ПРИКЛАДНА МАТЕМАТИКА

УДК 519.6 *http://dx.doi.org/10.30970/vam.2024.32.12428*

МІЖМОВНА АУДІОТРАНСКРИПЦІЯ З ВИКОРИСТАННЯМ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ

С. Матвіїв, Ю. Музичук

Львівський національний університет імені Івана Франка,
вул. Університетська 1, Львів, 79000, Україна
e-mail: sofiia.matviiv.pmp@lnu.edu.ua, yuriy.muzychuk@lnu.edu.ua

Розглянуто задачу міжмовної аудіотранскрипції – автоматичного перетворення аудіозапису мовлення однією мовою на текст іншою мовою. Основна мета полягає у розробці системи для транскрибування аудіо за допомогою рекурентних нейронних мереж (RNN). Проаналізовано сучасні підходи до вирішення цієї задачі, враховуючи використання нейромереж типу LSTM для поліпшення точності розпізнавання мовлення. Було проведено чисельні експерименти з використанням датасету LibriSpeech та оцінено ефективність моделі за допомогою метрик точності, Word Error Rate (WER) та BLEU для перекладу. Отримані результати свідчать про високу точність моделі та можливість її подальшого вдосконалення для поліпшення якості транскрипції й перекладу мовлення.

Ключові слова: міжмовна аудіотранскрипція, рекурентні нейронні мережі, розпізнавання мовлення, машинний переклад, точність моделі, сучасні моделі міжмовного транскрибування.

1. Вступ

Обмін інформацією швидко набуває все більшого значення у сучасному цифровому світі і з ним розуміння мовлення на різних мовах світу є ключовим у цій справі. Однак для багатьох людей мовний бар'єр може стати перешкодою для ефективного спілкування та доступу до інформації.

Міжмовна аудіотранскрипція має широке застосування, починаючи від автоматичного субтитрування відео до розпізнавання мовлення та перекладу тексту в реальному часі. Ця технологія значно полегшує доступ до інформації для людей з різними мовними потребами.

Проте, незважаючи на значний прогрес у цій галузі, існують деякі виклики, з якими зіштовхуються дослідники. Наприклад, різні діалекти та швидкість мовлення можуть ускладнювати процес транскрипції аудіо в текст і знижувати точність систем. Крім того, існує необхідність у подальшому вдосконаленні систем розпізнавання мовлення для їх оптимальної роботи у реальному часі та у різних умовах.

2. Формулювання задачі

У цьому розділі наведено формулювання задачі міжмовної аудіотранскрипції, огляд моделей, які існують для міжмовного транскрибування та загальна інформація про побудову рекурентної нейронної мережі та мережу LSTM.

3. Формулювання задачі

Завдання полягає у перетворенні мовлення, записаного у вигляді аудіосигналу, на його текстовий еквівалент у іншій мові. Іншими словами, це автоматичне перетворення аудіосигналу з однієї мови на іншу у вигляді письмового тексту.

Основні етапами розв'язання цієї задачі такі:

1. **Попередня обробка аудіосигналу**. Фільтрація шуму, нормалізація гучності та інші операції для підготовки аудіоданих для подальшого аналізу.
2. **Розпізнавання мовлення (ASR)**. Застосування алгоритмів автоматичного розпізнавання мовлення для перетворення аудіосигналу у текст.
3. **Машинний переклад (MT)**. Переклад тексту з однієї мови на іншу.
4. **Генерація текстового результату**. Транскрипція тексту та його переклад генерується у вигляді текстового документа або виводиться на екран користувача.

Ця задача складна через велику різницю між мовами у фонетиці, акцентах, лексиці та синтаксисі. Розв'язання цієї задачі потребує використання передових методів машинного навчання, таких як глибокі нейронні мережі, зокрема рекурентні нейронні мережі, для досягнення точних і надійних результатів.

4. Огляд сучасних моделей міжмовного транскрибування

4.1. Модель на основі трансформерів

Однією з найвідоміших моделей такого типу є Whisper AI [5]. Це система розпізнавання мовлення, розроблена OpenAI, яка використовує гібридний підхід, поєднуючи рекурентні нейронні мережі і трансформери. У випадку з Whisper комбінація RNN і трансформерів застосовують для використання сильних сторін обох архітектур. Компонент рекурентної нейронної мережі може фіксувати тимчасові залежності в мовних даних, тоді як компонент трансформатора може моделювати контекстні зв'язки та залежності між різними частинами аудіовхідного сигналу. Використовуючи цей гібридний підхід, Whisper прагне підвищити точність і продуктивність систем розпізнавання мовлення та її транскрибуваннння. Тренування на такому великому і різноманітному наборі даних дало змогу цій моделі бути стійкою до акцентів, фонового шуму та технічної мови [1].

Підсумовуючи, Whisper AI є сучасною моделлю розпізнавання мови, що використовує передові методи глибинного навчання для досягнення високої точності. Завдяки підтримці багатьох мов і гнучкості налаштування, вона підходить для широкого спектра застосувань [2]. Однак, як і будь-яка інша технологія, Whisper AI має свої переваги та недоліки.

4.2. Модель, яка використовує навчання з підкріпленням

Модель DeepSpeech [4], розроблена компанією Mozilla, є відкритим проєктом і базується на глибоких нейронних мережах, що дає змогу їй ефективно працювати з різними мовами та акцентами.

DeepSpeech використовує рекурентні нейронні мережі для обробки вхідних аудіосигналів. Останні версії можуть охоплювати вдосконалення на основі Long Short-Term Memory (LSTM) або Gated Recurrent Unit (GRU), що допомагають краще зберігати контекст і обробляти послідовності даних.

Таблиця 1

Переваги та недоліки Whisper AI
Advantages and disadvantages of Whisper AI

Переваги	**Недоліки**
Висока точність – whisper AI використовує сучасні методи глибинного навчання для забезпечення високої точності розпізнавання мови	**Високі вимоги до апаратного забезпечення** – для ефективної роботи моделі потрібні потужні обчислювальні ресурси
Підтримка багатьох мов: – модель підтримує розпізнавання багатьох мов, що робить її корисною для глобальних застосувань	**Великі обсяги даних для навчання** – модель потребує великих обсягів навчальних даних для досягнення високої точності
Гнучкість і налаштовуваність – модель можна налаштувати під специфічні потреби користувачів і різні сценарії використання	**Час на навчання** – навчання моделі займає значний час, що може бути перешкодою для швидкого впровадження
Інтеграція з різними платформами – whisper AI можна інтегрувати з різними системами та платформами для забезпечення зручності використання	**Чутливість до якості даних** – якість розпізнавання може значно знижуватися у разі використання неякісних або зашумлених аудіозаписів

Для обробки даних змінної довжини використовують функцію втрат CTC (Connectionist Temporal Classification). Вона дає змогу моделі вчитися вирівнювати послідовності аудіо з текстовими транскрипціями навіть без чіткого відповідника між аудіофрагментами та символами. Після того як модель генерує ймовірності для кожної можливої послідовності символів, застосовується алгоритм декодування (наприклад, beam search), щоб знайти найвірогіднішу послідовність символів, яка відповідає вхідному аудіо. DeepSpeech продовжує розвиватися, його подальші версії вдосконалюються, що робить модель ще потужноішою та ефективнішою для міжмовного транскрибування аудіо. Наведемо декілька переваг і недоліків цієї моделі.

4.3. Спеціалізована мультимовна модель

Модель Jasper [3] є архітектурою на основі рекурентних нейронних мереж, спеціально розробленою для завдань розпізнавання мови. Вона відома своєю здатністю ефективно працювати з аудіозаписами різних мов та акцентів. У цій моделі використовується комбінація згорткових і рекурентних шарів. Це поєднання допомагає моделі ефективно створювати акустичні та фонетичні характеристики мовлення. Основними компонентами архітектури є блоки, які містять згорткові шари. Їх використовують для вилучення характеристик з аудіоданих. А рекурентні шари використовуються для моделювання контексту та послідовностей. В кінці архітектури моделі Jasper зазвичай розташований класифікаційний шар, який застосовують для прогнозування текстової транскрипції аудіозапису.

Модель Jasper є популярним вибором для систем автоматичного розпізнавання

Таблиця 2

Переваги та недоліки DeepSpeech
Advantages and disadvantages of DeepSpeech

Переваги	**Недоліки**
Відкритий код – deepSpeech є проєктом з відкритим кодом, що дає змогу спільноті розробників брати участь у його вдосконаленні	**Великі обчислювальні ресурси** – навчання та розгортання моделей на основі глибинного навчання потребують значних обчислювальних ресурсів
Гнучкість – модель може бути адаптована до різних мов і акцентів, що робить її універсальною	**Великі обсяги даних** – для досягнення високої точності необхідні великі обсяги якісних навчальних даних
Висока точність – завдяки використанню сучасних методів глибинного машинного навчання, модель досягає високої точності розпізнавання мови	

мови, які потребують швидкої відповіді. Щоб підсумувати, наведемо переваги та недоліки для цієї моделі.

5. Побудова рекурентної нейронної мережі для транскрибування аудіо

Рекурентна нейронна мережа [9] – це різновид штучної нейронної мережі, яка використовує послідовні дані. Зазвичай такі алгоритми глибокого навчання використовують до таких проблем: переклад мови, обробка природної мови, розпізнавання мовлення та підписи до зображень; вони входять в такі популярні програми, як Siri, голосовий пошук і Google Translate. Подібно до нейронних мереж прямого зв'язку та згорткових нейронних мереж (CNN), рекурентні нейронні мережі використовують тренувальні дані для навчання. У традиційних глибоких нейронних мережах припускають, що входи та виходи незалежні один від одного, вихід рекурентних нейронних мереж залежить від попередніх елементів у послідовності.

Тоді, коли мережі прямого зв'язку мають різні ваги для кожного вузла, рекурентні нейронні мережі мають однаковий параметр на кожному рівні мережі. Саме ця властивість характерна для цих мереж. Але щоб полегшити навчання з підкріпленням, ці ваги можуть коригувати за допомогою процесів зворотного поширення та градієнтного спуску.

Основною ідеєю RNN є те, що кожен нейрон у шарі використовує свій внутрішній стан, отриманий з попередніх кроків, для обробки поточного вхідного сигналу. Це дає змогу мережі зберігати інформацію про попередні стани та використовувати її для прийняття рішень на основі контексту.

Припустимо, у нас є послідовність вхідних даних з часом. Нехай $x^{(t)}$ позначає вхідні дані в момент часу t, де t може бути будь-яким індексом у послідовності. Кожна одиниця рекурентного шару має свій внутрішній стан, який називається прихованим станом або станом нейрона. Позначимо прихований стан у момент часу t як $h^{(t)}$.

У кожного нейрона в RNN є ваги, які використовують для зв'язку між вхідними

Таблиця 3

Переваги та недоліки моделі Jasper
Advantages and disadvantages of the Jasper model

Переваги	**Недоліки**
Висока точність розпізнавання мовлення – jasper використовує сучасні методи глибинного навчання, що забезпечує високу точність розпізнавання мови	**Високі вимоги до обчислювальних ресурсів** – для ефективної роботи моделі потрібні потужні обчислювальні ресурси
Можливість налаштування для різних мов – jasper підтримує розпізнавання багатьох мов, що робить її корисною для глобальних застосувань	**Складність налаштування для специфічних випадків. Обмежена підтримка деяких мов** – модель може не підтримувати деякі рідкісні мови
Підтримка різних платформ – модель можна інтегрувати з різними системами та платформами для забезпечення зручності використання	**Висока складність початкового налаштування** – початкове налаштування моделі може бути складним і потребувати значних зусиль
Відкрите програмне забезпечення – jasper є відкритим вихідним програмним забезпеченням, що допомагає користувачам змінювати та поліпшувати модель	**Витрати на обчислювальні ресурси** – використання моделі може призводити до високих витрат на обчислювальні ресурси
Активна спільнота розробників – модель підтримується великою спільнотою розробників, що сприяє її розвитку та поліпшенню	**Потребує глибоких знань у галузі машинного навчання** – для ефективного використання моделі потрібні глибокі знання в ділянці машинного навчання

даними, прихованим станом і вихідними результатами. Позначимо ваги між вхідними даними і прихованим станом як W_{xh}, ваги між прихованим станом і самим собою як W_{hh}, а ваги між прихованим станом і вихідними результатами як W_{hy}. Також кожен нейрон може мати зміщення – це додатковий параметр, що додається до вхідної суми в кожному нейроні перед застосуванням активаційної функції. Він допомагає моделі краще підлаштовуватися до даних. Позначимо зміщення для прихованого стану та вихідних результатів як b_h та b_y, відповідно.

Тепер оновлення прихованого стану та вихідних результатів в RNN можуть бути описані так.

Прихований стан

$$h^{(t)} = \text{activation}(W_{xh} \cdot x^{(t)} + W_{hh} \cdot h^{(t-1)} + b_h).$$

Вихідний результат:=

$$y^{(t)} = \text{activation}(W_{hy} \cdot h^{(t)} + b_y),$$

де activation – це функція активації, яка застосовується до зваженого входу нейронів, така як гіперболічний тангенс, сигмоїда або ReLU.

Щоб мережа могла розпізнати аудіо нам потрібна спектрограма перетворення Фур'є (STFT). Перетворивши аудіо, на спектрограму STFT за допомогою Librosa (відкрита бібліотека Python для аналізу аудіо та музики), вибираємо довжину сегмента, який є фіксованим часовим проміжком подачі вхідних даних у модель RNN. Далі перетворюємо спектрограму в масив, де кожен рядок у масиві це рівень частоти, а стовпець – часовий проміжок. Значення в масиві представляють амплітуду звуку. Щоб уникнути об'єднання двох аудіо в один сегмент, у кінці кожного запису додається тиша зі значеннями 0, відповідно до розміру аудіо. Кожен аудіозапис матиме однакову кількість рівнів частоти, але не однакову загальну тривалість. Через це масив транспонується так, щоб кожне аудіо мало однакову кількість стовпців. Далі можна легко об'єднати ці аудіо разом та вводити у модель, використовуючи таку форму масиву.

Проте у практичному застосуванні рекурентна нейронна мережа може мати проблему зникнення градієнта, це коли градієнт помилки зникає або стає надто малим при зворотному поширенні помилки через тривалу залежність в часі. Для того, щоб рекурентна мережа могла ліпше розуміти як працює музика чи мова, їй потрібно довго пам'ятати свої досягнення. Для вирішення цієї проблеми зникнення градієнта найчастіше використовують мережі довгострокової пам'яті.

6. Мережа LSTM

Щоб дозволити глибоким нейронним мережам і довгій пам'яті демонструвати успіхи у розумінні музики чи мови, нам потрібен спеціальний метод або система методів, яка дає можливість зменшити вплив градієнтів із малими значеннями під час зворотного поширення в часі, щоб подолати проблему зникання градієнта. У мережі LSTM [8] використовують додаткові компоненти, такі як "ворота" (gates) [6], які допомагають контролювати потік інформації через мережу. Основна ідея полягає в тому, що LSTM може зберігати та використовувати довгострокові залежності в послідовностях.

Припустимо, у нас є послідовність вхідних даних з часом. Нехай $x^{(t)}$ позначає вхідні дані в момент часу t, де t може бути будь-яким індексом у послідовності. Також існує прихований стан, який позначається як $h^{(t)}$. Він буде передаватися з одного кроку до наступного, дозволяючи мережі зберігати довготривалі залежності. Компоненти LSTM.

– Ворота забування (Forget Gate) – контролюють, яку інформацію потрібно забути з попереднього схованого стану $h^{(t-1)}$ і вхідних даних $x^{(t)}$. Це досягається за допомогою сигмоїдної функції σ, ваг W_f та U_f та зміщення b_f. Ворота мають значення між 0 та 1, де 0 означає, що інформацію потрібно повністю забути, а 1 – повністю зберегти

$$f^{(t)} = \sigma(W_f \cdot x^{(t)} + U_f \cdot h^{(t-1)} + b_f).$$

– Ворота входу (Input Gate) – визначає, яку нову інформацію треба зберегти в схованому стані $h^{(t)}$. Використовується сигмоїдна функція σ, матриці ваг W_i та U_i, які визначають вплив вхідних даних і попереднього прихованого стану на ворота, компонент пам'яті, зміщення b_i. Ворота входу також використовують

функцію гіперболічного тангенса tanh для створення пропозиції нових значень, які можуть бути додані до прихованого стану

$$i^{(t)} = \sigma(W_i \cdot x^{(t)} + U_i \cdot h^{(t-1)} + b_i)$$

$$g^{(t)} = \tanh(W_g \cdot x^{(t)} + U_g \cdot h^{(t-1)} + b_g).$$

– Компонент пам'яті – це обчислювальний блок, який оновлює попередній прихований стан $h^{(t-1)}$ на основі воріт забування і воріт входу. Використовуємо поелементне множення ($\odot$) між воротами забування та попереднім прихованим станом, а також між воротами входу та пропозицією нових значень. Цей компонент допомагає мережі вирішувати, яку інформацію треба забути, а яку зберегти

$$c^{(t)} = f^{(t)} \odot c^{(t-1)} + i^{(t)} \odot g^{(t)},$$

де $c^{(t)}$ – прихований стан у момент часу t.

– Вихідний блок використовують, щоб визначити, яку інформацію зі схованого стану $h^{(t)}$ буде виведено як вихід. Застосовується сигмоїдна функція σ та ваги W_o та U_o та зміщення b_o. Прихований стан $h^{(t)}$ проходить через функцію активації, яка може бути, наприклад, tanh, щоб отримати вихід $y^{(t)}$.

Такий алгоритм дає змогу мережі LSTM вивчати лише потрібний і корисний контекст. Через ці причини модель LSTM зараз так часто і широко застосовують для прогнозування поведінки, що залежить від часу.

7. Програмна реалізація

У цьому розділі наведено основні частини програмної реалізації моделі міжмовного транскрибування, приклади роботи моделі та аналіз результатів роботи моделі.

8. Середовище розробки

Програму було створено за допомогою Jupyter Notebook. Для навчання моделі обрали мову Python з бібліотекою Tensorflow. Крім того, у цьому середовищі встановлені всі потрібні для створення і навчання рекурентної нейронної мережі бібліотеки (numpy, keras, librosa та ін.).

9. Набір даних

Використовується датасет LibriSpeech [7], який містить аудіозаписи англійською мовою (1000 годин). Для роботи було взято частину цього датасету – 100 годин мовлення. На початку роботи з датасетом для кожного аудіофайлу обчислюють коефіцієнти мел-кепстральних ознак (Mel-Frequency Cepstral Coefficients). MFCC є важливими ознаками, які часто використовують у розпізнаванні мови, оскільки вони ефективно представляють спектральні характеристики звуку.

10. Архітектура моделі рекурентної нейронної мережі

На рис. зображена схема архітектури моделі. Модель складається з таких компонентів.

- **Вхідний шар**. Вхідні дані представляються у вигляді послідовності коефіцієнтів MFCC.
- **LSTM шари**. Обробляють послідовні дані.
- **TimeDistributed Dense шар**. Застосовує повнозв'язний шар до кожного тимчасового кроку.
- **Activation шар**. Використовує функцію softmax для виходу.

Тренується нейронна мережа з використанням оптимізатора Adam.

Layer (type)	Output Shape	Param #
the_input (InputLayer)	[(None, None, 13)]	0
lstm_0 (LSTM)	(None, None, 128)	72704
lstm_1 (LSTM)	(None, None, 128)	131584
lstm_2 (LSTM)	(None, None, 128)	131584
time_distributed (TimeDistributed)	(None, None, 29)	3741
softmax (Activation)	(None, None, 29)	0

Рис. 1. Архітектура моделі рекурентної нейронної мережі

11. Приклади роботи моделі

11.1. Приклад 1. Вдала робота моделі

Оригінальне речення:	"She had never seen a river so wide before"
Транскрипція моделі:	"Вона ніколи раніше не бачила такої широкої річки"

11.2. Приклад 2. Невдалий переклад окремого слова

Оригінальне речення:	"The weather was surprisingly warm for a winter day"
Транскрипція моделі:	"Погода була несподівано гарною для зимового дня"
Проблема:	У цьому випадку модель правильно розпізнала зміст речення, але припустилася помилки у перекладі. Замість точного перекладу слова "warm" як "теплою" модель використала слово "гарною". Це може ввести в оману, оскільки теплою було б точнішим перекладом для опису погоди взимку

11.3. Приклад 3. Помилка в контекстуальному розумінні

Оригінальне речення:	"He quickly realized that the solution was far from perfect"
Транскрипція моделі:	"Він швидко зрозумів, що рішення було майже ідеальним"
Проблема:	Модель неправильно переклала фразу "far from perfect". Замість "далеко не ідеальне" вона використала "майже ідеальним". Цей переклад повністю змінює значення речення на протилежне

12. Оцінка моделі

На підставі проведених тестувань та аналізу результатів отримали такі показники.

Точність моделі становила 65%, що означає, що модель правильно передбачила 65% символів у транскрипції. Це свідчить про досить високу точність.

Оцінка WER для моделі становила 0.25 (25%). Це означає, що в середньому 25% слів у транскрипції були помилково вставлені, видалені або замінені. Хоча цей показник прийнятний, зниження WER могло значно поліпшити якість транскрипцій.

Модель переклала текст з англійської мови на українську. BLEU-оцінка для перекладу становила 0.55 (55%). Це підтверджує, що переклад моделі добре відповідає правильному перекладу.

Розроблена рекурентна нейронна мережа демонструє високий рівень ефективності в задачах транскрипції та перекладу аудіоданих. З показниками точності в 65%, WER в 25% та BLEU в 55%, модель демонструє, що може забезпечити достатню якість для багатьох практичних застосувань. Проте існують можливості для подальшого вдосконалення, особливо в зниженні кількості помилок у транскрипції та поліпшенні точності перекладу.

13. Висновки

Зібрано й подано загальні відомості про рекурентні нейронні мережі, огляд кількох сучасних моделей міжмовного транскрибування та наведено оцінки роботи створеної рекурентної нейронної мережі.

Отримані результати свідчать про можливість успішного використання рекурентних нейронних мереж у сфері обробки аудіоданих. Моделі розпізнавання мови, побудовані на базі рекурентних нейронних мереж, демонструють високу точність та ефективність у визначенні текстових транскрипцій звукових сигналів.

У майбутньому можливе розширення функціональності системи шляхом використання розгалуженіших моделей нейронних мереж, поліпшення методів аудіообробки та мовленнєвого аналізу, а також інтеграція додаткових мов ті мовленнєвих моделей.

Список використаної літератури

1. Radford A. Robust Speech Recognition via Large-Scale Weak Supervision / Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, Ilya Sutskever. – URL:

https://arxiv.org/pdf/2212.04356
2. Dickson B. How will OpenAI's Whisper model impact AI applications? / Ben Dickson. – URL: https://venturebeat.com/ai/how-will-openais-whisper-model-impact-ai-applications
3. Li A. Jasper: An End-to-End Convolutional Neural Acoustic Model / Ason Li, Vitaly Lavrukhin, Boris Ginsburg, Ryan Leary, Oleksii Kuchaiev, Jonathan M. Cohen, Huyen Nguyen, Ravi Teja Gadde. – URL: https://research.nvidia.com/publication/2019-04_jasper-end-end-convolutional-neural-acoustic-model
4. Mozilla DeepSpeech. DeepSpeech: An Open-Source Speech-To-Text Engine. – URL: https://github.com/mozilla/DeepSpeech
5. WhisperUI. How OpenAI Whisper works. – URL: https://whisperui.com/blog/how-openai-whisper-works
6. Gers F.A. Learning to Forget: Continual Prediction with LSTM Felix A. Gers, Jurgen Schmidhuber, Fred Cummins. – URL: https://www. researchgate.net /publication/12292425 _Learning _to _Forget _Continual _Prediction _with _LSTM
7. Papers with Code LibriSpeech. – URL: https://paperswithcode.com/dataset/librispeech
8. Olah C. Understanding LSTM Networks / Christopher Olah. – URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs
9. IBM. What is a recurrent neural network? URL: https://www.ibm.com/topics/recurrent-neural-networks

Стаття: надійшла до редколегії 27.10.2024
доопрацьована 07.11.2024
прийнята до друку 14.11.2024

CROSS-LINGUAL AUDIO TRANSCRIPTION USING RECURRENT NEURAL NETWORKS

S. Matviiv, Yu. Muzychuk

Ivan Franko National University of Lviv,
1, Universytetska str., 79000, Lviv, Ukraine
e-mail: sofiia.matviiv.pmp@lnu.edu.ua, yuriy.muzychuk@lnu.edu.ua

The study addresses the task of cross-lingual audio transcription, where speech in one language is automatically converted into written text in another. A system based on recurrent neural networks, specifically those utilizing long short-term memory (LSTM) units, has been developed to perform this task. The need for multilingual access to spoken content is emphasized, given ongoing challenges posed by language barriers. Existing models such as Whisper, DeepSpeech, and Jasper have been reviewed to assess their architectural features and inform the proposed approach. The implemented system includes preprocessing of audio, speech recognition, machine translation, and text generation. Training was conducted on a subset of the LibriSpeech dataset using mel-frequency cepstral coefficients extracted from audio inputs. The model architecture incorporates LSTM layers, a TimeDistributed dense layer, and Softmax output. Implementation was carried out in Python using TensorFlow and supporting libraries. Evaluation based on Word Error Rate and BLEU score shows strong performance. The findings indicate the system's effectiveness and potential for further development in multilingual speech processing applications.

Key words: cross-lingual audio transcription, recurrent neural networks, speech recognition, machine translation, model accuracy, modern cross-lingual transcription models.