**Speech to Text Models Comparison :**

Automatic speech-to-text recognition involves converting an audio file to editable text. Computer algorithms facilitate this process in four steps: analyse the audio, break it down into parts, convert it into a computer-readable format, and use the algorithm again to match it into a text-readable format.

In the past, this was a task only reserved for proprietary systems. This was disadvantageous to the user due to high licensing and usage fees, limited features, and a lack of transparency.

As more people researched these tools, creating your language processing models with the help of **open-source voice recognition systems became possible**. These systems, made by the community for the community, are easy to customize, cheap to use, and transparent, giving the user control over their data.

Here are the top open-source speech recognition engines we can start on:

**1. Whisper**

[Whisper](https://openai.com/research/whisper/) is Open AI’s newest brainchild that offers transcription and translation services.  Released in September 2022, this AI tool is one of the most accurate automatic speech recognition models. It stands out from the rest of the tools in the market due to the large number of training data sets it was trained on: 680 thousand hours of audio files from the internet. This diverse range of data improves the human-level robustness of the tool.

The main advantage of Whisper AI is it also comes with API version with little bit of costs, where as the library version is free to use (We may need to pay for cloud, etc)

**Pros**

* It supports content formats such as MP3, MP4, M4A, Mpeg, MPGA, WEBM, and WAV.
* It can transcribe 99 languages and translate them all into English.
* The tool is free to use.

**Cons**

* The larger the model, the more GPU resources it consumes, which can be costly.
* It will cost you time and resources to install and use the tool.
* It does not provide real-time transcription.

**2. Project DeepSpeech**

[Project DeepSearch](https://github.com/mozilla/DeepSpeech) is an **open-source speech-to-text** engine by Mozilla. This voice-to-text command and library is released under the Mozilla Public License (MPL). Its model follows the Baidu Deep Speech research paper, making it end-to-end trainable and capable of transcribing audio in several languages. It is also trained and implemented using Google’s TensorFlow.

**Pros**

* DeepSpeech is easy to customize since it’s a code-native solution.
* It provides special wrappers for Python, C, .Net Framework, and Javascript, allowing you to use the tool regardless of the language.
* It can function on various gadgets, including a Raspberry Pi device.
* Its per-word error rate is remarkably low at 7.5%.
* Mozilla takes a serious approach to privacy concerns.

**Cons**

* Mozilla is reportedly ending the development of DeepSpeech. This means there will be less support in case of bugs and implementation problems.

**3. Kaldi**

[Kaldi](https://github.com/kaldi-asr/kaldi) is a speech recognition tool purposely created for speech recognition researchers. It’s written in C++ and released under the Apache 2.0 license, one of the least restrictive licenses. Unlike tools like Whisper and DeepSpeech, which focus on deep learning, Kaldi primarily focuses on speech recognition models that use old-school, reliable tools. These include models like HMMs (Hidden Markov Models), GMMs (Gaussian Mixture Models), and FSTs (Finite State Transducers.)

**Pros**

* Kaldi is very reliable. Its code is thoroughly tested and verified.
* Although its focus is not on deep learning, it has some models that can help with transcription services.
* It is perfect for academic and industry-related research, allowing users to test their models and techniques.
* It has an active forum that provides the right amount of support.
* There are also resources and documentation available to help users address any issues.
* Being open-source, users with privacy or security concerns can inspect the code to understand how it works.

**Cons**

* Its classical approach to models may limit its accuracy levels.
* Kaldi is not user-friendly since it operates on a Command-line interface.
* It's pretty complex to use, making it suitable for users with technical experience.
* You need lots of computation power to use the toolkit.

**4. Athena**

[Athena](https://github.com/athena-team/athena) is another sequence-to-sequence-based **speech-to-text open-source** engine released under the Apache 2.0 license. This toolkit suits researchers and developers with their end-to-end speech processing needs. Some tasks the models can handle include automatic speech recognition (ASR), speech synthesis, voice detection, and keyword spotting. All the language models are implemented on TensorFlow, making the toolkit accessible to more developers.

**Pros**

* Athena is versatile in its use, from transcription services to speech synthesis.
* It does not depend on Kaldi since it has its pythonic feature extractor.
* The tool is well maintained with regular updates and new features.
* It is open source, free to use, and available to various users.

**Cons**

* It has a deep learning curve for new users.
* Although it has a WeChat group for community support, it limits the accessibility to only those who can access the platform.

**5. Coqui**

[Coqui](https://github.com/coqui-ai/STT) is an advanced deep learning toolkit perfect for training and deploying STT models. Licensed under the Mozilla Public License 2.0, you can use it to generate multiple transcripts, each with a confidence score. It provides pre-trained models alongside example audio files you can use to test the engine and help with further fine-tuning. Moreover, it has well-detailed documentation and resources that can help you use and solve any arising problems.

**Pros**

* The STT models it provides are highly trained with high-quality data.
* The models support multiple languages.
* There is a friendly support community where you can ask questions and get any details relating to STT.
* It supports real-time transcription with extremely low latency in seconds.
* Developers can customize the models to various use cases, from transcription to acting as voice assistants.

**Cons**

* Coqui stopped to maintain the STT project to focus on their text-to-speech toolkit. This means you may have to solve any problems that arise by yourself without any help from support.

**6. Vosk**

One of the most compact and lightweight speech-to-text engines today is [Vosk](https://alphacephei.com/vosk/). This open-source toolkit works offline on multiple devices, including Android, iOS, and Raspberry Pi. It supports over 20 languages and dialects, including English, Chinese, Portuguese, Polish and German.

Vosk provides users with small language models that do not take up much space. Ideally, around 50MB. However, a few large models can take up to 1.4GB. The tool is quick to respond and can convert speech to text continuously.

**Pros**

* It can work with various programming languages such as Java, Python, C++, Kotlyn, and Shell, making it a versatile addition for developers.
* It has various use cases, from transcriptions to developing chatbots and virtual assistants.
* It has a fast response time.

**Cons**

* The engine's accuracy can vary depending on the language and accent.
* You need coding expertise to integrate and use the tool.

**7. ESPnet**

[ESPnet](https://espnet.github.io/espnet/) is an **open-source speech-to-text** **software** released under the Apache 2.0 license. It provides end-to-end speech processing capabilities that cover tasks ranging from ASR, translation, speech synthesis, enhancement, and diarization. The toolkit stands out for leveraging Pytorch as its deep learning framework and following the Kaldi data processing style. As a result, you get comprehensive recipes for various language-processing tasks. The tool is also multi-lingual as it is capable of handling various languages. Use it with the readily available pre-trained models or create your own according to your needs.

**Pros**

* The toolkit delivers a stand-out performance compared to other speech-to-text software.
* It can process audio in real time, making it suitable for live transcription services.
* Suitable for use by researchers and developers.
* It is one of the most versatile tools to deliver various speech-processing tasks.

**Cons**

* It can be complex to integrate and use for new users.
* You must be familiar with Pytorch and Python to run the toolkit.

**6. Julius**

[Julius](https://github.com/julius-speech/julius) is one of the oldest speech-to-text projects, dating back to 1997, with roots in Japan. It is available under the BSD -3-license, making it accessible to developers. It strongly supports Japanese ASR, but being a language-independent program, the model can understand and process multiple languages, including English, Slovenian, French, Thai, and others. The transcription accuracy largely depends on whether you have the right language and acoustic model. The project is written in the most common language, C, allowing it to work in Windows, Linux, Android, and macOS systems.

**Pros**

* Julius can perform real-time speech-to-text transcription with low memory usage.
* It has an active community that can help with ASR problems.
* The models trained in English are readily available on the web for download.
* It does not need internet access for speech recognition, making it suitable for users needing privacy.

**Cons**

* Like any other open-source program, you need users with technical experience to make it work.
* It has a huge learning curve.

Feel free to explore cloud based models, which are available in API version, few examples can be Amazon Transcribe, Google Cloud Speech To Text, AssemblyAI, LemonFox.ai, etc.