Суспільна Думка

Застосування технологій нейронних мереж для аналізу суспільної думки

ЗМІСТ

ПЕРЕЛІК ПОЗНАЧОК ТА СКОРОЧЕНЬ

NLP	Natural language processing
ML	Machine learning
AI	Artificial intelligence
CSV	Comma-separated values
NN	Neural network
API	Application programming interface
GPT-3	Generative Pre-trained Transformer 3

ВСТУП

Сучасний світ перебуває в стані постійних змін та вдосконалення, що важливо враховувати для забезпечення нашого соціального прогресу та ефективного взаємодії у глобальному співтоваристві. Однією з найважливіших складових цього прогресу є здатність аналізувати та розуміти суспільну думку, що формується відповідно до подій та явищ, що відбуваються навколо нас. Технології штучного інтелекту, зокрема нейронні мережі, виявляються надзвичайно ефективним інструментом для аналізу та інтерпретації глибоких аспектів громадської думки.

Обрана тема дослідження - "Застосування технологій нейронних мереж для аналізу суспільної думки" - є важливою та актуальною в сучасному світі, оскільки вона спрямована на вдосконалення процесів аналізу та інтерпретації соціальних відгуків, що є ключовим чинником для прийняття обґрунтованих рішень у політичних, економічних, та соціокультурних сферах.

Метою цієї наукової роботи є дослідження та використання технологій нейронних мереж, зокрема алгоритмів та методів обробки природної мови, для підвищення ефективності вивчення та аналізу суспільної думки.

Однією з ключових задач цього дослідження є аналіз повідомлень, опублікованих у соціальних медіа, зокрема на платформі Twitter. Ця платформа надає унікальну можливість отримати доступ до широкого спектру глобальних думок та дискусій, охоплюючи найактуальніші теми сучасності.

Наукова новизна роботи полягає у тому, що на відміну від відомих робіт було детально описано методи аналізу тексту та класифікації повідомлень, зібраних з Twitter, використовуючи передові алгоритми та моделі обробки природної мови. Також була проведена оцінка ефективності застосування моделі глибокого навчання, зокрема Generative Pre-trained Transformer (GPT), у контексті аналізу громадської думки.

Результати цього дослідження можуть мати важливе практичне значення для поліпшення методів аналізу суспільної думки та прийняття обґрунтованих рішень на основі глибокого розуміння глобальних відгуків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Проблематика

Сучасні геополітичні конфлікти, зокрема війна в Україні та її взаємозв'язок, мають найбільший вплив на сприйняття та ставлення громадськості в усьому світі. Він демонструє боротьбу добра зі злом і породжує глибокі соціокультурні та геополітичні зміни, які відображаються в суспільній свідомості та громадській думці.

Можна виділити деякі важливі аспекти війни між Україною та росією, які впливають на громадську думку. По-перше, ця війна - загроза безпеці та територіальній цілісності не тільки України, але й Європи і навіть світу. По-друге, ця війна має вплив на геополітичні відносини обох країн з іншими, особливо нас цікавлять наші відносини з країнами-партнерами, в першу чергу - країнами Європейського Союзу та США. По-третє, вона викликає поділ громадської думки закордонних партнерів, щодо допомоги Україні та ймовірних результатів війни, а також ефективності рішень політичних та дипломатичних відомств.

Спостереження за змінами в громадській думці та у ставленні мешканців різних країн до цієї війни - є ключовим аспектом для подальшого аналізу перспектив допомоги цих країн нашій державі, адже громадська думка може впливати як на офіційні позиції країн-партнерів та і на їх лідерів. Безумовно, що громадська думка відображається у соціальних мережах, які можуть бути використані у якості інформаційної бази цього аналізу.

Також важливо враховувати, що аналіз громадської думки не обмежується лише поглядами на військові або офіційні політичні події. Важливу роль відіграють думки звичайних громадян, які можуть відрізнятися від офіційних позицій уряду своєї країни та відображати різноманіття соціальних, економічних та культурних поглядів.

Аналіз громадської думки є складною задачею, оскільки він передбачає не тільки виявлення різних точок зору, але й розуміння факторів, що впливають на формування цих поглядів. Такі фактори можуть включати історичний та культурний зв'язок, медійну експозицію, геополітичну ситуацію, освітній рівень населення та інші чинники.

Отже, цей розділ дослідження присвячений аналізу зміни відношень громадськості до війни між Україною та росією, а також впливу цієї війни на глобальну громадську думку. Аналізується сприйняття цих подій та їхній вплив на міжнародні стосунки, а також реакція звичайних громадян, що є важливим аспектом вивчення даної теми.

1.2 Існуючі рішення

У сфері аналізу суспільної думки існує кілька підходів та методів, що дозволяють оцінити глибину та розмаїття громадської думки та відношень:

1) Велика кількість інформації щодо громадської думки знаходиться в соціальних мережах, де користувачі активно обговорюють політичні та соціальні події. Аналіз текстів, коментарів, постів у соціальних мережах надає можливість визначити загальну настанову та основні теми обговорень.

2) Спеціалізовані агентства проводять опитування серед населення, щоб виявити думку та ставлення до різних тем та ситуацій у світі. Ці дані надають можливість зрозуміти погляди громадян та їхню установку.

3) Вивчення та аналіз інформації, яка публікується в медійних джерелах, дозволяє визначити тенденції та акценти у висвітленні будь-яких тем, а також виявити можливу спрямованість деяких засобів масової інформації.

4) Аналітичні організації та експерти готують звіти та аналізи щодо громадської думки, де подаються власні оцінки та аналізи.

5) Використання комп'ютерних методів та алгоритмів обробки природної мови. Сучасні методи обробки природної мови на основі застосування моделей та алгоритмів штучного інтелекту дозволяють обробляти великі обсяги даних та проводити аналіз громадської думки за різними параметрами.

Зазначені методи та підходи становлять основу для аналізу суспільної думки. Комбінування цих методів може дати глибше та більш повне розуміння динаміки та особливостей громадської думки.

Проте, у цій задачі, де тематика досить специфічна, є необхідність зміни таких понять, як настрій та відношення до якоїсь із подій. Тому в даній роботі було вирішено аналізувати настрій громадян під іншим кутом, так як для розуміння відношення людей до новин стосовно війни в Україні потрібно дізнатися не класичний настрій або забарвлення повідомлення, а натомість зрозуміти які пости та повідомлення можуть класифікуватися як корисні або не корисні конкретно для України. Наприклад, повідомлення про підтримку, допомогу та тексти про успіхи наших військ будуть класифікуватися як позитивний клас, натомість негативним будуть позначатися пости, що схвалюють заяви російського уряду, які дуже часто виявляються фейками та дезінформацією та тексти у підтримку війни або росії. Ситуацію щодо дезінформації у соціальних мережах та новинах добре демонструє книга «Calling Bullshit: The Art Of Skepticism In A Data-Driven World»[1]

1.3 Постановка задачі

Аналіз предметної галузі показує, що розробка класифікатора новин дуже актуальна, враховуючі продовження війни, зріст фейкової інформації у інформаційному просторі, а також утворення нових тем для обговорень.

Для виконання аналізу, задачу було розбито на кілька пунктів:

1) проаналізувати предметну область, зрозуміти актуальність теми;

2) усвідомити задачі: обробки природної мови; машинного навчання та його виду; необхідного набору даних та його обробки, а також методів його створення та шляхів пошуку та здобування;

3) обрати модель реалізації системи аналізу інформації;

4) підготувати набір даних, проаналізувати, обробити дані для навчання моделі;

5) розробити модель нейронної мережі та потрібні їй процеси навчання та евалюації;

6) оцінити модель на тестовій вибірці;

7) проаналізувати забарвлення текстів.

2 ПІДГОТОВКА ДО РЕАЛІЗАЦІЇ

2.1 Обґрунтування вибору технологічних рішень

Розробка буде вестися за допомогою мови програмування Python.

Усі частини проекту, а саме агрегація, обробка, аналіз даних, розробка моделі, її тренування та тестування, буде проводитися в Jupyter Notebook [2]. Завдяки своїй популярності серед дослідників, науковців та розробників даних, Jupyter Notebook має широку підтримку в багатьох областях, включаючи машинне навчання, обробку даних, візуалізацію, статистику та багато інших. Він інтегрується з багатьма популярними бібліотеками Python, такими як NumPy [3], Pandas [4], Matplotlib [5], TensorFlow [6], PyTorch [7] та інші.

Для агрегації даних буде використоно найпопулярнішу бібліотеку для роботи з даними – Pandas.

Також буде використано бібліотеку scikit-learn [8], ще відому як sklearn. Вона включає в себе алгоритми класифікації, регресії, кластеризації, підбору моделей та інші.

Для обробки тексту дуже гарно підходять бібліотеки RegEx [9].

Для подальшого аналізу даних буде використано бібліотеки NumPy, Matplotlib, Seaborn [10], WordCloud [11].

Для створення моделі нейронної мережі було обрано фреймворк Tensorflow. Tensorflow є потужним фреймворком для розробки нейронних мереж і моделей машинного навчання. Він надає зручний інтерфейс для визначення, тренування та використання нейронних мереж. Tensorflow має широкий набір інструментів для роботи з даними і оптимізації моделей, що дозволяє ефективно і швидко реалізувати штучну нейронну мережу.

Також, у работі використовується пакет gensim[12]. Це бібліотека для обробки тексту та моделювання тематик, яка включає у себе декілька алгоритмів, включаючи Word2Vec, Doc2Vec, LDA (Latent Dirichlet Allocation) та інші. В роботі буде використовуватися Word2Vec, який є одним з ключових алгоритмів у цій бібліотеці для векторизації слів та отримання їхнього семантичного представлення.

Також у роботі проводиться експеримент з використанням моделі gpt-3.5-turbo для анотації та класифікації текстів, що є дуже актуальним питанням у сфері NLP останніх років[13-20]. OpenAI надає доступ до своїх інтелектуальних послуг через API під назвою openai[21], включаючи GPT-3. GPT-3 є однією з останніх версій нейромережевих моделей, яка може генерувати текст та виконувати різні завдання мовного процесу. Для роботи з цим API, було використано книги «OpenAI GPT For Python Developers: The art and science of developing intelligent apps with OpenAI GPT-3, DALL·E 2, CLIP, and Whisper»[22] і «Natural Language Processing with Transformers. Building Language Applications with Hugging Face»[23]

Загалом, вибір цих технологічних рішень обґрунтовується їхньою популярністю, широким функціоналом, зручним інтерфейсом, наявністю багатофункціональних бібліотек та підтримкою спільноти розробників.

2.2 Підбір даних

Для проведення аналізу суспільної думки щодо війни в Україні було важливо зібрати наявні дані, що відображають громадське ставлення до даної проблеми. Проте, завдання зі збору даних виявилося надзвичайно складним, оскільки платформа Twitter, яка є важливим джерелом для аналізу громадської думки, почала обмежувати доступ до даних стосовно війни в Україні.

У останній період часу спостерігалося блокування облікових записів, які здійснювали збір даних щодо конфлікту, а також приховування коментарів та постів на цю тему. Додатково, доступ до API та можливість власного збору даних були суттєво обмежені платформою Twitter. Це значно ускладнило завдання зібрання необхідних для навчання моделі та дослідження даних.

У таких умовах було здійснено пошук та аналіз наявних відкритих датасетів. Були знайдені два набори даних, які відповідали різним вимогам дослідження.

Перший датасет "Unveiling Global Narratives: A Multilingual Twitter Dataset of News Media on the Russo-Ukrainian Conflict"(надалі буде згадуватися, як zenodo) [24] містив 1.5 мільйони твітів для 60 мов. Його

було обрано для моделювання тому, що в ньому присутні стовбці, створені за допомогою моделі RoBERTa[25] яка відображала сутність тексту, та показувала наскільки цей текст схильний бути «в підтримку» або «проти» України або росії, та війни в цілому. Кожен запис у цьому датасеті представлений у форматі JSONL.

Друге джерело було отримане з платформи Kaggle, "Ukraine Conflict Twitter Dataset"(надалі буде згадуватися, як tweets)[26], і містить 44 мільйони твітів до червня 2023 року, що стосувалися війни між Україною та росією. Цей набор даних було вирішено використовувати задля аналізу даних та класифікації даних з останніх двох місяців за допомогою GPT-3.

Обрані датасети відображають масштаб та різноманітність громадської думки та нададуть можливість глибше дослідити відношення суспільства до даної проблеми.

3 РЕАЛІЗАЦІЯ

3.1 Структура проекту

Проект складається з трьох файлів програмного коду. Призначення кожного з файлів детально наведено в таблиці 3.1.

Таблиця 3.1

Файл	Призначення файлу
PartnerSentiment.ipynb	Файл з підготовкою даних до роботи та аналізом даних.
ClassifySentiment.ipynb	Файл, в якому відбувається обробка текстових даних, та створення процесу використання openai API
ModelingSentiment.ipynb	Файл, в якому відбувається обробка текстових даних, процес векторизації тексту та моделювання.

3.2 Підготовка даних

Реалізація проекту розпочалася зі створення файлу PartnerSentiment.ipnb, основні частини якого описані у підрозділах А.1, А.2, А.3 додатка А.

В першу чергу, були імпортовані усі необхідні модулі, про які згадувалося у розділі 2.1.

Далі було реалізовано функціонал котрий ітеративно трансформує дані zenodo з формату jsonl до датафрейму, результати зображені на рисунку 3.1.

[1]:

	tweet_id	text	lang	country	sentiment	This statement is in favour of Russia	This statement is against Russia	This statement is against Ukraine	This statement is in favour of Ukraine	This statement is in favour of war	This statement is against war	This statement is in favour of military conflict	This statement is against military conflict
0	1540451336676986373	Weekend selection : Zelensky was not prepared ...	cs	Slovakia	neutral	0.0132	0.6298	0.4643	0.8667	0.0220	0.2177	0.0511	0.1404
1	1516696238212661253	The Ukrainian war , Charles Michel of Kiev , t...	ro	Italy	neutral	0.0011	0.0841	0.3233	0.2243	0.2862	0.0274	0.4485	0.0199
2	1587766197298548738	The invention of the Shark drone is a new chap...	te	India	neutral	0.6999	0.6533	0.8056	0.0088	0.1121	0.0003	0.1737	0.0004
3	1586193152175529984	Will nuclear war be over ?	te	India	negative	0.0106	0.2205	0.0108	0.0022	0.0008	0.0016	0.0007	0.0019
4	1580452730212331520	Boys fight : Students fight in coaching center...	te	India	negative	0.0032	0.9286	0.5969	0.0020	0.0029	0.6562	0.0119	0.4063
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1524827	1517911176956366848	# GerrariEZ in Pamplona , a bicycle march agai...	eu	Spain	negative	0.0004	0.9680	0.7573	0.0006	0.0002	0.9900	0.0002	0.9960
1524828	1501840878129799173	What 's happening in Ukraine by Elena Belokl a...	eu	Spain	neutral	0.0048	0.0185	0.0057	0.0046	0.0041	0.0020	0.0042	0.0035
1524829	1501787814266032129	RT : War in Europe , contradictions of imperia...	eu	Spain	negative	0.0412	0.4120	0.3494	0.0335	0.1720	0.0462	0.0971	0.3945
1524830	1500846404507512836	RT : PRESENCE i	eu	Spain	negative	0.0160	0.2094	0.2122	0.0238	0.1754	0.0228	0.1447	0.4320
1524831	1494329083370430479	RT : The Ukrainian army is bombing Donbass .	eu	Spain	negative	0.3360	0.2382	0.2786	0.6910	0.1880	0.1536	0.0851	0.1952

1524832 rows × 13 columns

Рисунок 3.1 – Вигляд датафрейму zenodo

Далі було оброблено дані tweets. Спочатку вони були у вигляді папки з архівованими файлами типу csv, тож для їх конкатенації за допомогою pandas було створено датафрейм, а також розроблено функціонал для конкатенації певних колонок, фільтрації тільки англійських текстів та витягування хештегів з тексту за допомогою regex-паттерну. Результат підготовки tweets, наведено на рисунку 3.2.

	tweetid	location	tweetcreatedts	followers	text	language	hashtags	username
0	1509681950042198030	Hawaii	2022-04-01 00:00:00.000000	392	⚡The Ukrainian Air Force would like to address...	en	#ProtectU #StopRussia #UkraineUnderAtta	Yaniela
1	1509681950151348229	NaN	2022-04-01 00:00:00.000000	881	Chernihiv oblast. Ukrainians welcome their lib...	en	#russianinvasion #StandWithUkraine #UkraineUnd...	gregffff
2	1509681950683926556	NaN	2022-04-01 00:00:00.000000	72	America us is preparing for something worse th...	en	#RussianUkrainianWar #China #Taiwan	ThanapornThon17
3	1509681951116046336	International Web Zone	2022-04-01 00:00:00.000000	377	JUST IN: #Anonymous has hacked & released ...	en	#Anonymous #OpRussia #DDoSecrets	I_Protest_2021
4	1509681951304990720	Hunter Account	2022-04-01 00:00:00.000000	25	***PUBLIC MINT NOW LIVE***\n\nFor \n@billionai...	en	#nft #mint	Marsh_Win_01
...	...	...	...	...	...	...	...	...
44416748	1509677720514433028	North Logan, UT	2022-03-31 23:43:11.000000	4101	Amb. William Taylor, former U.S. Ambassador to...	en	#Ukraine #TheReidOut	snarky_op
44416749	1509677722590605313	NaN	2022-03-31 23:43:12.000000	150	@lapatina_ @berehulak this just hurts my heart...	en	#Ukraine	AnnMRobie
44416750	1509677723136020480	NaN	2022-03-31 23:43:12.000000	348	@Abiy Ahmed, Eritrean dictator Isaias Afewerki...	en	#Tigray #Ukraine #TigrayGenocide #supportHR	tsion_tigray
44416751	1509677724490641410	The America's	2022-03-31 23:43:12.000000	845	#MARIUPOL—However, even at #Ilyich, #Ukraineli...	en	#MARIUPOL #Ilyich #Ukraine #Azov #Azov #DNR	TLBSociety
44416752	1509677727288373255	NaN	2022-03-31 23:43:13.000000	452	Cocktail Bar Punks\n\n10,000 #NFT poker faces ...	en	#NFT #Polygon #Ukraine	iqnitedit

44416753 rows × 8 columns

Рисунок 3.2 – Вигляд датафрейму tweets

3.3 Початковий аналіз даних

Після того як дані були підготовлені, було розпочато аналіз. Було створено нову колонку зі скороченою датою до формату «рік-місяць», а далі згруповано за цим критерієм, задля того щоб подивитися скільки унікальних текстових повідомлень було у кожному місяці з початку війни і до Червня 2023 року. Розподіл зображено на рисунку 3.3.

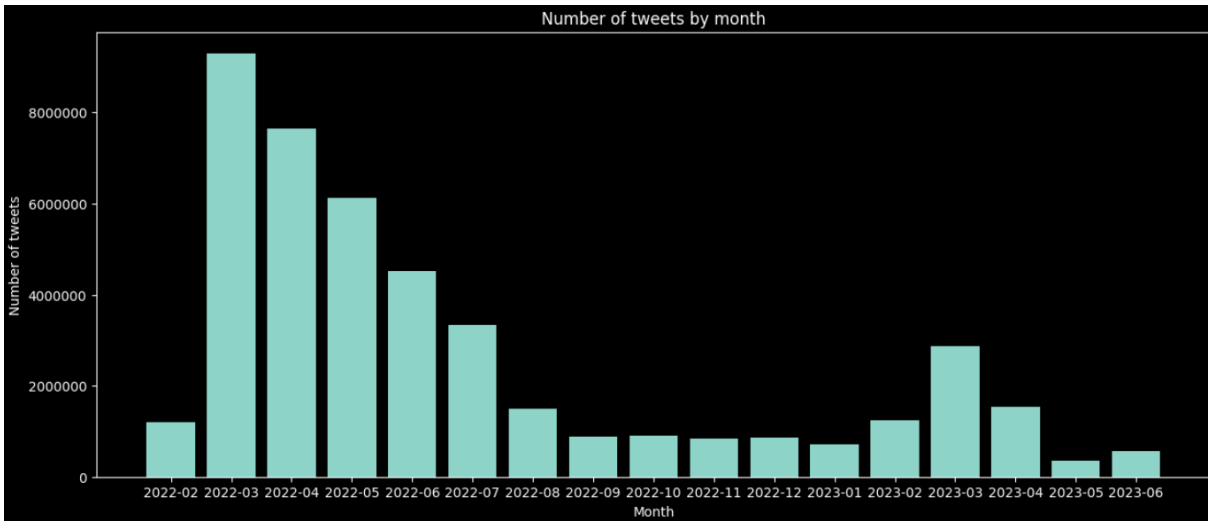

Рисунок 3.3 – Кількість твітів у місяць

Завдяки візуалізації видно, що пік обговорення війни був на самому початку, навіть у лютому дані якого ведуться з 24 числа, за тиждень перевершили результати осені 2022 та Травень і Червень 2023 року. Також, спостерігається підйом з Лютого по Квітень 2023, це напевно пов’язано з річницею війни та чутками про весняний контрнаступ.

Наступним кроком було проаналізувати кількість твітів від кожної країни в місяць. У Твітері є можливість виставляти свою локацію як назву країни, або міста. Тому за допомогою regex усі локації були зведені до форми країни, а ті записи де локації немає не були враховані в цей етап аналізу, результати якого приведено на рисунку 3.4.

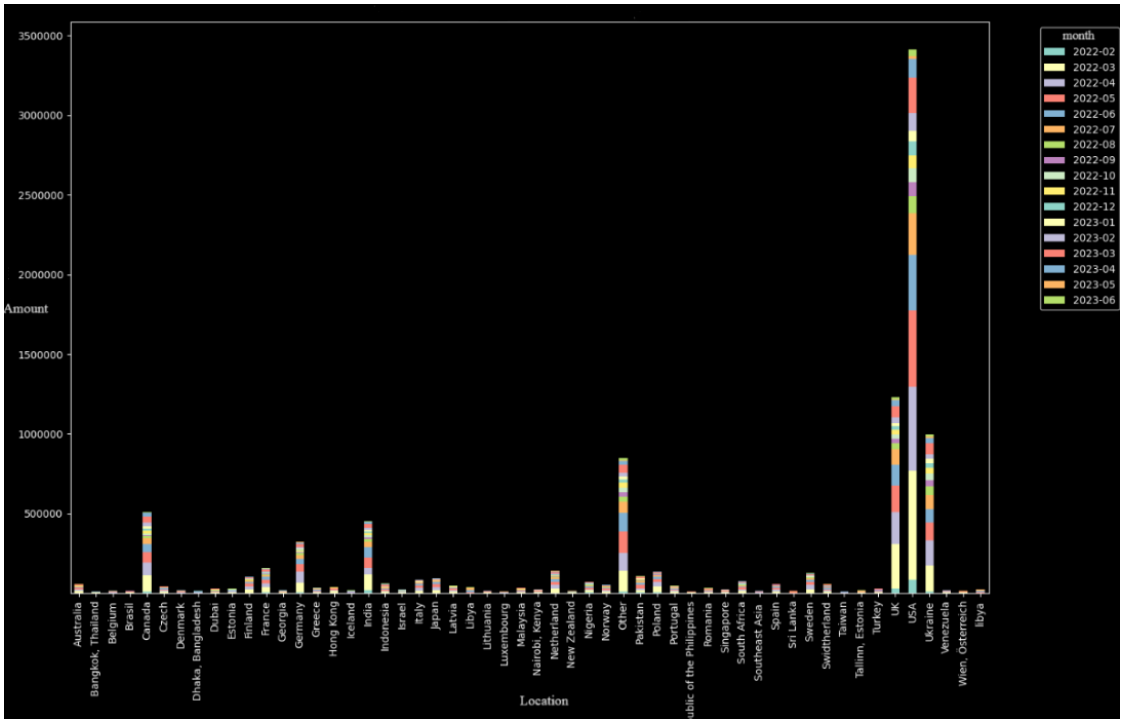

Рисунок 3.4 – Кількість твітів від кожної країни в місяць

На рисунку видно, що активніше за всіх війну обговорюють жителі Сполучених Штатів Америки і перевершують, наступних по кількості жителів Великої Британії більше ніж у 2 рази. Також у лідерах з'являються Україна, Канада, Індія, Германія, Франція, Польща та скандинавські країни.

Далі було проаналізовано хештеги, котрі користувачі додавали до своїх повідомлень. Завдяки бібліотеці WordCloud було створено графіки розподілу слів для кожного місяця. Демонстрацію найцікавіших візуалізацій можна побачити на рисунках 3.5-3.8.

Рисунок 3.5 – Популярні хештеги в перші місяці війни

Рисунок 3.6 – Популярні хештеги в перші місяці війни

Рисунок 3.7 – Поява сезонних хештегів

Рисунок 3.8 – Найактуальніші хештеги

Проаналізувавши графіки видно, що в перші місяці війни країна отримала значну підтримку і війна в Україні стала найактуальнішою темою для спілкування в найбільших платформах світу. Також, варто

зазначити що всі військові злочини також були помічені та винесені в соціум. На рисунку 3.7 видно, що інколи на ряду з темами про Україну з’являються і інші події, такі як конфлікт між Китаєм і Тайванем, сезонні явища, наприклад чемпіонат світу з футболу і новорічні свята. На Червень 2023 року світ продовжує обговорювати війну, нові військові злочини російських терористів та нові перемоги та досягнення України.

Більш повну картину історій змін найпопулярніших тем для обговорення можна вивести створивши таймлайн хештегів. Його суть проста, було взято топ 10 тем з кожного місяця та продемонстровано як вони змінюються. Даний таймлайн можна побачити на рисунку 3.9.

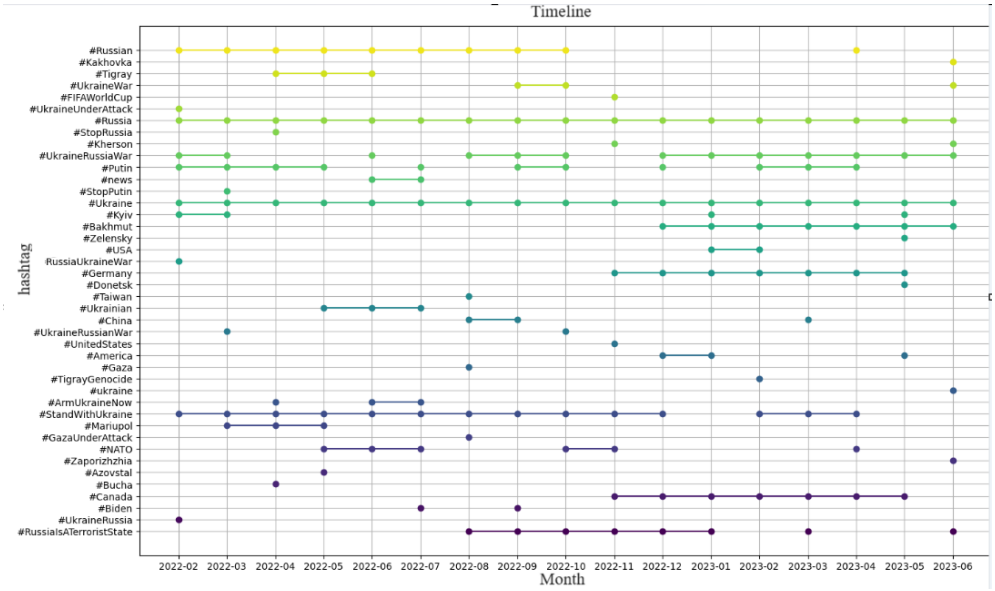

Рисунок 3.9 – Таймлайн

Також було проведено ще один аналіз хештегів, а саме аналіз хештегів користувачів у яких більше 500 тисяч підписників. Це було зроблено для того, щоб побачити які ідеї піднімають популярні люди і чи вони на користь Україні. Результати цього аналізу зображені на рисунку 3.10.

Рисунок 3.10 – Хештеги популярних користувачів

3.4 Обробка даних та моделювання

Наступним етапом стала підготовка даних zenodo до навчання. Всі дії було виконано у файлі ModelingSentiment.ipynb, зміст якого наведено у додатку Б. І тут стало питання з фінальною змінною для кожного рядка, так як в даних немає значення яке б задовольняло завдання. Але слід пам’ятати, що замість цільової змінної існують колонки завдяки яким можна створити цільове значення.

Було вирішено зробити це шляхом складання правил і після багатьох спроб знайти найкращі значення результат виглядав наступним чином:

сума колонок «This statement is against russia» і «This statement in favour of Ukraine» має бути більша ніж сума протилежних значень помножена на 2.5. Такі записи ідентифікувалися, як позитивний клас.

сума колонок «This statement is against russia» і «This statement in favour of Ukraine» помножена на 1.7 має бути менше протилежних значень. Записи задовольняючі цю умову були позначені як негативний клас.

Відповідно, записи котрі не задовольнили жодну з умов, позначені міткою нейтрального класу.

Проставивши значення цільової змінної, було перевірено розподіл емоційного забарвлення, який зображений на рисунку 3.11.

```
[8]: df.mySentiment.value_counts()

[8]: mySentiment
     Neutral     1194233
     Positive     197985
     Negative     132614
     Name: count, dtype: int64
```

Рисунок 3.11 – Розподіл міток класу

Згенерувавши таргет, потрібно було переходити до роботи із текстом. Тому, було створено функціонал для того щоб відчистити текст від небажаних символів, посилань та перевести його у нижній реєстр. Функція для обробки тексту зображена на рисунку 3.12.

```
def preprocess(text):
    text = str(text).lower()
    text = re.sub(r"< user_mention_1 >|< url_1 >",'', text)
    text = re.sub(r"\n|\r|\d", '', text)
    text = re.sub('https:[/a-z0-9.]+|&[a-z;]+|@[a-z]+','',text)
    return text
```

Рисунок 3.12 – Функція для обробки тексту

Існує багато підходів для обробки тексту і кожен з них дуже тісно пов’язаний з задачею, подальшим переведенням даних до числового формату і самою природою даних. В даній роботі векторизація тексту буде відбуватися за допомогою моделі word2vec, яка буде тренуватися на повному корпусі після обробки, тому є можливість не вирізати з текстів так звані стопворди, які не несуть будь-який сенс, бо фінальний вигляд векторів буде створюватися за допомогою взяття середнього з усього тексту. Але, перед переведенням даних до числової форми варто подивитися розподіл довжини текстів. Для цього було створено процес підрахунку кількості токенів в повідомленні. Сам процес та графік, що відображає розподіл довжин текстів зображений на малюнку 3.13.

```
[13]: df['len_text'] = df['clean_text'].apply(lambda x: len(x.split()))
      df.len_text.value_counts().sort_index().plot()

[13]: <Axes: xlabel='len_text'>
```

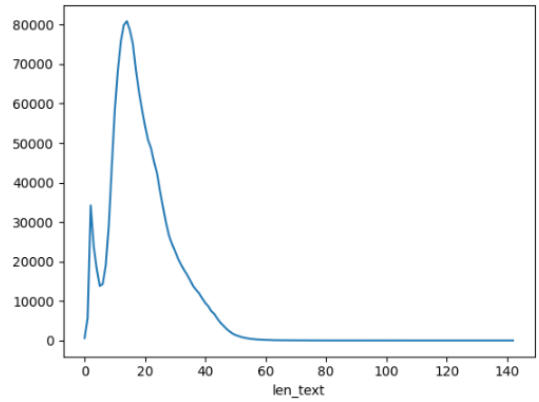

Рисунок 3.13 – Розподіл довжин текстів

Було виявлено що майже 900 записів мають довжину меншу за 5 і не несуть за собою багато сенсу та будь-яку думку, тому їх було виключено з набору даних.

Наступним кроком було створення word2vec моделі, шляхом розбиття усіх значень обробленого тексту на токени і надання їх до моделі для тренування. У результаті модель має словниковий запас у кількості 44351 унікальний токен та кожен вектор має 100 значень. Результат тренування зображений на рисунку 3.14.

```
[21]: model = Word2Vec(sentences=sentences.values,
                      sg=1,
                      workers=4)

      model.wv.vector_size, len(model.wv.index_to_key)

[21]: (100, 44351)
```

Рисунок 3.14 – Створена модель word2vec

Далі, датасет було поділено на тренувальну та валідаційну вибірки. Частина валідаційних даних становила 20% від усього корпусу. Після чого було створено функціонал для переведення тексту до векторної форми і після її успішного застосування, отримано тренувальний та валідаційний набори векторів. Також у функції передбачена поява слів котрі не містяться у словниковому запасі моделі, в таких випадках

замість цих слів до суми векторів додається сто-вимірний вектор заповнений значеннями 0.5. Зміст функції приведений на рисунку 3.15.

```
[23]:
import numpy as np

def text_to_vector(text):
    words = text.split()
    vectors = []
    for word in words:
        try:
            vector = model.wv.get_vector(word)
            vectors.append(vector)
        except KeyError:
            vectors.append([0.5 for i in range(100)])
    if not vectors:
        return None
    return np.mean(vectors, axis=0)
```

Рисунок 3.15 – Функція зведення текстів до векторів

На основі отриманих списків векторів, було створено два датафрейми для тренування та перевірки, після чого було реалізовано процес тренування Гауссівського наївного байєсівського класифікатора.

Далі відбулася перевірка моделі на валідаційних даних, за допомогою матриці плутанини, результати наведені на рисунку 3.16.

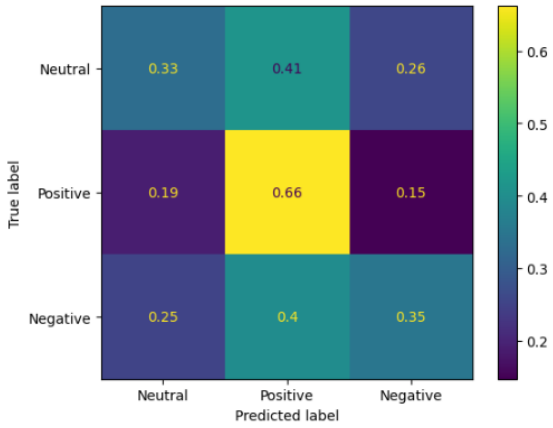

Рисунок 3.16 – Матриця плутанини наївного байєсівського класифікатора

Результати показали, що загальна точність моделі дорівнює 44%, а матриця плутанини дає зрозуміти що модель нормально працює з позитивним класом, але нейтральний та негативний дуже часто сприймаються як позитивний. Тому, було побудовано досить просту нейронну мережу, яка складалася з 4 шарів нейронів та процес тренування і оцінки її показників, що показано на рисунку 3.17.

```
label_mapping = {'Positive': 1, 'Neutral': 0, 'Negative': 2}
y_train_numeric = y_train.map(label_mapping)
y_test_numeric = y_test.map(label_mapping)

print("fd dtype:", fd.dtypes)
print("y_train_numeric dtype:", y_train_numeric.dtype)

fd = fd.astype(np.float32)
fd_t = fd_t.astype(np.float32)

classifier = tf.keras.Sequential([
    tf.keras.layers.Dense(128, activation='relu', input_shape=(fd.shape[1],)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(3, activation='softmax')
])

classifier.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

classifier.fit(fd, y_train_numeric, validation_data=(fd_t, y_test_numeric), epochs=100, batch_size=32)
```

Рисунок 3.17 – Побудова нейронної мережі та її тренування

Наступним кроком була перевірка моделі на валідаційних даних за допомогою матриці плутанини. Результати вийшли набагато кращі, але все ж таки не ідеальними, їх зображено на малюнку 3.18.

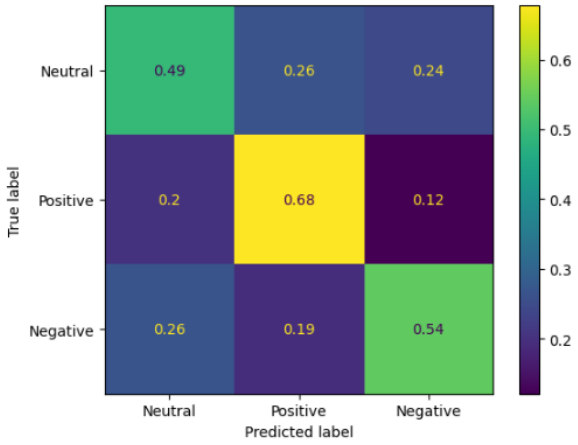

Рисунок 3.18 – Матриця плутанини нейронної мережі

Як тепер видно, результати більш точними, а також позитивний та негативний класи тепер зливаються між собою набагато менше, що має дуже позитивний влив на роботу моделі. Також, у даній роботі, було проведено експеримент мета якого була спробувати провести анотацію даних за допомогою сучасних великих мовних моделей. Результати цього дослідження, знаходяться у розділі 3.5.

3.5 Використання великих мовних моделей для анотації

Надалі робота відбувалася у файлі ClassifySentiment.ipynb, зміст якого знаходиться у додатку В.

Для експерименту було обрано модель gpt-3.5-turbo, розроблену компанією openAI. Її використання можливе за допомогою API, воно також не безкоштовне тому для проведення експерименту та подальшого аналізу настроїв було значно скорочено датасет tweets з метою економії коштів та швидкістю опрацювання, були вжиті наступні міри:

були узяті дані лише за Травень та Червень місяці 2023 року;

за для економії також текст був попередньо оброблений із використанням функції з рисунку 3.12;

було викинуто дублікати, бо у датасеті присутні однакові тексти від різних користувачів;

також, були оброблені значення локації по тій самій методиці, про яку йшлося у розділі 3.3, але додатково було виключену твіти з України, так як головна задача проаналізувати настрій жителів країн партнерів.

Далі було створено словник під назвою mySentiment та процес роботи з openAI API реалізація якого зображена на рисунку 3.19.

```
openai.api_key = "YOUR_API_KEY"
import time
def get_completion(prompt, model="gpt-3.5-turbo"):
    messages = [{"role":"system", "content":"""You must classify input on 3 classes: Positive, Neutral, Negative.
                Positive - Text aimed for favour of Ukraine, end of the war, support of Ukraine, or somehow belittles russia, that is negativ
                Negative - Is opposite to Positive class and good for russia.
                If Ukraine win fight, kill or bombing russian forces, soldiers, vehicle or airtransport with weapon, dron or technic it`s Pos
                {"role": "user", "content": prompt}]
    response = openai.ChatCompletion.create(
            model=model,
            messages=messages,
            timeout=7,
            request_timeout=7,
            max_tokens=1)
    return response.choices[0].message["content"]

def useGPT():
    try:
        for message in tqdm(tmp.clean_text.values):
                prompt = f"""{message}"""
                mySentiment[message] = get_completion(prompt)
    except KeyboardInterrupt:
        return 'Done'

    except Exception:
        useGPT()
```

Рисунок 3.19 – Робота з openAI API

На рисунку видно, що для роботи з API , треба обов’язково створити свій ключ на сайті openAI, та далі використовувати його безпосередньо у середовищі програмування.

Далі було створено функціонал для посилання запитів до моделі gpt-3.5-turbo, а також процес оновлення словника де у циклі ітеративно відправляється текст з раніше створеного набору даних. Важливо зазначити що разом з текстом у запиті треба описати моделі її задачу і пояснити необхідні правила. Також в даному випадку дозволяється встановити параметр моделі max_tokens на значення 1, бо нам потрібен у відповідь лише клас повідомлення і для цього достатньо одного токену. Як результат після виконання коду, було отримано словник з 59470 записами анотованими за допомогою великої мовної моделі. Їх було збережено та порівняно з результатами нейронної мережі з попереднього розділу. Результати порівнянь наведені на рисунках 3.20 і 3.21 та проведені у файлі «ModelingSentiment.ipynb» з додатку Б.

На рисунку значення «prediction» означає прогноз GPT, відповідно «prediction_model» - прогноз нейронної мережі створеної під час виконання роботи.

```
prediction  prediction_model
Positive    Positive           0.150613
Neutral     Neutral            0.143738
Positive    Negative           0.137373
Neutral     Positive           0.118910
Positive    Neutral            0.113166
Neutral     Negative           0.102590
Negative    Negative           0.096684
            Neutral            0.093027
            Positive           0.043899
Name: proportion, dtype: float64
```

```
for i in test_data[(test_data['prediction']=='Positive')&(test_data['prediction_model']=='Positive')]['clean_text'].sample(10):
    print(i, end='\n')
```

```
PBS NewsHour full episode, Jan. 18, 2023 #UkraineRussianWar #UkraineWillWin #Luhansk #Donetsk #Bakhmut
5/ UKR Eastern Forces Spokesperson Colonel Serhiy Cherevaty stated that UKR forces advanced up to 1,700m in the past day,  the UKR 3rd Separate Assault Brigade said the brigade’s counterattacks expanded the UKR salient in the #Bakhmut area to 2,000m meters wide by 700m deep.
Ukrainian troops of the 7th Separate Battalion celebrate the liberation of Neskuchne.Video probably from yesterday or the day before.Source: #Ukraine #Counteroffensive #Donetsk
 Stay Strong. Stay Vigilant. #StayUnited #SnakeIslandSalute to rf. Aussie F/A18s coming out of retirement for UA. #ArmedForces of #Ukraine are incredible  inspiring #ShieldOfEurope #Freedom Warriors. My prayers are for you. Trust God, trust your instincts. lv #ThePrussianPrincess
Ukrainian refugee sings with Lithuanians in support for Ukraine #Canada #Germany #Bakhmut #Kyiv #Ukraine
Last week, we welcomed  star and acclaimed Ukrainian footballer, Oleksandr Zinchenko, to the #UNITED24 team.  Together with 7, they will play Game4Ukraine, a charity match at Stamford Bridge, to raise funds for a school restoration:
❗Good Morning Everyone!  is unfortunately hiding my Kherson posts from you, some with explosions going off behind me won't even appear on your feed. Please can you help us combat this "shadow/algorithm ban" 1. RT every post you see 📌2.Follow FLK ✅3.🔔ON #Ukraine
#BREAKING US is providing up to $325 million in more military aid for #Ukraine, a defense official tells VOA, in a package expected to be announced tomorrow.PDA 40 to include:*#Strykers  #Bradleys that can replace those damaged  destroyed*Munitions for NASAMS  #HIMARS
Go get them boys 🔥🐻🔥! #славаУкраїні #RussiaInvadedUkraine #RussiaIsATerroristState #NAFOExpansionIsNonNegotiable #NAFO #NAFOfellas #NAFOCatsDivision #UkraineRussiaWar
Russian forces retreat from Soledar, where they suffered serious losses in heavy battles #UkraineRussianWar #UkraineWillWin #Luhansk #Donetsk #Bakhmut
```

```
for i in test_data[(test_data['prediction']=='Positive')&(test_data['prediction_model']=='Negative')]['clean_text'].sample(10):
    print(i)
```

```
Want to support #Ukraine UA, fight RU disinformation, bonk V, and have fun doing so? Join #NAFO and become a #Fella! It's easy — visit  for details!
ukraine army using capture Russian battle tank and advancing toward kharkiv region. #russia #ukraine#Canada #Germany #America #Ukraine #Bakhmut
It's actually ridiculous how pro-russian trolls go on about  #Ukraine being a Nazi state. Many families in Ukraine suffered horrifying death tolls fighting the Nazis in WWII. The male-female ratio in Ukraine after WWII was 1:10 - that gives you some idea of the amount of people…
🔥#Breaking, 🎈 In Ankara TR, during the events of the Parliamentary Assembly of the Black Sea Economic Community, the representative of Russia  tore the flag of Ukraine  from the hands of a UA Member of Parliament.!🔗#Ukraine  #Bakhmut #Russia
Ambush Footage!! Ukraine Paratroopers Destroyed 870 Russian Wagner Group which crossed Bakhmut track#Canada #Germany #America #Ukraine #Bakhmut
Aid group provides drinking water after Kakhovska dam blast:    #Ukraine
Evidence indicates Russia blew #NovaKakhovka dam. Norwegian Seismic specialists, recorded a signal indicating explosion on RU controlled dam, coinciding with dam burst. #Ukraine security service released phone recording - Russian soldier admits RU blew dam
WATCH Ukraine releases footage claiming to show fighting in Soledar #shorts #UkraineRussianWar #UkraineWillWin #Luhansk #Donetsk #Bakhmut
The Armed Forces of #Ukraine made a breakthrough on the #Zaporozhye front by 7 km: the Russians are retreating, abandoning villages#ukrainecounteroffensive
Dramatic footage!! Ukrainian troops strike Russian positions in north Bakhmut until run away #UkraineRussianWar #UkraineWillWin #Luhansk #Donetsk #Bakhmut
```

Рисунок 3.20 – Порівняння результатів різних моделей

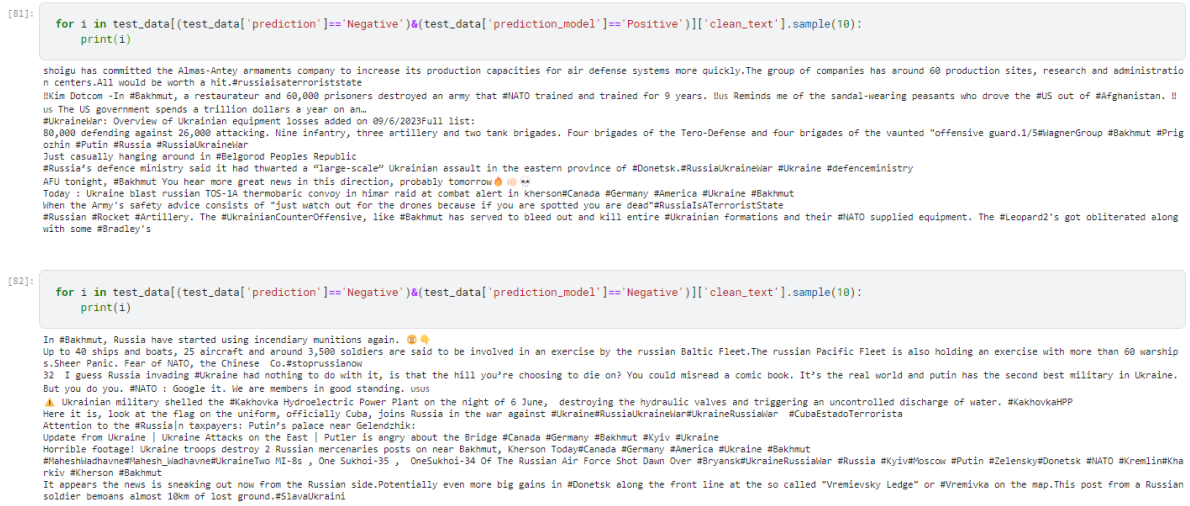

Рисунок 3.21 – Порівняння результатів різних моделей

Як видно на рисунку 3.20 результати моделей співпадають лише приблизно у 39% тестових даних, ще у 42% відбувається змішення позитивного, або негативного класу з нейтральним, а в інших випадках моделі дають абсолютно різні відповіді.

Якщо проаналізувати випадки в яких прогнози різняться, дуже гарно проглядається те що результати GPT набагато кращі за результати розробленої нейронної мережі, що дає можливість сказати що експеримент вдався і можна вважати анотацію даних за допомогою великих мовних моделей ефективним підходом, який не тільки економить час, який витратила б людина для анотації настільки великої кількості даних, але також може бути і більш економним рішенням для деяких проектів, наприклад не потрібно витрачати багато часу і ресурсів для тренування власного класифікатора, бо іноді достатньо лише описати завдання і правила для моделі.

3.6 Аналіз даних із урахуванням прогнозів

Фінальним етапом роботи був аналіз цільової змінної. Треба пам'ятати що класифікація за допомогою великих мовних моделей, відбувалася лише на даних з останніх двох наявних місяців, тому аналіз проводився не на всій вибірці, а лише на найактуальнішій частині.

По-перше, було продемонстровано зміну розподілу прогнозів за Травень та Червень місяці 2023 року. Результати зображені на рисунку 3.22.

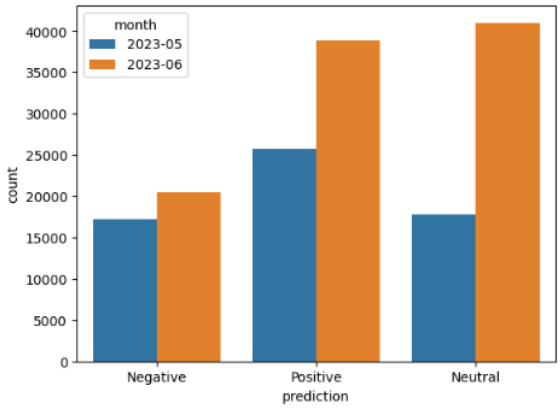

Рисунок 3.22 – Розподіл цільової змінної

Згідно з діаграмою, у Червні місяці кількість позитивних та нейтральних повідомлень помітно зросла, в той же час негативні мали лише невеликий кількісний зріст, що є гарною ситуацією для України.

Також, було проаналізовано середню оцінку емоційного забарвлення окремо для кожної країни. Результати для найкращих 10 країн знаходяться на рисунку 3.23, для найгірших 10 на рисунку 3.24.

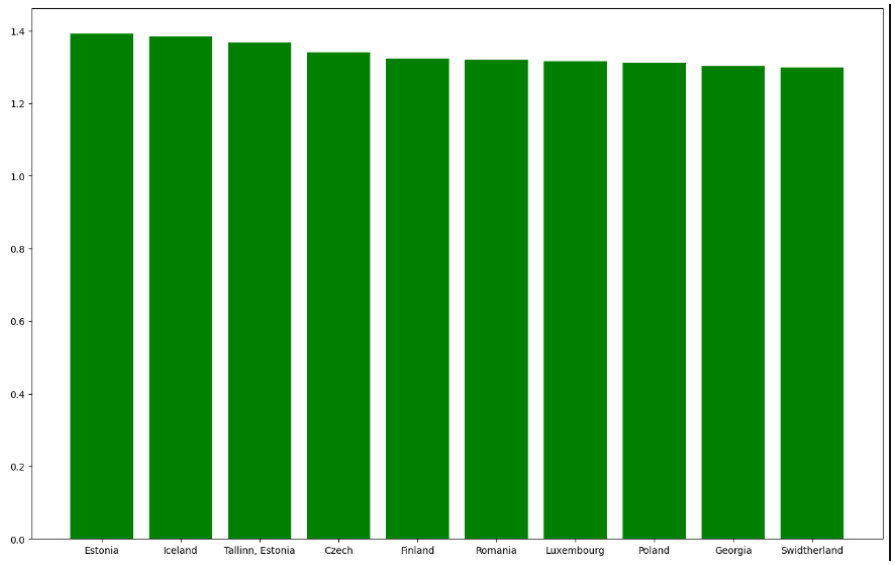

Рисунок 3.23 – Країни з найбільш позитивним настроєм

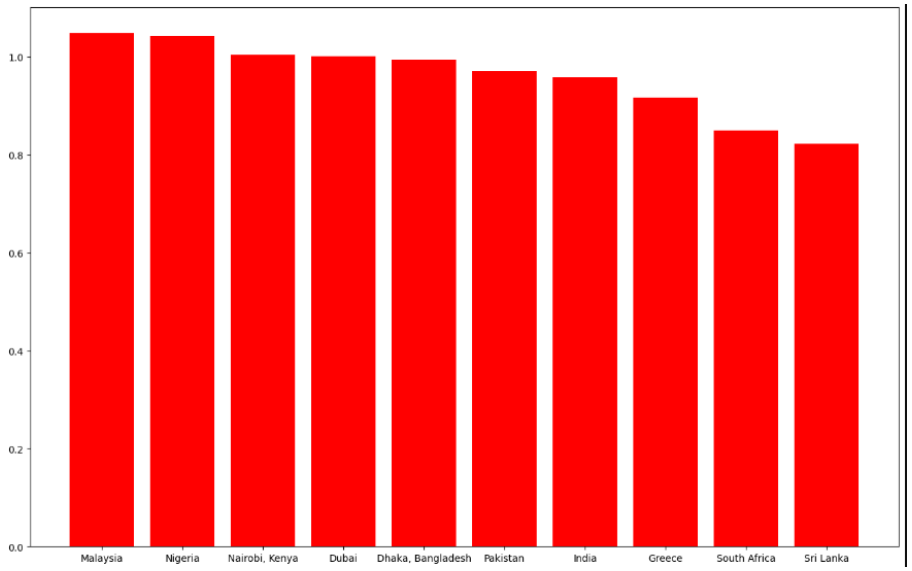

Рисунок 3.23 – Країни з найменш позитивним настроєм

Як видно з аналізу, перші міста займають європейські країни, деякі з них безпосередні країни-сусіди України. В той час як, країнами з найменьшим середнім настроєм до України стали країни Азії та Африки, а також з’явилася Греція. Тоді було вирішено поділити країни за їх континентом та перевірити, як різні континенти відносяться до України і війни. Результати знаходяться на рисунку 3.24.

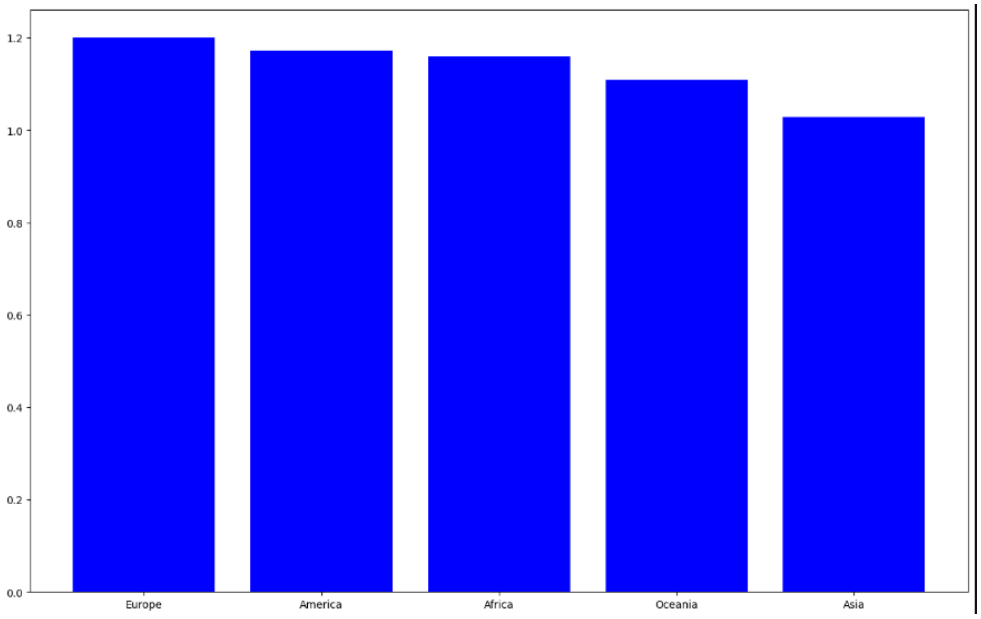

Рисунок 3.24 – Ставлення континентів до України

В цілому, усі континенти мають показники вище за одиницю, яка означає нейтралітет, але у будь-якому разі Азія має достатньо великий відрив від Європи та Америки.

Також, було проведено аналіз хештегів окремо для кожного класу цільової змінної. Це допоможе бачити, які теми для обговорень визивають у суспільства настрій за чи проти України. Результати зображені на рисунку 3.25. Також, є можливість дивитися до таких тегів для кожної окремою країни та бачити, за що жителі цієї країни занепокоєні та чому вони залишають негатив у своїх повідомленнях. Аналіз негативних тегів зображено на рисунку 3.26.

Рисунок 3.25 – WordCloud по міткам класу

Рисунок 3.26 – WordCloud для негативних тегів в США

ВИСНОВКИ

В науковій роботі були проведені дослідження, що зосереджені на застосуванні нейронних мереж для аналізу суспільної думки, зокрема в контексті аналізу даних, зібраних з соціальної мережі Twitter. Загалом, робота включала розробку технічних рішень з використанням сучасних нейромережевих технологій, спрямованих на об'єктивне вивчення та аналіз глибинних аспектів суспільних реакцій на війну в Україні.

Під час проведення досліджень були застосовані методи та алгоритми NLP, AI, ML та було реалізовано проєкт, який вимагав розробки та застосування багатьох моделей нейронних мереж.

Підсумки проведеної роботи показують, що була досягнута мета наукової роботи - дослідження та використання технологій нейронних мереж, зокрема алгоритмів та методів обробки природної мови, для підвищення ефективності вивчення та аналізу суспільної думки.

Всі поставлені дослідницькі задачі та задачі розробки були розв'язані.

В першу чергу, було здійснено важливий крок у зборі та підготовці великого обсягу даних, який становив вражаючу кількість - 44 мільйони твітів з Twitter. Цей обсяг даних був ключовим для надання об'єктивної картини суспільної думки.

Далі було використано дві основні моделі для аналізу цих даних: Гауссівський Наївний Байєсівський Класифікатор та нейронну мережу. Проведений нами порівняльний аналіз дозволив з'ясувати переваги та особливості кожної моделі в контексті аналізу суспільної думки.

Окремо варто відзначити експеримент з використанням великої мовної моделі GPT-3.5-turbo для класифікації тексту. Цей експеримент привів до успішних результатів і відкрив нові можливості у використанні масштабних мовних моделей для аналізу суспільних реакцій.

В рамках даного дослідження також було створено візуалізацію результатів аналізу суспільної думки та настроїв щодо війни в Україні.

Це було зроблено з метою надання найглибших інсайтів у суспільну реакцію та сприяння більшому розумінню глобальних соціокультурних процесів.

В цілому, дане дослідження відкриває двері до подальших досліджень та допомагає розширити наше розуміння суспільних динамік та реакцій. Отримані результати та методи можуть бути використані для поліпшення методів аналізу громадської думки та сприяння більшому впливу на соціокультурні та політичні процеси в сучасному світі.

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1 Calling Bullshit: The Art Of Skepticism In A Data-Driven World / C. T. Bergstrom, J.D. West. – Random House, 2020 – 336 p.

2 Jupyter Notebook : головна сторінка сайту бібліотеки JupyterNotebook // https://jupyter.org/, 07.10.2023.

3 NumPy : головна сторінка сайту бібліотеки NumPy // https://numpy.org/, 07.10.2023.

4 Pandas : головна сторінка сайту бібліотеки Pandas // https://pandas.pydata.org/, 07.10.2023.

5 Matplotlib : головна сторінка сайту бібліотеки Matplotlib // https://matplotlib.org/stable/index.html, 07.10.2023.

6 TensorFlow : головна сторінка сайту фреймворку TensorFlow // https://www.tensorflow.org/, 07.10.2023.

7 PyTorch : головна сторінка сайту фреймворку PyTorch // https://pytorch.org/, 07.10.2023.

8 Scikit-Learn : головна сторінка сайту бібліотеки Scikit-Learn // https://scikit-learn.org/stable/index.html, 07. 10.2023.

9 RegEx : головна сторінка сайту бібліотеки RegEx // https://regexr.com/, 07.10.2023.

10 Seaborn : головна сторінка сайту бібліотеки Seaborn // https://seaborn.pydata.org/, 07.10.2023.

11 WordCloud : документація бібліотеки WordCloud // https://pypi.org/project/wordcloud/, 07.10.2023.

12 Gensim : документація бібліотеки Gensim // https://pypi.org/project/gensim/, 10.10.2023.

13 Штучний інтелект у природній мові: мовні моделі, малоресурсні мови та дегуманізація в текстах : посилання на статтю // https://dou.ua/forums/topic/45583/, 12.10.2023.

14 Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. / Ahmed H, Traore I, Saad S. 2017.

15 Sentiment Analysis on Social Media Data Using Intelligent Techniques. / Panguila K., Chandra J. 2019.

16 Integration of Sentiment Analysis of Social Media in the Strategic Planning Process to Generate the Balanced Scorecard. / Grande-Ramirez J., Roldan-Rayes E. 2022.

17 Sentiment analysis on social media tweets using dimensionality reduction and natural language processing. / Okeyo G., Kimwele M. 2022.

18 Coding Adventures: Sentiment Analysis, NLP, and Neural Networks Explored. / The Journey. 2023.

19 Unlocking AI Challenges: From Sentiment Analysis to Text Summarization, Your Comprehensive Guide to AI Exploration and Application. / The Journey. 2023.

20 Sentiment Analysis on TripAdvisor Hotel Reviews with ChatGPT. / Niggl D. 2023.

21 openAI : головна сторінка платформи openAI // https://platform.openai.com/overview, 12.10.2023.

22 OpenAI GPT For Python Developers: The art and science of developing intelligent apps with OpenAI GPT-3, DALL·E 2, CLIP, and Whisper/ A. E. Amri – Leanpub, 2023 – 236 p.

23 Natural Language Processing with Transformers. Building Language Applications with Hugging Face/ L. Tunstall, L. von Werra, T. Wolf – O`Riley, 2022 – 406 p.

24 zenodo : посилання на статтю та дані //https://arxiv.org/pdf/2306.12886.pdf, 07.10.2023.

25 RoBERTa : модель від стенфордського університету // https://huggingface.co/docs/transformers/model_doc/roberta, 07.10.2023.

26 tweets : датасет для аналізу // https://www.kaggle.com/datasets/bwandowando/ukraine-russian-crisis-twitter-dataset-1-2-m-rows/data, 07.10.2023.

ДОДАТОК А

Початковий файл PartnerSentiment

А.1 Підготовка zenodo

```
file_path = 'dataset_zenodo.jsonl'
data = []
with open(file_path, 'r') as file:
   for line in tqdm(file):
      data.append(json.loads(line.strip()))
df = pd.DataFrame(index=[i for i in range(len(data))], columns=['tweet_id','text','lang','country','sentiment','This statement
is in favour of Russia','This statement is against Russia',\
                  'This statement is against Ukraine','This statement is in favour of Ukraine',\
                  'This statement is in favour of war','This statement is against war',\
                  'This statement is in favour of military conflict','This statement is against military conflict'])
for ind in tqdm(range(len(data))):
   try:
      df.iloc[ind] = [data[ind]['tweet_id'], " ".join([i['text'] for i in data[ind]['stanza_output'][0]]),
data[ind]['lang'],data[ind]['country'],list(data[ind]['sentiment'].keys())[0],\
         data[ind]['stance'][0]['entail_prob'],\
         data[ind]['stance'][1]['entail_prob'],\
         data[ind]['stance'][2]['entail_prob'],\
         data[ind]['stance'][3]['entail_prob'],\
         data[ind]['stance'][4]['entail_prob'],\
         data[ind]['stance'][5]['entail_prob'],\
         data[ind]['stance'][6]['entail_prob'],\
         data[ind]['stance'][7]['entail_prob']]
   except:
      pass
```

А.2 Підготовка tweets

```
data_folder = r'C:\Users\hp\Documents\Codes\myproj\Paper\data'
df1 = pd.DataFrame(columns=['username', 'acctdesc', 'location', 'following', 'followers',
     'totaltweets', 'usercreatedts', 'tweetid', 'tweetcreatedts','retweetcount', 'text', 'hashtags', 'language','favorite_count'])
for filename in tqdm(os.listdir(data_folder)):
   if filename.endswith('.csv.gzip'):
      d = pd.read_csv(data_folder+'\\'+filename, compression='gzip', header=0,usecols=['username', 'acctdesc', 'location',
'following', 'followers',
     'totaltweets', 'usercreatedts', 'tweetid', 'tweetcreatedts','retweetcount', 'text', 'hashtags', 'language','favorite_count'])
      d = d[d['language']=='en']
      df1 = pd.concat([df1,d])
```

А.3 Аналіз даних

```
df['month'] = df['tweetcreatedts'].apply(lambda d: str(d)[:7])
res = df[['month']].groupby('month').apply(lambda g: pd.Series([len(g)]))
fig, ax = plt.subplots(figsize=(15, 6))
plt.style.use('dark_background')
plt.bar(res.index, res[0].values)
ax.ticklabel_format(axis='y', style='plain')
ax.set_xlabel('Month')
ax.set_ylabel('Number of tweets')
ax.set_title('Number of tweets by month')

result = tmp.groupby(['location', 'month']).size().unstack()
```

```
ax = result.plot(kind='bar', stacked=True, figsize=(16, 10))

ax.ticklabel_format(axis='y', style='plain')
ax.set_title('Кількість твітів від різнних каїн за місяць месяц')
ax.set_xlabel('Країна')
ax.set_ylabel('Кількість твитов')
ax.legend(title='Місяць', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()

res = df[['month','hashtags']].groupby('month').apply(lambda g: pd.Series([' 
'.join(g.hashtags.values)]))
for date in tqdm(res.index):
   wordCloud = WordCloud(width = 1000, height = 1000,
            background_color ='white',
            max_words=100,
            collocations=False,
            min_font_size = 10).generate(res.loc[date][0])
   plt.figure(figsize=(10,10))
   plt.imshow(wordCloud)
hash = []
popular_hash = {}
for i in res.index:
   popular_hash.update({i:Counter(res.loc[i][0].split(' ')).most_common(10)})
   hash += [j[0] for j in Counter(res.loc[i][0].split(' ')).most_common(10)]

from collections import defaultdict
hashtags_data = defaultdict(list)
all_hashtags = set(hashtag for hashtags_count in data.values() for hashtag, _ in hashtags_count)
N=0

for month, hashtags_count in data.items():
   hashtags_count_dict = dict(hashtags_count)
   N=0
   for hashtag in all_hashtags:
      N+=1
      if hashtags_count_dict.get(hashtag):
         count = N
      else:
         count=0
      hashtags_data[hashtag].append(count)
for hashtag in hashtags_data:
   hashtags_data[hashtag] += [0] * (17 - len(hashtags_data[hashtag]))
for hashtag, counts in hashtags_data.items():
   print(f"{hashtag}: {counts}")

import matplotlib.pyplot as plt

data = hashtags_data

months = list(res.index)
modified_data = {hashtag: [count if count != 0 else None for count in counts] for hashtag, counts in
data.items()}

plt.figure(figsize=(16, 10))

for idx, (hashtag, counts) in enumerate(modified_data.items()):
   plt.plot(months, counts, label=hashtag, marker='o', color=plt.cm.viridis(idx / len(modified_data)))

plt.xlabel('Місяць')
```

```
plt.ylabel('Хештег')
plt.title('Таймлайн хештегів')
plt.yticks(np.arange(1,len(modified_data)+1), list(modified_data.keys()))
plt.grid(True)
plt.show()

res = df[['username','location','followers','hashtags']]\
      .groupby('username').apply(lambda g: pd.Series([g.location.unique()[0], g.followers.max(), '
'.join(g.hashtags)]))

wordCloud = WordCloud(width = 1000, height = 1000,
            background_color ='white',
            max_words=100,
                collocations=False,
            min_font_size = 10).generate(' '.join(res[2]))
plt.figure(figsize=(10,10))
plt.imshow(wordCloud)
```

ДОДАТОК Б

Початковий файл ModelingSentiment

Б.1 Обробка даних

```
                import pandas as pd
df = pd.read_csv('/kaggle/input/zenodo-source/zenodo_filtered.csv')
rule1= (df['This statement is against Russia']+df['This statement is in favour of Ukraine'])>2.5*(df['This statement is against
Ukraine']+df['This statement is in favour of Russia'])
rule2 = (df['This statement is against Russia']+df['This statement is in favour of Ukraine'])*1.7<(df['This statement is against
Ukraine']+df['This statement is in favour of Russia'])
mask1 = df[rule1]
mask2 = df[rule2]
df['mySentiment'] = 'Neutral'
df.loc[mask1.index, 'mySentiment'] = 'Positive'
df.loc[mask2.index, 'mySentiment'] = 'Negative'
def preprocess(text):
   text = str(text).lower()
   text = re.sub(r"< user_mention_1 >|< url_1 >",'', text)
   text = re.sub(r"\n|\r|\d", '', text)
   text = re.sub('https:[/a-z0-9.]+|&[a-z;]+|@[a-z]+','',text)
   return text
df['clean_text'] = df['text'].apply(preprocess)
df['len_text'] = df['clean_text'].apply(lambda x: len(x.split()))
df.len_text.value_counts().sort_index().plot()
df = df[df['len_text']>=5]
zenodo['clean_text'] = zenodo['text'].apply(preprocess)
sentences = zenodo['clean_text'].apply(str.split)
sentences.values
model = Word2Vec(sentences=sentences.values,
             sg=1,
             workers=4)

model.wv.vector_size, len(model.wv.index_to_key)
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df['clean_text'], df['mySentiment'], test_size=0.2, random_state=1)
print(y_train.value_counts(), y_test.value_counts())
import numpy as np

def text_to_vector(text):
   words = text.split()
   vectors = []
   for word in words:
      try:
         vector = model.wv.get_vector(word)
         vectors.append(vector)
      except KeyError:
         vectors.append([0.5 for i in range(100)])
   if not vectors:
      return None
   return np.mean(vectors, axis=0)
X_train_vectors = [text_to_vector(text) for text in X_train]
X_test_vectors = [text_to_vector(text) for text in X_test]
from tqdm.notebook import tqdm
fd = pd.DataFrame(index = [i for i in range(len(X_train_vectors))],columns=[i for i in range(100)])
for i in tqdm(range(len(X_train_vectors))):
   fd.iloc[i] = X_train_vectors[i]
fd_t = pd.DataFrame(index = [i for i in range(len(X_test_vectors))],columns=[i for i in range(100)])
for i in tqdm(range(len(X_test_vectors))):
   fd_t.iloc[i] = X_test_vectors[i]
```

Б.2 Моделюювання

```
from sklearn.naive_bayes import GaussianNB
```

```
classifier = GaussianNB()
classifier.fit(fd, y_train.replace({'Positive':1, 'Neutral':0, 'Negative':2}))
y_pred = classifier.predict(fd_t)

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

accuracy = accuracy_score(y_test.replace({'Positive':1, 'Neutral':0, 'Negative':2}), y_pred)
print(f"Accuracy: {accuracy}")

print("Classification Report:")
print(classification_report(y_test.replace({'Positive':1, 'Neutral':0, 'Negative':2}), y_pred))

cm = confusion_matrix(y_test.replace({'Positive':1, 'Neutral':0, 'Negative':2}), y_pred, normalize='true')
cmd = ConfusionMatrixDisplay(cm, display_labels=['Neutral','Positive','Negative'])
cmd.plot()

label_mapping = {'Positive': 1, 'Neutral': 0, 'Negative': 2}
y_train_numeric = y_train.map(label_mapping)
y_test_numeric = y_test.map(label_mapping)

print("fd dtype:", fd.dtypes)
print("y_train_numeric dtype:", y_train_numeric.dtype)

fd = fd.astype(np.float32)
fd_t = fd_t.astype(np.float32)

classifier = tf.keras.Sequential([
   tf.keras.layers.Dense(128, activation='relu', input_shape=(fd.shape[1],)),
   tf.keras.layers.Dense(64, activation='relu'),
   tf.keras.layers.Dense(32, activation='relu'),
   tf.keras.layers.Dense(3, activation='softmax')
])

classifier.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

classifier.fit(fd, y_train_numeric, validation_data=(fd_t, y_test_numeric), epochs=100, batch_size=32)
preds = classifier.predict(fd_t)
predictions = []
for i in preds:
   predictions.append(np.argmax(i))
cm = confusion_matrix(y_test.replace({'Positive':1, 'Neutral':0, 'Negative':2}), predictions, normalize='true')
cmd = ConfusionMatrixDisplay(cm,  display_labels=['Neutral','Positive','Negative'])
cmd.plot()
def prep_pipe(data):
   data = data.apply(preprocess)
   vectors = [text_to_vector(text) for text in data]
   from tqdm.notebook import tqdm
   fd = pd.DataFrame(index = [i for i in range(len(vectors))],columns=[i for i in range(100)])
   for i in tqdm(range(len(vectors))):
      fd.iloc[i] = vectors[i]
   return fd.astype(np.float32)
test_data = pd.read_csv('/kaggle/input/fianlcross/sentimentDF.csv')
preds = classifier.predict(test_input)
   predictions = []

   for i in preds:

      predictions.append(np.argmax(i))

   test_data['prediction_model']= predictions
```

Б.3 Порівняння моделей

```
test_data[['prediction_gpt','prediction_ffnn']].value_counts(normalize=True)
for i in test_data[(test_data['prediction']=='Positive')&(test_data['prediction_model']=='Positive')]['clean_text'].sample(10):
   print(i, end='\n')
for i in
test_data[(test_data['prediction']=='Positive')&(test_data['prediction_model']=='Negative')]['clean_t
ext'].sample(10):
   print(i)
```

```
for i in
test_data[(test_data['prediction']=='Negative')&(test_data['prediction_model']=='Positive')]['clean_t
ext'].sample(10):
    print(i)
for i in
test_data[(test_data['prediction']=='Negative')&(test_data['prediction_model']=='Negative')]['clean_
text'].sample(10):
    print(i)
```

ДОДАТОК С

Початковий файл ClassifySentiment

С.1 Обробка даних

```
#res['sum'] = sum_list
res = res[res[0]>11000]

location_list = list(res.sort_values(0, ascending=False).index)
from tqdm.notebook import tqdm
tmp = df[df['location'].isin(location_list)][['location','text','month']]
for li in tqdm(lists.items()):
   for val in tqdm(li[1]):
      tmp['location'].replace({val:li[0]}, inplace=True)
interest_zone = list(lists.keys()) + ['Brasil',
 'Denmark',
 'Dhaka, Bangladesh','Estonia',
 'Georgia',
 'Hong Kong',
 'Iceland',
 'Indonesia',
 'Israel',
 'Libya',
 'Lithuania',
 'Luxembourg',
 'Nairobi, Kenya',
 'New Zealand','Republic of the Philippines','Singapore','Sri Lanka',
 'Taiwan',
 'Tallinn, Estonia',
 'Venezuela']
interest_zone.remove('Ukraine')
interest_zone.remove('Other')
import regex as re
tmp['clean_text'] = tmp['text'].apply(lambda x: re.sub('\n|https:[/a-zA-Z0-9.]+|&[a-zA-Z;]+|@[a-zA-Z]+','',x))
tmp = tmp[tmp.month.isin(['2023-05','2023-06'])]
```

С.2 Робота з API

```
mySentiment={}
openai.api_key = "YOUR_API_KEY"
import time
def get_completion(prompt, model="gpt-3.5-turbo"):
   messages = [{"role":"system", "content":"""You must classify input on 3 classes: Positive, Neutral,
Negative.
            Positive - Text aimed for favour of Ukraine, end of the war, support of Ukraine, or somehow
belittles russia, that is negative in its direction. Success of Ukraine good news for Ukraine and
Ukrainians aid for Ukraine. #UkraineWillWin, #StandWithUkraine #StopPutin,#RussiaIsANaziState
and #StopWar topics, and so on.
            Negative - Is opposite to Positive class and good for russia.
            If Ukraine win fight, kill or bombing russian forces, soldiers, vehicle or airtransport with
weapon, dron or technic it`s Positive"""},
            {"role": "user", "content": prompt}]
   response = openai.ChatCompletion.create(
         model=model,
         messages=messages,
         timeout=7,
         request_timeout=7,
```

```
        max_tokens=1)
    return response.choices[0].message["content"]

def useGPT():
    try:
        for message in tqdm(tmp.clean_text.values):
            prompt = f"""{message}"""
            mySentiment[message] = get_completion(prompt)
    except KeyboardInterrupt:
        return 'Done'

    except Exception:
        useGPT()
```