УДК 004.032.26 **doi: 10.26906/SUNZ.2024.1.158**

С. І. Шаповалова, А. Ю. Софієнко

Національний технічний університет України «КПІ імені Ігоря Сікорського», Київ, Україна

ЦИФРОВІ ПРЕДСТАВЛЕННЯ TELEGRAM-КАНАЛІВ

Анотація. Предметом дослідження цієї статті є цифрові представлення текстових інформаційних ресурсів на прикладі Telegram-каналів. Мета роботи – визначити оптимальний для подальшої тематичної класифікації метод формування цифрових представлень Telegram-каналів. У статті вирішуються наступні завдання: означення підходів до формування вхідного вектору; визначення етапів обробки текстових даних для цифрового представлення Telegram-каналу; створення датасету цифрових представлення Telegram-каналів; розмітка датасету для розв'язання задачі класифікації; визначення гіперпараметрів оптимальних моделей класифікації. Отримано такі результати: створений датасет цифрових представлень Telegram-каналів, сформованих на основі мережі SBERT, за трьома підходами: агрегація векторів публікацій, конкатенація ключових слів за методом TF-IDF та поєднання перших двох підходів; визначено, що підхід конкатенації ключових слів за методом TF-IDF та поєднання перших двох підходів до формування цифрових представлень Telegram-каналів на основі текстових публікацій є найбільш ефективним для подальшої класифікації за тематикою; визначено оптимальні за точністю гіперпараметри моделей тематичної класифікації: Logistic Regressio та нейромережі глибокого навчання. Перспективним напрямком подальших досліджень є оцінювання застосування запропонованих цифрових представлень до задач кластеризації та пошуку.

Ключові слова: обробка текстів природною мовою, BERT, тематична класифікація повідомлень, representation learning.

Вступ

Онлайн-соціальні мережі (OSN) становлять життєво важливий аспект сучасної комунікації, який є актуальним у повсякденному житті. Онлайн-платформи та соціальні мережі стали ключовим джерелом інформації для більшої частини населення світу. Невпинне зростання кількості джерел та обсягів інформації призводить до нових проблем. Перенасиченість великою кількістю інформації зробила пошук, фільтрацію, пошук релевантних джерел інформації, їх структуризацію та аналіз складним та ресурсовитратним завданням. Оскільки публікації в соціальних мережах – це сильно розріджені текстові дані з великою кількістю шумів, для подальшої роботи з ними необхідний інструмент якісного вилучення "дистильованих" фактів.

Сучасним підходом до аналізу даних є застосування алгоритмів машинного навчання, які приймають на вхід цифрові величини – вектори ознак, якість та чистота яких напряму впливає на результат роботи алгоритмів. Згідно з дослідженням компанії Gradus Research Company [1], яке проводилось на замовлення Національної суспільної телерадіокомпанії України, понад 55% українців отримують інформацію з месенджерів та соціальних мереж. Telegram – лідер серед месенджерів – 89% співвітчизників користуються ним для перегляду новин.

Аналіз останніх досліджень і публікацій

Соціальні мережі стали не тільки потужними інструментами для спілкування та обміну інформацією, а й об'єктом інтенсивних досліджень в галузі обробки природної мови на основі машинного навчання. Лише за останні декілька років було проведено велику кількість досліджень, пов'язаних з використанням інформації з онлайн соціальних мереж.

Довгий час для створення цифрових представлень інформації природною мовою використовувалися класичні алгоритми Bag of Words та TF-IDF, що вираховували лише частоту використання слів чи словосполучень (англ. N-gram) в документах, при цьому втрачаючи семантичну інформацію. Проте в останнє десятиліття спостерігається стрімкий розвиток більш складних та контекстно-орієнтованих методів обробки природної мови. На відміну від традиційних статистичних методів, до розуміння семантики та контексту використовуються нейронні мережі, що створюють представлення (англ. embedding) слів та текстів з врахуванням контекстної інформації.

Починаючи з 2013 року, для створення векторів слів було представлено такі підходи:

- Word2Vec (Google, 2013) [2],
- GloVe (Stanford University, 2014) [3],
- FastText (Facebook AI, 2016) [4].

Ці алгоритми використовують неглибокі мережі для побудови представлень для слів. З розвитком глибоких нейронних мереж в 2018 році дослідниками з Google AI було представлено архітектуру BERT [5], що здатна оперувати реченнями та фрагментами тексту, а не окремими словами. Через рік розроблено SBERT (Sentence-BERT) [6] – модифікацію BERT, що використовує для навчання сіамську та триплет мережу, представлення якої можна порівнювати за косинусної відстанню, що є важливим в задачах пошуку та кластеризації.

Дослідники використовують різні архітектури та моделі, такі як глибокі нейронні мережі та методи графового аналізу, для обробки даних з таких платформ як Twitter, Facebook, Instagram та інші. В роботі [7] було застосовано алгоритм Spherical k-means для кластеризації distilBERT-представлень з метою визначення основних тем дезінформації, виявлення ключових трендів. Було отримано високу точність в оцінках на наявному датасеті. Подібним чином в роботі [8] були використані ембедінги, для класифікації акаунтів ботів та людей в мережі Twitter.

З моменту впровадження архітектури з основою BERT у сфері обробки природної мови, цей підхід

отримав значний успіху завдяки своїй здатності до розуміння контексту та синтаксичних залежностей. Однак, незважаючи на великий потенціал BERT, її вхід базується на токенах та має фіксоване обмеження на розмір, що може суттєво впливати на використання мережі в конкретних сценаріях. Архітектура BERT обмежена за кількості вхідних токенів, які можуть бути оброблені моделлю. У класичній версії моделі ця обмеженість складає 512 токенів. В розширених версіях, таких як BERT-large, вхід розширено до 1024 токенів, проте цього недостатньо для опрацювання об'ємних текстових даних.

Таким чином, для оптимального використання BERT-моделей та подолання їх обмежень потрібне вдосконалення методів створення цифрових представлень. Постає необхідність у розробці методів агрегації або скорочення тексту, щоб зберегти важливу інформацію та забезпечити ефективне використання BERT на великих корпусах тексту.

Однак, важливо врахувати, що агрегація тексту може призвести до втрати семантичної інформації, тому необхідно вдосконалення методів ембедінгу для збереження значущості текстового контексту під час агрегації.

Отже, для побудови цифрових представлень Telegram-каналів необхідне подолання обмежень розміру входу нейронної мережі SBERT.

Постановка задачі

Метою статті є визначення оптимального для подальшої тематичної класифікації методу формування цифрових представлень Telegram-каналів.

Завдання:

– означити підходи до формування цифрових представлень Telegram-каналів на основі мережі SBERT;

– визначити етапи обробки текстових даних для цифрового представлення Telegram-каналу;

– створити датасет цифрових представлення Telegram-каналів;

– розмітити датасет для розв’язання задачі класифікації;

– експериментально визначити гіперпараметри оптимальних моделей класифікації.

Вхідною інформацією є необроблені текстові дані публікацій в Telegram-каналах українською та російською мовами.

Результатом є цифрове представлення, яке придатне для подальшої обробки та вирішення завдань із застосуванням машинного навчання для класифікації, фільтрації, агрегації, категоризації, рекомендації.

Модель штучного інтелекту - навчена нейронна мережа SBERT, ресурс HuggingFace [9].

Подолання обмеженості розміру входу моделі SBERT для формування цифрових представлень Telegram-каналів здійснюється підбором оптимальних параметрів за критерієм точності тематичної класифікації відповідних методів за такими підходами:

– агрегація векторів публікацій;

– вектор конкатенації TF-IDF ключових слів;

– агрегація векторів публікацій та ключових слів.

Результати досліджень

1. Підходи до формування цифрового представлення Telegram-каналів. Модель SBERT має ліміт входу 512 або 1024 токенів. Для подолання цього обмеження може бути використано декілька підходів.

1. *Агрегація векторів публікацій*: входом моделі є публікація. Замість подання моделі для обробки всього корпусу текстів Telegram-каналу використовуються окремі публікації. Кожна з них окремо подається до SBERT. Таким чином для всіх текстів отримуються вектори прихованих станів. Обчислюється середнє значення цих векторів, що дозволяє узагальнити та зберегти суттєву інформацію з різних повідомлень в одному компактному представленні.

2. *Вектор конкатенації TF-IDF ключових слів*: входом моделі є текст -конкатенація ключових слів. Такий метод передбачає виділення важливих термінів у тексті за допомогою TF-IDF та об'єднання цих термінів у вектор. Цей метод дозволяє враховувати суттєві та репрезентативні елементи тексту, винятково акцентуючи увагу до ключових словосполучень ресурсу.

3. *Агрегація векторів публікацій та ключових слів*: входом моделі є публікації та конкатенація ключових слів. Цей підхід враховує ключові терміни та контекст окремих публікацій, що можуть спільно вносити вагомий внесок у розуміння тексту моделлю SBERT.

2. Конвеєр обробки текстових даних Telegram-каналу. Створення числового представлення Телеграм-каналу здійснюється за декілька етапів і потребує вирішення як загальних так і специфічних задач обробки тексту. На рис. 1 представлено схему конвеєра обробки тексту (англ. data pipeline) для визначення характеристичних ознак тексту.

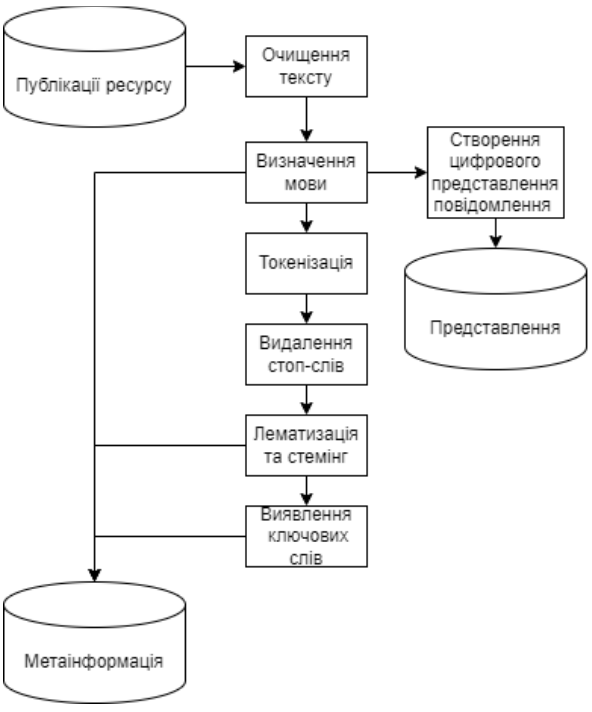

Рис. 1. Конвеєр обробки тексту

Очищення вхідного тексту. Першим етапом обробки публікації з ресурсу є очищення від шуму: тегів, спецсимволів, веб-посилань. Цей етап дозволяє подальшим алгоритмам та моделям краще концентруватися на суттєвому змісті.

Фільтрація на основі визначення мови публікації. Задачею цього етапу є виокремлення повідомлень російською та українською мовами. Кожна мова має специфічні особливості: граматику, лексику та правила, які впливають на обробку тексту. Визначення мови допомагає вибрати відповідні методи та алгоритми для обробки тексту конкретною мовою, що забезпечує більш точні та ефективні результати. На цьому етапі для ідентифікації мови було використано бібліотеку fastText від Facebook's AI Research, що здатна визначити ISO код мови тексту з-поміж 176 доступних.

Очищений від шуму текст та його мова зберігаються як метаінформація публікації. Оскільки основними мовами набору даних дослідження є українська та російська, публікації іншими мовами далі в обробку не йшли.

Токенізація та видалення стоп-слів. Наступним етапом конвеєру обробки є розбиття слів на базові елементи – токени. В роботі для токенізації було використано Python бібліотеку NLTK[10]. З упорядкованої колекції токенів відфільтровуються стоп-слова — токени, що не несуть суттєвої інформації. Список стоп-слів обирався залежно від мови тексту. За основу було взято наявні списки з бібліотеки NLTK. Оскільки на цьому ресурсі не представлено українську та недостатньо представлено російську, колекція стоп-слів була доповнена прикладами з ресурсів [11, 12] . Також додано слова з власних спостережень.

Лематизація, стемінг та морфологічний аналіз. Оскільки може існувати велика кількість форм одного й того ж слова, для полегшення аналізу тексту слова приводяться до базової форми. В результаті декілька слів в різній формі трактуються як єдине представлення. Для отримання початкової форми слів було використано бібліотеку Python pymorphy2 [13]. Однією з функцій цієї бібліотеки є здатність визначати морфологічну характеристику слова, яку було використано для виявлення власних назв з метою доповнення ключовими словами метаінформації про публікацію.

Визначення ключових слів ресурсу. Для швидкої оцінки тематики ресурсу необхідно виокремити ключові слова, які відрізняються від загальних та мають важливу інформаційну цінність. Для визначення ключових термінів в контексті всього ресурсу було використано статистичний метод TF-IDF (Term Frequency-Inverse Document Frequency) з бібліотеки scikit-learn [14]. На цьому етапі метаінформація також доповнюється ключовими словами. Для цього визначається топ-100 слів з найвищим значення TF-IDF метрики, яка відображає важливість слова в поточній публікації.

3. Створення датасету. Наявність якісного датасету є одним з ключових факторів отримання хорошого результату машинного навчання.

На сьогоднішній день існує декілька публічно доступних наборів даних з Telegram:

1) багатомовний Pushshift Telegram [15];

2) перською мовою Dataset-for-teenagers-chat-in-Telegram-groups [16];

3) болгарською мовою TRACES Bulgarian Telegram Dataset [17];

4) на тематику криптовалют Crypto telegram groups [18].

Проте існуючі дані непридатні для поставленої задачі через тематичну спрямованість та мови представлення. Тому було створено датасет з текстових даних та посилань з відкритих Telegram-каналів українською та російською мовами. Для збору датасету було розроблено кравлер (англ. crawler) - спеціальну програму, яка автоматично обходила канали за стратегією в ширину та завантажувала дані. Процес збору включав в себе переходи за посиланнями на канали, вилучення публікації та метаінформації, збереження їх до бази даних. В результаті було створено датасет із 9753 Telegram-каналів.

В подальшому ці дані були оброблені та очищені з метою видалення дублікатів, а також непотрібних даних з набору. Процес відбувався за етапами, наведеними на рис. 1.

4. Розмітка датасету. Зібраний датасет є нерозміченим, тобто він не містить в собі відомостей про приналежність кожного спостереження до конкретного класу або категорії. Для отримання необхідних міток було застосовано класи тематики Telegram-каналів з ресурсу tgstat.com [19].

Використовуючи цей ресурс, було отримано мітки 38 класів для 5691 з 9753 каналів. Класи решти Telegram-каналів в створеному датасеті ресурс tgstat.com не надає. Перелік та розподіл міток зображено на рис. 2.

У процесі обробки та аналізу даних виявлено, що деякі мітки включали одні й ті ж класи даних, що призводило до небажаних розбіжностей. Так, категорії "Новини", "Політика" та "Блоги" частково перетинаються, що негативно впливає на якість роботи алгоритмів.

З метою покращення якості та уніфікації датасету було виключено такі амбівалентні мітки з вхідного набору даних. Розподіл класів та категорій оновленого датасету зображено на рисунку 3.

Таке оновлення спряло зменшенню надмірної складності даних та усуненню можливої багатозначності при прогнозуванні на моделі штучного інтелекту.

Діаграма на рис. 3 демонструє, що розподіл прикладів по класах став більш збалансованим, однак все ще нерівномірний. Це може призвести до проблеми коректності класифікації. Модель може бути схильною до прийняття більшого класу як "переважаючого", і неадекватно розрізняти менший клас.

Для вирішення проблеми незбалансованості на етапі навчання було визначено ваги класів. Вони були представлені масивом коефіцієнтів для обчислення помилки в процесі навчання.

Гіпотетично модель має краще адаптуватись до різниці у кількості прикладів для кожного класу і покращити якість прогнозів. Ваги було встановлено зворотньо-пропорційно до частоти класів у навчальному наборі.

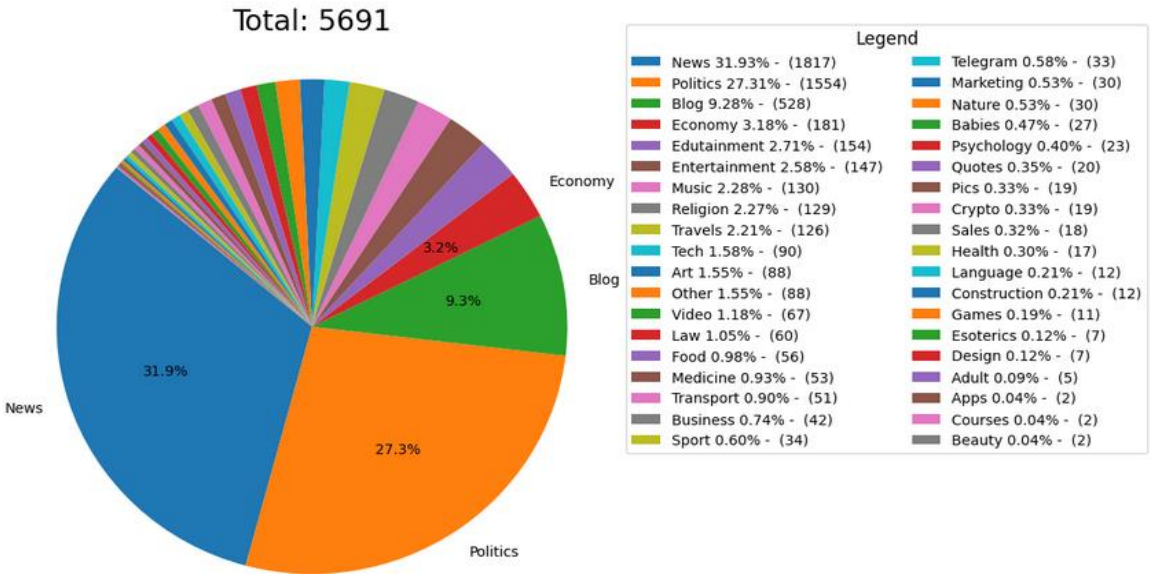

Рис. 2. Кругова діаграма розподілу класів розміченої частини набору даних

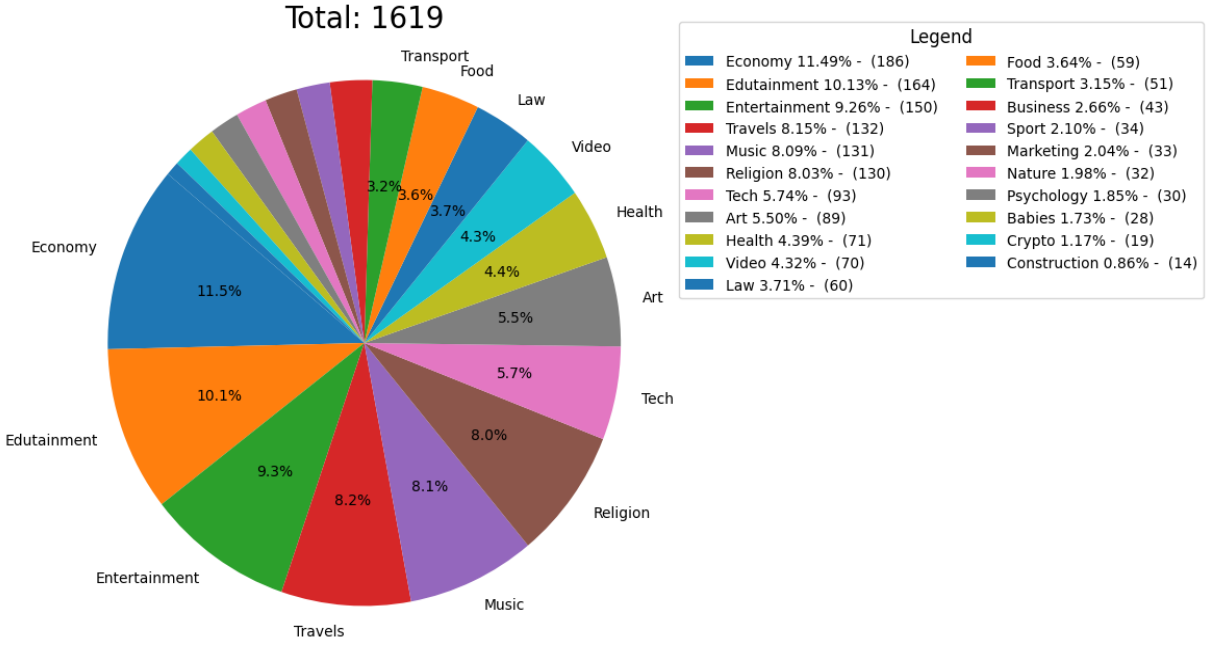

Рис. 3. Кругова діаграма розподілу класів після видалення амбівалентних категорій

Гіпотетично модель має краще адаптуватись до різниці у кількості прикладів для кожного класу і покращити якість прогнозів. Ваги було встановлено зворотньо-пропорційно до частоти класів у навчальному наборі.

5. Постановка задачі обчислювальних експериментів. Задача – мультикласова класифікація Telegram-каналів за тематикою (предметом обговорення) на основі базових підходів штучного інтелекту.

Варіанти цифрових представлень Telegram-каналів:

- агрегація векторів публікацій;
- вектор конкатенації TF-IDF ключових слів;
- агрегація векторів публікацій та ключових слів.

Навчальна вибірка Telegram-каналів та їх представлень становила 1295 каналів, тестова - 324 канали.

Для проведення класифікації були використані два основних набори моделей:

- класичні алгоритми машинного навчання: Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors та Support Vector Classifier (SVC), реалізації яких представлено в бібліотеці Python - scikit-learn.

- повнозв'язні нейронні мережі, реалізовані з використанням фреймворку Keras та TensorFlow [20].

Застосування обох підходів дозволило дослідити та порівняти ефективність методів глибокого навчання та традиційних алгоритмів машинного навчання для задачі класифікації тематики Telegram-каналів.

6. Пошук гіперпараметрів моделей класифікації. Пошук оптимальних гіперпараметрів є важливою складовою процесу налаштування моделей машинного навчання, метою якої є досягнення найкращої продуктивності. Гіперпараметри визначають структуру та математичні параметри поточної моделі. Наприклад, для нейромережі – це кількість

шарів, кількість нейронів в кожному шарі, швидкість навчання, функції активації та інші.

В табл. 1 наведено значення гіперпараметрів моделей за класичними алгоритмами машинного навчання, комбінації яких було застосовано для визначення оптимальних значень поточної моделі.

Гіперпараметрами моделювання нейромережі обрано кількість прихованих шарів та функції активації нейронів за шарами.

Іншими параметрами моделювання були:

- оптимізатори процесу навчання;
- розмір батчу;
- наявність вагів класів.

Пошук оптимальних значень гіперпараметрів нейромережі відбувався шляхом автоматизованого перебору. Всього було перевірено 6048 комбінацій гіперпараметрів для 3 цифрових представлень Telegram-каналів.

В табл. 2 наведено всі параметри моделювання нейромережі та їх значення.

Таблиця 1 – **Гіперпараметри моделей машинного навчання**

Модель	Назва параметрів	Значення
Logistic Regression	C	0.001, 0.01, 0.1, 1, 10, 100
Decision Tree	max_depth	None, 10, 20, 30, 40
	min_samples_split	2, 5, 10
Random Forest	n_estimators	50, 100, 200
	max_depth	None, 10, 20, 30
	min_samples_split	2, 5, 10
K-Nearest Neighbors	n_neighbors	3, 5, 7, 9
	weights	uniform, distance
SVC (Support Vector Classifier)	C	0.1, 1, 10
	kernel	linear, rbf

Таблиця 2 – **Гіперпараметри моделей нейронної мережі**

Параметри моделювання	Значення
Ваги класів (class_weights)	False, True
Кількість прихованих шарів (hidden layers)	[], [64], [128], [256], [512] [512,256,128, 64], [256,128, 64], [128, 64]
Функції активації (Hidden layers activations)	relu, tanh, leakyrelu, sigmoid, elu, prelu
Розмір батчу (batch_sizes)	[256, 128, 64]
Оптимізатори процесу навчання (optimizers)	Adam, SGD, RMSprop

7. Результати класифікації моделями з оптимальними гіперпараметрами. На всіх моделях з визначеними оптимальними гіперпараметрами було розв'язано задачу тематичної класифікації.

В табл. 3 наведено результати класифікації за метриками accuracy, precision, recall, які отримано відповідною моделлю (Classifier) з оптимальними значеннями гіперпарамтрів (Best Parameters) та типом цифрового представлення (Embeding Type).

Найкращі результати за класичними алгоритмами машинного навчання було отримано моделлю Support Vector Classifier з параметром регуляризації C = 10 та ядром радіальної базисної функції (RBF) на представленнях, створених агрегацією векторів публікацій.

В табл. 4 представлено найкращі комбінації нейромережевих моделей та цифрових представлень Telegram-каналів (Embeding Type) та використання вагів класів:

TRUE – наявність коефіцієнтів класів при визначенні помилки в процесі навчання,

FALSE - відсутність).

Таблиця 3 – **Результати класифікації телеграм каналів моделями машинного навчання**

Classifier	Best Parameters	Embeding Type	Accuracy	Precision	Recall
Logistic Regression	C: 0.1	Concatenated	0.783	0.789	0.708
Decision Tree	max_depth: 30; min_samples_split: 2	Concatenated	0.561	0.453	0.461
Random Forest	max_depth: None; min_samples_split: 5; n_estimators: 200	Concatenated	0.774	0.737967	0.633
K-Nearest Neighbors	n_neighbors: 9; weights: distance	Concatenated	0.759	0.722	0.669
SVC	C: 10; kernel: 'rbf'	Concatenated	0.777	0.734	0.699
Logistic Regression	C: 0.1	TfIdfVector	0.722	0.724	0.680
Decision Tree	max_depth: 40; min_samples_split: 10	TfIdfVector	0.389	0.354	0.312
Random Forest	max_depth: None; min_samples_split: 2, n_estimators: 200	TfIdfVector	0.688	0.744	0.545
K-Nearest Neighbors	n_neighbors: 9; weights': 'distance	TfIdfVector	0.710	0.684	0.624
SVC	C: 10; kernel: rbf	TfIdfVector	0.728	0.735	0.681
Logistic Regression	C: 0.1	MessagesMean	0.787	0.762	0.683
Decision Tree	max_depth: 30; min_samples_split: 2	MessagesMean	0.531	0.408	0.409
Random Forest	max_depth: 20; min_samples_split: 2, n_estimators: 200	MessagesMean	0.756	0.712	0.611
K-Nearest Neighbors	n_neighbors: 7; weights: distance	MessagesMean	0.744	0.705	0.647
SVC	C: 10; kernel: rbf	MessagesMean	0.762	0.749	0.674

Таблиця 4 **–Результати класифікації телеграм каналів нейронними мережами**

Model	Embeding Type	Class Weights	Accuracy	Precision	Recall	Auc	Prc
Adam-[64]-prelu-64	MessagesMean	FALSE	0.981	0.866	0.701	0.973	0.852
Adam-[128]-tanh-256	TfIdfVector	FALSE	0.976	0.822	0.642	0.954	0.776
Model-Adam-[64]-sigmoid-128	Concatenated	FALSE	0.983	0.865	0.756	0.980	0.8611

На другому наборі моделей найкращі результати по всіх метриках були досягнуті нейронною мережею з одним прихованим шаром на 64 нейрони, які використовували функцію активації Sigmoid та цифрових представленнях створених агрегацією векторів публікацій. Під час навчання цієї моделі використовувався оптимізатор Adam з налаштуваннями за замовчуванням. На вхід моделі цифрові представлення подавались в батчах, де кожен батч мав розмірність 128 елементи. Звідти позначка моделі Adam-[64]-sigmoid-128.

На рис. 4 представлено залежність значень метрик точності від епох навчання моделі Adam-[64]-sigmoid-128 з найкращим кінцевим результатом.

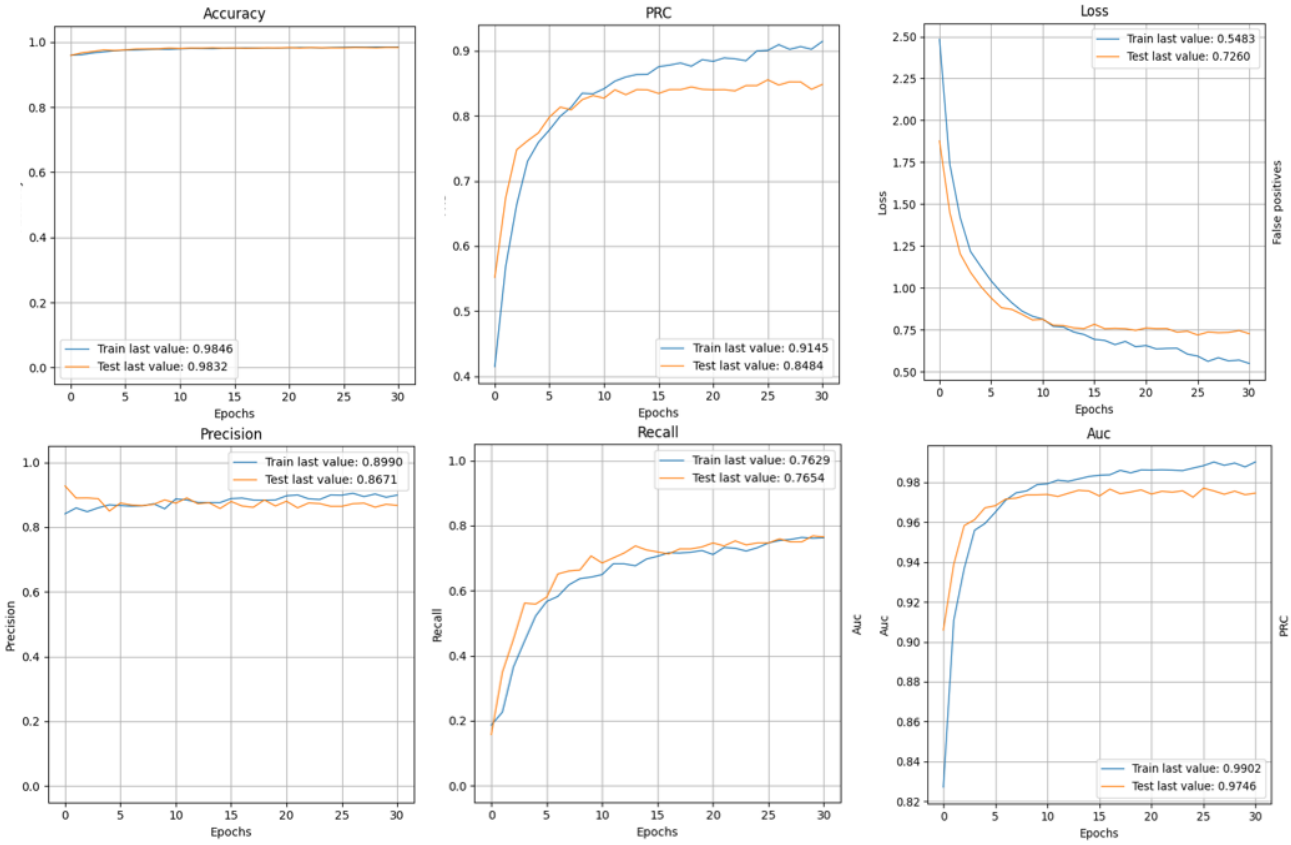

Рис. 4. Графіки метрики навчання нейронної мережі з найкращими результатами

З перших ітерацій модель Adam-[64]-sigmoid-128 продемонструвала хороші результати, що стало можливим завдяки використанню цифрових представлень Telegram-каналів, отриманих на попередньо навченій моделі SBERT, яка спеціалізується на ефективних векторних представленнях речень.

Використання, як входу класифікатора, вектора створеного на основі двох підходів, що включав поєднання інформації з окремих публікацій та статистично виділені ключові та словосполучення, продемонстрував кращі результати ніж кожен з підходів поодинці.

Необхідно відзначити, що отримана перевага має незначний характер, проте вона значуща.

Такий спосіб створення цифрового представлення дозволив узагальнити, зберегти ключові теми та суттєву інформацію з різних публікацій Telegram-каналів в одному компактному представленні.

Висновки

1. Для ембедінгу Telegram-каналів на мережі SBERT визначено такі підходи:

агрегація векторів публікацій,

конкатенація ключових слів за методом TF-IDF,

поєднання перших двох підходів.

Таке рішення дозволило вирішити проблему обмеженості кількості вхідних токенів у моделі SBERT та підвищити ефективність обробки текстової інформації, забезпечуючи можливість в обмеженому за розміром векторі представити більшість ключових особливостей повідомлень Telegram-каналів.

2. Представлено конвеєр обробки текстових даних для цифрового представлення Telegram-каналів.

3. Створено датасет з 9753 цифрових представлень повідомлень Telegram-каналів.

4. На основі ресурсу tgstat.com визначено мітки 38 класів для 5691 з 9753, що склали навчальну вибірку.

5. Експериментально визначено оптимальні за точністю гіперпараметри моделей тематичної класифікації:

- за нейромережевою моделлю з одним прихованим шаром на 64 нейрони, які використовували функцію активації Sigmoid та оптимізатор Adam та забезпечили 98.3% точноності;

- за моделлю машинного навчання Logistic Regression з рівнем регуляризації C = 0.1, яка забезпечила 78.7% точноності.

СПИСОК ЛІТЕРАТУРИ

1. Скринінг українського суспільства протягом повномасштабної війни. Національна рада України з питань телебачення і радіомовлення. URL: https://www.nrada.gov.ua/wp-content/uploads/2022/05/GradusResearch_Report_Suspilne_50K_27042022.pdf .
2. Mikolov, T., Chen, K., Corrado, G., & Dean, J. Efficient estimation of word representations in vector space. 2013. *arXiv preprint arXiv:1301.3781*.
3. Pennington, J., Socher, R., & Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* 2014, October. (pp. 1532-1543).
4. Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, *5*, 135-146. 2017.
5. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 2018.
6. Reimers, N., & Gurevych, I.. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *arXiv preprint arXiv:1908.10084*. 2019
7. Barbaro, F., & Skumanich, A. Addressing socially destructive disinformation on the web with advanced AI tools: Russia as a case study. In *Companion Proceedings of the ACM Web Conference 2023* (pp. 204-207). 2023, April.
8. Wei, F., & Nguyen, U. T. Twitter Bot Detection Using Neural Networks and Linguistic Embeddings. *IEEE Open Journal of the Computer Society*. 2023.
9. Hugging Face – The AI community building the future. Hugging Face. URL: https://huggingface.co/ (date of access: 30.11.2023).
10. NLTK : Natural Language Toolkit. NLTK :: Natural Language Toolkit. URL: https://www.nltk.org/ (date of access: 30.11.2023).
11. Ukrainian-Stopwords. GitHub. URL: https://github.com/skupriienko/Ukrainian-Stopwords (date of access: 30.11.2023).
12. stopwords-iso/stopwords-ru. GitHub. URL: https://github.com/stopwords-iso/stopwords-ru (date of access: 30.11.2023).
13. Korobov M.: Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts,pp 320-332. 2015.
14. scikit-learn: machine learning in Python. scikit-learn. URL: https://scikit-learn.org/ (date of access: 30.11.2023).
15. The Pushshift Telegram Dataset / B. Jason et al. Zenodo. URL: https://zenodo.org/records/3607497 (date of access: 30.11.2023).
16. Dataset-for-teenagers-chat-in-Telegram-groups: Dataset for teenagers' chat in Telegram groups (Persian). GitHub. URL: https://github.com/imRezaAlie/Dataset-for-teenagers-chat-in-Telegram-groups (date of access: 30.11.2023).
17. Temnikova I. TRACES Bulgarian Telegram Dataset Annotated with Linguistic Markers of Lies. Zenodo. URL: https://zenodo.org/records/7614294 (date of access: 30.11.2023).
18. Crypto telegram groups. Kaggle: Your Machine Learning and Data Science Community. URL: https://www.kaggle.com/datasets/aagghh/crypto-telegram-groups (date of access: 30.11.2023).
19. Telegram channels and groups catalog. *TGStat.com*. URL: https://tgstat.com/ (date of access: 30.11.2023).
20. Keras: The high-level API for TensorFlow | TensorFlow Core [Electronic resource] // TensorFlow. – Mode of access: https://www.tensorflow.org/guide/keras (date of access: 08.12.2023)

Received (Надійшла) 09.12.2023
Accepted for publication (Прийнята до друку) 24.01.2024

Digital representations of Telegram channels

S. Shapovalova, A. Sofiienko

Abstract. The subject of research of this article is digital representations of textual information resources on the example of Telegram channels. The purpose of the work is to determine the optimal method of forming digital representations of Telegram channels for further thematic classification. The following tasks are solved in the article:definition of approaches to the formation of the input vector; determination of the stages of text data processing for the digital representation of the Telegram channel; creation of a dataset of digital representations of Telegram channels; dataset marking for solving the classification problem; determination of hyperparameters of optimal classification models. The following results were obtained: a dataset of digital representations of Telegram channels formed on the basis of the SBERT network was created using three approaches: aggregation of publication vectors, concatenation of keywords using the TF-IDF method, and a combination of the first two approaches; it was determined that the approach of concatenation of keywords using the TF-IDF method and the combination of the first two approaches to the formation of digital representations of Telegram channels based on text publications is the most effective for further classification by topic; the optimal hyperparameters of the thematic classification models are determined in terms of accuracy: Logistic Regressio and deep learning neural networks. A promising direction of further research is the evaluation of the application of the proposed digital representations to clustering and search tasks.

Keywords: natural language text processing, BERT, thematic classification of messages, representation learning.