Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

Контент-аналіз україномовних текстів, написаних природньою мовою

Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення» - 121

Керівник курсової роботи
Кандидат фізико-математичних наук,
доцент
Жежерун О. П.

_____

(підпис)

“____” _____ 2021 р.

Виконав студент ІПЗ-4:
Брус А. І.

“____” _____ 2021 р.

Київ 2021

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

к.ф-м.н., доцент

О. П. Жежерун

_____

(підпис)

“____” _____ 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студента Бруса Андрія Ігоровича факультету інформатики 4 курсу

Тема: Контент-аналіз україномовних текстів, написаних природньою мовою

Зміст ТЧ до курсової роботи:

Календарний план

Вступ

Розділ 1. Основи контент-аналізу та алгоритми обробки текстів, написаних природньою мовою

Розділ 2. Аналіз тональності тексту як приклад застосування контент-аналізу

Розділ 3. Опис практичної частини та аналіз результатів

Висновки

Список використаної літератури

Дата видачі “____” _____ 2021 р. Керівник _____

(підпис)

Завдання отримав _____

(підпис)

Тема: Контент-аналіз україномовних текстів, написаних природньою мовою

Календарний план виконання роботи:

№	Назва етапу	Термін виконання	Примітка
1.	Отримання теми курсової роботи	25.10. 2020	
2.	Пошук технічної та наукової літератури за темою роботи	28.11. 2020	
3.	Ознайомлення з літературою	17.12. 2020	
4.	Огляд методології контент-аналізу, його видів та застосування	15.01. 2020	
5	Ознайомлення з поняттям та алгоритмами сентиментального аналізу	30.01.2020	
6.	Ознайомлення з бібліотеками TensorFlow, Keras, gensim	08.02. 2021	
7.	Формування корпусу для тестування	10.02. 2021	
8.	Розробка моделей	01.03. 2021	
9.	Написання текстової частини роботи	06.03. 2021	
10.	Перегляд змісту роботи керівником	29.03. 2021	
11.	Внесення змін до курсової роботи відповідно до зауважень наукового керівника	05.04. 2021	
12.	Створення презентації	10.04. 2021	

Студент Брус А. І.

Керівник Жежерун О. П.

“____” _____

Зміст

Вступ

Контент-аналіз, або ж аналіз змісту, вже більше ста років використовується для досліджень текстів та зображень, їх значення, прихованого змісту, схожості, належності до певних категорій. Контент-аналіз дозволяє видобувати з тексту інформацію про автора, його думки та ставлення до певних тем чи явищ, та використовувати цю інформацію в комерційних, наукових, політичних і маркетингових цілях. З розвитком соцмереж фокус контент-аналізу перемістився з дослідження ЗМІ на аналіз повідомлень, постів та коментарів. Тепер методи контент-аналізу все більше застосовуються для аналізу коротких текстів, а новітні технології все частіше використовуються для автоматизації контент-аналізу.

Актуальність теми. У сучасному світі значний відсоток комунікації між людьми ведеться саме в соцмережах та месенджерах. З огляду на це, все більшого попиту набувають задачі аналізу повідомлень, постів та коментарів, іншими словами – коротких текстів, написаних природньою мовою. Месенджери та соцмережі є ідеальним джерелом для аналізу поведінки людей, політичних та соціальних досліджень. Усе частіше з’являються спроби відслідковувати агресію, булінг та ксенофобію у соцмережах за допомогою автоматизованих рішень. Таким чином, такі напрями досліджень контент-аналізу, як сентиментальний та семантичний аналіз, є надзвичайно перспективними. У той же час, зважаючи на відносно невелику кількість робіт про обробку україномовних текстів, дослідження методів аналізу та обробки текстів, написаних українською мовою, досі є доволі актуальними.

Мета і завдання дослідження. Метою даної роботи є дослідження методології контент-аналізу та існуючі методи його автоматизації, розробка і порівняння ефективності моделей для аналізу тональності коротких текстів, написаних природньою мовою, та аналіз ефективності визначення більш тонких емоційних забарвлень текстів.

Завдання дослідження:

1. Дослідити поняття та методологію контент-аналізу;
2. Оглянути основні методи автоматизації контент-аналізу;
3. Дослідити існуючі архітектури та алгоритми для аналізу тональності текстів;
4. На основі обраних алгоритмів реалізувати системи для аналізу тональності короткого тексту на предмет наявності в ньому негативного емоційного забарвлення;
5. Протестувати систему та проаналізувати результати;

Наукове та практичне значення результатів. Теоретичні відомості та напрацювання можуть бути використані для подальших досліджень у цьому напрямку. Розроблені моделі можуть бути покращені та використані в системах дослідження тональностей україномовних повідомлень у соціальних мережах та месенджерах, а також для соціологічних досліджень на основі результатів такого аналізу.

Розділ 1. Основи контент-аналізу та алгоритми обробки текстів, написаних природньою мовою

1.1 Загальний огляд методології контент-аналізу

Контент-аналіз (букв. аналіз змісту) – один із найпоширеніших та найбільш широко визначених методів аналізу, кількісно-якісний метод, що спеціалізується на аналізі текстової та інколи графічної інформації з метою квантифікації (кількісному виділенні якісних ознак) текстів чи зображень та інтерпретації подальших результатів. Знаходиться на перетині соціології та лінгвістики, та вважається одним з основних методів дослідження в соціальних науках, проте із розвитком комп'ютерних технологій, мас-медіа та соціальних мереж контент-аналіз все частіше використовує такі новітні технології, як нейронні мережі (як частину NLP), Text mining (інтелектуальний аналіз тексту, напрям штучного інтелекту) та інші методи автоматизації та оптимізації досліджень.

У сучасному світі контент-аналіз набуває все більшого розповсюдження як метод аналізу тексту повідомлень та коментарів соцмереж, проте може бути застосований до будь-якого тексту, написаного природньою мовою (книги, дослідження, статті, новини, текстові повідомлення), а також до зображень та відеофайлів.

Контент-аналіз вирізняється строгістю процедури, проте завдяки цьому гарантує високий ступінь точності та об'єктивності аналізу. Основними кроками при проведенні контент-аналізу є [1]:

- Визначення основної одиниці аналізу – слово або значення слова, фрази, речення, теми, ситуації, описуваної текстом, повідомлення. Часто декілька одиниць розглядаються на різних етапах дослідження, а потім поєднуються для більш точного результату.
- Визначення основних категорій чи концепцій для подальшого визначення належності одиниць аналізу до однієї з цих категорій. Категорії мають бути

вичерпними та взаємовиключними, тобто кожна одиниця аналізу обов'язково має належати до однієї та тільки однієї категорії.

- Визначити, що шукати – наявність певної концепції чи її підрахунок (частоту, з якою вона зустрічається).
- Визначення правил належності до категорій чи відповідності концепціям, а також певних додаткових правил (чи враховується синонімічність, чи потрібна попередня обробка тексту тощо).
- Вирішити, чи повністю відкинути нерелевантну інформацію (стоп-слова, сполучники), та як саме її відкинути.
- Безпосередньо аналіз тексту – пошук слів, визначення їх значень, розподіл по категоріях, статистичний аналіз.
- Аналіз результатів – порівняння та інтерпретація отриманих результатів, внесення змін до алгоритмів та повторні дослідження.

Незважаючи на строгі вимоги, порядок не є фіксованим та може змінюватись в залежності від предмету та області дослідження. Новітні технології, такі як машинне навчання, нейронні мережі та попередня обробка тексту, також можуть замінити одразу декілька етапів дослідження.

1.2 Особливості контент-аналізу україномовних повідомлень

Контент-аналіз у класичному розумінні розроблявся з метою аналізу текстів у сфері ЗМІ. Основна методологія була розроблена на початку XX ст. для виявлення елементів впливу та небажаної пропаганди у журналах та газетах. Аналіз спілкування як прямого прояву природньої мови у ті часи був майже неможливим та використовувався лише зрідка для дослідження листувань. Проте з появою соціальних мереж та месенджерів у сучасному світі контент-аналіз повідомлень, відгуків та постів набуває все більшої популярності завдяки великому обсягу та доступності даних для різноманітних досліджень.

Згідно з визначенням Берельсона, одного з основоположників контент-аналізу, «контент-аналіз – метод, що досліджує об'єктивний, систематичний та кількісний опис явного змісту спілкування». Повідомлення користувачів

соціальних мереж (особисті повідомлення, пости, твіти, коментарі) використовуються для висловлення власної думки, позиції автора, часто мають емоційне забарвлення, а також дають уявлення про зовнішні події чи стани. Таким чином, вони є ідеальним джерелом застосування контент-аналізу.

Аналіз коротких текстів значно відрізняється від аналізу довгих текстів, що містять десятки речень, абзаци та тематичні розділи. Короткі тексти можуть складно піддаватись або взагалі не піддаватись категоризації. У той же час, глибинні нейронні мережі та векторні представлення документів краще працюють з даними невеликого розміру, тож їх застосування до коротких текстів може дати кращі результати в таких задачах, як семантичний аналіз.

Аналіз україномовних повідомлень є значно складнішим, ніж аналіз англомовних. Непрямий порядок слів ускладнює знаходження зв’язків між словами, а велика кількість форм слова (роди, відмінки, числа) часто потребує попередньої обробки для нормалізованого представлення слів. Також розвиток алгоритмів обробки україномовних текстів сповільнює нестача корпусів україномовних текстів та актуальних україномовних даних.

Англомовні дослідники часто використовують пости та коментарі із соцмережі Twitter для аналізу, оскільки Twitter надає офіційне API для стягування повідомлень та метаданих про них, що дозволяє дуже просто створювати великі корпуси даних. На жаль, в Україні Twitter непоширений, а Facebook не дозволяє стягувати інформацію про пости у тій мірі, як це дозволяє Twitter. Перспективним у цьому плані є месенджер Telegram, що має офіційне API для автоматизованої роботи з клієнтом Telegram. Популярність цього месенджеру стрімко зростає, та багато новинних та інформаційних ресурсів уже використовують Telegram в якості платформи для публікацій. Наявність API відкриває широкі можливості для використання даних каналів та повідомлень Telegram для формування україномовних корпусів даних, подібних до англомовних відповідників із Twitter.

1.3 Використання машинного навчання та штучного інтелекту для контент-аналізу

Natural Language Processing (NLP, обробка природньої мови) – один з основних напрямів штучного інтелекту в поєднанні з комп'ютерною лінгвістикою, що знайшов своє використання у контент-аналізі. Методи NLP можуть бути використані для контент-аналізу наступним чином:

- Data mining (видобування даних) – використання нейронних мереж, еволюційних алгоритмів та дерев рішень для кластеризації та класифікації даних [2, 3];
- Інформаційний пошук – нейронні мережі, векторно-просторові представлення, приховані семантичні індекси та нечіткі множини можуть використовуватись для пошуку та підрахунку входжень певних елементів, а також визначення прихованих елементів чи концепцій, на які спрямоване дослідження [4, 5, 6];
- Векторні представлення слів та речень – використовуються для визначення семантичних значень слів та синонімічних відносин між ними, що полегшує завдання класифікації та семантичного аналізу [7, 8].

Інтелектуальний аналіз тексту (в англомовному середовищі більш відомий як Text mining) – підрозділ Data mining, що спеціалізується на видобутку інформації з корпусів текстових документів, ефективно застосовуючи методи обробки природньої мови та машинного навчання. Саме термін text mining зазвичай асоціюється із поняттям контент-аналізу, хоча насправді визначення контент-аналізу відрізняється від інтелектуального аналізу тексту та є значно ширшим, хоч і включає в себе text mining [9, 10].

Такі методи представлення слів, як Bag Of Words та Word2Vec, широко використовуються в сучасному інтелектуальному аналізі та методах автоматизації контент-аналізу загалом.

Bag of words – представлення набору слів (речення, документу) у вигляді невпорядкованого набору слів, що не містить ніякої інформації про зв'язок між ними. Цей набір виглядає як вектор, де значення кожної позиції відповідає кількості екземплярів певного слова у цьому документі, а сама позиція ставиться у відповідність цьому слову. Мішок слів широко використовується у задачах класифікації, де частота виникнення певного слова виступає ознакою належності тексту до однієї з категорій.

Word2vec – один з методів векторного представлення, або вкладання, слів (word embeddings) – методика обробки природньої мови, представлена Томашем Міколовим та його командою дослідників у 2013 році під керівництвом компанії Google [7, 8]. В основі концепту вкладання слів лежить твердження про те, що семантику (значення) слова можна визначити із контексту, тобто із певної кількості слів навколо нього (перед та після).

Методики вкладання слів будують множини векторних представлень слів на основі вхідних текстів, де кожне унікальне слово представлене нормалізованим вектором певного розміру. З математичної точки зору, вкладання слів є математичним вкладанням багатовимірного простору слів, в якому кожний вимір відповідає певному слову, до неперервного векторного простору скінченної розмірності (зазвичай залежної від реалізації). Word2vec у своїй реалізації використовує двошарову нейронну мережу та розмірність векторного простору w=300 за замовчуванням як оптимальну [7].

Word2vec має дві реалізації векторного представлення слів:

- Skip-gram (словосполучення з пропусками) – визначає навколишній контекст на основі заданого слова із врахуванням порядку слів. Оптимальним вважається вікно розмірності n=5, тобто в загальному враховуються 11 (2n+1) слів включно із заданим.

- CBOW (Continuous Bag Of Words, неперервний мішок слів) – обернений до попереднього метод, що передбачає слово, виходячи з контексту (не враховуючи

порядок слів, як і всі реалізації мішків). Насправді являє собою звичайний мішок (вектор) розміру 2n+1, що включає передбачуване слово та n слів до і після нього.

В обох реалізаціях використовується нейронна мережа, що визначає коефіцієнти так, щоб мінімізувати косинусну міру подібності між словами, що зустрічаються в схожому контексті, а отже, ймовірно, мають схоже значення або спільні ознаки). Косинусна міра подібності P визначається так:

$$P = \frac{w_1 * w_2}{\|w_1\|\|w_2\|},$$

де w_1 та w_2 – вектори представлення слів.

Розділ 2. Аналіз тональності тексту як приклад застосування контент-аналізу

2.1 Загальний опис аналізу тональності тексту

Sentiment analysis (аналіз тональності) – один із найпоширеніших напрямів контент-аналізу з використанням NLP, що спеціалізується на аналізі емоційних забарвлень текстів та позицій авторів, виражених у цих текстах. Став широко використовуватись після того, як було доведено, що будь-який текст, написаний природньою мовою, певним чином відображає емоції, думки або позицію автора, а отже, може бути використаний для отримання цієї інформації [11, 12].

Незважаючи на те, що метод називається аналізом тональності, насправді він надає змогу знаходити не тільки тональні відтінки текстів (тобто ті, що прямо відображають ставлення автора до певного об’єкту), а й емоційні відтінки, що автор міг використати підсвідомо, та відповідність текстів певній тематиці.

Аналіз тональності стрімко розвивається завдяки наявності майже невичерпних джерел для досліджень – новинних ресурсів, блогів, текстів соцмереж та торгових платформ з можливістю коментування продуктів. Багато компаній стимулюють розвиток цих технологій, надаючи різноманітні рішення для отримання даних та їх обробки (Google, Twitter, IBM), що призводить до розробки все більш точних та ефективних методів аналізу [13].

Поряд із вже класичним аналізом тональності коментарів (позитивні/нейтральні/негативні), сентиментальний аналіз може вирішувати багато інших задач. У сучасному світі все більше уваги приділяється аналізу повідомлень у соцмережах на предмет наявності агресії, ксенофобії та булінгу [14, 15]. Аналіз тональності текстів може бути застосований для визначення ознак таких явищ за допомогою аналізу менш помітних емоційних забарвлень повідомлень.

Методи аналізу тональності поділяються на два типи за способом дії:

- На основі семантичних правил;
- На основі статистичних методів.

Перші використовують набори правил у сукупності зі словниками тональностей слів для визначення загальної тональності текстів. Проте їх використання ускладнюється тим, що тональність конкретного слова у реченні не завжди означає відповідну тональність цілого речення. Це потребує розробки та застосування нових лінгвістичних паттернів на основі морфологічно-синтаксичного аналізу (лем, словоформ та ін.). Додаткових правил обробки потребують також заперечні частки та слова (такі, як «не», «ніколи», «відмінний від», «протилежний» та ін.), що також впливає на час та складність розробки.

Методи другої групи використовують засоби машинного навчання, навчені з учителем та без учителя (англ. – supervised та unsupervised), такі, як наївні байєсові класифікатори та SVM, а також згорткові та рекурентні нейронні мережі. Ці засоби дозволяють створювати простіші моделі більш широкого використання, проте потребують розмічених корпусів текстів для навчання. Відповідно, результативність таких алгоритмів доволі сильно залежить від якості розмітки текстів.

У даній роботі було надано перевагу методам другої групи, оскільки вони не потребують глибоких знань та додаткових досліджень у сфері лінгвістики для коректної побудови правил та лінгвістичних патернів. Байєсів класифікатор є суто статистичним методом, тоді як SVM використовує неймовірнісний бінарний лінійний алгоритм з учителем.

Штучні нейронні мережі, такі, як рекурентні та згорткові, працюють за схожим до біологічних мереж принципом. При визначенні емоційного забарвлення тексту людина часто визначає ознаки цього забарвлення підсвідомо, знаючи тональність певних слів та уловлюючи зміну цієї тональності в залежності від контексту. Штучні нейронні мережі моделюють роботу мозку, тож також здатні самостійно вловлювати вагу тональностей певних слів та враховувати контекст.

2.2 Наївний байєсів класифікатор

Наївний байєсів класифікатор (НБК) – метод статистичної класифікації на основі теореми Байєса. Цей алгоритм є одним з найпростіших та найшвидших, і показує високу точність на даних великого обсягу.

Ідеальний байєсів класифікатор однозначно класифікує текст, якщо таке можливо, а в протилежному випадку повертає вектор, що містить імовірності належності тексту до кожного з можливих класів. Проте для коректної роботи байєсів класифікатора потрібен корпус даних, що міститиме усі можливі комбінації ознак. Оскільки кількість комбінацій зростає експоненційно зі збільшенням кількості ознак, а в природній мові кількість таких ознак неможливо обрахувати точно, до того ж, в деяких випадках ця кількість може прямувати до нескінченності, практичне застосування ідеального байєсового класифікатора до обробки природніх текстів є дуже обмеженим та часто недоцільним.

НБК використовує припущення про те, що вплив певної ознаки на результат аналізу ніяк не залежить від інших ознак (використання цього припущення і робить його наївним). Це дозволяє знехтувати усіма можливими комбінаціями ознак, та робити висновки виключно на основі впливу конкретної ознаки на ймовірність належності до певного класу. Таке припущення значно спрощує реалізацію алгоритму та підвищує його швидкодію, проте ігнорування відносин між ознаками негативно впливає на точність класифікації.

НБК працює на основі теореми Байєса та вираховує ймовірність події *P*, у даному випадку належності до певного класу, за формулою:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$, де:

- *P(h)* – ймовірність того, що гіпотеза h є вірною, незалежно від даних D (апріорна ймовірність),

- *P(D)* – ймовірність появи даних *D* незалежно від гіпотези (апріорна ймовірність),
- *P(h|D)* – ймовірність того, що гіпотеза *h* є вірною при даних *D* (постеріорна ймовірність),
- *P(D|h)* – ймовірність появи даних *D* за умови, що гіпотеза *h* є вірною (постеріорна ймовірність),

Для визначення конкретного класу НБК використовує правило постеріорного максимуму, тобто підраховує значення ймовірності кожного класу для документу та вибирає найбільше значення. Результуючий клас носить назву класу постеріорного максимуму, а його значення $\hat{h}$ для кожного документу, виходячи з попередньої формули, обраховується наступним чином:

$$\hat{h} = \underset{h_j}{argmax} \prod_{k=1}^{p} P(d_k|h_j)P(h_j), \text{де:}$$

- p – кількість ознак, що враховуються при класифікації,
- d_k – k-й документ в колекції,
- $P(h_j),$ – ймовірність появи ознаки класу h_j
- $P(d_k|h_j)$ – ймовірність появи ознаки, відповідної класу h_j, в документі d_k,

Оскільки припущення про незалежність ознак одна від одної робить $P(D)$ константою, ця константа може бути упущена, та зазвичай використовується лише для нормалізації результуючих даних (до прикладу, приведення результату у відповідність проміжку [-1, 1], де -1, 0 та 1 – три класи емоційного забарвлення: негативне, нейтральне та позитивне).

2.3 Supporting vector machine

Supporting vector machine (SVM, метод опорних векторів) – лінійний алгоритм машинного навчання, що часто застосовується в задачах класифікації. Суть цього методу полягає в тому, що він навчається знаходити гіперплощину, яка розділяє векторний простір, у якому знаходяться представлення даних, на

частини, що відповідають класам. Зазвичай використовується для поділу на два класи та, відповідно, шукає одну гіперплощину, проте існують модифікації, що дозволяють розподіл на три та більше класів (хоча це супроводжується зменшенням точності результатів) [16].

У *n*-вимірному Евклідовому просторі гіперплощина – це (*n*-1)-вимірний підпростір, що розділяє цей простір на дві частини. Таким чином, при роботі з векторними представленнями слів із класичної реалізації word2vec ці слова формуватимуть Евклідів простір розмірності *n*=300, тоді як SVM шукатиме для цього векторного простору гіперплощину розмірності *n*=299.

Нехай існує задача розподілу даних на два класи – негативний і позитивний. Векторні представлення (розмірності *n*=2) елементів, що підлягають класифікації, можна представити графічно, як показано на рисунку 1:

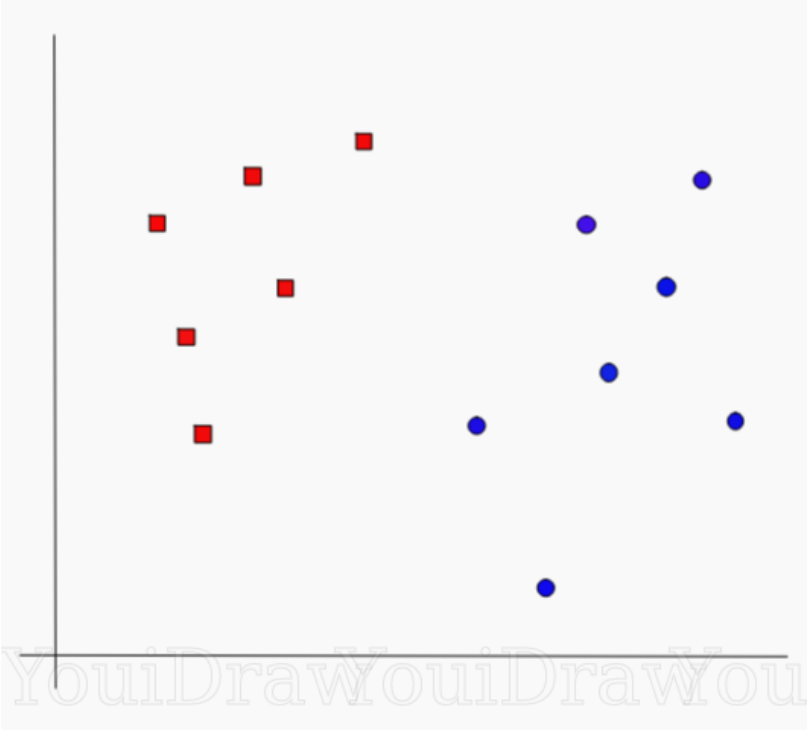

Рис.1

Таким чином, точки, позначені червоними квадратами, відповідають за позитивне забарвлення, а синіми колами – за негативне. Тоді завданням SVM буде знайти оптимальну гіперплощину, що розділить ці дві сукупності точок з максимальною точністю. У цьому випадку гіперплощина виступатиме прямою.

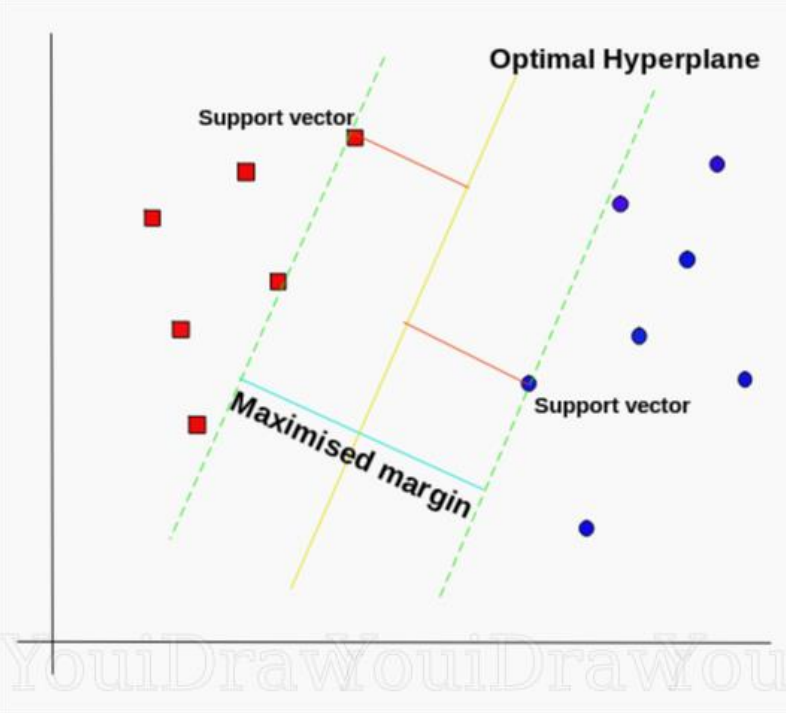

Рис. 2

Під час роботи SVM при ініціалізації створює довільну гіперплощину, а потім шукає так звані опорні вектори – такі точки у векторному просторі, які б знаходились найближче до існуючої гіперплощини розділення. Наступним кроком алгоритм вираховує відстань (або проміжок) між опорними векторами та площиною розділення. SVM намагається максимізувати цю відстань та скорегувати координати гіперплощини, а гіперплощина з найбільшим проміжком буде вважатись оптимальною (рисунок 2).

Таким чином, при навчанні SVM намагається знайти таку гіперплощину, що якомога більше розділяла б два класи та біла рівновіддаленою від них. Проте, при роботі зі словами чи документами їх векторні представлення навряд чи можуть бути розподілені в такому зручному вигляді. Часто виникає ситуація, описана на рисунку 3, коли дані неможливо розділити лінійно.

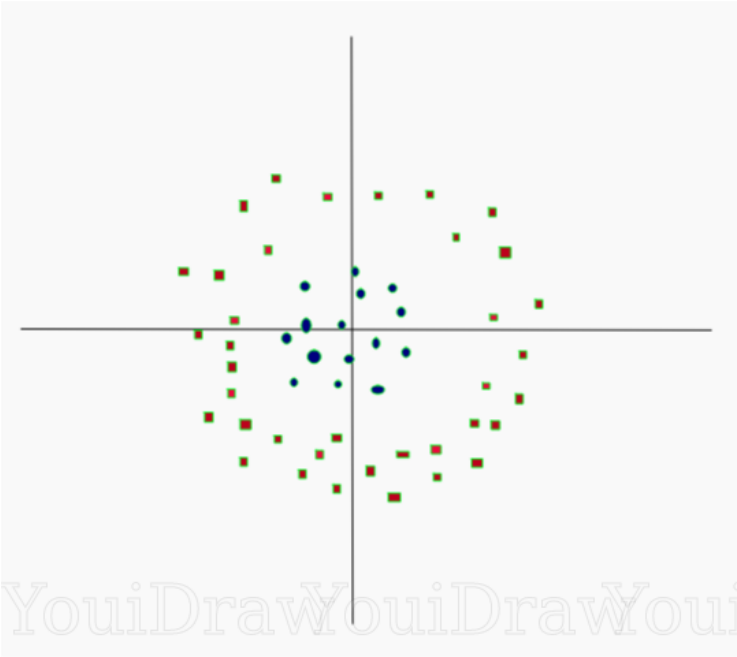

Рис. 3

У таких випадках, SVM розширює початковий простір та будує додатковий вимір, а також підбирає правило, за яким довільно розподілені елементи початкового *n*-вимірного простору у (*n*+1)-вимірному просторі можуть бути розділені на дві частини. Таким чином, для рисунку 3 можна додати третій вимір (вісь Oz), а координати елементів на ній визначити наступним чином:

$$z = x^2 + y^2,$$

де *x* та *y* – координати елементу у початковому 2-вимірному просторі. Це дасть SVM змогу розділити точки так, як показано на рисунку 4:

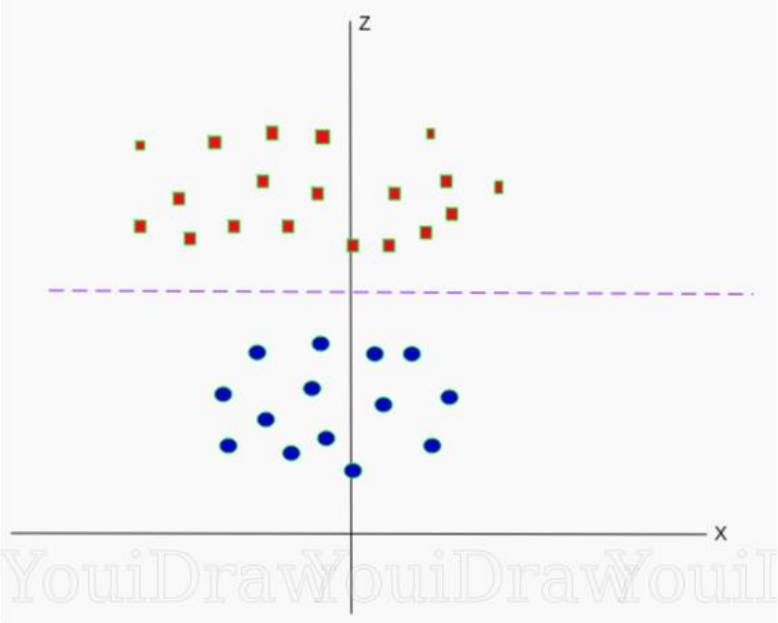

Рис. 4

Незважаючи на складність знаходження правильної функції для потрібного перенесення координат у (n+1)-вимірний простір, існує велика кількість евристичних методів, що можуть бути вбудовані в SVM та автоматично визначати потрібну функцію з високою точністю [17].

2.4. Рекурентні нейронні мережі

Рекурентні нейронні мережі (англ. – Recurrent neural networks, RNN) – вид рекурсивних нейронних мереж, що «пам’ятає» свій попередній стан та робить обчислення з урахуванням цього стану. Кожен нейрон RNN містить прихований стан, що записується при поточному проходженні та буде використаний цим нейроном на наступній ітерації. Архітектура RNN стала першою, що дала нейронним мережам примітивну функцію пам’яті, схожу з пам’яттю нейронів людини. Рекурентна структура RNN дозволяє опрацьовувати послідовності даних нефіксованої довжини, що є однією з основних причин успішності цієї архітектури в задачах обробки та класифікації текстів, написаних природньою мовою.

В розгорнутому стані структура RNN виглядає наступним чином (рисунок 5):

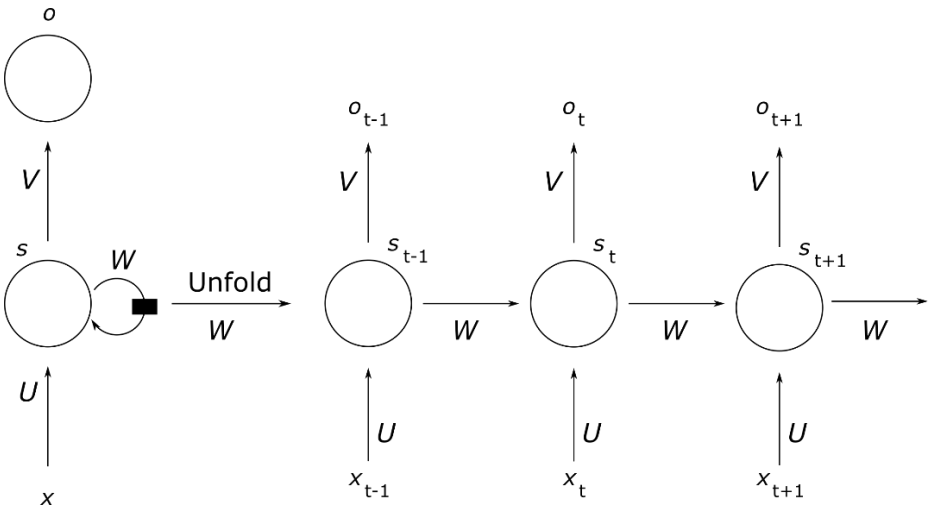

Рис. 5

Таким чином, в розгортці RNN має вигляд *n*-шарової мережі, де кожен шар відповідає за одне зі слів вхідної послідовності довжини *n*. Основні елементи RNN, показані на рисунку 5:

- x_t – вхідні дані на кроці *t*;
- U – матриця ваг для вхідних даних поточного стану;
- s_t – прихований стан на кроці *t*, вираховується за формулою:

$$s_t = \sigma(Ux_t + Ws_{t-1})$$

- W – матриця ваг із попереднього стану;
- σ – функція активації;
- o_t – вихідні дані на кроці *t*.

Нульовий вектор s_0 вводиться для обрахування першого прихованого кроку.

Контроль похибки та тренування здійснюється за допомогою доповненого зворотного поширення (англ. – back propagation) – алгоритму зворотного поширення у часі (англ. – back propagation through time), що використовує метод схоластичного градієнтного спуску та рухається не назад, як back propagation, а вздовж розгорнутої у часі мережі.

Основна проблема RNN полягає у тому, що прихований стан зберігає пам’ять лише про попередню ітерацію. Протягом навчання після декількох ітерацій пам’ять про першу з них буде повністю перезаписана. Це погіршує якість семантичного та тонального аналізу тексту, оскільки існує загроза, що RNN прийматиме остаточне рішення класифікації лише на основі останніх частин тексту, ігноруючи дані про початок документу. Ця проблема носить назву зникаючого та вибухаючого градієнту.

2.5 LSTM

LSTM (Long-Short Term Memory) - модифікація RNN, розроблена з метою подолати проблему зникнення та вибуху градієнту. При кожній ітерації використовується так званий забувальний фільтр (англ. – forget gates), що вирішує, чи варто нейрону перезаписувати збережений стан, чи знехтувати поточним станом на користь пам’яті про минулий.

Нейрон LSTM містить у собі не просто інформацію про свій попередній стан, а окрему комірку пам’яті, що може зберігати інформацію впродовж усього процесу навчання. Це також дозволяє краще виявляти зв’язки між віддаленими частинами речення. Така властивість LSTM є особливо цінною з огляду на те, що в українській мові можливий непрямий порядок слів, тож пов’язані між собою слова можуть розташовуватись у протилежних кінцях речення.

2.6 Згорткові нейронні мережі

Згорткові нейронні мережі (Convolutional neural networks, CNN) отримали свою назву через використання шару з операцією «згортки» (англ. – convolution layer). Вперше використана Алексом Крижевським у 2012 році для конкурсу ImageNet з метою класифікації об’єктів на фото. Розробка Word2Vec у 2013-2014 роках дала можливість застосовувати операції згортки до вхідних послідовностей слів, представлених у векторному вигляді, що зробило CNN однією з найпопулярніших та найбільш ефективних нейронних мереж для класифікації, тонального та семантичного аналізу.

Принцип роботи CNN розроблений під впливом теорії рецептивних полів та принципу роботи клітин мозку людини з органами зору. Рецептивні поля – частини вхідної інформації про зображення, які обробляє нейрон замість того, щоб обробляти все зображення одразу. Рецептивні поля сусідніх нейронів частково перетинаються, та при об'єднанні дозволяють отримати загальне враження про вхідне зображення.

CNN зазвичай являє собою поєднання згорткових та агрегувальних шарів, та, за потреби, повноз'єднаних прихованих шарів для зведення усіх результатів.

Приклад архітектури CNN наведено на рисунку 6:

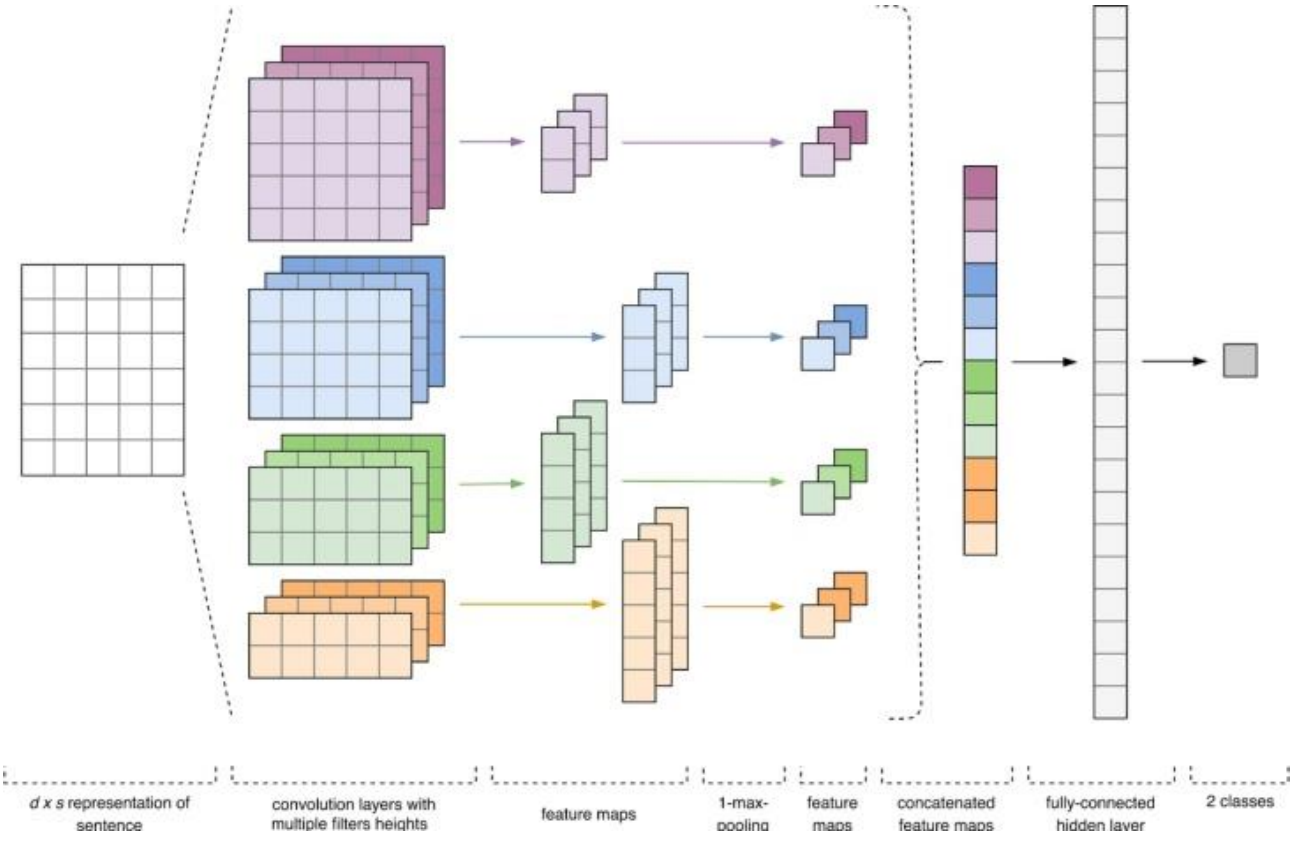

Рис. 6

2.6.1 Згортковий шар

У цьому шарі, як і в мозку людини, нейрон отримує вхідну інформацію не від усіх нейронів попереднього шару, як у класичних нейронних мережах, а тільки від певної її частини. Такий підхід відповідає принципу роботи людського мозку, який сприймає не весь текст одразу, а лише декілька слів, розташованих поруч

(словосполучення, слово), формуючи відповідний контекст та конкретне уявлення про нього. Надалі вже уявлення про контексти та значення речень в сукупності дозволяють сформулювати уявлення про абзац, а сукупність абзаців – про весь текст.

Вхідна послідовність слів представляється як матриця *S:*

$$S \in R^{w \times l}, \text{ де}$$

- l – довжина вхідної послідовності,
- w – розмірність векторного представлення слова

Операція згортки використовує матрицю $C \in R^{w \times n}$ до вікна розміром у n слів для усіх можливих вікон у реченні, а значення c_i для вікна $S[*, i, i+n]$ вираховується наступним чином:

$$c_i = \sigma\left(\sum (S[*, i, i+n] \circ H) + b\right), \text{ де}$$

- σ – нелінійна функція активації,
- $\circ$ - добуток Адамара,
- b – параметр упередженості

На виході шар згортки видає карту ознак c:

$$c = [c_1, c_2, \dots, c_{l-n+1}], \ c \in R^{l-n+1},$$

У шарі згортки параметрами ваг виступають розміри матриці згортки, в той же час розмірність самої матриці та параметри упередженості є спільними для усіх нейронів. Це призводить до мінімізації кількості вільних параметрів, а також робить неможливим зникнення та вибух градієнту, що є поширеною проблемою при використанні зворотного поширення помилки.

2.6.2 Агрегувальний шар

Агрегувальний шар (англ. – pooling) слугує для зменшення розмірності оброблюваних даних. Він отримує на вхід карту ознак із згорткового шару та визначає найбільш важливі ознаки за допомогою функції максимальної агрегації.

Для того, щоб втрати при застосуванні шару агрегації були якомога меншими, зазвичай використовують найменший можливий фільтр розмірності 2. Операція агрегації застосовується наступним чином:

$$p_i = \max(c_{2\times i-1}, c_{2\times i})\text{, де}$$

- $c_{2\times i-1}$ та $c_{2\times i}$ – ознаки з вхідної карти ознак,
- p_i – результуюча ознака.

Результуюча карта ознак після проходження цього шару виглядатиме так:

$$p = \left[p_1, p_2, \dots, p_{\frac{l-n+1}{2}}\right],$$

$$\text{де } p \in R^{\frac{l-n+1}{2}}$$

Зменшення розмірностей обчислюваних даних та, відповідно, кількості обчислень також слугує для контролю навчання моделі.

2.6.3 Повноз'єднаний та класифікаційний шари

Повноз'єднаний шар (fully connected layer) використовується для зведення усіх результатів воєдино. Кожен нейрон повноз'єднаного шару пов'язаний з усіма нейронами попереднього шару. Шар-класифікатор (classification layer) застосовується для приведення результату до цілого числа, що відповідатиме певному класу (зазвичай шляхом округлення або використання спеціальних функцій). Шар втрат (loss layer) інколи застосовується перед шаром класифікатором для регуляції втрат та відхилень.

Розділ 3. Опис практичної частини та аналіз результатів

3.1 Постановка завдання та опис з точки зору контент-аналізу

Метою цієї роботи є аналіз ефективності різних алгоритмів аналізу тональності коротких текстів, а також їх можливість вловлювати наявність тонких емоційних забарвлень незалежно від контексту вживання.

Враховуючи методологію контент-аналізу, завдання аналізу тональності коротких текстів можна описати наступним чином.

Основними одиницями аналізу у даних дослідженнях є речення (як вид короткого тексту) та слово. Тоді як на аналіз речення спрямоване саме дослідження, аналіз слів необхідний на етапі формування списку стоп-слів та формування корпусу текстів для досліджень.

Основною концепцією, на наявність якої аналізуватимуться одиниці аналізу, є негативне емоційне забарвлення. Відповідно, речення аналізуватимуться на відповідність одній із двох категорій:

- такі, що містять негативне емоційне забарвлення
- такі, що не містять негативного забарвлення

Негативно забарвленим реченням вважається таке речення, в якому явно відчувається негативне ставлення автора до певного явища, події або теми.

Оскільки наявність негативного емоційного забарвлення не може бути визначена лише за допомогою пошуку певних забарвлених слів, а частіш за все визначається людиною інтуїтивно виходячи з контексту, побудова чітких правил для розподілу є доволі складним завданням.

Під час формування тестового корпусу даних та розмітки текстів були сформовані наступні правила, за якими аналітик може віднести речення до категорії негативно забарвленого тексту:

1. наявність слів, що явно свідчать про негативне забарвлення речення (до прикладу, «ненавиджу», «недолюблюю», «зневажаю»).
2. контекст, що явно свідчить про відповідне ставлення автора, за відсутності явно негативно забарвлених слів.
3. Прояви агресії та погрози варто відносити до негативно забарвлених текстів.

В інших же випадках речення варто відносити до категорії текстів, що не містять негативного забарвлення.

Використання останніх двох правил значно підвищує суб'єктивність рішення, оскільки засноване на сприйнятті особи, що проводить аналіз. Проте, оскільки контент-аналіз сам по собі є суб'єктивним методом дослідження, такі рішення допустимі [18, 1].

При автоматизованому аналізі тексту є доцільним використання словника стоп-слів. Враховуючи тематику дослідження, виникає потреба в розробці спеціалізованого словника стоп-слів, що не міститиме таких часток та вставних слів, як «не», «ніколи», «дуже», що зазвичай не потрібні при задачах визначення тематики тексту та його класифікації, проте можуть сильно впливати на тональність текстів.

Практична частина включає в себе розмітку тренувального корпусу, аналіз поширених стоп-слів та створення власного словника, а також реалізація та тестування чотирьох описаних моделей для аналізу тональності коротких україномовних текстів та їх класифікацію відповідно до наявності обраної тональності, та порівняння результатів роботи цих моделей.

3.2 Опис використаних технологій та корпусів даних

З огляду на досвід розробки та кількість джерел і засобів розробки, було обрано мову програмування Python. Бібліотека genism використовувалась для обробки моделей векторних представлень слів.

Для розробки моделі CNN було використано бібліотеки Tensorflow для роботи з засобами машинного навчання та Keras для роботи з шарами нейронних мереж, оскільки остання містить вбудовані реалізації усіх необхідних шарів та багатьох функцій активації.

Для розробки моделей SVM та НБК використано бібліотеку sci-kit learn, що містить велику кількість реалізацій алгоритмів машинного навчання, в тому числі й класів для SVM, класичного та модифікованих алгоритмів НБК.

Для тренування моделей використовувалась готова україномовна модель word2vec, яку надає портал lang-ua [19]. Токенізація аналізованих текстів реалізована за допомогою LanguageTool uk для Python, API, що розроблене спеціально для української мови.

Список стоп-слів було сформовано вручну. У нього ввійшли сполучники, деякі частки та вставні слова. До нього не ввійшли такі слова, як «навіть», «за вийнятком», «зовсім» та інші вставні слова і частки, що можуть впливати на емоційне забарвлення тексту.

Основний датасет також сформовано та розмічено вручну. До нього ввійшли випадково вибрані речення із корпусу текстів порталу lang-ua, а також відібрані речення з постів та повідомлень у соцмережі Telegram. Розмір корпусу – 600 речень, з них 392 позначені як нейтральні, а 208 – з негативною коннотацією.

3.3 Попередня обробка текстів

Для кожної з моделей перед аналізом кожен текст проходить попередню обробку, що включає в себе приведення усіх слів до нижнього регістру та токенізацію тексту. Для токенізації тексту були використані скрипти LanguageTool API [20].

Зазвичай перед аналізом тексту засобами машинного навчання та нейронними мережами проводиться також лематизація та стеммінг даних, що дає змогу зменшити об’єм словника та краще розпізнавати схожі терміни. Проте в задачах аналізу тональності тексту лематизація та стемінг радше шкодять, адже різні

форми слова часто можуть нести різне тональне забарвлення (до прикладу, слово «люблю» однозначно свідчить про позитивне ставлення, тоді як слово «любила» може мати негативний відтінок). Тому у даних дослідженнях стеммінг та лематизація не проводились.

Розмір тренувальної вибірки для всіх моделей складає 80% від загального обсягу розмічених текстів, тестової – 20%. Для розподілу тренувальної вибірки використано метод train_test_split() із бібліотеки scikit-learn.

3.4 Підготовка моделей векторних представлень слів

Для навчання моделей SVM, CNN та CNN-LSTM використовується україномовна модель word2vec, що знаходиться у відкритому доступі на порталі lang-ua [19]. Оскільки завжди є ймовірність появи в текстах тренувального корпусу нових слів, відсутніх у готовій моделі (або ж просто слів з помилками), постає необхідність якимось чином враховувати їх. Зважаючи на малу вибірку даних, повторне тренування моделі на тому ж корпусі даних дасть недостовірні результати, тож було вирішено обрати метод ініціалізації нових слів випадковими векторами.

Для пришвидшення обробки кожне слово тестового датасету переводиться у числове представлення з використанням словника та id терміну у цьому словнику. Власна векторна модель, що використовуватиметься для навчання моделей, далі ініціалізується випадковими числами за законом нормального розподілу. Наприкінці, словам із власної векторної моделі, що присутні у готовій векторній моделі, копіюються із неї відповідні значення.

Таким чином, відомі слова отримують коректні значення, а невідомі залишаються випадково ініціалізованими. В подальшому при розблокуванні embedding-шару значення невідомих слів будуть скореговані моделлю, що має призвести до підвищення точності результатів.

3.5 Опис моделей аналізу

3.5.1 НБК

Для реалізації НБК було обрано бібліотеку scikit-learn, а саме клас Multinomial NB (Multinomial Naïve Bayes) – одну із стандартних реалізацій наївного байєсового класифікатора, оптимізовану для обробки мультиноміально розподілених даних.

У якості функції втрат використовується функція zero-one, що визначає похибку як кількість неправильних результатів.

Результат передбачень порівнюється із розміченими даними, та класифікуються наступним чином:

- TP – True Positive, кількість правильно визначених позитивних результатів
- FP – False Positive, кількість результатів, визначених позитивними, які насправді є негативними
- TN – True Negative, кількість вірно визначених негативних результатів
- FN – False Negative, кількість визначених негативними результатів, що насправді є позитивними

Метрики, що використовуються для оцінки точності моделі:

- Precision – відношення правильно визначених позитивних результатів до загальної кількості визначених позитивно результатів:

$$precision = \frac{TP}{TP + FP}$$

- Recall – відношення правильно визначених позитивних результатів до всіх позитивно відмічених:

$$recall = \frac{TP}{TP + FN}$$

- Accuracy - відношення кількості вірно визначених результатів до кількості усіх експериментів:

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

3.5.2 SVM

Модель SVM також побудована на відповідній реалізації з бібліотеки scikit-learn. Оскільки дані розподіляються лише між двома категоріями, було обрано клас LinearSVC (Linear Support Vectors Classifier).

У якості функції втрат використовується $L1$ – функція найменшого абсолютного відхилення, що визначається як:

$$L1 = \sum_{i=1}^{n} \left| Y_{\text{р}} - Y_{\text{п}} \right|, \qquad \text{де:}$$

- $Y_{\text{р}}$ – реальне значення,
- $Y_{\text{п}}$ – передбачене значення

Метрики, використані для оцінки точності – accuracy, precision, recall.

3.5.3 CNN

Серед розглянутих у другому розділі моделей аналізу тональності текстів згорткова нейронна мережа є єдиною, яка має обмеження на розмір вхідних даних, оскільки приймає на вхід матрицю розмірності $w \times l$, де w – розмірність векторного простору вкладень слів, а l – довжина вхідної послідовності. Для даного дослідження було обрано максимальну довжину послідовності $l = 21$ на основі досліджень середньої довжини речень в українській мові [21].

Шар вхідних даних (input layer) ініціалізується з параметром розмірності вхідних даних ($l = 21$).

Ваги першого шару моделі (embedding layer) ініціалізуються готовою моделлю word2vec та надалі блокуються, щоб при подальшому навчанні результати роботи цього шару залишались незмінними. Для запобігання перенавчання наступним додано шар втрат, що використовує dropout-регуляризацію з ймовірністю оновлення вершини $p = 0{,}2$.

Наступними ідуть по 10 згорткових шарів для кожної з розмірностей згорткових матриць: $h = 2, 3, 4, 5$. Це дозволить моделі паралельно обробляти 2-, 3-, 4- та 5-

грамми. Параметри обрані на основі результатів дослідження оптимальної кількості шарів та висоти матриць, описаних у статті [22].

Після кожної десятки згорткових шарів використовується 10 шарів агрегації. Функція активації шарів згортки та активації – ReLU. Потім додано іще один шар втрат, аналогічний вже доданому, а після нього – повнозв'язний шар розмірності $n = 24$ нейрони.

Результуюча карта ознак передається останньому, вихідному, шару із сигмоїдальною функцією активації, що повертає або 0, або 1 (де 1 – наявність негативного забарвлення, 0 – його відсутність).

Бінарна крос-ентропійна функція була використана в якості функції втрат, метрики результативності – accuracy, precision, recall.

3.5.4 CNN+LSTM

Архітектури CNN та LSTM можуть бути поєднані для покращення результату [23]. Таке поєднання дозволяє одночасно охоплювати декілька сусідніх ознак (слів) за допомогою CNN та при цьому пам'ятати ознаки, що вже давно вийшли за межі «поля зору» CNN.

У цій архітектурі одразу ж після embedding-шару додається одновимірний згортковий шар, а одразу ж після нього – шар агрегації, що далі з'єднуватиметься з шаром LSTM.

Вхідний шар такий же, як у попередній архітектурі, та ініціалізується з такою ж розмірністю вхідних даних ($l = 21$). Перший embedding-шар також повністю ідентичний описаному у попередньому пункті.

Наступним додано шар згортки з розмірністю матриці згортки h=3. Функція активації – ReLU. Одразу ж після нього використовується шар агрегації.

Потім додається вже класичний рекурентний шар LSTM, що ініціалізується з параметром m=150 , де m – кількість комірок пам'яті, що містить цей шар. Останнім є однонейронний вихідний шар із сигмоїдальною функцією активації.

Функція втрат - бінарна кросс-ентропійна, метрики – accuracy, presicion, recall.

3.6 Аналіз результатів

Тестування усіх моделей проводилось на власноруч розміченому корпусі нелематизованих речень.

Для досягнення максимальної ефективності навчання CNN та CNN-LSTM проводилось у 150 епох для кожної моделі. Потім визначалась найбільш результативна епоха, для неї розблоковувався embedding-шар та проводилось додаткове тренування із 20 епох. Це дозволило підвищити точність передбачень моделей.

Модель	accuracy	precision	recall
LinearSVC	0.54953	0.452	0.282
Multinomial Naïve Bayes, BOW	0.54782	0.411	0.2016
Multinomial Naïve Bayes, word2vec	0.55124	0.4959	0.4123
CNN	0.5711	0.6102	0.547
CNN з розблокованим embedding-шаром	0.60452	0.60502	0.4857
CNN-LSTM	0.5835	0.5211	0.5134
CNN-LSTM з розблокованим embedding-шаром	0.61549	0.593	0.4332

Таблиця 1.

З таблиці 1 видно, що нейронні мережі досягли кращих результатів, ніж наївний байєсовий класифікатор та метод опорних векторів. Використання векторних представлень слів для НБК незначною мірою покращило результати у порівнянні з мішком слів.

Нейронні мережі показали кращі результати при виборі найкращої епохи та розблокування шару векторного представлення. Загалом, CNN-LSTM з розблокованим embedding-шаром показала найкращий результат. Це свідчить про те, що модифакація класичної CNN із додаванням шарів LSTM справді покращує результат. Проте, для більшої достовірності результатів все ж варто

проводити дослідження на корпусах більшої розмірності. На жаль, на момент написання роботи великих корпусів, розмічених для сентиментального аналізу, ще не створено.

Низькі показники перш за все спричинені невеликим розміром тестової колекції. Суб'єктивна класифікація речень при формуванні сету даних також могла мати незначний вплив на точність передбачень. Іще одним фактором, що міг вплинути на результат, є специфічна тематика дослідження, коректність визначення і застосування правил при визначенні належності речення до категорії.

Наступні дії можуть покращити результат:

- Зміна архітектур нейронних мереж, кількості та порядку шарів
- Використання інших функцій втрат
- Зміна чи заміна правил та розширення навчального корпусу
- Використання семантичних та лінгвістичних правил, їх комбінація з розглянутими алгоритмами

Незважаючи на низькі результати, усі алгоритми продемонстрували можливість вірно передбачати категорію у більш ніж половині випадків, що вже робить їх успішними, хоч вони і досі значно програють людському аналізу.

Висновки

У даній роботі досліджено методологію контент-аналізу основні алгоритми сентиментального аналізу. На практиці розглянуто використання алгоритмів аналізу тональності текстів як одного із засобів контент-аналізу текстів. Розглянуто перспективи використання контент-аналізу та аналізу тональностей для обробки повідомлень соцмереж та месенджерів. Описано основні проблеми обробки української природної мови. Також створено власний дата сет для аналізу, який може бути розширений та покращений для підвищення якості натренованих на ньому моделей.

Детально розглянуто та реалізовано чотири моделі для тонального аналізу коротких текстів – НБК, SVM, CNN та LSTM. Проаналізовано результати їх роботи. Результати свідчать про те, що поєднання CNN та LSTM найкраще показують себе у задачах визначення тональності коротких текстів. Загалом ці моделі можуть бути покращені, треновані на більших масивах даних, а також можуть використовуватись як низько ефективний, проте швидкий метод аналізу та категоризації коротких україномовних текстів, написаних природньою мовою.

Скорочення та абревіатури

NLP – Natural Language Processing, або ж обробка природньої мови – напрям штучного інтелекту та комп’ютерної лінгвістики, що досліджує проблеми та методологію синтезу, аналізу та розпізнавання людської (природної) мови, а також роботу з великими об’ємами лінгвістичних даних.

НБК – Наївний Байєсівський Класифікатор, статистичний алгоритм, що базується на теоремі Байєса та ігнорує зв’язки між різними ознаками. Вирізняється швидкодією і простотою реалізації.

SVM – Support Vector Machine, метод опорних векторів – лінійний алгоритм машинного навчання, часто застосовується в задачах класифікації**.**

RNN – Recurrent neural network, глибинна рекурсивна нейронна мережа, що використовує один і той же шар рекурентно та при кожному кроці враховує свій стан на попередній ітерації.

LSTM – Long Short-Time Memory, розширення архітектури RNN, що покликана усунути проблему вибуху та зникання інградієнту при обробці великих наборів даних.

CNN – Convolutional neural network, глибинна нейронна мережа, що вирізняється використанням шарів згортки та агрегації

Список використаної літератури

1 The uses of content analysis data in studying social change. Harold D. Lasswell, 1968. [Електронний ресурс] – Режим доступу до ресурсу: https://www.jstor.org/stable/41853539?seq=1

2 Data mining with computational intelligence, Wang, Lipo; Fu, Xiuju, 2005. [Електронний ресурс] – Режим доступу до ресурсу: https://cdn.preterhuman.net/texts/science_and_technology/artificial_intelligence/Data%20Mining%20with%20Computational%20Intelligence%20-%20Lipo%20Wang%20,%20Xiuju%20Fu.pdf

3 Encyclopedia of artificial intelligence / Juan Ramon Rabunal Dopico, Julian Dorado de la Calle, and Alejandro Pazos Sierra, 2007

4 G. Salton, A. Wong, and C. S. Yang (1975), A vector space model for automatic indexing «Communications of the ACM», vol. 18, nr. 11, pages 613—620. «(The article in which the vector space model was first presented)»

5 Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman. Indexing by latent semantic analysis. Journal of the American Society for Information Science (1990)

6 Hsinchun Chen Machine learning for information retrieval: Neural networks, symbolic learning, and genetic algorithms. Journal of the American Society for Information Science. Volume 46 Issue 3, ст. 194—216

7 Word2vec, Tomas Mikolov, 2013. [Електронний ресурс] – Режим доступу до ресурсу: https://code.google.com/archive/p/word2vec/

8 Distributed Representations of Words and Phrases and their Compositionality. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. [Електронний ресурс] – 2014. Режим доступу до ресурсу: https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

9 Класичний контент-аналіз та аналіз тексту: термінологічні та методологічні відмінності, Іванов О. В., 2013. [Електронний ресурс] – Режим доступу до ресурсу: http://ekmair.ukma.edu.ua/handle/123456789/7244?show=full

10 Compatibility between Text Mining and Qualitative Research in the Perspectives of Grounded Theory, Content Analysis, and Reliability. Yu, Chong Ho; Jannasch-Pennell, Angel; DiGangi, Samuel, 2011. [Електронний ресурс] – Режим доступу до ресурсу: https://eric.ed.gov/?id=EJ926322

11 The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. Tausczik, Y.R., Pennebaker, J.W.. Journal of Language and Social Psychology 29, 24–54; 2010

12 The Secret Life of Pronouns. What Our Words Say About Us. Nerbonne, J, Literary and Linguistic Computing 29, 139–142, 2014

13 A benchmark comparison of state-of-the-practice sentiment analysis methods. Ribeiro, F.N., Araújo, M., Gonçalves, P., André Gonçalves, M., Benevenuto, F. SentiBench - EPJ Data Science p 5, 23; 2016

14 Detection of hate speech in Arabic tweets using deep learning. Al-Hassan, A., Al-Dossari, H., Multimedia Systems (2021). [Електронний ресурс] – 2016. Режим доступу до ресурсу: https://doi.org/10.1007/s00530-020-00742-w

15 Racism Detection in Twitter Using Deep Learning and Text Mining Techniques for the Arabic Language, A. Alotaibi and M. H. Abul Hasanat, 2020.

16 Three level sentiment classification using SVM with an imbalanced Twitter dataset, Alan Coyne, 2020. [Електронний ресурс] – Режим доступу до ресурсу: https://towardsdatascience.com/a-three-level-sentiment-classification-task-using-svm-with-an-imbalanced-twitter-dataset-ab88dcd1fb13

17 Support Vector Machines(SVM) — An Overview, Rushikesh Pupale, 2018. [Електронний ресурс] – Режим доступу до ресурсу: https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989

18 Content Analysis for the Social Sciences and Humanities. Holsti O.R., 1969. [Електронний ресурс] – Режим доступу до ресурсу: http://www.questia.com/PM.qst?a=o&d=54363997

19 Lang-uk. [Електронний ресурс] – Режим доступу до ресурсу: https://lang.org.ua/uk/

20 LanguageTool API NLP uk. Андрій Рисін. [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/brown-uk/nlp_uk

21 Статистичний розподіл і флуктуації довжин речень в українських, російських і англійських корпусах. О. С. Кушнір, О. С. Брик, В. Є. Дзіковський, Л. Б. Іваніцький, І. М. Катеринчук, Я. П. Кісь. [Електронний ресурс] – 2016. Режим доступу до ресурсу: http://science.lpnu.ua/sites/default/files/journal-paper/2018/jun/12983/22228-239.pdf

22 A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. Zhang Y., Wallace B., 2015. [Електронний ресурс] – Режим доступу до ресурсу: https://arxiv.org/abs/1510.03820

23 A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis, Rehman, A.U., Malik, A.K., Raza, B. et al, 2019.