Assignment

Multilingual Speech Recognition Model Evaluation Report

Introduction

This report presents the evaluation of a multilingual speech recognition model that utilizes a pre-trained multilingual speech recognition system, specifically Multilingual Whisper. The model's performance is assessed using a variety of evaluation metrics, including Word Error Rate (WER), Character Error Rate (CER), METEOR, N-gram-based metrics, and BLEU. The purpose of this evaluation is to gauge the accuracy and linguistic quality of the model's transcriptions across multiple languages.

Data and Testing Environment

Audio Data: The evaluation was conducted using a diverse dataset of audio recordings in multiple languages.

ASR Model: The ASR model is based on Multilingual Whisper, a pre-trained multilingual speech recognition model.

Testing Environment: The evaluation was performed using the Python programming language, utilizing libraries such as NLTK, spaCy, and others.

Evaluation Metrics

1. **Word Error Rate (WER)**

WER is a standard metric for evaluating the accuracy of transcriptions. It measures the percentage of words in the recognized text that are different from the reference (true) text. A lower WER indicates better accuracy.

2. **Character Error Rate (CER)**

CER assesses the quality of the transcriptions at the character level. It measures the percentage of characters in the recognized text that do not match the reference text. A lower CER indicates better character-level accuracy.

3. **METEOR**

METEOR is a metric that considers both precision and recall, accounting for synonyms, stemming, and linguistic variations. It provides a holistic assessment of translation quality.

4. **N-gram Metrics**

N-gram metrics, including precision, recall, and F1-score, evaluate the overlap of n-grams (contiguous sequences of words) between the recognized and reference texts. These metrics capture fluency and coherence in the transcriptions.

5. **BLEU (Bilingual Evaluation Understudy)**

BLEU evaluates translation quality by measuring the similarity between n-grams in the recognized and reference texts. It is commonly used in machine translation tasks but can be adapted for ASR evaluation.

Evaluation Results

The following table summarizes the results of the evaluation using the mentioned metrics:

Metric	Score
WER	0.3684210526315789
CER	0.14910025706940874
METEOR	0.65
N-gram (N=2)	0.5116279069767442(F1 score)
N-gram (N=3)	0.4094488188976378 (F1 score)
N-gram (N=4)	0.31999999999999995 (F1 score)
N-gram (N=5)	0.22764227642276422 (F1 score)
BLEU	0.41956118868575765

Discussion

WER and CER provide insights into the accuracy of the ASR model's transcriptions. Lower scores indicate better performance in terms of word and character accuracy.

METEOR offers a more comprehensive evaluation that considers synonyms, stemming, and linguistic variations.

N-gram metrics assess the quality of transcriptions by evaluating the overlap of n-grams between the reference and recognized text.

BLEU measures the similarity of n-grams and provides an additional perspective on translation quality.

The combined evaluation using these metrics offers a well-rounded assessment of the multilingual speech recognition model. The choice of metrics depends on specific use cases and priorities.

RAG model to summarize text

Approach

I loaded the RAG tokenizer and model.

I preprocessed the text data by tokenizing it and converting it to a format that is compatible with the RAG model.

I generated the summary using the generate() method of the RAG model.

I decoded the summary and printed it to the console.

Here is a more detailed explanation of each step:

Step 1: Load the RAG tokenizer and model

I used the AutoTokenizer.from_pretrained() and AutoModelForSeq2SeqLM.from_pretrained() methods to load the RAG tokenizer and model from the Hugging Face Transformers library.

Step 2: Preprocess the text data

I used the rag_tokenizer.tokenize() method to tokenize the text data. This converts the text data into a list of tokens, which are the basic units of input to the RAG model.

I then used the inputs["input_ids"].repeat(5, 1) code to repeat the input IDs 5 times. This is because the RAG model expects to receive 5 context input documents for each input document.

Step 3: Generate the summary

I used the rag_model.generate() method to generate the summary of the text data. This method takes the input IDs and the context input IDs as input and generates a summary as output.

Step 4: Decode and print the summary

I used the rag_tokenizer.decode() method to decode the summary. This converts the summary from a list of token IDs to a string.

I then used the print() function to print the summary to the console.

Conclusion

The multilingual speech recognition model was evaluated using WER, CER, METEOR, N-gram, and BLEU metrics, which provide valuable insights into its performance. The model demonstrates good performance characteristics, and the choice of metrics can be tailored to meet specific requirements. and perform summarization of the text using rag model This report serves as a basis for understanding the model's capabilities and identifying areas for improvement in multilingual ASR applications .