Software Requirements Specification

for

AITrans

**Version 1.0 approved**

**Prepared by Oleksii Trofimenko**

**Kharkiv National University of Radioelectronics**

**01.06.2025**

# 1. Постановка мети

Метою цієї роботи є аналіз та порівняння ефективності й продуктивності великої мовної моделі LLaMA при застосуванні спеціалізованих методів інтеграції глосарію та прогресивного оновлення контексту в автоматизованому перекладі художніх текстів. Основну увагу зосереджено на моделі LLaMA3.3-70B-Instruct, а також на розроблених підходах інтеграції глосарію й оновлення контексту. У результаті дослідження було створено програмний продукт, який забезпечує автоматичний переклад художньої літератури українською, французькою та німецькою мовами, враховуючи стилістичні та термінологічні особливості оригіналу.

У процесі реалізації проєкту використовувались сучасні інформаційні технології. Основну частину реалізації становить програмний код, створений за допомогою мови Python та спеціалізованих бібліотек для обробки природної мови (nltk, BookNLP), а також великих мовних моделей (huggingface-hub, sentence-transformers). Крім того, було використано середовище розробки conda для забезпечення стабільності та керування залежностями. Веб-інтерфейс був розроблений на основі веб-фреймворку Flask із залученням HTML, CSS та JavaScript для максимальної зручності взаємодії з користувачами.

Для оцінювання ефективності розробленої системи перекладу застосовувалися різноманітні інструменти та метрики, такі як BLEU, CHRF++ та CoCon, а також додаткові семантичні показники для оцінювання термінологічної узгодженості. Архітектура системи побудована за модульним принципом, що забезпечує чітке розмежування функцій між ядром для обробки природної мови, модулем інтеграції глосарію і користувацьким веб-інтерфейсом.

# 2. Опис програмної системи

Розроблена програмна система реалізує процес автоматизованого перекладу художніх текстів із залученням сучасних мовних моделей та методів обробки природної мови. Архітектура проєкту побудована як клієнт-серверний веб-додаток, що дозволяє забезпечити інтерактивну взаємодію з користувачем, запускати перекладні сценарії та візуалізувати результати роботи системи в реальному часі. Система також виконує роль експериментального середовища для вивчення впливу різних стратегій подання контексту та глосаріїв на якість перекладу.

Клієнтська частина системи створена з використанням веб-фреймворку Flask, що дозволило побудувати легкий, модульний і адаптивний інтерфейс. Завдяки використанню HTML, CSS та JavaScript, інтерфейс забезпечує зручну навігацію, підтримує динамічне оновлення вмісту та дозволяє запускати запити до моделі безпосередньо з браузера. Усі запити надсилаються до серверної частини, яка відповідає за основну обчислювальну логіку.

Серверна частина реалізована на Python та включає модулі для обробки вхідного тексту, генерації глосарію, реалізації механізму інтеграції контексту, а також взаємодії з LLaMA-моделлю. Обробка текстів здійснюється із застосуванням бібліотек NLP, зокрема nltk і BookNLP, які дозволяють виокремити мовні структури, персонажів, діалоги та думки. Після обробки тексту система формує запит до LLaMA-моделі, інтегруючи попередньо отриманий глосарій та релевантні фрагменти контексту.

Отримані результати перекладу відображаються в інтерфейсі користувача разом із додатковою інформацією, такою як відповідність глосарію, попередній і наступний контекст, а також семантичні метрики. Таким чином, створене середовище дозволяє не лише проводити переклад, а й досліджувати вплив різних конфігурацій на якість фінального результату. Система орієнтована на розширення, що дозволяє інтегрувати нові моделі, додавати інші мови або змінювати способи обробки контексту без перебудови всієї архітектури.

# 3. Функціонал програмної системи

Функціонал програмної системи охоплює ключові компоненти процесу перекладу та взаємодії з глосаріями й текстами. Основні маршрути, реалізовані в системі, наведено нижче:

|  |  |  |
| --- | --- | --- |
| **Шлях** | **Тип** | **Призначення** |
| / | зовнішній | Створення глосарію у форматі JSON, відображення вікна статистики, збереження книги на сервері |
| /translation | зовнішній | Інтерфейс із полем для тексту, завантаженням і аналізом глосарію, чекбоксом передачі контексту, редагуванням полів глосарію |
| /start\_translation | внутрішній | Запуск процесу перекладу |
| /translation\_progress/<tid> | внутрішній | Інтерфейс перевірки семантичної подібності текстів, два текстові поля та кнопка для оцінювання |
| /compare | зовнішній | Обробка порівняння текстів на семантичну збіжність |
| /compare\_text | внутрішній | Створення глосарію через BookNLP та аналіз отриманих файлів |
| /create\_glossary | внутрішній | Видалення окремої сутності з глосарію |
| /analyze\_glossary | внутрішній | Аналіз глосарію через LLM для очищення і корекції термінів |
| /download\_glossary/<book\_id> | внутрішній | Завантаження глосарію певної книги у вигляді файлу |
| /process | внутрішній | Запуск обробки глосарію |
| /progress/<state\_id> | внутрішній | Відстеження прогресу обробки глосарію |

Уся взаємодія здійснюється через інтуїтивно зрозумілий веб-інтерфейс. Основна мета програмної системи — забезпечити можливість дослідження ефективності методів інтеграції контексту та глосарію в умовах, наближених до реального використання, при цьому не відволікаючись на складні сценарії керування доступами або багатокористувацьку логіку.

# 4. Основні обмеження

Серед загальних обмежень розробленої системи можна виокремити такі аспекти. По-перше, для функціонування веб-додатка необхідне стабільне підключення до інтернету, оскільки сервер обмінюються даними з провайдером на якому розгорнуто модель. По-друге, система не реалізує механізм автентифікації або авторизації, оскільки її основна мета полягає у дослідженні якості перекладу, а не в управлінні користувацькими сесіями. Окрім цього, система розрахована на послідовну роботу одного користувача без паралельного доступу або підтримки багатокористувацької взаємодії. Аналіз якості перекладу проводиться на обмеженому обсязі даних, що не охоплює великомасштабні сценарії чи багатотомні корпуси. Безпека переданих даних не була предметом дослідження і тому не реалізована на рівні повноцінного захисту.

# 5. Припущення та залежності

У роботі враховано низку припущень і залежностей, що впливають на коректну роботу системи. Передбачається, що користувач працює у середовищі з актуальною версією веб-браузера, який підтримує JavaScript та забезпечує обробку інтерфейсів HTML/CSS. Серверна частина потребує встановленого середовища Python, налаштованого середовища conda для різних модулів, доступу до необхідних бібліотек та стабільного з’єднання з інтернетом. Система передбачає базову налаштованість без складної конфігурації, однак потребує наявності певних технічних навичок у користувача при роботі з інтерфейсом і запуску процесів. Усі експерименти проводяться у середовищі з обмеженим обсягом ресурсів, тому результати можуть відрізнятись при масштабуванні.