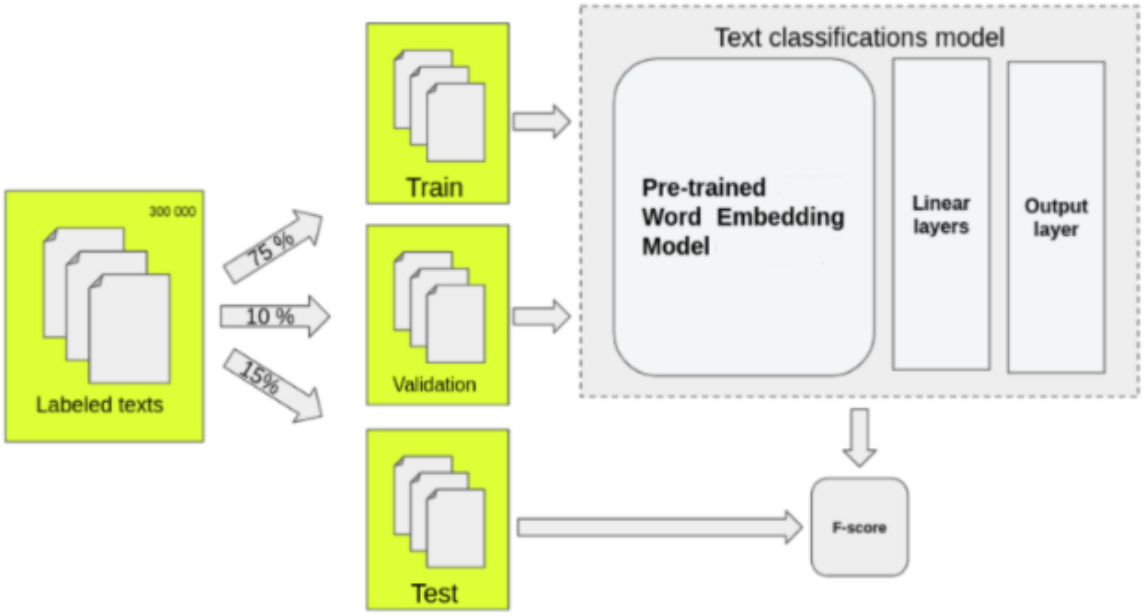


**Рис. 14 Послідовність кроків (Pipeline) навчання моделі запропонованої в результаті досліджень**

Запропонована під час дослідження архітектура текстового класифікатора складається з (Див. Рис. 14):

1) **Шару Ембендингів(Pre-trained GTP-2 Word Embedding Layer)** - створює вектор ембендингів для кожної послідовності слів.
2) **Повнозв'язних шарів (Linear layers)** - для перетворення даних у менші вектори функцій.
3) **Вихідного слою (Output layers) -** шар з вихідним вектором**.**

Параметри навчання даної моделі були підібрані наступні:

- learning_rate: 1e-05
- mini_batch_size: 8
- patience: 3
- anneal_factor: 0.5

- max_epochs: 10
- shuffle: True

## 3. Результати роботи розробленої моделі на наборі даних SARC

В результаті досліджень було розглянуто декілька архітектур класифікатора текстів, які включали в себе різні попередньо навчені трансформери для побудови векторів ембендингів, а саме:

- BERT
- GTP-2
- DialoGPT
- RoBERTa

Це одні з найкращих моделей для задачі представлення ембендингів для моделей класифікації текстів.

Для порівняння результатів, було обрано F1-міру, так як вона найкраще відображає якість моделей

| **Word Embedding Model** | **F1-score** |
|---|---|
| BERT(base) | 0.613 |
| BERT(large) | 0.542 |
| GPT-2(medium) | 0.586 |
| GPT-2(large) | 0.542 |
| DialoGPT(medium) | 0.603 |
| DialoGPT(large) | 0.558 |
| RoBERTa(base) | 0.607 |
| DistilBERT(base) | 0.549 |
| DeBERT(base) | 0.61 |

Більшість моделей показали хороші результати, зважаючи на те, що вони не враховували контекст розмови. Найкращою моделлю була модель на основі попередньо тренованого трансформера на наборі даних Reddit -

BERT(base), яка показала 61% F-міру на тестовій вибірці датасету. Дана модель простіше тренувалася, ніж аналогічна модель з більшою кількістю параметрів.

# ВИСНОВКИ

В ході написання даної кваліфікаційної роботи було проведено аналіз основних проблем розпізнавання сарказму і текстових повідомленнях, досліджено та проведено порівняльний аналіз основних методів, алгоритмів та інструментів класифікації текстів на вміст сарказму.

Проведено порівняльний аналіз доступних наборах розмічених даних. Для навчання було обрано публічний датасет SARC, так як він є найбільш підходящим для навчання і має велику кількість збалансованих текстів.

Розроблено та порівняно нейронні мережі на основі популярних архітектур трансформерів. Найкращою виявилася архітектура BERT(base) і показала 61% F-міру на тестовій вибірці датасету.

Доцільно й далі продовжувати дослідження методів розпізнавання сарказму в текстах, адже ця тема стає все більш актуальною з кожним роком і суттєво може покращити результати роботи методів сентимент аналізу. Також, варто застосувати побудову контекстних ембендингів та стилометричного аналізу користувачів, для покращення ефективності та отримання більш точних результатів роботи моделей.

# ПЕРЕЛІК ПОСИЛАНЬ


1. Jorgensen, Julia. "The functions of sarcastic irony in speech." Journal of pragmatics 26.5 (1996)
2. Лосев А.Ф., Шестаков В.П. Історія естетичних категорій. - 1965
3. Любимова Т. Б. Комическое, его виды и жанры / Т. Б. Любимова. – М.: Знание, 1990. – 64 с
4. Kreuz, Roger. Irony and sarcasm. MIT Press, 2020.
5. Riloff, Ellen, et al. "Sarcasm as contrast between a positive sentiment and negative situation." Proceedings of the 2013 conference on empirical methods in natural language processing. 2013.
6. Mulay, Preeti, et al. "Detection of Personality Traits of Sarcastic People (PTSP): A Social-IoT Based Approach." Internet of Things and Big Data Analytics for Smart Generation. Springer, Cham, 2019
7. Lee, Christopher J., and Albert N. Katz. "The differential role of ridicule in sarcasm and irony." Metaphor and symbol 13.1 (1998)
8. Pexman, Pamela. "How do we understand sarcasm?" Frontiers for Young Minds 6.56 (2018).
9. Kreuz, R., 2020. Irony and sarcasm. MIT Press.
10. Harris, M., and Pexman, P. M. 2003. Children’s perceptions of the social functions of verbal irony. Discourse Process. 36:147–65.
11. Shamay-Tsoory, S. G., Tomer, R., Berger, B. D., Goldsher, D., and Aharon-Peretz, J. 2005. Impaired “affective theory of mind” is associated with right ventromedial prefrontal damage. Cogn. Behav. Neurol
12. Francisco Rangel and Paolo Rosso. On the Impact of Emotions on Author Profiling. Information Processing & Management, 52(1):73 – 92, 2016. Emotion and Sentiment in Social and Expressive Media.
13. Feinman J. Sarcasm Detection in Text / J. Feinman, J. Kasakyan, J. Stolzenberg.

14. Hancock, Jeffrey T. "Verbal irony use in face-to-face and computer-mediated conversations." Journal of Language and Social Psychology 23.4 (2004)
15. Carvalho, P.; Sarmento, L.; Silva, M.J.; De Oliveira, E. Clues for detecting irony in user-generated contents: Oh...!! it's so easy. In Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion; Association for Computing Machinery: New York, NY, USA, 2009.
16. González-Ibánez, Roberto, Smaranda Muresan, and Nina Wacholder. "Identifying sarcasm in Twitter: a closer look." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011.
17. Riloff, E.; Qadir, A.; Surve, P.; De Silva, L.; Gilbert, N.; Huang, R. Sarcasm as contrast between a positive sentiment and negative situation. In Proceedings of the 2013 Conference on EMNLP, Seattle, WA, USA, 18–21 October 2013; pp. 704–714.
18. Joshi, A.; Sharma, V.; Bhattacharyya, P. Harnessing context incongruity for sarcasm detection. In Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP, Beijing, China, 26–31 July 2015; pp. 757–762.
19. Wallace, B.C.; Charniak, E. Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP, Beijing, China, 26–31 July 2015; pp. 1035–1044.
20. Amir, S.; Wallace, B.C.; Lyu, H.; Carvalho, P.; Silva, M.J. Modelling Context with User Embeddings for Sarcasm Detection in Social Media. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; pp. 167–177.
21. Rajadesingan, A.; Zafarani, R.; Liu, H. Sarcasm detection on twitter: A behavioral modeling approach. In Proceedings of the Eighth ACM

International Conference on Web Search and Data Mining, Shanghai, China, 2–6 February 2015; pp. 97–106.

22. Котельников Е.В. Распознавание эмоциональной составляющей в текстах: проблемы и подходы / Е.В. Котельников, М.В. Клековкина, Т.А. Пескишева, О.А. Пестов; Изд-во ВятГГУ, 2012.
23. Opinion Mining on the Web by Extracting Subject-Aspect-Evaluation Relations Nozomi Kobayashi, Ryu Iida, Kentaro Inui, Yuji Matsumoto – 2006
24. Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM." (1999).
25. Bing Liu. Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012.
26. A. Graves, M. Liwicki, S. Fernandez, J. Schmidhuber. A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009.
27. WANG, Rui, et al. Chinese NER with Height-Limited Constituent Parsing. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2019. - p. 7160-7167.
28. Dos Santos, Cicero, and Maira Gatti. "Deep convolutional neural networks for sentiment analysis of short texts. "*Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 2014.
29. Bouma, Gerlof. "Normalized (pointwise) mutual information in collocation extraction." *Proceedings of GSCL* (2009).
30. Graves, Alex; Mohamed, Abdel-rahman; Hinton, Geoffrey.Speech Recognition with Deep Recurrent Neural Networks 2013.

31. Ramos, Juan. "Using tf-idf to determine word relevance in document queries." *Proceedings of the first instructional conference on machine learning*. Vol. 242. 2003.
32. Feature Selection and Weighting in Sentiment Analysis O'Keefe, Tim; Koprinska, Irena – 2006.
33. A Comparison of the Accuracy of Support Vector Machine and Naıve Bayes Algorithms In Spam Classification– Rita McCue, Jürgen Schmidhuber – 29.11.2009.
34. Vaswani, Ashish, et al. "Attention is all you need." arXiv preprint arXiv:1706.03762 (2017).
35. https://nlds.soe.ucsc.edu/sarcasm2
36. Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." arXiv preprint arXiv:1706.03762 (2017).
37. Carvalho, P.; Sarmento, L.; Silva, M.J.; De Oliveira, E. Clues for detecting irony in user-generated contents: Oh...!! it's so easy. In Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion; Association for Computing Machinery: New York, NY, USA, 2009.