DOI: https://doi.org/10.26565/2304-6201-2024-62-03

УДК (UDC) 004,94

Єгоров Єгор Сергійович
магістр
Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022
e-mail: ees010301@gmail.com
https://orcid.org/0000-0001-8731-7198

Шматков Сергій Ігорович
д.т.н., професор; завідуючий кафедри теоретичної і прикладної системотехніки факультету комп'ютерних наук;
Харківський національний університет імені В. Н. Каразіна, майдан Свободи,4, Харків, Україна, 61022
e-mail: s.shmatkov@karazin.ua
https://orcid.org/0000-0002-0298-7174

Розробка моделі нейронної мережі для усунення неоднозначності омографів у текстових даних

У цій науковій статті досліджується створення моделі нейронної мережі, спрямованої на вирішення неоднозначності омографів у текстових даних. Різні типи нейронних мереж ретельно вивчаються на предмет їхнього потенціалу у вирішенні цієї проблеми. Стаття заглиблюється в методологічні аспекти архітектури нейронної мережі, що включає аналіз, проектування, реалізацію, тестування, оцінку та оптимізацію. Важливість кожного етапу цього процесу наголошується на важливості глибокого розуміння типів нейронних мереж та їх застосування, а також розумного вибору технології. Корисність розробленої моделі поширюється на домени, що використовують автоматичне розпізнавання мови, підтримку прийняття рішень на основі тексту, вдосконалення систем пошуку та обробку природної мови. Дослідники та практики в області обробки природної мови та класифікації текстових даних знайдуть цю статтю цінною.

Актуальність: у сучасному світі розробка моделі нейронної мережі для вирішення неоднозначності омографів у текстових даних визначається викликами, які стоїть перед галуззю обробки природної мови. Ця модель здатна виправляти помилки, пов'язані з невірним розумінням значень слів у тексті, що забезпечить більшу точність та якість аналізу текстового матеріалу. Використання її можливе в різних сферах, включаючи автоматичне визначення мови, підтримку прийняття рішень на основі текстової інформації, удосконалення систем пошуку та класифікації даних.
Мета: підвищення якості обробки тексту, зокрема, поліпшення точності розпізнавання слів, які мають більше одного значення, за допомогою розробки та впровадження нейронної мережі, яка буде мати можливість усувати неоднозначності омографів у текстових даних в реальному часі.

Методи дослідження: для дослідження обраної області було використано методи системний аналіз, методи глибокого навчання, теорія нейронних мереж, методи обробки та підготовки даних, імітаційне моделювання. Програмне забезпечення розроблено за допомогою мови Python і використані пакети sklearn, keras і т.д..

Результати: головним результатом роботи була розробка моделі нейронні мережі, яка усуває неоднозначності омографів у текстових даних в реальному часі, що дає можливість розвитку її на інших мовах.
Висновки: розглянуто проблему неоднозначності омографів у тектстових даних. Оскільки це завдання обробки природної мови була розроблена модель нейронної мережі з довгою короткостроковою пам'яттю з використанням ембедінгових моделей, LSTM-шару та повнозв'язаних шарів. Дослідження засвідчило важливість інноваційних підходів у вирішенні проблем неоднозначності омографів у текстових даних, а використання нейронних мереж та технологій штучного інтелекту стає обіцяючим напрямком для подальших досліджень та впроваджень у цій області.
Ключові слова: *нейронні мережі, омограф, модель, ембедінг, PYTHON, LSTM.*

Як цитувати: Єгоров Є. С., Шматков С. І. Розробка моделі нейронної мережі для усунення неоднозначності омографів у текстових даних. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2024. вип. 62. С.30-36. https://doi.org/10.26565/2304-6201-2024-62-03

How to quote: Yehorov Y., Shmatkov S "A neural network model for homograph disambiguation in textual data ", *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 62, pp. 30-36, 2024. [In Ukrainian] https://doi.org/10.26565/2304-6201-2024-62-03

1. Вступ

У сучасному інформаційному суспільстві, в якому величезні об'єми текстової інформації обробляються щоденно, розробка ефективних інструментів для автоматичної обробки та аналізу текстових даних стає надзвичайно важливою задачею. Однак існують різні лексичні неоднозначності, які ускладнюють цей процес, і однією з основних проблем є неоднозначність омографів — слів, що мають однакову лексичну форму, але різне значення.

2. Обґрунтування та вибір типу нейронної мережі з довгою короткостроковою пам'яттю (LSTM)

a) **Обробка послідовностей:** LSTM мережі ефективно працюють з послідовностями даних, такими як текст, часові ряди тощо. Їхній механізм довгої пам'яті дозволяє зберігати та використовувати інформацію з попередніх кроків послідовності, що робить їх ідеальними для прогнозування на основі історичних даних.

b) **Вирішення проблеми зникаючого градієнту:** LSTM мережі мають механізми воротного контролю, що дозволяють ефективно управляти градієнтами у процесі зворотного розповсюдження помилки. Це допомагає уникнути проблеми зникаючого градієнту та дозволяє тренувати глибокі мережі більш ефективно.

c) **Управління довгостроковою залежністю:** LSTM здатні ефективно вирішувати завдання, де потрібно управляти довгостроковими залежностями між даними. Це особливо корисно у задачах, де важливо враховувати контекст на більш довгій відстані в часі або послідовності.

d) **Гнучкість в архітектурі:** LSTM мережі можуть бути модифіковані та адаптовані для різних завдань, включаючи класифікацію, прогнозування та генерацію послідовностей. Вони можуть бути використані як частина більш складних архітектур або вбудовані в інші моделі, що робить їх універсальними для різноманітних застосувань (рис. 1). [1-2]

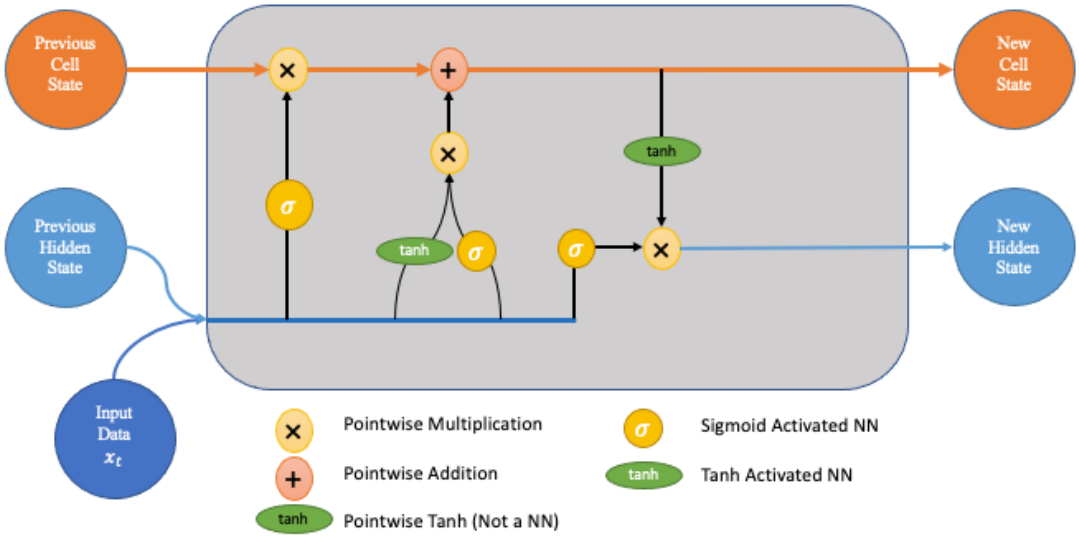

Рисунок 1 – Діаграма LSTM

3. Архітектура нейронної мережі (рис. 2)

- Input Layer (Вхідний шар): цей шар відображає вхідні дані в модель. Даним вхідним шаром є Embedding з параметрами, які визначають розмір словника (vocab_size), розмірність вектора ембедінгу (embedding_dim) і довжину вхідної послідовності (input_length).
- Embedding Layer (Шар ембедінгу): цей шар використовується для перетворення індексів слів у вектори ембедінгу. Вектор ембедінгу - це числовий вектор, який представляє слово у просторі високої розмірності. Величина input_length вказує на максимальну довжину вхідних послідовностей.
- LSTM Layer (Шар LSTM): шар довготривалої короткострокової пам'яті (LSTM) використовується для моделювання довготривалих залежностей в послідовностях.

Він приймає вхід від шару ембедінгу і генерує внутрішній стан, який передається наступним шарам.

- Dense Layer (Повнозв'язаний шар): цей шар використовується для класифікації або регресії. Для вирішення поставленої задачі використовується функція активації softmax для визначення ймовірностей кожного класу.[3]
- Output Layer (Вихідний шар): шар визначає вихідні ймовірності для різних класів у задачі класифікації.
- Зв'язки між шарами: стрілки між шарами представляють потік даних від одного шару до іншого. Вони також вказують напрямок передачі інформації. Враховуючи вищезазначені фактори, обрана архітектура LSTM забезпечує ефективневиявлення та усунення неоднозначності омографів у текстових даних, здатність адаптуватися до різних контекстів та уникнення проблем зниклих градієнтів під час тренування.[4-5]

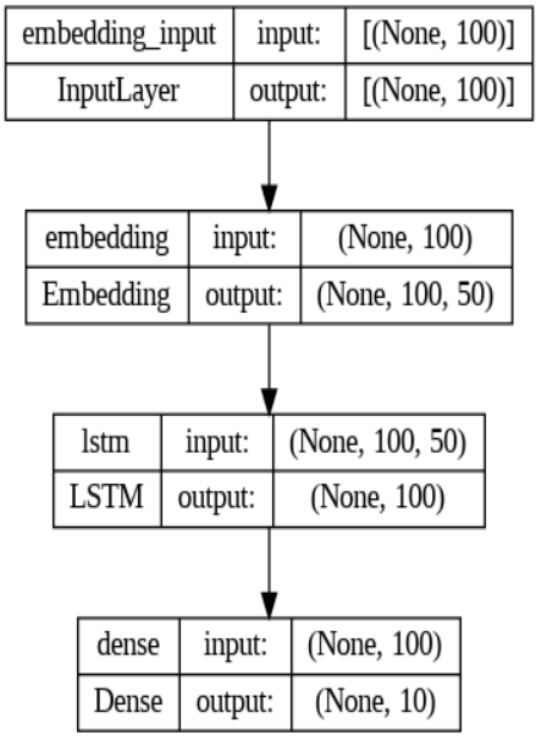

Рисунок 2 – Модель нейронної мережі для усунення неоднозначності омографів у текстових даних

4. Навчання нейронної мережі, та тестування

Кроки:

- Попередня обробка:

  Очищення даних: Видалення непотрібної або неінформативної інформації, такої як зайві символи, пробіли, знаки пунктуації. Це допомагає у покращенні якості даних та ефективності моделі.

  Токенізація: Розбиття тексту на окремі "токени" або слова. Це необхідно для подальшого представлення тексту у вигляді числових векторів, зрозумілих для моделі.

  Лематизація та стемінг: Зменшення слова до його базової форми (леми) або кореневої форми (стему). Це допомагає моделі у виборі ключових слов для розпізнавання семантики.[6]
- Векторизація тексту:

  Цей процес перетворює слова і токени в числові представлення, що використовуються для подальшого навчання моделі.
- Ініціалізація:

  Визначає початкові параметри моделі та її компонентів.
- Етап оптимізації:

  За допомогою оптимізатора Adam проходить процес адаптації ваг моделі під

конкретні дані.

- Етап форвардного та зворотнього поширення:

  Форвардне поширення: вхідні дані подаються в модель, де вони проходять через різні шари та отримують прогнозовані значення. Після цього відбувається обрахунок втрат та початок зворотного поширення.

  Зворотнє поширення: обчислення градієнтів втрат відносно параметрів моделі. Ці градієнти використовуються для актуалізації ваг шарів у відповідності із заданими правилами оптимізації. Цей процес дозволяє моделі "вчитися" з входов та виправляти помилки, підлаштовуючи параметри.

- Оцінка:

  Валідація та тестування: Оцінка моделі на валідаційному наборі для налаштування параметрів та на тестовому наборі для оцінки загальної ефективності.
  Метрики ефективності: Використання відповідних метрик для вимірювання точності,чутливості, специфічності та інших параметрів продуктивності[7].

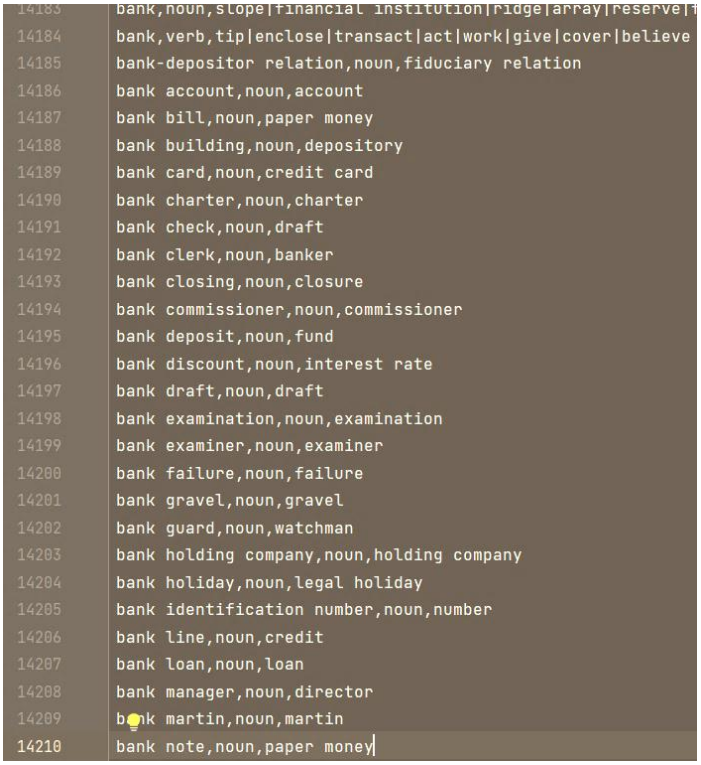

Рисунок 3 – Вигляд у якому зберігаються дані

Проведений експеримент з різними конфігураціями (табл. 1) моделі показав, що варіант (друга колонка) з параметрами LSTM = 64, Dense = 128, batch_size = 64, Epochs = 10, Adam = 0.001 демонструє найвищі значення обох метрик - високу точність (Accuracy = 0.8802) та значення AUC-ROC (0.8916). Це свідчить про те, що ця конфігурація моделі ефективно вирішує задачу класифікації омографів, надаючи найкращі результати у порівнянні з іншими варіантами.

Також важливо зазначити, що збільшення кількості LSTM-шарів не завжди призводить до поліпшення результатів, а зменшення кількості може вплинути на ефективність моделі.

Збільшення кількості епох не завжди призводить до покращення, а швидкість навчання може виявитися ключовим фактором: збільшення швидкості може призвести до зростання точності таAUC-ROC.

Таблиця 1 - Результати експериментів з різними параметрами моделі

	LSTM = 64 Demes = 128 Batch_size = 64 Epochs = 10 Adam = 0.001	LSTM = Demes = 128 Batch_size = 64 Epoch Adam = 0.001	LSTM = 32 Demes = 64 Batch_size = 64 Epochs = 10 Adam = 0.001	LSTM = 128 Demes = 256 Batch_size = 64 Epochs = 5 Adam = 0.01	LSTM = 64 Demes = 256 Batch_size = 64 Epochs = 10 Adam = 0.01	LSTM = 64 Demes = 128 Batch_size = 64 Epochs = 10 Adam = 0.0001
Accuracy						
AUC-ROC						

Навчання моделі пройшло досить успішно. Починаючи з високого значення точності на першій епосі, помітно певний зріст наступних епох, що свідчить про те, що модель ефективно вчиться на навчальних даних. Однак, можливо, варто звернути увагу на те, що на останніхепохах можна спостерігати певний спад у точності та інших метриках на валідаційному наборі. Це може бути ознакою перенавчання моделі або необхідності підгонки гіперпараметрів.

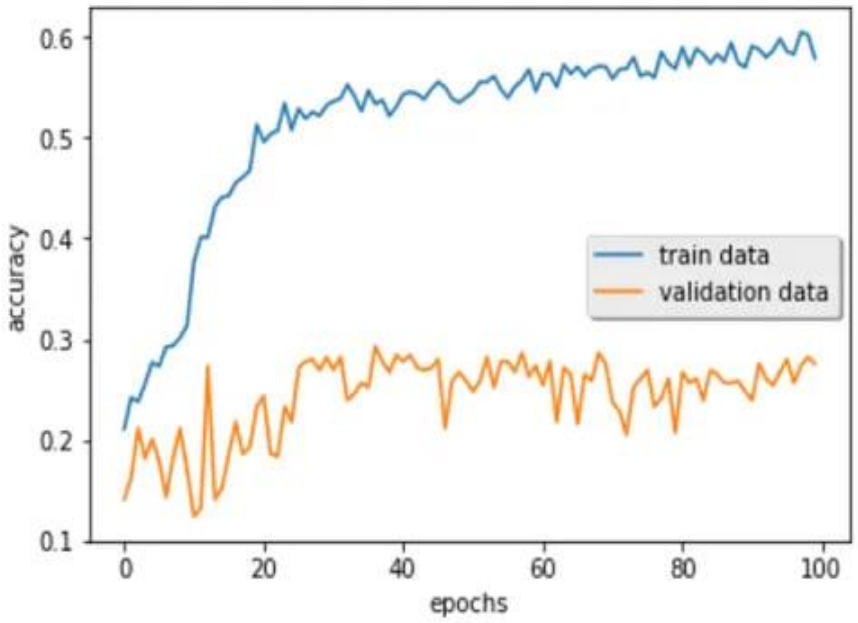

Рисунок 4 – Графік класової точночті моделі

Результати та висновки

Модель демонструє високий рівень ефективності та продуктивності завдяки розробленій архітектурі, різноманітності та якості даних для навчання, а також обраним гіперпараметрам. Архітектура включає LSTM-шари, ембедінги та дропаут-шари, які ефективно працюють з текстовими даними, що підтверджується високою точністю 89% та значенням AUC-ROC 0.8916. Навчальні дані складаються з 10 тисяч текстових даних з датасету, попередньо оброблених шляхом видалення HTML-сутностей, URL-адрес та інших непотрібних символів. Проведені експерименти та аналіз результатів дозволили визначити оптимальні гіперпараметри моделі.

Модульна структура та архітектура програмної реалізації також забезпечують легку інтеграцію та адаптацію моделі для використання на різних платформах. Наприклад, модель була успішно

впроваджена у середовищі Telegram, що дозволяє в реальному часі проводити аналіз і усувати неоднозначності омографів у текстових даних, які надходять через обрану платформу.

СПИСОК ЛІТЕРАТУРИ

1. Saiful Islam, Md.; Hossain, Emam (2020-10-26). "Foreign Exchange Currency Rate Prediction using a GRU-LSTM Hybrid Network":
   https://www.sciencedirect.com/science/article/pii/S2666222120300083?via%3Dihub (Last accessed: 21.05.2024).
2. Fully Connected Layers in Convolutional Neural Networks:
   https://indiantechwarrior.com/fully-connected-layers-in-convolutional-neural-networks/ (Last accessed: 21.05.2024).
3. Стрілець В. Є., Шматков С. І., Угрюмов М. Л. та ін. – Харків, Методи машинного навчання монографія, Харківський національний університет імені В. Н. Каразіна, 2020.
4. Zell, Andreas (1994). Simulation Neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.). Addison-Wesley. p. 73
5. McCrae, John P.; Labropoulou, Penny; Gracia, Jorge; Villegas, Marta; Rodríguez-Doncel, Víctor; Cimiano, Philipp (2015). "One Ontology to Bind Them All: The META-SHARE OWL Ontology for the Interoperability of Linguistic Datasets on the Web". In Gandon, Fabien; Guéret, Christophe; Villata, Serena; Breslin, John; Faron-Zucker, Catherine; Zimmermann, Antoine (eds.). The Semantic Web: ESWC 2015 Satellite Events. Lecture Notes in Computer Science. Vol. 9341. Cham: Springer International Publishing. pp. 271–282.
6. Kilgarriff,A.:Getting to know your corpus. In: Text, Speech and Dialogue, Springer (2012)3–15
7. "Deep Learning" Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016, MIT Press

REFERENCES

1. Saiful Islam, Md.; Hossain, Emam (2020-10-26). "Foreign Exchange Currency Rate Prediction using a GRU-LSTM Hybrid Network":
   https://www.sciencedirect.com/science/article/pii/S2666222120300083?via%3Dihub (Last accessed: 21.05.2024).
2. Fully Connected Layers in Convolutional Neural Networks:
   https://indiantechwarrior.com/fully-connected-layers-in-convolutional-neural-networks/ (Last accessed: 21.05.2024).
3. Strelets V. E., Shmatkov S. I., Ugryumov M. L. and others. – Kharkiv, Methods of machine learning monograph, V. N. Karazin Kharkiv National University, 2020. [in Ukrainian]
4. Zell, Andreas (1994). Simulation Neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.). Addison-Wesley. p. 73
5. McCrae, John P.; Labropoulou, Penny; Gracia, Jorge; Villegas, Marta; Rodríguez-Doncel, Víctor; Cimiano, Philipp (2015). "One Ontology to Bind Them All: The META-SHARE OWL Ontology for the Interoperability of Linguistic Datasets on the Web". In Gandon, Fabien; Guéret, Christophe; Villata, Serena; Breslin, John; Faron-Zucker, Catherine; Zimmermann, Antoine (eds.). The Semantic Web: ESWC 2015 Satellite Events. Lecture Notes in Computer Science. Vol. 9341. Cham: Springer International Publishing. pp. 271–282.
6. Kilgarriff,A.:Getting to know your corpus. In: Text, Speech and Dialogue, Springer (2012)3–15
7. "Deep Learning" Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016, MIT Press

Yehorov Yehor *Master student; V.N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv-22, Ukraine, 61022.*
e-mail: ees010301@gmail.com
https://orcid.org/0000-0001-8731-7198

Shmatkov Segii *Professor; V.N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv-22, Ukraine, 61022.*
e-mail: s.shmatkov@karazin.ua
https://orcid.org/0000-0002-0298-7174

Development of a neural network model to resolve homograph ambiguity in text data

This paper explores the creation of a neural network model aimed at resolving the ambiguity of homographs in textual data. Various neural network types are scrutinized for their potential in addressing this challenge. The paper delves into the methodological aspects of neural network architecture, encompassing analysis, design, implementation, testing, evaluation, and optimization. Each stage of this process is underlined for its significance, stressing the importance of a thorough understanding of neural network types and their applications, as well as judicious technology selection. The utility of the developed model extends to domains utilizing automatic language recognition, text-based decision support, enhancement of search systems, and natural language processing. Researchers and practitioners in the field of natural language processing and text data classification will find this paper valuable.

Relevance: in today's world, the development of a neural network model to resolve the ambiguity of homographs in textual data is determined by the challenges facing the field of natural language processing. This model is able to correct errors associated with incorrect understanding of the meanings of words in the text, which will ensure greater accuracy and quality of the analysis of the text material. Its use is possible in various areas, including automatic language detection, decision support based on text information, improvement of search systems and data classification.

The goal: to improve the quality of text processing, in particular, to improve the accuracy of recognition of words that have more than one meaning, through the development and implementation of a neural network that will be able to resolve homograph ambiguities in text data in real time.

Research methods: system analysis, deep learning methods, neural network theory, data processing and preparation methods, simulation modeling were used to study the selected area. The software is developed using the Python language and uses the sklearn, keras and other packages.

Results: the main result of the work is the development of a neural network model that eliminates homograph ambiguities in text data in real time, which makes it possible to expand it for the other languages.

Conclusions: the problem of ambiguity of homographs in textual data has been considered. For this natural language processing task, a neural network model with long short-term memory was developed using embedding models, LSTM layer, and fully connected layers. The study proves the importance of innovative approaches in solving homograph ambiguity problems in textual data, and that the use of neural networks and artificial intelligence technologies becomes a promising direction for further research and implementation in this area.

***Keywords**: neural networks, homograph, model, embedding, PYTHON, LSTM*

Надійшла у першій редакції 03.04.2024, в останній -17.05.2024.
The first version has been received 3.04.2024, the final version – on 17.05.2024.