ISSN (p) 0321-2211, ISSN (e) 2663-3450
Автоматизація та інтелектуалізація приладобудування

DOI: 10.20535/1970.69(1).2025.331890
УДК 004.021, 004.62, 004.855.5

РОЗРОБКА СИСТЕМИ ГЕНЕРАЦІЇ НА ОСНОВІ АЛГОРИТМУ ШТУЧНОГО ІНТЕЛЕКТУ. ЧАСТИНА 2

Вережинський В. А., Даниленко А. В., Рупіч С. С., Цибульник С. О.
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
E-mail: serhii.rupich@gmail.com

Під штучним інтелектом часто розуміють певний алгоритм або ансамбль моделей машинного навчання, що надає змогу вирішувати прикладні задачі різного характеру, найчастіше з присутньою характерною нелінійністю, з урахуванням специфіки області дослідження та можливості застосування нетривіальних підходів. Однією з таких задач є завдання генерації нових об'єктів чи поєднань. В даній роботі розглядається повний цикл розробки системи генерації, починаючи від збору даних, завершуючи готовою моделлю з елементами штучного інтелекту. Зазвичай моделі та алгоритми, що використовуються, потребують набір початкових даних, який забезпечує навчання та пошук певних закономірностей, та в подальшому може бути використаний для інших досліджень. Перша частина статті присвячена вирішенню задачі формування набору даних. Було розглянуто методи захисту інтернет-джерел, розроблено парсер для автоматичного збору даних з електронних джерел та ресурсів, виконано препроцесінг отриманої інформації та виконано композицію датасету комбінацій слів. Це найбільш трудомісткий та важливий етап у розробці системи генерації, адже всі подальші результати, їх якість та достовірність, повністю залежать від повноти, чистоти та правильності сформованих даних.

Друга частина присвячена наступним етапам розробки системи генерації, а саме візуалізації даних, налаштуванню параметрів математичної моделі та перевірці згенерованих результатів. На етапі формування датасету інформація представляється у табличному вигляді, тому важливим етапом є візуальний аналіз. В даній роботі для візуалізації використовуються вектори ембедингу, що отримані через просту нейронну мережу Word2Vec, оскільки дані представляються у текстовому вигляді. Ці вектори трансформуються у двовимірний простір за допомогою методу t-SNE. В якості моделі генерації на останньому етапі розробки системи застосовується алгоритм класичного машинного навчання Ланцюг Маркова. Такий підхід надає можливість створювати відносно просту, але ефективну систему, що повною мірою відповідає процесу моделювання для знаходження нових закономірностей та поєднань.

***Ключові слова**: генерація; аналіз; препроцесинг; математичний алгоритм; нейронна мережа; машинне навчання.*

Вступ

Сучасні системи штучного інтелекту (ШІ) та машинного навчання значною мірою покладаються на якість та репрезентативність даних, що використовуються для навчання моделей. Особливо актуальним є розроблення ефективних методів генерації даних, які дозволяють отримувати структуровані та інформативні набори для різних задач. Однак, процес створення таких систем супроводжується рядом викликів, зокрема, необхідністю аналізу складних текстових даних, налаштування параметрів моделей та перевірки достовірності результатів.

Розробка систем генерації з використанням штучного інтелекту потребує ретельного підготовчого етапу, де якість даних визначає успіх усієї системи. У першій частині роботи висвітлено ключові кроки створення навчальної множини: автоматизований збір даних (парсинг) та їх подальшу обробку [1]. Зокрема, розглянуто технічні аспекти роботи з інтернет-ресурсами, включаючи обхід захисту вебсайтів, а також реалізацію стабільного парсингу за допомогою Docker та Docker Compose.

На основі зібраних json-файлів проведено масштабну класифікацію та уніфікацію понад 3000 словосполучень, видалення зайвих даних і формування структурованих датасетів. Ці дії забезпечили підґрунтя для майбутнього навчання генеративної моделі, де чистота та узгодженість даних є критичними для отримання коректних результатів. Отримані навчальні набори стануть основою для наступних етапів роботи, пов'язаних із побудовою та оптимізацією математичної моделі системи.

Сучасні технології штучного інтелекту, зокрема системи генерації даних, відіграють ключову роль у вирішенні складних завдань аналітики, прогнозування та автоматизації. Процес їх розробки включає низку взаємопов'язаних етапів, кожен з яких вимагає ретельного підходу для досягнення оптимальних результатів. Якщо формування датасету є базовим кроком, то подальші етапи – візуа-

лізація, налаштування моделей та генерація результатів – визначають практичну єфективність системи, її здатність до інтерпретації та адаптації до реальних умов.

Візуалізація отриманих даних

Представлення даних у табличному вигляді зручно використовувати для аналізу та застосовувати у моделюванні, але не завжди забезпечується повнота розуміння про структуру та взаємозв'язки датасету. Тому після препроцесінгу необхідно зробити візуальне представлення отриманих даних. Щоб якнайкраще побачити приховані закономірності у датасеті, можна навчити на них нейронну мережу (НМ) типу word2vec, та взяти ваги з ембедингу (векторного представлення слова), де розмір ембедингу буде відповідати (довжина_датасету, 512), де 512 – розмір прихованого шару мережі. Таким чином, кожен рядок ембедингу буде відповідати одному з інгредієнтів.

Word2Vec – примітивна нейронна мережа, що складається з 1-3 прихованих шарів [2-4]. На відміну від інших архітектур НМ, головна ціль використання word2vec – не результати, отримані на виході із мережі, а стан нейронів (ваги) у прихованих шарів. Кожен рядок прихованого шару описує окреме слово з датасету, на якому тренувалася НМ. Таким чином, при достатній якості датасету для навчання, отримується вектор для кожного слова. Розмір цього вектору підбирається самостійно дослідником в процесі дослідження задачі, це є гіперпараметр. Найчастіше, він встановлюється в межах від 128 до 1024 чисельних значень. Головна відміна таких векторів в тому, що близькі слова за значенням будуть знаходитися ближче одне до одного у 128-1024-вимірному просторі. Для виміру дистанції між цими векторами використовуються найчастіше косинусні відстанні (cosine distance) або мангеттенська метрика (manhattan distance). Також є можливість маніпулювати такими векторами за допомогою найпростіших математичних операцій.

Для вирішення завдання векторного представлення слів у роботі було використано просунуту версію word2vec – fastText від Facebook [5]. Бібліотека fastText – це бібліотека з відкритим вихідним кодом, що надає можливість вивчати користувачу представлення тексту та текстові класифікатори.

На рис. 1 наведено загальну схему архітектури word2vec [2-4]. Можна побачити, що між шарами мережі є вагові коєфіцієнти, які встановлюють необхідні взаємозв'язки між ними. В даному випадку, вагові коєфіцієнти між вхідним та прихованим шарами відповідають саме за необхідний ембедінг.

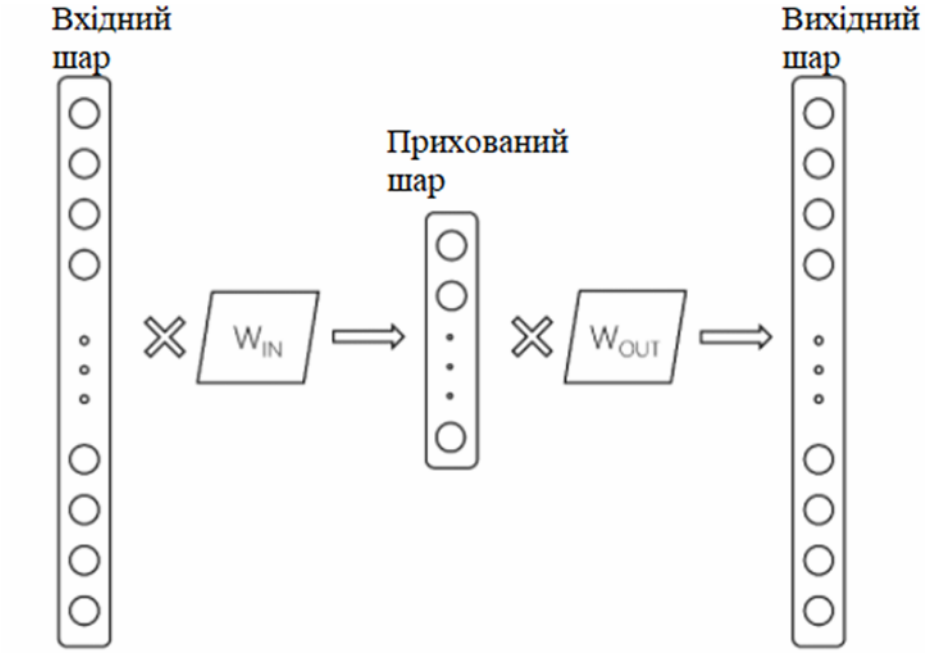

Рис. 1. Архітектура word2vec

Оскільки векторне представлення слова належить до багатовимірного простору, тому для візуалізації необхідно використовувати відповідні інструменти. Метод машинного навчання візуалізації даних t-SNE [6] використовується для нелінійного зменшення розмірності. Тобто, багатовимірні дані вкладаються у трьох або двовимірний простір з ціллю подальшої візуалізації.

На рис. 2 наведено вектори з ембедингу. Показано наскільки близько розташовані дані у двовимірному просторі при перенесенні з багатовимірного. Близькі за значеннями або спорідненні слова знаходяться в околі невеликих кластерів.

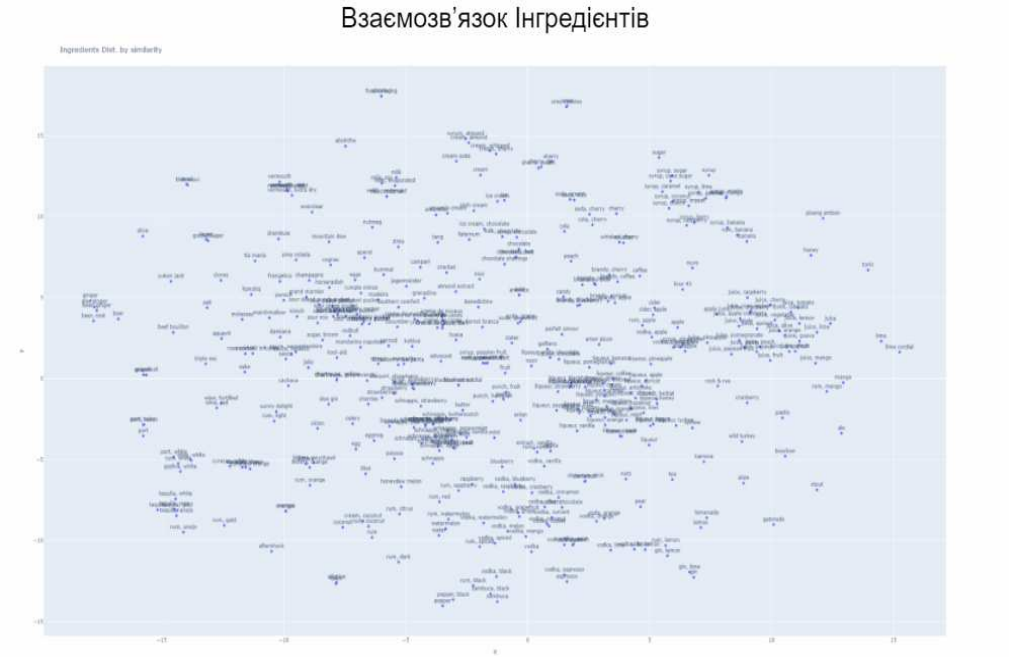

Рис. 2. Вектори з ембедингу fastText, трансформовані у двовимірний простір за допомогою t-SNE

Результати роботи алгоритму

На останньому етапі залишилося підібрати або розробити модель ШІ, що генерувала би якісні рецепти, при цьому використовуючи мінімальну кількість ресурсів для обрахунків. Для рішення задач натуральної мови існують такі мережі як LSTM, GRU, BERT, GPT-2, GPT-3, VAE та інші. Також наразі високу ефективність демонструють великі мовні моделі типу GPT, Gemini, DeepSeek тощо. Проте через те, що задача доволі специфічна, використання вже навчених мереж має низку обмежень, а для того, щоб навчити ті ж LSTM чи GRU, потрібно буде в 10-20 разів більше даних, ніж було отримано в процесі парсинку. Тому краще починати з найбільш простих моделей. Наприклад, ідеально підійдуть марковські ланцюги (Markov chains) для розв'язання даної задачі [7].

Марковські ланцюги є класичною ймовірнісною моделлю машинного навчання, яка знаходить застосування в задачах генерації текстів природної мови на основі заданого контексту. Основна перевага цього підходу полягає в його ресурсоефективності: алгоритм не вимагає значних обчислювальних потужностей під час інференсу (статистичного прогнозування), що робить його привабливим для систем із жорсткими обмеженнями на обсяг пам'яті або енергоспоживання (наприклад, у вбудованих пристроях або IoT-рішеннях). На відміну від НМ, які демонструють вищу точність за рахунок здатності враховувати складні контекстуальні залежності, але потребують значних ресурсів для файнтюнінгу та інференсу (зокрема, GPU-прискорення), марковські ланцюги базуються на простіших статистичних закономірностях.

Генерація тексту в марковських ланцюгах реалізується через переходи між станами, де кожен стан відповідає певному слову або послідовності слів. Простір станів формується на етапі навчання моделі на основі аналізу частот переходів між елементами тексту в тренувальному датасеті. Наприклад, у n-грамовій моделі ймовірність появи кожного наступного слова визначається попередніми n-1 словами, що обмежує контекстну чутливість, але зберігає низьку складність обчислень. Це робить марковські ланцюги особливо корисними для генерації коротких, структурованих текстів. На рис. 3 наведений принцип дії Ланцюга Маркова [8-9].

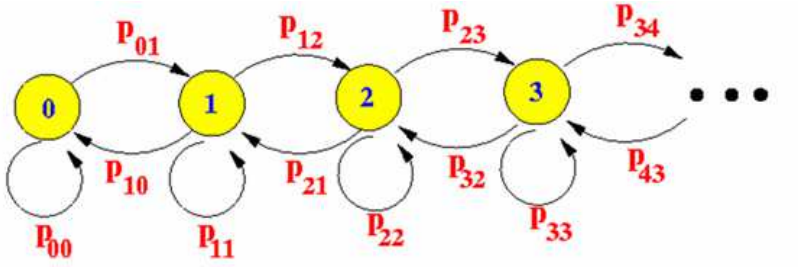

Рис. 3. Ланцюг Маркова

Однак обмеженість контекстного "горизонту" марковських моделей призводить до неприродньо довших текстів через ігнорування довгострокових залежностей.

Тому в сучасних системах, орієнтованих на якісну генерацію (наприклад, трансформери або RNN), вони застосовуються переважно як допоміжний інструмент або на початкових етапах прототипування.

Реалізація та результат генерації

Після навчання моделі Markov chains, при застосуванні алгоритму отримуємо випадково згенеровані словосполучення. На рис. 4 наведено частину реалізації.

В результаті отримуються готові логічні комбінації слів. На рис. 5 показаний приклад набору випадково згенерованих словосполучень.

```
import markovify
from utils import DIR

DATA_DIR = DIR.DATA_DIR


with DATA_DIR.joinpath('common/sentences.txt').open('r') as file:
    sentences = file.read()


def sentence_transform_out(sentence: str) -> str:
    return sentence.replace('COMMA', ', ').replace('SPACE', ' ')

sentence_transform_out(sentences[:50].replace(',',';'))

model = markovify.Text(sentences)
model.compile(inplace=True)


recipes = model.make_sentence()

recipes_cl = [sentence_transform_out(sentence.replace(" ", "; ").replace(",",
"""").strip()) for sentence in recipes.split('.')]

import random

for i in range(10):
    print(random.choice(recipes_cl)[2:])
```

Рис. 4. Реалізація моделі генерації

```
water; oil; liquid; fresh water; organic gaseous mixture; liquid compound;
hybrid liquid; syrup; black water; organic oil; soda; blood;
ink; paint; perfume; soap; chemical solution; biochemical compound;
gas; glue; mercury; gasoline; chemical compound; DNA synthesis;
composite substance; chemical pair; volatile mixture; formula mix; molten metal; liquid soap;
melted butter; olive oil; coconut oil; liquid sugar; salty water; fresh water;
salt water; fizzy drink; liquid medicine; liquid oxygen; liquid nitrogen; liquid chocolate;
caramel sauce; liquid wax; reagent mixture; polymer combination; inorganic substance; liquid detergent;
liquid fertilizer; protein compound;
Total words generated: 50
```

Рис. 5. Випадково згенеровані словосполучення

Висновки

У дослідженні представлено комплексний підхід до розробки системи генерації рецептів на основі штучного інтелекту, з акцентом на критичну роль якості даних та обґрунтований вибір алгоритмів. На етапі підготовки даних було реалізовано автоматизований збір інформації з використанням Docker та Docker Compose, що дозволило ефективно подолати технічні обмеження вебресурсів і сформувати структурований датасет з понад 3 тисяч комбінацій слів. Очищення та уніфікація даних забезпечили їх придатність для подальшого моделювання, підкреслюючи необхідність ретельного препроцесингу в задачах генерації [1].

Візуалізація за допомогою векторних ембедингів (fastText) та методу t-SNE продемонструвала можливість виявлення семантичних зв'язків між інгредієнтами, що є ключовим для інтерпретації структури датасету. Незважаючи на доступність складних архітектур (LSTM, GPT тощо), обмеженість обсягу даних та ресурсів обумовила вибір марковських ланцюгів як оптимального рішення. Ця модель, хоч і має обмежений контекстний "горизонт", довела свою ефективність у генерації коротких структурованих текстів за мінімальних витрат на інференс.

Переваги запропонованої системи полягають у поєднанні єфективності та мінімізації обчислювальних витрат. Проте подальше вдосконалення можливе за рахунок інтеграції більш складних архітектур, які враховують довгострокові залежності та підвищують якість генерації.

Дослідження також вказує на важливість балансу між складністю моделі, обсягом даних та практичними вимогами до системи, як основи для будь-якої системи ШІ, що генерує контент. Отримані результати можуть бути адаптовані для інших задач, пов'язаних з аналізом та синтезом структурованих текстів.

Література

[1] В. А. Вережинський, А. В. Даниленко, С. С. Рупіч, С. О. Цибульник, "Розробка системи генерації на основі алгоритму штучного інтелекту. Частина 1", *Вісник КПІ. Серія Приладобудування,* вип. 65(1), с. 110-116, 2023. DOI: 10.20535/1970.65(1).2023.283455

[2] word2vec [Online]. Available: https://code.google.com/archive/p/word2vec/

[3] T. Mikolov, K. Chen, G. Corrado, J. Dean, "Efficient Estimation of Word Representations in Vector Space," In *Proceedings of Workshop at*

ICLR. 2013.
[4] T. Mikolov, W.-T. Yih, G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," in *Proceedings of NAACL HLT*. 2013.
[5] fastText – Library for efficient text classification and representation learning [Online]. Available: https://fasttext.cc/
[6] Van der Maaten L., Hinton G., "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, № 11, 2008.
[7] W. K. Ching, M. K. Ng, "Markov chains", *Models, algorithms and applications*, vol. 650. pp. 111-139, 2006.
[8] Kulasiri D. et al., "Visualizing Markov Process Through Graphs and Trees," Chemical Master Equation for Large Biological Networks: State-space Expansion Methods Using AI., pp. 55-80, 2021.
[9] Dr. Salim El Rouayheb, Chapter 9: Markov Chains. ECE541: Stochastic Signals and Systems, 2018. [Online]. Available: https://eceweb1.rutgers.edu/~csi/ECE541/Chapter9.pdf

UDC 004.021, 004.62, 004.855.5

V. Verezhynksyi, A. Danylenko, S. Rupich, S. Tsybulnyk
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

DESIGN OF GENERATION SYSTEM BASED ON ARTIFICIAL INTELLIGENCE ALGORITHM. PART 2

Artificial intelligence is often understood as a specific algorithm or an ensemble of machine learning models that enable solving applied problems of various kinds, most often characterized by inherent nonlinearity, while considering the specifics of the research domain and the possibility of applying non-trivial approaches. One such task is the generation of new objects or combinations. This study examines the full development cycle of a generation system, from data collection to the final model incorporating artificial intelligence elements. Typically, the models and algorithms used require an initial dataset that facilitates training and pattern discovery and can later be applied to other research areas.
The first part of the article is dedicated to solving the problem of dataset formation. Methods for protecting online sources were explored, a parser for automatic data collection from electronic sources and resources was developed, preprocessing of the obtained information was performed, and a dataset of word combinations was compiled. This is the most labor-intensive and crucial stage in the development of the generation system, as all subsequent results – their quality and reliability – depend entirely on the completeness, cleanliness, and correctness of the structured data.
The second part focuses on the next stages of system development, specifically data visualization, tuning of mathematical model parameters, and validation of the generated results. During dataset formation, the information is represented in a tabular format, making visual analysis an essential step. In this study, embedding vectors obtained through a simple Word2Vec neural network are used for visualization, as the data is presented in textual form. These vectors are transformed into a two-dimensional space using the t-SNE method. At the final stage of system development, a classical machine learning algorithm the Markov Chain is employed as the generation model. This approach allows for the creation of a relatively simple yet effective system that fully aligns with the modeling process for discovering new patterns and combinations.

Keywords: generation; analysis; preprocessing; mathematical algorithm; neural network; machine learning.

Надійшла до редакції
23 березня 2025 року

Рецензовано
19 квітня 2025 року