Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ"
Факультет інформатики
Кафедра інформатики

Кваліфікаційна робота
Освітній ступінь – магістр

на тему: "АВТОМАТИЧНЕ ТРАНСКРИБУВАННЯ ТА ІДЕНТИФІКАЦІЯ У ЗМІШАНОМУ УКРАЇНСЬКО-АНГЛІЙСЬКОМУ МОВЛЕННІ"

Виконав: студент 2-го року
навчання Спеціальності
122 Комп’ютерні науки
Нікітін Руслан Валерій Умарович

Керівник: Ігнатенко Олексій
Петрович доктор фіз.-мат. наук,
доцент

Рецензент: Солопова Вероніка Максимівна
доктор філософії з комп'ютерних наук

Магістерська робота захищена
з оцінкою _____

Секретар ЕК _____

"____" _____2026 р.

Київ – 2026

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ"
Факультет інформатики
Кафедра інформатики

ЗАТВЕРДЖУЮ
Зав. кафедри інформатики
кандидат фіз.-мат. наук, доцент
Гороховський Семен Самуїлович

_____
(підпис)
“_____” _____2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на магістерську роботу

студенту 2 р.н. магістерської програми Комп’ютерні науки
Нікітіну Руслану Валерію Умаровичу

Дослідити ефективність сучасних архітектур систем автоматичного розпізнавання мовлення на задачі транскрибування змішаного українсько-англійського мовлення.

Зміст текстової частини кваліфікаційної роботи:

Зміст
Анотація
Вступ
Розділ 1. Теоретичні засади
Розділ 2. Створення та аналіз датасету
Розділ 3. Бенчмарк ASR-моделей та аналіз результатів
Розділ 4. Розробка системи транскрибування повного циклу
Висновки
Список використаних джерел
Додатки

Дата видачі “_____” _____ 2025 р.
Керівник О. П. Ігнатенко,
доктор фіз.-мат. наук, доцент _____
(підпис)

Завдання отримав _____
Р. В. У. Нікітін (підпис)

Тема: Автоматичне транскрибування та ідентифікація у змішаному українсько-англійському мовленні

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проєкту (роботи)	Термін виконання
1.	Отримання завдання магістерської роботи	07.11.2025
2.	Огляд і систематизація літератури за темою дослідження	15.11.2025
3.	Формулювання теоретичних засад та постановки задачі	13.12.2025
4.	Проєктування, збір та анотування спеціалізованого датасету змішаного мовлення	17.01.2026
5.	Розробка методології оцінювання та мапи зворотної текстової нормалізації	03.02.2026
6.	Проведення бенчмарку сучасних ASR-архітектур	05.03.2026
7.	Проєктування та розробка серверної частини застосунку	28.03.2026
8.	Реалізація інтерфейсу користувача та тестування системи	18.04.2026
9.	Передзахист магістерської роботи	30.04.2026
10.	Аналіз даних, порівняння методів і формування висновків	15.05.2026
11.	Написання та оформлення магістерської роботи	30.04.2026
12.	Доопрацювання текстової частини	02.05.2026
13.	Підготовка презентації та виступу	24.05.2026
14.	Захист магістерської роботи	8.06.2026

Студент Нікітін Р. В. У.

Керівник Ігнатенко О. П. “____”_____20__р.

ЗМІСТ

СПИСОК УМОВНИХ СКОРОЧЕНЬ

1. IT – інформаційні технології
2. СУБД – система управління базами даних
3. AI (Artificial Intelligence) – штучний інтелект
4. API (Application Programming Interface) – прикладний програмний інтерфейс
5. ASR (Automatic Speech Recognition) – автоматичне розпізнавання мовлення
6. BLEU (Bilingual Evaluation Understudy) – метрика оцінювання якості машинного перекладу
7. BLOB (Binary Large Object) – масив двійкових даних (тип даних у СУБД)
8. BPE (Byte Pair Encoding) – алгоритм кодування парами байтів (підслівна токенізація)
9. CER (Character Error Rate) – частка помилок на рівні символів
10. CS-WER (Code-Switching Word Error Rate) – частка помилок на рівні слів для змішаного мовлення (ізольована оцінка іншомовних включень)
11. CTC (Connectionist Temporal Classification) – конекціоністська часова класифікація (функція втрат для навчання акустичних нейромереж)
12. CUDA (Compute Unified Device Architecture) – програмно-апаратна архітектура паралельних обчислень на графічних процесорах
13. ECAPA-TDNN (Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Network) – архітектура нейронної мережі із часовою затримкою для ідентифікації мовців
14. EN (English) – англійська мова
15. FST (Finite-State Transducer) – скінченний перетворювач
16. FTS5 (Full-Text Search 5) – модуль повнотекстового пошуку для SQLite
17. GELU (Gaussian Error Linear Unit) – функція активації в штучних нейронних мережах
18. GPU (Graphics Processing Unit) – графічний процесор
19. ITN (Inverse Text Normalization) – зворотна текстова нормалізація
20. LDC (Linguistic Data Consortium) – Консорціум лінгвістичних даних
21. LID (Language Identification) – автоматична ідентифікація мови

22. LLM (Large Language Model) – велика мовна модель

23. LSTM (Long Short-Term Memory) – архітектура рекурентної нейронної мережі з довгою короткостроковою пам'яттю

24. METEOR (Metric for Evaluation of Translation with Explicit ORdering) – метрика оцінювання якості машинного перекладу

25. PCM (Pulse-Code Modulation) – імпульсно-кодова модуляція (формат нестисненого цифрового аудіо)

26. QA (Quality Assurance) – забезпечення якості програмного забезпечення

27. REST (Representational State Transfer) – архітектурний стиль взаємодії компонентів розподіленого застосунку

28. RTF (Real-Time Factor) – коефіцієнт реального часу (метрика швидкодії оброблення аудіо)

29. SDK (Software Development Kit) – набір засобів розробки програмного забезпечення

30. Seq2Seq (Sequence-to-Sequence) – архітектура нейронних мереж для перетворення послідовностей

31. STT (Speech-to-Text) – системи перетворення мовлення в текст

32. TDT (Token-and-Duration Transducer) – архітектура розпізнавання мовлення з передбаченням тривалості токенів

33. UK (Ukrainian) – українська мова

34. VAD (Voice Activity Detection) – детектування активності мовлення

35. WER (Word Error Rate) – частка помилок на рівні слів

36. WFST (Weighted Finite-State Transducer) – зважений скінченний перетворювач

АНОТАЦІЯ

Магістерська робота присвячена оцінюванню та порівнянню сучасних архітектур автоматичного розпізнавання мовлення на задачі транскрибування змішаного українсько-англійського мовлення. Досліджено особливості роботи локальних Encoder-Decoder моделей, комерційних API-сервісів та мультимодальних великих мовних моделей, а також методи ізольованого оцінювання помилок (метрика CS-WER) і зворотної текстової нормалізації.

Розроблено спеціалізований еталонний корпус uk-en-code-mixed-asr-2h з параметрами: 448 аудіозаписів загальною тривалістю понад 2 години, IT-домен, 99,6 % внутрішньореченнєвого перемикання кодів. Реалізовано настільний застосунок для транскрибування повного циклу, який об'єднує підсистеми детектування активності мовлення із двопороговим гістерезисом, розпізнавання тексту та діаризації з міжсесійною ідентифікацією мовців на основі векторів ECAPA-TDNN.

Експериментальне порівняння 49 конфігурацій моделей показало такі результати (загальний WER після ITN-нормалізації): найкраща мультимодальна Audio-LLM Gemini 3.1 Pro – 0,105, найкращий комерційний API ElevenLabs Scribe v2 – 0,134, найкраща локальна модель Whisper large-v3-turbo – 0,167. Встановлено, що показник помилок на англійських термінах системно перевищує помилки на українських словах у 3-10 разів. Застосування алгоритму ITN знизило кількість помилок на англіцизмах у середньому на 13 %, що доводить орфографічну, а не акустичну природу збоїв (кирилична транслітерація термінів). Дослідження моделей наскрізного перекладу виявило неспроможність традиційних архітектур коректно обробляти IT-сленг через дослівне калькування. Натомість мультимодальні Audio-LLM продемонстрували високу ефективність у семантичній адаптації українського професійного жаргону в технічну англійську мову, що робить переклад життєздатною альтернативою транскрибуванню з ITN.

Ключові слова: *автоматичне розпізнавання мовлення, ASR,*

code-switching, змішане українсько-англійське мовлення, CS-WER, зворотна текстова нормалізація (ITN), діаризація, ідентифікація мовця.

ВСТУП

Питання автоматичного розпізнавання мовлення є одним із ключових у сучасній обробці природної мови та системах штучного інтелекту. Розпізнавання змішаного мовлення становить класичну проблему для дослідження мовних моделей, зокрема явищ інтеграції іншомовної технічної лексики в синтаксис матричної мови.

Дослідження в цій галузі мають значну історію: від ранніх прихованих марковських моделей до сучасних застосувань глибокого навчання та великих мовних моделей. Водночас стрімкий розвиток від CTC-архітектур і Encoder-Decoder систем до новітніх мультимодальних Audio-LLM потребує систематичного порівняльного аналізу їх застосування на специфічних мовних парах.

Актуальність підсилюється тим, що для українсько-англійського мовлення задача ускладнюється фундаментальною асиметрією систем письма (кирилиця та латиниця). Різні підходи до транскрибування мають принципово різну природу: акустичні CTC-моделі намагаються встановити пряму відповідність між звуком та літерою, Encoder-Decoder моделі жорстко залежать від токенізатора та початкового мовного тегу, а Audio-LLM покладаються на глибокий семантичний контекст. Порівняльний аналіз цих підходів на спільній задачі транскрибування IT-лексики дає змогу виявити їх сильні та слабкі сторони й надати обґрунтовані рекомендації щодо застосування.

Метою дослідження є розробка та експериментальний аналіз методів автоматичного транскрибування змішаного українсько-англійського мовлення з метою виявлення характеристик, переваг та обмежень сучасних ASR-архітектур, а також створення системи повного циклу для обробки такого мовлення.

Для досягнення поставленої мети визначено такі завдання:

- Проаналізувати сучасні архітектури систем розпізнавання мовлення, зокрема CTC, Encoder-Decoder та Audio-LLM.

- Розробити та анотувати спеціалізований еталонний датасет українсько-англійського змішаного мовлення для IT-домену.
- Розробити методологію об’єктивного оцінювання (метрика CS-WER) та алгоритм зворотної текстової нормалізації (ITN) для усунення орфографічної варіативності.
- Провести серію експериментів в однакових умовах для об’єктивного порівняння результативності 35 ASR-моделей.
- Проаналізувати отримані результати, визначити причини галюцинацій та специфіку помилок на іншомовних токенах.
- Спроєктувати та розробити десктопний застосунок для транскрибування з підтримкою VAD та діаризації.

Об’єктом дослідження є процес автоматичного транскрибування та ідентифікації дикторів у змішаному українсько-англійському мовленні засобами алгоритмів машинного навчання.

У роботі застосовано методи комп’ютерного моделювання, обробки природної мови та експериментального аналізу. Використано метрики на основі відстані Левенштейна (WER, CER, CS-WER) для оцінки точності розпізнавання; детерміновані алгоритми текстової нормалізації на базі регулярних виразів для зворотного мапінгу; рекурентні нейронні мережі для сегментації аудіо; архітектуру ECAPA-TDNN для вилучення голосових векторів та метод косинусної схожості для ідентифікації мовців.

Експерименти проведено в уніфікованому середовищі з фіксованими параметрами на базі створеного датасету uk-en-code-mixed-asr-2h (448 аудіозаписів, 16 тематичних IT-категорій). Експериментальні дані проаналізовано статистичними методами для оцінювання загального WER, ізольованого рівня помилок на англійських словах (CS-WER EN) та швидкодії моделей.

Наукова новизна роботи полягає у комплексному порівнянні понад 35 моделей різних архітектур в однакових експериментальних умовах на задачі code-switching. Вперше введено в науковий обіг відкритий корпус начитаного

українсько-англійського IT-мовлення.

Отримано нові емпіричні дані про специфіку помилок ASR-моделей: встановлено, що мультимодальні Audio-LLM (зокрема Gemini 3.1 Pro) забезпечують найвищу точність (WER 0,105 після ITN), проте схильні до галюцинацій. Доведено, що показник помилок на англійських термінах (CS-WER EN) системно перевищує помилки на українських словах у 3–10 разів. Експериментально підтверджено, що застосування ITN знижує помилки на англіцизмах у середньому на 13%, що свідчить про переважно орфографічну (кирилична транслітерація), а не акустичну природу збоїв.

Розроблений програмний інструментарій дає змогу здійснювати транскрибування та крос-сесійну ідентифікацію мовців у реальному часі. Використання двопорогового гістерезису VAD вирішує проблему стрибків пам’яті, а зниження порогу косинусної схожості адаптує систему до зміни акустичних ознак голосу під час перемикання мов. Результати бенчмарку можуть застосовуватися IT-компаніями для обґрунтованого вибору ASR-провайдерів у корпоративних продуктах.

Робота складається з чотирьох розділів.

Перший розділ присвячено теоретичним засадам. Розглянуто code-switching як лінгвістичне явище, проаналізовано особливості метрик оцінювання та архітектур систем розпізнавання мовлення.

У другому розділі наведено результати проєктування та збору спеціалізованого датасету uk-en-code-mixed-asr-2h. Описано протокол збору, виконано статистичний та словниковий аналіз корпусу.

Третій розділ присвячено експериментальним дослідженням. Представлено результати бенчмарку 49 конфігурацій ASR-моделей, виконано аналіз впливу зворотної нормалізації на точність розпізнавання, досліджено обмеження моделей наскрізного перекладу та наведено типологію галюцинацій.

У четвертому розділі описано розробку системи транскрибування повного циклу. Наведено архітектуру бекенду, рішення для оптимізації роботи VAD та модулів діаризації з урахуванням специфіки змішаного мовлення.

РОЗДІЛ 1: ТЕОРЕТИЧНІ ЗАСАДИ

1.1 Code-switching як лінгвістичне явище

Явище code-switching полягає у перемиканні між двома або більше мовами в межах одного мовленнєвого акту. Найчастіше його можна спостерігати у мовленні білінгвів або в професійних спільнотах, де частина понять стабільно вживається іншою мовою. У контексті цієї роботи йдеться про українсько-англійське змішане мовлення, зокрема мовлення з англійськими технічними термінами всередині українських речень.

Варто розрізняти code-switching, code-mixing та borrowing. Code-switching зазвичай означає перемикання між мовами на рівні речень, фраз або окремих фрагментів мовлення. Code-mixing частіше описує змішування мов усередині одного речення без чіткої зміни комунікативної ситуації. Borrowing, або запозичення, стосується іншомовних слів, які вже закріпилися в мові-реципієнті й сприймаються як частина її лексики. Наприклад, слово "комп’ютер" є радше запозиченням, тоді як вислів "треба зробити pull request" є прикладом code-mixing.

У літературі зазвичай виділяють кілька типів code-switching. Перший тип - це inter-sentential code-switching, тобто перемикання між реченнями. У такому випадку одне речення вимовляється однією мовою, а наступне – іншою. Наприклад: "Я вже завершив задачу. I will open the pull request today." Другий тип - це intra-sentential code-switching, тобто перемикання всередині одного речення. Саме цей тип є найважливішим для цієї роботи, оскільки більшість прикладів українсько-англійського професійного мовлення мають саме таку структуру: "Я вже завершив задачу. Я відкрию pull request сьогодні.". Третій тип - це intra-word code-switching, коли іншомовна основа поєднується з морфологічними елементами іншої мови. Використовуючи той самий контекст, це може звучати так: "Я вже завершив задачу, залишилося тільки її задеплоїти". Для українсько-англійського мовного контакту такі форми є надзвичайно поширеними: англійські корені органічно приймають українські префікси та

закінчення, утворюючи слова на кшталт "заапрувити", "пофіксити", "закомітити". [1, 2]

Українсько-англійський code-switching є окремо цікавим для задач автоматичного розпізнавання мовлення через асиметрію алфавітів. Українська мова використовує кирилицю, тоді як англійська – латиницю. Через це модель має не лише розпізнати слово на слух, а й правильно визначити, якою графікою його потрібно записати. Наприклад, слово "deploy" може бути розпізнане як "деплой", "діплой", “диплой”, “де плой”, “тіплой” або залишене латиницею. Для людини такі варіанти часто зрозумілі з контексту, однак для ASR-системи це різні токени, що впливають на підрахунок помилок.

Ця проблема стосується і токенізації. Більшість сучасних моделей використовують підслівні токенізатори, навчені на великих корпусах тексту. Якщо токенізатор недостатньо добре представлений українсько-англійськими прикладами, він може неефективно розбивати іншомовні слова або змішані форми. Особливо це помітно на словах, де англійська основа отримує українські афікси. Для ASR це створює додаткову складність, оскільки помилка може виникнути не лише на рівні акустичного розпізнавання, а й на рівні перетворення розпізнаного сигналу в текстову послідовність.

Соціолінгвістично українсько-англійське змішане мовлення особливо поширене в IT-середовищі. Причин для цього декілька. По-перше, значна частина технічної документації, назв інструментів і робочих термінів існує англійською мовою. По-друге, для деяких понять немає усталеного або зручного українського відповідника. Через це фрази на зразок "змерджити pull request", "задеплоїти на staging", "перевірити logs" або "пофіксити bug" є природними для усного мовлення українських розробників.

Для ASR-систем таке мовлення створює низку типових помилок. Модель може пропустити англійське слово, замінити його фонетично подібним українським словом, записати англійський термін кирилицею або перекласти його замість транскрибування. У результаті транскрипція може формально

залишатися зв’язною, але втрачати важливі технічні деталі. Саме тому українсько-англійське code-switching потребує окремого дослідження, а не лише перевірки загальної підтримки української та англійської мов у моделі.

1.2 Огляд існуючих датасетів

Для дослідження автоматичного розпізнавання змішаного мовлення потрібні корпуси, у яких зафіксовано перемикання між мовами в межах одного мовленнєвого фрагмента. Такі корпуси дають змогу не лише навчати ASR-моделі, а й порівнювати їх в однакових умовах: на тих самих аудіозаписах, з однаковими референсними транскрипціями та однаковими метриками. Без таких даних оцінка якості моделі залишається неповною, оскільки загальна підтримка двох мов не означає, що система добре працюватиме з їхнім змішуванням.

Таблиця 1.2 – Порівняння датасетів

Датасет	Мови	Тип мовлення	Тривалість	Кількість мовців	Публічний доступ
ASCEND	ZH-EN	spontaneous	10,62 год	23	так
SEAME	ZH/MS-EN	spontaneous	192 год	156	частково
Miami Bangor	ES-EN	spontaneous	35 год	84	так
FAME!	FR-NL	broadcast	18,5 год	~541	так
uk-en-code-mixed-asr-2h	UK-EN	read	2,1 год	1	так

Порівняння цих корпусів узагальнено в таблиці 1.2. За типом перемикання кодів датасети суттєво різняться. ASCEND та Miami Bangor охоплюють обидва основні типи – inter-sentential та intra-sentential. SEAME орієнтований цільово на intra-sentential перемикання, що робить його найближчим за структурою до

задачі цього дослідження серед існуючих корпусів. FAME! є найширшим за охопленням: крім міжреченнєвого та внутрішньореченнєвого перемикання, він містить і intra-word випадки.

Корпус uk-en-code-mixed-asr-2h є найвужчим за типом перемикання навмисно: 446 із 448 зразків (99,6 %) є випадками intra-sentential code-switching, а решта два – монолінгвальними записами.

Одним із відомих датасетів є ASCEND, створений для китайсько-англійського code-switching. Він містить 10,62 години чистого спонтанного мовлення від 23 двомовних мовців. Корпус побудовано на багатораундових діалогах, що робить його корисним для оцінювання моделей у наближених до реальної розмови умовах. Важливо, що ASCEND орієнтований саме на спонтанне мовлення, тоді як значна частина попередніх code-switching ресурсів використовувала начитані фрази. [3]

SEAME є одним із найбільших і найчастіше цитованих корпусів для мандаринсько-англійського code-switching. Він був зібраний у Сінгапурі та Малайзії, тобто в регіонах, де змішування мандаринської та англійської є природною частиною усного мовлення. У різних джерелах обсяг SEAME подається по-різному: у статті 2010 року про корпус згадується 30 години спонтанного транскрибованого мовлення, тоді як LDC-реліз [4] описує приблизно 192 години мовлення від 156 мовців. [5]

Miami Bangor Corpus присвячений іспансько-англійському білінгвальному мовленню. Його було записано й транскрибовано у 2008–2011 роках у межах проєкту, спрямованого на перевірку моделей code-switching на іспансько-англійських даних. У корпусі представлене природне мовлення 84 мовців, які проживали в Маямі. Це робить Miami Bangor важливим ресурсом не лише для ASR, а й для лінгвістичного аналізу білінгвального мовлення. [6]

FAME! відрізняється від попередніх корпусів тим, що побудований на

радіомовленні, а не на побутових діалогах чи інтерв'ю. Він охоплює фризько-нідерландське мовлення й був створений для дослідження code-switching у регіональному broadcast-контексті. Корпус містить 18,5 години анотованих аудіозаписів, 309 ідентифікованих спікерів, з яких 21 з'являється кілька разів, та 233 неідентифікованих. У ASR-експериментах використовуються фризькі та нідерландські сегменти для навчання й тестування. Такий формат корисний тим, що показує code-switching у медійному середовищі, де якість аудіо, стиль мовлення та структура реплік відрізняються від звичайної розмови. [7]

Окремо варто згадати FLEURS. Це не спеціалізований датасет code-switching, а мультилінгвальний speech benchmark, побудований на основі FLoRes-101. Він охоплює 102 мови й містить приблизно 12 годин мовлення на кожну мову. FLEURS можна використовувати для ASR, speech language identification, speech translation та інших задач, пов'язаних із багатомовним мовленням. Однак наявність багатьох мов у корпусі не означає наявність перемикання між мовами в межах одного висловлювання. Через це FLEURS є корисним для оцінювання мультилінгвальних моделей, але не розв'язує задачу оцінювання code-switching. [8]

Ближчим до задачі цієї роботи є CS-FLEURS, створений саме для code-switched speech recognition і speech translation. Корпус охоплює 52 мови та 113 унікальних code-switched мовних пар. Він містить кілька тестових наборів, зокрема X-English пари з реальними голосами, synthetic speech через generative text-to-speech, а також lower-resourced X-English пари. Крім тестових наборів, CS-FLEURS надає training set обсягом 128 годин synthetic speech для 16 X-English мовних пар. Хоча цей бенчмарк охоплює й українсько-англійське змішування, відповідні аудіодані повністю згенеровані штучно за допомогою text-to-speech систем (зокрема моделей конкатенативного або генеративного синтезу). На практиці якість такого синтезу для українсько-англійського перемикання кодів є вкрай низькою: алгоритми генерують неприродну

просодію, мають виражений "роботизований" акцент і суттєво спотворюють фонетику англійських IT-термінів, вбудованих в українське речення. Через це використання цього сегменту CS-FLEURS для оцінювання ASR-систем у реальних умовах є дуже обмеженим. Незважаючи на це, він залишається одним із наймасштабніших сучасних ресурсів для code-switching, але його структура орієнтована на широку мультилінгвальну оцінку, а не на доменно специфічне українсько-англійське професійне мовлення. [9]

Порівняння цих корпусів показує, що українсько-англійська пара залишається прогалиною. Її не можна повністю замінити даними з китайсько-англійського або іспансько-англійського code-switching, оскільки кожна мовна пара має власні фонетичні особливості. Для української та англійської окремою проблемою є різні системи письма: українська використовує кирилицю, а англійська – латиницю. Через це ASR-модель має не лише розпізнати іншомовний фрагмент на слух, а й правильно визначити, як його записати: латиницею, транслітерацією або перекладом. Саме на таких межах часто виникають помилки токенізації, пропуски англійських слів або заміни на фонетично подібні українські слова.

Ця прогалина є важливою саме зараз через активне використання англійської технічної лексики в українському IT-середовищі. У робочих обговореннях природно трапляються фрази на зразок "впав production", "API лежить" тощо. Для людини такі вислови є зрозумілими, але для ASR-системи вони створюють змішаний акустичний і текстовий сигнал. Якщо модель неправильно розпізнає англійський термін, транскрипція може втратити практичну цінність: її складніше шукати, аналізувати або використовувати як технічний запис зустрічі.

Для заповнення цієї прогалини в межах даного дослідження було розроблено спеціалізований датасет uk-en-code-mixed-asr-2h, орієнтований на українсько-англійське змішане мовлення в IT-домені. Корпус містить 448

аудіозаписів загальною тривалістю понад 2 години (начитане мовлення одного диктора). За типом перемикання датасет майже повністю складається з intra-sentential code-switching, що точно відображає специфіку вставлення англійських термінів в україномовні речення. Вибір формату начитаного мовлення (read speech) зумовлений необхідністю мати абсолютно точну еталонну транскрипцію для безпомилкового обчислення спеціалізованих метрик оцінювання (зокрема CS-WER). Детальний опис концепції, протоколу збору та статистичний аналіз цього корпусу наведено в Розділі 2. [10]

1.3 Моделі автоматичного розпізнавання мовлення

Еволюція систем автоматичного розпізнавання мовлення пройшла шлях від прихованих марковських моделей до повністю нейромережевих архітектур. У контексті розпізнавання змішаного мовлення архітектура моделі відіграє вирішальну роль, оскільки саме від неї залежить баланс між акустичним сигналом і мовним контекстом. Сучасні ASR-системи можна поділити на кілька фундаментальних класів: моделі на основі CTC, Transducer-архітектури, авторегресивні системи Encoder-Decoder та новітні мультимодальні Audio-LLM.

1.3.1 CTC-моделі та архітектура wav2vec 2.0

Connectionist Temporal Classification або CTC – це функція втрат і відповідний механізм декодування, який дає змогу навчати нейронні мережі вирівнювати вхідні аудіокадри з вихідними текстовими токенами без потреби у покадровій розмітці. Оскільки аудіосигнал значно довший за текстову послідовність, CTC використовує спеціальний порожній токен є. Після генерації ймовірностей для кожного кадру дублікати та порожні токени колапсуються у фінальний текст.

Сучасним стандартом CTC-підходу стала архітектура wav2vec 2.0, яка лежить в основі моделей MMS[11] та спеціалізованих українських систем. Її головною інновацією є навчання без вчителя на масивах нерозміченого аудіо.

Архітектура складається з трьох основних компонентів:

- екстрактор ознак - це багатошарова згорткова нейронна мережа, яка перетворює сиру аудіохвилю на латентні акустичні ознаки;
- модуль квантизації, що переводить неперервні представлення у дискретні кодові вектори з наперед заданого словника;
- контекстна мережа на базі Transformer, яка формує глобальні представлення на основі маскованих ознак.

Під час попереднього навчання частина латентних векторів маскується, а Transformer повинен передбачити правильний квантований вектор серед набору негативних кандидатів. Для задач транскрибування поверх контекстної мережі додається лінійна CTC-голова, після чого модель проходить донавчання на розміченому мовленні. [12]

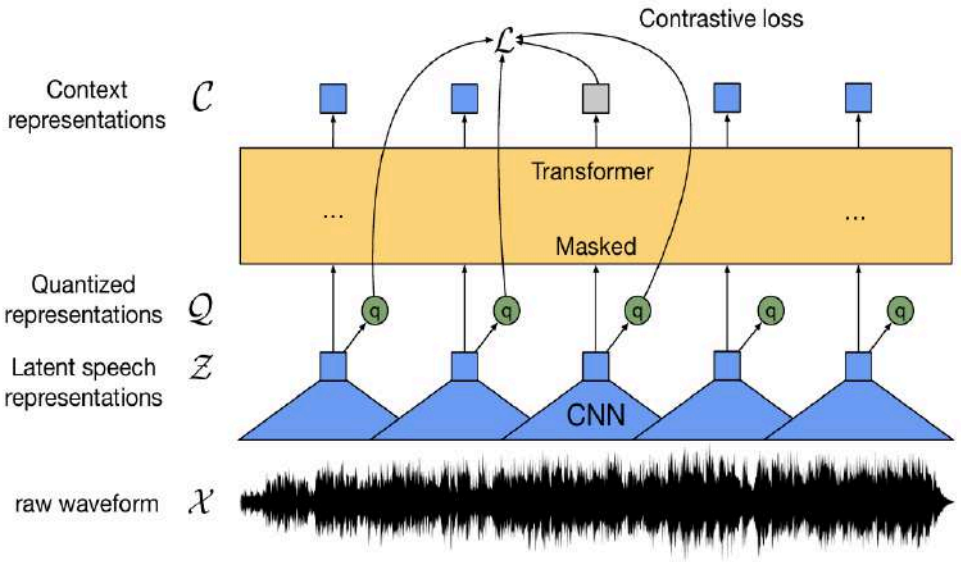

Рисунок 1.3.1 – Архітектура моделі wav2vec 2.0 [12]

Попри високу швидкість роботи завдяки неавторегресивному декодуванню, CTC-моделі мають відносно слабку внутрішню мовну модель. Вони встановлюють пряму відповідність між звуком та літерою. Через це при роботі зі змішаним українсько-англійським мовленням моделі, навчені переважно на українських корпусах, намагаються фонетично адаптувати

англійські звуки до кириличного алфавіту. Це неминуче призводить до втрати оригінального написання англіцизмів.

Подальшим розвитком ідеї wav2vec 2.0 стала архітектура XLS-R (Cross-lingual Speech Representation) [13], яка масштабувала самонавчання на 128 мов. На базі цих багатомовних моделей створюються спеціалізовані рішення для конкретних мов шляхом донавчання. Для української мови помітними представниками цього підходу є відкриті моделі розробника Yehor Smoliakov (зокрема w2v-xls-r-uk та w2v-bert-uk-v2.1) [14].

1.3.2 Авторегресивні моделі Encoder-Decoder (Whisper)

Архітектури з механізмом Attention розділяють процес розпізнавання мовлення на дві незалежні, але пов'язані частини. Енкодер аналізує весь аудіосигнал цілком, а декодер авторегресивно генерує текст по одному токену, використовуючи механізм Cross-Attention для фокусування на релевантних ділянках аудіо під час створення кожного наступного слова.

Найвідомішим представником цього класу є сімейство моделей Whisper від OpenAI. Вхідним сигналом для Whisper є 80-канальна логарифмічна мел-спектрограма. Енкодер складається з двох шарів одномірної згортки із функцією активації GELU та стеку Transformer-блоків із синусоїдальними позиційними кодуваннями. Декодер також побудований на Transformer-архітектурі та використовує унікальний мультизадачний формат токенів.

Для української транскрипції Whisper очікує на вході декодера сувору службову послідовність <|*startoftranscript*|> <|*uk*|> <|*transcribe*|> <|*notimestamps*|>. У задачах змішаного мовлення ця архітектура має як переваги, так і обмеження. Потужний Transformer-декодер спирається на байтовий BPE-токенізатор. Це дозволяє моделі пам’ятати правильне написання складних англійських технічних термінів. Однак жорсткий мовний тег <|uk|>

змінює розподіл ймовірностей у декодері, змушуючи його надавати екстремальну перевагу кириличним токенам. У результаті модель стикається з конфліктом між акустикою та мовним контекстом і часто генерує транслітеровані варіанти слів замість оригінальних. [15]

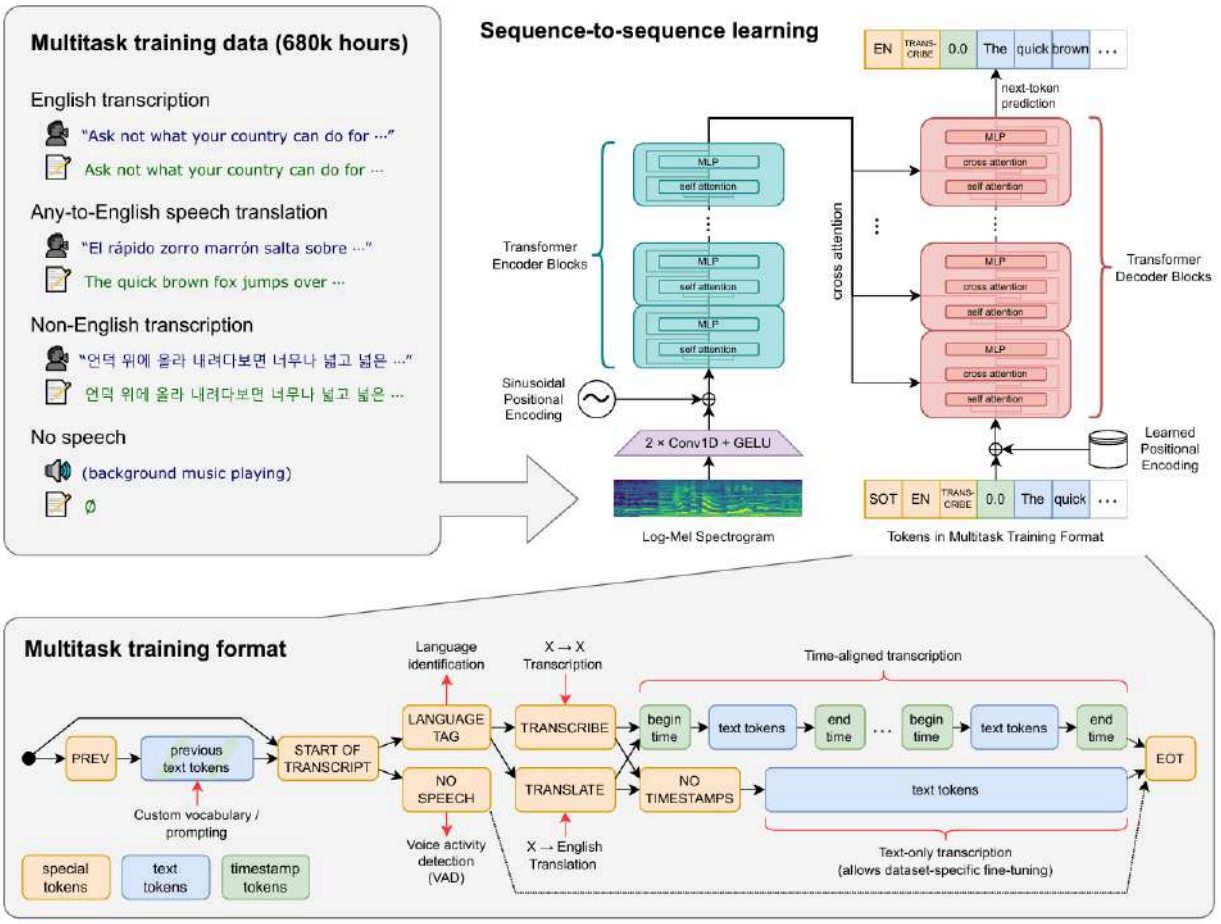

Рисунок 1.3.2 – Архітектура та мультизадачний формат токенів моделі Whisper [15]

1.3.3 Transducer-архітектури та Conformer

Transducer-архітектури поєднують переваги CTC-підходу та авторегресивних систем. Вони забезпечують потокове розпізнавання мовлення при збереженні потужної вбудованої мовної моделі. Базова архітектура складається з трьох компонентів:

- акустичний енкодер, який обробляє вхідний аудіосигнал;

- мережа передбачення, що виконує функцію мовної моделі та обробляє попередньо згенеровані токени;
- об'єднувальна мережа, яка зводить інформацію з двох попередніх модулів для генерації наступного символу.

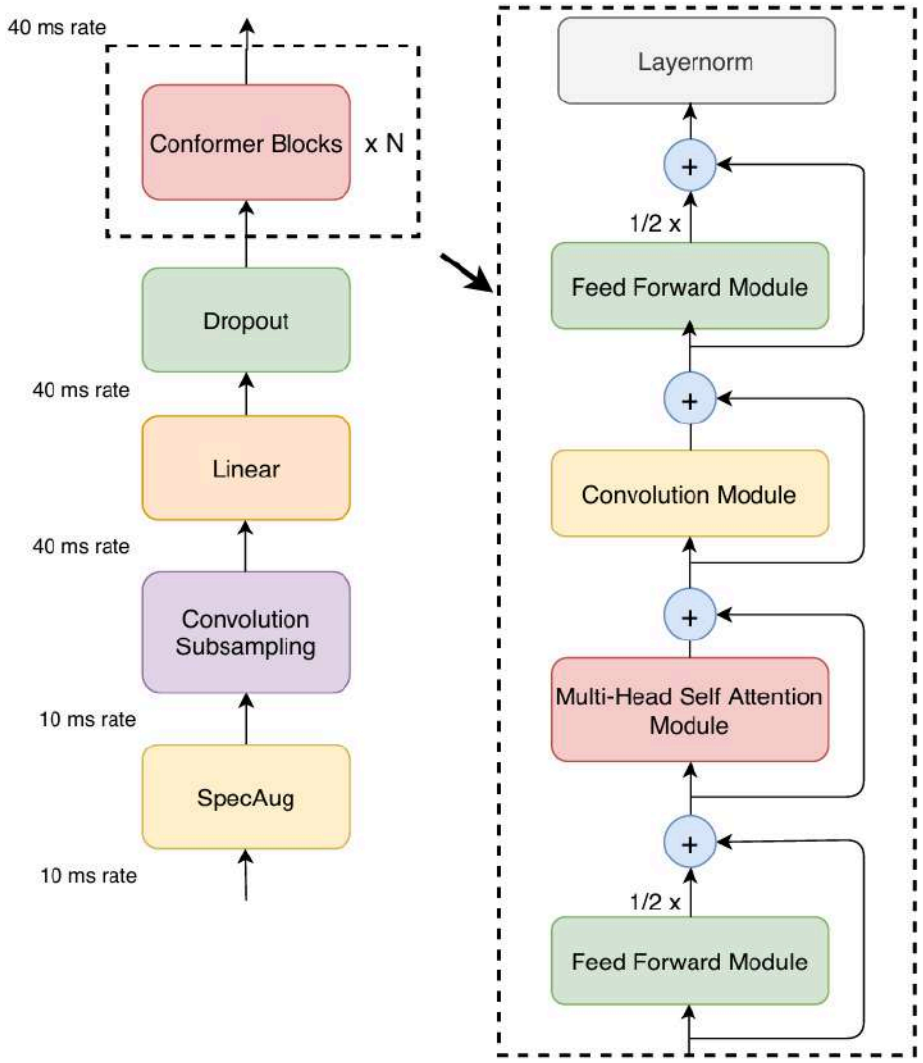

Рисунок 1.3.3 – Структура блоку Conformer [16]

У сучасних реалізаціях енкодер базується на архітектурі Conformer [16]. Екосистема NVIDIA NeMo [17] активно використовує цю архітектуру для створення таких моделей, як Citrinet та Canary. Окремим різновидом цього напряму є TDT (Token-and-Duration Transducer) [18]. На відміну від класичних систем, які роблять передбачення для кожного аудіокадру, TDT прогнозує не лише наступний токен, а й кількість кадрів, які можна безпечно перескочити. Це

значно зменшує обчислювальне навантаження та робить моделі лінійки Parakeet придатними для роботи в реальному часі.

Варто також згадати монолінгвальні адаптації конформерів, такі як Ukrainian Pods Conformer [19], що тренуються виключно на українських даних. Крім того, для задач з обмеженими обчислювальними ресурсами широко використовуються легковагові офлайн-рішення, зокрема інструментарій Vosk [20], який забезпечує швидке розпізнавання на локальному обладнанні.

1.3.4 Audio-LLM

Audio-LLM є новітнім класом нейромереж, у яких розпізнавання мовлення відбувається не в ізольованій акустичній системі, а всередині великої мовної моделі. На відміну від каскадних систем, де текст спочатку транскрибується окремою моделлю, а потім виправляється мовною моделлю, тут реалізовано механізм раннього злиття.

Типова архітектура Audio-LLM має трикомпонентну структуру:

Аудіоенкодер → Проекційний шар → LLM-декодер

Аудіоенкодер (наприклад, попередньо натренований блок із моделі Whisper) перетворює мовлення на неперервні латентні ознаки. Оскільки акустичні вектори та текстові токени існують у різних математичних просторах, використовується проекційний шар (адаптер). Він стискає ці ознаки та перетворює їх на віртуальні токени, зрозумілі для текстової мовної моделі. Після цього LLM генерує текст, маючи доступ одночасно до акустичної інформації та глибокого семантичного контексту.

Теоретично такий підхід є ідеальним для розпізнавання змішаного мовлення. Оскільки великі мовні моделі навчалися на терабайтах технічної документації та програмного коду, вони "розуміють" специфічну IT-лексику краще за будь-яку класичну акустичну модель. У межах роботи розглядаються

кілька ключових представників цього класу:

Сімейство моделей Voxtral від компанії Mistral AI позиціонується як відкриті системи розуміння мовлення (speech understanding), орієнтовані не лише на транскрибування, а й на безпосередній аналіз аудіо. Ці моделі доступні у двох розмірах: Voxtral Small із 24 мільярдами параметрів для продукційних сценаріїв та Voxtral Mini із 3 мільярдами параметрів для локального або edge-використання. Обидві версії підтримують довгий аудіоконтекст, автоматичне визначення мови, надання відповідей на запитання та самаризацію без залучення окремого ASR-модуля. У документації розробника підтверджено підтримку англійської, іспанської, французької, португальської, гінді, німецької, нідерландської та італійської мов, а в еталонних оцінюваннях FLEURS також використовується арабська. Для цього дослідження Voxtral становить інтерес як інструмент для перевірки того, чи впливає фізичний обсяг параметрів моделі на здатність коректно зберігати іншомовні терміни без їхньої фонетичної транслітерації. Voxtral Small підтримує українську мову, Mini – ні. [21]

Архітектура сімейства Gemini від Google DeepMind відрізняється тим, що розроблялася як нативно мультимодальна система від самого початку створення. Вона здатна одночасно працювати з текстом, зображеннями, відео та аудіо. У контексті транскрибування такий підхід важливий тим, що модель не обмежується суто акустичним розпізнаванням, а залучає глибокий семантичний контекст. Покоління Gemini 2.5, 3.0, 3.1, 3.5 окремо вирізняється здатністю обробляти мільйони токенів контексту, включно з годинами безперервного аудіо. Тому в даному дослідженні Gemini розглядається як універсальний інструмент, що аналізує аудіосигнал крізь призму своїх знань про програмування та структуру технічного тексту. [22, 23, 24]

Наступними представниками цього класу є моделі MiMo 2.0 Omni та її оновлена версія MiMo v2.5. Вони є нативно омнімодальними системами, які приймають текст, зображення, відео й аудіо в єдиній архітектурі. Базова модель

підтримує вікно контексту до 256 тисяч токенів, тоді як версія v2.5 розширює його до одного мільйона. Включення цих систем до бенчмарку дозволяє перевірити, чи здатні великі мовні моделі без явної підтримки української мови виконувати точне транскрибування змішаного мовлення, і чи не призводить їхня схильність до складних міркувань до генерування абстрактного тексту замість прямої фіксації почутого. [25]

Спеціалізовані рішення від OpenAI представлені моделями GPT-4o Transcribe та її зменшеною версією GPT-4o Mini Transcribe. Вони є окремими кінцевими точками універсальної моделі GPT-4o, які фокусуються саме на перетворенні мовлення в текст. Архітектура GPT-4o обробляє всі модальності єдиною нейронною мережею без проміжних перетворень, що мінімізує втрату акустичної інформації. [26]

Окремий клас масивних багатомовних систем становлять моделі наскрізного перекладу (Speech-to-Text Translation), найяскравішим представником яких є SeamlessM4T від Meta [27]. Ця архітектура спроєктована для виконання мультимодального перекладу в єдиному конвеєрі, що дає змогу перевірити гіпотезу прямого перекладу змішаного мовлення як альтернативи транскрибуванню. [27]

Попри неперевершене розуміння контексту, клас Audio-LLM має системний недолік. Якщо акустичний сигнал є нечітким, містить паузи або сильний акцент, потужний авторегресивний декодер може відірватися від аудіосигналу. У таких випадках модель починає галюцинувати, генеруючи технічно правильний, але фізично не вимовлений диктором текст, спираючись виключно на статистичну ймовірність наступного слова.

1.3.5 Комерційні API-моделі

Окрему групу становлять комерційні провайдери розпізнавання мовлення. Їхні внутрішні архітектури є пропрієтарними та закритими, тому їх

дослідження базується на офіційній документації та технічних звітах компаній. Включення цих систем у бенчмарк є критично важливим, оскільки саме вони найчастіше інтегруються у реальні бізнес-продукти.

Комерційні API відрізняються від відкритих моделей тим, що вони є не просто нейромережами, а комплексними конвеєрами обробки. Зазвичай такий конвеєр включає детектор активності голосу, модуль ідентифікації мови, акустичну модель та жорстку систему текстової нормалізації. Саме модуль ідентифікації мови часто стає причиною збоїв при змішаному мовленні: якщо система визначає українську мову на перших секундах аудіо, вона може примусово застосувати український словник до всього запису, повністю руйнуючи англійські терміни.

Досліджувані системи можна поділити на спеціалізовані рішення та інфраструктурні платформи:

ElevenLabs Scribe v2 підтримує адаптацію до контексту та дозволяє передавати системі словник специфічних термінів. Це критично важливо для IT-домену, оскільки такі підказки здатні примусово активувати розпізнавання слів на зразок deploy чи pipeline, навіть якщо вони звучать з українським акцентом. [28]

Deepgram Nova-3 позиціонується як рішення для складних корпоративних сценаріїв із наднизькою затримкою. Їхня архітектура оптимізована для високошвидкісного висновування, що робить її цікавим кандидатом для порівняння швидкодії з локальними моделями. [29]

AssemblyAI Universal-3 Pro спеціалізується на обробці складних випадків, зокрема рідкісних слів, галузевої термінології та мовного змішування. Компанія використовує архітектуру на базі Conformer, доповнену масивними навчальними даними. [30]

Soniox stt-async-v4 підтримує багатомовне розпізнавання та гнучке керування мовними підказками на рівні окремих запитів. [31]

Speechmatics пропонує платформи з підтримкою як пакетної обробки, так і розпізнавання в реальному часі. Їхня система відома агресивною текстовою нормалізацією, вплив якої на змішане мовлення є предметом даного дослідження. [32]

Інфраструктурні рішення від гігантів індустрії представлені платформами Azure Speech-to-Text від Microsoft та Chirp 3 від Google Cloud. Модель Chirp 3 побудована на базі універсальної мовної моделі з мільярдами параметрів. Обидві платформи пропонують автоматичну ідентифікацію мови, проте їхня поведінка на межах внутрішньореченнєвого перемикання кодів потребує незалежної оцінки. [33, 34, 35]

Також на ринку з'являються локальні гравці, зокрема бета-версія українського сервісу Interfaze STT [36], що фокусується на специфіці національного мовлення та адаптації до місцевого контексту.

1.4 Метрики оцінювання та нормалізація тексту

Для об'єктивного порівняння систем автоматичного розпізнавання мовлення традиційно використовуються метрики, що базуються на відстані Левенштейна між еталонною транскрипцією (*reference*) та результатом розпізнавання (*hypothesis*).

1.4.1 Word Error Rate (WER)

Основним індустріальним стандартом є WER - це відсоток помилок на рівні слів. WER обчислюється за такою формулою:

$$WER = \frac{S+D+I}{N},$$

де S (*Substitutions*) – кількість хибних замін слів;

D (*Deletions*) – кількість пропущених слів;

I (*Insertions*) – кількість зайвих, тобто вставлених, слів, яких не було в оригіналі;

N – загальна кількість слів в еталонній транскрипції.

Значення WER може перевищувати 100%, або 1,0 у випадку, коли модель генерує велику кількість зайвих слів, тобто якщо параметр *I* стає більшим за *N*. [37]

1.4.2 Character Error Rate (CER)

Хоча WER є стандартом, він має суттєві обмеження для флективних мов, зокрема української. Метрика WER однаково жорстко штрафує як повну втрату змісту слова, наприклад "програма" замість "база", так і незначну граматичну помилку в закінченні, наприклад "розробник" замість "розробники".

З огляду на це, додатково використовується метрика CER, яка обчислюється за тією ж логікою відстані Левенштейна, але базовою одиницею є не слово, а окремий символ:

$$CER = \frac{S+D+I}{N},$$

де S, D та I позначають кількість замін, вилучень і вставок на рівні символів, а N – загальну кількість символів в еталонній транскрипції.

CER дозволяє оцінити, наскільки коректно модель зловила фонетичну та графічну основу висловлювання. [37]

1.4.3 Code-Switching Word Error Rate (CS-WER)

Проблема обмеженості стандартної метрики WER для оцінювання змішаного мовлення активно піднімається в сучасних дослідженнях. Зокрема, розробники метрики *PolyWER* зазначають, що традиційний підхід є надто суворим для випадків перемикання кодів. Коли системи розпізнавання стикаються з іншомовними вставками, вони часто виводять їх у різних форматах: оригінальною графікою, фонетичною транслітерацією алфавітом

матричної мови або навіть перекладом. Якщо модель правильно розпізнає слово на слух, але записує його транслітерацією, базовий WER жорстко штрафує такий результат. Для подолання цього недоліку в метриці *PolyWER* пропонується зіставляти вивід моделі з кількома еталонними варіантами одночасно [38].

Спираючись на подібну логіку, для задачі українсько-англійського змішаного мовлення в цьому дослідженні метрика WER також недостатня. Загальний показник помилок у реченні з п'ятнадцяти слів може залишатися низьким, навіть якщо єдине ключове англійське технічне слово було розпізнане хибно або повністю пропущене. Оскільки фокусом роботи є здатність моделей розпізнавати саме іншомовні включення, тут застосовується спеціалізована метрика CS-WER.

Алгоритм обчислення CS-WER складається з таких етапів:

1. За допомогою алгоритмів вирівнювання (*alignment*) зіставляються еталонний текст і згенерована гіпотеза.
2. З еталонної транскрипції виокремлюються лише ті токени, які належать до іноземної мови, наприклад *pipeline*, *deploy*, *bug*.
3. Відсоток помилок розраховується виключно для цих іншомовних токенів відносно їх загальної кількості N_{en} у референсі:

$$CS\text{–}WER = \frac{S_{en}+D_{en}+I_{en}}{N_{en}}$$

де S_{en} – кількість замін серед англійських токенів;

D_{en} – кількість пропущених англійських токенів;

I_{en} – кількість зайвих англійських токенів;

N_{en} – загальна кількість англійських токенів в еталонній транскрипції.

Такий підхід дозволяє ізолювати проблему перемикання кодів від загальної якості розпізнавання української мови та показати, наскільки добре модель працює саме з технічною термінологією.

1.4.4 Real-Time Factor (RTF)

Для оцінки швидкодії та практичної придатності моделей, особливо для інтеграції у вебзастосунок, використовується часова метрика RTF. Вона визначається як відношення часу, витраченого моделлю на обробку аудіо, до фактичної тривалості самого аудіозапису:

$$RTF = \frac{T_{processing}}{T_{audio}}$$

де $T_{processing}$ – час, витрачений моделлю на транскрибування аудіо файлу;

T_{audio} – фактична тривалість аудіозапису.

Наприклад, якщо модель витрачає 2 секунди на транскрибування аудіофайлу тривалістю 10 секунд, показник RTF дорівнює:

$$RTF = \frac{2}{10} = 0,2$$

Значення $RTF < 1$ означає, що система здатна обробляти мовлення швидше, ніж воно звучить. Це є базовою вимогою для побудови streaming-систем або ASR-систем реального часу. Якщо RTF більше за 1, модель працює повільніше за реальний час і є придатною переважно для офлайн-обробки.

1.4.5 Проблема нормалізації тексту (ITN)

Окремим викликом під час обчислення перелічених метрик є нормалізація тексту. Різні ASR-моделі мають різні формати виведення. Наприклад, Whisper генерує текст із розділовими знаками та великими літерами, тоді як CTC-моделі, зокрема wav2vec2, зазвичай видають текст у нижньому регістрі без пунктуації.

Пряме порівняння необроблених виводів може призвести до штучного завищення показників помилок. Тому перед підрахунком WER, CER та CS-WER до всіх гіпотез і референсів застосовується уніфікований процес нормалізації:

- приведення тексту до нижнього регістру;

- видалення знаків пунктуації;
- нормалізація числівників;
- уніфікація символу українського апострофа, зокрема заміна символів типографського апострофу та гравісу на єдиний стандартний прямий апостроф.

Застосування єдиної процедури нормалізації забезпечує коректність порівняння моделей і дозволяє оцінювати саме якість розпізнавання мовлення, а не відмінності у форматуванні текстового виводу.

1.5 Теоретичні основи ідентифікації мовців та архітектура ECAPA-TDNN

Завдання автоматичної ідентифікації мовця (Speaker Identification) полягає у розпізнаванні особи диктора на основі акустичних характеристик його голосу. Фундаментальним підходом до вирішення цієї задачі є перетворення змінного за тривалістю мовленнєвого сигналу у компактний вектор фіксованої довжини (embedding або голосовий профіль), який описує унікальні біометричні та акустичні ознаки людини.

Сучасним стандартом і однією з найпотужніших архітектур у цій галузі є ECAPA-TDNN (Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Network), представлена дослідниками Гентського університету у 2020 році. [39] Вона є еволюційним розвитком класичних нейронних мереж із часовою затримкою (TDNN) та архітектури x-vector.

Ефективність архітектури ECAPA-TDNN базується на трьох головних компонентах:

- Squeeze-and-Excitation (канальна увага). Модель сама вчиться фільтрувати інформацію: вона підсилює ознаки, що характеризують унікальний голос людини, і пригнічує фонові шуми чи зміст слів.
- Multi-scale Feature Aggregation (багатомасштабне об'єднання). Алгоритм

не покладається лише на фінальний результат мережі, а збирає дані з різних етапів обробки. Це дозволяє їй "чути" одночасно і дрібні артикуляційні нюанси, і загальний тембр мовця.

- 1D Dilated Convolutions (розширені згортки). Вони різко розширюють "поле зору" моделі, дозволяючи їй аналізувати ширший аудіоконтекст без додаткового навантаження на процесор. Це робить систему легкою та швидкою.

Для встановлення того, чи належать два голосові фрагменти одній людині, отримані вектори (найчастіше розмірністю 192 або 512) порівнюються між собою за допомогою метрики косинусної схожості (cosine similarity). [39]

У контексті дослідження змішаного мовлення ідентифікація мовців та застосування ECAPA-TDNN становить особливий інтерес. Під час переходу з рідної матричної мови на іноземну (або навпаки) диктор часто підсвідомо змінює артикуляційну базу, темп мовлення, висоту тону та просодику для адаптації до іншомовної фонетики. Здатність архітектури ECAPA-TDNN витягувати глибинні, незалежні від мови (language-agnostic) акустичні ознаки є критично важливою для того, щоб запобігти хибній класифікації одного й того ж диктора як двох різних людей під час внутрішньореченнєвого перемикання кодів.

РОЗДІЛ 2: СТВОРЕННЯ ТА АНАЛІЗ ДАТАСЕТУ

Для проведення об’єктивного порівняння сучасних ASR-систем на задачі українсько-англійського перемикання кодів було розроблено спеціалізований корпус мовлення *uk-en-code-mixed-asr-2h*. Цей розділ описує методологію його створення, технічні характеристики, структуру метаданих, а також містить детальний статистичний і лінгвістичний аналіз зібраного матеріалу.

2.1 Концепція та дизайн корпусу

Аналіз наявних ресурсів, наведений у підрозділі 1.2, показав, що у відкритому доступі практично відсутні якісні датасети змішаного українсько-англійського мовлення. Для заповнення цієї прогалини було прийнято рішення створити власний корпус, дизайн якого підпорядкований головній меті роботи – тестуванню ASR-моделей.

Ключовим методологічним рішенням став вибір формату read-speech, тобто начитаного мовлення, замість спонтанного. Спонтанне мовлення є більш природним, проте воно містить значну кількість хезитацій, наприклад звуків "е-е" або "м-м", обірваних слів, перебивань та фонового шуму. Для створення еталонного бенчмарку такі фактори є небажаними, оскільки вони вносять неоднозначність у референсну транскрипцію. Модель може отримати нижчий результат, тобто вищий WER, не через те, що неправильно розпізнала термін, а через те, що транскрибатор не записав звук вагання мовця.

Начитане мовлення дозволяє здійснювати повний контроль над текстом і гарантує, що референсна транскрипція максимально точно відповідає акустичному сигналу. Це є критичною вимогою для точного обчислення метрики CS-WER на межах перемикання мов.

Домен корпусу - це інформаційні технології. Саме в цій сфері українсько-англійське внутрішньореченнєве перемикання кодів є особливо

поширеним та органічним. Щоб забезпечити лексичну різноманітність, тексти для начитування були згенеровані та відредаговані вручну з охопленням 16 різних тематичних категорій, серед яких: DevOps, QA, Frontend, Backend, Databases, Cloud Infrastructure та Machine Learning. Це дозволило включити в корпус широкий спектр технічних термінів: від поширених, таких як API, баг, деплой, до вузькоспеціалізованих, таких як garbage collector, index lookup тощо.

2.2 Протокол збору та технічні характеристики

Запис аудіоматеріалу здійснювався одним диктором, чоловічим голосом, у контрольованих акустичних умовах, що дозволяє мінімізувати вплив реверберації та фонового шуму на результати бенчмарку. Інструкція для запису передбачала природний темп мовлення, збереження звичної інтонації без театральності та чітку артикуляцію як українських слів, так і англійських термінів без штучного акцентування останніх.

2.2.1 Технічні специфікації аудіо:

- Частота дискретизації (Sample Rate): 48 kHz.
- Кількість каналів: 1, Mono.
- Формат файлів: OGG Opus.

Вибір кодека Opus зумовлений його високою ефективністю стиснення мовленнєвих даних при збереженні прозорої акустичної якості, що є стандартом для сучасних систем передачі голосу.

2.2.2 Структура датасету та метадані:

Корпус сформований як готовий до використання Hugging Face Dataset. Кожен аудіозапис, тобто утеранс, супроводжується набором метаданих для полегшення автоматизованого тестування. Структура запису включає:

- id – унікальний ідентифікатор висловлювання (наприклад, 1_azure);
- file_name – відносний шлях до аудіофайлу;

- transcription – еталонна (reference) транскрипція зі збереженням оригінального змішування кирилиці та латиниці;
- duration –тривалість запису в секундах;
- language –загальний мовний маркер (mixed для змішаного мовлення або uk для монолінгвального контрольного зразка);
- cs_type – тип перемикання кодів (intra для внутрішньореченнєвого перемикання або mono для монолінгвального);
- speaker_id – анонімізований ідентифікатор диктора (наприклад, Valera або ValeraAlt);
- en_level – рівень володіння англійською мовою диктора (наприклад, C1);
- topic – приналежність до однієї з 16 тематичних IT-категорій (доменів);
- translation_en – повний переклад висловлювання англійською мовою, який додатково використовується для оцінювання STT-моделей із функцією перекладу.

Важливою архітектурною особливістю датасету uk-en-code-mixed-asr-2h є відсутність поділу на навчальну, валідаційну та тестову вибірки, тобто train/validation/test split. Увесь зібраний обсяг даних виступає єдиним цілісним тестовим набором. Датасет не призначений для fine-tuning акустичних моделей, бо його мета - це zero-shot оцінювання. Тобто моделі тестуються "з коробки", що дозволяє об’єктивно виміряти їхню здатність генералізувати знання на змішане мовлення, яке вони, ймовірно, не бачили під час попереднього навчання.

Крім того, поточний обсяг корпусу (трохи більше двох годин) є об’єктивно недостатнім для повноцінного донавчання сучасних нейромереж без ризику перенавчання на акустичні особливості єдиного диктора. У найкращому випадку наявний масив даних можна використати для персоналізованої адаптації моделі під голос конкретного користувача або для точкового

налаштування словника термінів. Проте експерименти з таким донавчанням, а також фізичне масштабування датасету для формування повноцінної навчальної вибірки, виходять за межі поточної роботи і розглядаються як перспективний напрям подальших досліджень.

2.3 Статистичний та лінгвістичний аналіз корпусу

Для підтвердження репрезентативності зібраного корпусу було проведено його детальний дослідницький аналіз.

2.3.1 Базові метрики обсягу:

Загалом датасет містить 448 утерансів. Загальна тривалість чистого мовлення становить 7595 секунд, що дорівнює 126,6 хвилини, або 2,11 години.

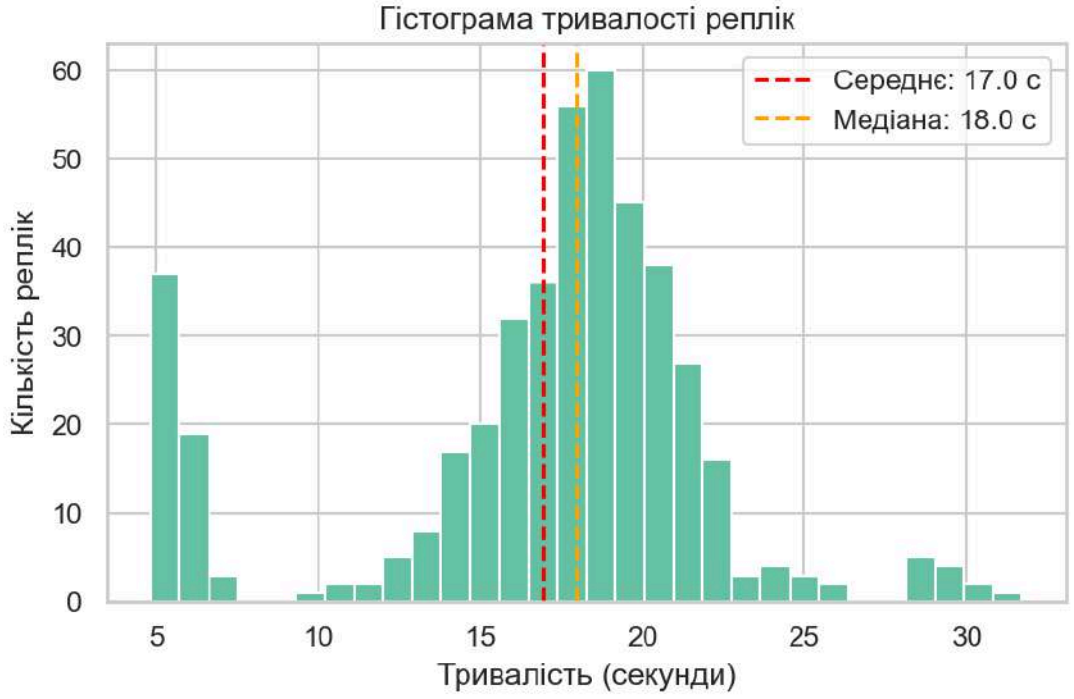

Рисунок 2.3.1 – Гістограма тривалості реплік

Тривалість окремих записів коливається від 4,8 секунди до 31,7 секунди. Середня тривалість одного утеранса становить 17,0 секунди. Цей показник є оптимальним для тестування сучасних Encoder-Decoder моделей та Audio-LLM: 17 секунд забезпечують достатній мовний контекст для роботи механізмів уваги, але водночас не перевантажують контекстне вікно і не викликають

проблем із нестачею пам’яті, які часто виникають під час обробки довгих аудіо.

2.3.2 Текстова статистика та швидкість мовлення:

У середньому одне висловлювання складається з 40,7 слова, або близько 276 символів. Середня швидкість мовлення диктора становить 2,4 слова на секунду, що відповідає нормальному темпу ділової розмови.

2.3.3 Характеристики перемикання кодів:

Аналіз розподілу за типом мовного змішування демонструє домінування цільового явища: 99,6 % корпусу, тобто 446 зразків із 448, містять intra-sentential code-switching, тобто перемикання всередині речення, також є intra-word приклади: “тригерив”, “evict-нув”, “underestimate-ує” тощо.

Лише 2 зразки (0,4 % від загального обсягу) є повністю монолінгвальними українськими реченнями, які використовуються у бенчмарку як контрольний baseline. Прикладом такого суто україномовного запису (ID: 95_dotnet_short) є фраза: "*Глянув на той декоратор для кешування, який обговорювали на архітектурному рев’ю. Він кешував результат методу, але ключ будувався тільки з назви методу без параметрів, і різні виклики з різними аргументами повертали однаковий кешований результат. Переписав генерацію ключа з урахуванням всіх параметрів через серіалізацію аргументів.*"

Як видно з прикладу, усі технічні поняття тут є адаптованими запозиченнями і записані виключно кирилицею. Наявність таких контрольних зразків є важливою, оскільки вона дозволяє виміряти базову якість розпізнавання українського IT-контексту моделлю до того, як вона зіткнеться з проблемою прямого перемикання кодів та латиницею.

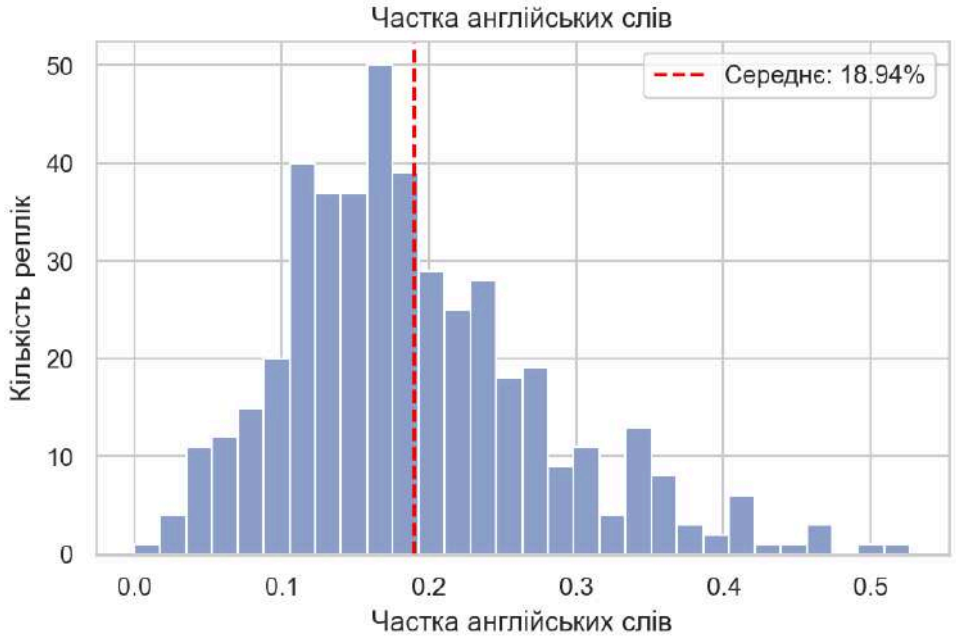

Рисунок 2.3.3 – Частка англійських слів

Лінгвістичний аналіз показав, що в середньому кожна фраза в корпусі містить 18,9 % англійських слів, що становить 21,9 % символів латиницею від загальної довжини тексту. Таке співвідношення є репрезентативним для реальної робочої комунікації розробників: зазвичай це 1-3 вузькоспеціалізовані англійські терміни, вбудовані в синтаксичну структуру українського речення. Лише 3 зразки в усьому корпусі містять понад 50 % латиниці, тобто є випадками сильного змішування.

2.3.4 Словниковий аналіз:

Загальна кількість входжень англійських слів у корпусі становить 3207. З них було виокремлено 1251 унікальний англійський токен. Наявність такого широкого словника гарантує, що ASR-моделі під час бенчмарку тестуються на різноманітній лексиці, а не лише на кількох найчастотніших термінах.

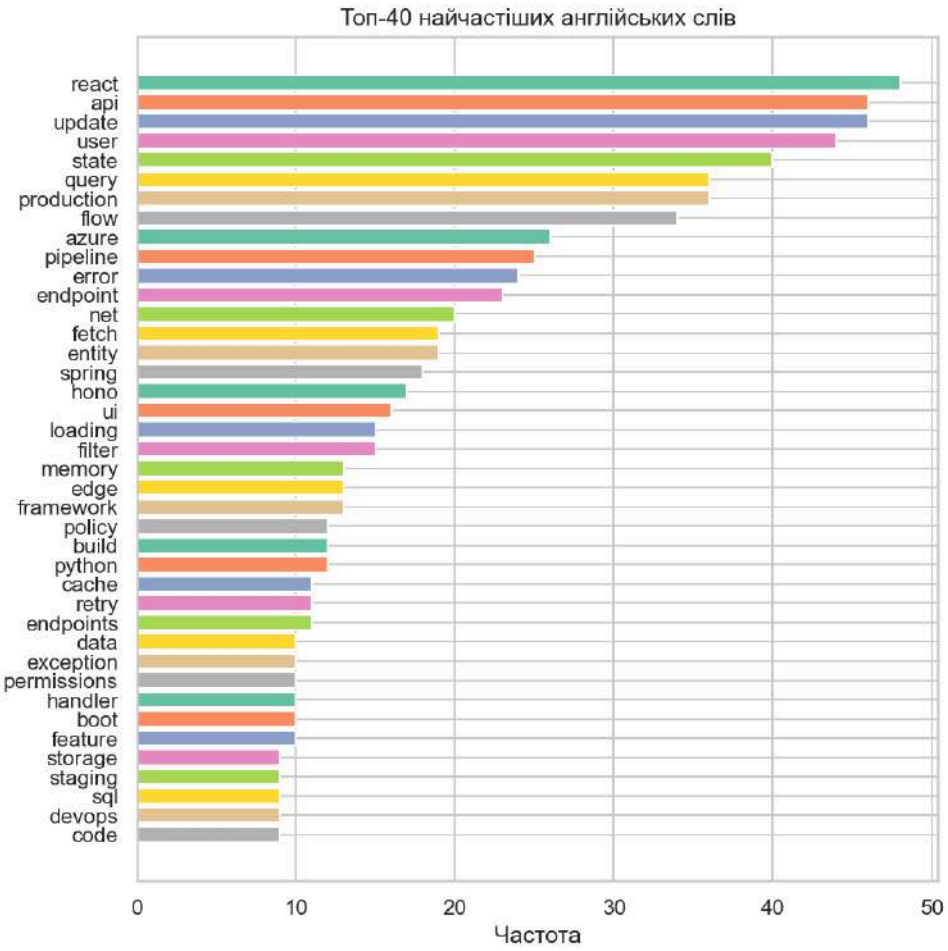

Рисунок 2.3.4 – Топ-40 найчастіших англійських слів у датасеті

Ці статистичні дані підтверджують, що зібраний датасет має достатню щільність перемикання кодів та лексичну варіативність для того, щоб виступати надійним інструментом оцінювання мовних моделей.

2.4 Обмеження зібраного корпусу

Попри те, що uk-en-code-mixed-asr-2h розв’язує проблему відсутності бенчмарків для українського IT-домену, він має низку об’єктивних обмежень, які необхідно враховувати під час інтерпретації результатів експериментів.

2.4.1 Один біологічний мовець

Увесь обсяг аудіо записаний одним диктором. Хоча до корпусу навмисно включено контрольну мікровибірку з 6 реплік (ідентифікатор ValeraAlt), які

були записані тим самим диктором зі штучно зміненим, більш низьким та грубим тембром голосу для перевірки базової акустичної варіативності, загальна однорідність даних зберігається.

Через це датасет ідеально підходить для оцінювання мовних моделей на предмет розпізнавання термінології та перемикання алфавітів, але він не дозволяє повноцінно перевірити стійкість ASR-систем до справжніх гендерних відмінностей, різноманіття регіональних акцентів або багатомовцевого середовища.

2.4.2 Студійна якість та начитане мовлення.

Як було зазначено в описі дизайну корпусу, записи не містять фонового шуму, ефекту відлуння чи накладання голосів, тобто overlapping speech, які є типовими для реальних мітингів у Zoom або Google Meet. Крім того, у корпусі відсутні обмовки та перебивання.

Зазначені обмеження не зменшують наукової цінності датасету, оскільки першочергова роль – створення контрольованих умов для базового порівняння архітектур. Проте для створення повністю універсальної системи розпізнавання мовлення в майбутньому знадобиться збір розширеного корпусу багатоканального спонтанного мовлення.

2.5 Словник зворотної текстової нормалізації

Оскільки головною проблемою розпізнавання українсько-англійського змішаного мовлення є транслітерація, невід'ємною частиною інфраструктури датасету стало створення словника зворотної текстової нормалізації. Цей словник зіставляє канонічні англійські терміни з масивом їхніх можливих кириличних репрезентацій.

Критично важливою методологічною особливістю цього словника є те, що він не базується на апостеріорному аналізі виводів ASR-моделей.

Використання реальних помилок протестованих систем для формування словника призвело б до витоку даних і штучного завищення результатів бенчмарку. Натомість мапа транслітерацій формувалася апріорі як лінгвістичне передбачення усіх теоретично можливих варіантів вимови та запису термінів. В основу прогнозування лягли правила практичної транслітерації, морфологічні особливості (потенційне додавання моделями дефісів чи пробілів) та аналіз типових артикуляційних патернів україномовних IT-спеціалістів. [40]

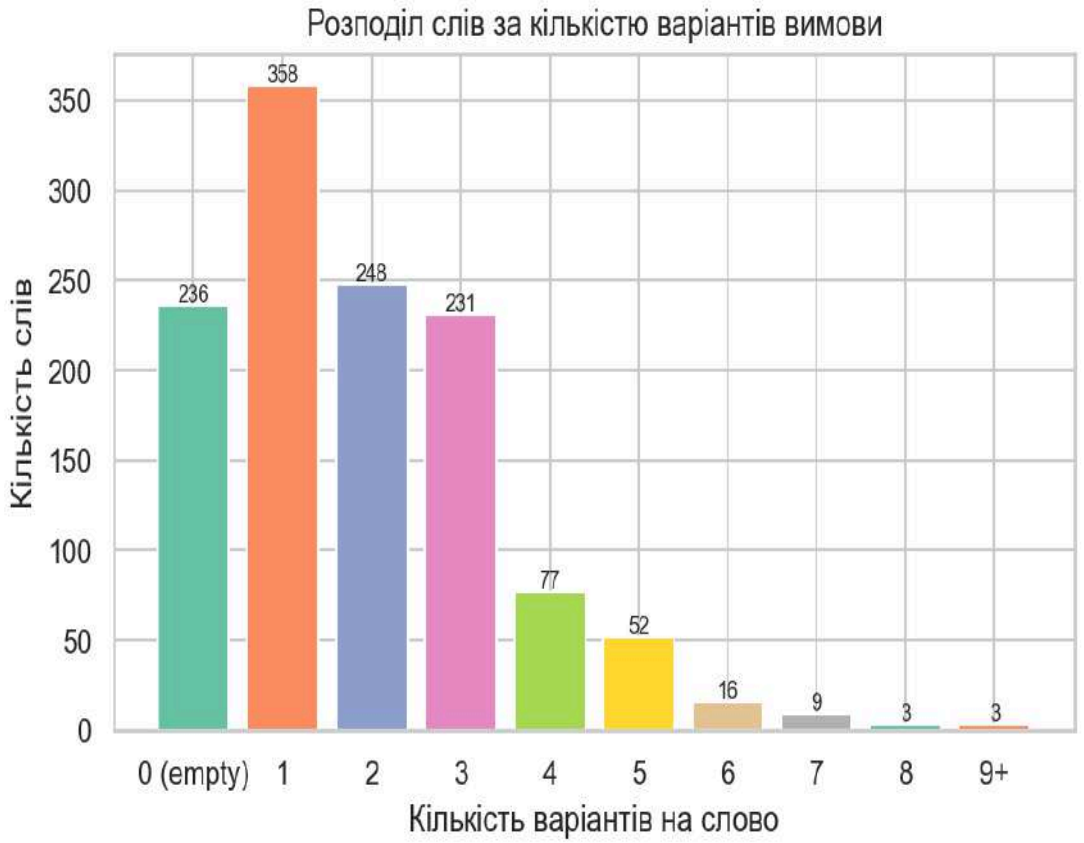

Рисунок 2.5 – Розподіл англійських термінів за кількістю згенерованих варіантів вимови/транслітерації у словнику

У результаті було сформовано масив, де на кожен англійський термін припадає в середньому 2,9 варіанта кириличного запису. Як видно з гістограми розподілу (Рисунок 2.5), найбільшу групу (358 слів) становлять терміни з одним однозначним варіантом транслітерації. Однак значна частина словника має від 2 до 4 варіацій. Окрему групу (236 слів, категорія "0") становлять терміни, для

яких кириличні відповідники не генерувались з різних причин. Деякі слова мають прямий відповідник або транслітерація вже є іншим словом.

Наявність довгого "хвоста" розподілу (від 5 до 9+ варіантів) зумовлена високою фонетичною неоднозначністю складних термінів. Оскільки поточна реалізація ITN-мапи базується на "плоскому" словнику та використанні регулярних виразів для пошуку збігів, намагання охопити всі можливі префікси, суфікси та відмінкові закінчення призводить до комбінаторного перевантаження.

Показовими прикладами такого апріорного генерування варіативності є такі словникові статті:

- "thread": "тред", "сред", "фред"
- "threshold": "трешолд", "срешолд", "фрешолд"
- "thundering": "тандерінг", "сандерінг", "фандерінг"
- "prometheus": "прометеус", "промітіус", "промітеус"

Ці приклади ілюструють чотири основні виміри фонетичної та орфографічної варіативності, закладені в мапу:

1. Адаптація відсутніх фонем: англійський глухий міжзубний приголосний /θ/ (дограф th) не має відповідника в українській фонетичній системі. Залежно від індивідуального акценту диктора та інтерпретації цього звуку акустичною моделлю, він може транскрибуватися трьома різними кириличними літерами: "т", "с" або "ф" (тред / сред / фред).
2. Акустична неоднозначність та кінцеве оглушення: в усному мовленні неносіїв мови часто спрацьовує правило оглушення дзвінких приголосних у кінці слова. Через це IT-термін thread (/θrɛd/ – потік виконання) фонетично майже повністю зливається зі словом threat (/θrɛt/ – загроза). Така

акустична близькість заплутує ASR-системи, змушуючи їх або обирати хибний семантичний кандидат, або генерувати неоднозначну транслітерацію.

3. Варіативність голосних звуків: англійські довгі голосні та нестандартні для кирилиці дифтонги передаються системами розпізнавання вкрай нестабільно. Наприклад, звук /iː/ у слові Prometheus (/prəˈmiːθiəs/) може бути фонетично наближеним до оригіналу у вигляді транслітерацій “промітіус” або “промітеус”.

4. Візуальне прочитання: несвідоме спотворення вимови за принципом "читаю так, як пишеться", ігноруючи правила англійської орфоепії. Це надзвичайно поширене явище у професійному сленгу, коли замість правильного фонетичного запису утворюється візуальна калька (наприклад, форма “прометеус”).

Такий лінгвістично обґрунтований, предиктивний підхід до формування словника дозволив створити надійний, незалежний від конкретних моделей інструмент для об'єктивного вимірювання показника CS-WER, який компенсує недоліки токенізаторів без прямого втручання в акустичну оцінку систем.

РОЗДІЛ 3: БЕНЧМАРК ASR-МОДЕЛЕЙ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

У цьому розділі подано результати порівняльного тестування сучасних систем автоматичного розпізнавання мовлення на створеному корпусі uk-en-code-mixed-asr-2h. Основна мета експерименту полягає не лише у визначенні моделі з найнижчим загальним WER, а й у глибшому аналізі того, як різні архітектури працюють із перемиканням кодів між українською та англійською мовами.

Окрему увагу приділено трьом аспектам: точності розпізнавання англійських термінів у межах українського мовлення, схильності моделей до галюцинацій та перевірці ефективності зворотної текстової нормалізації. Такий підхід дозволяє оцінити не лише загальну якість транскрибування, а й практичну придатність моделей для українського ІТ-контексту, де змішування мов є типовим явищем.

3.1 Методологія тестування

Для забезпечення репрезентативності та об’єктивності дослідження було побудовано єдиний пайплайн тестування. Він охоплює 35 моделей (49 конфігурацій) різних класів, уніфіковану процедуру текстової нормалізації та двоетапний алгоритм підрахунку помилок. Експерименти з локальними моделями проводилися на базі графічного прискорювача NVIDIA GeForce RTX 3080 (10 ГБ відеопам’яті).

Такий підхід дає змогу порівнювати між собою моделі з різною архітектурою, різними форматами виведення тексту та різними механізмами мовного декодування.

3.1.1 Відбір та класифікація протестованих моделей

Усі протестовані моделі можна умовно поділити на три основні групи.

Першу групу становлять локальні відкриті моделі, які можна запускати без використання комерційних API. До неї увійшли моделі сімейства Whisper від OpenAI [15]: від полегшених версій tiny, base, small і medium до потужніших large-v2, large-v3 та оптимізованої large-v3-turbo.

Також до цієї групи було включено CTC-моделі та конформери, зокрема багатомовні системи mms-1b-all [11], wav2vec2-xlsr-multilingual-56 [13], а також спеціалізовані українські моделі Yehor/w2v-xls-r-uk, Yehor/w2v-bert-uk-v2.1 [14] та Ukrainian Pods Conformer [19]. Окремо тестувалися моделі з екосистеми NVIDIA NeMo [17]: Citrinet_1024, Parakeet-TDT-0.6b-v3 [18] та Canary-1b-v2. До локальних систем також увійшли офлайн-рішення Vosk uk v3 [20] у стандартній та small-версіях, а також модель наскрізного перекладу SeamlessM4T-v2-large [27].

Другу групу становлять комерційні спеціалізовані ASR API. До неї увійшли хмарні рішення Azure Speech від Microsoft [33] та Chirp 3 від Google Cloud [34, 35], а також спеціалізовані ASR-провайдери: Deepgram Nova-3 [29], AssemblyAI Universal-3 Pro [30], Soniox stt-async-v4 [31], ElevenLabs Scribe v1 і v2 [28], Speechmatics STT [32] та бета-версія сервісу Interfaze STT [36].

Третю групу становлять мультимодальні Audio-LLM. У межах цієї категорії було протестовано моделі сімейства Gemini [22, 23, 24], зокрема легші версії Gemini Flash Lite, флагманські Gemini Pro та Gemini Flash. Також до тестування було включено GPT-4o-transcribe і GPT-4o-mini-transcribe [26], а також моделі Voxtral-Mini-3B, Voxtral-Small-24B [21], MiMo-v2.5 та MiMo-v2-Omni [25].

Локальні моделі запускалися на обладнанні з GPU з використанням фреймворків PyTorch, Transformers та NeMo. Комерційні API та Audio-LLM тестувалися через REST API із застосуванням відповідних SDK.

Для систем, які підтримують мовні підказки, тестування проводилося у

кількох режимах: з автоматичним визначенням мови, з примусовим встановленням української мови та, де це було можливо, з комбінованими підказками для української й англійської мов.

3.1.2 Процедура текстової нормалізації

Оскільки різні ASR-системи генерують текст у різних форматах, пряме порівняння необроблених гіпотез з еталонними транскрипціями могло б штучно завищити WER. Наприклад, Whisper може повертати текст із великими літерами та пунктуацією, тоді як wav2vec2 генерує текст у нижньому регістрі без розділових знаків.

Щоб усунути вплив форматування, до кожної гіпотези та кожного еталонного тексту застосовувався єдиний конвеєр нормалізації. Спочатку весь текст переводиться у нижній регістр. Далі видаляється пунктуація: крапки, коми, знаки питання, знаки оклику, дужки та інші символи, які не змінюють фонетичного змісту висловлювання.

Окремим важливим етапом була уніфікація українського апострофа. У цифрових текстах апостроф може бути представлений різними символами: типографічним апострофом, машинописним апострофом, гравісом або спеціальним модифікатором літери. У межах пайплайну всі ці варіанти зводені до одного стандартного символу.

Після цього виконувалося очищення пробілів: множинні пробіли та символи табуляції замінені одним пробілом, а зайві пробіли на початку й у кінці рядка видалені.

Лише після проходження цієї нормалізації тексти передаються до алгоритму вирівнювання на базі бібліотеки jiwer. Це дозволяє порівнювати саме зміст транскрипції, а не поверхневі відмінності у форматуванні.

Для окремого оцінювання точності розпізнавання англійських термінів і

українського контексту було застосовано алгоритм мовного тегування слів.

3.1.3 Алгоритм обчислення метрик та ізоляції CS-WER

На першому етапі кожному слову в еталонній транскрипції присвоювався мовний тег: EN або UK. Слово отримувало тег EN у двох випадках. По-перше, якщо воно містило хоча б один символ латинського алфавіту, наприклад Azure, API або QA. По-друге, якщо воно входило до створеної бази зворотної транслітерації як український варіант прочитання англійського терміна, наприклад "ажур", "ей пі ай" або "деплой".

Усі інші слова позначалися як UK. Після цього бібліотека jiwer виконувала вирівнювання між еталонною транскрипцією та гіпотезою моделі. Помилки заміни, пропуску та вставки зараховувалися до відповідних лічильників залежно від мовного тегу еталонного слова.

Для вставок, тобто зайвих слів, згенерованих моделлю, застосовується додаткове правило. Якщо вставлене слово містить латинські літери, воно зараховується як помилка в англійському сегменті. Якщо латиниці не було, вставка зараховується як помилка в українському сегменті.

Такий підхід дозволяє чітко визначити, де саме модель припускається помилок: у розпізнаванні українського мовлення, у роботі з англійськими термінами або через надмірну генерацію зайвого тексту. Крім того, уніфікована нормалізація мінімізує вплив орфографії та пунктуації на фінальний результат.

3.2 Аналіз результатів прямого транскрибування

У межах першого етапу бенчмарку було протестовано на необроблених аудіоданих без застосування зворотної текстової нормалізації. Метою цього етапу було визначити, наскільки добре сучасні ASR-системи розпізнають українсько-англійське змішане мовлення без додаткової постобробки.

Повна таблиця результатів наведена в додатку А. У цьому підрозділі розглядаються ключові закономірності за основними класами архітектур.

3.2.1 Мультимодальні Audio-LLM: лідери бенчмарку та проблема галюцинацій

Найкращі результати на задачі транскрибування змішаного мовлення продемонстрували мультимодальні великі мовні моделі. Перше місце в бенчмарку посіла Gemini 3.1 Pro із загальним показником WER 0,1359 без застосування ITN.

Її висока якість розпізнавання українського мовлення підтверджується найнижчим показником помилок на українських словах 0,0742 (тобто близько 7,4 %). Високі результати також показали Gemini 3 Flash із WER 0,1442 та GPT-4o Transcribe із WER 0,1730.

Перевага цього класу моделей пояснюється наявністю потужного мовного декодера. На відміну від класичних CTC- або Transducer-систем, Audio-LLM краще враховують контекст речення й можуть точніше передбачати появу англійського технічного терміна в українському висловлюванні.

Водночас мультимодальні мовні моделі мають суттєвий недолік – схильність до надмірної генерації тексту (галюцинацій) та зациклення. Показовим прикладом фонетичної галюцинації стала модель MiMo-v2.5 від Xiaomi. Її загальний WER склав 0,9377 (майже 93,8 % помилок), а показник помилок на українських словах досяг 0,9145. Оскільки моделі сім’ї MiMo офіційно не підтримують розпізнавання української мови, вони намагаються генерувати текст на основі акустичної схожості з відомими їм мовами, що призводить до катастрофічного падіння якості та генерації довгих беззмістовних речень.

Інший тип надмірної генерації продемонстрували новітні моделі сім’ї Gemini, які оснащені механізмом внутрішніх міркувань (reasoning). Без задання

жорстких лімітів ці моделі часто потрапляють у нескінченний цикл. Яскравим прикладом такої поведінки є процес генерації моделі Gemini 3.1 Pro у середовищі Google Vertex. Замість транскрибування аудіо система почала генерувати нескінченний ланцюг внутрішніх роздумів англійською мовою: "Transcribing Ukrainian Verbatim", "Capturing Ukrainian Accurately", "Finalizing Ukrainian Transcript" тощо. Модель повністю зациклилася на декларації своїх намірів ідеально точно передати українське мовлення, однак до фактичного виведення розпізнаного тексту так і не перейшла.

Ця архітектурна проблема успішно розв'язується лише заданням параметра температури на рівні 0 та встановленням жорсткого обмеження бюджету токенів на режим міркування (наприклад, не більше ніж 200 токенів). Використання температури на рівні 1,0 не лише збільшує кількість помилок у транскрипціях, а й призводить до системних збоїв під час API-запитів (перевищення часу очікування, або timeout) та проблем із локальним обробленням відповіді. Крім того, виявлено технічну особливість: якщо кількість токенів міркування моделі більша або дорівнює максимальній кількості токенів, виділених на генерацію (max_tokens), то сама транскрипція повертається порожньою.

Водночас моделі сім’ї MiMo продемонстрували протилежну проблему: етап міркування в них завершується коректно, однак після цього починається нескінченне зациклення самого тексту відповіді. Саме через цей дефект перші ітерації тестування моделей MiMo показували аномальний рівень помилок (WER > 3,0).

Це означає, що модель генерує значно більше слів, ніж було в оригінальному аудіо. У таких випадках система фактично переходить від транскрибування до довільної генерації тексту, особливо на фонових шумах або нечітких фрагментах мовлення.

3.2.2 Комерційні ASR API

Серед спеціалізованих комерційних ASR-систем найкращий результат показала ElevenLabs Scribe v2, яка посіла друге місце в загальному рейтингу з WER 0,1414. Стабільно високі результати також продемонстрували Soniox Async v4 із WER 0,1756 та Google Chirp 3 із WER 0,1846.

Водночас варто зазначити нестабільність роботи моделі Google Chirp 3 на наданому наборі даних: під час тестування було зафіксовано 15 збоїв (стовпець Fails у загальній таблиці результатів). У цих випадках API повертало порожню відповідь або внутрішню помилку сервера. Наявність таких збоїв свідчить про потенційні проблеми з обробкою специфічних аудіофрагментів або надмірну чутливість внутрішніх фільтрів платформи, що варто враховувати під час інтеграції цього рішення у продукційні системи.

Ці системи краще пристосовані до реальних умов транскрибування, ніж більшість локальних моделей. Вони демонструють відносно низький загальний рівень помилок і краще працюють із фрагментами, де українське мовлення поєднується з англійськими технічними термінами.

Водночас популярні корпоративні рішення виявилися менш ефективними саме на задачі внутрішньореченнєвого перемикання кодів (intra-sentential code-switching). Наприклад, Azure Speech у конфігурації uk-UA показав WER 0,4509. Додавання явного англійського мовного тегу в режимі uk-UA + en-US LID не покращило результат, а навпаки погіршило його до 0,4905.

Це свідчить про те, що класичні механізми визначення мови на рівні висловлювання не завжди здатні стабільно працювати з перемиканням кодів. Вони можуть добре визначати основну мову аудіо, але не завжди коректно обробляють короткі англійські вставки всередині українського речення.

3.2.3 Локальні відкриті моделі та обмеження монолінгвальних систем

Серед відкритих моделей, придатних для локального запуску, найкращий результат продемонструвала Whisper Large v3 Turbo. У режимі примусового українського мовного тегу її WER склав 0,1953.

Цей результат робить Whisper Large v3 Turbo найсильнішою локальною моделлю серед протестованих систем. Вона забезпечує прийнятну якість транскрибування без необхідності передавати аудіо до зовнішніх API, що є важливим для сценаріїв із підвищеними вимогами до приватності та безпеки даних.

Модель Parakeet TDT 0.6B v3 від NVIDIA показала задовільну якість на українському тексті: показник помилок на українських словах становив 0,1533. Однак на англійських термінах вона майже повністю провалилася: показник помилок на англійських словах (EN-WER) досяг 0,9018.

Особливо показовими є результати спеціалізованих українських моделей, зокрема W2V-BERT Ukrainian v2.1 та Ukrainian Pods Conformer. Їхній загальний WER становив 0,5845 та 0,4763 відповідно. Для цих моделей показник помилок на англійських термінах перевищував 1,0 (1,0034 та 1,0097 відповідно).

Це означає, що системи не просто пропускали англійські слова. Вони часто намагалися фонетично адаптувати їх до українського словника, генеруючи нерелевантні або неіснуючі слова. Така поведінка є типовою для монолінгвальних моделей, які не мають достатньої підтримки змішаного мовлення.

Отримані результати підтверджують, що монолінгвальні ASR-моделі мають суттєві обмеження для сучасного українського IT-контексту. У середовищах, де англійські терміни регулярно з’являються всередині українських речень, такі системи не забезпечують достатньої стабільності.

3.3 Мапа транслітерацій та зворотна текстова нормалізація

Попри відносно низькі показники загального WER у моделей-лідерів, аналіз помилок на англійських словах виявив суттєву проблему. Для всіх протестованих моделей помилки на англійських термінах були значно вищими, ніж помилки на українських словах.

Наприклад, у Gemini 3.1 Pro показник помилок на українських словах становив 0,0742, тоді як показник помилок на англійських термінах досягав 0,4256. Тобто англійські терміни розпізнаються приблизно в 5,7 раза гірше, ніж український контекст.

Детальний аналіз транскрипцій показав, що значна частина таких помилок не була пов’язана з неправильним акустичним розпізнаванням. У багатьох випадках модель правильно визначала англійський термін на слух, але записувала його кирилицею відповідно до української вимови. Наприклад, deploy міг бути записаний як "деплой", а pipeline – як "пайплайн".

Оскільки еталонна транскрипція містила канонічне англійське написання, стандартний WER класифікував такі випадки як повноцінні помилки заміни. Через це якість моделей у роботі з англійськими термінами виглядала нижчою, ніж вона фактично була на акустичному рівні.

Щоб усунути цю проблему та точніше оцінити якість розпізнавання, було розроблено модуль зворотної текстової нормалізації (ITN). Його завдання полягає в тому, щоб перетворювати поширені українські фонетичні записи англійських IT-термінів на їхнє канонічне англійське написання.

3.3.1 Побудова мапи транслітерацій

Основою модуля стала спеціально сформована мапа зворотної транслітерації. Вона містить 1240 англійських IT-термінів, для кожного з яких було зібрано поширені українські варіанти вимови, транслітерації та

морфологічних форм. Загалом база охоплює 3606 варіантів написання. Наприклад, для англійського слова update було додано такі українські форми: "апдейт", "ап-дейт", "ап дейт", "апдейту", "апдейтом". У результаті система містить 4838 унікальних пар відповідності між українськими фонетичними формами та канонічними англійськими термінами. Така база дає змогу автоматично нормалізувати значну частину типових англійських технічних слів, які в ASR-виводі часто з’являються кирилицею.

Програмна реалізація розробленого алгоритму ITN виконується у три послідовні кроки:

1. Побудова інвертованого словника: на основі зібраної бази формується обернена структура даних (inverse map), де ключами виступають українські транслітерації ("деплой", "ажур", "пасворд"), а значеннями – канонічні англійські терміни.
2. Компіляція регулярного виразу: усі українські ключі зі словника об'єднуються в єдиний комплексний регулярний вираз (regex). Для запобігання хибним частковим замінам ключі обов'язково сортуються за зменшенням довжини (наприклад, форма "ап дейт" перевіряється раніше за "ап"). Для точного пошуку ізольованих термінів шаблон обгортається операторами меж слова.
3. Нормалізація тексту: як еталонна транскрипція (reference), так і згенерована гіпотеза (prediction) проходять через функцію підстановки. Регулярний вираз виявляє кириличні форми та замінює їх на латинські, після чого обчислюється фінальний показник WER.

Ефективність цього підходу пояснюється тим, що ASR-моделі часто правильно розпізнають англійський термін фонетично, але записують його кирилицею. Пошук та заміна таких форм ще до етапу вирівнювання зрівнює гіпотезу з еталоном і скасовує несправедливі штрафи метрики WER.

З математичного погляду створена мапа зворотної транслітерації функціонує як базова реалізація детермінованого скінченного перетворювача (FST). Кожна пара відповідності є шляхом у графі перетворень. Використання регулярних виразів та пошуку за хеш-таблицею забезпечує абсолютну детермінованість: алгоритм виконує заміну лише тоді, коли знаходить точний збіг. Це повністю виключає ризик генерації галюцинацій та непередбачуваних модифікацій тексту, що є критичною перевагою порівняно з авторегресивними мовними моделями в задачах транскрибування.

3.3.2 Результати застосування зворотної текстової нормалізації

Застосування фільтра ITN до результатів бенчмарку на другому етапі показало суттєве покращення для переважної більшості протестованих моделей (41 із 49). Середнє абсолютне покращення загального WER для цих моделей становило 0,0237, що свідчить про ефективність словникової нормалізації в усуненні формальних орфографічних помилок.

Водночас для низки моделей (зокрема Voxtral, AssemblyAI, GPT-4o Mini, Canary, Speechmatics та MiMo) після нормалізації спостерігалося незначне погіршення загального показника WER (до +0,0268) та CER. Найбільша деградація зафіксована в моделі MiMo-v2.5, де загальний WER зріс на 0,0636, а показник помилок на англійських словах (EN-WER) залишився практично незмінним. Детальний аналіз показав, що таке погіршення виникає через хибні спрацьовування (false positives) алгоритму регулярних виразів на сильно спотвореному або галюцинованому тексті. Коли модель генерує потік беззмістовних слів (як у випадку MiMo), алгоритм ITN може випадково знайти збіги з українськими транслітераціями і замінити їх на англійські терміни, що лише ускладнює подальше вирівнювання з еталонним текстом і збільшує кількість помилок вставки (insertions).

Незважаючи на ці специфічні випадки, для цільових моделей-лідерів найсуттєвіший позитивний ефект спостерігався саме в розпізнаванні

англійських термінів. Для моделі ElevenLabs Scribe v1 показник помилок на англійських словах (EN-WER) знизився з 0,7490 до 0,4528. Абсолютне зменшення склало 0,2962, що у відносному вираженні означає покращення на 39,5 %. Для GPT-4o Transcribe відносне зниження помилок на англійських словах становило 18,4 %. Навіть для найкращої моделі Gemini 3.1 Pro застосування ITN дало помітний ефект: показник помилок на англійських словах знизився з 0,4256 до 0,2442 (на 0,1814 в абсолютному вираженні). Загальний WER для цієї моделі після нормалізації зменшився до 0,1053, а CER – до 0,0443.

Показовим прикладом такого покращення є результати моделі ElevenLabs Scribe v1 на аудіозаписі 10_azure. Еталонна нормалізована транскрипція виглядала так: *"завів managed identity в azure, тому сервіс тепер ходить без password у конфігах."* До застосування нормалізації модель згенерувала повністю транслітерований кириличний текст: *"вив менедж та ідентиті важур, тому сервіс не проходить без пасворд у конфігах."* Без нормалізації стандартна метрика класифікувала всі чотири англійські слова як помилкові. Після проходження через ITN-фільтр транслітеровані форми *"ідентиті"* та *"пасворд"* були автоматично розпізнані та відновлені до канонічних identity та password. Це знизило показник помилок в англійському сегменті (CS-WER_EN) для цього висловлювання з 1,0 до 0,5.

Водночас цей самий приклад яскраво ілюструє межі словникового підходу: сильно спотворені межі слів, коли акустична модель розриває англійський термін на два українські (як managed → "менедж та") або, навпаки, склеює прийменник із терміном (в azure → "важур"), алгоритм нормалізації відновити не зміг.

3.3.3 Обмеження підходу та перспективи розвитку

Попри ефективність, словниковий підхід до зворотної текстової нормалізації має обмеження. Він добре працює з наперед відомими формами,

але не завжди здатний обробляти продуктивні змішані утворення. Найскладнішими є українські дієслівні форми, утворені від англійських коренів, наприклад: "задеплоїв", "пофіксив", "закомітив". Такі слова поєднують англійську основу з українськими префіксами, суфіксами та граматичними закінченнями.

Оскільки програмна реалізація алгоритму спирається на регулярні вирази з жорсткими межами слова, система фізично не здатна вихопити англійський корінь зсередини українського слова. Цей захисний механізм є необхідним для уникнення деструктивних замін (наприклад, щоб не замінити випадковий склад усередині іншого слова), однак він робить алгоритм нечутливим до префіксальних та суфіксальних утворень.

Для коректної обробки таких конструкцій простого словника недостатньо. Ефективним архітектурним кроком для вирішення проблеми морфологічної сегментації є компіляція створеного словника у повноцінний зважений скінченний перетворювач (Weighted Finite-State Transducer, WFST) за допомогою спеціалізованих бібліотек рівня OpenFst.

Застосування архітектури WFST дасть змогу:

- Усунути комбінаторний вибух: замість створення тисяч комбінацій для слова deploy достатньо створити три окремі графи (граф українських префіксів, граф англійських коренів та граф українських закінчень) і виконати над ними математичну операцію композиції.
- Врахувати ймовірності (ваги): додавання ваг (costs) до різних шляхів графа дозволить системі обирати найімовірніший варіант перетворення на основі контексту.
- Зберегти швидкодію: обхід скомпільованого графа WFST має лінійну складність, що дозволить виконувати складну нормалізацію в режимі реального часу з мінімальною затримкою.

Іншим важливим напрямом майбутніх досліджень є вивчення внутрішніх механізмів того, як акустичні моделі передбачають та фонетично адаптують іншомовні слова. Особливий науковий інтерес у цьому контексті становить поведінка строго монолінгвальних систем, зокрема моделі Ukrainian Pods Conformer. Оскільки ця архітектура навчалася виключно на українськомовному корпусі та не має вбудованих знань про латинську орфографію, під час зіткнення зі змішаним мовленням вона здійснює природну "наївну" акустичну транслітерацію. Використання виводів таких моделей як автоматичного генератора транслітерацій дасть змогу зібрати масиви нетривіальних фонетичних адаптацій для навчання систем морфологічної сегментації.

3.4 Аналіз ефективності моделей наскрізного перекладу (Speech-to-Text Translation)

Проведені експерименти виявили фундаментальну різницю в підходах до перекладу між традиційними трансляційними архітектурами та сучасними мультимодальними Audio-LLM.

Експеримент із вузько спеціалізованою моделлю SeamlessM4T v2 Large частково підтвердив початкові побоювання. Хоча модель успішно повертала англійські терміни в їхню канонічну латинську форму (наприклад, "деплой" коректно ставав deploy), загальний технічний і семантичний зміст висловлювань зазнавав катастрофічних спотворень. Головна причина провалу таких моделей – прямий, дослівний переклад українського професійного сленгу та метафор. Яскравими прикладами стали:

Неправильне трактування метафор стану: фраза "staging ліг" була дослівно перекладена як "staging lifted" або "staging lay down".

Спотворення сленгових понять: слово "...підняв Azure…" стабільно перекладалося як "... pick up Azure…" замість професійних "spun up" або "deployed".

Натомість експерименти з мультимодальними Audio-LLM (зокрема Gemini 3.1 Pro у режимі перекладу) продемонстрували кардинально інші результати та довели високу ефективність цього підходу. Завдяки глибокому розумінню контексту, модель не виконувала дослівного перекладу, а здійснювала семантичну адаптацію українського IT-жаргону в природну ділову англійську мову.

Аналіз результатів перекладу Gemini 3.1 Pro виявив таке:

- Точне розуміння IT-ідіом: фраза "стейджинг ліг" була коректно інтерпретована як "staging went down", а фраза "залишити хвости" – як "technical debt".
- Відновлення англійських коренів з українських неологізмів: слово "захардкодив" модель успішно розпізнала та переклала як "hardcoded", без втрати морфологічної основи.

Водночас дослідження виявило методологічну проблему: оцінювання систем семантичного перекладу (Audio-LLM) за допомогою стандартних метрик WER та CER є нерелевантним. Алгоритми Audio-LLM схильні використовувати синоніми (наприклад, "short update" замість еталонного "quick update"), стягнення ("it's" замість "it is") та змінювати синтаксичну структуру речення без втрати технічного змісту. Оскільки метрика WER штрафує за будь-яке відхилення від еталонного рядка, вона штучно завищує показник помилок (показуючи WER понад 25 %), хоча фактичний переклад є абсолютно коректним та придатним для використання.

Таким чином, гіпотеза про використання наскрізного перекладу для подолання проблем транслітерації є підтвердженою для класу Audio-LLM. Цей підхід є потужною альтернативою сирому транскрибуванню з ITN, проте його використання вимагає переходу від метрик точного лексичного збігу (WER) до семантичних метрик оцінювання якості перекладу (BLEU, METEOR або

LLM-as-a-judge). Зведені результати кількісного оцінювання моделей перекладу наведено в Додатку Б. Розширені приклади порівняння дослівного калькування із семантичною адаптацією наведено в Додатку В.

3.5 Нейромережевий постпроцесинг

Альтернативним напрямом розв'язання проблеми змішаного мовлення є застосування нейромережевого постпроцесингу. Розуміючи обмеження жорстких словників, провідні технологічні компанії активно досліджують перехід до нейромережевої зворотної нормалізації.

Дослідники з Apple довели, що класичні генеративні архітектури (Seq2Seq) є неприпустимими для продакшн-систем через ризик "незворотних помилок" та галюцинацій. Для уникнення цього вони запропонували розглядати ITN як задачу маркування послідовностей (sequence labeling), де модель лише присвоює словам теги дій. У 2021 році дослідники з Amazon AWS AI підтвердили, що нейромережеві підходи на базі маркування можуть працювати швидше та масштабуватися краще за складні багаторівневі системи регулярних виразів. [41, 42]

Розвиваючи ідею гібридизації для роботи в режимі реального часу, дослідники Microsoft запропонували розділити процес ITN на тегування (Tagging) та перетворення (Transduction). Нейромережа лише позначає діапазони тексту, а перетворення виконується детермінованим автоматом WFST. Новітнім проривом є дослідження команди L. Ho (Interspeech 2025), у якому архітектура динамічно адаптується до розміру аудіофрагмента, що дозволяє виконувати миттєву (instant) нормалізацію великими мовними моделями безпосередньо під час мовлення. [43, 44]

Огляд цих робіт підтверджує, що для подальшого вдосконалення системи та оброблення складних українських дієслів доцільно перейти до гібридного підходу: нейромережевий тегер має визначати межі англійського кореня, а

детермінована мапа транслітерацій або WFST – виконувати безпечну заміну.

Висновок до розділу 3

Проведені експерименти та детальний аналіз результатів підтверджують ключову тезу дослідження: сучасні ASR-системи (особливо мультимодальні Audio-LLM) розпізнають англійські терміни на акустичному рівні значно краще, ніж це відображає стандартна метрика WER. Головна проблема транскрибування змішаного мовлення полягає не стільки в акустичному нерозпізнаванні сигналу, скільки в орфографічній транслітерації, помилках токенізації та хибному виборі системи письма (графіки).

Застосування детермінованого алгоритму зворотної текстової нормалізації довело свою ефективність, суттєво знизивши відсоток помилок на іншомовних включеннях (у середньому на 13–30 %). Водночас виявлено межі застосовності "плоских" словників на базі регулярних виразів, які не здатні обробляти складну українську морфологію (дієслівні префікси та суфікси). Для розв'язання цієї проблеми математично та архітектурно обґрунтовано доцільність переходу до зважених скінченних перетворювачів та гібридних підходів нейромережевої нормалізації.

Перевірка гіпотези щодо використання наскрізного перекладу виявила фундаментальні відмінності між архітектурами. Традиційні трансляційні моделі не придатні для обробки IT-домену через дослівне калькування жаргонізмів. Натомість великі мовні моделі (Audio-LLM) здійснюють точну семантичну адаптацію українського професійного сленгу в англійську мову, що робить переклад життєздатною та потужною альтернативою прямому транскрибуванню, за умови використання семантичних метрик оцінювання замість WER.

Отже, для якісної обробки явища code-switching недостатньо лише розгортання потужної акустичної моделі. Критично необхідною є розробка

комплексного конвеєра, що поєднує акустичне розпізнавання зі спеціалізованими механізмами зворотної текстової нормалізації та постобробки, адаптованими до реальних умов використання іншомовної термінології.

РОЗДІЛ 4: РОЗРОБКА СИСТЕМИ ТРАНСКРИБУВАННЯ ПОВНОГО ЦИКЛУ

4.1 Архітектура системи

Розроблена система є настільним застосунком під назвою "Voice Diary", що розв'язує завдання транскрибування та ідентифікації мовців у змішаному українсько-англійському мовленні в режимі, наближеному до реального часу, в умовах локального виконання без залежності від хмарних сервісів. Технологічний стек містить: середовище Electron 41 як оболонку, React 19 + TypeScript + Vite на стороні клієнтської частини, а також FastAPI + SQLite на стороні серверної частини. Обмін даними здійснюється через REST API та протокол WebSocket.

Фундаментальним архітектурним рішенням є архітектура на основі провайдерів, де кожен із трьох ключових модулів машинного навчання реалізований за окремим інтерфейсом Protocol мови Python (лістинг 4.1).

Лістинг 4.1 – Інтерфейси модулів машинного навчання

```
class ASRProvider(Protocol):

  def transcribe(self, audio: np.ndarray, language_hint: str) -> Utterance:
...

class DiarizationProvider(Protocol):

    def segment(self, audio: np.ndarray) -> list[Segment]: ...

class EmbeddingProvider(Protocol):

    def embed(self, audio: np.ndarray) -> np.ndarray: ...
```

Такий підхід запозичений у відкритому репозиторії проєкту OpenOats [45], де реалізовано аналогічний шаблон проєктування TranscriptionBackend із методами prepare(), transcribe() і reset(). Провайдер-орієнтована архітектура дає

змогу замінювати окремі модулі (модель розпізнавання, модуль діаризації, модель побудови векторів) без переписування координатора або інтерфейсу. Для порівняння: застосунок Meetily [46] використовує аналогічний підхід на базі типажів (trait) TranscriptionProvider у мові Rust, а система Vexa [47] виконує конфігурацію компонентів через змінні середовища (environment variables).

Поточний набір провайдерів містить:

- для автоматичного розпізнавання мовлення (ASR): faster-whisper large-v3-turbo, конвеєр HuggingFace Transformers, ElevenLabs Scribe (хмарний резервний варіант);
- для діаризації: pyannote/speaker-diarization-3.1, NeMo Sortformer v2.1 (альтернатива з вимогою CUDA);
- для векторних представлень: speechbrain/spkrec-ecapa-voxceleb (архітектура ECAPA-TDNN, розмірність 192).

Фактичний порядок оброблення аудіоданих у серверній частині:

1. отримання аудіофрагмента (chunk) у форматі PCM (100 мс, float32, 16 кГц);
2. детектування пауз і застосування гейту за допомогою Silero VAD;
3. формування завершеного висловлювання;
4. сегментація за мовцями за допомогою PyAnnote 3.1;
5. транскрибування за допомогою ASR-моделі faster-whisper;
6. обчислення голосового вектора моделлю ECAPA-TDNN;
7. ідентифікація мовця або перенесення його до черги невідомих;
8. збереження в базу даних SQLite та передавання через WebSocket до інтерфейсу користувача.

Вибір базової ASR-моделі для застосунку здійснювався на основі масштабного бенчмарку, результати якого детально описано в Розділі 3.

4.2 Модуль Voice Activity Detection та оптимізація пам'яті

Voice Activity Detection є воротами усього процесу. Без нього можливі два крайні варіанти: або чекати повного завершення сесії й передавати весь аудіозапис на ASR та діаризацію пакетно (втрата near-real-time), або запускати всі моделі на кожному 100-мілісекундному чанку (катастрофічна обчислювальна вартість та погіршення результату). VAD вирішує цю дилему: він виявляє паузи між репліками і передає на важкі моделі лише закриті мовленнєві сегменти.

4.2.1 Вибір VAD

Для реалізації обрано Silero VAD – це компактна LSTM-модель, що повертає per-frame speech probability (0..1) на вхідних вікнах 512 семплів (32 мс) при 16 кГц. [48] Порівняння з альтернативами наведено в таблиці 4.2.1.

Таблиця 4.2.1 – Порівняння методів VAD

Підхід	Швидкість	Якість на шумі	Ресурс	Рекомендація
Silero VAD	Швидкий	Добра	~1 MB	Обрано
WebRTC VAD	Дуже швидкий	Гірша	Ультралегкий	Ultra-edge сценарії
pyannote VAD	Повільний	Найкраща	~2 GB	Research/eval

4.2.2 Еволюція: від single-threshold до dual-threshold hysteresis

Початкова реалізація застосунку використовувала стандартний ітератор з єдиним порогом (threshold=0,5). Це породжувало дві взаємопов'язані проблеми.

Першою проблемою є швидке перемикання станів. Один поріг використовується і для входу в стан мовлення, і для виходу з нього. Під час паузи або вагання мовця ймовірність коливається в межах 0,48–0,52. Кожен перетин порогу провокує зміну стану: система генерує дрібні "сміттєві"

висловлювання або безпідставно розрізає репліку.

Другою проблемою є стрибки використання пам'яті. У разі тривалого монологу без пауз репліка потрапляла в PyAnnote як єдиний 30-секундний блок. Матриця уваги архітектури Transformer зростає квадратично, що призводило до переповнення відеопам'яті.

Аналіз відкритих рішень (таблиця 4.2.2) показав, що конкурентні системи застосовують алгоритм гістерезису.

Таблиця 4.2.2 – Аналіз альтернативних рішень керування VAD

Проєкт	Onset (start)	Offset (end)	min_silence
Meetily	0,50	0,35	400 ms
Vexa	0,60	0,45	250 ms
OpenOats	FluidAudio .default	same	.default
Voice Diary (до)	0,50	0,50 (same)	500 ms

Поняття гістерезису в теорії керування та обробці сигналів означає залежність стану системи від її передісторії. У контексті VAD це означає використання двох різних порогів замість одного. Високий поріг (onset) використовується для переходу зі стану "тиша" в стан "мовлення" -- це надійно захищає систему від хибних спрацьовувань на випадкові шуми (кліки, кашель). Низький поріг (offset) використовується для зворотного переходу зі стану "мовлення" в стан "тиша" – він утримує мікрофон "відкритим", запобігаючи передчасному обрізанню аудіо під час природних мікропауз, вагань мовця або перемикання між мовами (де артикуляція часто сповільнюється).

Реалізоване рішення: виклик ітератора замінено на звернення до необробленої (raw) моделі Silero з реалізацією власного скінченного автомата (state machine):

- Якщо ймовірність ≥ 0,60 → фронт наростання, вхід у мовлення.
- Якщо ймовірність у межах [0,45; 0,60] → смуга гістерезису, система залишається в стані мовлення.
- Якщо ймовірність < 0,45 → спад, починається відлік часу тиші. Якщо тиша триває понад 300 мс – вихід із мовлення.

Для розв'язання проблеми пам'яті максимальну довжину фрагмента було зменшено до 13 секунд. Оскільки споживання пам'яті зростає квадратично, пікове навантаження знизилося до очікуваних значень.

4.2.3 Специфіка code-switching для VAD

Завдяки гістерезису [0,45; 0,60] та мінімальній тиші у 300 мс мікропаузи між словами не закривають сесію. Крім того, відповідно до аналізу рішень конкурентів, реалізовано режим деградації: у разі критичної помилки Silero система не обриває з'єднання, а примусово скидає буфер (force-flush), переходячи в режим "постійне мовлення".

4.3 Модуль діаризації та ідентифікація мовців

4.3.1 Speaker Diarization

Для ідентифікації різних дикторів застосовується сегментація аудіо за мовцями (speaker diarization). Основним серверним рушієм обрано pyannote/speaker-diarization-3.1. Система працює в необмеженому (unconstrained) режимі: кількість мовців визначається автоматично. Як альтернатива з підтримкою CUDA впроваджено NeMo Sortformer v2.1. Унікальною перевагою розробленої системи є міжсесійна (cross-session) ідентифікація мовців: на відміну від Vexa чи Meetily, застосунок Voice Diary зберігає голосові профілі у SQLite, що дає змогу автоматично впізнавати знайомих дикторів у нових записах. [49]

4.3.2 Speaker Identification та проблема code-switching

Для побудови векторів застосовується модель ECAPA-TDNN, що повертає 192-вимірний вектор. Функцією близькості є показник косинусної схожості.

Центральна проблема: під час переходу україномовного диктора на англійську змінюються його темп мовлення, просодика. Це призводить до зміщення вектора у просторі ознак. Початковий поріг розпізнавання 0,82 виявився занадто жорстким.

Сегменти з переважаючою англійською мовою давали scores у діапазоні 0,508-0,741, що нижче початкового порогу. Система помилково класифікувала їх як "невідомих мовців" і переміщала у чергу на ручне розв'язання, хоча фактично йшлося про того самого мовця, що щойно перейшов на англійську.

Для вирішення цієї проблеми поріг знижено до 0,5. Емпіричний аналіз показує, що реалістичний діапазон для noisy single-mic аудіо 0,55-0,65. Поріг налаштовується користувачем через Settings у реальному часі без перезапуску (POST /config/threshold).

Unknown Queue та cascade re-identification. Сегменти, що не досягли порогу, потрапляють у чергу unknown_queue. Через інтерфейс Unknown Queue користувач бачить кандидатів (отриманих через GET /utterances/{id}/candidates з cosine-ranked результатами) і вручну резолвить невідомого мовця на існуючий або новий контакт. Після підтвердження система автоматично виконує cascade re-identification, тобто переперевіряє всі попередні нерозв'язані сегменти тієї ж сесії з новим голосовим профілем.

4.4 ASR-модуль та захист від галюцинацій

Для транскрипції використовується faster-whisper - це оптимізована реалізація архітектури Whisper із квантизацією у форматі int8, що забезпечує 4-кратне прискорення висновування порівняно з оригіналом.

Однією з практичних проблем моделі Whisper є генерація заучених фраз ("Дякую за увагу", "[Музика]", "Субтитры сделал...") на тлі тиші або шуму. Запозичивши ідею з проєкту Vexa [47], було реалізовано чорний список (blocklist), сформований на базі відкритого датасету sachaarbonel/whisper-hallucinations та доповнений артефактами. Проєкт Vexa використовує блокліст з ~150 слів, які трапляються найчастіше при транскрибуванні англійською мовою.[50] Датасет sachaarbonel/whisper-hallucinations був окремо доповнений даними, які були виявлені під час тестування. Додатково впроваджено критерії якості: відхилення тексту з аномальним співвідношенням стиснення та штрафування за повторення N-грам.

Для боротьби з хибним визначенням мови (наприклад, під час переходу на слово "deployment" модель Whisper може розпізнати його як нідерландський текст через збіг фонем, що руйнує зміст), було реалізовано "білий список" мов (en, uk). Якщо впевненість моделі падає нижче 0,5, координатор запускає примусове транскрибування дозволеними мовами та обирає варіант із найвищою логарифмічною ймовірністю. Модель також генерує часові мітки на рівні слів, що критично важливо для вирівнювання тексту із сегментами діаризації.

4.5 Інтерфейс користувача

Клієнтська частина реалізована на базі React 19 та TypeScript. Взаємодія з серверною частиною відбувається через REST API (керування налаштуваннями) та два незалежних WebSocket-з'єднання (бінарний потік аудіо від мікрофона та системного аудіо). Усі моделі машинного навчання завантажуються як патерн Singleton, проте їхнє висновування виконується в ізольованому пулі потоків для забезпечення потокобезпечності.

Основні екрани застосунку:

- "Current Session": транскрибування в реальному часі.

- "All Sessions": історія збережених записів.
- "Unknown Queue": інтерфейс ручного прив'язування нерозпізнаних голосових векторів до контактів.
- "Contacts": галерея акустичних профілів.
- "Settings": гнучке керування порогами VAD та вибір рушіїв.

4.6 Схема бази даних

Дані зберігаються у СУБД SQLite. Структура бази даних містить сім реляційних таблиць та одну віртуальну таблицю. Головна таблиця sessions безпосередньо пов'язана з таблицею utterances, яка зберігає текстові транскрипції. Для забезпечення швидкого повнотекстового пошуку дані з таблиці висловлювань синхронізуються з віртуальною таблицею utterances_fts (на базі розширення FTS5) за допомогою тригерів бази даних AFTER INSERT/UPDATE/DELETE. Дані про мовців організовано через таблиці contacts, voice_profiles та чергу unknown_queue. Голосові вектори зберігаються в полях embedding як масиви з рухомою комою (формат BLOB, тип float32, L2-нормалізовані).

ВИСНОВКИ

У магістерській дипломній роботі досліджено актуальну проблему – здатність сучасних систем автоматичного розпізнавання мовлення транскрибувати українсько-англійське мовлення з перемиканням кодів та створено програмну систему повного циклу для обробки таких аудіоданих.

Створено спеціалізований еталонний корпус даних. Для проведення об’єктивного тестування зібрано датасет uk-en-code-mixed-asr-2h, що складається з 448 аудіозаписів загальною тривалістю 2,11 години на IT-тематику. Аналіз корпусу підтвердив його репрезентативність: 99,6 % зразків містять внутрішньореченнєве перемикання кодів, а частка англійських слів становить 18,9 %.

Проведено масштабний бенчмарк 35 сучасних ASR-моделей. Встановлено, що лідерами у розпізнаванні змішаного мовлення є мультимодальні Audio-LLM, зокрема Gemini 3.1 Pro із загальним WER 0,105, та спеціалізовані комерційні API, зокрема ElevenLabs Scribe v2. Найкращою локальною моделлю визнано faster-whisper large-v3-turbo з WER 0,167 та RTF 0,028. Доведено, що монолінгвальні українські моделі, зокрема W2V-BERT і Conformer, принципово не здатні обробляти технічне перемикання кодів без додаткового fine-tuning.

Виявлено фундаментальну асиметрію CS-WER. Аналіз помилок за мовними сегментами показав, що показник помилок на англійських термінах перевищує показник помилок на українських словах у 3-10 разів для всіх протестованих архітектур. Це доводить, що обробка іншомовних включень є головним вузьким місцем сучасних систем розпізнавання мовлення.

Доведено критичну важливість зворотної текстової нормалізації. Створення та застосування кастомної мапи транслітерацій, що містить 1240

англійських термінів та 4838 унікальних пар відповідностей, покращило показник помилок на англійських термінах для 100 % моделей. Середнє зниження помилок на англіцизмах становило 13-30 %. Це емпірично доводить, що ASR-моделі мають високу акустичну точність, а значна частина їхніх помилок спричинена орфографічною транслітерацією латинських термінів кирилицею.

Досліджено гіпотезу про ефективність наскрізного перекладу мовлення в текст (Speech-to-Text Translation). Виявлено дихотомію в ефективності різних архітектур. Традиційні моделі перекладу демонструють неспроможність коректно обробляти IT-жаргонізми через дослівне калькування. Натомість сучасні мультимодальні Audio-LLM, зокрема Gemini 3.1 Pro, розв'язують цю проблему: вони успішно відновлюють латинське написання термінів та семантично адаптують український сленг у правильну технічну англійську мову. Встановлено, що для об'єктивного оцінювання Audio-LLM у задачах перекладу метрика WER є нерелевантною через жорстке штрафування за синонімію та перефразування.

Розроблено десктопний застосунок Voice Diary. Створено програмний комплекс із провайдер-орієнтованою архітектурою. Успішно вирішено проблему стрибків пам’яті під час обробки довгих аудіо за допомогою налаштування VAD-гістерезису за принципом подвійного порогу. Реалізовано крос-сесійну ідентифікацію мовців на базі ECAPA-TDNN з оптимізованим порогом косинусної схожості 0,5, адаптованим до зміни акустичних ознак під час перемикання мов.

Впроваджено алгоритми захисту від типових галюцинацій моделей сімейства Whisper. Це підвищило стабільність роботи системи під час обробки довгих або акустично складних записів і зменшило ризик появи нерелевантного згенерованого тексту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Poplack S. Sometimes I'll start a sentence in Spanish Y TERMINO EN ESPANOL: toward a typology of code-switching // Linguistics. – 1980. – Vol. 18, No. 7-8. – P. 581–618.

2. Дровосєкова Н. Між двох мов: як емоційний інтелект диктує перемикання кодів у соціальних мережах // Mundus Philologiae. – 2024. – № 3 [Електронний ресурс]. – Режим доступу: https://mundphil.kubg.edu.ua/index.php/journal/article/view/31

3. Lyu D. et al. ASCEND: A Spontaneous Chinese-English Dataset for Code-switching in Multi-turn Conversation // Proceedings of the 13th Language Resources and Evaluation Conference. – 2022.

4. Lyu D. C. et al. An analysis of a mandarin-english code-switching speech corpus: SEAME // Interspeech 2010. – 2010.

5. Lyu D. C. et al. SEAME Mandarin-English Code-Switching Speech Corpus. LDC2015S04 // Linguistic Data Consortium. – 2015 [Електронний ресурс]. – Режим доступу: https://catalog.ldc.upenn.edu/LDC2015S04

6. Deuchar M. et al. BL Corpus: A Bangor Corpus of Bilingual Talk [Електронний ресурс]. – Режим доступу: https://talkbank.org/biling/access/Bangor/Deuchar2014.pdf

7. Yılmaz E. et al. A longitudinal bilingual Frisian-Dutch radio broadcast database // Proceedings of the 10th LREC 2016. – 2016 [Електронний ресурс]. – Режим доступу: http://www.lrec-conf.org/proceedings/lrec2016/pdf/704_Paper.pdf

8. Conneau A. et al. FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech // arXiv preprint arXiv:2205.12446. – 2022.

9. Zhao Y. et al. CS-FLEURS: A Benchmark for Code-Switched Speech Recognition // arXiv preprint arXiv:2509.14161. – 2025.

10. Nikitin R. uk-en-code-mixed-asr-2h dataset [Електронний ресурс]. – Режим доступу: https://huggingface.co/datasets/vnikitin/uk-en-code-mixed-asr-2h

11. Pratap V. et al. Scaling Speech Technology to 1,000+ Languages // arXiv preprint arXiv:2305.13516. – 2023.

12. Baevski A., Zhou Y., Mohamed A., Auli M. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations // Advances in Neural Information Processing Systems. – 2020. – P. 12449–12460.

13. Babu A. et al. XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale // arXiv preprint arXiv:2111.09296. – 2021.

14. Smoliakov Y. Ukrainian ASR Models [Електронний ресурс]. – Режим доступу: https://huggingface.co/Yehor

15. Radford A. et al. Robust Speech Recognition via Large-Scale Weak Supervision // arXiv preprint arXiv:2212.04356. – 2022.

16. Gulati A. et al. Conformer: Convolution-augmented Transformer for Speech Recognition // Interspeech 2020. – 2020. – P. 5036–5040.

17. Kuchaiev O. et al. NeMo: a toolkit for building AI applications using Neural Modules // arXiv preprint arXiv:1909.09577. – 2019.

18. Kim J. et al. Efficient Sequence Transduction by Jointly Predicting Tokens and Durations // arXiv preprint arXiv:2304.06795. – 2023.

19. Ukrainian Pods. Ukrainian Pods Conformer [Електронний ресурс]. – Режим доступу: https://huggingface.co/taras-sereda/uk-pods-conformer

20. Alpha Cephei. Vosk Speech Recognition Toolkit [Електронний ресурс]. – Режим доступу: https://alphacephei.com/vosk/

21. Mistral AI. Voxtral: Open Speech Understanding Models // arXiv preprint arXiv:2507.13264. – 2025.

22. Google DeepMind. Gemini 3.1 Pro Model Card [Електронний ресурс]. – Режим доступу: https://deepmind.google/models/model-cards/gemini-3-1-pro/

23. Google AI. Gemini API Models [Електронний ресурс]. – Режим доступу: https://ai.google.dev/gemini-api/docs/models

24. Google DeepMind. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context // arXiv preprint arXiv:2507.06261. – 2025.

25. Xiaomi. MiMo-v2-Omni: Multimodal Models for Omni-Interaction [Електронний ресурс]. – Режим доступу: https://mimo.xiaomi.com/mimo-v2-omni

26. OpenAI. Hello GPT-4o [Електронний ресурс]. – Режим доступу: https://openai.com/index/hello-gpt-4o/

27. Seamless Communication et al. SeamlessM4T: Massively Multilingual & Multimodal Machine Translation // arXiv preprint arXiv:2308.11596. – 2023.

28. ElevenLabs. Speech to Text [Електронний ресурс]. – Режим доступу: https://elevenlabs.io/docs/overview/capabilities/speech-to-text

29. Deepgram. Introducing Nova-3 Speech-to-Text API [Електронний ресурс]. – Режим доступу: https://deepgram.com/learn/introducing-nova-3-speech-to-text-api

30. AssemblyAI. Universal-3 Pro [Електронний ресурс]. – Режим доступу: https://assemblyai.com/docs/getting-started/universal-3-pro

31. Soniox. Speech-To-Text Models [Електронний ресурс]. – Режим доступу: https://soniox.com/docs/stt/models

32. Speechmatics. Documentation [Електронний ресурс]. – Режим доступу: https://docs.speechmatics.com/

33. Microsoft. Azure AI Speech – Speech-to-text [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/azure/ai-services/speech-service/speech-to-text

34. Zhang Y. et al. Google USM: Scaling Automatic Speech Recognition Beyond 100 Languages // arXiv preprint arXiv:2303.01037. – 2023.

35. Google Cloud. Cloud Speech-to-Text: Chirp Model [Електронний ресурс]. – Режим доступу: https://docs.cloud.google.com/speech-to-text/docs/models/chirp-3

36. Interfaze. Автоматичне розпізнавання мовлення [Електронний ресурс]. – Режим доступу: https://interfaze.ai/

37. Jurafsky D., Martin J. H. Speech and Language Processing. 3rd ed. – 2024 [Електронний ресурс]. – Режим доступу: https://web.stanford.edu/~jurafsky/slp3/

38. Zhao Y. et al. PolyWER: Evaluation of Code-Switching ASR using Multiple Reference Variants // Findings of the Association for Computational Linguistics: EMNLP 2024. – 2024 [Електронний ресурс]. – Режим доступу: https://aclanthology.org/2024.findings-emnlp.356

39. Desplanques B., Thienpondt J., Demuynck K. ECAPA-TDNN: Emphasized Channel Attention, propagation and aggregation in TDNN based speaker verification // Interspeech 2020. – 2020. – P. 3830–3834.

40. Про впорядкування транслітерації українського алфавіту латиницею : постанова Кабінету Міністрів України від 27.01.2010 р. № 55 [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/55-2010-п

41. Pusateri E., Chen B., Stahlberg C. Inverse Text Normalization as a Labeling Problem // Apple Machine Learning Research. – 2017 [Електронний ресурс]. – Режим доступу: https://machinelearning.apple.com/research/inverse-text-normal

42. Sunkara M. et al. Neural inverse text normalization // ICASSP 2021. – IEEE, 2021. – P. 7578–7582 [Електронний ресурс]. – Режим доступу: https://www.amazon.science/publications/neural-inverse-text-normalization

43. Gaur Y. et al. Streaming, fast and accurate on-device Inverse Text Normalization for Automatic Speech Recognition // arXiv preprint arXiv:2211.03721. – 2022.

44. Ho L. et al. Dynamic Context-Aware Streaming Pretrained Language Model For Inverse Text Normalization // arXiv preprint arXiv:2505.24229. – 2025.

45. OpenOats. OpenOats Transcription System [Електронний ресурс]. – Режим доступу: https://github.com/yazinsai/OpenOats

46. Meetily. Meetily: Open-source transcription [Електронний ресурс]. – Режим доступу: https://github.com/Zackriya-Solutions/meetily

47. Vexa. Vexa Speech Engine [Електронний ресурс]. – Режим доступу: https://github.com/Vexa-ai/vexa

48. Silero Team. Silero VAD: pre-trained enterprise-grade Voice Activity Detector [Електронний ресурс]. – Режим доступу: https://github.com/snakers4/silero-vad

49. Bredin H. et al. pyannote.audio: neural building blocks for speaker diarization // ICASSP 2020. – IEEE, 2020. – P. 7124–7128.

50. Arbonel S. whisper-hallucinations dataset [Електронний ресурс]. – Режим доступу: https://huggingface.co/datasets/sachaarbonel/whisper-hallucinations

ДОДАТОК А

(обов’язковий)

Детальні результати експериментів

Таблиця А.1 -- Результати бенчмарку моделей

#	Model	EN-WER	EN ITN	Δ EN	UK-WER	UK ITN	Δ UK	WER	WER ITN	Δ WER	CER	CER ITN	Δ CER	Fails
1	Gemini 3.1 Pro	0,4256	0,2442	-0,1815	0,0742	0,0737	-0,0005	0,1359	0,1053	-0,0306	0,0784	0,0443	-0,0342	0
2	ElevenLabs Scribe v2	0,3530	0,2700	-0,0830	0,0962	0,0959	-0,0003	0,1414	0,1337	-0,0077	0,0583	0,0457	-0,0125	0
3	ElevenLabs Scribe v2 (force_uk)	0,3552	0,2769	-0,0783	0,0967	0,0965	-0,0002	0,1422	0,1365	-0,0057	0,0572	0,0457	-0,0115	0
4	Gemini 3 Flash	0,3888	0,2644	-0,1244	0,0921	0,0917	-0,0004	0,1442	0,1277	-0,0166	0,0743	0,0507	-0,0236	0
5	Gemini 2.5 Pro	0,4278	0,2803	-0,1475	0,0856	0,0854	-0,0002	0,1458	0,1220	-0,0237	0,0756	0,0466	-0,0289	0
6	Gemini 3.1 Flash Lite	0,4075	0,3058	-0,1017	0,1029	0,1026	-0,0003	0,1564	0,1475	-0,0089	0,0705	0,0560	-0,0145	0

#	Model	EN-WER	EN ITN	Δ EN	UK-WER	UK ITN	Δ UK	WER	WER ITN	Δ WER	CER	CER ITN	Δ CER	Fails
7	Gemini 3.5 Flash	0,4790	0,3370	-0,1420	0,1012	0,1007	-0,0005	0,1676	0,1422	-0,0254	0,0857	0,0571	-0,0286	0
8	GPT-4o Transcribe	0,4621	0,2778	-0,1843	0,1114	0,1110	-0,0004	0,1730	0,1524	-0,0206	0,0935	0,0644	-0,0291	0
9	Soniox Async v4 (uk+en hints + langid)	0,4621	0,3862	-0,0759	0,1115	0,1114	-0,0001	0,1731	0,1621	-0,0110	0,0697	0,0549	-0,0148	0
10	Soniox Async v4 (uk+en hints)	0,4634	0,3837	-0,0797	0,1115	0,1114	-0,0001	0,1733	0,1626	-0,0107	0,0710	0,0558	-0,0153	0
11	Soniox Async v4	0,4718	0,3902	-0,0816	0,1125	0,1124	-0,0001	0,1756	0,1653	-0,0103	0,0729	0,0576	-0,0153	0
12	Whisper Large v3 (force_uk)	0,4487	0,3172	-0,1315	0,1042	0,1039	-0,0003	0,1781	0,1529	-0,0252	0,0894	0,0635	-0,0258	0
13	Chirp 3	0,3634	0,2842	-0,0792	0,1135	0,1132	-0,0003	0,1846	0,1547	-0,0299	0,0977	0,0616	-0,0361	15
14	Gemini 2.5 Flash	0,5360	0,4130	-0,1231	0,1120	0,1115	-0,0005	0,1865	0,1702	-0,0163	0,0845	0,0639	-0,0205	0

#	Model	EN-WER	EN ITN	Δ EN	UK-WER	UK ITN	Δ UK	WER	WER ITN	Δ WER	CER	CER ITN	Δ CER	Fails
15	Whisper Large v3 (auto)	0,4487	0,3185	-0,1303	0,1073	0,1071	-0,0003	0,1877	0,1633	-0,0244	0,0919	0,0669	-0,0250	0
16	Whisper Large v3 Turbo (force_uk)	0,5066	0,3595	-0,1472	0,1138	0,1136	-0,0002	0,1953	0,1672	-0,0281	0,0979	0,0693	-0,0286	0
17	Whisper Large v3 Turbo (auto)	0,5069	0,3601	-0,1469	0,1153	0,1151	-0,0002	0,2002	0,1723	-0,0279	0,0990	0,0707	-0,0283	0
18	Whisper Large v2 (force_uk)	0,4891	0,3770	-0,1122	0,1242	0,1240	-0,0001	0,2003	0,1786	-0,0216	0,0925	0,0704	-0,0222	0
19	Interfaze STT (beta)	0,5569	0,4478	-0,1091	0,1445	0,1442	-0,0004	0,2170	0,2118	-0,0052	0,0961	0,0775	-0,0186	0
20	Gemini 2.5 Flash Lite	0,6031	0,4183	-0,1848	0,1427	0,1422	-0,0006	0,2236	0,2019	-0,0217	0,1149	0,0856	-0,0293	0
21	Voxtral Small (24B)	0,4680	0,3694	-0,0987	0,1811	0,1809	-0,0002	0,2315	0,2331	+0,0016	0,1244	0,1195	-0,0049	0
22	AssemblyAI Universal-2	0,5706	0,4849	-0,0857	0,1635	0,1632	-0,0003	0,2351	0,2392	+0,0041	0,0943	0,0850	-0,0093	0

#	Model	EN-WER	EN ITN	Δ EN	UK-WER	UK ITN	Δ UK	WER	WER ITN	Δ WER	CER	CER ITN	Δ CER	Fails
23	ElevenLabs Scribe v1 (force_uk)	0,1326	0,1326	0,1326	0,1326	0,1326	0,1326	0,1326	0,1326	0,1326	0,1326	0,1326	0,1326	0
24	ElevenLabs Scribe v1	0,7490	0,4528	-0,2962	0,1349	0,1321	-0,0028	0,2427	0,1943	-0,0484	0,1508	0,0941	-0,0567	0
25	Whisper Large v2 (auto)	0,4949	0,3827	-0,1122	0,1569	0,1569	+0,0000	0,2452	0,2244	-0,0207	0,1088	0,0875	-0,0213	0
26	Whisper Medium (force_uk)	0,5811	0,4204	-0,1607	0,1546	0,1547	+0,0001	0,2480	0,2173	-0,0307	0,1247	0,0937	-0,0309	0
27	GPT-4o Mini Transcribe	0,5469	0,3971	-0,1499	0,1982	0,1980	-0,0001	0,2594	0,2597	+0,0003	0,1322	0,1124	-0,0198	0
28	Whisper Medium (auto)	0,5676	0,4141	-0,1535	0,1850	0,1850	+0,0001	0,2773	0,2490	-0,0283	0,1342	0,1058	-0,0284	0
29	Parakeet TDT 0.6B v3	0,9018	0,6437	-0,2581	0,1533	0,1524	-0,0010	0,2849	0,2520	-0,0329	0,1713	0,1249	-0,0464	0
30	Nova-3 (uk + smart_format)	0,8709	0,7122	-0,1587	0,1731	0,1729	-0,0002	0,2957	0,2797	-0,0160	0,1548	0,1280	-0,0268	0

#	Model	EN-WER	EN ITN	Δ EN	UK-WER	UK ITN	Δ UK	WER	WER ITN	Δ WER	CER	CER ITN	Δ CER	Fails
31	Canary 1B v2 (uk ASR)	0,7122	0,6565	-0,0557	0,2297	0,2297	-0,0000	0,3146	0,3186	+0,0040	0,1467	0,1435	-0,0032	0
32	SeamlessM4T v2 Large (uk ASR)	0,8566	0,7415	-0,1150	0,2261	0,2261	-0,0000	0,3368	0,3190	-0,0179	0,2319	0,2038	-0,0281	0
33	Voxtral Mini	0,5978	0,4805	-0,1172	0,3092	0,3092	+0,0001	0,3599	0,3805	+0,0206	0,1561	0,1494	-0,0067	0
34	Whisper Small (force_uk)	0,7358	0,5781	-0,1577	0,2553	0,2557	+0,0003	0,3638	0,3340	-0,0298	0,1808	0,1513	-0,0295	0
35	Nova-3 (uk)	0,9143	0,7839	-0,1304	0,2535	0,2534	-0,0001	0,3696	0,3601	-0,0094	0,1697	0,1435	-0,0262	0
36	Whisper Small (auto)	0,7352	0,5811	-0,1541	0,2698	0,2702	+0,0004	0,3872	0,3591	-0,0280	0,1920	0,1641	-0,0280	0
37	Speechmatics (enhanced)	1,2385	1,0997	-0,1389	0,2232	0,2212	-0,0021	0,4004	0,4006	+0,0001	0,2692	0,2634	-0,0058	0
38	Speechmatics (standard)	1,2529	1,1034	-0,1495	0,2285	0,2268	-0,0017	0,4069	0,4086	+0,0016	0,2700	0,2643	-0,0058	0
39	Azure Speech (uk-UA)	0,9900	0,8785	-0,1115	0,3364	0,3353	-0,0011	0,4509	0,4353	-0,0156	0,3022	0,2802	-0,0220	0

#	Model	EN-WER	EN ITN	Δ EN	UK-WER	UK ITN	Δ UK	WER	WER ITN	Δ WER	CER	CER ITN	Δ CER	Fails
40	Ukrainian Pods Conformer	1,0097	0,8997	-0,1099	0,3629	0,3628	-0,0001	0,4763	0,4736	-0,0028	0,2580	0,2440	-0,0140	0
41	Azure Speech (uk-UA + en-US LID)	0,9651	0,8623	-0,1027	0,3900	0,3890	-0,0010	0,4905	0,4870	-0,0035	0,3379	0,3269	-0,0111	0
42	MMS-1B (ukr adapter)	1,0134	0,8499	-0,1635	0,4107	0,4099	-0,0008	0,5160	0,5043	-0,0118	0,2498	0,2181	-0,0317	0
43	Whisper Base (force_uk)	0,9542	0,8203	-0,1339	0,4731	0,4732	+0,0001	0,5801	0,5562	-0,0239	0,3052	0,2817	-0,0235	0
44	W2V-BERT Ukrainian v2.1	1,0034	0,9131	-0,0903	0,4958	0,4957	-0,0001	0,5845	0,5619	-0,0225	0,3179	0,2959	-0,0220	0
45	Whisper Base (auto)	0,9469	0,8203	-0,1267	0,5185	0,5187	+0,0002	0,6307	0,6082	-0,0225	0,3294	0,3073	-0,0222	0
46	Whisper Tiny (force_uk)	0,9907	0,9083	-0,0823	0,6171	0,6175	+0,0004	0,7000	0,6854	-0,0146	0,3703	0,3552	-0,0151	0
47	Whisper Tiny (auto)	0,9922	0,9116	-0,0805	0,6582	0,6587	+0,0005	0,7405	0,7268	-0,0137	0,3964	0,3819	-0,0145	0
48	MiMo-v2-Omni	1,0125	0,9489	-0,0635	0,8085	0,8088	+0,0003	0,8402	0,8371	-0,0031	0,5776	0,5643	-0,0133	0

#	Model	EN-WER	EN ITN	Δ EN	UK-WER	UK ITN	Δ UK	WER	WER ITN	Δ WER	CER	CER ITN	Δ CER	Fails
49	MiMo-v2.5	1,1444	1,1130	-0,0314	0,9145	0,9146	+0,0000	0,9377	1,0012	+0,0636	0,6821	0,7276	+0,0455	3

Примітка. Стовпець Fails позначає кількість аудіозаписів (із 448), для яких модель не змогла згенерувати транскрипцію (через тайм-аут API, внутрішню помилку сервера, зациклення або спрацювання фільтрів безпеки). Для таких записів розрахунок метрик не проводився (або їм присвоювався максимальний штраф), що впливає на фінальну надійність моделі.

ДОДАТОК Б

(обов’язковий)

Результати оцінювання моделей наскрізного перекладу

Таблиця Б.1 -- Результати оцінювання моделей наскрізного перекладу

Model	WER	CER	Fails
Gemini 3.1 Pro (UK→EN)	0,2546	0,1537	0
Gemini 3 Flash (UK→EN)	0,3153	0,1980	0
Soniox Async v4 (UK→EN)	0,3663	0,2220	0
Canary 1B v2 (UK→EN)	0,5062	0,3197	0
SeamlessM4T v2 Large (UK→EN)	0,5768	0,4162	0
Azure Speech uk-UA+LID (UK→EN)	0,6355	0,4323	0
Azure Speech uk-UA (UK→EN)	0,6355	0,4323	0

Наведені дані ілюструють методологічну проблему оцінювання систем семантичного перекладу (Audio-LLM) за допомогою метрики WER. Найкраща за змістом модель Gemini 3.1 Pro демонструє WER 0,2546 (понад 25 % помилок). Це відбувається не через поганий переклад, а через те, що алгоритм вирівнювання жорстко штрафує модель за використання синонімів та правильну зміну синтаксичної структури речення. Водночас вищий WER для традиційних архітектур (наприклад, 0,5768 для SeamlessM4T) зумовлений саме грубими помилками калькування та втратою сенсу.

ДОДАТОК В

(обов’язковий)

Приклади дослівного та семантичного перекладу IT-сленгу

Таблиця В.1 -- Порівняння Canary 1B v2 та семантичної адаптації Gemini 3.1 Pro

ID запису	**Еталонна українська транскрипція**	**Еталонний переклад**	**Canary 1B v2**	**Gemini 3.1 Pro**
4_azure	Підключив Azure Queue Storage, і background worker тепер стабільно забирає jobs без дублювання.	I connected Azure Queue Storage, and the background worker now reliably picks up jobs without duplicates.	If you're a background worker, you're unstable, and jobs are taken away without duplication by hiring a hostel.	I connected Azure Queue Storage, and the background worker now reliably picks up jobs without duplication.
10_sql	Моя думка по транзакціях: ізоляційний рівень READ COMMITTED за замовчуванням в PostgreSQL дає змогу уникнути dirty reads.	My view on transactions: the default READ COMMITTED isolation level in PostgreSQL allows you to avoid dirty reads.	My opinion on transactions. The isolation level of the REIT committee for customs in PostgresQuel allows you to avoid dirty tricks.	My thoughts on transactions. The default read committed isolation level in PostgreSQL allows you to avoid dirty reads.
12_azure	Сьогодні перевіряв Azure DevOps pipeline, бо один stage падав під час test run.	Today I checked the Azure DevOps pipeline because one stage was failing during the test run.	Today I checked the DevOps pipeline for an update on one stage of the test.	Today I checked the Azure DevOps pipeline because one stage was failing during the test run.
9_dotnet_short	Додав health checks у .NET сервіс, тому інфра швидше бачить проблемні інстанси.	I added health checks to the .NET service, so the infrastructure sees problematic instances faster.	He added health checks to the .NET service, so the info sees problem instances more quickly.	I added health checks to the .NET service, so the infrastructure sees problematic instances faster.

ID запису	Еталонна українська транскрипція	Еталонний переклад	Canary 1B v2	Gemini 3.1 Pro
18_react_short	Сьогодні спростив conditional rendering, бо JSX став занадто важким для читання.	Today I simplified conditional rendering because the JSX became too hard to read.	Conditional rendering has been made easier today because JSX has become too complicated to read.	Today I simplified conditional rendering because the JSX became too hard to read.