

ЗАСТОСУВАННЯ ЛОКАЛЬНОЇ LLM-МОДЕЛІ ДЛЯ СТАНДАРТИЗАЦІЇ НАЗВ МЕТИЗІВ

Вінницький національний технічний університет

Анотація

Дослідження присвячене автоматизації стандартизації номенклатури металевих виробів за допомогою локальних LLM, що забезпечує високу точність роботи зі специфічними даними. Деталізовано технічну реалізацію системи: від конфігурації Python-скриптів для взаємодії з Lm studio через API до специфікації JSON-структур та архітектури збереження обробленої інформації в базі.

Ключові слова: локальна LLM, мовна модель, метизи, стандартизація, стандартизація назв.

Abstract

The research is devoted to the automation of standardization of the nomenclature of metal products using local LLMs, which ensures high accuracy of work with specific data. The technical implementation of the system is detailed: from the configuration of Python scripts for interaction with Lm studio via API to the specification of JSON structures and the architecture of storing processed information in the database.

Keywords: local LLM, language model, hardware, standardization, name standardization.

Вступ

У сучасних умовах розвитку промислових інформаційних систем (ERP, CAD/PDM) постає гостра потреба в уніфікації технічних назв виробів. Зокрема, завдання стандартизації метизів — таких як болти, гайки та гвинти — вимагає точного багатомовного перекладу для інтеграції в міжнародні ланцюги постачання. Традиційні системи нейронного машинного перекладу (NMT), попри їхню ефективність, часто не враховують специфіку технічної лексики та вимоги форматних стандартів DIN, ISO чи ДСТУ [1].

Натомість останні досягнення у сфері великих мовних моделей (LLM) відкривають нові можливості для одночасної стандартизації та перекладу в межах єдиного інструменту. Для забезпечення термінологічної стійкості доцільно використовувати підхід LLM-BT, що базується на механізмі зворотного перекладу (back-translation) для верифікації термінів [2]. Окремим аспектом є впровадження локальних LLM-серверів, що дозволяє підприємствам зберігати автономність та безпеку даних, уникаючи залежності від хмарних API.

У даній роботі запропоновано архітектуру автономної системи, побудовану на Python-скриптах, локальному сервері LM Studio та базі даних SQLite. Система реалізує повний цикл обробки: від формування специфічних підказок (prompt-engineering) до валідації результатів за схемою «стандарт → розмір → переклад». Такий підхід дозволяє не лише автоматизувати поповнення каталогів, а й кількісно оцінити коректність вихідних даних за допомогою розроблених метрик якості.

Реалізація поставленої задачі

Програмна реалізація системи виконана мовою Python та базується на автоматизованому циклі обробки даних. Спеціальний скрипт здійснює вибірку записів із локальної бази даних, генерує для кожного з них структурований запит (prompt) і передає його через POST-інтерфейс до сервера LM Studio. Після генерації відповіді моделлю, Python-клієнт виконує парсинг отриманого JSON-файла,

вилучає стандартизовані технічні назви та переклади, після чого автоматично оновлює відповідні поля в базі даних[3].

Для ілюстрації взаємодії компонентів на рис. 1 наведено діаграму компонентів та розгортання (Component / Deployment Diagram). Вона візуалізує архітектуру рішення, описуючи потоки даних між Python-клієнтом, сервером LM Studio та локальною моделлю в межах єдиної обчислювальної системи.

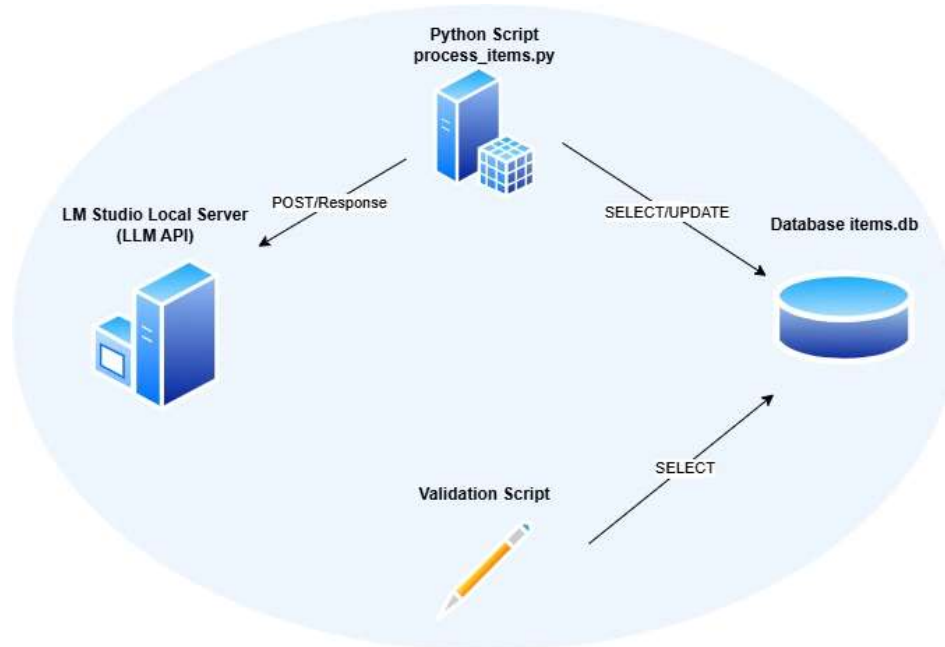


Рис. 1. Схема підключення (Component / Deployment Diagram)

Навчання LLM на доменних даних

Адаптація великих мовних моделей (LLM) до вузькоспеціалізованих доменів є критично важливою для коректної обробки технічної термінології та стандартів. Хоча універсальні моделі мають високий ступінь загальних знань, вони часто припускаються помилок у роботі зі специфічними форматами. Для подолання цього бар'єру зазвичай використовують три стратегії: RAG (для доступу до знань у реальному часі), PEFT (наприклад, LoRA) для ефективної адаптації та класичне fine-tuning на цільових датасетах (James Tang, 2025) [4-5].

Через високу ресурсомісткість повного перенавчання, у цьому дослідженні було обрано метод PEFT (LoRA), який дозволяє навчати модель на малих обсягах даних без втрати її базових можливостей. Експеримент проводився на моделі mistral-7b-instruct-v0.2 з використанням корпусу з понад 100 прикладів (prompt-completion), що містили шаблони стандартів DIN, ISO та ГОСТ. Такий підхід дозволив скоротити час донавчання до кількох годин на стандартних GPU та суттєво мінімізувати помилки у форматуванні назв метизів.

Для оцінки коректності роботи моделі було реалізовано алгоритм автоматичної верифікації. Він базується на сегментації оновленої назви (українською мовою) на структурні компоненти: назва, стандарт, номер стандарту, розмір та додаткові опції. Технічно цей процес реалізовано мовою Python з використанням бібліотеки re. Зокрема, за допомогою методу re.compile() було створено регулярні вирази, які дозволяють деконструювати рядок і порівняти виділені секції із заданим еталонним шаблоном.

Значна роль у проведеному дослідженні полягала у розробленні необхідних для донавчання моделі prompt-ів. Основна задача prompt – максимально роз'яснити моделі mistral-7b-instruct-v0.2 інструкцію по стандартизації назв метизів. Приклад prompt наведено на рис.2.

```
prompt = (
    f"Ти – модель для стандартизації назв метизів. Задача стандартизувати назву метиза і зробити переклад її на 2 мови. "
    f"Вхід(приклад): 'Болт оцинкований М10х120 DIN 933'. "
    f"Вихід(приклад): ukr_name: 'Болт DIN 933 М10х120 оцинкований'; eng_name: 'Bolt DIN 933 M10x120 galvanized'. "
    f"В конкретному прикладі Болт - Назва виробу(може бути Гайка, Шайба тощо), DIN 933 - Стандарт(може бути ГОСТ та ISO), М10х120 - розмір виробу, оцинкований - інші додатковий параметр"
    f"Тепер стандартизуй: '{raw_name}' у форматі '<назва виробу> <стандарт> <розмір> <додаткові параметри при наявності>'. "
    f"Забезпеч, щоб переклади були точними: "
    f"- ukr_name українською, "
    f"- eng_name англійською. "
    f"Відповідь у JSON без додаткових полів, одразу ukr_name, eng_name."
)
```

Рис. 2. Приклад prompt

На базі розроблених промптів було здійснено апробацію моделі із залученням тестової вибірки зі 100 записів, попередньо внесених до бази даних items.db. Результати обробки для 5 характерних прикладів, що ілюструють якість стандартизації, систематизовано в Таблиці 1.

Таблиця 1

Виконання стандартизації через prompt

Назва метиза	Оновлений запис ukr_name	Оновлений запис eng_name	Час обробки запису
Болт М12х85 DIN 603	Болт DIN 603 М12х85	Bolt DIN 603 M12x85	10.14 секунд
Гайка М12 DIN 934	Гайка DIN 934 М12	Screw DIN 934 M12	8.99 секунд
Шпилька М8х80 DIN 976	Шпилька DIN 976 М8х80	Screw DIN 976 M8x80	9.30 секунд
Болт М10х70 DIN 931	Болт DIN 931 М10х70	Bolt DIN 931 M10x70	12.21 секунд
Шайба 10 DIN 125	Шайба DIN 125 10	Washer DIN 125 M10	10.04 секунд

Загальні часові витрати на обробку масиву зі 100 записів склали 987,59 секунди (приблизно 16 хвилин 45 секунд). Результати верифікації за допомогою спеціалізованого промпту продемонстрували виняткову точність: при загальній кількості 100 ітерацій не було виявлено жодного некоректного запису (0,0% похибки)[3].

Для об’єктивної оцінки ефективності розробленого методу прямої екстракції даних через API-запити, планується проведення порівняльного тестування з використанням галузевого стандарту структурування даних — бібліотеки Instructor. На відміну від базового алгоритму, що фокусується на гнучкості промпт-інжинірингу та швидкості обробки в середовищі SQLite, метод Instructor впроваджує рівень суворої валідації на базі Pydantic-схем. Це дозволяє забезпечити типізацію витягнутих параметрів метизів (категорія, діаметр, довжина) та автоматизувати обробку виключень у разі логічних помилок моделі. Поточний етап дослідження включає аналіз продуктивності обох підходів на ідентичному датасеті з 100 одиниць номенклатури, що дає змогу виявити баланс між швидкістю генерації та надійністю отриманих структурованих даних у локальних системах на базі Mistral-7B.

Висновки

У результаті дослідження науково обґрунтовано ефективність застосування локальних великих мовних моделей (LLM) для автоматизованої нормалізації та багатомовного перекладу номенклатури метизної продукції. Експериментально доведено, що цільове донавчання моделі mistral-7b-instruct на спеціалізованому доменному корпусі в поєднанні з оптимізованим промпт-інжинірингом (промпт №2) забезпечує 100% точність обробки даних. Це підтверджує спроможність запропонованого підходу вирішувати складні завдання технічної стандартизації в автономному режимі.

Отже, розроблена методика інтеграції локальних LLM у промислові інформаційні системи дозволяє суттєво підвищити достовірність технічних даних. Запропоноване рішення може слугувати архітектурною основою для впровадження інтелектуальних модулів автоматизації в аналітичні та виробничі комплекси.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Machine translation in the era of large language models: a survey of historical and emerging problems / d. Ataman et al. *Information*. 2025. Vol. 16, no. 9. P. 723. URL: <https://doi.org/10.3390/info16090723> (дата звернення: 12.02.2026).
2. Llm-bt-terms: back-translation as a framework for terminology standardization and dynamic semantic embedding. *Arxiv.org*. URL: <https://arxiv.org/abs/2506.08174> (дата звернення: 12.02.2026).
3. Старжинський В., Бісікало О. Використання локальних llm-моделей для стандартизації та багатомовного перекладу технічних назв виробів. *Measuring and computing devices in technological processes*. 2025. № 4. С. 407–415. URL: <https://doi.org/10.31891/2219-9365-2025-84-49> (дата звернення: 12.02.2026).
4. Shah k. How to build domain specific llms?. Best artificial intelligence blogs. Url: <https://www.projectpro.io/article/domain-specific-llms/1111> (дата звернення: 12.02.2026).
5. Tang j. Fine-tuning llms for specific domains: why, how, and what to consider. Medium. Url: <https://medium.com/@jamestang/fine-tuning-llms-for-specific-domains-why-how-and-what-to-consider-8b2fd5781615> (дата звернення: 12.02.2026).

Старжинський Валерій Юрійович – аспірант групи 126-23а, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: 3372292@gmail.com

Бісікало Олег Володимирович — доктор технічних наук, професор, завідувач кафедри Автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м. Вінниця.

Valeriy Starzhinskii - — Department of intelligent information technologies and automation, Vinnytsia National Technical University, Vinnytsia, e-mail: 3372292@gmail.com

Bisikalo Oleh — Doctor of Technical Sciences, Professor, Head of the Department of Automation and Intelligent Information Technologies, Vinnytsia National Technical University, Vinnytsia.