

Contents

Text Quality	1
Benchmarked Models	1
Long Form English	2
Fleurs.....	2
Latin Script Languages.....	2
Non-Latin Script Languages.....	3
Performance.....	3
Memory Usage, Presets	3
Memory Usage, Custom.....	4
Transcribe Speed	4

Text Quality

The tables in this section list [word error rate](#) metrics for open speech datasets I found on the internets.

Despite Wikipedia presenting a tidy formula that implies the figure is precise, computing error rates is tricky in practice because natural languages offer many correct ways to express the same speech. Take the English example of “I have spent \$1,500” versus “I’ve spent fifteen hundred dollars”: both are perfectly valid, yet reliably normalising text at scale to avoid counting these linguistic shenanigans as errors is quite difficult.

Rather than comparing directly against published figures, I ran the reference speech-to-text implementation on these datasets myself. This keeps error rates consistent across versions, since the same methodology is applied throughout. The reference implementation is written in Python, using Transformers, PyTorch, and the CUDA runtime.

Benchmarked Models

Key	Model	Preset
Python	Voxtral Mini 4B Realtime 2602, 8.25 GB	Reference
High	Voxtral Mini FP16, 8.5 GB	High
Medium	Voxtral Mini BCML text, 4.6 GB	Medium
Low	Voxtral Mini BCML, 3.4 GB	Low

The tables also include **Mistral** rows with the figures [published on Hugging Face](#) by Mistral AI. Note that Mistral’s numbers use a different text normalisation algorithm, as well as a different subset of the FLEURS dataset.

The discrepancies between my measurements and Mistral's published figures do not necessarily indicate bugs in my QA pipeline.

Long Form English

For this benchmark, I extracted all 11 audio files from the test-00000-of-00001-7a1bb92f62e929b8.parquet archive in the [tedlium-long-form dataset](#), then transcribed them to UTF-8 text files using both the transcribe application and the reference Python implementation. Error rates were then computed using the EnglishTextNormalizer class for text normalisation.

All transcriptions used a 2,400 ms delay (best quality, and the only option the software supports for file transcription). Here are the measured results.

Model	Python	High	Medium	Low
Error rate	2.81%	2.82%	2.85%	2.83%

Fleurs

[Fleurs](#) is the speech version of the FLoRes machine translation benchmark. I used the **test** subset downloaded from [Hugging Face](#).

Latin Script Languages

English was normalised using the EnglishTextNormalizer Python class from the [whisper normalizer package](#); all other languages were normalised with BasicTextNormalizer from the same package.

Delay	Model	German	English	Spanish	French	Italian	Dutch	Portuguese
2400 ms	<i>Mistral</i>	4.15%	4.05%	2.71%	5.23%	2.37%	5.91%	3.93%
	Python	4.11%	4.07%	2.67%	5.20%	2.25%	5.57%	3.86%
	High	4.08%	4.09%	2.66%	5.27%	2.18%	5.59%	3.85%
	Medium	4.18%	4.14%	2.72%	5.37%	2.25%	5.87%	3.97%
	Low	4.22%	4.16%	2.72%	5.56%	2.31%	6.03%	3.99%
960 ms	<i>Mistral</i>	4.87%	4.34%	2.98%	5.68%	2.46%	6.76%	4.57%
	Python	4.81%	4.53%	2.78%	5.74%	2.58%	6.57%	4.27%
	High	4.74%	4.53%	2.79%	5.74%	2.54%	6.57%	4.23%
	Medium	4.71%	4.50%	2.82%	6.00%	2.58%	7.17%	4.27%
	Low	4.92%	4.52%	2.82%	6.14%	2.54%	7.26%	4.27%
480 ms	<i>Mistral</i>	6.19%	4.90%	3.31%	6.42%	3.27%	7.07%	5.03%
	Python	5.50%	5.08%	3.25%	6.87%	3.44%	7.17%	4.96%
	High	5.42%	5.08%	3.26%	6.84%	3.36%	7.16%	4.96%
	Medium	5.72%	4.95%	3.28%	7.09%	3.47%	7.46%	5.24%
	Low	5.83%	5.05%	3.29%	7.14%	3.45%	7.62%	5.25%

Non-Latin Script Languages

Text was normalised using different methods depending on the language. See [ErrorRateMethod.pdf](#) for technical details.

Delay	Model	Arabic	Hindi	Chinese	Japanese	Korean
2400 ms	<i>Mistral</i>	14.71%	10.73%	8.48%	5.50%	14.30%
	Python	10.46%	13.37%	5.65%	5.33%	8.10%
	High	10.70%	13.35%	5.67%	5.30%	8.12%
	Medium	10.84%	13.40%	5.92%	5.40%	8.40%
	Low	10.47%	13.52%	5.92%	5.47%	8.29%
960 ms	<i>Mistral</i>	20.32%	11.82%	8.99%	6.80%	14.90%
	Python	10.42%	14.41%	6.17%	5.65%	8.21%
	High	10.44%	14.41%	6.16%	5.75%	8.22%
	Medium	11.05%	14.52%	6.46%	5.53%	8.29%
	Low	10.71%	14.84%	6.39%	5.82%	8.40%
480 ms	<i>Mistral</i>	22.53%	12.88%	10.45%	9.59%	15.74%
	Python	11.29%	15.11%	7.03%	6.36%	8.63%
	High	11.25%	14.88%	7.01%	6.40%	8.56%
	Medium	11.54%	15.51%	7.25%	6.35%	8.51%
	Low	11.48%	15.72%	7.38%	6.41%	8.56%

Performance

Memory Usage, Presets

The table below lists memory usage of the transcription engine with Voxtral models when selecting a preset in the engine options. The figures reflect memory used by the speech-to-text engine measured in isolation; GUI frontends will consume additional memory on top.

Model	Preset	VRAM	RAM
Voxtral BCML	low	3.3 GB	2.3 MB
	medium	3.4 GB	771 MB ¹
	high	4.3 GB	2.3 MB
Voxtral BCML text	low	4.5 GB	2.3 MB
	medium	4.6 GB	771 MB
	high	5.6 GB	2.3 MB
Voxtral FP16	low	8.4 GB	2.3 MB
	medium	8.5 GB	771 MB
	high	9.4 GB	2.3 MB

¹ The system RAM figure for the medium preset is not a bug. That preset loads one large but sparsely accessed matrix into system memory in order to conserve the precious VRAM.

Memory Usage, Custom

The table below lists memory usage of the transcription engine with Voxtral models after selecting individual options on the first screen.

Model	Embeds	Audio cache	Language cache	VRAM	RAM
Voxtral BCML	Balanced	Quantised	Quantised	3.3 GB	771 MB
	Performance	Uncompressed	Quantised	3.4 GB	2.3 MB
	Balanced	Quantised	Uncompressed	3.7 GB	771 MB
	Performance	Quantised	Uncompressed	3.7 GB	2.3 MB
	Balanced	Uncompressed	Uncompressed	3.8 GB	771 MB
	Performance	Uncompressed	Uncompressed	3.8 GB	2.3 MB
	Quality	Quantised	Quantised	3.8 GB	2.3 MB
	Quality	Uncompressed	Quantised	3.9 GB	2.3 MB
	Quality	Quantised	Uncompressed	4.2 GB	2.3 MB
Voxtral BCML text	Balanced	Quantised	Quantised	4.5 GB	771 MB
	Performance	Uncompressed	Quantised	4.6 GB	2.3 MB
	Balanced	Quantised	Uncompressed	4.9 GB	771 MB
	Performance	Quantised	Uncompressed	4.9 GB	2.3 MB
	Balanced	Uncompressed	Uncompressed	5.0 GB	771 MB
	Performance	Uncompressed	Uncompressed	5.0 GB	2.3 MB
	Quality	Quantised	Quantised	5.0 GB	2.3 MB
	Quality	Uncompressed	Quantised	5.2 GB	2.3 MB
	Quality	Quantised	Uncompressed	5.4 GB	2.3 MB
Voxtral FP16	Balanced	Quantised	Quantised	8.4 GB	771 MB
	Performance	Uncompressed	Quantised	8.5 GB	2.3 MB
	Balanced	Quantised	Uncompressed	8.8 GB	771 MB
	Performance	Quantised	Uncompressed	8.8 GB	2.3 MB
	Balanced	Uncompressed	Uncompressed	8.9 GB	771 MB
	Performance	Uncompressed	Uncompressed	8.9 GB	2.3 MB
	Quality	Quantised	Quantised	8.9 GB	2.3 MB
	Quality	Uncompressed	Quantised	9.0 GB	2.3 MB
	Quality	Quantised	Uncompressed	9.3 GB	2.3 MB

Transcribe Speed

The table below lists transcribe rates relative to real-time across several GPUs I have benchmarked. Specifically, I transcribed an audio file of 3m:19s on multiple computers and divided the audio duration by the time taken to transcribe it, excluding the time it took to load the model from disk. A value of 1.0 means real-time transcription speed; higher is better.

For reliable use with dictation software, the transcription rate should exceed 1.0. It is difficult to test thoroughly given the limited number of GPUs available to me, but in my experience a rate of 1.15 is sufficient for real-time use.

Unlike the memory figures, throughput depends on many factors that are difficult to control. For integrated GPUs, the most important variable is system memory configuration: DDR version, speed, and channel count. For all GPUs, throughput is also affected by resource-intensive applications running alongside the speech-to-text engine, GPU driver version, operating system version, and other factors such as ambient temperature.

Take these numbers with a grain of salt, and make sure to evaluate the software on your particular machine before purchasing. I provide 30 days evaluation licences partly for that reason.

The Dictation application has a real-time indicator labelled "GPU load", which displays the inverse of this value expressed as a percentage. If the figure stays below 90% even whilst speaking quickly, the GPU is capable enough for your chosen combination of model and settings. Note that if you are using a laptop with two GPUs, you may need to select the faster discrete GPU on the application's first screen.

I have measured the numbers using the software version 0.9.6.

GPU	Model	Preset	Rate
Ryzen 5 5600U iGPU²	Voxtral BCML	low	1.36
	Voxtral BCML text	medium	1.09
	Voxtral FP16	high	0.81
Ryzen 7 8845HS iGPU³	Voxtral BCML	low	3.21
	Voxtral BCML text	medium	3.28
	Voxtral FP16	high	1.55
Ryzen 7 8700G iGPU	Voxtral BCML	low	3.58
	Voxtral BCML text	medium	3.57
	Voxtral FP16	high	1.54
GeForce 1650	Voxtral BCML	low	2.88
	Voxtral BCML text	medium	1.39
	Voxtral FP16	high	OoM ⁴
GeForce 1080 Ti	Voxtral BCML	low	6.67
	Voxtral BCML text	medium	6.6
	Voxtral FP16	high	5.63
GeForce 4070 Ti Super	Voxtral BCML	low	10.9
	Voxtral BCML text	medium	10.58
	Voxtral FP16	high	9.48

² The laptop was running on AC power during the benchmark. It is a 2021 machine and the battery has degraded noticeably over the years.

³ The laptop was running on battery power during the benchmark.

⁴ "Out of Memory": an error message which says "Insufficient memory to continue the execution of the program." That GPU has only 4 GB of VRAM, so the failure with the uncompressed model was not unexpected.