

CS-399

Mini-Project I

LLMs for ASR Error Correction

Submitted to :
Dr. Nataraj K S Sir

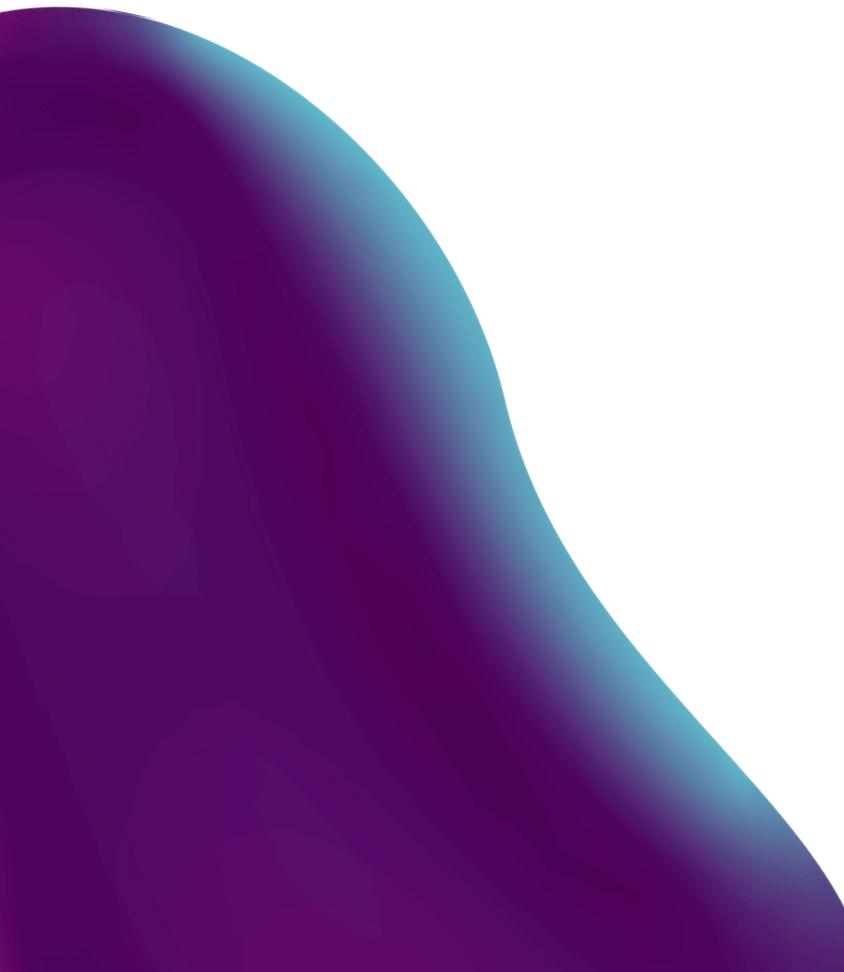
Table of Contents

- Problem Statement
- What is ASR
- Related Works
- Whisper model
- How ASR works in Whisper
- ASR Architecture
- Data and Methods
- Results and Discussion
- Conclusion

Problem Statement

Our project aims to explore the efficacy of using ChatGPT, a generative large language model (LLM), for automatic speech recognition (ASR) error correction.

While ASR error correction presents a promising avenue for post-processing speech recognition outputs, current supervised training methods reliant on decoding results from specific ASR systems can be computationally intensive and lack versatility.



Leveraging the capabilities of ChatGPT, known for its proficiency in natural language processing tasks, the project seeks to investigate both unconstrained and constrained approaches to ASR error correction using ASR N-best outputs.

Furthermore, our project aims to evaluate the performance of this generative LLM approach across different ASR architectures (transducer and attention-encoder-decoder based) and multiple test sets, including zero and 1-shot settings.

What is ASR?

Objective: Automatic Speech Recognition Systems (ASR) aim to convert spoken language into text, crucial for human-computer interaction.

Working Principle: ASR systems utilize advanced algorithms and machine learning techniques to understand human speech, decoding phonetic nuances and contextual meanings.

Related Work

Two stage error correction by “Multi-stage large language model correction for speech recognition”

1. Traditional language models to identify low-confidence segments within ASR transcripts.
2. Utilizing LLMs to address these identified uncertainties

Post-processing paradigms for ASR outputs using LLMs by “Generative speech recognition error correction with large language models and task-activating prompting.”



“Can generative large language models perform ASR error correction?” explores the **effectiveness of generative LLMs like ChatGPT.**



SoftCorrect by “Softcorrect: Error correction with soft detection for automatic speech recognition.”

“Large language models are efficient learners of noise-robust speech recognition.” for tackling the issue of **Noise robustness in ASR**

The background features three large, semi-transparent circles with a gradient from dark purple to bright blue. One circle is positioned at the top left, another is centered in the middle, and a third is located at the bottom left.

Whisper Model

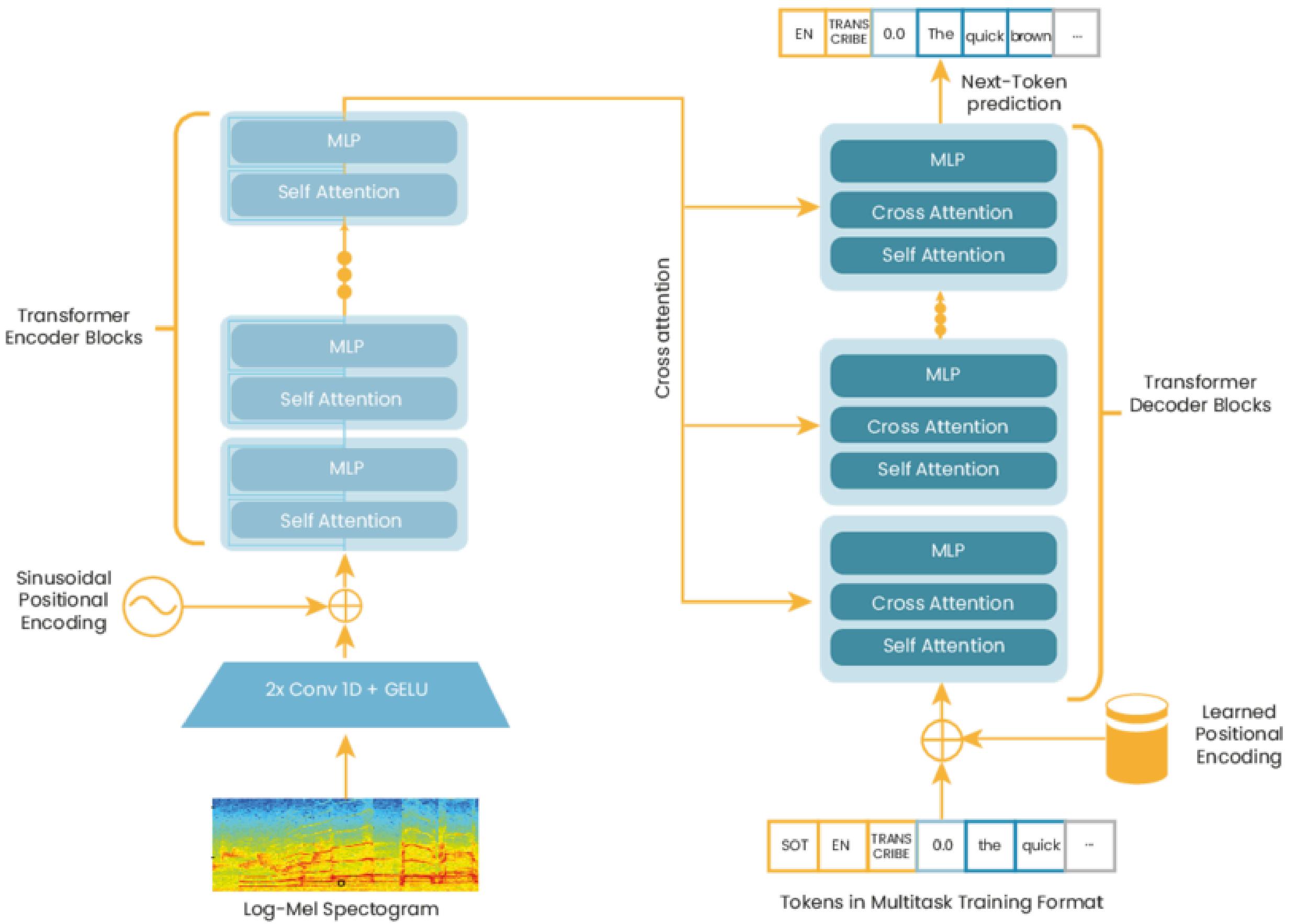


OpenAI/Whisper

Whisper, created by OpenAI, is a powerful automatic speech recognition (ASR) model with some unique capabilities:



- **Large-scale training:** Whisper is trained on a massive dataset of labeled speech data (over 680,000 hours!), making it versatile and adaptable to various speech patterns and accents.
- **Multilingual:** Whisper can not only transcribe speech in English but also translate speech in multiple languages to English. This breaks down language barriers and widens its application potential.
- **Weakly-supervised learning:** Unlike some ASR models that require heavily labeled data, Whisper leverages a weakly-supervised learning approach. This means it can learn effectively even from data where the exact words aren't precisely labeled
- **Open-source:** OpenAI released Whisper as an open-source model, allowing for wider adoption and contribution from the developer community.



Transformer architecture: Whisper utilizes a transformer-based architecture, a powerful neural network structure commonly used in natural language processing tasks. This likely contributes to its accuracy and efficiency.

How ASR works in Whisper?

Input Preparation

Whisper takes audio as input, which is first converted into a spectrogram. This spectrogram represents the frequencies present in the audio over time.

Encoder

The converted spectrogram is fed into the encoder part of the transformer model. This encoder is like a powerful feature extractor, analyzing the spectrogram and capturing the important characteristics of the speech.

Decoder

The decoder takes the encoded information from the encoder and uses it to predict the text word by word. It considers the previously predicted words and the encoder's output to make an educated guess about the next word in the sequence.

Local Agreement

Whisper employs a unique technique called "local agreement." Here, the decoder outputs are analyzed across a sliding window of audio chunks. Only words that are consistently predicted across consecutive chunks are considered confident and included in the final transcript. This helps to reduce errors and improve accuracy.

Weakly Supervised Learning

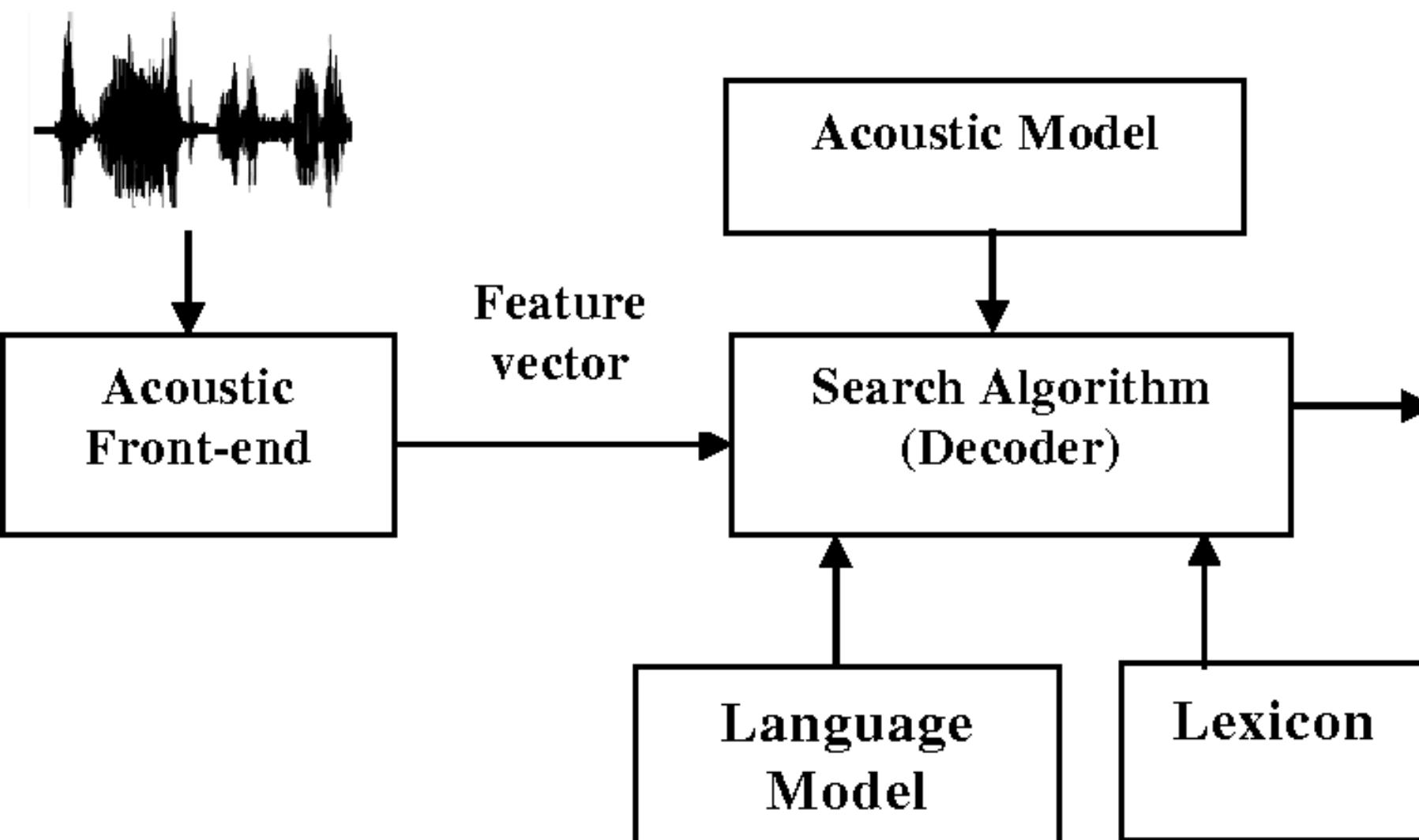
Unlike some traditional ASR models that require large amounts of precisely labeled data, Whisper is trained using a technique called "weakly-supervised learning."

Additional Features

Whisper's model is trained to perform multiple tasks beyond just speech-to-text. It can also predict speaker information, language used, and even provide partial timestamps for different phrases within the audio.

ASR Architecture

Speech Utterance

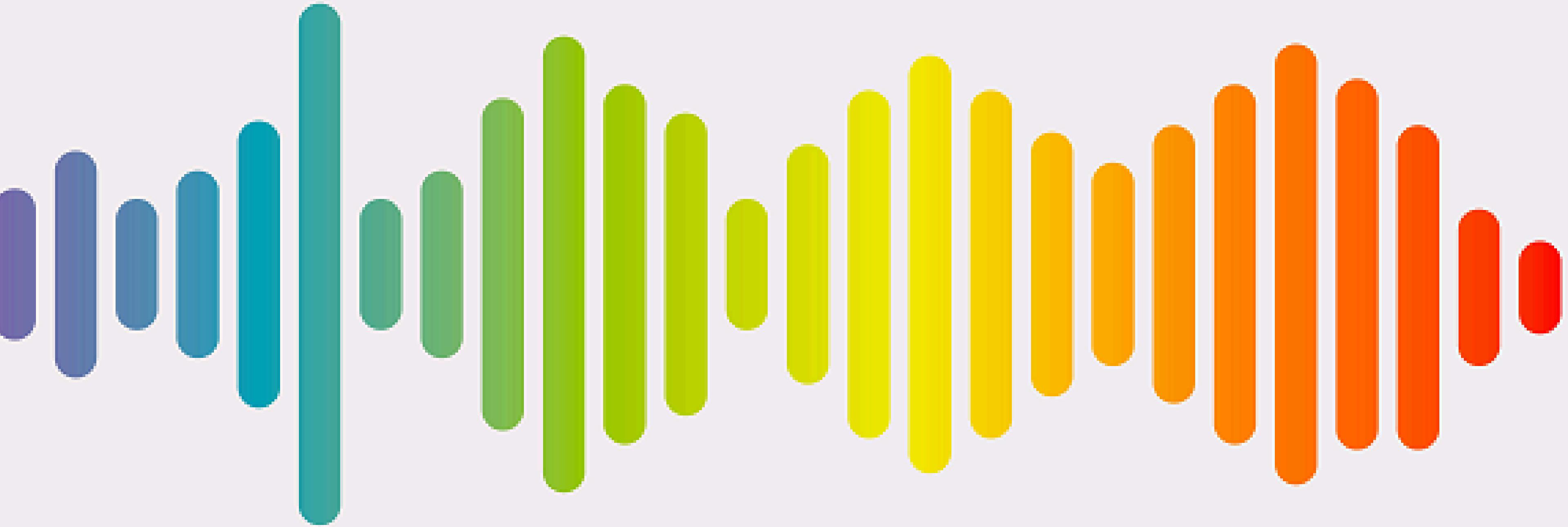


The block diagram illustrates an Automatic Speech Recognition (ASR) system. Speech is captured as a sound wave and converted into features like MFCCs.

These features are then analyzed by acoustic and language models, along with a word dictionary, to find the most probable sequence of words (sentence) that corresponds to the spoken utterance.

The background features a dark purple gradient with three semi-transparent, rounded rectangular shapes. One large shape is centered in the middle, another is in the upper left corner, and a third is in the lower left corner. All shapes have a soft blue-to-purple gradient.

Data and Methods



Dataset: */libriSpeech_asr*

Methods

Enhanced Output

The modification allows printing multiple hypotheses, their probabilities, and temperatures during transcription, offering a deeper understanding of the model's behavior.

Temperature Insight

:Access to temperature values reveals the diversity of generated hypotheses, aiding in understanding model variations.

Efficient Iteration

The code efficiently iterates over hypotheses and temperatures simultaneously using the zip function, simplifying access to both sets of data.

Methods

Detailed Hypothesis Info

Each hypothesis includes temperature, decoded text, and log probability, enhancing understanding and decision-making.

Improved Readability

An empty line is added after printing each hypothesis to enhance readability, making it easier for users to distinguish between different hypotheses.

Analysis Empowerment

Users can analyze model behavior and make informed decisions based on confidence levels, improving usability.

Methods

Real-World Applicability

Particularly useful in scenarios with multiple plausible hypotheses, aiding reliability assessment.

Flexible Parameter Exploration

The modification facilitates experimentation with different temperature values, allowing users to explore their impact on hypothesis diversity and transcription accuracy, thereby supporting model fine-tuning and optimization efforts.

Overall Impact

The modification significantly enhances the transcription process by providing users with more insights into the model's predictions and the confidence levels associated with each hypothesis, ultimately improving the usability and reliability of the transcription system.

Error Correction Methods

1. Zero-shot Unconstrained
2. Zero-shot Constrained
3. Task-Activating Prompting (TAP)

Results

Windows PowerShell X + -

```
PS C:\Users\Santhosh\Desktop\Mini Project\samples> python run.py
C:\Users\Santhosh\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\whisper\t
r
rize.py:115: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
Detecting language using up to the first 30 seconds. Use '--language' to specify the language
Detected language: English
Temperature: 0.0
Hypothesis: He hoped nothing serious had happened to drag her out at such an untoward time.
Log probability: -0.21963510059175037

Temperature: 0.2
Hypothesis: He hoped nothing serious had happened to drag her out at such an untoward time.
Log probability: -0.21963510059175037

Temperature: 0.4
Hypothesis: He hoped nothing serious had happened to drag her out at such an underwater time.
Log probability: -0.2935778216311806

Temperature: 0.6
Hypothesis: He hoped nothing serious had happened to drag her out of such an untoward time.
Log probability: -0.28874183836437406

Temperature: 0.8
Hypothesis: He hoped nothing serious had happened to drag her out at such an untawored time.
Log probability: -0.5559662864321754

Temperature: 1.0
Hypothesis: He hoped nothing serious had happened to drag her out of such an untoward time.
Log probability: -0.27312274206252324

[00:00.000 --> 00:04.200] He hoped nothing serious had happened to drag her out of such an untoward time.
  He hoped nothing serious had happened to drag her out of such an untoward time.
PS C:\Users\Santhosh\Desktop\Mini Project\samples>
```

Sample 1



ChatGPT

Based on the provided log probabilities, the hypothesis with a temperature of 0.4 appears to be the most plausible, as it has the highest log probability (-0.2887) among the options given. Therefore, the best hypothesis is:

"But as the rain gave not the least sign of cessation he observed, I think we shall have to go back. Never."



Windows PowerShell X +

```
PS C:\Users\Santhosh\Desktop\Mini Project\samples> python run.py
C:\Users\Santhosh\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\whisper\tube.py:115: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
Detecting language using up to the first 30 seconds. Use '--language' to specify the language
Detected language: English
Temperature: 0.0
Hypothesis: But as the rain gave not the least sign of cessation he observed, I think we shall have to go back. Never.
Log probability: -0.2760465217359138

Temperature: 0.2
Hypothesis: But as the rain gave not the least sign of cessation he observed, I think we shall have to go back. Never.
Log probability: -0.2760465217359138

Temperature: 0.4
Hypothesis: But as the rain gave not the least sign of cessation he observed, I think we shall have to go back. Never.
Log probability: -0.2886997569691051

Temperature: 0.6
Hypothesis: But as the rain gave not the least sign of cessation, he observed, I think we shall have to go back... Never.
Log probability: -0.5182345895206227

Temperature: 0.8
Hypothesis: But as the rain gave not the least sign of cessation, he observed, I think we shall have to go back.
Log probability: -0.4804588635762533

Temperature: 1.0
Hypothesis: But as the rain gave not the least sin of cessation, He observed, I think we shall have the way back. Never.
Log probability: -1.031834956577846

[00:00.000 --> 00:06.340] But as the rain gave not the least sin of cessation, He observed, I think we shall have the
[00:06.340 --> 00:07.340] way back.
[00:07.340 --> 00:08.340] Never.
  But as the rain gave not the least sin of cessation, He observed, I think we shall have the way back. Never.
PS C:\Users\Santhosh\Desktop\Mini Project\samples>
```

Sample 2



ChatGPT

Based on the provided log probabilities, the hypothesis with a temperature of 0.2 appears to be the most plausible, as it has the highest log probability (-0.2196) among the options given.

Therefore, the best hypothesis is:

"He hoped nothing serious had happened to drag her out at such an untoward time."



```
PS C:\Users\Santhosh\Desktop\Mini Project\samples> python run.py
C:\Users\Santhosh\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\whisper
rbe.py:115: UserWarning: FP16 is not supported on CPU; using FP32 instead
    warnings.warn("FP16 is not supported on CPU; using FP32 instead")
Detecting language using up to the first 30 seconds. Use '--language' to specify the language
Detected language: English
Temperature: 0.0
Hypothesis: I cannot understand how you should know me while I have no knowledge of you.
Log probability: -0.15226897440458598

Temperature: 0.2
Hypothesis: I cannot understand how you should know me while I have no knowledge of you.
Log probability: -0.15226897440458598

Temperature: 0.4
Hypothesis: I cannot understand how you should know me while I have no knowledge of you.
Log probability: -0.17462030209993062

Temperature: 0.6
Hypothesis: I cannot understand how you should know me while I have no knowledge of you.
Log probability: -0.21124111978631271

Temperature: 0.8
Hypothesis: I cannot understand how you should know me while I have no knowledge of you.
Log probability: -0.25987198478297185

Temperature: 1.0
Hypothesis: I cannot understand how you should know me, while I have no knowledge of you.
Log probability: -0.2385744571685791

[00:00.000 --> 00:04.180] I cannot understand how you should know me, while I have no knowledge of you.
I cannot understand how you should know me, while I have no knowledge of you.
PS C:\Users\Santhosh\Desktop\Mini Project\samples>
```

Sample 3



ChatGPT

In this scenario, the log probabilities are consistent across different temperature settings, with all temperatures yielding the same log probability of approximately -0.2939.

Therefore, the best hypothesis among these options is:

"Oh, but you know me, about me at least."



```
PS C:\Users\Santhosh\Desktop\Mini Project\samples> python run.py
C:\Users\Santhosh\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\whisper
ride.py:115: UserWarning: FP16 is not supported on CPU; using FP32 instead
    warnings.warn("FP16 is not supported on CPU; using FP32 instead")
Detecting language using up to the first 30 seconds. Use '--language' to specify the language
Detected language: English
Temperature: 0.0
Hypothesis: Oh, but you know me, about me at least.
Log probability: -0.2939096768697103

Temperature: 0.2
Hypothesis: Oh, but you know me, about me at least.
Log probability: -0.2939096768697103

Temperature: 0.4
Hypothesis: Oh, but you know me, about me at least.
Log probability: -0.3190212567647298

Temperature: 0.6
Hypothesis: Oh, but you know me, about me at least.
Log probability: -0.364605967203776

Temperature: 0.8
Hypothesis: Oh, but you know me, about me at least.
Log probability: -0.364605967203776

Temperature: 1.0
Hypothesis: Oh, but you know me, about me at least.
Log probability: -0.42574828465779624

[00:00.000 --> 00:03.140] Oh, but you know me, about me at least.
Oh, but you know me, about me at least.
PS C:\Users\Santhosh\Desktop\Mini Project\samples>
```

Sample 4



ChatGPT

Based on the provided log probabilities, the hypothesis with a temperature of 0.0 and 0.2 have the same log probability (-0.1523), and the hypothesis with a temperature of 0.4 has a slightly lower log probability (-0.1746). However, as the temperature increases, the log probability decreases, indicating less confidence in the prediction.

Therefore, the best hypothesis among these options is:

"I cannot understand how you should know me while I have no knowledge of you."

Conclusion

- The hypothesis that "LLMs can correct errors in ASR models" holds promise due to LLMs' advanced linguistic capabilities, including understanding language patterns, semantics, and context.
- LLMs' adaptability to diverse language domains and continuous learning from extensive datasets position them as effective tools for enhancing transcription accuracy, offering a compelling solution to the challenge of error correction in ASR transcriptions.

Thank You!

Team Members

Dungavath Santhosh Naik: 21BCS036

Karan Raghuram Naik: 21BDS051

Nariyam Likhith: 21BCS072

Yuvraaj Bhama: 21BDS071