

УДК: 004.5:004.8

*Оврамець І. В., здобувач вищої освіти,  
Антонов Ю. С., канд. фіз.-мат. наук, доцент,  
доцент кафедри інформаційних технологій,  
Донецький національний університет імені Василя Стуса*

## **ОСОБЛИВОСТІ АРХІТЕКТУРНОЇ РЕАЛІЗАЦІЇ АВТОМАТИЗОВАНОЇ СИСТЕМИ ГЕНЕРАЦІЇ ВІДПОВІДЕЙ НА ПОВІДОМЛЕННЯ**

*Анотація. У роботі розглянуто архітектурні особливості побудови автоматизованої системи генерації відповідей на повідомлення з використанням технологій обробки природної мови та моделей штучного інтелекту. Запропоновано підхід до формування релевантних відповідей на основі аналізу історії переписки.*

*Ключові слова: штучний інтелект, генерація відповідей, обробка природної мови, автоматизація, системи комунікації.*

**Вступ.** У сучасному світі цифрових технологій та глобальної комунікації обсяг інформації, що передається через месенджери, електронну пошту та соціальні мережі, неупинно зростає [1]. Щодня користувачі обмінюються мільярдами повідомлень, значна частина яких потребує швидких, змістовних та релевантних відповідей. У таких умовах автоматизація процесу спілкування стає не лише актуальним, а й необхідним напрямом розвитку інформаційних систем.

Використання технологій штучного інтелекту для генерації відповідей на повідомлення відкриває нові можливості для підвищення ефективності комунікації. Такі системи здатні аналізувати контекст переписки, визначати наміри користувача та формувати змістовні відповіді, наближуючи взаємодію людини з комп'ютером до природного діалогу.

Метою роботи є дослідження архітектурних особливостей побудови автоматизованої системи генерації відповідей на повідомлення з використанням методів обробки природної мови та технологій штучного інтелекту.

**Основний текст.** Процес обробки природної мови включає кілька послідовних етапів. Спочатку здійснюється токенізація, тобто розбиття тексту на окремі слова або фрази. Далі виконується лематизація або стемінг, які дають змогу звести слова до їх базової форми. Наступним кроком є синтаксичний аналіз, що полягає у виявленні граматичних зв'язків між словами, після чого проводиться семантичний аналіз, який допомагає системі зрозуміти зміст повідомлення. Завершальним етапом є визначення наміру користувача (intent detection) та генерація відповіді (response generation), де система формує релевантну та логічно пов'язану відповідь.

Для реалізації зазначених етапів використовуються різні інструменти та бібліотеки. Найпоширенішими є NLTK (Natural Language Toolkit) – набір інструментів для базового аналізу тексту, spaCy – продуктивна бібліотека для лінгвістичної обробки, Stanford NLP – потужний інструментарій для синтаксичного розбору, а також сучасні бібліотеки на базі трансформерних моделей, як-от BERT, RoBERTa, GPT, та фреймворки Hugging Face Transformers і LangChain [1]. Вони забезпечують глибоке контекстне розуміння тексту та використовуються для генерації осмислених відповідей у системах діалогів.

З метою забезпечення ефективної роботи всіх етапів обробки природної мови система має модульну архітектуру, у якій кожен компонент виконує окрему функцію та може бути незалежно оновлений або масштабований. Такий підхід підвищує гнучкість системи та спрощує її підтримку в процесі експлуатації.

Особливу роль відіграє модуль генерації відповідей, який інтегрує результати аналізу тексту та на основі визначеного наміру користувача формує осмислену відповідь. У межах запропонованої архітектури цей модуль реалізовано як окремий Docker-контейнер, що забезпечує ізоляцію середовища виконання та зручність розгортання [2]. Завдяки контейнеризації система може бути легко інтегрована з різними платформами, як-от Telegram, Slack, Discord, Microsoft Teams або вебчати корпоративних сайтів.



Рисунок 1.2 – Схема окремого модуля рекомендацій

Використання Docker дає змогу швидко розгорнути й масштабувати систему без необхідності зміни базової інфраструктури. Кожен контейнер може бути розміщений на окремому сервері або у хмарному середовищі, що підвищує стабільність роботи та забезпечує безперервну взаємодію між компонентами через REST API [3]. Такий підхід сприяє створенню розподіленої системи, здатної обробляти велику кількість запитів у реальному часі.

Додатково архітектура передбачає можливість використання баз даних для збереження історії переписки, що дає змогу системі навчатися на попередніх діалогах і формувати більш персоналізовані відповіді. Отже, поєднання методів обробки природної мови, трансформерних моделей і контейнеризації створює ефек-

тивну, масштабовану та адаптивну систему, здатну до інтеграції в різні середовища сучасних цифрових комунікацій.

**Висновки.** Внаслідок проведеного дослідження було розглянуто архітектурні особливості побудови автоматизованої системи генерації відповідей на повідомлення. Визначено основні етапи обробки природної мови – від токенизації та лематизації до семантичного аналізу, визначення наміру користувача та формування відповіді.

Проаналізовано сучасні інструменти та бібліотеки для реалізації зазначених процесів, зокрема NLTK, spaCy, Stanford NLP, Hugging Face Transformers і LangChain, що забезпечують глибоке контекстне розуміння текстів і можливість створення змістовних діалогових систем.

Окрему увагу приділено архітектурному підходу до контейнеризації системи. Запропоновано реалізувати модуль генерації відповідей у вигляді окремого Docker-контейнера, що забезпечує ізоляцію, масштабованість і простоту інтеграції з різними платформами, як-от Telegram, Slack, Discord, Microsoft Teams чи вебчати.

Розроблена концепція поєднує методи обробки природної мови, технології штучного інтелекту та контейнеризацію, створюючи основу для гнучкої, продуктивної й масштабованої системи автоматичної генерації відповідей. Подальші дослідження можуть бути спрямовані на вдосконалення контекстного розуміння, адаптацію моделей під конкретного користувача та впровадження механізмів самонавчання системи на основі історії діалогів.

#### Список використаних джерел

1. Timsina P. Building Transformer Models with PyTorch 2.0: NLP, computer vision, and speech processing with PyTorch and Hugging Face (English Edition). 2024. 247 p.
2. Altinok D. Mastering spaCy: An end-to-end practical guide to implementing NLP applications using the Python ecosystem. 2021. 372 p.
3. Gough J., Bryant D. Mastering API Architecture: Design, Operate, and Evolve API-Based Systems. 2022. 286 p.
4. The Effect of WhatsApp Usage on Employee Innovative Performance at the Workplace: Perspective from the Stressor–Strain–Outcome Model / N. M. Kasim, M. A. Fauzi, M. F. Yusuf, W. Wider. Behavioral sciences (Basel, Switzerland). 2022. Vol. 12(11). P. 456. DOI: 10.3390/bs12110456. URL: <https://pubmed.ncbi.nlm.nih.gov/36421752/> (дата звернення: 07.09.2025).