

**КОМП'ЮТЕРНІ НАУКИ**

УДК 519.876.5

**ГЕНЕРУВАННЯ РИМОВАНИХ РЯДКІВ ЗА ДОПОМОГОЮ  
ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ****Ю. Калічун**

*Львівський національний університет імені Івана Франка,  
вул. Університетська, 1, Львів, 79000,  
e-mail: ykalichun@gmail.com*

Розглянуто та проаналізовано передові методи генерування природної мови за допомогою штучних нейронних мереж. Виявлено переваги і недоліки цих методів і на підставі цієї інформації спроектовано та натреновано штучний інтелект, здатний писати слова до пісень. В межах цього дослідження було зібрано тренувальний набір з 35 тисяч рядків з українських пісень. Було спроектовано модель штучного інтелекту з використанням GRU. Описано результат роботи цієї моделі і проаналізовано згенеровані тексти. Також було зазначено можливі напрями поліпшення генерованих слів до композицій і їх використання.

*Ключові слова:* генерування природної мови, рекурентні нейронні мережі, навчання без вчителя, GRU, LSTM, вебскрапінг, BeautifulSoup4.

**1. ВСТУП**

Останні підходи в нейронних мережах, зокрема прогресивне глибоке навчання, сильно вплинули на сферу штучного інтелекту (ШІ), регулярно вдосконалюючи рішення багатьох складних завдань у різних унікальних областях. Генерування природної мови (NLG) не є особливим випадком. У багатьох задачах NLG використання глибокого навчання дало результати, які ефективно перевершили результати інших ШІ та статистичних методів, які використовувались раніше.

Однією з актуальних задач NLG зараз є генерування римованого тексту, тобто віршів. Крім звичайних вимог, таких як логічна зв'язність, до тексту додається необхідність римування, що ускладнює процес генерування в рази.

**2. КОРОТКИЙ ОГЛЯД ВІДОМИХ ПРОЄКТІВ****2.1. GHOSTWRITER**

GhostWriter демонструє ефективність LSTM (довготривалої короткочасної пам'яті) у задачі створення реп тексту. Мета цієї моделі – створити текст, який за стилем подібний до певного артиста, але не схожий на існуючі тексти.

На відміну від інших робіт, які визначають явні шаблони для генерування тексту, ця модель визначає власну схему рими, довжину рядка та довжину вірша. Експерименти свідчать, що модель довготривалої короткочасної пам'яті створює тексти краще, ніж базові моделі.

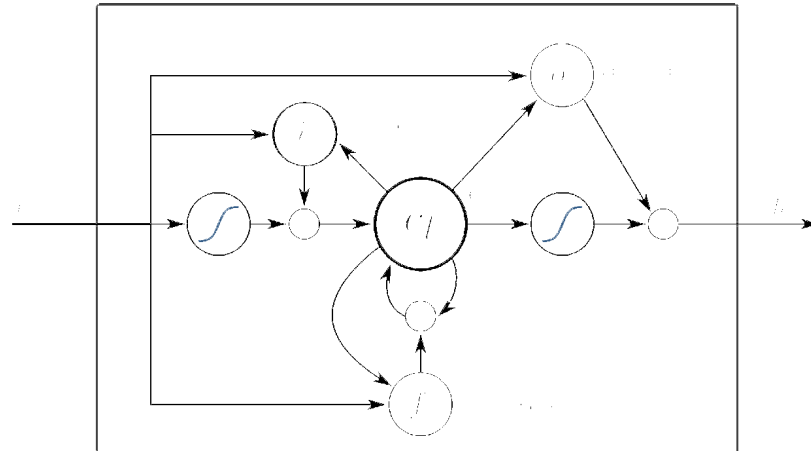


Рис. 1. Модель вузла LSTM

## 2.2. TEXT-BASED LSTM NETWORKS FOR AUTOMATIC MUSIC COMPOSITION BY KEUNWOO CHOI, GEORGE FAZEKAS, MARK SANDLER

Пропонований спосіб охоплює два LSTM шари та глибоке навчання з використанням фреймворку Keras. Запропонована мережа призначена для вивчення залежностей всередині текстових документів.

У експериментах word-RNN (Recurrent Neural Networks) дають хороші результати в обох випадках, тоді як RNN на підставі символів (char-RNN) вдається вивчити лише музичні акорди. Запропонована система може бути використана для повністю автоматичної композиції або як напівавтоматична система, яка допомагає людям складати музику, керуючи параметром різноманітності моделі.

## 2.3. MUSENET – 2019

Це модель генерування музики від OpenAI. Вона використовує найсучаснішу архітектуру NLP – масштабну модель трансформатора для прогнозування наступного токена в послідовності. Вона може поєднувати стилі різних відомих композиторів, а також різних музичних жанрів.

## 3. ПРОЄКТУВАННЯ

Проблема RNN полягає в тому, що в них справді важко закодувати далекоглядні залежності через архітектуру мережі. Кожен новий стан – це попередній стан, помножений на сукупність ваг. Це означає, що якщо вага навіть трохи менше одиниці, то на кожному кроці стан буде зменшуватися. Під час багаторазового множення на багатьох кроках це значення стану знизиться до 0. І навпаки, якщо вага навіть трохи більше одиниці, то дуже швидко це значення стану вибухне. Через це дуже важко "запам'ятати" стан на більш довгому діапазоні.

### 3.1. LSTM

Щоб вирішити цю проблему, були винайдені моделі довгострокової пам'яті (LSTM). Як і прості RNN, LSTM отримують вектор і повертають вектор. Однак, на відміну від RNN, стан, що передається в наступну ітерацію, відрізняється від результату на кожному кроці. Перш ніж вирішити, що вивести, а що передати як стан, LSTM виконують сукупність попередніх обчислень. Вони виконують чотири незалежних обчислення, використовуючи перцептрони на кожному кроці, і використовують результат цих обчислень, щоб визначити новий стан і вихідні значення. Обчислення, які використовуються: забуття (Forget gate), активація оновлення (Input gate), кандидат (Cell) і активація виводу (Output gate). Модель одного вузла LSTM разом з зазначеними обчисленнями зображена на рис. 1.

### 3.2. GRU

У 2014 році були винайдені рекурентні вузли з закритим доступом. Вони є спрощеною версією LSTM, тому можуть тренуватися швидше без втрати продуктивності. Є дві великі відмінності між GRU та LSTM. По-перше, вихідні значення дорівнюють стану, який GRU передає в наступний блок, як це зображено на рис. 2.

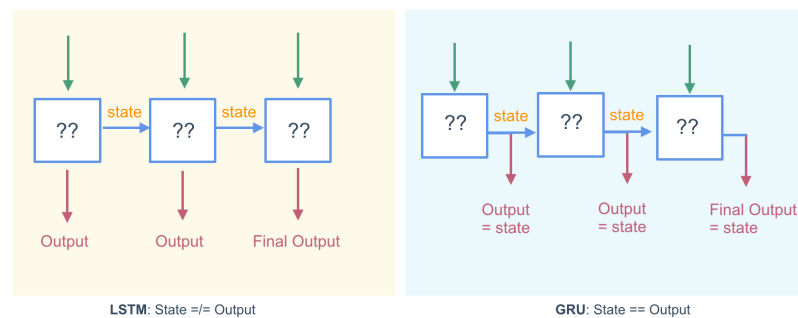


Рис. 2. Передача в наступний блок

По-друге, обчислюються лише три вектори, а не чотири. Якщо подивитись на LSTM, то можна помітити зайвий вектор, який керує рішенням, які дані зберегти, а які забути. Для оптимізації пам'яті і швидкодії GRU об'єднують вектор забуття разом з вектором оновлення активаційної функції. Порівняння можна побачити на рис. 3.

Саме через вищезгадані переваги за основу моделі було обрано GRU. Далі, шляхом спроб і помилок, було визначено оптимальну кількість вентильних рекурентних вузлів – 256. З такою кількістю було отримано хороші результати з порівняно коротким процесом навчання.

## 4. РЕАЛІЗАЦІЯ

### 4.1. ВИКОРИСТАНІ ТЕХНОЛОГІЇ, МЕТОДИКИ ТА ДАНІ

Для створення ШНМ було використано такі технології:

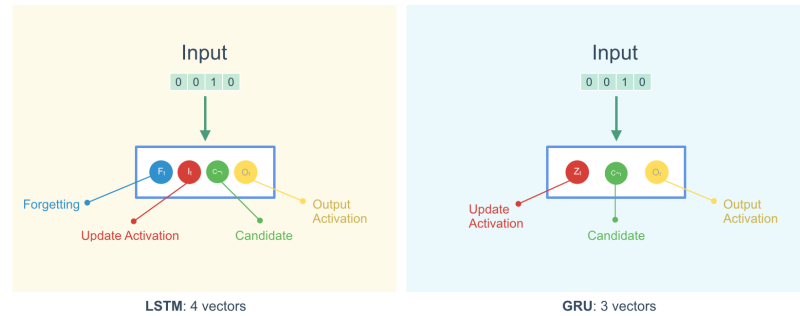


Рис. 3. Вектори забуття та оновлення активаційної функції

- Tensorflow v2.0 – фреймворк для створення та тренування моделей нейронних мереж
- Python 3.7
- Colaboratory – безплатне хмарне середовище з Jupyter notebook і відеокартами Nvidia Tesla
- BeautifulSoup4 – бібліотека для вебскрапінгу

#### 4.2. ВЕБСКРАПІНГ ТЕКСТІВ

Першим кроком у створенні будь-якої нейронної мережі є пошук датасету. В випадку українських пісень готового датасету не було, тому в рамках цієї роботи було написано програму на мові Python, яка збирає тексти українських пісень за допомогою бібліотеки BeautifulSoup4.

Джерелом текстів став веб ресурс Spivanyuk.org [5] Насамперед з сторінки "Всі пісні" було зібрано посилання на 7 тисяч сторінок з піснями. Після чого, за допомогою іншого алгоритму вебскрапінгу, було оброблено ці сторінки і зібрано 35 тисяч рядків з українських пісень. Отримавши текст, його потрібно було відфільтрувати, щоб зробити дані однорідними, видалити пусті рядки і зайвий текст, як от "Приспів:", "Співати 2 рази" і т.д. Для очищення датасету було використано звичайний текстовий редактор і регулярні вирази.

#### 4.3. СТВОРЕННЯ ТРЕНУВАЛЬНОГО ДАТАСЕТУ

Насамперед було створено словник з усіма словами з текстового файлу. Ці слова було пронумеровано і замінено слова на нові унікальні ідентифікатори (токени). Далі для створення датасету було порізано текстовий файл на окремі блоки по 21 слову і перші 20 слів було взято за вхідну послідовність, а останнє за очікуваний результат нейронної мережі.

#### 4.4. АРХІТЕКТУРА МОДЕЛІ

Архітектура нейронної мережі складається з Embedding layer, 1 шару Gated Recurrent Unit (GRU, вид рекурентної нейронної мережі) та Dense layer. Шар

відсіку(Dropout) у цій нейронній мережі не використовується, бо перетренування не є ймовірним. Детальніша архітектура та код для створення моделі зображено на рис. 4, 5.

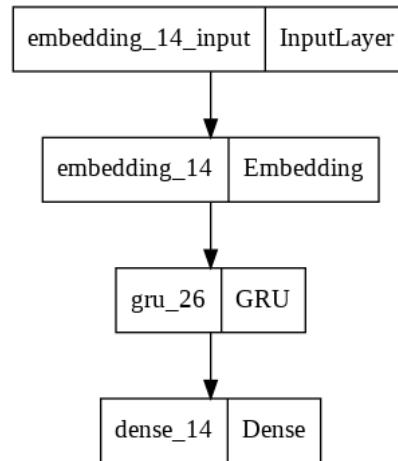


Рис. 4. Архітектура моделі

```

def createModel(vocab_size, embedding_dim, rnn_units, batch_size):
    model = tf.keras.Sequential([
        tf.keras.layers.Embedding(vocab_size, embedding_dim,
                                  batch_input_shape=[batch_size, None]),
        tf.keras.layers.GRU(rnn_units,
                             return_sequences=True,
                             stateful=True,
                             recurrent_initializer='glorot_uniform'),
        tf.keras.layers.Dense(vocab_size)
    ])
    return model
  
```

Рис. 5. Код моделі

#### 4.5. ТРЕНУВАННЯ МОДЕЛІ

Для оновлення ваг нейронної мережі використовується метод стохастичного градієнта Adam. Під час навчання за допомогою перехресної ентропії обчислюється помилка, а після цього застосовується метод зворотного поширення помилки, і оновлення ваг.

Тренування однієї епохи займає від 20 до 40 секунд залежно від того, наскільки великий набір даних обрано для тренування. Для досягнення цілей було взято 35 тис. рядків тексту з пісень і натреновано 20 епох. Загалом тренування зайняло близько 10 хвилин на відеокарті Nvidia Tesla K80.

## 5. РЕЗУЛЬТАТ

Результати роботи нейронної мережі можна побачити нижче. З низки текстів було обрано ті, які найліпше римуються і не містять співпадінь з вхідним датасетом.

можна бавити слова збираю кричать поганці  
наче диявола сіроманці  
а он із заходу -- папські солдати --  
іезуїти та уніати

так будь зі мною зі мною  
я мовчу та сумно на душі  
і не схоже на сни  
в небесах я і ти  
може це перемога --  
ну от і братія мовчить  
не вмієш думати ти  
не можеш звикнути до болю  
на божий світ з'являються вірші  
і дивний світ придуманий тобою

але на жаль наснилось все на світі  
ті вчені  
я не знаю чи живемо  
чи чекаем ми --  
зупинив час ти  
і в п'ятницю ввечері  
зустрітись приречені

Після аналізу результатів було суб'єктивно оцінено, що серед усього згенерованого тексту приблизно 30% становлять грамотно складені вірші - з римою і логічними словами. І з цих грамотних віршів приблизно 25% – не є плагіатом з датасету.

## 6. ВИСНОВОК

У межах цього дослідження було згенеровано тексти до пісень нейронною мережею. Наразі згенеровані вірші важко назвати шедевром поезії, але часто в них трапляються унікальні уривки з хорошим римуванням і логічним змістом. Позаяк для цієї роботи використовували випадкову вибірку пісень, тому важко вловити, чий саме стиль наслідує ця модель. Вона поєднує все, чому навчилася серед різних текстів різних виконавців. Для поліпшення результату треба збільшити датасет хоча б 50 разів – тоді буде більший набір римувань для одного початку вірша і як мінімум зменшиться відсоток плагіату.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Tensorflow 2.0 Essential Documentation [Електронний ресурс] // Tensorflow. – 2021. – Режим доступу: <https://www.tensorflow.org/guide>
2. Sherman Wezley Ghost Writing with TensorFlow [Електронний ресурс] / Wezley Sherman // Towards Data Science. – 2020. – Режим доступу: <https://towardsdatascience.com/ghost-writing-with-tensorflow-49e77e26978f>
3. Chadalavada V. Ghost Writing with TensorFlow [Електронний ресурс] / Venkat Chadalavada // Medium. – 2020. – Режим доступу: <https://medium.com/@venkat.chadalavada/lyrics-generator-using-lstm-e174bfc6c2f>
4. MuseNet [Електронний ресурс] // OpenAI. – 2019. – Режим доступу: <https://openai.com/blog/musenet/>
5. Spivanyuk.org [Електронний ресурс] // Spivanyuk.org. – 2019. – Режим доступу: <http://spivanyuk.org/>
6. Paialunga P. Song lyrics generation with Artificial Intelligence (RNN) [Електронний ресурс] / Piero Paialunga // Towards Data Science. – 2020. – Режим доступу: <https://towardsdatascience.com/song-lyrics-generation-with-artificial-intelligence-rnn-cdba26738530>
7. Sigurur Skúli How to Generate Music using a LSTM Neural Network in Keras [Електронний ресурс] / Skúli Sigurur // Towards Data Science. – 2017. – Режим доступу: <https://towardsdatascience.com/how-to-generate-music-using-a-lstm-neural-network-in-keras-68786834d4c5>
8. Keunwoo C. Text-based LSTM networks for Automatic Music Composition [Електронний ресурс] / Choi Keunwoo, George Fazekas, Mark Sandler // arxiv. – 2016. – Режим доступу: <https://arxiv.org/abs/1604.05358>
9. Text generation with an RNN [Електронний ресурс] // Tensorflow. – 2021. – Режим доступу: [https://www.tensorflow.org/text/tutorials/text\\_generation](https://www.tensorflow.org/text/tutorials/text_generation)
10. Калічун Ю. Генерування музики за допомогою штучних нейронних мереж / Ю. Калічун, Н. Колос // Вісн. Львів. ун-ту. Сер. прикл. матем. та інф. – 2021. – Вип. 29. – С. 115–122.

*Стаття: надійшла до редколегії 01.09.2022  
доопрацьована 15.10.2022  
прийнята до друку 05.10.2022*

## LYRICS GENERATION USING NEURAL NETWORKS

**Y. Kalichun**

*Ivan Franko National University of Lviv,  
Universytetska str., 1, Lviv, 79000, Ukraine,  
e-mail: [ykalichun@gmail.com](mailto:ykalichun@gmail.com)*

The paper presents a study on the use of artificial neural networks (ANNs) for natural language generation (NLG), with a focus on generating song lyrics. NLG is a subfield of artificial intelligence that deals with the automatic production of natural language text. It has a wide range of applications, including machine translation, dialogue systems, and content generation. ANNs are a type of machine learning model that are inspired by the structure and function of the human brain. They have been successfully applied to a variety of tasks, including image classification, speech recognition, and natural language processing. In this paper, the author reviews the existing literature on NLG and ANNs, discussing the benefits and limitations of different methods. He then presents a case study in which he designs and trains an AI model to generate lyrics using a dataset of 35,000 lines

---

from Ukrainian songs. The AI model is based on a type of ANN called a gated recurrent unit (GRU), which has been shown to be effective in tasks involving sequential data. The author reports on the results of their model, including how well it was able to generate coherent and diverse lyrics, and compares it to other approaches. He also considers the potential applications and implications of his work, such as the use of the generated lyrics in compositions and ways to improve the quality of the generated texts. Finally, he describes how the web scraping and the BeautifulSoup4 library were used to collect the dataset. In summary, this paper provides a detailed examination of the use of ANNs and unsupervised learning techniques for NLG, with a focus on the task of generating song lyrics.

*Key words:* natural language generation, recurrent neural networks, unsupervised learning, GRU, LSTM, web scraping, BeautifulSoup4.