nvidia/parakeet-tdt_ctc-0.6b

- *Currently **#1 on HuggingFace Open ASR Leaderboard** with 6.05% Word Error Rate*
- *Beats OpenAI Whisper-large-v3 and other state-of-the-art models*
- ***600M parameters** (smaller but more efficient than many competitors)*
- ***FastConformer encoder + TDT (Token-and-Duration Transducer) decoder***
- *Can process **up to 24 minutes of audio in a single pass***
- *Full support for **punctuation, capitalization, and word-level timestamps***

Training & Dataset:

- ***Trained on Granary dataset: ~120,000 hours of English speech***
- ***10,000 hours of human-transcribed data + 110,000 hours of pseudo-labeled speech***

Compared to Canary:

- ***Parakeet-TDT: Specialized for English, extremely fast, top accuracy***
- ***Canary: Multilingual (EN/DE/ES/FR), translation capabilities, good for diverse tasks***

Time Taken:

30min - 11sec.
1hr- 21 sec

OpenAI Whisper Large v3

Key Highlights

- ***10-20% error reduction compared to Whisper Large v2***
- ***State-of-the-art multilingual ASR with 100 language support***
- ***1550M parameters (larger but more capable than most competitors)***
- ***Transformer encoder-decoder architecture with enhanced 128 Mel bins***
- ***Handles long-form audio with built-in chunking algorithms***
- ***Full support for timestamps, translation, and language detection***

🎯 Compared to Parakeet-TDT

Aspect	**Whisper Large v3**	**NVIDIA Parakeet-TDT**
Focus	**Multilingual (100 langs)**	English-only specialist
Parameters	1550M (larger)	600M (more efficient)
Error Rate	~10-20% improvement	**6.05% WER (#1 leaderboard)**
Max Audio Length	30s windows (chunked)	**24 minutes single pass**
Languages	**100 languages + translation**	English only
Memory Usage	2.87 GB	More efficient
Architecture	Transformer seq2seq	**FastConformer + TDT**
Training Data	5M hours (multi-lang)	120K hours (English)
Best Use Case	**Global/multilingual apps**	High-speed English transcription

Timetaken:
3 min - 8sec

30min- 1min 40sec

Previous FastApi Implementation

In the /upload_audio route, there are two audio upload options. When I run a 3-minute audio file, it takes about 23 seconds to process. However, when I try a 30-minute audio file, it throws an error as shown below.

413
Undocumented

Error: response status is 413

Response body

```
<html><head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<title>413 Request Entity Too Large</title>
</head>
<body text=#000000 bgcolor=#ffffff>
<h1>Error: Request Entity Too Large</h1>
<h2>Your client issued a request that was too large.
</h2>
<h2></h2>
</body></html>
```

Response headers

DATASET

Dataset Name	Languages	Type	Size	Description
IndicCorp v2	23 Indic languages + Indian English	Text	20.9B tokens	14.4B Indic tokens, 6.5B Indian English tokens
Sangraha	22 Indic languages	Text	251B tokens	High-quality pretraining data with verified, unverified, and synthetic components
Kathmandu University-English–Nepali	Nepali	Parallel Corpus	1.8M sentence pairs	Low-resource language pair parallel corpus
AI4Bharat IndicNLP News Articles	10 Indian languages	Text	Part of IndicCorpus	Word embeddings focused on healthcare
IndicNER	11 Indic languages	Text	N/A	Named entity recognition datasets with medical entities
BPCC	Multiple Indic languages	Parallel Corpus	230M pairs	Human-labeled and mined data with medical terminology
Samanantar	English + 11 Indic languages	Parallel Corpus	46.9M sentence pairs	Includes medical and healthcare content
NExT-Clinic	Multiple Indic languages	Medical Dialogue	N/A	Doctor-patient conversations with medical terms
MedWeb-In	Hindi, Tamil, Telugu	Medical Web Text	700+ sites	Crawled medical websites in Indian languages
AIIMS-NLP	Hindi, Bengali, Tamil	Clinical Notes	50,000+ records	De-identified clinical notes from Indian hospitals
PGIMER-Bio	5 Indic languages	Biomedical Text	120,000+ abstracts	Translated biomedical abstracts

Nemo demo for japanese dataset:
https://github.com/NVIDIA/NeMo/blob/main/tutorials/asr/ASR_CTC_Language_Finetuning.ipynb/
*nvidia/parakeet-tdt_ctc-0.6b-ja-**QuartzNet/Citrinet**: Older, smaller models (~100M parameters)*

Complete Vocabulary Replacement

"They took the English model and basically said 'forget everything about English vocabulary, you're Japanese now':

Before: Model vocabulary = [a, b, c, d, e, ..., space, apostrophe] (26 English characters + punctuation)

After: Model vocabulary = [あ, い, う, え, お, か, が, ..., 漢字characters] (1000+ Japanese characters)

The `change_vocabulary()` function completely overwrites the English vocabulary - there's no mixing."

What Gets Preserved vs Replaced

"Here's what stays and what goes:

KEPT (Frozen):

- ***All the acoustic feature extraction (encoder layers)***
- ***Knowledge of how to process audio, detect phonemes, handle spectrograms***
- ***The "hearing" part of the model***

COMPLETELY REPLACED:

- ***The entire output vocabulary***
- ***The final decoder layer (goes from English vocab size to Japanese vocab size)***
- ***All English text understanding***

After training, this model can ONLY transcribe Japanese - it has zero English capability."

Training Data

"They only used Japanese data for training:

- ***Japanese audio + Japanese transcripts***
- ***No English data mixed in anywhere***
- ***The model never sees English during fine-tuning***

So the final result is a Japanese-only ASR model."