УДК 004.4

Добрушин Ю.В., магістрант
Віктор А.С., PhD, доцент
Державний університет інформаційно-комунікаційних технологій

МЕТОДОЛОГІЧНІ ЗАСАДИ АВТОМАТИЗАЦІЇ ІНТЕГРАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ СЕМАНТИЧНОГО АНАЛІЗУ

Постановка задачі. Сучасна IT-інфраструктура характеризується гетерогенністю та стрімким зростанням кількості програмних сервісів, що взаємодіють через API. Процес їх інтеграції залишається значною мірою ручним, трудомістким та схильним до помилок.

Ключовою проблемою є семантичний розрив, коли однакові за змістом сутності (наприклад, "ідентифікатор клієнта") мають різні синтаксичні назви в системах (customerId, client_id, user_number). Існуючі автоматизовані підходи, що базуються на синтаксичному порівнянні або простих онтологіях, демонструють недостатню точність та гнучкість. Це зумовлює необхідність розробки нового методу автоматизованого зіставлення (mapping) елементів API, який би спирався на глибоке розуміння їх семантичного значення.

Мета дослідження. Метою даної роботи є розробка та теоретичне обґрунтування методу автоматизованого семантичного зіставлення компонентів програмних інтерфейсів. Метод має базуватися на застосуванні сучасних моделей обробки природної мови (NLP), зокрема архітектури Transformer, для аналізу текстових описів API та створення їх векторних представлень (embeddings). Основна задача – довести, що запропонований підхід дозволяє досягти суттєво вищої точності зіставлення порівняно з існуючими аналогами.

Результати дослідження. В рамках дослідження запропоновано комплексний метод, що складається з трьох основних етапів.

Перший етап – збір та передобробка даних. Метод передбачає автоматичний парсинг стандартних специфікацій API, таких як OpenAPI (Swagger), для вилучення ключової інформації: назв кінцевих точок (endpoints), параметрів запитів, полів у тілі запиту/відповіді та їх текстових описів [1].

Другий етап – семантичне векторизування. На цьому етапі вилучені текстові описи кожного елемента API подаються на вхід попередньо навченої мовної моделі (наприклад, BERT або RoBERTa) [2]. Модель перетворює текст у багатовимірний числовий вектор (embedding), який кодує семантичне значення опису в контексті. Це є ключовим кроком,

що дозволяє перейти від синтаксичного порівняння рядків до порівняння змісту.

Третій етап – зіставлення на основі подібності. Для знаходження відповідностей між елементами двох різних API обчислюється метрика подібності між їх векторними представленнями, найчастіше – косинусна подібність [3]. Пари елементів з найвищим показником подібності вважаються семантичними відповідниками.

Згідно з аналізом сучасних досліджень, методи на основі Transformer-моделей демонструють точність (F1-score) у задачах семантичного зіставлення в діапазоні 85-94%. Це являє собою значне покращення порівняно з традиційними синтаксичними підходами (наприклад, на основі відстані Левенштейна або n-грам), ефективність яких зазвичай не перевищує 50-70% [4].

Таким чином, запропонований метод дозволяє не просто покращити, а кардинально підвищити якість автоматичної інтеграції, знижуючи кількість помилок, що виникають через семантичну неоднозначність.

Висновки та перспективи. Запропонований метод створює теоретичну основу для розробки нового покоління інтелектуальних систем інтеграції ПЗ. Його головна перевага полягає у здатності розуміти функціональне призначення елементів API, а не лише їх назви. Це дозволяє значно прискорити процес розробки, знизити кількість помилок та зменшити вимоги до кваліфікації інтеграторів.

Подальші дослідження можуть бути спрямовані на практичну реалізацію методу у вигляді програмного прототипу, його тестування на широкому наборі реальних API та розширення для аналізу не лише текстових описів, а й прикладів коду з документації для збільшення успішних варіантів зіставлення endpoints при інтеграції.

Список використаних джерел:

1. OpenAPI Initiative. (2021). OpenAPI Specification v3.0.3. Retrieved from: https://spec.openapis.org/oas/v3.0.3

2. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, 4171–4186.

3. Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing (3rd ed. draft). Prentice Hall.

4. Mudgal, S., et al. (2018). Deep learning for entity matching: A design space exploration. Proceedings of the 2018 International Conference on Management of Data (SIGMOD '18), 19–34.