ДОДАТОК А

Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії

5. Golian N., Afanasieva I.., Nazarenko D. Investigation of the Deep Learning Approaches to Classify Emotions in Texts. COLINS, 2021, pp. 206-224.

6. Kyrychenko I., Tereshchenko G., Gruzdo I., Cherednichenko O. Application of paragraphs vectors model for semantic text analysis // Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2020), April 23-24, 2020, Lviv, Ukraine. Vol. I. P. 283-293. 2020. SCOPUS.

ДОДАТОК Б

Звіт з результатами перевірки на унікальність тексту в базі ХНУРЕ

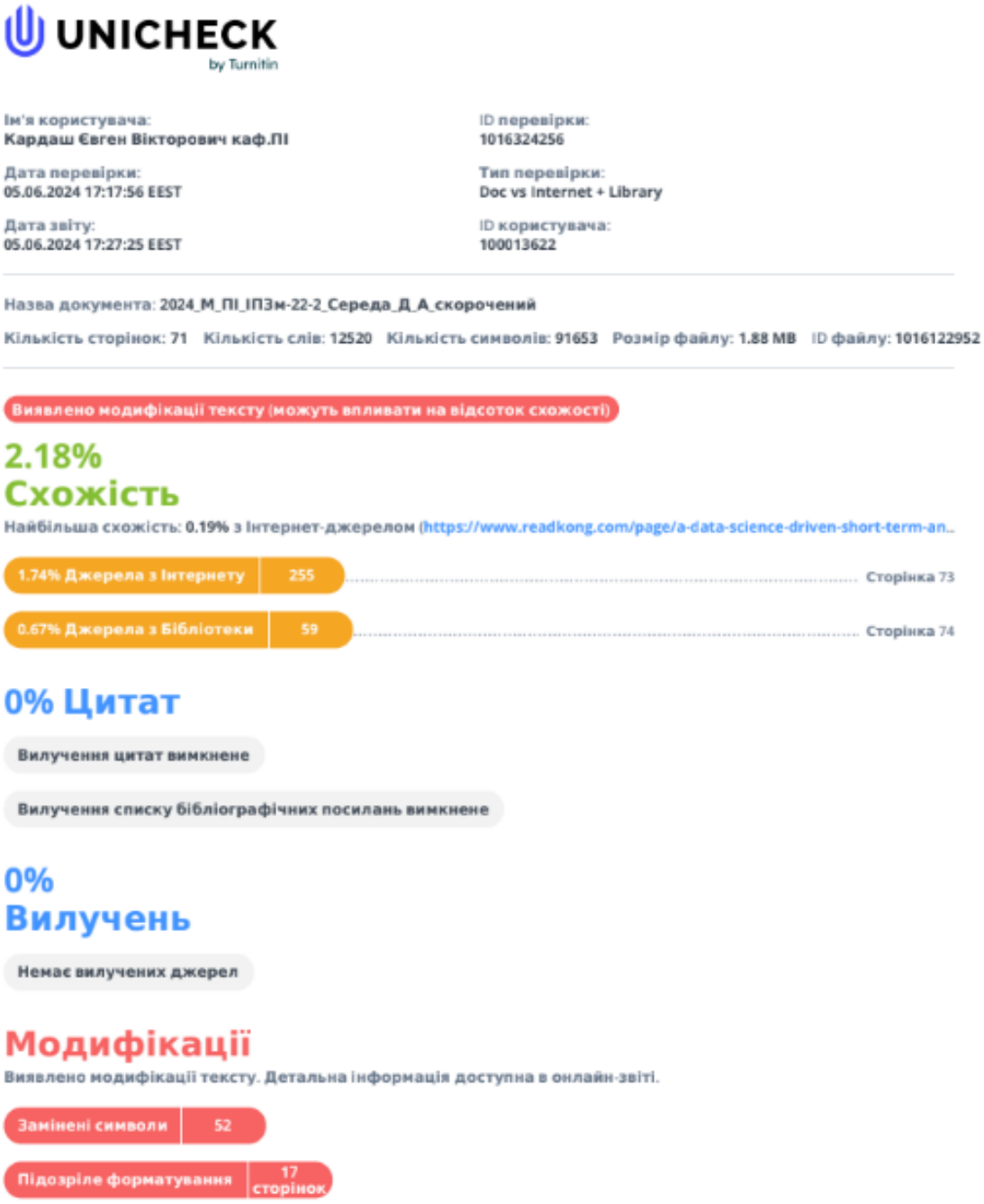


Рисунок Б.1 – Перевірка на доброчесність

ДОДАТОК В

Слайди презентації

Рисунок В.1 – Титульний слайд

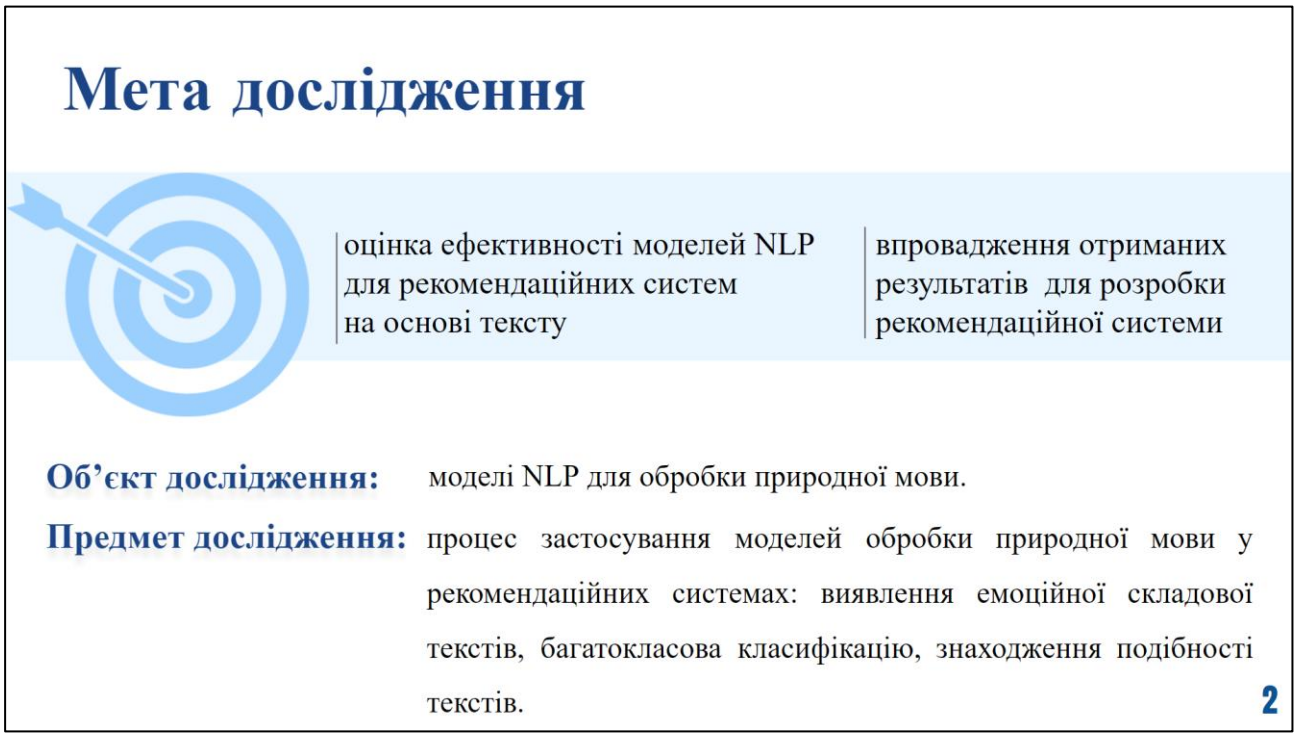


Рисунок В.2 – Мета дослідження

Рисунок В.3 – Аналіз предметної області

Рисунок В.4 – Постановка задачі

Вибір моделей для експериментів

Викреслення моделей за принципом Парето

	Розмір вбудованого словника	Економія за обсягом пам'яті	Open Source	Простота навчання	Ефективність контекстуалізації
BERT	30522	69.5	1	2	5
GPT-2	50257	64	1	3	4
~~GPT-3~~	~~50257~~	~~0~~	~~0~~	~~1~~	~~4~~
~~Word2Vec~~	~~50300~~	~~69.9~~	~~1~~	~~5~~	~~2~~
XLNet	32000	68.6	1	3	5
FastText	50300	69.9	1	5	3

5

Рисунок В.5 – Вибір моделей для експериментального дослідження

Вибір моделей для експериментів

Розрахунок коефіцієнтів корисності за лінійною адитивною згорткою

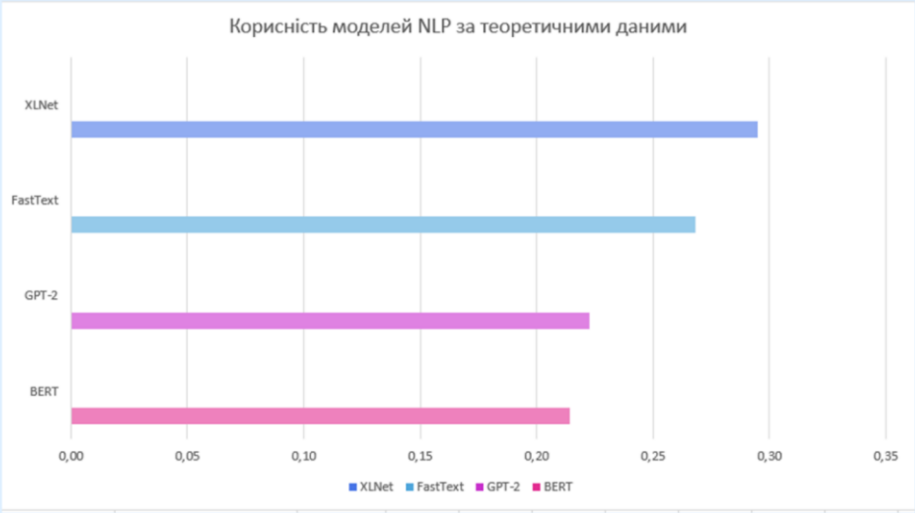
	Розмір вбудованого словника	Економія за об'ємом пам'яті	Open Source	Простота навчання	Ефективність контекстуалізації	K
BERT	0	0,932	1	0	1	**0,21**
GPT-2	0,998	0	1	0,333	0,5	**0,22263**
FastText	1	1	1	1	0	**0,26836**
XLNet	0,7	0,5833	1	0,67	0,75	**0,29478**
βj	0,27	0,133333333	0,07	0,2	0,33	
αj	0,370644922	0,397566891	0,25	0,499251123	0,444444444	

6

Рисунок В.6 – Вибір моделей для експериментального дослідження

Вибір моделей для дослідження

Обрані моделі



7

Рисунок В.7 – Вибір моделей для експериментального дослідження

Експериментальні дослідження

Експериментальні дослідження будуть складатися з наступних етапів:

– формування датасетів, для трьох задач:
  1. сентиментальний аналіз;
  2. багатокласова класифікація фільмів за жанрами;
  3. передбачення подібності текстів;
– очищення та обробка датасетів;
– програмування та навчання обраних для дослідження моделей;
– оцінка моделей для трьох задач;
– аналіз результатів та моделей за отриманими метриками.

8

Рисунок В.8 – План проведення експериментальних досліджень

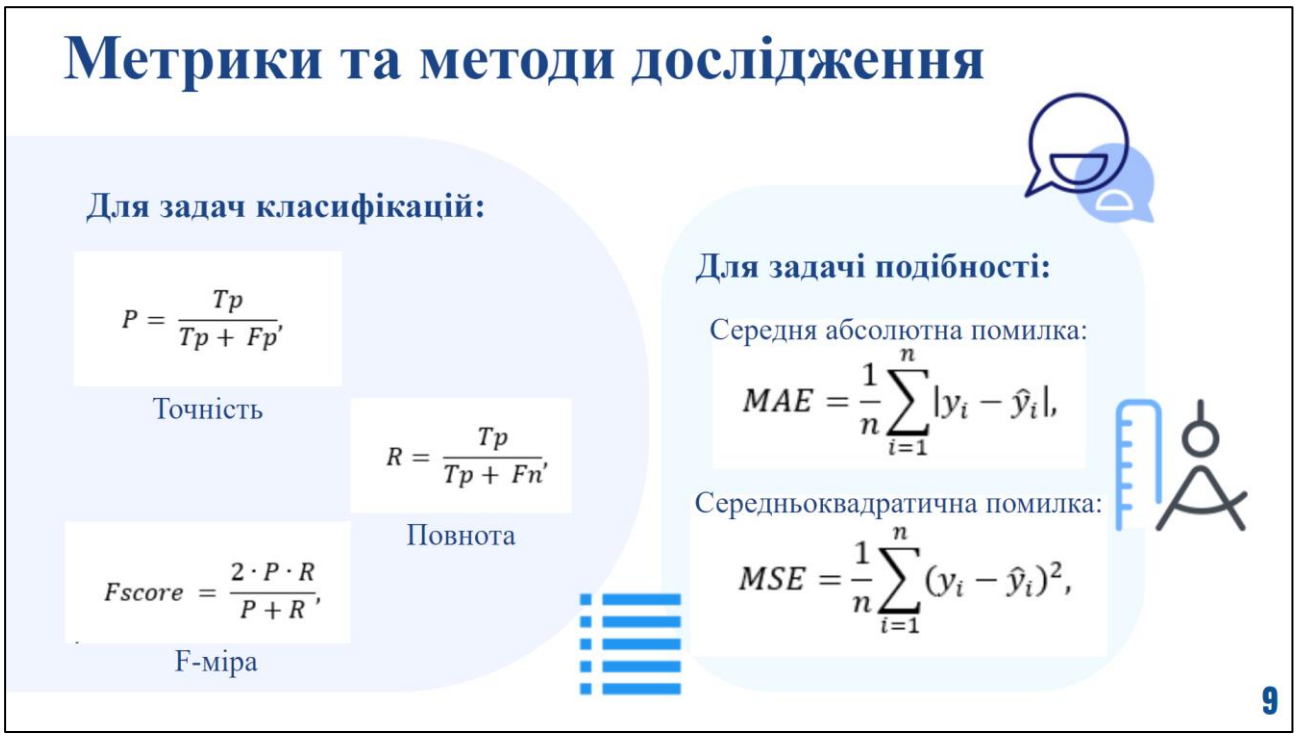


Рисунок В.9 – Метрики та методи дослідження

Рисунок В.10 – Інструменти та технології

Рисунок В.11 – Набір даних для емоційного аналізу

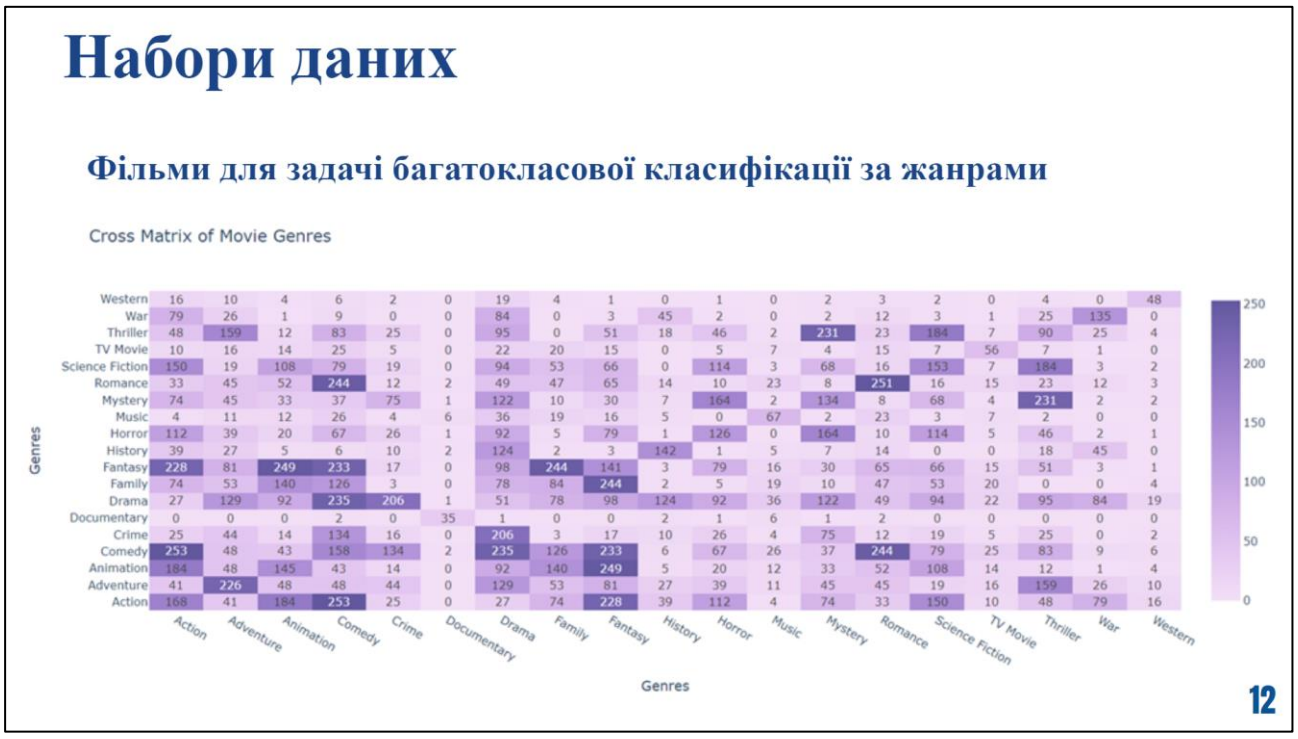


Рисунок В.12 – Набір даних для класифікації за жанрами

Набори даних

Пара текстів та подібностей для знаходження подібностей

```
df = pq.read_table("train-00000-of-00001.parquet").to_pandas()
df.head(10)
```

	sentence1	sentence2	similarity_score
0	A plane is taking off.	An air plane is taking off.	5.00
1	A man is playing a large flute.	A man is playing a flute.	3.80
2	A man is spreading shreded cheese on a pizza.	A man is spreading shredded cheese on an uncoo...	3.80
3	Three men are playing chess.	Two men are playing chess.	2.60
4	A man is playing the cello.	A man seated is playing the cello.	4.25
5	Some men are fighting.	Two men are fighting.	4.25
6	A man is smoking.	A man is skating.	0.50
7	The man is playing the piano.	The man is playing the guitar.	1.60
8	A man is playing on a guitar and singing.	A woman is playing an acoustic guitar and sing...	2.20
9	A person is throwing a cat on to the ceiling.	A person throws a cat on the ceiling.	5.00

13

Рисунок В.13 – Набір даних для подібності текстів

Налаштування програмного середовища

Попередня обробка тексту

```
class CustomTokenizer:
    def __init__(self):
        self.pattern = re.compile(r"\w+|[^\w\s]")

    def tokenize(self, text):
        tokens = self.pattern.findall(text)
        return tokens

def preprocess_text(text):
    text = text.lower()
    text = re.sub(r"[^a-zA-Z0-9]", " ", text)
    tokenizer = CustomTokenizer()
    tokens = tokenizer.tokenize(text)
    tokens = [token for token in tokens if token not in stop_words]
    tokens = [lemmatizer.lemmatize(token) for token in tokens]
    return " ".join(tokens)
```

14

Рисунок В.14 – Попередня обробка тексту

Налаштування програмного середовища

Завантаження попередньо навчених моделей та токенізаторів

BERT

```
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=6)
```

GPT-2

```
model_config = GPT2Config.from_pretrained('gpt2', num_labels = 6)
tokenizer = GPT2Tokenizer.from_pretrained('gpt2')

tokenizer.padding_side = "left"
tokenizer.pad_token = tokenizer.eos_token
model = GPT2ForSequenceClassification.from_pretrained('gpt2', config=model_config)
model.resize_token_embeddings(len(tokenizer))
model.config.pad_token_id = model.config.eos_token_id
```

XLNet

```
tokenizer = XLNetTokenizer.from_pretrained('xlnet-base-cased')
model = XLNetForSequenceClassification.from_pretrained('xlnet-base-cased', num_labels=6)
```

15

Рисунок В.15 – Завантаження моделей

Налаштування програмного середовища

Токенізація та підготовка даних

Токенізація

```
train_encodings = tokenizer(train_texts, truncation=True, padding=True, max_length=128)
test_encodings = tokenizer(test_texts, truncation=True, padding=True, max_length=128)

train_labels = torch.tensor(train_labels)
test_labels = torch.tensor(test_labels)

train_dataset = TensorDataset(torch.tensor(train_encodings['input_ids']),
                              torch.tensor(train_encodings['attention_mask']), train_labels)
train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True)

test_dataset = TensorDataset(torch.tensor(test_encodings['input_ids']),
                             torch.tensor(test_encodings['attention_mask']), test_labels)
test_loader = DataLoader(test_dataset, batch_size=16, shuffle=False)
```

Перетворення в тензори

Підготовка даних до подання до моделей

16

Рисунок В.16 – Токенізація

Налаштування програмного середовища

Навчання та оцінка моделей-трансформерів

```
optimizer = torch.optim.AdamW(model.parameters(), lr=2e-5)
model.train()
for epoch in range(7):
    for batch in train_loader:
        optimizer.zero_grad()
        input_ids, attention_mask,
        outputs = model(input_ids=
                        attention_
        loss = outputs.loss
        loss.backward()
        optimizer.step()
```

```
model.eval()
predictions = []
true_labels = []
with torch.no_grad():
    for batch in test_loader:
        input_ids, attention_mask, labels = batch
        outputs = model(input_ids=input_ids, attention_mask=
                        attention_mask)
        logits = outputs.logits
        predictions.extend(torch.argmax(logits,
                                        axis=1).cpu().numpy())
        true_labels.extend(labels.cpu().numpy())


report = classification_report(true_labels, predictions)
```

17

Рисунок В.17– Навчання та оцінка

Налаштування програмного середовища

Навчання та оцінка fasttext

```
model = train_supervised(input="train.txt", autotuneValidationFile="valid.txt")
model.save_model("trained_model.bin")

def load_test_data(filepath):
    texts = []
    true_labels = []
    with open(filepath, 'r') as f:
        for line in f:
            parts = line.strip().split(' ', 1)
            labels = [label.replace('__label__', '') for label in parts[0].split()]
            text = parts[1]
            texts.append(text)
            true_labels.append(labels)
    return texts, true_labels

test_texts, test_true_labels = load_test_data('test.txt')
predicted_labels = []
for text in test_texts:
    labels, _ = model.predict(text, k=1)
    predicted_labels.append([label.replace('__label__', '') for label in labels])
```

18

Рисунок В.18 – Навчання fasttext

Результати експериментів

Результати для задач класифікацій

BERT

Завдання	Точність	Повнота	F-score
Емоційний зміст	0,806	0,848	0,82
За жанрами	0.866	0,86	0,858

GPT-2

Завдання	Точність	Повнота	F-score
Емоційний зміст	0,724	0,764	0,738
За жанрами	0,82	0,8	0,81

19

Рисунок В.19 – Результати експериментів

Результати експериментів

Результати для задач класифікацій

XLNet

Завдання	Точність	Повнота	F-score
Емоційний зміст	0.84	0.80	0.82
За жанрами	0.83	0.87	0.849

Fasttext

Завдання	Точність	Повнота	F-score
Емоційний зміст	0.84	0.80	0.85
За жанрами	0.66	0.68	0.66

20

Рисунок В.20 – Результати експериментів

Результати експериментів

Результати для знаходження подібності текстів

Модель	MSE	MAE
BERT	0.81	0.71
GPT-2	1.488	1.037
XLNet	1.447	1.0193
FastText	1,25	0.93

21

Рисунок В.21 – Результати експериментів

Аналіз результатів

Для задачі емоційного аналізу

Модель	F-Score показник	Час виконання (год.)
BERT	**0.85**	**12**
GPT-2	0.738	17
XLNet	0.82	13,5
FastText	**0,85**	**4,5**

22

Рисунок В.22 – Аналіз результатів

Аналіз результатів

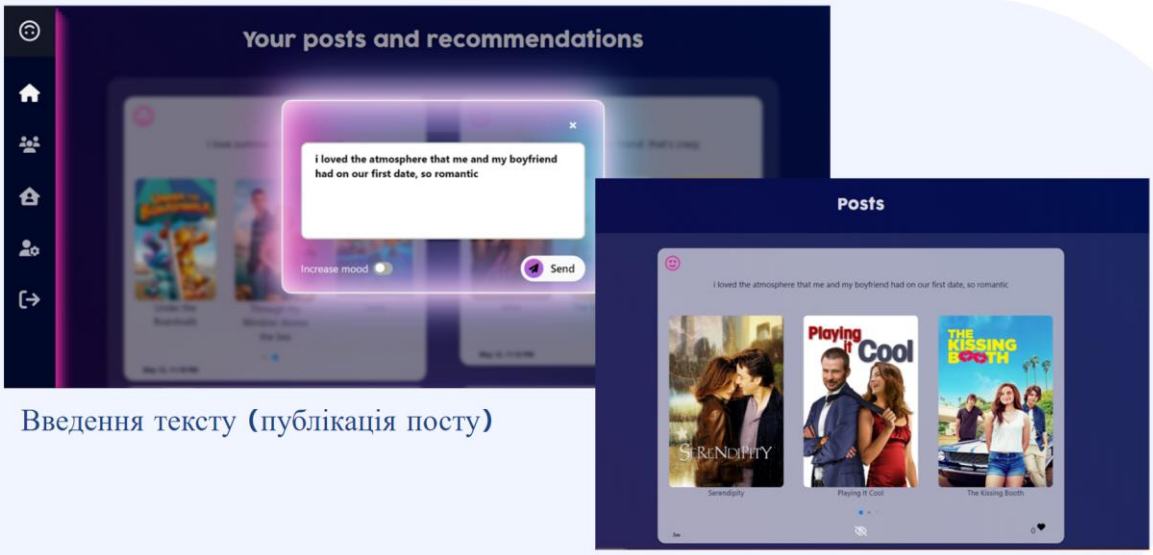
Для багатокласової класифікації жанрів

Модель	F-score показник	Час виконання (год.)	K
BERT	0,82	6,5	**0,275337522**
GPT	0,81	0	0,161277477
XLNet	0,849	2	0,203525447
FastText	0,66	6	0,234859553
βj	0,625	0,25	
αj	0,318572794	0,068965517	

23

Рисунок В.23 – Аналіз результатів

Використання результатів дослідження



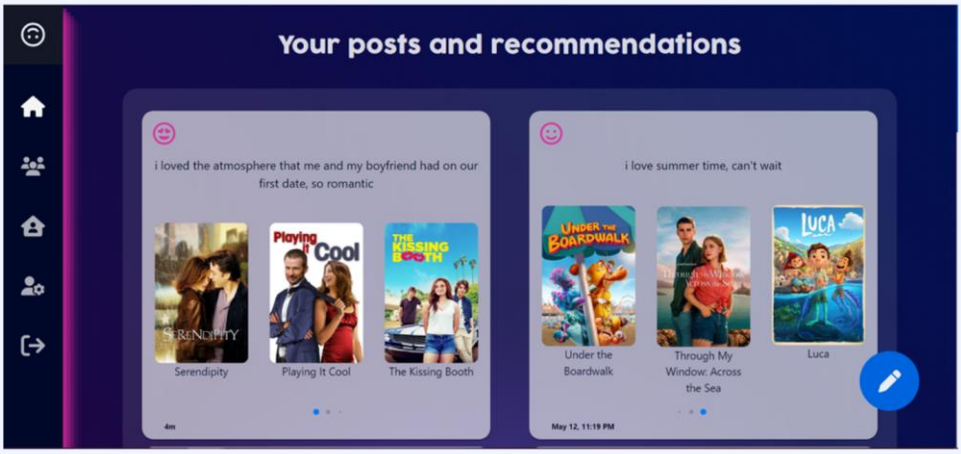
Введення тексту (публікація посту)

Опублікований пост з рекомендаціями

24

Рисунок В.24 – Розроблена рекомендаційна система

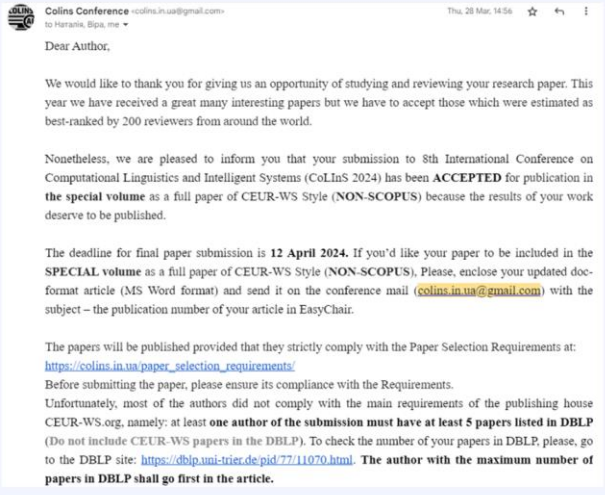
Використання результатів дослідження



Перегляд історії публікацій та рекомендацій

Рисунок В.25 – Розроблена рекомендаційна система

Апробація



Colins Conference <colins.in.ua@gmail.com> Thu, 28 Mar, 14:56

to Haraniia, Bipa, me

Dear Author,

We would like to thank you for giving us an opportunity of studying and reviewing your research paper. This year we have received a great many interesting papers but we have to accept those which were estimated as best-ranked by 200 reviewers from around the world.

Nonetheless, we are pleased to inform you that your submission to 8th International Conference on Computational Linguistics and Intelligent Systems (CoLInS 2024) has been **ACCEPTED** for publication in **the special volume** as a full paper of CEUR-WS Style (**NON-SCOPUS**) because the results of your work deserve to be published.

The deadline for final paper submission is **12 April 2024.** If you'd like your paper to be included in the **SPECIAL volume** as a full paper of CEUR-WS Style (**NON-SCOPUS**), Please, enclose your updated doc-format article (MS Word format) and send it on the conference mail (colins.in.ua@gmail.com) with the subject – the publication number of your article in EasyChair.

The papers will be published provided that they strictly comply with the Paper Selection Requirements at: https://colins.in.ua/paper_selection_requirements/

Before submitting the paper, please ensure its compliance with the Requirements.

Unfortunately, most of the authors did not comply with the main requirements of the publishing house CEUR-WS.org, namely: at least **one author of the submission must have at least 5 papers listed in DBLP** (Do not include CEUR-WS papers in the DBLP). To check the number of your papers in DBLP, please, go to the DBLP site: https://dblp.uni-trier.de/pid/77/11070.html. **The author with the maximum number of papers in DBLP shall go first in the article.**

МІЖНАРОДНИЙ НАУКОВИЙ ЖУРНАЛ «ІНТЕРНАУКА»

ІНТЕРНАУКА

INTERNATIONAL SCIENTIFIC JOURNAL «INTERNAUKA»

CERTIFICATE

THE PUBLICATION

Series: MN № 78256

this is to certify that Stud.

Daria Sereda

has published the article titled:

«INVESTIGATION OF THE NATURAL LANGUAGE PROCESSING MODELS FOR TEXT-BASED RECOMMENDATION SYSTEMS»

in the

INTERNATIONAL SCIENTIFIC JOURNAL "INTERNAUKA". Issue 4(159)/2024

Web-site publishing:

https://doi.org/10.25313/2520-2057-2024-4-9846

Chief Editor: Dmytro KOVALENKO

26

Рисунок В.26 – Апробація

Висновки

У результаті виконання кваліфікаційної роботи магістра було:

- проведено аналіз предметної області та літератури;
- проведено аналіз існуючих методів NLP;
- проведено теоретичне дослідження на 6 моделях NLP для вибору моделей для експериментального дослідження;
- проведено експериментальні дослідження для трьох задач: емоційний аналіз, багатокласова класифікація фільмів за жанрами та знаходження подібності між текстами;
- проведено аналіз результатів та оцінка моделей та знайдено найефективніші моделі;
- надано оцінку використання результатів та розробка рекомендаційної системи з використанням найефективніших за результатами моделей NLP.

ОПУБЛІКОВАНІ НАУКОВІ РОБОТИ:

Sereda D., Golian N. Investigation of the natural language processing models for text-based recommendation systems // International scientific journal "Internauka". — 2024. — №4. https://doi.org/10.25313/2520-2057-2024-4-9846

27

Рисунок В.26 – Висновки

ДОДАТОК Г

Апробація результатів роботи

Рисунок Г.1 – Титульна сторінка журналу

Рисунок Г.2 – Сертифікат публікації

UDC 004.051

Sereda Daria
Student of the
Kharkiv National University of Radio Electronics

Golian Nataliia
Scientific Director, Associate Professor of the
Department of Software Engineering
Kharkiv National University of Radio Electronics

DOI: 10.25313/2520-2057-2024-4-9846

INVESTIGATION OF THE NATURAL LANGUAGE PROCESSING MODELS FOR TEXT-BASED RECOMMENDATION SYSTEMS

Summary. *This study investigates the application of natural language processing (NLP) methods in recommendation systems based on textual data. It identifies models with the highest utility coefficients for specific tasks based on criteria crucial for recommendation provision. The study underscores the importance and effectiveness of employing NLP methods in recommendation systems, utilizing the findings to develop a recommendation system.*

Key words: *recommendation systems, NLP models, natural language processing, text, BERT, fastText, movies, sentiment analysis.*

In today's digital landscape, users are inundated with a vast array of online content and platforms, making effective navigation essential. Recommender systems have become indispensable tools, guiding users through this abundance by offering personalized experiences. Leveraging advanced algorithms and machine learning, these systems analyze user behavior and preferences to deliver tailored recommendations that anticipate evolving needs.

There are two main types of recommender systems — content-based filtering and collaborative filtering. In content-based filtering systems, objects are determined based on their features. Based on the available attributes in the objects that the user rated, the interest profile of the new user is studied, in other words, this is a keyword-based recommendation system. Thus, content-based recommendation systems use algorithms that offer users similar items they liked in the past or are currently studying [1].

The integration of Natural Language Processing (NLP) techniques marks a significant shift in recommendation systems. NLP equips these systems to understand and extract insights from textual data, such as user reviews and product descriptions. By harnessing NLP, recommender systems can better comprehend user preferences and deliver more contextually relevant recommendations.

Transformer models, such as BERT and GPT, have demonstrated remarkable performance in natural language understanding tasks, making them well-suited for tasks like sentiment analysis and semantic understanding in recommender systems. Similarly, word embedding models, like fastText, offer efficient representations of words in vector space, facilitating meaningful comparisons and associations between words. By leveraging these sophisticated techniques, recommender systems can better comprehend user preferences and deliver more accurate and personalized recommendations. Therefore, exploring these advanced models is essential for enhancing the effectiveness and adaptability of recommendation algorithms in diverse application domains so this investigation is focused on relatively new transformer models and word embedding model.

The following metrics were chosen to evaluate the models.

Recommendation precision is a measure of how often the model correctly classifies positive predictions. It increases as the ratio of the number of correct positive predictions to the total number of positive predictions. Mathematically, it is defined as (1).

$$P = \frac{Tp}{Tp + Fp}, \quad (1)$$

where P is precision;

Tp is the number of correctly classified instances as positive.

Рисунок Г.3 – Копія статті

Fp is the number of incorrectly classified instances as positive.

Recall quantifies is the ability of the model to capture all relevant instances of the positive prediction. It is calculated using the formula (2).

$$R = \frac{Tp}{Tp + Fn}, \quad (2)$$

where R is recall;

Tp is the number of correct positive forecasts;

Fn is the number of false negative predictions.

The F-score is a combined measure of precision P and a Recall R. It is calculated using the harmonic mean between P and R (3).

$$Fscore = \frac{2 \cdot P \cdot R}{P + R}, \quad (3)$$

where P — precision value;

R — recall value.

To facilitate the investigation effectively, two main goals have been defined:

1. Sentiment analysis. Sentiment analysis enables a deeper understanding of the emotional context embedded within user-generated text, facilitating the creation of recommendations that resonate on a more personal level. By discerning the user's emotional state, recommendations can be customized to better suit their current needs, mood, and preferences.

2. Formulating recommendations based on users' preferences enhances the effectiveness of recommendation systems across various domains. By analyzing user-generated content or interactions and providing suggestions tailored to individual tastes, these systems offer more accurate and satisfying recommendations, thereby improving overall user satisfaction and engagement with the platform or service.

Within the scope of this study, the tasks above are formulated as follows:

1. The sentiment analysis within user-generated text;

2. Providing movie genre recommendations based on user-preferred movie attributes.

For the first task, a dataset containing users' tweets and labels for emotional states was utilized: sadness (0), joy (1), love (2), anger (3), fear (4), surprise (5) [2]. The training set comprises 15,969 unique values. 4666 tweets represent sadness, 5362 tweets — joy, 1304 — love, 2159 — anger, 1937 — fear and 572 represent surprise.

For the second task, the dataset was created with the use of The Movie Database (TMDb) API [3]. This dataset contains information about movies fetched from endpoints The resulting dataset consists of 20,000 rows and includes the following columns: title — the title of the movie; description — a brief overview or synopsis of the movie; genres — a list of genres associated with the movie: [genre 1, genre 2, ... genre n]. There are a total of 18 genres listed, with varying counts of movies for each genre.

Experiments were conducted on GPT-2, BERT, XLNet and FastText models. The process includes data pre-processing, model training, model testing and evaluation. The Python programming language was used to conduct experiments and evaluate models.

To draw conclusions regarding the use of models for text-related tasks in recommendation systems, two tables comparing the results were created. As metrics for comparison, F-score and training time were chosen. Table 1 represents results for sentiment analysis problem and table 2 — genre recommendations. As metrics for comparison, F-score and training time were chosen.

While BERT offers competitive performance in sentiment analysis, FastText outperforms models in terms of both accuracy and efficiency. Its ability to achieve a high F-Score with minimal computational resources makes FastText an attractive choice for sentiment analysis tasks, particularly in scenarios where timely processing and resource constraints are paramount.

Table 1

Comparison for sentiment analysis

Model	F-Score value	Execution time (h.)
BERT	0.858	12
GPT	0.738	17
XLNet	0.82	13,5
FastText	0,85	4,5

Table 2

Comparison for genre recommendations

Model	F-Score value	Execution time (h.)
BERT	0.82	11,5
GPT	0.81	18
XLNet	0.849	16
FastText	0,66	6

Рисунок Г.4 – Копія статті

In order to find the model that best fits, we will use a linear additive convolution with weighting coefficients to calculate the utility coefficient, as certain criteria have a greater impact on effectiveness and model selection. The convolution formula is given in (4).

$$Z^* = \max_{i=1,m} \sum_{j=1}^{n} \alpha_j \beta_j \alpha_{i_j} \quad (4)$$

where α_j — normalization factors,

β_j — weighting coefficients.

Some criteria contribute more and are more important than others, so it is necessary to introduce weighting coefficients. Using the ranking method, we will evaluate the importance of the criteria in the following order: the most important criterion is F-Score, its rank we set up as 5. Execution time on the other hand is also important, so its rank is 2. Rank sum = 8. The utility calculation of models using linear additive convolution is illustrated in figure 1.

Based on the utility scores calculated using the linear additive convolution, the BERT model appears to be the most suitable choice for recommendation solutions among the models evaluated.

In conclusion, the integration of BERT and FastText offers a comprehensive approach to recommendation systems, enabling them to deliver personalized and emotionally resonant suggestions to users, thereby enhancing the overall user experience and satisfaction. Therefore, the results of the conducted research can be used to develop recommendations systems, specifically a recommendation system for providing movie recommendations based on the user's post, taking into account emotions and post's context.

	F-score	Time saving (h)	K
BERT	0,82	6,5	**0,275337522**
GPT	0,81	0	0,161277477
XLNet	0,849	2	0,203525447
FastText	0,66	6	0,234859553
βj	0,625	0,25	
αj	0,318572794	0,068965517	

Fig. 1. Model's utility for genre recommendations

References

1. Aggarwal C.C. Recommender Systems: The Textbook. 2016. 519 p. (1st Edition).

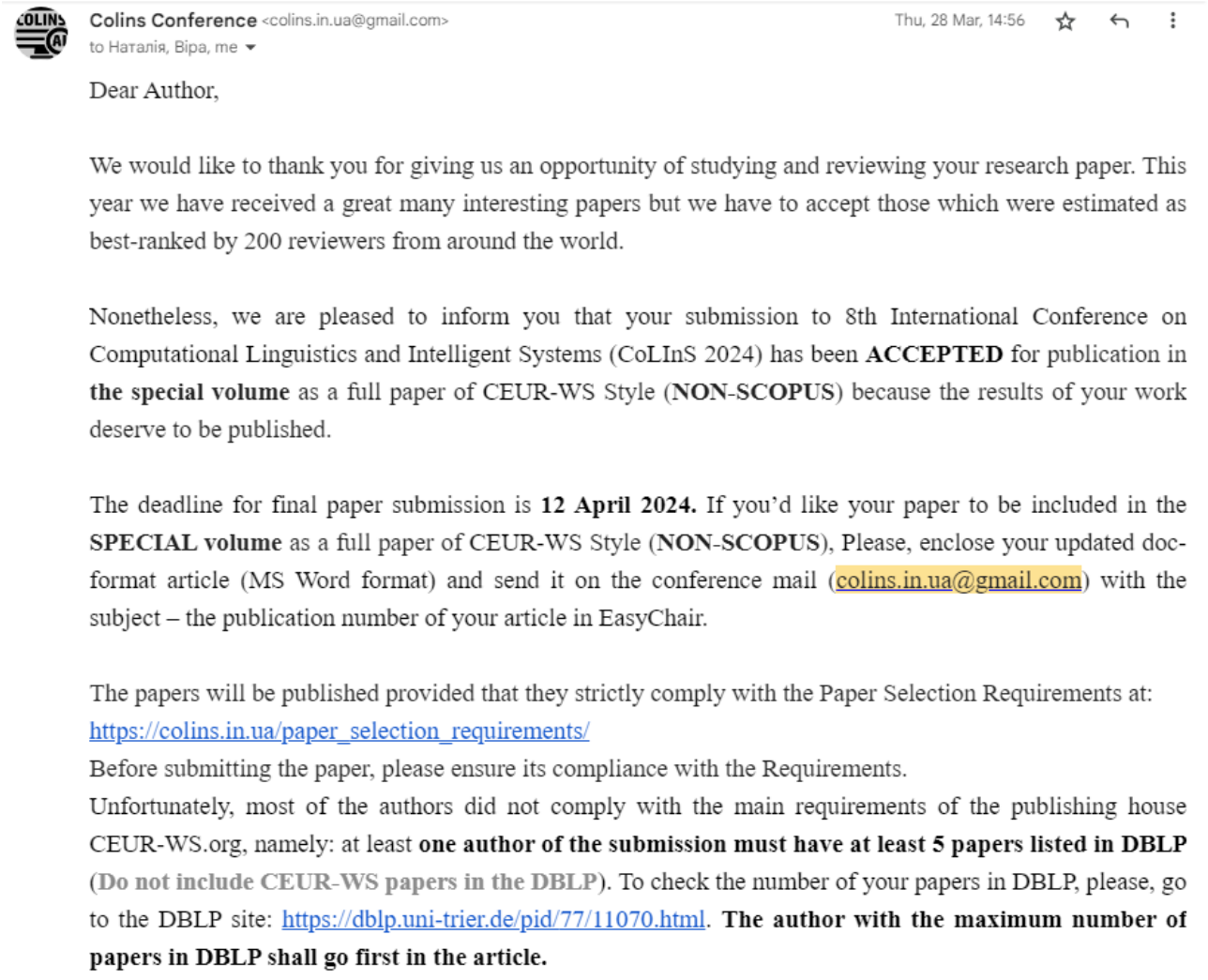
2. Riloff E., Tsujii J., Chiang D., Hockenmaier J. CARER: Contextualized Affect Representations for Emotion Recognition. Association for Computational Linguistics. 2018. URL: https://aclanthology.org/D18-1404 (date of access: 05.03.2024).

3. TMDb API. URL: https://developer.themoviedb.org/docs/getting-started (date of access: 06.03.2024).

Рисунок Г.5 – Копія статті

ДОДАТОК Д

Повідомлення про прийняття статті до публікації у конференції CoLiNs 2024 в якості CEUR-WS Style (NON-SCOPUS)



Colins Conference <colins.in.ua@gmail.com> Thu, 28 Mar, 14:56

to Наталія, Віра, me

Dear Author,

We would like to thank you for giving us an opportunity of studying and reviewing your research paper. This year we have received a great many interesting papers but we have to accept those which were estimated as best-ranked by 200 reviewers from around the world.

Nonetheless, we are pleased to inform you that your submission to 8th International Conference on Computational Linguistics and Intelligent Systems (CoLInS 2024) has been **ACCEPTED** for publication in **the special volume** as a full paper of CEUR-WS Style (**NON-SCOPUS**) because the results of your work deserve to be published.

The deadline for final paper submission is **12 April 2024.** If you'd like your paper to be included in the **SPECIAL volume** as a full paper of CEUR-WS Style (**NON-SCOPUS**), Please, enclose your updated doc-format article (MS Word format) and send it on the conference mail (colins.in.ua@gmail.com) with the subject – the publication number of your article in EasyChair.

The papers will be published provided that they strictly comply with the Paper Selection Requirements at:
https://colins.in.ua/paper_selection_requirements/
Before submitting the paper, please ensure its compliance with the Requirements.
Unfortunately, most of the authors did not comply with the main requirements of the publishing house CEUR-WS.org, namely: at least **one author of the submission must have at least 5 papers listed in DBLP (Do not include CEUR-WS papers in the DBLP)**. To check the number of your papers in DBLP, please, go to the DBLP site: https://dblp.uni-trier.de/pid/77/11070.html. **The author with the maximum number of papers in DBLP shall go first in the article.**

Рисунок Д.1 – Повідомлення про прийняття

ДОДАТОК Е

Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008: 2015

Експертний висновок результатів перевірки кваліфікаційної роботи

студент (посада) | програмної інженерії (кафедра) | ІПЗм-22-2 (група)

Середа Д.А.
(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
7.6.9	Якщо рядки або колонки таблиці виходять за межі формату сторінки, таблицю поділяють на частини, розміщуючи одну частину під іншою або поруч, чи переносять частину таблиці на наступну сторінку. У кожній частині таблиці повторюють її головку та боковик. У разі поділу таблиці на частини дозволено її головку чи боковик заміняти відповідно номерами колонок або рядків, нумеруючи їх арабськими цифрами в першій частині таблиці. Слово «Таблиця » подають лише один раз над першою частиною таблиці. Над іншими частинами таблиці з абзацного відступу друкують «Продовження таблиці » або «Кінець таблиці ____ » без повторення її назви.	42, далі за текстом.
	7.7 Переліки	
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
7.10.6	Пояснення познак, які входять до формули чи рівняння, треба подавати безпосередньо під формулою або рівнянням у тій послідовності, у якій їх наведено у формулі або рівнянні. Пояснення познак треба подавати без абзацного відступу з нового рядка, починаючи зі слова «де» без двокрапки. Познаки, яким встановлюють визначення чи пояснення, рекомендовано ви-рівнювати у вертикальному напрямку.	22
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні познаки	
	7.15 Додатки	

Експерт _____ (підпис) Вадим НЕЧВОЛОД (прізвище, ініціали)

09.06.2024