

УДК (UDC) 004.93

**Новіков
Олексій Едуардович**

студент ННІ комп'ютерних наук та штучного інтелекту,
Харківський національний університет імені В. Н. Каразіна, майдан
Свободи, 4, м. Харків, 61022
e-mail: novikov2020ki11@student.karazin.ua;
<https://orcid.org/0009-0003-3566-531X>

**Стрілець
Вікторія Євгенівна**

к.т.н., доцент кафедри комп'ютерних систем та робототехніки,
Харківський національний університет імені В. Н. Каразіна, майдан
Свободи, 4, м. Харків, 61022
e-mail: viktoria.strilets@karazin.ua;
<https://orcid.org/0000-0002-2475-1496>

Модель чат-бота для конфігурування персонального комп'ютера із застосуванням методів NLP

Мета роботи: підвищення зручності та ефективності вибору компонентів персонального комп'ютера шляхом використання Telegram чат-бота з методами NLP для врахування запитів користувача.

Методи дослідження: методи обробки природної мови NLP для інтерпретації користувачьких запитів та формування відповідей чат-бота; методи побудови діалогових систем; підходи до організації компонентів програмного забезпечення. Telegram чат-бот реалізовано на основі клієнт-серверної архітектури, де клієнтська частина забезпечує взаємодію з користувачем у Telegram, а серверна — логіку обробки даних і підбору компонентів ПК. Для реалізації використані технології: мова програмування Python, бібліотека python-telegram-bot для створення чат-бота, інструменти NLP для аналізу та інтерпретації запитів користувача та fuzzy matching для покращення пошуку.

У **результаті** створено модель Telegram чат-бота, який автоматизує процес підбору комплектуючих для персональних комп'ютерів, враховуючи індивідуальні потреби та побажання користувача. Чат-бот дозволяє швидко отримати рекомендації щодо вибору компонентів ПК, таких як процесор, відеокарта, оперативна пам'ять, накопичувач, материнська плата та блок живлення, з урахуванням цінової категорії, призначення (ігри, робота, мультимедіа) та бажаних характеристик. Чат-бот забезпечує зручну взаємодію через Telegram, а серверна частина відповідає за обробку запитів, аналіз тексту користувача та формування оптимальних конфігурацій з використанням методів NLP і fuzzy matching. Для обробки природної мови застосовані бібліотеки та інструменти: Stanza, NLTK (токенізація, стемінг, лематизація), TextBlob; для нечіткого пошуку – RapidFuzz. Використання мови Python та бібліотеки python-telegram-bot забезпечує надійну роботу системи, гнучкість у масштабуванні та можливість швидкого оновлення бази компонентів.

Висновки: створений Telegram чат-бот дозволяє автоматизувати процес підбору комплектуючих для персональних комп'ютерів з урахуванням індивідуальних потреб і побажань користувача. Чат-бот забезпечує можливість підбору компонентів під різноманітні сценарії використання – ігри, робота, мультимедіа, бюджетні або високопродуктивні конфігурації та інше. Це дозволяє користувачам швидко отримувати якісні рекомендації, зменшує ймовірність помилок при складанні конфігурацій і полегшує процес вибору комплектуючих. Отже розроблена модель підвищує зручність користування, спрощує процес вибору компонентів та сприяє більш ефективній взаємодії користувача з системою.

Ключові слова: чат-бот, telegram, автоматизація, NLP, NLTK, stanza, fuzzy matching, конфігуратор ПК.

Як цитувати: Новіков О.Е., Стрілець В.Є. Модель чат-бота для конфігурування персонального комп'ютера з застосуванням методів NLP. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління.* 2025. вип. 67. С.91-100. <https://doi.org/10.26565/2304-6201-2025-67-09>

How to quote: O. Novikov, and V. Strilets, "Chatbot model for personal computer configuration using NLP methods" *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems,* vol. 67, pp. 91-100, 2025. [5https://doi.org/10.26565/2304-6201-2025-67-09](https://doi.org/10.26565/2304-6201-2025-67-09) [in Ukrainian]

1 Вступ

У сучасних умовах розвитку інформаційних технологій роль персональних комп'ютерів та високопродуктивних систем зростає надзвичайно швидкими темпами. Вони використовуються у різних сферах: від навчання та наукових досліджень до бізнес-процесів і розваг. Разом з тим постає проблема правильного підбору апаратного забезпечення, яке має відповідати індивідуальним потребам користувачів. З огляду на постійне оновлення ринку комп'ютерних

комплектуючих, де щорічно з'являються сотні нових моделей із різними характеристиками, завдання вибору оптимальної конфігурації стає складним навіть для досвідчених користувачів.

Традиційний підхід, що передбачає самостійний пошук інформації, аналіз технічних параметрів та порівняння цін, потребує значних часових витрат і глибоких технічних знань. При цьому користувачі мають різні запити: для одних важливо зібрати недорогий ПК для офісних завдань, для інших – забезпечити максимальну продуктивність у сучасних іграх, а для третіх – створити надійну робочу станцію для проектування чи обробки великих обсягів даних. Додаткову складність створює необхідність перевірки сумісності комплектуючих, що часто викликає помилки у недосвідчених користувачів.

З цієї причини виникає потреба у створенні автоматизованих систем, здатних полегшити процес підбору обладнання. Використання методів, технологій обробки природної мови NLP [1, 3] та нечіткого пошуку fuzzy matching [2] відкриває нові можливості для побудови інтелектуальних інструментів взаємодії з користувачем. Такі системи дозволяють не лише інтерпретувати запити, сформульовані у довільній формі, але й надавати персоналізовані рекомендації, адаптовані під різні сценарії використання комп'ютера.

Таким чином, створення подібних інтелектуальних систем є актуальним завданням сучасної комп'ютерної науки та практики розробки програмного забезпечення. Зростання обсягів інформації та швидкий розвиток ринку комп'ютерних технологій потребують інструментів, здатних обробляти великі масиви даних та надавати користувачам релевантні результати у зручній формі. Використання методів обробки природної мови у поєднанні з механізмами пошуку та адаптивної фільтрації відкриває можливості для створення систем, що поєднують у собі простоту у використанні та глибину аналітичних можливостей.

2 Аналіз предметної області та формулювання задачі

Сучасний ринок комп'ютерних комплектуючих пропонує користувачам великий вибір інструментів для підбору та порівняння компонентів. Одним із найбільш поширених підходів є використання онлайн-магазинів та каталогів, які дозволяють фільтрувати товари за базовими характеристиками, такими як ціна, бренд або технічні параметри. Такі платформи, як Amazon, Newegg або Rozetka, надають зручний інтерфейс для пошуку компонентів, проте вони обмежені у плані персоналізації та інтерактивності. Користувачу доводиться самостійно аналізувати сумісність компонентів і вибрати оптимальні варіанти.

Крім того, існують спеціалізовані сайти порівняння компонентів та конфігуратори, наприклад, PCPartPicker [9] або Logical Increments [10], які надають більш структуровану інформацію та дозволяють перевіряти сумісність обраних компонентів. Ці сервіси полегшують процес складання ПК і дають можливість обирати готові конфігурації для різних сценаріїв використання — від ігор до офісної роботи або обробки графіки. Проте й вони не завжди можуть врахувати індивідуальні потреби користувача або обробити запити, сформульовані у довільній текстовій формі.

Додатково користувачі звертаються до форумів та спільнот, таких як Reddit або Tom's Hardware, де обмінюються порадами щодо сумісності та ефективності компонентів. Такі ресурси надають великий обсяг практичної інформації та відгуків від досвідчених користувачів. Однак цей підхід потребує значного часу на аналіз і не забезпечує автоматизації підбору комплектуючих.

Нарешті, популярність набирають чат-боти та автоматизовані консультанти у Telegram, Discord або інших платформах, які пропонують швидкі рекомендації або готові конфігурації ПК. Вони дозволяють користувачу взаємодіяти у зручній формі, отримувати відповіді на запитання та уточнювати побажання. На жаль більшість існуючих ботів працюють за простими правилами і не підтримують аналіз довільних текстових запитів. Це обмежує їхню здатність надавати персоналізовані рекомендації, враховувати сумісність компонентів та адаптуватися під унікальні потреби користувачів.

Попри наявність численних інструментів для підбору ПК, існуючі рішення не забезпечують повної персоналізації та інтерактивності. Користувачі часто змушені самостійно аналізувати сумісність компонентів, порівнювати характеристики та приймати рішення на основі обмеженої або частково структурованої інформації. Форми взаємодії з ботами та автоматизованими консультантами залишаються досить статичними і не дозволяють ефективно обробляти довільні текстові запити користувача.

У зв'язку з цим виникає задача створення інструменту, який дозволить користувачу швидко та ефективно підібрати комплектуючі ПК відповідно до власних потреб та побажань. Основна задача

полягає у створенні інтелектуальної системи, здатної аналізувати текстові запити користувача та формувати персоналізовані рекомендації щодо підбору комплектуючих ПК. Система повинна враховувати різноманітні параметри та вимоги користувача, пропонувати варіанти комплектуючих і забезпечувати інтерактивну взаємодію. Важливою вимогою є можливість обробки довільних запитів, навіть якщо вони сформульовані нечітко або неповно, що дозволяє надати максимально адаптовані рекомендації для широкого кола користувачів.

Для забезпечення здатності інтерпретувати довільні, нечіткі або неповні запити користувача ефективним є використання сучасних методів обробки природної мови NLP та алгоритмів наближеного порівняння fuzzy matching, що дозволяють розпізнавати ключові параметри та побажання користувача і формувати персоналізовані рекомендації. Такий підхід забезпечує підвищену точність підбору комплектуючих і дозволяє адаптувати результати до індивідуальних потреб кожного користувача.

Крім того, система має забезпечувати інтерактивну взаємодію з користувачем: звертати увагу на деталі, пропонувати оптимальні варіанти та надавати інформацію про технічні характеристики та співвідношення ціни і продуктивності. Виконання цих задач дозволяє значно спростити процес підбору ПК, економити час користувачів та забезпечувати більш персоналізований і зручний досвід порівняно з існуючими рішеннями, такими як статичні онлайн-каталоги чи базові чат-боти без NLP.

У роботі описану інтелектуальну систему було реалізовано як чат-бот, що поєднує сучасні методи обробки текстової інформації та інтелектуальні алгоритми пошуку. Основним завданням чат-бота є надання користувачеві можливості отримати релевантні рекомендації щодо вибору комп'ютерних комплектуючих, сформовані відповідно до його індивідуальних потреб та вподобань. На відміну від традиційних каталогів чи статичних систем пошуку, чат-бот забезпечує інтерактивну взаємодію, що дозволяє враховувати широкий спектр факторів: від загальних вимог до продуктивності до більш специфічних побажань, які користувач може сформулювати у довільній формі. Такий підхід створює передумови для персоналізації результатів і формує більш зручний користувацький досвід.

Важливою перевагою розробленої моделі чат-бота є її універсальність і можливість адаптації до різноманітних сценаріїв використання. Вона може бути корисною як для новачків, які не мають достатніх технічних знань і прагнуть швидко отримати готові рекомендації, так і для досвідчених користувачів, що хочуть проаналізувати різні варіанти та знайти найкраще рішення. Завдяки цьому чат-бот виступає не лише інструментом пошуку, а й своєрідним цифровим консультантом, що допомагає користувачам орієнтуватися у складному та динамічному ринку комп'ютерних технологій.

3 Методи дослідження та технології обробки запитів користувача

У задачі підбору комплектуючих для персонального комп'ютера користувачі формують свої потреби у вільній формі – природною мовою. Часто такі запити містять неточності, неповні характеристики або описові формулювання на кшталт «потрібна потужна відеокарта для ігор» чи «дешевий процесор з низьким енергоспоживанням». Традиційні методи пошуку за ключовими словами в такому випадку виявляються недостатніми, оскільки не враховують варіативність мови та можливі помилки.

Саме тому для коректної інтерпретації запитів доцільно застосовувати методи обробки природної мови NLP [1, 3] та алгоритми наближеного порівняння fuzzy matching [2]. Вони дозволяють:

- виділяти ключові характеристики з тексту,
- зводити слова до базової форми,
- враховувати контекст використання,
- знаходити найбільш релевантні компоненти навіть за умови неточного формулювання запиту.

Такий підхід забезпечує гнучкість та адаптивність системи, роблячи підбір комплектуючих більш зручним і точним для користувача.

3.1 Попередня обробка тексту

Першим кроком у роботі з текстовими даними є їхня попередня обробка, яка має підготувати користувацькі запити до подальшого аналізу. Запити у вільній формі можуть містити орфографічні помилки, різні варіанти написання слів, зайві символи чи неінформативні слова. Якщо одразу передавати такий текст у систему обробки, результати будуть неточними, тому важливо провести кілька етапів очищення й стандартизації.

Одним із базових кроків є токенизація [4] (tokenization) – процес розбиття тексту на окремі елементи (токени), найчастіше слова. Це дозволяє системі працювати не з суцільним рядком, а з окремими частинами, які можна порівнювати, аналізувати й перетворювати. У Python для цього часто використовується метод `nltk.word_tokenize` [6], який враховує пунктуацію та правила мови.

Другим важливим кроком є нормалізація [5] тексту. Вона включає зведення всіх символів до нижнього регістру (щоб, наприклад, слова «Intel» та «intel» розпізнавалися як однакові), видалення пунктуації, спеціальних символів, чисел, що не несуть корисної інформації, а також усунення стоп-слів (наприклад: «і», «та», «але», «для»). Такі слова не додають смислового навантаження у пошукових запитах і лише заважають точній обробці.

Таким чином, попередня обробка тексту створює основу для подальших етапів аналізу, роблячи дані більш чистими та структурованими. Це суттєво підвищує якість лінгвістичних і семантичних методів, які застосовуються на наступних етапах.

3.2 Зведення слів до основної форми

Після попередньої обробки важливим етапом є зведення слів до базової форми, що дозволяє уникнути проблеми варіативності словоформ. Наприклад, слова «процесор», «процесора», «процесори» мають різні закінчення, але фактично позначають один і той самий об'єкт. Якщо їх не привести до єдиної форми, система може сприймати їх як різні терміни, що призведе до втрати релевантності під час пошуку.

Для цього використовуються два підходи.

Стемінг [7] (stemming) – це спрощене відсікання закінчень слів без урахування граматики. Наприклад, «грає», «грав», «грати» будуть зведені до основи «гра». Такий метод швидкий, але не завжди точний, оскільки результат не завжди збігається зі словниковою формою. Інструменти:

- SnowballStemmer [7] для української мови;
- PorterStemmer [7] для англійської мови.

Лематизація [8] (lemmatization) – це точніший метод, що ґрунтується на словникових базах та граматичних правилах. Він дозволяє привести слово до його канонічної форми (леми). Наприклад, «грає», «грав», «грати» будуть зведені саме до «грати». Лематизація є кращим підходом для роботи з українською мовою, оскільки зберігає правильність форми. Інструменти:

- WordNetLemmatizer [8] для англійської мови;
- Stanza [3] для української мови (підтримує морфологічний та лематизаційний аналіз).

Отже, стемінг доцільно застосовувати у випадках, коли потрібна швидкість і допускається певна втрата точності, тоді як лематизація краще підходить для завдань, що потребують високої точності й коректності мовних форм. У роботі для запитів українською мовою більш доцільним є поєднання обох методів: стемінг для попереднього скорочення словоформ та лематизація для уточнення базового значення слова.

3.3 Лінгвістичний аналіз

На цьому етапі система переходить від роботи з окремими словами до розуміння їхніх граматичних та синтаксичних характеристик у контексті. Це важливо для правильного інтерпретування користувацьких запитів, оскільки одні й ті самі слова можуть мати різні значення залежно від ролі у реченні.

Морфологічний аналіз [9] полягає у визначенні частини мови (іменник, прикметник, дієслово тощо), а також граматичних характеристик — роду, числа, відмінка, часу. Наприклад, у запиті «підбери нову відеокарту для ігор» система має розпізнати, що слово «нову» є прикметником, який описує характеристику «відеокарти», а «для ігор» вказує на ціль використання.

Синтаксичний аналіз визначає залежності між словами в реченні. Це дозволяє встановити, які слова є головними, а які допоміжними. Наприклад, у реченні «хочу потужний процесор для роботи» система повинна зрозуміти, що слово «потужний» описує саме «процесор», а не «роботу».

Для виконання морфологічного та синтаксичного аналізу використовується stanza.Pipeline [3] – багатомовна NLP-бібліотека від Stanford NLP, яка підтримує українську мову. Вона дозволяє отримати повний розбір речення, а саме:

- визначити частини мови для кожного слова;
- встановити граматичні характеристики;
- побудувати дерево синтаксичних залежностей.

Таким чином, лінгвістичний аналіз дає змогу не лише розуміти окремі слова, але й інтерпретувати сенс усього запиту, що є критично важливим для правильної рекомендації ПК-компонентів.

3.4 Семантичний рівень

Після лінгвістичного аналізу система переходить до розуміння значення слів та фраз у контексті запиту, що дозволяє робити більш точні та персоналізовані рекомендації.

Аналіз тональності [10] (sentiment analysis) дозволяє визначати емоційне забарвлення запиту користувача. Наприклад, користувач висловлює позитивне враження, невдоволення або нейтральне запитання. Це може бути корисно для визначення нагальності або пріоритетності рекомендацій. Інструменти: TextBlob [11], який дозволяє отримати полярність та суб'єктивність тексту.

Виділення сутностей [12] (Named Entity Recognition, NER) дає змогу розпізнавати конкретні об'єкти та ключові елементи запиту, такі як назви товарів, брендів, категорії компонентів, моделі, технічні характеристики або терміни, що відносяться до часу чи організацій. Інструменти: stanza [3] та spacy [13], які підтримують багатомовний NER та дозволяють виділяти сутності навіть у складних українських текстах.

Використання семантичного рівня дає можливість системі не просто знаходити ключові слова, а й розуміти, що саме користувач має на увазі, навіть якщо запит сформульований нечітко або містить неточні формулювання. Це важливо для підбору компонентів комп'ютера, адже користувач може описувати свої потреби у вільній формі, без використання стандартних характеристик чи термінів.

3.5 Методи підвищення точності Fuzzy matching

Однією з ключових проблем у сфері підбору комп'ютерних компонентів є неструктурованість і неточність користувацьких запитів. Більшість користувачів формулює свої потреби у вільній формі, не дотримуючись єдиних стандартів написання назв моделей або технічних характеристик. Це призводить до того, що пряме порівняння рядків не дає бажаного результату.

У таких випадках fuzzy matching виступає ефективним методом для підвищення точності пошуку та відбору релевантних варіантів. Суть цього підходу полягає у знаходженні найбільш схожих рядків за певною метрикою, навіть якщо вони відрізняються за написанням. Для цього використовуються алгоритми обчислення відстані редагування (наприклад, Levenshtein distance), що враховують кількість операцій (вставка, видалення, заміна символів), необхідних для перетворення одного рядка в інший.

Завдяки цьому fuzzy matching дозволяє:

- розпізнавати орфографічні помилки у запитах;
- виявляти скорочення та різні варіанти написання назв моделей;
- враховувати різні мовні форми введення користувачем;
- підвищувати релевантність результатів, навіть якщо збіг не є буквальним.

Інструменти fuzzy matching:

- fuzzywuzzy [14] – класична бібліотека для Python, яка застосовує Levenshtein distance для порівняння рядків;
- rapidfuzz [15] – більш оптимізована та швидка альтернатива, що забезпечує високу продуктивність при роботі з великими наборами даних (наприклад, каталогами компонентів).

Практичне застосування fuzzy matching у завданні підбору ПК-компонентів дозволяє створити систему, яка буде толерантною до помилок користувача та надаватиме найбільш відповідні результати навіть у випадку неточного формулювання запиту.

4 Проектування архітектури моделі чат-бота

Реалізований чат-бот побудовано за принципом клієнт-серверної архітектури. Клієнтська частина представлена платформою Telegram – користувачі взаємодіють із ботом через стандартний інтерфейс месенджера. Серверна частина виконує всю бізнес-логіку: прийом повідомлень через Telegram API [16], обробку тексту NLP, пошук релевантних компонентів у базі даних, формування та відправку відповідей. Така архітектура дозволяє розділити інтерфейс взаємодії й обчислювальні ресурси, спростити масштабування та оновлення логіки без втручання у клієнтську частину (рис. 1).

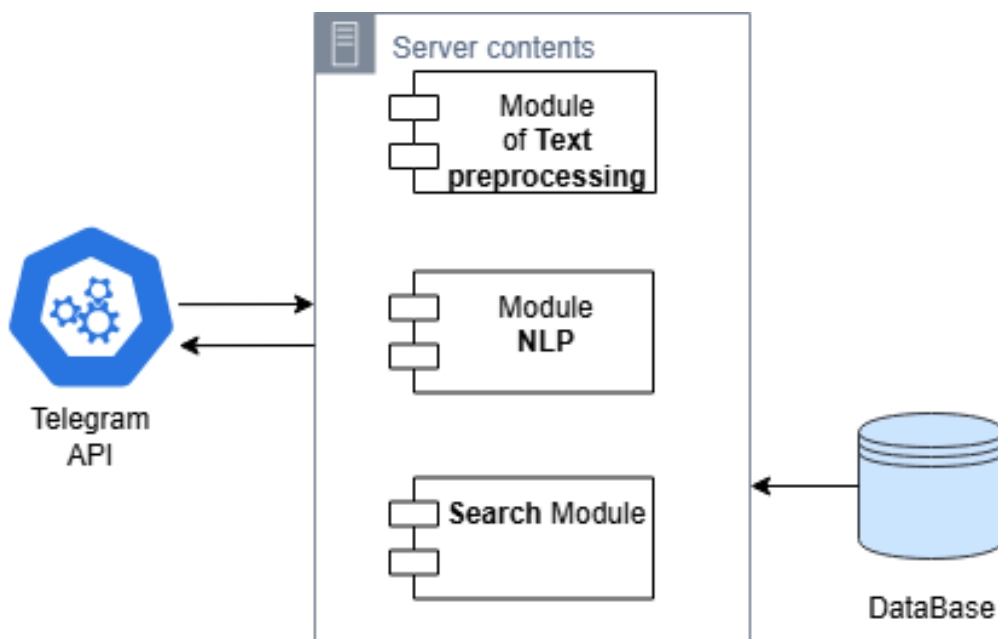


Рис. 1. Схема архітектури чат-бота
Fig. 1. The chatbot architecture diagram

Архітектура складається з таких основних частин:

- Telegram-інтерфейс – обробник вебхуків / полінгу, що отримує повідомлення від користувачів і надсилає відповіді. Зазвичай реалізовано через бібліотеку python-telegram-bot.
- модуль попередньої обробки тексту відповідає за нормалізацію, токенізацію, видалення стоп-слів.
- NLP-пайплайн виконує стемінг, лематизація, морфологічний та синтаксичний аналіз, міститьNER-модуль [12].
- пошуковий модуль, ранжування — fuzzy matching і логіка відбору компонентів.
- база даних зберігає каталог компонентів, їх характеристик, цінкових параметрів та тегів сумісності.

Запит користувача опрацьовується у кілька етапів. На рис. 2 показаний спрощений потік обробки запитів.

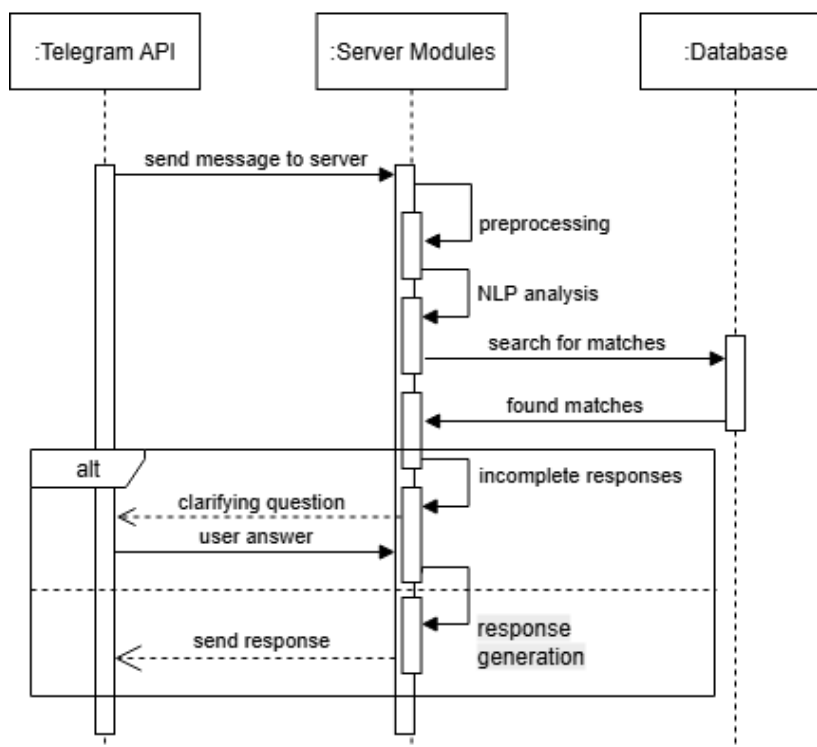


Рис. 2. Спрощена діаграма обробки повідомлень
Fig. 2. Simplified message processing diagram

Користувач надсилає повідомлення в Telegram чат-бот, після цього відбувається послідовність дій, які формують відповідь:

- Telegram пересилає повідомлення на сервер;
- попередня обробка: приведення до нижнього регістру, очищення, токенізація;
- NLP-аналіз: лематизація, стемінг, морфосинтаксичний і лінгвістичний аналіз;
- пошук відповідностей: поєднання структурованих параметрів із записами в базі даних; застосування fuzzy matching для наближених збігів;
- складання варіантів: ранжування знайдених компонентів за релевантністю (враховуючи сумісність, пріоритети користувача, ціновий діапазон тощо);
- діалогова взаємодія: якщо параметрів недостатньо, то бот ставить уточнювальні питання, якщо вистачає – формує відповідь з рекомендаціями.
- відправка відповіді користувачеві через Telegram.

5 Реалізація та приклади роботи системи

Розроблений чат-бот у Telegram забезпечує зручний процес взаємодії з користувачем під час вибору комп'ютерних комплектуючих. Користувач формує свій запит у довільній формі, після чого бот виконує його обробку, застосовуючи алгоритми NLP та методи нечіткого пошуку. У відповідь користувач отримує стислий опис рекомендованих компонентів, які найбільше відповідають критеріям запиту.

Для детальнішого ознайомлення бот формує і додатково надсилає файл із повним описом підібраних комплектуючих, що містить характеристики, ціну та інші параметри. Після цього користувачеві пропонується можливість надіслати цей файл на електронну пошту, і в разі згоди бот автоматично виконує відправлення.

На рис. 3а продемонстровано роботу чат-бота при надходженні на нього звичайного, коректного запиту користувача. На рис. 3б продемонстровано роботу чат-бота при надходженні на нього запиту з помилками. В обох випадках чат-бот демонструє правильне сприйняття запиту і коректну відповідь на нього.

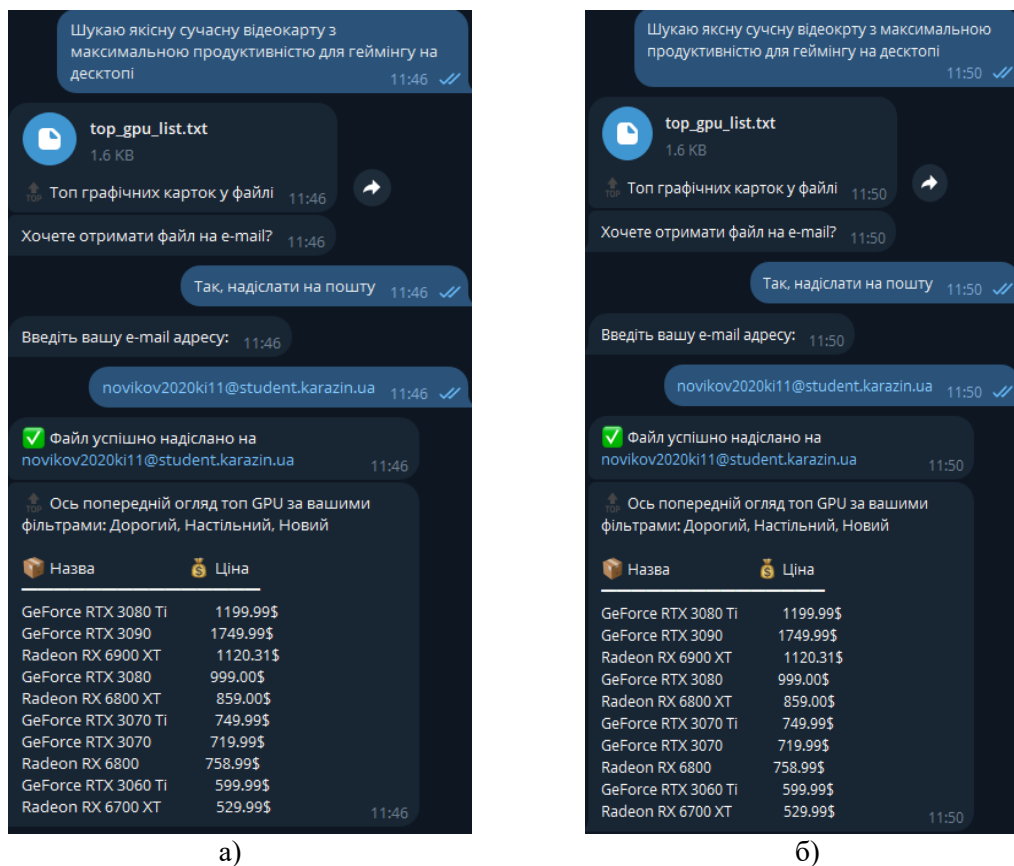


Рис. 3. Приклад роботи чат-бота: а) з коректним запитом; б) із запитом, який містить помилки
Fig. 3. Example of chatbot operation: a) with a correct query; b) with a query containing errors

На рис. 4 продемонстровано зміст файлу який формується і надається ботом користувачу.

```

Top GPU за фільтрами: Дорогий, Настільний, Новий
gpuName  G3Dmark  G2Dmark  total_performance  performance_per_dollar  gpuValue  TDP  powerPerformance  testYear  price  category
GeForce RTX 3080 Ti  26887  1031  27918  23.265194  22.41  350.0  76.82  2021  1199.99  Desktop
GeForce RTX 3090  26395  999  27394  15.653804  15.08  350.0  75.41  2020  1749.99  Desktop
Radeon RX 6900 XT  25458  1102  26560  23.707724  22.72  300.0  84.86  2020  1120.31  Desktop
GeForce RTX 3080  24853  1003  25856  25.881882  24.88  320.0  77.66  2020  999.00  Desktop
Radeon RX 6800 XT  23364  1078  24442  28.454016  27.20  300.0  77.88  2020  859.00  Desktop
GeForce RTX 3070 Ti  23367  1003  24370  32.493767  31.16  290.0  80.58  2021  749.99  Desktop
GeForce RTX 3070  22093  969  23062  32.031000  30.69  220.0  100.42  2020  719.99  Desktop
Radeon RX 6800  20667  1030  21697  28.586674  27.23  250.0  82.67  2020  758.99  Desktop
GeForce RTX 3060 Ti  20206  961  21167  35.278921  33.68  200.0  101.03  2020  599.99  Desktop
Radeon RX 6700 XT  18993  1014  20007  37.749769  35.84  230.0  82.58  2021  529.99  Desktop
    
```

Рис. 4. Зміст відправленого файлу
Fig. 4. Contents of the sent file

6 Оцінка ефективності та перспективи розвитку

Розроблений Telegram чат-бот продемонстрував ефективність у вирішенні задачі підбору комп'ютерних комплектуючих на основі запитів користувача у вільній формі. Використання методів обробки природної мови NLP та технологій наближеного порівняння fuzzy matching дозволило системі коректно інтерпретувати як точні, так і неточні або неповні запити. Завдяки цьому досягається висока точність відповідей навіть у випадках орфографічних помилок чи неточних формулювань.

Функціонал бота забезпечує зручність взаємодії: користувач отримує стислий перелік рекомендованих компонентів у чаті та може завантажити файл із розширеним описом. Додатковою перевагою є можливість надсилання цього файлу на електронну пошту, що робить систему більш практичною для кінцевого користувача. Оцінка швидкодії показала, що обробка запиту та формування відповіді відбувається у прийнятний час, що гарантує комфортне користування сервісом.

Подальший розвиток системи може включати кілька напрямів удосконалення. По-перше, інтеграція з онлайн-магазинами дозволить отримувати актуальні ціни на комплектуючі та

формувати персоналізовані кошики. По-друге, розширення датасету забезпечить охоплення більшої кількості моделей і брендів, що підвищить актуальність рекомендацій. Важливим кроком буде додавання посилань на сайти-постачальники у датасеті, щоб користувач одразу міг перейти до покупки обраного товару. Крім того, доцільно впровадити можливість створення списку обраних комплектуючих, що дозволить користувачеві зберігати цікаві варіанти для подальшого порівняння.

Перспективним є також впровадження сучасних моделей NLP на основі трансформерів (наприклад, BERT чи GPT), які здатні значно підвищити якість розуміння користувацьких запитів. Додатково можна розглянути інтеграцію голосових команд, а також реалізацію персоналізації на основі історії попередніх звернень користувача.

7 Висновки

У результаті проведеного дослідження було створено Telegram чат-бот, здатний аналізувати довільні текстові запити користувача та надавати персоналізовані рекомендації щодо підбору комп'ютерних комплектуючих. Система поєднує сучасні методи обробки природної мови NLP та алгоритми fuzzy matching, що забезпечує коректну інтерпретацію навіть неточних або неповних формулювань. На відміну від статичних онлайн-каталогів або базових чат-ботів без NLP, розроблена модель здатна гнучко реагувати на запити користувачів та формувати релевантні рекомендації, враховуючи широкий спектр характеристик комплектуючих.

Завдяки впровадженню NLP-технологій чат-бот здатний аналізувати тексти користувачів, виділяти ключові характеристики та побажання, а fuzzy matching дозволяє коригувати можливі орфографічні помилки чи неповні формулювання. Це значно підвищує точність підбору комплектуючих і робить систему доступною для користувачів із різним рівнем технічних знань.

Таким чином, розроблений Telegram чат-бот значно спрощує процес вибору комп'ютерних комплектуючих, роблячи його більш персоналізованим, інтерактивним та ефективним. Система демонструє практичну цінність і потенціал для подальшого розвитку, забезпечуючи користувачу швидкий доступ до необхідної інформації та полегшуючи прийняття рішень у процесі підбору ПК-компонентів.

REFERENCES

1. Text Processing and NLP in Python : website. URL: <https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk>
2. What Is Fuzzy Matching and How Can It Clean Up My Bad Data? : website. URL: <https://profisee.com/fuzzy-matching/>
3. Using Stanza for NLP Tasks in Python : website. URL: <https://stanfordnlp.github.io/stanza/>
4. Tokenization in NLP : website. URL: <https://www.geeksforgeeks.org/nlp/nlp-how-tokenizing-text-sentence-words-works/>
5. Text Normalization for Natural Language Processing : website. URL: <https://medium.com/data-science/text-normalization-for-natural-language-processing-nlp-70a314bfa646>
6. NLTK Documentation : website. URL: <https://www.nltk.org/>
7. What Is Stemming? | IBM : website. URL: <https://www.ibm.com/think/topics/stemming>
8. Lemmatization in NLP : website. URL: <https://medium.com/@kevinjagi83/lemmatization-in-nlp-2a61012c5d66>
9. What is Morphological Analysis in Natural Language Processing (NLP)? : website. URL: <https://www.geeksforgeeks.org/nlp/morphological-analysis-in-nlp/>
10. What is Sentiment Analysis? : website. URL: <https://www.ibm.com/think/topics/sentiment-analysis>
11. TextBlob Documentation : website. URL: <https://textblob.readthedocs.io/en/dev/>
12. What is Named Entity Recognition? : website. URL: <https://www.ibm.com/think/topics/named-entity-recognition>
13. Industrial-Strength Natural Language Processing : website. URL: <https://spacy.io/>
14. FuzzyWuzzy Python Library: Interesting Tool for NLP and Text Analytics : website. URL: <https://www.analyticsvidhya.com/blog/2021/06/fuzzywuzzy-python-library-interesting-tool-for-nlp-and-text-analytics/>
15. RapidFuzz Documentation : website. URL: <https://rapidfuzz.github.io/RapidFuzz/>
16. Telegram Bot API Documentation : website. URL: <https://core.telegram.org/bots/api>
17. How to Build a Telegram Bot in Python : website. URL: <https://core.telegram.org/bots/samples>

18. PCPartPicker : website. URL: <https://pcpartpicker.com/>
19. Logical Increments : website. URL: <https://www.logicalincrements.com/>
20. Rozetka : website. URL: <https://rozetka.com.ua/>
21. Amazon : website. URL: <https://www.amazon.com/>
22. Veres O., Hadzalo O. Application of Methods of Recommendations in the Analysis of Computer Components. SISN. 2023. Vol. 14. P. 84–98. <https://doi.org/10.23939/sisn2023.14.084> [in Ukrainian]
23. Chatwattana P., Yangthisarn P., Tabubpha A. The Educational Recommendation System with Artificial Intelligence Chatbot: A Case Study in Thailand : article. International Journal of Engineering Pedagogy (iJEP). 2024. Vol. 14, No. 5. P. 51–64. <https://doi.org/10.3991/ijep.v14i5.48491>
24. Bird S., Klein E., Loper E. Natural Language Processing with Python : textbook. O'Reilly Media. United States of America, 2009. 502 p. https://eSearchgate.net/publication/220691633_Natural_Language_Processing_with_Python

Strilets Viktoriia *Ph.D, associate professor of the Department of Computer Systems and Robotics, Education and Research Institute of Computer Sciences and Artificial Intelligence, V.N. Kharkiv National University, 6 Svobody sq., Kharkiv, Ukraine, 61022*

Novikov Oleksii *student of the Education and Research Institute of Computer Sciences and Artificial Intelligence, V.N. Karazin Kharkiv National University, 6 Svobody sq., Kharkiv, Ukraine, 61022*

Chatbot model for personal computer configuration using NLP methods

Objective: to improve the convenience and efficiency of selecting personal computer components by using a Telegram chatbot with NLP methods to process user requests.

Research Methods: methods of natural language processing NLP were used to interpret user queries and generate chatbot responses; methods for building dialogue systems; and approaches to organizing software components. The Telegram chatbot was implemented based on a client-server architecture, where the client side provides interaction with the user on Telegram, and the server side handles data processing and PC component selection logic. The implementation used the following technologies: Python programming language, the python-telegram-bot library for creating the chatbot, NLP tools for analyzing and interpreting user queries, and fuzzy matching to improve search results.

As a **result**, a Telegram chatbot was created to automate the process of selecting components for personal computers, taking into account individual user needs and preferences. The system allows users to quickly receive recommendations for selecting PC components such as CPU, GPU, RAM, storage, motherboard, and power supply, considering price category, intended purpose (gaming, work, multimedia), and desired specifications. The chatbot provides a convenient interaction through Telegram, while the server side handles request processing, text analysis, and generating optimal configurations using NLP methods and fuzzy matching. For natural language processing, the libraries and tools used include Stanza, NLTK (tokenization, stemming, lemmatization), and TextBlob; for fuzzy search, RapidFuzz was applied. Using Python and the python-telegram-bot library ensures reliable system performance, flexibility in scaling, and the ability to quickly update the component database.

Conclusions: The developed Telegram chatbot allows automating the selection of PC components according to individual user needs and preferences. The system enables component selection for various use cases — gaming, work, multimedia, budget or high-performance configurations, and more. This allows users to quickly receive optimal recommendations, reduces the likelihood of errors when assembling configurations, and simplifies the component selection process. The developed system improves user convenience, optimizes the component selection process, promotes more efficient user interaction with the system.

Keywords: chatbot, telegram, automation, NLP, NLTK, stanza, fuzzy matching, PC configurator.