DOI 10.34185/1562-9945-5-162-2026-09
УДК 004

О.С. Волковський, О.Ю. Нікішина

МОДЕЛЮВАННЯ СИСТЕМИ ЧЕРГ ДЛЯ ГЕНЕРАЦІЇ SEO-ОПТИМІЗОВАНИХ ТЕКСТІВ У ВИСОКОНАВАНТАЖЕНОМУ СЕРЕДОВИЩІ

Анотація. Робота присвячена розробці системи для автоматизованої генерації SEO-оптимізованих текстів з подальшим моделюванням системи черг для зниження витрат на створення якісного контенту, спрощення процесу генерації текстів та забезпечення їх відповідності сучасним SEO-вимогам. Дослідженню механізмів взаємодії з ШІ-сервісами для оптимізації процесу генерації текстів в умовах обмежень на кількість запитів та швидкість обробки даних. Проведені дослідження наявних рішень для автоматичної генерації контенту, у тому числі розглянуті сервіси на базі штучного інтелекту, такі як ChatGPT. У результаті були виявлені обмеження існуючих підходів, пов'язані переважно з їхньою недостатньою масштабованістю або обмеженнями API. Синтезована Q-схема для вирішення конкретної задачі, адаптований асинхронний моделюючий алгоритм, здійснене моделювання системи черг SEO-оптимізованих текстів. Вдосконаленні алгоритми взаємодії з ШІ-сервісами забезпечують стабільну та ефективну роботу, дозволяють не лише здешевити процес наповнення контентом, але й зробити його максимально ефективним, зберігаючи якість текстів на рівні, що не поступається роботі професійних копірайтерів.
Ключові слова. SEO-оптимізовані тексти, ШІ-сервіси, Q-схема.

Постановка проблеми. Останні роки стали переломними у сфері автоматизації завдяки стрімкому розвитку технологій штучного інтелекту (далі ШІ). Інструменти на основі машинного навчання, такі як нейромережі для генерації текстів, відкрили нові можливості для автоматизації контент-маркетингу та інших бізнес-процесів. Раніше написання статей, описів товарів і маркетингових матеріалів вимагало значних людських ресурсів, однак сьогодні з'явилися ШІ-сервіси, здатні виконувати цю роботу з високим рівнем якості та швидкості.

Комерційні ШІ-системи вже досягли рівня, при якому вони можуть замінити професійних копірайтерів у багатьох завданнях. Алгоритми глибокого навчання аналізують великі обсяги текстових даних, імітуючи стилістику та логіку написання людини. Це дозволяє створювати тексти, що відповідають вимогам якості, унікальності та SEO-оптимізації.

SEO (Search Engine Optimization) – це комплекс методів і стратегій, спрямованих на покращення позицій вебсайтів у результатах пошукових систем, таких як Google. Оптимізовані тексти повинні бути унікальними, містити релевантні ключові слова та

відповідати вимогам алгоритмів ранжування. Впровадження ШІ у процес створення контенту дозволяє не лише автоматизувати його написання, але й враховувати актуальні вимоги SEO для досягнення кращих результатів у пошукових системах.[1]

Автоматизація процесу наповнення сайтів контентом особливо актуальна для інтернет-магазинів, маркетингових агентств, блогерів та інформаційних платформ. Раніше для написання та оптимізації великого обсягу текстів потрібно було залучати команди копірайтерів та редакторів, що вимагало значних фінансових вкладень. Використання ШІ-рішень дозволяє зменшити витрати на створення контенту, роблячи цей процес більш доступним для малого та середнього бізнесу. Разом із розвитком ШІ-сервісів змінюється і ринок цифрового контенту. Пошукові системи постійно оновлюють свої алгоритми ранжування, що створює додаткові виклики для SEO-оптимізації. Автоматизована генерація текстів повинна не лише забезпечувати якісне наповнення сайтів, а й відповідати сучасним вимогам SEO, таким як унікальність, релевантність та використання ключових слів у правильному контексті. Це вимагає розробки гнучких алгоритмів, які враховують зміни у пошукових алгоритмах і адаптуються до нових вимог.

Крім того, проблема високонавантаженості систем автоматизації контенту стає все більш актуальною. Використання ШІ-генерації вимагає балансування між швидкістю обробки даних та обмеженнями сервісів. Багато платформ накладають ліміти на кількість одночасних запитів, що може сповільнювати процес масового створення контенту. Тому необхідно розробляти ефективні системи черг, які забезпечують рівномірний розподіл навантаження та мінімізують затримки в генерації текстів.

Ще одним викликом є підтримка різних форматів та стилістики текстів, оскільки різні галузі бізнесу мають свої вимоги до контенту. Наприклад, опис товарів для інтернет-магазину потребує чіткої структури та акценту на ключових характеристиках, тоді як статті для блогів вимагають більш природного стилю викладу. Розробка універсальної системи автоматизації повинна передбачати можливість налаштування параметрів генерації, адаптації під різні типи контенту та інтеграцію з іншими сервісами для пост обробки текстів.

В рамках цієї роботи є розглядається процес створення системи для автоматизованої генерації SEO-оптимізованих текстів з подальшим моделюванням системи черг для зниження витрат на створення якісного контенту. Крім того, розглядається механізми взаємодії з ШІ-сервісами, що дозволяють оптимізувати процес генерації текстів в умовах обмежень на кількість запитів та швидкість обробки даних.

Аналіз існуючиих рішень. На сьогоднішній день існує декілька інструментів для автоматичної генерації описів товарів, які використовуються компаніями для спрощення процесу наповнення сайтів контентом. Більшість із цих рішень інтегровані у популярні системи управління контентом (CMS) або онлайн-магазини, що дозволяє швидко створювати тексти без залучення копірайтерів. Проте існуючі системи мають ряд обмежень та проблем, які заважають їх широкому використанню в промислових масштабах.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

1. Обмеженість платформ. Більшість рішень створені для конкретних платформ, таких як Shopify, OpenCart, WordPress та інші. Це означає, що їх можливості обмежені архітектурою цих платформ, і вони не можуть бути легко адаптовані для інших потреб або масштабовані на великі обсяги контенту.

2. Недостатня гнучкість налаштувань. Більшість готових рішень не дозволяють користувачам впливати на параметри генерації тексту, такі як стилістика написання, довжина тексту, рівень деталізації або специфіка використання ключових слів. Це робить такі системи малопридатними для бізнесів, які потребують індивідуального підходу до контенту.

3. Відсутність SEO-оптимізації. Існуючі рішення майже не враховують SEO-оптимізацію. Зокрема, вони не здійснюють перевірку на унікальність текстів, що є критично важливим для підвищення позицій сайту у пошукових системах. В результаті, автоматично згенеровані описи можуть містити дубльований контент, що негативно впливає на ранжування сторінок у пошукових системах.

4. Неможливість масової генерації. Більшість інструментів не передбачають можливості масової генерації текстів та управління процесом створення контенту у високонавантажених середовищах. Більшість сервісів не підтримують систему черг, що призводить до затримок у генерації або перевантаження сервісу при обробці великої кількості запитів одночасно.

5. Проблема довжини та якості текстів. Більшість систем створюють короткі та загальні описи, які не містять достатньої кількості унікальної інформації про товар. Це знижує їхню ефективність як інструментів для підвищення конверсії та залучення клієнтів.

Окремо варто розглянути WordPress-плагін "ШІ Content Creator - Easy ChatGPT powered article generator"[2]. Це рішення базується на інтеграції з ChatGPT, що дозволяє автоматично генерувати тексти безпосередньо через інтерфейс WordPress. Проте, з точки зору SEO, цей підхід має значні обмеження. Користувач не може задати чітку кількість ключових слів, що ускладнює налаштування контенту під вимоги пошукових алгоритмів. Також відсутня перевірка на унікальність, що може призвести до створення дублікатів контенту та негативно вплинути на ранжування сайту. По суті, цей плагін є лише спрощеним інтерфейсом для відправки запитів до ШІ-моделі, без додаткової обробки тексту, що обмежує його ефективність у промисловому використанні. SE Ranking [3] є комплексним SEO-інструментом, що включає можливості аналізу текстів та генерації контенту.

Основні особливості:

1. Інтеграція ключових слів – сервіс дозволяє гнучко налаштовувати ключові слова, що сприяє створенню SEO-оптимізованого контенту.

2. Аналітика текстів і рейтинги – користувач отримує розширену аналітику текстів, включаючи їх відповідність пошуковим запитам та рейтинги якості.

3. Обмеження у генерації – система не передбачає масову генерацію текстів, а працює з кожним запитом окремо, що уповільнює процес контентного наповнення великих сайтів.

4. Мовні обмеження – SE Ranking підтримує лише англійську мову, тоді як ChatGPT працює з багатьма мовами одночасно.

5. Відсутність перевірки унікальності – сервіс не має вбудованого інструменту для перевірки тексту на унікальність, що є важливим фактором для SEO.

Виклад основного матеріалу дослідження. Для створення текстів у системі був обраний сервіс ChatGPT від OpenШІ. Основними причинами вибору саме цього сервісу є:

- Висока якість текстів та здатність адаптувати стиль написання під вимоги користувача.
- Гнучкість у налаштуванні параметрів генерації, що дозволяє створювати як короткі, так і розгорнуті описи товарів.
- Можливість інтеграції через API, що дозволяє автоматизувати процеси генерації текстів у веб-додатку.

Альтернативні рішення, такі як GitHub Copilot та DeepSeek, були розглянуті, але вони не підходять для цієї задачі з наступних причин:

- GitHub Copilot – орієнтований переважно на генерацію програмного коду та не адаптований для створення комерційного контенту.
- DeepSeek – показує малу стабільність та обмеження щодо довжини текстів, що критично важливо для SEO-оптимізованого контенту.
- Grok – Орієнтацією на неформальний стиль та гумор, що може бути неприйнятним для створення комерційного контенту

Інтеграція сервісів перевірки унікальності

Одним із важливих аспектів роботи системи є перевірка унікальності згенерованого контенту. Для цього планується інтеграція таких сервісів, як:

- Copyleaks – надає потужні інструменти для перевірки текстів на дублювання, підтримує API та забезпечує швидку обробку запитів.
- Duplichecker – популярний сервіс зручний для швидкої перевірки невеликих обсягів тексту.
- Prepostseo – один із варіантів для перевірки унікальності з додатковими інструментами SEO-аналізу.

Вибір декількох сервісів дозволяє підвищити точність перевірки та забезпечити коректну обробку контенту, що критично важливо для SEO-просування сайтів. Оскільки система буде працювати у високонавантаженому середовищі, важливим етапом розробки є реалізація ефективної системи черг. Черги дозволять рівномірно розподіляти навантаження між запитами, уникати перевантаження сервісу генерації текстів та забезпечувати стабільну роботу навіть при великій кількості користувачів.

Основні принципи організації черг:

- Пріоритизація запитів – можливість прискореного оброблення критично важливих запитів.
- Обмеження одночасних запитів – забезпечення рівномірного навантаження на API ChatGPT та сервіси перевірки унікальності.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

• Автоматичне повторення запитів у разі помилки – запобігання втраті даних через технічні збої.

Таким чином, підхід до розробки додатку включає вибір веб-додатку як основної платформи, використання ChatGPT для генерації текстів, інтеграцію сервісів перевірки унікальності та реалізацію ефективної системи черг для стабільної роботи у високонавантажених умовах.

Вибір технологій розробки є критично важливим етапом створення інформаційних систем, оскільки саме від нього залежать продуктивність, масштабованість, надійність і можливість подальшого розширення функціоналу. У рамках цієї роботи використовується стек технологій, до складу якого входять PHP, CodeIgniter, MySQL, jQuery, HTML, Bootstrap. Вибір кожної з цих технологій зумовлений їхніми особливостями, що дозволяють створити ефективну систему генерації контенту.

Моделювання системи черг ґрунтується на розробці алгоритмів, що виконують потоки подій та функціонування компонентів системи. Основна мета моделюючого алгоритму — відображати поведінку системи таким чином, щоб вона була максимально наближена до реального процесу, але водночас залишалася ефективною з точки зору обчислювальних ресурсів.

Для оцінки продуктивності системи та визначення реальних часових витрат на генерацію та перевірки унікальності SEO-оптимізованих текстів була проведена серія практичних вимірювань. Дослідження проводилося у середовищі, що максимально наближене до робочих умов системи. Усі вимірювання здійснювалися на одному апаратному забезпеченні, з мінімальними фоновими навантаженнями, що могли б вплинути на результати

Для генерації текстів використовувався API ChatGPT (GPT-4o), який отримував запит на створення розгорнутого тексту обсягом від 10 000 до 14 000 символів. Час генерації замірявся починаючи з моменту надсилання запиту і до моменту отримання повного тексту. Було здійснено п'ять незалежних спроб, результати яких наведено в таблиці 1.

Таблиця 1

Час повернення відповіді сервісу ChatGPT

№ спроби	**Час генерації (секунди)**
1	34
2	41
3	57
4	39
5	46

Середній час генерації тексту склав приблизно 43,4 секунди. Варто зазначити, що відхилення у результатах можуть бути спричинені коливаннями навантаження на сервер ChatGPT у певні моменти часу, а також складністю запиту

Важливим аспектом є оцінка кількості токенів, що використовуються при генерації тексту. В середньому 1 000 символів відповідає приблизно 250 токенам. Таким чином, для тексту обсягом 10 000–14 000 символів потрібно від 2 500 до 3 500 токенів.

Вартість використання API GPT-4o складає $0.005 за 1 000 вхідних токенів та $0.015 за 1 000 вихідних токенів [9] . Для одного запиту середня вартість буде:

1. Вхідні токени: 300 × $0.005 = $0.0015
2. Вихідні токени: 3 000 × $0.015 = $0.045
3. Загальна вартість одного запиту: $0.0465

Для порівняння, використання GPT-4-turbo коштує $0.01 за 1 000 вхідних токенів та $0.03 за 1 000 вихідних, що майже вдвічі дорожче. Таким чином, для завдань дипломного проєкту вибір GPT-4o є більш економічно обґрунтованим, оскільки він забезпечує прийнятний баланс між якістю та витратами.

Після генерації тексту наступним етапом є перевірка унікальності за допомогою спеціалізованих онлайн-сервісів. У межах експерименту було обрано популярний сервіс Copyleaks, який пройшов тестування у п’яти незалежних ідентичних запитах. Час обробки вимірювався аналогічним способом – від моменту завантаження тексту до отримання результату. Результати наведено у таблиці 2.

Таблиця 2

Час повернення відповіді сервісу Copyleaks

№ спроби	Час (хвилини)
1	12,0
2	19,2
3	9,2
4	14,4
5	16,4

У середньому процес перевірки унікальності займав 14,2 хвилини. Це досить суттєвий показник, оскільки в умовах масової генерації контенту такі часові витрати можуть стати «вузьким місцем» у всій системі. Зважаючи на це, доцільно розглянути можливості оптимізації етапу перевірки, наприклад, за рахунок використання швидших сервісів або паралельної обробки текстів.

Асинхронна обробка текстів. У таблиці 3 наведено кількість текстів, які можуть оброблятися паралельно за допомогою різних сервісів або інструментів:

Таблиця 3

Час повернення відповіді сервісу Copyleaks

Інструмент/сервіс	Максимальна кількість паралельних обробок
Copyleaks	50
Duplichecker	15
ChatGPT (асистенти)	30

Отримані результати свідчать про необхідність ефективного планування ресурсів для забезпечення стабільної роботи системи. Використання черг дозволяє рівномірно розподіляти навантаження, що особливо важливо у випадку роботи з великою кількістю запитів у реальному часі. Також слід враховувати, що затримки у відповідях API можуть змінюватися залежно від зовнішніх факторів, таких як завантаженість серверів

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

або обмеження швидкості запитів, тому подальші експерименти можуть включати аналіз впливу цих чинників.

Практичні виміри дозволили встановити орієнтовні часи виконання двох основних операцій - генерації тексту та його унікалізації. Середній час генерації одного тексту складає 43,4 секунди, у той час як перевірка унікальності займає в середньому 14,2 хвилини. Система має підтримувати обробку десятків текстів одночасно, при цьому ChatGPT (асистенти) здатен генерувати до 30 текстів паралельно, Copyleaks перевіряє до 50 текстів, а Duplichecker - до 15.

Враховуючи ці характеристики, була передбачена система з використанням моделі черг, яка дозволить рівномірно розподіляти навантаження між генераторами, сервісами унікалізації та зовнішніми перевірками. Таким чином, завдання моделювання полягає в аналізі та оптимізації роботи такої системи на основі реальних часових параметрів і технічних обмежень.

На основі розрахунків продуктивності системи для 100 заявок дійшли наступних висновків: коефіцієнти завантаженості на етапах генерації та SEO-допрацювання ($\rho \approx 0{,}075$ та $\rho \approx 0{,}015$ відповідно) вказують на низький рівень навантаження, що означає ефективне функціонування цих компонентів із великим запасом потужності - ресурси використовуються лише на 7–8%. Водночас етап унікалізації демонструє суттєве навантаження ($\rho = 1{,}04$), що майже досягає критичного рівня. Це свідчить про те, що Copyleaks працює на межі своїх можливостей, а сам процес унікалізації стає головним обмежувальним чинником усієї системи. Саме тому для подальшого моделювання була обрана асинхронна циклічна модель. Усі задачі обробляються у фоновому режимі, що дозволяє системі запускати десятки паралельних процесів без блокування головного потоку. Кожна задача має свій статус виконання (is_processing = 1, is_processing = 2), що дозволяє контролювати обробку без очікування завершення інших задач. Модель підтримує обробку до 30 активних задач на кожен сервіс одночасно. Окрема увага була приділена забезпеченню повної автономності системи обробки. Замість побудови складної інфраструктури на базі RabbitMQ чи подібних брокерів повідомлень, реалізовано просту, але надійну модель через фонові PHP-процеси. Ключова роль тут відведена механізму блокування (flock) через файл generate_worker.lock, який гарантує, що одночасно не буде запущено кілька воркерів. Цей механізм дозволяє створити легку, незалежну чергову систему, яка не потребує постійного контролю ззовні та не залежить від сторонніх сервісів або інтерфейсів взаємодії. У випадку збоїв система самостійно завершує обробку задач, не порушуючи цілісності даних.

У наведеній нижче Q-схемі представлено логіку роботи системи, яка генерує, перевіряє та доопрацьовує SEO-тексти у високонавантаженому асинхронному середовищі. Схема на рис. 1 відображає послідовні та умовні етапи, через які проходить кожне завдання, починаючи з моменту його створення до моменту виведення з системи.

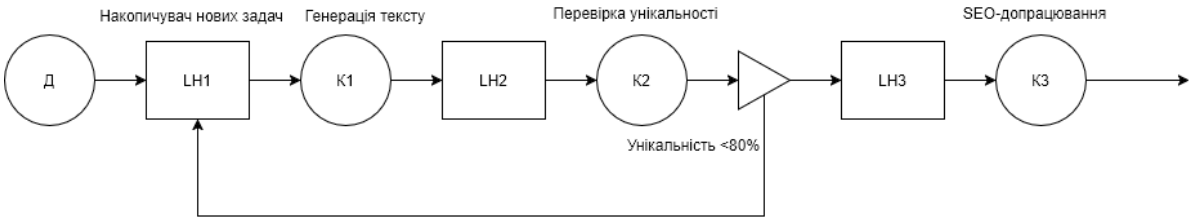

Рисунок 1 – Q-схема моделюючого алгоритму

На основі побудованої Q-схеми був розроблений асинхронно-циклічний алгоритм , який враховує специфіку високонавантаженого середовища та передбачає послідовну взаємодію між ключовими підсистемами: генерацією, перевіркою унікальності, унікалізацією, а також SEO-допрацюванням тексту.

У випадку, якщо унікальність є недостатньою (менше 80%), завдання переходить у фазу повторної генерації із залученням додаткових інструкцій для підвищення унікальності

Алгоритм побудований у вигляді циклу, який імітує роботу асинхронної обчислювальної системи. Центральним елементом цієї структури є визначення моменту наступу події, що відповідає за перехід між етапами обробки в залежності від черг та станів обслуговування.

Система автоматизованої генерації SEO-оптимізованого контенту реалізована в рамках архітектурної моделі Model–View–Controller (MVC), що забезпечує чітке розділення логіки: моделі відповідають за доступ до даних, контролери - за бізнес-процеси та маршрутизацію, а представлення - за інтерфейс користувача.

Окрім основної функціональності, пов’язаної з моделюванням черг та асинхронною генерацією контенту, у межах системи реалізований CRM-компонент, який забезпечує:

- автентифікацію користувачів;
- можливість створення проєктів і розділення згенерованих текстів за ними;
- налаштування параметрів генерації, зокрема вибір асистентів;
- зручний інтерфейс для перегляду, фільтрації, редагування та завантаження згенерованих результатів.

CRM частина реалізована у вигляді класичних сторінок з контролерами, кожен з яких відповідає за свою частину логіки. Наприклад, GenerateController обробляє всі операції, пов’язані з чергами задач, їх запуском, зміною станів і логуванням. Кожен такий контролер працює з відповідними моделями та викликає представлення для взаємодії з користувачем. Механізм обробки задач побудований на зберіганні даних у реляційній базі даних. Кожна задача має статус (is_processing), що вказує на її стан у життєвому циклі: очікування, обробка, завершення, помилка тощо. Статуси відповідають етапам Q-схеми та дозволяють відслідковувати динаміку виконання. Кожна задача інкапсульована в окремий CLI-процес, що запускається через системну команду. Всередині процесу реалізується повна логіка: запит до API генерації, обробка відповіді, оновлення даних у базі, перевірка унікальності та SEO, зміна статусу задачі.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Для управління чергою задач створений фоновий процес-модератор, що виконує роль диспетчера. Він постійно сканує базу на наявність задач зі статусом очікування (is_processing = 1) і приймає рішення про запуск нових CLI-процесів. При цьому враховується максимальне допустиме навантаження: не більше 30 задач одночасно на кожен сервіс генерації. Диспетчер перевіряє наявність задач у черзі, щоб визначити, чи є завдання, які потребують обробки. Одночасно контролюється кількість активних задач, звертаючи увагу на стан is_processing, який не повинен перевищувати значення 2. Перед запуском кожної задачі він змінює її статус, що дозволяє відстежувати поточний стан виконання. Для обробки команд диспетчер ініціалізує CLI-процес, забезпечуючи коректну взаємодію з системою. Також він реалізує захист від дублювання завдань за допомогою файлового блокування. Якщо ліміт досягнутий, система очікує 10 секунд (sleep(10)), після чого повторно перевіряє стан черги. Завдяки цьому досягається баланс між швидкістю обробки та стабільністю роботи зовнішніх сервісів (API генерації, уніkалізації тощо).

Система була розгорнута на виділеному сервері з характеристиками, що значно перевищують мінімальні вимоги до неї. Сервер оснащено процесором Intel(R) Xeon(R) E-2146G з тактовою частотою 3.50 GHz та 12 ядрами, а також оперативною пам’яттю обсягом 32 ГБ, з яких у середньому використовувалося близько 11.5 ГБ під час активної роботи системи.

У процесі тестування та реального використання не було зафіксовано жодного суттєвого навантаження на ресурси системи. Це пояснюється тим, що основні обчислювальні задачі реалізовані в рамках окремої системи черг, яка функціонує незалежно від основного інтерфейсу. Черги не створюють зайвих фонових навантажень і дозволяють ефективно розподіляти задачі в часі, що запобігає піковим навантаженням.

З архітектурної точки зору також було вжито рішень, спрямованих на оптимізацію використання ресурсів. У якості фреймворку для побудови веб-застосунку використано CodeIgniter, який вирізняється високою продуктивністю та мінімальним споживанням пам’яті. Для взаємодії з базою даних замість повноцінного ORM було використано Query Builder з прямими SQL-запитами до MySQL. Це дозволило зменшити кількість звернень до бази та пришвидшити обробку кожного запиту за рахунок уникнення надлишкової абстракції.

На рис. 2 показаний час, витрачений на обробку кожної з 100 заявок. Візуально видно, що зростання часу відбувається поступово до певного порогу, після чого він стабілізується. Це пояснюється тим, що перші запити проходять без черги в Copyleaks, але вже після кількох десятків запитів починається накопичення запитів у черзі самого сервісу. У результаті час перевірки починає зростати, але після стабілізації черги утворюється приблизно постійний середній час обробки. Це дозволяє зробити висновок, що навіть з урахуванням уповільнення, система в цілому стабільно виходить на певний темп виконання.

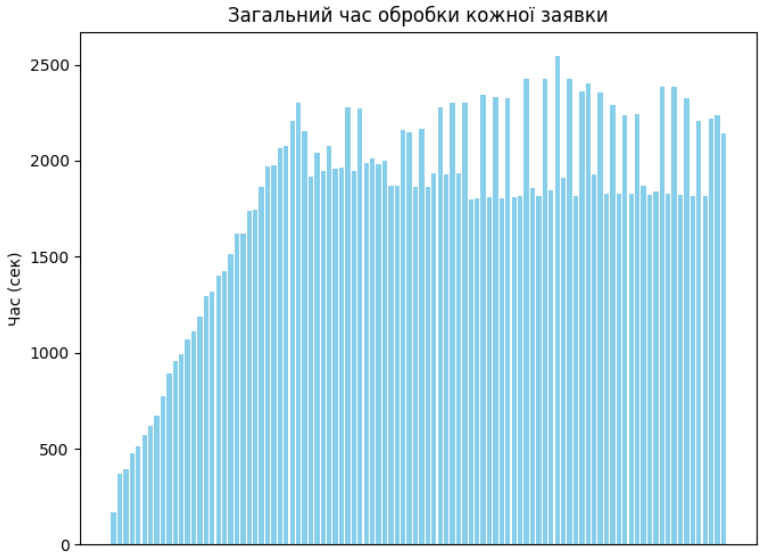

Рисунок 2 – Графік часу обробки всіх заявок

На рис. 3 показаний розклад часу по трьох основних етапах: генерація, унікальність та SEO. Найбільший обсяг часу припадає саме на етап перевірки унікальності (помаранчева зона). Генерація займає невеликий сегмент (синій), а SEO - взагалі мізерну частку (зелений). Це підтверджує гіпотезу, що основним вузьким місцем системи є саме взаємодія з зовнішнім сервісом Copyleaks. Варто зазначити, що всі етапи однаково запускалися в асинхронному режимі, тому кожне стрибкоподібне зростання часу є прямим наслідком чергового затору на етапі унікальності.

На рис. 4 кругова діаграма, яка показує частку кожного з етапів у загальному часі. Більше 96% усієї тривалості припадає саме на етап перевірки унікальності. Це демонструє критичність даного компонента у загальній архітектурі. Таке відображення дає змогу наочно ідентифікувати "вузьке місце" системи та визначити, в якому напрямку слід проводити оптимізацію.

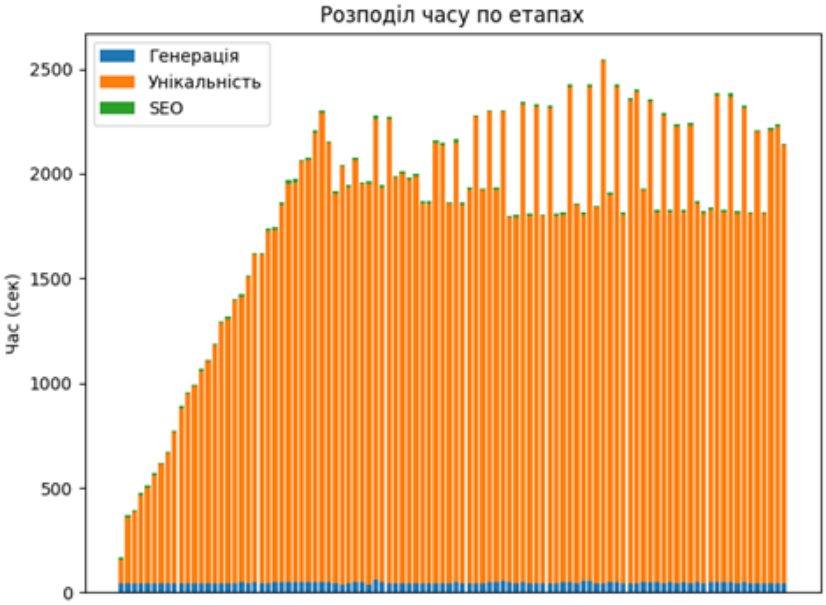

Рисунок 3 – Графік розподілу часу по етапах

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

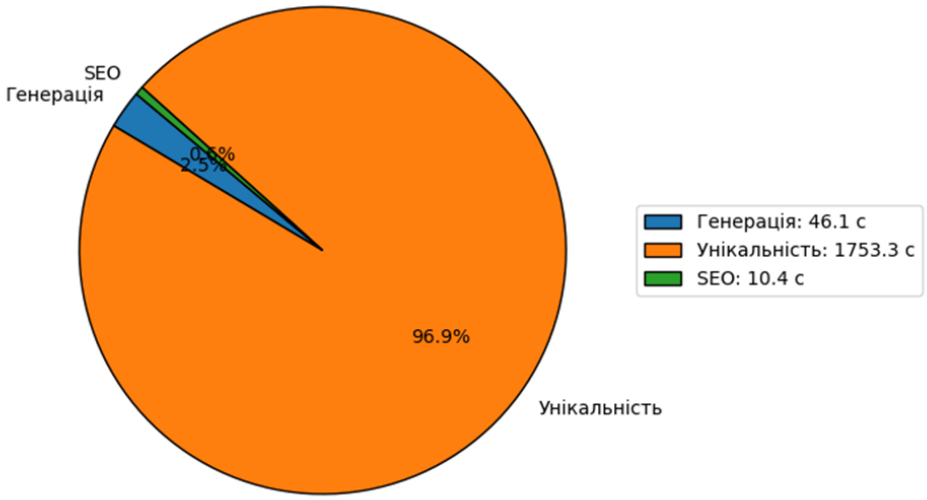

Рисунок 4 – Розподілення затраченого часу між фазами

Висновки. Створена високопродуктивна автономна система, здатна генерувати тексти описів товарів, перевіряти їх на унікальність та відповідність SEO-вимогам у режимі реального часу шляхом побудови програмного застосунку на базі веб-фреймворку CodeIgniter із використанням моделі асинхронної черги, що дозволяє одночасно обробляти велику кількість заявок. Завдяки використанню асинхронного моделюючого алгоритму Q-схеми для чергової обробки вдалося досягти оптимального балансу між швидкістю генерації, навантаженням на систему та якістю кінцевого результату.

ЛІТЕРАТУРА / REFERENCES

1. Internet stattia “What Is SEO – Search Engine Optimization” [URL: https://searchengineland.com/guide/what-is-seo]
2. Rozshyrennia AI Content Creator – Easy ChatGPT powered article generator [URL: https://wordpress.com/plugins/ai-content-creator]
3. Platforma SE Ranking [URL: https://seranking.com/ua/ai-writer.html]Received 05.01.2026.

Received 05.01.2026.
Accepted 07.01.2026.

Modeling a queue system for generation of seo-optimized texts in a highly loaded environment

The work is dedicated to the development of a system for automated generation of SEO-optimized texts with subsequent modeling of the queue system to reduce the costs of creating high-quality content, simplify the process of generating texts and ensure their compliance with modern SEO requirements. Research on mechanisms of interaction with AI services to optimize the process of generating texts under conditions of restrictions on the number of requests and data processing speed. Studies of existing solutions for automatic content generation were conducted, including the consideration of services based on artificial intelligence, such as ChatGPT. As a result, limitations of existing approaches were identified, mainly related to their insufficient scalability or API limitations. A Q-scheme was synthesized to solve

a specific problem, an asynchronous modeling algorithm was adapted, and a modeling of the queue system of SEO-optimized texts was carried out. Improved algorithms for interacting with AI services ensure stable and efficient operation, allowing not only to reduce the cost of the content filling process, but also to make it as efficient as possible, while maintaining the quality of the texts at a level that is not inferior to the work of professional copywriters.

Волковський Олег Степанович – к.т.н., доцент кафедри комп'ютерних наук та інформаційних технологій Дніпровського національного університету ім. Олеся Гончара.
ORCID: https://orcid.org/0000-0002-8635-6571
Нікішина Олександра Юріївна – ст. викладач кафедри комп'ютерних наук та інформаційних технологій Дніпровського національного університету ім. Олеся Гончара.
ORCID: https://orcid.org/0009-0007-2486-1859

Volkovskyi Oleh Stepanovych – Candidate of Technical Sciences, Associate Professor of the Department of Computer Science and Information Technologies, Oles Honchar Dnipro National University.
ORCID: https://orcid.org/0000-0002-8635-6571
Nikishyna Oleksandra Yuriivna – Senior Lecturer of the Department of Computer Science and Information Technologies, Oles Honchar Dnipro National University.
ORCID: https://orcid.org/0009-0007-2486-1859

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)