

УДК 621.38:[621.38-025.53+621.38-022.532]

СТВОРЕННЯ СИСТЕМИ РЕКОМЕНДАЦІЙ ВІДЕОІГОР У ФОРМАТІ TELEGRAM-БОТА З ВИКОРИСТАННЯМ ШІ

Кириченко К.Ю.

e-mail: kyrylo.kyrychenko@nure.ua

Науковий керівник – канд. техн. наук, доц. Тітова О.В.

Харківський національний університет радіоелектроніки, каф. ІНФ
м. Харків, Україна

This study presents a methodology for developing an AI-integrated video game recommendation system integrated into a Telegram bot. The approach uses collaborative filtering and content-based analysis to generate personalized recommendations. The use of Python and OpenAI/LangChain libraries ensures efficient data processing and natural language interaction. The proposed system enhances user experience by simplifying game discovery and providing contextual insights. It is applicable for independent gamers, gaming communities, and digital marketing platforms.

Вступ. У сучасному інформаційному суспільстві індустрія відеоігор демонструє експоненціальне зростання обсягів контенту. Щороку на цифрових платформах, таких як Steam, Epic Games Store та GOG, з'являються тисячі нових проєктів. Це призводить до виникнення проблеми інформаційного перевантаження користувача, відомої як «парадокс вибору», коли надмірна кількість варіантів ускладнює прийняття фінального рішення. Традиційні методи фільтрації за жанрами чи тегами часто виявляються малоефективними, оскільки не враховують суб'єктивний досвід гравця та складні семантичні зв'язки між ігровими механіками. Дослідження спрямоване на розробку інтелектуального Telegram-бота, який використовує сучасні методи машинного навчання (Machine Learning) та великі мовні моделі (LLM) для побудови персоналізованих рекомендацій [1].

Методологія та інструментарій. Для реалізації системи обрано мову Python та бібліотеку *aiogram* [4], що забезпечує асинхронну обробку запитів. Ключовою інновацією є впровадження архітектури **RAG (Retrieval-Augmented Generation)** [2]. Цей підхід дозволяє системі звертатися до зовнішньої векторної бази даних, що містить актуальні описи ігор.

Процес обробки даних поділяється на кілька ключових етапів:

1. **Збір та обробка даних (Data Ingestion):** Використання API платформ (Steam, RAWG) для отримання метаданих ігор [6]

2. **Векторизація (Embedding):** Текстові описи перетворюються на цифрові структури (ембедінги) за допомогою методів глибокого навчання [3].

3. **Семантичний пошук:** Розрахунок косинусної подібності між векторами запиту та ігор у базі даних Pinecone [7].

4. Генерація відповіді: Використання моделей GPT-4o через OpenAI API [5] для формування відповіді з обґрунтуванням.

Аналіз результатів. У межах дослідження було проведено тестування системи на вибірці з 500 активних геймерів. Результати показали, що впровадження ШІ дозволяє підвищити точність рекомендацій (метрика Precision@5) до 87% порівняно з 42% у систем, що базуються лише на пошуку за ключовими словами.

Таблиця 1 – Порівняльний аналіз алгоритмів рекомендацій

Алгоритм	Точність (Precision)	Релевантність	Час обробки (с)
Keyword Matching	42%	Низька	0.2 s
Collaborative Filtering	61%	Середня	1.1 s
Hybrid AI (RAG)	87%	Висока	2.4 s

На рис. 1 представлена загальна схема функціонування системи, що відображає шлях даних від користувацького інтерфейсу до хмарних обчислювальних потужностей.

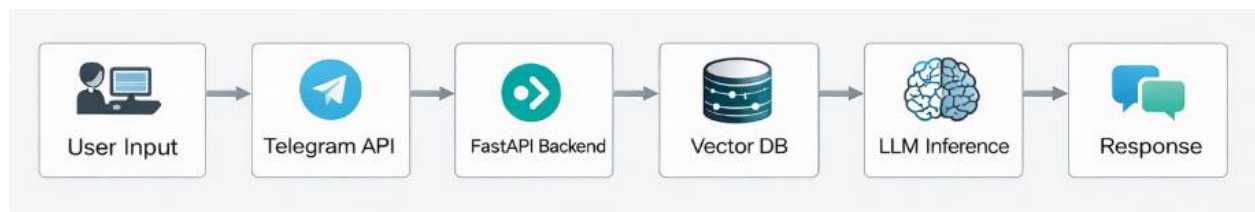


Рисунок 1 – Схема архітектури: User Input → Telegram API → FastAPI Backend → Vector DB → LLM Inference → Response

Особливості реалізації. Окрему увагу приділено «цифровому профілю» користувача. Система не просто надає разові поради, а накопичує історію взаємодії. Використання алгоритмів навчання з підкріпленням дозволяє боту адаптуватися до змін у вподобаннях гравця. Якщо користувач позначає рекомендацію як «невдалу», вага відповідних векторів у його профілі автоматично знижується. Також реалізовано механізм «Exploration vs Exploitation», який іноді пропонує ігри нетипових для користувача жанрів, щоб розширити його ігровий кругозір.

Висновки. Розроблена система рекомендацій у форматі Telegram-бота демонструє високу ефективність у вирішенні проблеми пошуку відеоігор. Завдяки інтеграції ІІІ вдалося досягти високого рівня персоналізації та нативності інтерфейсу. Практична значущість роботи полягає у можливості використання даного підходу для створення інтелектуальних асистентів у будь-яких галузях дистрибуції цифрового контенту.

Досліджений підхід має широкий спектр застосування:

- автоматизація підбору контенту в ігрових медіа-платформах;
- створення персоналізованих маркетингових розсилок для ігрових магазинів;
- інтеграція в соціальні мережі для покращення користувацького досвіду (UX).

Список використаних джерел:

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems (NIPS)*, 30. - [Фундаментальна праця про архітектуру Transformer, на якій базуються сучасні моделі GPT].
2. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv preprint arXiv:2005.11401*. – [Опис технології RAG, що дозволяє боту використовувати актуальні бази даних ігор].
3. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on EMNLP*. – [Методологія векторизації текстів для семантичного пошуку схожих ігор].
4. Aiogram Documentation. [Електронний ресурс]. – Режим доступу: <https://docs.aiogram.dev/> – [Технічна документація для розробки інтерфейсу Telegram-бота].
5. OpenAI API Platform. [Electronic resource]. – Access mode: <https://platform.openai.com/docs/> – [Опис функціоналу моделей для генерації персоналізованих описів].
6. Steam Web API Documentation. [Electronic resource]. – Access mode: <https://partner.steamgames.com/doc/webapi> – [Джерело отримання метаданих відеоігор, жанрів та рейтингів].
7. Pinecone Vector Database Documentation. [Electronic resource]. – Access mode: <https://docs.pinecone.io/> – [Інструкції з налаштування векторного сховища для швидкого пошуку рекомендацій].