УДК 316.4

А. О. Азарова
В. І. Тетерев

ПРОГРАМНО-АПАРАТНИЙ КОМПЛЕКС ДЛЯ ВИВЧЕННЯ УКРАЇНСЬКОЇ МОВИ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Вінницький національний технічний університет

Анотація

У роботі представлено процес побудови програмно-апаратного комплексу для вивчення української мови, створеного на основі технологій штучного інтелекту. Обґрунтовано актуальність впровадження інтерактивних мовних систем, що забезпечують автоматичне розпізнавання мовлення, аналіз тексту, генерацію персоналізованих навчальних траєкторій та голосову підтримку користувача. Розроблено архітектуру комплексу, алгоритми взаємодії та перспективи подальшого розвитку. Запропоноване рішення забезпечує індивідуалізацію навчання та підвищує ефективність опанування української мови.

Ключові слова: програмно-апаратний комплекс, штучний інтелект, навчальні системи, NLP, українська мова, розпізнавання мовлення.

Abstract

The study presents the process of building a software and hardware complex for learning the Ukrainian language, created on the basis of artificial intelligence technologies. The relevance of implementing interactive language systems that provide automatic speech recognition, text analysis, generation of personalized learning trajectories and user voice support is substantiated. The architecture of the complex, interaction algorithms and prospects for further development are developed. The proposed solution ensures individualization of learning and increases the efficiency of mastering the Ukrainian language.

Keywords: *software and hardware complex, artificial intelligence, NLP, educational systems, speech recognition, Ukrainian language.*

Сучасні тенденції розвитку мовних технологій вказують на швидке зростання інтересу до використання штучного інтелекту в освітніх системах, зокрема під час вивчення української мови. Опанування мови потребує регулярної практики, якісного зворотного зв’язку та адаптації навчального навантаження до індивідуальних особливостей кожного користувача. Ці умови неможливо забезпечити традиційними методами, якщо навчання відбувається самостійно або без участі викладача.

Сучасні системи вивчення мови і застосуванням інформаційних технологій не є ефективним строго формалізованим інструментом, спроможним ефективно аналізувати мовлення, виявляти помилки, формувати ґрунтовні рекомендації та пропонувати персоналізовані завдання. Отже, актуальним є розроблення відповідного програмно-апаратного комплексу, спрямованого на розв’язання цих викликів. Варто зауважити, що найбільш продуктивним підходом для створення такого комплексу є залучення можливостей сучасних нейромережевих моделей штучного інтелекту та доступного апаратного забезпечення.

Розглянемо запропонований авторами метод для створення програмно-апаратного комплексу. Для вирішення такого комплексного завдання було застосовувано системний підхід, що зумовлює модульний принцип структури комплексу, що дозволяє забезпечити його гнучкість, масштабованість і легкість супроводу.

Візуалізацію структури комплексу представимо на рис. 1. На схемі відображено всі основні підсистеми, включно з аудіообробкою, логікою аналізу, механізмами генерації завдань та збереженням результатів. Модулі мають ієрархічну взаємодію, що дозволяє ізольовано модифікувати будь-який компонент без впливу на інші елементи системи.

Загальна архітектура комплексу включає набір незалежних функціональних компонентів, які взаємодіють через внутрішні програмні інтерфейси без потреби у зовнішньому сервері. Кожен модуль виконує чітко окреслену задачу: розпізнавання мовлення, аналіз граматики, генерація завдань, синтез мовлення, управління базою даних, а також організація інтерфейсу взаємодії з користувачем.

Усі компоненти функціонують локально, у межах десктоп-додатку, без потреби в Інтернет-з'єднанні. Вибір архітектури на користь офлайн-реалізації обґрунтований необхідністю забезпечити приватність даних, швидкодію та незалежність від серверної інфраструктури. Уся логіка обробки мовлення, аналізу та навчання відбувається на стороні клієнта.

Програмно-апаратний комплекс має задовольняти таким ключовими вимогами до архітектури, як:

- повна автономність;
- модульність;
- використання відкритих технологій;

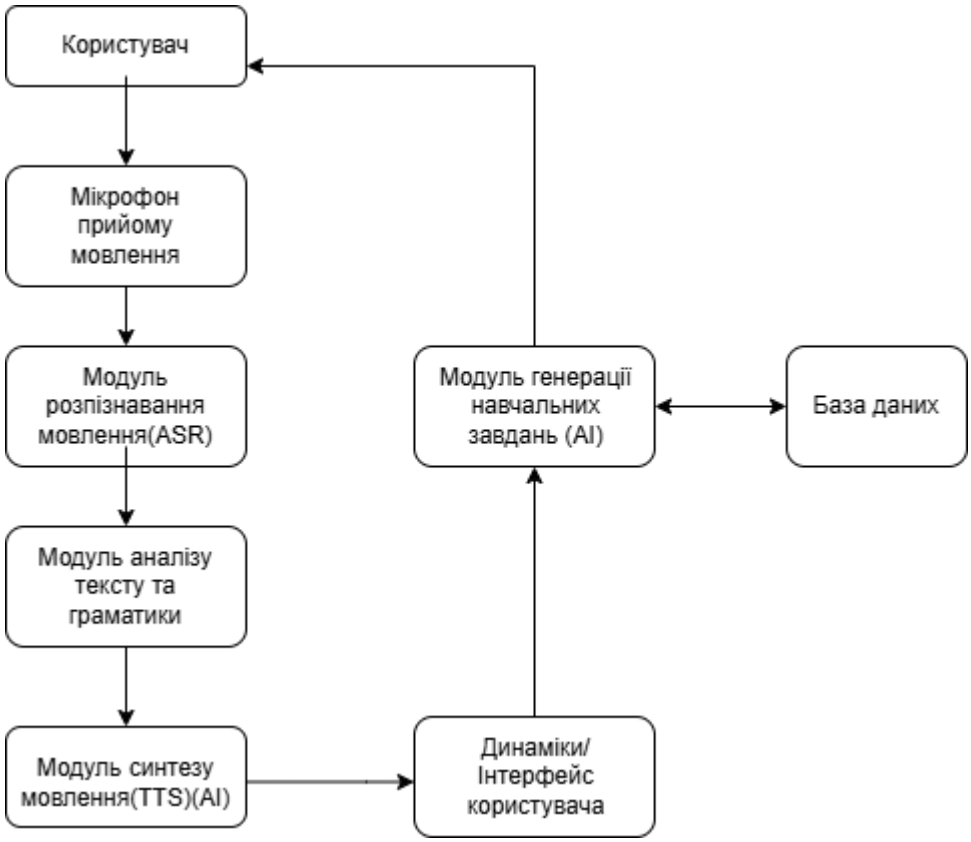

Рисунок 1 – Структурна схема програмно-апаратного комплексу для вивчення української мови з використанням штучного інтелекту

- простота інсталяції;
- підтримка користувацької персоналізації.

Запропонована архітектура дозволяє ефективно реалізувати всі освітні функції без складних зовнішніх залежностей. Вона оптимізована під операційну систему Windows, з урахуванням обмежених обчислювальних ресурсів звичайних користувачів.

Розглянемо роботу модулів, представлених на схемі (рис. 1).

Модуль розпізнавання мовлення забезпечує перетворення вхідного аудіопотоку на текстовий формат у реальному часі. У системі застосовується українська модель розпізнавання на основі бібліотеки Vosk, що використовує нейромережі з трансформерною архітектурою. Модель попередньо навчена на великому корпусі україномовних аудіо-даних і адаптована до побутової лексики та академічного стилю мовлення.

Процес розпізнавання мовлення опишемо так:

$$W = \arg\max\_W \left(P(X|W) \times P(W) \right) / P(X),$$

де X – послідовність акустичних ознак;

W – гіпотетична послідовність слів;

$P(X \mid W)$ – ймовірність появи сигналу для заданої фрази;

$P(W)$ – ймовірність побудови граматично правильної конструкції.

Для практичної реалізації застосовується декодер на основі пошуку за графом з інтеграцією мовної моделі у вигляді n-грам. Це дозволяє компенсувати акустичні помилки за рахунок граматичного контексту. Модуль підтримує адаптивну корекцію відповідно до словника користувача.

Оцінка ефективності проводиться через коефіцієнт точності Word Accuracy:

$$A = (N - (S + D + I)) / N \times 100\%,$$

де N – загальна кількість слів у еталонному тексті;

S – кількість замінених слів;

D – кількість пропущених слів;

I – кількість вставлених слів.

Середня затримка між завершенням мовлення і появою тексту не перевищує 0,5 с.

Усі розрахунки ведуться локально, без потреби в API або Інтернет-з'єднанні. Система чутлива до якості мікрофона, проте ефективно працює навіть у побутових умовах із середнім рівнем шуму.

Модуль аналізу тексту та граматики виконує перевірку синтаксичної, морфологічної та лексичної правильності тексту. Його реалізовано шляхом інтеграції мовної моделі Gemini 2.0 Flash, яка забезпечує багаторівневе оброблення українського мовлення, включно з аналізом частин мови, узгодженням слів, виявленням порушень побудови речень, стилістичних невідповідностей і пунктуаційних помилок. Ця модель здійснює контекстну інтерпретацію висловлювань, що дозволяє визначати не лише формальні порушення, а й семантичну природність конструкцій та доречність слововживання.

Архітектура модуля передбачає можливість розширення, зокрема підключення локальних граматичних правил або офлайн-аналізаторів, що забезпечить роботу без Інтернет-з'єднання у майбутніх версіях. Такий підхід поєднує переваги формального правило-орієнтованого аналізу та сучасних нейро-мережевих моделей, підвищуючи точність виявлення мовних відхилень і якість запропонованих виправлень. Результати аналізу оцінюються за допомогою коефіцієнта граматичної правильності:

$$G = (n_p / n) \times 100\%,$$

де n_p – кількість правильно вжитих граматичних конструкцій;

n – загальна кількість перевірених фрагментів.

Система також виконує категоризацію помилок за типами:

- орфографічні;
- граматичні (узгодження, відмінювання);
- синтаксичні (побудова речення);
- стилістичні (невідповідність контексту).

Усі операції виконуються локально, без викликів до зовнішніх API. Завдяки поєднанню правил і контекстної генерації вдається підвищити точність виявлення складних конструкцій, які часто залишаються непоміченими у традиційних інструментах перевірки.

Модуль генерації навчальних завдань забезпечує адаптивне створення вправ на основі результатів граматичного аналізу та історії помилок користувача. Побудова завдань здійснюється з використанням функції:

$$T = w1 \times G + w2 \times L + w3 \times P,$$

де G – рівень граматичної коректності;

L – поточний рівень володіння мовою;

P – набір попередніх помилок;

w1,w2,w3 – вагові коефіцієнти, що визначають внесок кожного параметра.

Ця функція дозволяє системі автоматично визначати складність завдання, його тип та необхідну кількість прикладів для тренування. Генерація реалізується шляхом поєднання логіки шаблонів (із фіксованими вправами) і контекстної генерації через Gemini 2.0 Flash.

Типи завдань включають:

- завершення речень із пропущеними словами;
- виправлення помилок у тексті;
- вибір правильної форми з кількох варіантів;
- переклад фраз рідною мовою;
- усне відтворення зразка після прослуховування.

Генератор адаптує контент відповідно до частоти та типу допущених помилок, рівня складності, тривалості взаємодії та індивідуальних параметрів користувача. Якщо певна конструкція трапляється часто з помилками, то система формує спеціальне завдання на її закріплення.

Оцінювання ефективності реалізується за допомогою коефіцієнта успішності:

$$S = (nc / n_t) \times 100\%,$$

де nc – кількість правильно виконаних завдань;

n_t – загальна кількість наданих завдань.

Якщо значення S перевищує 85%, система переходить до завдань складнішого рівня. У разі зниження цього показника – генерується додатковий блок вправ з повторенням.

Модуль синтезу мовлення реалізує зворотне перетворення тексту на аудіосигнал, необхідний для озвучування інструкцій, прикладів або результатів. Автори пропонують за основу обрати бібліотеку Coqui TTS, яка підтримує багатомовні моделі та дозволяє здійснювати генерацію мовлення природною інтонацією і тембром.

Для української мови використано одну з відкритих моделей, що базується на Tacotron 2 із WaveRNN. Модель адаптована до освітнього домену та працює локально. Параметри синтезу

(швидкість, інтонація, сила наголосу) можуть налаштовуватись користувачем через графічний інтерфейс. Завдяки цьому кожен приклад або підказка може бути озвучена з урахуванням індивідуального темпу навчання.

Процес синтезу відбувається в реальному часі із затримкою не більше 1–1.5 с після генерації тексту. Для збереження аудіофайлів застосовується формат WAV із можливістю подальшого експорту у MP3. Алгоритм працює в межах системи без потреби у зовнішніх сервісах або API.

Синтезоване мовлення використовується:

- для зачитування завдань,
- демонстрації правильної вимови,
- супроводу інтерфейсу голосовими підказками,
- формування адаптивного усного контролю (диктант, повторення).

Застосування Coqui TTS дозволяє забезпечити високий рівень якості звуку при помірному навантаженні на систему. Вбудовані моделі займають від 100 до 300 МБ пам'яті і сумісні з більшістю конфігурацій ПК. Робота модуля стабільна на платформі Windows і не вимагає інсталяції зовнішніх залежностей.

Модуль керування базою даних відповідає за зберігання всієї навчальної інформації, налаштувань користувача, результатів взаємодії, історії помилок та прогресу. Основою слугує вбудована СУБД SQLite, яка забезпечує швидкий доступ до даних без необхідності у встановленні окремого сервера.

Структура БД містить таблиці:

- користувачів (ідентифікатор, ім'я, рівень, мова);
- вправ (тип, складність, дата генерації, результат);
- граматичних помилок (тип, контекст, виправлення);
- аудіозаписів (файл, тривалість, зміст);
- конфігурацій (налаштування TTS, розмір шрифтів, мова інтерфейсу).

Усі запити виконуються через стандартну бібліотеку Python sqlite3.

Операції читання/запису оптимізовані для роботи з малими обсягами пам'яті.

База даних інтегрована з усіма модулями комплексу і слугує єдиним джерелом інформації про користувача. Такий підхід спрощує ведення статистики, дозволяє реалізувати адаптивне навчання та забезпечує повну автономність роботи додатку.

До перспектив розвитку комплексу може належати доповнення його AI-модулем автоматичної сертифікації рівня володіння мовою відповідно до стандартів CEFR, а також інтерактивним діалоговим режимом, де штучний інтелект моделюватиме живого співрозмовника. Крім того, можливе створення мобільної версії та впровадження технологій прогнозування навчального успіху, що підвищить ефективність навчання та зробить систему універсальним інструментом як для індивідуальної, так і для групової роботи.

Таким чином, програмно-апаратний комплекс використовує штучний інтелект на всіх ключових етапах – від розпізнавання мовлення до аналізу граматики, генерації навчальних матеріалів та синтезу мовлення, що робить його сучасним інтелектуальним рішенням для повноцінного вивчення української мови.

Список використаних джерел

1. Whisper: Robust and Efficient Speech Recognition at the Edge. URL: https://cdn.openai.com/papers/whisper.pdf (дата звернення: 21.11.2025)
2. OpenAI Whisper. URL: https://openai.com/index/whisper/ (дата звернення: 21.11.2025)
3. What is Generative AI? URL: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-generative-ai (дата звернення: 21.11.2025)Automatic Speech Recognition (ASR) System Development. URL: https://mobidev.biz/blog/automatic-speech-recognition-asr-system-development (дата звернення: 21.11.2025)
4. Jurafsky D., Martin J. Speech and Language Processing (3rd edition draft). URL: https://stanford.edu/~jurafsky/slp3/ (дата звернення: 21.11.2025)

Азарова Анжеліка Олексіївна – к.т.н., проф. каф. ОТ Факультету інформаційних технологій та комп'ютерної інженерії Вінницького національного технічного університету, м. Вінниця.

Тетерев Віталій Ігорович – ст. гр. 1КІ-24м Факультету інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця.

Azarova Anzhelika A. – Candidate of technical sciences, Professor of Management and security information systems department, Deputy dean of the Faculty of management and information security of research and international cooperation, Vinnytsia National Technical University, Vinnytsia.

Teterev Vitalii I. – *student of group 1KI-24m, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia.*