sciendo

Running Automatic Speech Recognition (ASR) Model in the Context of Real Time Data Streaming Architecture

Robert Cristian NECULA
The Bucharest University of Economic Studies, Bucharest, Romania
robert.necula@gmail.com

Pavel-Cristian CRACIUN
The Bucharest University of Economic Studies, Bucharest, Romania
craciunpavel18@stud.ase.ro

Abstract. *In the context of huge volume of data generated in the last couple of years and the need of processing audio stream in real time data processing flows, this paper aims to explore, identify and experiment Automatic Speech Recognition (ASR) models in the context of low latency applications depending on different contexts using specific data sets. Our work aims to explore whether if such models can be deployed in the real time stream audio workflow and for that we compare multiple size of the Whisper model for multiple data sets and analyze the results. Additionally, we tested the models for English and Romanian language. In the future, for the multi-language context it is highly important to analyze using model dedicated fine-tuned for specific language and compare with the results of multi-language models. This paper contributes to the field by providing evidence, comparison, statistics, level of accuracy in order to use ASR models in the context of low latency applications, exploring multiple versions of Whispers models and help researcher in dimensioning the environments for real time infrastructure and extend streaming capabilities with Artificial Intelligence models. Equally, the paper can used for IT professional in running critical systems in different industries to emphasize the techniques to be used in running ASR model in production.*

Keywords: real time, automatic speech recognition (ASR), whispers, multi-language, knowledge distillation, natural language processing.

Introduction

Real-time computing concepts have sparked research interest for at least the last two decades. Author K. G Shin and P. Ramanathan in their paper described a real-time system as a set of cooperating tasks, and these tasks are invoked or activated at regular intervals and have deadlines by which they must complete their execution. This paper identified three major components as (a) *Time*, (b) *Reliability of task execution,* (c) *Environment* of active components of the tasks and argued that their interaction characterizes any real-time systems.

The concept of *Time* is playing a very important role in designing real time architecture because the managed information had the concept that is relying with the information lifecycle. Basically, the information can be used if the data processing is under a specific deadline. For example, the temperature of car brake can have a validity for few seconds.

A hard deadline can be described as the time limit after which the results produced by the corresponding task cease to be useful without affecting the system. A deadline, on the other hand, has been argued to be less importance in terms of invoking other concurrent tasks.

Thus, in the current paper we are focusing on the speech recognition model in the context of streaming architecture and get conclusions how to deal with these models in our interest area.

DOI: 10.2478/picbe-2025-0101

Our study is going to analyze speech recognition models and run in different scenarios to help researcher, developers or solution architects to integrate these models in their studies or products. Additionally, we are testing the multi-language models with samples for Romanian language, from the data accuracy and performance, to see how their models are performing not only for English but also for others languages.

Progress in speech recognition has been boosted by the development of unsupervised pre-training techniques materialized by Wav2Vec 2.0 (Baevski et al., 2020).
However, these models learn directly from raw audio without human labeling which can allow to use large amount of datasets scaled up to 1,000,000 hours of training data (Zhang et al., 2021).

These pre-trained audio encoders learn high-quality representations of speech, but because they are purely unsupervised, they are not performing in decode mapping to the usable outputs, which in specific use case can request performing fine-tuning process. Fine-tuning is a complex process who involved additional cost, time and skills which is, in many cases, hard to implement.

Alec Radford et all, 2022 suggest that while unsupervised pre-training has improved the quality of audio encoders dramatically, the lack of an equivalently high-quality pretrained decoder, combined with a recommended protocol of dataset-specific fine-tuning, is an important limit which can create barriers and constrains in use it. The goal of a speech recognition system, especially in the context of low latency applications, should work reliable "out of the box" in a large range of environments, experiments and use cases without requiring supervised fine-tuning of a decoder for every deployment distribution.

Many studies have demonstrated that speech recognition models that are pre-trained in a supervised approach using many datasets provide higher robustness and can be used in more general use case very effectively (Narayanan et al., 2018, Likhomanenko et al. ,2020 and Chan et al., 2021).

Considering this context, we have focused on research made by Alec Radford et el, 2022 who named Whisper.
The Whisper ASR model has significantly advanced speech recognition by using substantial amount of training data and innovative features such as multitask tokenizers and end-to-end punctuated transcription.

Whisper mode is a supervised pre-training beyond English-only speech which is both multilingual and multitask. It was trained for 680,000 hours of audio, 117,000 hours cover 96 other languages. The dataset also includes 125,000 hours of X→en translation data.
Whisper large model with 1.5 billion parameter sequence-to-sequence (Seq2Seq) pre-trained on 680,000 hours of weakly supervised speech recognition data, demonstrates a strong ability to generalize to many different datasets and domains. However, increasing size of pre-trained ASR models poses challenges when deploying these systems in low-latency settings or resource-constrained hardware (Hendrycks et al., 2020; Zhang et al., 2022).

For this reason, considering our first objective to manage these models in low latency applications, we have identified the recent efforts in natural language processing (NLP) about compressing transformer-based models and define the concept of knowledge distillation (KD). KD has successfully been applied to reduce the size of models such as BERT without a significant performance loss on non-generative classification tasks.

Sanchit Gandhi et al. applied distillation to the Whisper model in the context of Seq2Seq ASR. They addressed the challenge of maintaining robustness to different acoustic conditions of a large-scale open-source dataset covering 10 distinct domains. By pseudo-labelling the data, they ensured consistent transcription formatting across the dataset and provide sequence-level

distillation signal. Additionally, they proposed a simple word error rate (WER) based heuristic for filtering the pseudo-labelled data and demonstrate that it is an effective method for ensuring good downstream performance of the distilled model.
Therefore, we have tested in our experiments in comparison with Whisper model, aiming to have a complete picture about the improvements, performance and reliability of the models.

Literature review

By design, Whisper model is a multitask format, who can perform on the same input audio signal, multiple tasks like transcription, translation, voice activity detection, alignment, and language identification. Very often, these tasks are managed individually, resulting in a relatively complex system around the core speech recognition model.

Whisper models use encoder-decoder Transformer (Vaswani et al., 2017) which has validated and scalable architecture. As Alec Radford et all described, all audio is re-sampled to 16,000 Hz, and an 80-channel log- magnitude Mel spectrogram representation is computed on 25-millisecond windows with a stride of 10 milliseconds

We were interested to see that *Whisper models* has implemented an audio language detector, which was created by fine-tuning a prototype model trained on a prototype version of the dataset on VoxLingua107 (Valk & Aluma¨e, 2021) to determine that the spoken language matches the language of the transcript according to CLD2 (***Compact Language Detector 2).***

Whisper models family have multiple shapes depending on the size and the objective you want to use.

Table 1. Architecture details of the Whisper model family

Model	Layers	Width	Heads	Parameters
Tiny	4	384	6	39M
Base	6	512	8	74M
Small	12	768	12	244M
Medium	24	1024	16	769M
Large	32	1280	20	1550M

Source: Redford et al. Robust Speech Recognition via Large-Scale Weak Supervision.

There are multiple ways of improvement of Whisper models. By design, Whisper has a receptive field of 30-seconds. In order to transcribe audios files longer than this, one of two long-form algorithms are required:

- **Sequential**: uses a time window for buffered inference, transcribing 30-second slices one by one
- **Chunked**: splits long audio files into shorter ones (with a small overlap between segments), transcribes each segment independently, and stitches the resulting transcriptions at the boundaries

The sequential long-form algorithm should be used in the context of transcription when accuracy is the most important factor, and speed is less. The chunked long-form algorithm should be used when transcription speed is the most important factor or you are transcribing a single long audio file. By default, Transformers uses the sequential algorithm.

In the context of real time processing the transcription speed is very important that way the experiments went in this direction, more specifically using chunked algorithm.

More than that, we extend our research and identified Distil-Whisper models which is a more performant model who can perform good results in terms of accuracy and time results.

Basically, Sanchit Gandhi et al. use an alternative strategy, first proposed by Patry (2022), in which the long audio file is chunked into smaller pieces with a small overlap between nearby segments. The model is run over each chunk and then inferred text is joined finding the longest common sequence between overlaps. This approach enables accurate transcription across chunks without having to transcribe them sequentially.

This algorithm is semi auto-regressive which means that the chunks can be processed in any order, provided nearby chunks are subsequently joined correctly at their boundaries. This mechanism allows to process in parallel.

This allows chunks to be transcribed in parallel through batching. In practice, this yields up to 9 times improvements in inference speed compared to sequential transcription over long audio files. In this work, use the chunked long-form transcription algorithm when evaluating both the Whisper and Distil-Whisper models.

Methodology

In the experimental methodology section, we detail the techniques and methods used in this research, approaching ASR for small sentences of speech stream, long sentences with complex wording and, more specific, Romanian sentences for a multilanguage model.

Before we will start our research of the default Whisper model, it's important to understand the concept of evaluation metrics for ASR systems. For instance, the Word Error Rate (WER) is a common metric for assessing speech recognition systems. It measures the errors in predictions compared to target text transcriptions, categorized into substitutions (S), insertions (I), and deletions (D). WER is computed as:

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

Figure 1. WER definition

Source: H. Nanjo and T. Kawahara A new ASR evaluation measure and minimum Bayes-risk decoding for open-domain speech understanding.

where N is the total number of words in the sentence. A lower WER indicates fewer errors, with 0% being a perfect score. The WER is either calculated as a number or percentage.

Each use case has a separate approach and evaluation method.

Small sentences

In this use case we evaluate a subset of audio sample provided by librispeech_asr-noise data set. We have selected a short subset of data with less than 64 words. Considering the scope of the research and the subset size, we use only sequential approach and run the model into CPU environment.

One important step in evaluation is the results normalization. In many cases the transcript associated with the audio sample is not totally clean, considering lack of punctations (comma) or other grammatical orthography. For supervised models, the human transcripts have lack of punctations which can generate wrong evolution of the mode. We opt to address this problem with extensive standardization of text, before the WER calculation to minimize penalization of non-semantic differences.

Input: x is an audio sample
Parameters: N no of audio sample in dataset
Output: WER, duration
1. Set processor type = CPU
2. Select Whisper model
3. Load [dataset]
4. While n < N
5. Get [audio sample] from [dataset]
6. Result = pipeline [audio sample]
7. Result= Normalize [Results]
8. Calculate WER
9. Evaluate Duration, WER, STDEV[Duration]
10. n=n+1
11. End While
12. Return WER, duration, STDEV[Duration]

Figure 2. Small sentence procedure logic workflow
Source: Authors' own research results/contribution.

In order to understand the spread of the duration we calculate and evaluate Standard deviation of the duration for each Whisper algorithm. We noticed that Standard deviation is increasing with the in the same way with the complexity of the model (whisper-tiny -> whisper-large-v3).

$$W = \{\text{set of Whisper models}\}; \qquad i \in W$$

$$STDEV_i = \sqrt{\frac{\sum_j^N \left(\left(x_j - \mu\right)^2\right)}{N}}; \; N = nr\ of\ audio\ sample$$

Figure 3. Standard deviation
Source: Alexandru Isaic-Maniu, Constantin Mitrut, Vergil Voineagu Statistica.

Long sentences
In this case, we have identified a clean audio file in the native Britain English with 20 min+ duration in order to test the performance of the model in very well setup environment.
We have evaluated manually the transcript in order to assure accuracy of the results and fair comparison with different models outputs.
We choose to have 3 different scenarios:

- Running **Sequential method** on CPU technology
- Running **Chunked** on CPU technology
- Running **Chunked** on GPU technology (limited resource)

Input: x is an audio sample
Parameters: local file (path)
Output: WER, duration
1. Set processor type = CPU or GPU
2. Select Whisper model
3. Load local file
4. Get [audio sample] from [local file]
5. Load [audio transcript]

```
6. Set up parameters for
        Chunked <- 25 or 30
        Batch size <- 8
7. Result =  pipeline [audio sample
8. Calculate WER
9. Evaluate Duration, Performance KPI, WER
10. Return WER, duration, performance KPI
```

Figure 4. Long sentence procedure logic workflow

Source: Authors' own research results/contribution.

Romanian sentences

This is an extended scenarios where we have tested the performance of multi-language Whisper models for Romanian language. Therefore, our aim was to see if these models can be used, or it is need to fine-tuned the models with full Romanian data sets.

We have used a dedicated data set produced by Research Institute for Artificial Intelligence "Mihai Drăgănescu".

```
Input: x is an audio sample
Parameters: N no of Romanian audio sample in dataset
Output: WER, duration
1. Set processor type = CPU
2. Select Whisper model
3. Load [dataset]
4. While n < N
5.      Get [audio sample] from [dataset]
6.      Result =  pipeline [audio sample]
7.      Result= Normalize [Results]
8.      Calculate WER
9.      Evaluate Duration, WER
10.     n=n+1
11. End While
12. Return WER, duration
```

Figure 5. Romanian sentence procedure logic workflow

Source: Authors' own research results/contribution.

We evaluated the result from accuracy perspective. As we did for small English sentences, we used normalization function to avoid wrong calculation of the WER due to missing punctations in data set.

Dataset Usage

We have used sample open-source data for small audio files (< 30s) provided by the hugging face dateset distil-whisper/librispeech_asr-noise, test-pub-noise, split 40.

Complementary, we also use large audio file to understand the reliability of the model even for batching stream audio using small parts but also for large file to see how long the process can take and evaluate the method can be approach in the real time streaming flow process. The large audio is about 20:45 min.

For Romanian transcript we have used an open data set named USPDATRO.

Underrepresented Speech Dataset from Open Data: Case Study on the Romanian Language (USPDATRO) is a manually created Romanian language speech corpus.
It was created specifically using speech types that are underrepresented in other speech datasets.

Sources for this dataset are represented by open data available on multimedia platforms under a Creative Commons license. The data was manually transcribed and aligned at segment level. In addition to the text and audio files, we offer text annotations (lemmatization, part of speech tags, dependency parsing) in CoNLL-U Plus format.

Each datasource is mentioned by URL in the metadata.csv file with associated license (a Creative Commons variant).

The dataset is produced under Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy Web.

Results and discussions

The experiments are divided in 3 different use cases and running models for:

- Small sentences, between 8 – 64 words with average about 24-25 words
- Long sentence, for testing performance
- Romanian sentences, to see the results for Romanian language for models trained for multi-languages test cases

Small sentences

The experiments have been performed for a large spectrum of sentences to understand the behavior for different scenarios. For the dataset we have been used, we measured the duration (seconds) of running Whisper models counting the number of words provided by the sentences. The reason was to understand performance of the models using small sentences. We noticed that there is a flat cost of execution regardless of the length of sentences, so the duration from 8 – 64 words in the sentences has a small variation against average duration.

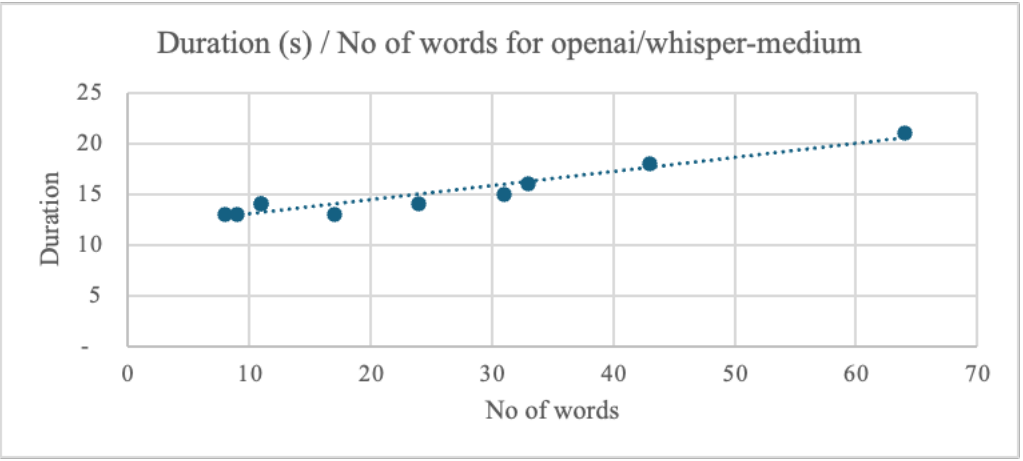

Figure 6. Duration variation against no of words for Whisper medium model

Source: Authors' own research results/contribution.

As a complete picture, the experiments have covered most of the models and we concluded that the whisper-tiny has the fastest time, but the WER is higher. The longest time has been performed by the openai/whisper-large-v3 but it provides the best WER.

Table 2. Small sentences Duration and WER results

Model	Duration (s)	WER	Average of words
openai/whisper-tiny	8.20	0.0385	25
openai/whisper-small	10.10	0.0097	25
openai/whisper-medium	15.10	0.0074	25
openai/whisper-large-v3	32.70	0.0032	25
openai/whisper-large-v3-turbo	19.50	0.0032	25
distil-whisper/distil-large-v2	14.20	0.0129	25
distil-whisper/distil-large-v3	13.40	0.0074	25

Source: Authors' own research results/contribution.

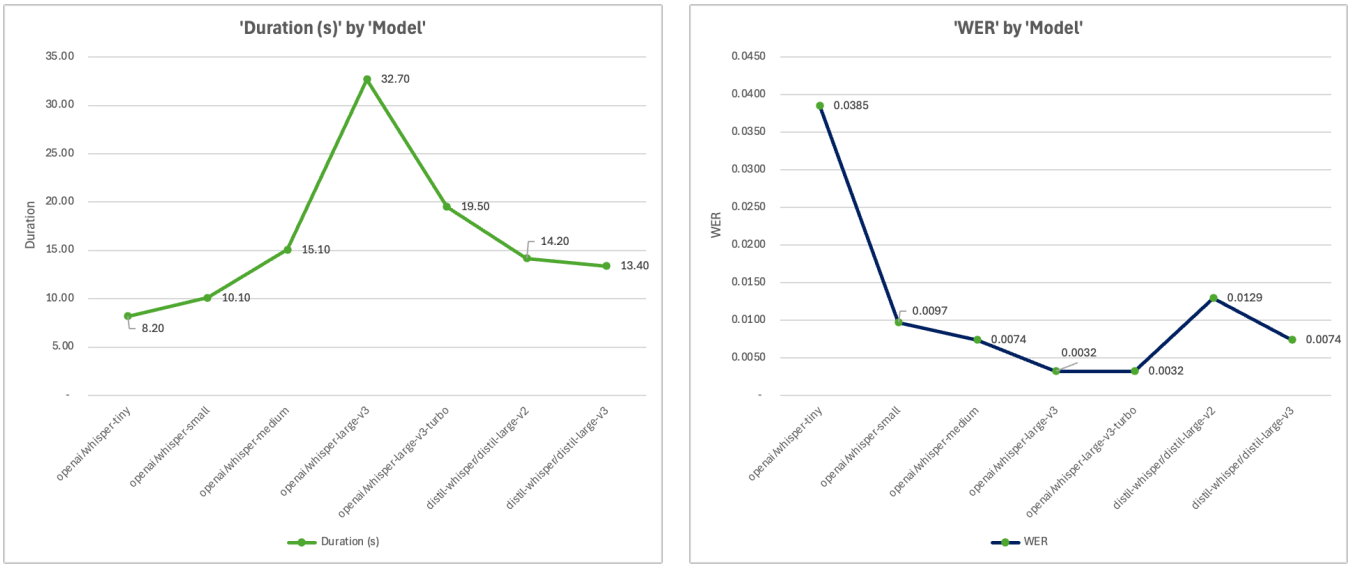

Figure 7. Duration / WER by model

Source: Authors' own research results/contribution.

The experiment measures the duration (execution time of Whisper models) and WER for small sentences data sets. We noticed that the smallest Whisper models provide rapid execution but with a penalty for WER. The larger Whisper models provide excellent WER results but with a longer duration.

If the user is looking for low latency, as the real time data processing is requesting, and accepts a higher WER, the whisper-tiny or whisper-small can be a solution. If the higher WER is not accepted but duration is still a constrain, whisper-large-v3-turbo is most suitable model.

Both models distil-whisper/distil-large-v3 and distil-whisper/distil-large-v2 perform good results (duration(s) and WER) and can be taken in consideration for real time data processing.

Long sentences

The experiments have been performed for a specific data set with large audio file in English language having native accent. We have used two long-form algorithms, sequential and chunked, as we described in the previous chapter.

Table 3. Long sentences Duration, Execution performance and WER results Sequential form algorithm

Sequential			
Model (M)	**Duration (s)**	**Execution performance (x)**	**WER**
openai/whisper-tiny	37	15.20	0.0448
openai/whisper-medium	407	1.40	0.0146
openai/whisper-large	1,463	0.40	0.0092
openai/whisper-large-v2	1,892	0.30	0.0178
openai/whisper-large-v3	561	1.00	-
openai/whisper-large-v3-turbo	208	2.70	0.0103
distil-whisper/distil-large-v2	212	2.60	0.0751
distil-whisper/distil-large-v3	175	3.20	0.0427

Source: Authors' own research results/contribution.

$$\text{Execution performance (x)}_{\text{Model M}} = \frac{\text{Duration (s)}_{\text{whisper-large-v3}}}{\text{Duration (s)}_{\text{Model M}}}$$

Figure 8. Execution formula

Source: Authors' own research results/contribution.

Figure 9. Execution performance (x) / WER by model – Sequential form

Source: Authors' own research results/contribution.

The experiment measures the duration (execution time of Whisper models) and WER for long sentences data sets. We noticed that the smallest Whisper models provide rapid execution with acceptable WER. The larger Whisper models provide excellent WER results but with a good execution time. Distillated Whisper models have issues with providing an acceptable WER.

For faster execution with an acceptable WER, closed to 4.5%, whisper-tiny is 15 x faster than whisper-large-v3. If the researcher is looking for faster execution whisper-tiny can be a possible option and the result as most than acceptable. Considering the processing of long audio

file, where mostly the objective is to process in batch approach, whisper-large-v3-turbo is the most suitable algorithm for this.

For chunked form we have tested with different parameters of execution, but is highly important to understand that these are dependent by the computer power of the system (no of cores, no of GPUs).

In our research output we have used the chunk_length_s=25 and batch size=8. Considering the limitation of the environment from GPU perspective, we didn't performed measures for different chunk length and batch size.

Table 4. Long sentences Duration, Execution performance and WER results Chunked form algorithm

Chuncked			
Model	**Duration (s)**	**Execution performance(x)**	**WER**
openai/whisper-tiny	34	33.65	0.1026
openai/whisper-medium	447	2.56	0.0805
openai/whisper-large	3,927	0.29	0.0686
openai/whisper-large-v2	3,559	0.32	0.0627
openai/whisper-large-v3	1,144	1.00	
openai/whisper-large-v3-turbo	408	2.80	0.0864
distil-whisper/distil-large-v2	205	5.58	0.0746
distil-whisper/distil-large-v3	208	5.50	0.0648

Source: Authors' own research results/contribution.

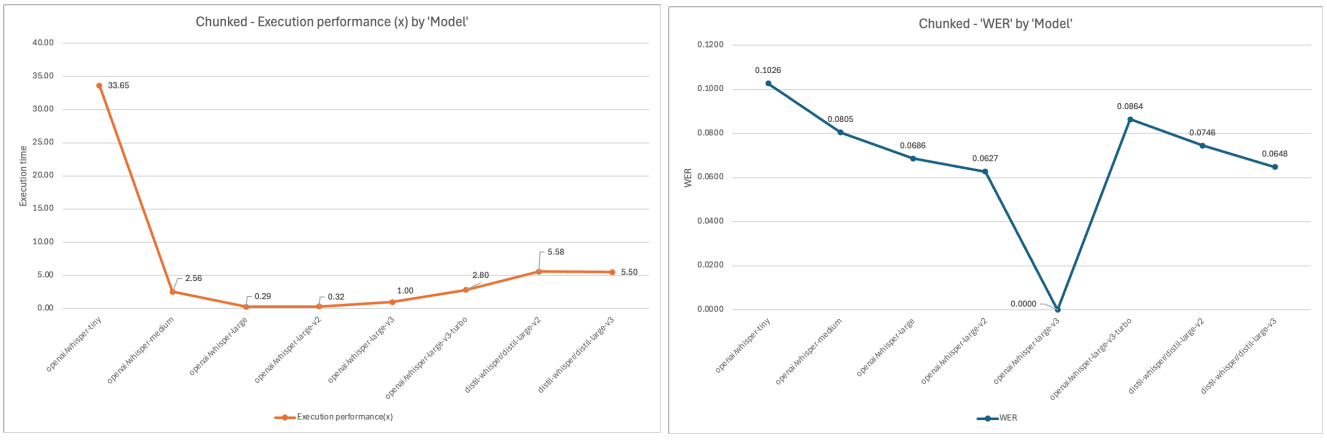

Figure 10. Execution performance (x) / WER by model – Chunked form

Source: Authors' own research results/contribution.

For Chunked form, for faster execution whisper-tiny is 33 x faster, but WER is closed to 10%. Whisper-large-v3-turbo provide faster execution than the whisper-large-v3, but WER is higher. Depending on the use case, distil-whisper-large-v3 provide faster execution, 5.5 x and lower WER.

Additional performance testing has been made, comparing execution on CPU running sequential form and GPU running in chunked form.

The results are presented below:

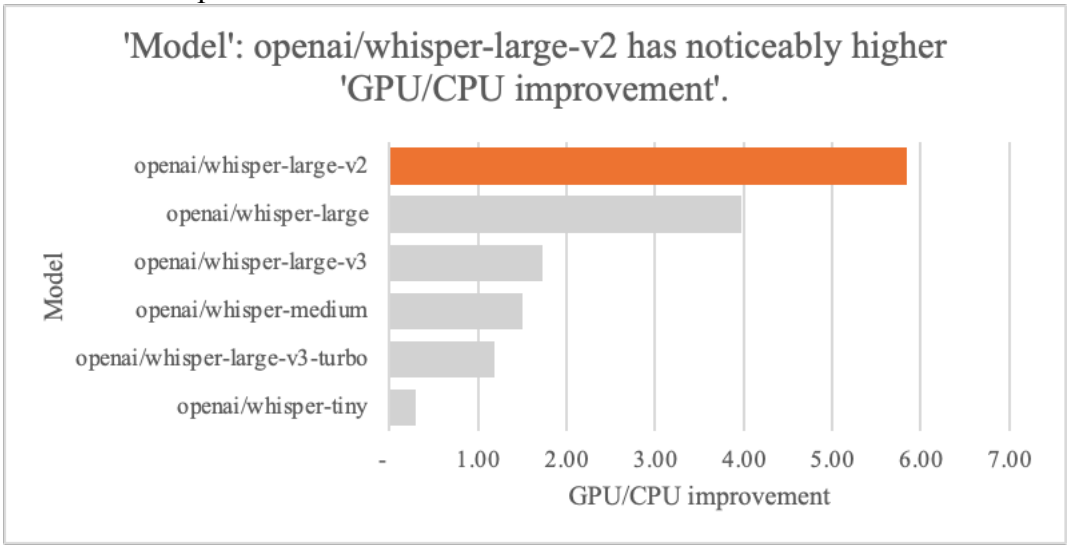

Figure 11. GPU/ CPU benchmark execution performance

Source: Authors' own research results/contribution.

It is noticeable remark that whisper-large-v2 perform a high improvement on GPU hardware. On limited GPU resources, as we used in our experiments, whisper-large-v3 offers performance improvement closed to 2x.

Romanian sentences

Whisper models offer multi-language speech recognition cover multiple language. We have tested for specific set of Romanian sentences and we obtain very good results. For instance, for sentences about 25 words, whisper-large-v3 and whisper-large-v3-turbo provide most important results, whisper-large-v3-turbo being 2 x faster than whisper-large-v3 and acceptable WER (between 3-11%). Whisper-tinny is faster but results are not suitable for enterprise speech recognition systems, offering a 55% WER results.

Conclusion

For real time data processing system, whisper models can be used in multiple forms depending on the requirements and acceptable criteria and performing real time data flows. Therefore, for low latency flows or small sentences, whisper-tiny or whisper-large-v3-turbo can be used, complemented by distil-whisper models (v2 or v3), performing faster results and acceptable WER.

For long sentences, suitable for batch processing, whisper-large-v3-turbo and distil-whisper models can be used with very good results as we described in the previous section. One important remark is about GPU processing which can provide an important benefit for real time data processing in terms of performance corelated with chucked form of the model.

The recommendation is to align the GPU resources (no of cores, memory) used with the parameters of the whisper model to get maxim benefits.

For Romanian sentences, for small sentences, real time speech to text data flow, whisper models can pe approached with acceptable results. However, for enterprise use cases, it is recommended to have a fine-tunned model, trained for specific Romanian data sets, having the specific of the technical words you are going to use.

Limitation of the current research is related with the GPU resources available and the experiments parameters (chunk and batch size) we have used for measuring the outputs. This can be extended in the future research with understanding the behavior of Whisper models using more GPU resources and analyze different values for chunk and batch size. The future research is planning to cover the analysis of model dedicated fine-tuned for specific language (more specific for Romanian language) and compare with the results of multi-language models.

References

Kang. G Shin and Parameswaran Ramanathan. (2020). Real-Time Computing: A New Discipline of Computer Science and Engineering, Proceedings. IEEE, vol. 82, no. 1, pp. 6-24, 1994.

Alec Radford, Jong Wook Kim, Tao Xu , Greg Brockman, Christine McLeavey, Ilya Sutskever. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. arXiv:2212.04356

Likhomanenko, T., Xu, Q., Pratap, V., Tomasello, P., Kahn, J., Avidov, G., Collobert, R., and Synnaeve, G. (2020). Rethinking evaluation in asr: Are our models robust enough? arXiv preprint arXiv:2010.11745

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. arXiv:2006.11477.

Chan, W., Park, D., Lee, C., Zhang, Y., Le, Q., and Norouzi, M. SpeechStew. (2021). Simply mix all available speech recognition data to train one large neural network. arXiv preprint arXiv:2104.02133.

Zhang, Y., Park, D. S., Han, W., Qin, J., Gulati, A., Shor, J., Jansen, A., Xu, Y., Huang, Y., Wang, S., et al. (2021). BigSSL: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition. arXiv:2109.13226.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems, pp. 5998–6008.

Valk, J. and Alumae¨, T. (2021) Voxlingua107: a dataset for spoken language recognition. In 2021 IEEE Spoken Language Technology Workshop (SLT), pp. 652–658. IEEE.

Sanchit Gandhi, Patrick von Platen & Alexander M. Rush. (2017). Distil-Whisper: Robust knowledge distillation via large-scale pseudo labelling, arXiv:2311.00430

Nicolas Patry. (2022) Making automatic speech recognition work on large files with Wav2Vec2 in Transformers. https://huggingface.co/blog/asr-chunking. Accessed: 25 Nov.,

H. Nanjo and T. Kawahara (2005) A new ASR evaluation measure and minimum Bayes-risk decoding for open-domain speech understanding 2024.

Hendrycks, D., Liu, X., Wallace, E., Dziedzic, A., Krishnan, R., and Song, D. (2020). Pretrained transformers improve out-of-distribution robustness. arXiv preprint arXiv:2004.06100.

Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy Web, Romanian datasets, http://www.racai.ro