Introduction

Speech-to-text (STT) is a technology that enables the translation of spoken language into text. It is a cornerstone of voice-based applications, which usually receive voice commands. Each application must recognize audio and transcribe it to text to perform actions or communicate back. There is a large number of use cases for this technology, ranging from video transcription to conversations with AI.

Virtual assistants are created to communicate with clients directly and possibly replace humans in that role completely, providing a matching user experience. To achieve this goal, the agent often needs to be able to react to the request in real time to match human performance, thus low latency and high accuracy is incredibly important to ensure the satisfactory user experience. Both inaccurate transcription that leads to incorrect actions and high latency reduces user's satisfaction with a product.

Brief introduction of Whisper and Deepgram. Whisper is a powerful open source transformer-based model for Automatic Speech Recognition. Deepgram is a cloud-based voice AI platform that provides APIs for various tasks in speech processing, among which is STT.

Model overview

Whisper

- Whisper is an open-source, multilingual transformer-based Automatic Speech Recognition (ASR) model developed by OpenAI.
- Supports over 100 languages
- Trained on the enormous amount of data (million of hours) of different quality making it robust to noise, accents, jargon and other background sound effects
- Customizable model with a lot of available pre-trained models
- SoTA model for ASR, scoring top places at ASR leaderboards[HF ASR Leaderboard]

Deepgram

- Deepgram Nova-2 is a state-of-the-art ASR model developed by Deepgram, designed to provide real-time transcription services with high accuracy and low latency.
- Provides real-time transcription with high accuracy
- Various features for STT(diarization, formatting, etc)
- Provides SDKs for several languages, comprehensive and extensive documentation
- Cloud-based platform
- Community of developers

Comparison

It makes sense to compare two models: Whisper and Nova-2, which is the closest competitor by Deepgram.

Accuracy

In order to evaluate the models in different environments, I have gathered and classified several popular benchmarks for ASR. Whisper large-v3 is a popular well-studied model. Deepgram Nova-2 is a model developed by a private company, so community has no access to the source code, which limits evaluation abilities. Thus, there are lots of sources where Whisper and its versions are evaluated on different datasets, but it is challenging to find the data on Nova-2. A large source for Nova-2 is Deepgram's website, but I feel

sceptical about it due to unclear statements about the data/how the model was trained[7], etc. Thus all the data is gathered from third parties.

The datasets are grouped in clusters based on the audio quality, where green datasets contain a high-quality data with single speaker, it is distinctive and noise-free.As we go down the data losses quality. This data is more close to real-life audio. Background noise, overlapping voices, sound effects, laughter from the audience and other not relevant information is present in the tracks. In terms of the task your company is trying to solve, the scores on those more complex datasets represent how well the model is able to deal with clients under different conditions. There are also two more rows included in the evaluation, which are “Noise” and “Telephony”. That data is gathered by AssembleAI and is not publicly available, however it is stated that it represents specifically chosen noisy data and telephone conversation, that looks looks similar to what you are doing.

The results are reported for English language only, but they do not differ much for other languages(except for some rare ones for Whisper turbo; you can check them out on the Whisper’s github).

Dataset Desription		
Dataset	**Quality**	**Description**
LibriSpeech	High	Very clear, audiobooks, professional narration.
TED-LIUM	High	Clear TED talks in controlled settings.
SPGISpeech	High	Clear navigation instructions, studio recorded.
FLEURS	High	Professional studio recordings, multiple languages.
Rev16	High	Clean base audio with controlled reverb conditions.
VoxPopuli	Moderate	Variable parliament acoustics, background noise.
GigaSpeech	Variable	Mixed sources from YouTube, podcasts, audiobooks.
Earnings-22	Variable	Conference calls, overlapping speech, background noise.
Meanwhile	Variable	TV shows with background effects and overlapping speech.
Common Voice	Variable	Crowdsourced recordings with varying quality.
AMI	Moderate-Low	Meeting recordings with overlaps and room noise.

Each color represents the source, where the data has been taken from for WER comparison.

	HuggingFace	https://huggingface.co/spaces/hf-audio/open_asr_leaderboard
	Whisper's github	https://github.com/openai/whisper
	artificialanalysis	https://artificialanalysis.ai/speech-to-text/models/deepgram
	AssemblyAI	https://www.assemblyai.com/benchmarks
	Other	https://github.com/Wordcab/rtasr
	No Data	

WER			
Datsaset/Model	Whisper Large-v3	Whisper Large-v3 turbo	Deepgram Nova-2
LibriSpeech	2,01	2,10	3,13
TED-LIUM	3,86	3,57	8,98
SPGISpeech	2,94	2,97	N/A
FLEURS	N/A	N/A	15,5
VoxPopuli	9,54	11,87	N/A
GigaSpeech	10,02	10,14	N/A
Earnings-22	11,29	11,63	11,05
Meanwhile	9,75	5,56	N/A
Common Voice	9,30	10,60	15,10
AMI	15,95	16,13	N/A
Noisy	11,86	N/A	16,03
Telephony (internal)	12,55	N/A	13,22

The results of WER are gathered in this table. Deepgram's website reports different metrics for WER on its own data, which is not public, so I ignored it.
From the table above we can see that there is a rather minor difference between Whisper large-v3 and Whisper Turbo. Whisper Turbo is a very fresh model that has been published in October 2024, so some results are not available for this model.

Whisper models work well for clear data achieving a very low WER, the performance decreases when it comes to noisy data.The worst results are on AMI data, which is an extremely challenging dataset. The results for other datasets with Medium-quality audio is quite good with Whisper scoring approximately 10% WER on most of them and achieving exceptional results on Meanwhile(Late night show with laughter and background noise, recorded in studio though). Common Voice is another very important dataset which has become a typical benchmark for ASR models since it contains a lot of noise, jargones and accents and is regarded as one of major ASR models benchmarks.

Summarizing, Whisper model shows robust results under challenging conditions in a presence of noise and other sound effects.

Nova-2 has no problem with clear speech, however the performance significantly decreases when it comes to noisy real-life data. The results for Nova-2 are tolerable, but it is clear that it struggles with noise and this needs to be taken into account (processing confidence scores, for instance).

Latency

Deepgram Nova-2 is a model capable of real-time transcription with minimal latency.

Whisper processes data in chunks of 30s, so the model needs to be modified to process audio streaming(fortunately, it can be achieved with transformer library, which makes the process easier). The processing time depends on several factors, among which

the choice between CPU/GPU processing and GPU type. For a top A100 GPU with 80GB it is reported to achieve 1,5s latency for Whisper large-v3 and ~0.46 for Whisper turbo for one audio chunk [4].

However, A100 is a very expensive GPU, and the performance is going to degrade upon the usage of less powerful GPU. Whisper requires a lot of VRAM (10GB for large-v3 and 6GB for turbo), which makes it quite expensive to use. Moreover,the usage of less powerful GPUs might result in significant degradation in speed, which might make the usage of whisper for real-time transcription not possible.

On the other hand, Nova-2 does not suffer from those problems, as it is a cloud based model, where you pay as you use the resources and do not have to worry about anything.

Ease of integration

Whisper might require some tailoring to correctly and efficiently integrate it into the pipeline. It is not guaranteed that it would perform very well on the custom data out of the box, so it might need some fine-tuning or choosing the right whisper model (there are several open-source options for Whisper, apart from official ones). All the data processing steps should also be accounted for including preprocessing and postprocessing. Apart from that, the model is pretty easy to use in the scripts, as there are convenient inference scripts on HuggingFace/GitHub, along with fine-tuning approaches and strategies. There are lots of pre-trained models available on github and huggingface, that can be easily installed and used.

Local deployment requires several steps to do like preparing the environment, setting up the model, containerization and creating endpoint. There are some tools to ease the process, but it still requires time and effort of the engineers. After the deployment, the model needs to be thoroughly monitored, maintained and optimized.

Deepgram does not need any deployment, as the audio could be processed on the run via API calls.Deepgram offers SDKs in several language, along with a comprehensive documentation and guides, that makes its usage easy. One does not need to process data or worry about maintenance in this case, so the integration is much easier.

Resource Requirements

The resource requirements for Whisper depend on the speed needed to process the audio. There is no limit for CPU, so the inference time is the only thing that would be affected by less powerful CPU(of course, there should be enough RAM, ideally >8GB). The GPU needs to have CUDA support and the situation here is the same as with CPU. Powerful GPUs even enable real-time audio processing.

Whisper large-v3 requires ~10GB of VRAM, Whisper turbo - ~6GB. Here is a small table to compare the performance on different GPUs/CPUs for Whisper.

Computing Device	**large-v3**	**Distil large-v3**	**large-v3-turbo**
CPU: Ryzen 6850U	00:26:12	00:13:30	00:18:30
CPU: Apple M1	00:33:15	00:21:40	00:??:??

CPU: Intel i9-10940X	00:10:25	00:04:36	00:??:??
CPU: Intel i7-8750H	00:??:??	00:??:??	00:19:16
GPU: RTX 2080 Ti	00:01:44	00:01:06	00:??:??
GPU: RTX 2070 Max-Q	00:05:59	00:??:??	00:04:37

The biggest models provides the highest accuracy, so it makes sense to use Whisper large for offline processing. Regarding the speech-to-text streaming, Whisper turbo is the best model in terms of accuracy/inference time tradeoff.

Nova-2 is a cloud-based model, thus it does not need any resources allocated. So, the only thing that client does is sending/receiving/displaying data. The only limit for this technology is internet speed. However, some users reported several issues, when the latency suddenly increase to 20-30s per 10s audio or the model gets stuck at some audio part. In this case the client has no control over the network, and the only thing to do is to contact the support or wait for the company to resolve the issue.

Cost and Scalability

Whisper requires investment in local deployment environment, which includes CPUs, GPUs, storage, electricity cost, the cost to develop the infrsutructure, maintenance cost. The scalability for this model requires possible modification of the infrastructure and more investments in hardware.

Deepgram proposes paid plans for its models, when you pay for as much you used. The platform provides an easily scalable solution. Of course, the costs might skyrocket with the drastic increase of data for processing.

Both approaches need to be carefully evaluated based on the current data load, current infrustucture design and expected growth.

Use case in Ale

There are two types of audio processing in Ale's pipeline(assumption):

- offline transcription
    - audio analysis (for example to identify issues with real-time transcriber by processing calls)
    - transcribing meetings/ video calls
- real-time transcription
    - responding to voice commands/ performing the actions upon voice commands, etc

Whisper is a large and accurate model that requires proper GPU setup to run in real-time mode. It is also free and customizable, so could be tailored to any task. It seems like an obvious choice for offline processing. On the other hand, Deepgram proposes a real-time transcription without the need to build the environment for a specific model.
As Ale is a early-stage startup, I would like to propose a mixed approach. As it has been already discussed, there are way to make Whisper work with real-time transcription.

However, it requires significant investment into the infrastructure and maintenance. It comes down to the following pros and cons:

Whisper:

Pros:

- free
- customizable
- no need to send the data to third-party company
- accurate and powerful (scores top results among competitors on all the benchmarks)
- absolute control over the model

Cons:

- initial investment into the infrastructure
- cost of scalability and maintenance
- cost of tailoring and fine-tuning
- requires powerful GPUs to work in real-time

Deepgram:

Pros:

- less accurate than Whisper large model, but still performs well on all the benchmarks
- provides real-time transcription
- easy to integrate, scale and no need to maintain
- a quick, easy and powerful solution
- a lot of features available
- supports custom model training[8]

Cons:

- becomes too expensive as the amount of data increases
- no absolute control over the model, so bugfixing is transferred to Deepgram, which might take time[3]
- Less accurate in noisy environments, so not as robust as Whisper

Recommendation:

Deepgram is more suitable for a quick market entry with minimal infrastructure investment. It provides quite high accuracy and real-time processing of data. Moreover, Deepgram provides access to other features, different models, well-written and easy to use documentation. Those are nice-to-have features, that might not be the main focus at this stage.

Whisper is a long-term solution. It is a more powerful model with more control, which is also open-source. It is more robust in noisy environments and generally more accurate. There are also various Whisper versions developed by community, that are worth evaluation. It is also cheaper to use when the amount of data increases.

Start with Deepgram, keep the research on Whisper, move to Whisper eventually.

References and Benchmarks

The benchmarks are provided above for a more coherent text, so I will only list references here.

[1]Comparing CPUs and GPUs	https://github.com/JuergenFleiss/aTrain#benchmarks
[2]Whisper Github(common voice benchmark):	https://github.com/openai/whisper
[3]Problems with Deepgram:	https://www.reddit.com/r/AskProgramming/comments/1cxo5ch/seeking_speechtotext_api_recommendations/
[4]Whisper Inference Speed on A100	https://docs.inferless.com/how-to-guides/deploy-a-whisper-large-v3-turbo
[5]Whisper paper	https://arxiv.org/pdf/2212.04356
[6]Deepgram documentation	https://arxiv.org/pdf/2212.04356
[7]Deepgram model comparison	https://deepgram.com/learn/nova-2-speech-to-text-api
[8]Custom model training	https://offers.deepgram.com/hubfs/Deepgram-Custom-Model-Training-Product%20Sheet.pdf?hsLang=en