УДК 004.415.2:004.8:004.912

ГІБРИДНА АРХІТЕКТУРА ВЕБЗАСТОСУНКУ ДЛЯ ОПИТУВАНЬ З ШІ-АНАЛІЗОМ ВІДКРИТИХ ВІДПОВІДЕЙ

Мельник В.А.[1]; Крісілов В.А., д.т.н., проф.[2]
[1,2] Національний університет «Одеська політехніка»
[1,2] Україна, Одеса
[1] 7982174@stud.op.edu.ua; [2] krisilov@op.edu.ua

Анотація. *Розглянуто проєктування вебзастосунку для організації опитувань з ШІ-аналізом відкритих відповідей на основі Firebase та Gemini API. Визначено вимоги, проаналізовано ризик хибного підсумку методом дерева відмов, порівняно чотири архітектури й обрано гібридну: локальну кластеризацію та хмарне формулювання підсумку. Описано плани реалізації й тестування.*
Ключові слова: *опитування; відкриті відповіді; мовна модель; Firebase; гібридна архітектура.*

Вступ. Відкриті запитання в опитуваннях містять найцінніші відомості, але їх ручне узагальнення забирає багато часу. Мовні моделі здатні автоматизувати кодування таких відповідей: у дослідженні [1] найкраща модель досягла точності 80,9 % проти 88,6 % у другого експерта-людини. Водночас моделі схильні до галюцинацій під час реферування [2], а переконливий підсумок легко прийняти без перевірки. SurveyMonkey [3] і Qualtrics [4] пропонують ШІ-аналіз відкритих відповідей (теми, тональність), проте за наявними описами він залежить від окремих модулів чи дозволів і не передбачає підсумку з прив’язкою кожної теми до відповідей-доказів.

Мета роботи. Проєктування вебзастосунку для організації опитувань, що формує підсумок відкритих відповідей, де кожна тема підтверджена реальними відповідями, та обґрунтувати його архітектуру.

Основна частина. Систему складають веб-клієнт, Firebase Authentication, Cloud Firestore, модуль статистики, сервіс аналізу на FastAPI, ШІ-аналізатор (Gemini API) і резервний аналіз ключових слів. Цілі структуровано за чотирма рівнями [5]: організаційні, провідні індикатори, результати для користувачів, властивості моделі. Ключова вимога: для відкритого запитання з ≥ 10 відповідей система подає підсумок до 7 тем із частками, кожна з яких підтверджена реальними відповідями. Цільові показники: macro-F1 $\geq 0{,}75$ на 200 розмічених вручну відповідях, не більше 5 % непідтверджених тверджень, затримка ≤ 15 с.

Основним ризиком обрано спотворення підсумку: вигадані чи пропущені теми, хибні частки. Дерево відмов [5] має десять базових подій і сім мінімальних перерізів. Запропоновано чотири системні засоби поза моделлю: верифікація підсумку (теми й цитати мають існувати у відповідях, а частки рахує окремий модуль); ізоляція відповідей від інструкцій проти ін’єкцій [6]; контроль повноти входу (дедуплікація, лічильник «оброблено k із N»); людський нагляд із фіксованою версією моделі та резервом.

Для ШІ-компонента порівняно чотири архітектури: клієнтську (виклик Gemini API з браузера), хмарну з локальним backend, локальну на scikit-learn без мовної моделі та гібридну. У табл. 1 наведено їх якісне порівняння.

Таблиця 1 – Порівняння архітектур-кандидатів

Архітектура	Хибні теми	Конфіденційність	Складність
А1 Клієнтська	без перевірки	низька	низька
А2 Хмарна	галюцинації	низька: усі тексти	низька
А3 Локальна ML	пропуски синонімів	найвища	середня
А4 Гібридна	прив'язані до груп	середня: представники	найвища

Обрано гібридну архітектуру: сервіс видаляє порожні тексти й дублікати, групує відповіді кластеризацією TF–IDF, передає Gemini API лише представників груп із JSON-схемою, перевіряє результат і сам обчислює частки тем. Це зменшує витрати безкоштовної квоти та ризик вигаданих тем, бо кожна тема спирається на реальну групу відповідей; за недоступності моделі система перемикається на ключові слова. Ціною є найвища складність розробки. Також розроблено схему Firestore, діаграму послідовності сценарію аналізу, макети чотирьох екранів і діаграму класів. Заплановано реалізацію прототипу, розмічений набір із 200 відповідей, вимірювання показників точності й вірності, перевірку стійкості до ін'єкцій і порівняння з хмарною архітектурою.

Висновки. Визначено цілі, компоненти, вимоги й ризик застосунку для опитувань з ШІ-аналізом, запропоновано чотири системні засоби пом'якшення ризику, обґрунтовано гібридну архітектуру й підготовлено проєктні моделі системи. Досягнення цільових показників буде перевірено на прототипі.

Використання інструментів ШІ. Під час підготовки тез використано Claude Sonnet 5.5 (Anthropic) для перевірки пунктуації, граматики та правопису українського тексту, а також для перекладу анотації англійською мовою. Перевірку змісту та остаточне редагування виконано автором. Gemini API є складником описаної системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Mellon, J., Bailey, J., Scott, R., Breckwoldt, J., Miori, M., & Schmedeman, P. (2024). Do AIs know what the most important issue is? Using language models to code open-text social survey responses at scale. Research & Politics, 11(1). https://doi.org/10.1177/20531680241231468

[2] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., … Fung, P. (2023). Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12), Article 248. https://doi.org/10.1145/3571730

[3] SurveyMonkey. (n.d.). Get quick insights with Analyze with AI. https://help.surveymonkey.com/en/surveymonkey/analyze/analyze-with-ai/

[4] Qualtrics. (n.d.). Text iQ functionality. https://www.qualtrics.com/support/survey-platform/data-and-analysis-module/text-iq/text-iq-functionality/

[5] Kästner, C. (2025). Machine learning in production: From models to products. Cambridge, MA: MIT Press.

[6] OWASP GenAI Security Project. (2025). OWASP top 10 for LLM applications 2025. https://genai.owasp.org/

Melnyk V.A., Krisilov V.A.

Hybrid architecture of a web application for surveys with AI-based analysis of open-ended responses

Abstract. *The design of a web application for organizing surveys with AI-based analysis of open-ended responses, built on Firebase and the Gemini API, is considered. Requirements are defined, the risk of a misleading summary is analyzed with fault tree analysis, four architectures are compared, and a hybrid one is chosen: local clustering plus cloud-based summary generation. Implementation and testing plans are described.*

Keywords: *surveys; open-ended responses; language model; Firebase; hybrid architecture.*