# Characteristics and limitations of Speech-to-Text

11/12/2021 • 5 minutes to read • 



In this article



Speech-to-Text recognizes what's spoken in an audio input and generates transcription outputs. This requires proper setup for an expected language used in the audio input and spoken styles. Non-optimal settings may lead to lower accuracy.

## Language of Accuracy

The industry standard to measure Speech-to-Text accuracy is Word Error Rate (WER) . WER counts the number of incorrect words identified during recognition, then divides by the total number of words provided in correct transcript (often created by human labeling).

To understand the detailed WER calculation, please see this document about Speech-to-Text evaluation and improvement.

## Transcription Accuracy and System Limitations

Speech-to-Text uses a unified speech recognition machine learning model to understand what is spoken in a wide range of contexts and topic domains, such as command-and-control, or dictation and conversations. Therefore, you don't need to consider using different models for your application or feature scenarios.

However, you need to specify a language (or locale) for each audio input. This must match with actual language spoken in an input voice. Please check the list of supported locales for more details.

There are many factors that lead to a lower accuracy in transcription.

- **Acoustic quality:** Speech-to-Text enabled applications and devices may use a wide variety of microphone types and specifications. The unified speech models have been created based on various voice audio device scenarios, such as telephones, mobile phones and speaker devices. However, voice quality may be degraded by the way users speak to microphones, even if they use high-quality microphones. For example, if users stay far from a microphone, the input quality would be too low, while speaking too close to a microphone would cause audio quality deterioration. Both cases may adversely impact Speech-to-Text accuracy.
- **Non-speech noise:** If an input audio contains a certain level of noise, it has an impact on accuracy. Noise may come from the audio devices used to make a recording, while audio input itself may contain noise, such as background noise and environmental noise.
- **Overlapped speech:** There could be multiple speakers within range of an audio input devices, and they may speak at the same time. Also, other speakers may speak in the background while the main user is speaking.
- **Vocabularies:** The Speech-to-Text model has knowledge of wide variety of words in many domains. However, users may speak company specific terms and jargon (which are "out of vocabulary"). If a word appears in the audio that doesn't exist in a model, it will result in a mis-transcription.
- **Accents:** Even within one locale (such as "English -- United States"), many people have different accents. Very particular accents may also lead to a mis-transcription.
- **Mismatched locales:** Users may not speak the languages you expect. If you specified English -- United States (en-US) for an audio input, but a user spoke Swedish, for instance, accuracy would be worse.

Due to these acoustic and linguistic variations, customers should expect a certain level of inaccuracy in the output text when designing an application.

# Best practices to improve system accuracy

As discussed above, acoustic conditions, such as background noise, side speech, distance to microphone, speaking styles and characteristics can adversely affect the accuracy of what is being recognized.

Please consider the following application/service design principles for better speech experiences.

- **Design UIs to match input locales:** Mismatched locales will deteriorate accuracy. The Speech SDK supports automatic language detection, but it only detects one out of four locales specified at runtime. You still need to understand in what locale your users will speak. Your user interfaces should clearly indicate which languages the users may speak by such measures as a drop-down list of languages allowed to be used. Please see supported locales.

- **Allow users to try again:** Misrecognition may occur due to some temporary issues, such as unclear or fast speech or a long pause. If your application expects specific transcriptions, such as predefined action commands (such as "Yes" and "No"), and did not get any of them, users should be able to try again. A typical method is to tell users "Sorry, I didn't get that. Please try again".

- **Confirm before taking an action by voice:** Just as with keyboard/click/tap-based user interfaces, if an audio input can trigger an action, users should be given an opportunity to confirm the action, especially by displaying or playing back what was recognized/transcribed. A typical example is sending a text message by voice. An app repeats what was recognized and asks for confirmation: "You said, 'Thank you'. Send it or change it?".

- **Add custom vocabularies:** The general speech recognition model provided by Text-to-Speech covers a broad vocabulary. However, scenario specific jargon and named entities (people names, product names, etc.) may be underrepresented and what words and phrases are likely to be spoken can vary significantly depending on the scenario. If you can anticipate which words and phrases will be spoken (for instance, when a user selects an item from a list), you may want to use the phrasal list grammar, see "Improving recognition accuracy" in Get started with Speech-to-Text..

- **Use Custom Speech:** If Speech-to-Text accuracy in your application scenarios remains low, you might want to consider customizing the model for your acoustic and linguistic variations. You may create your own models by training with your own voice audio data or text data. See more details about Custom Speech.

# Fairness

At Microsoft, we strive to empower every person on the planet to achieve more. An essential part of this goal is working to create technologies and products that are fair and

inclusive. Fairness is a multi-dimensional, sociotechnical topic and impacts many different aspects of our product development. You can learn more about Microsoft's approach to fairness here .

One dimension we need to consider is how well the system performs for different groups of people. Research has shown that without conscious effort focused on improving performance for all groups, it is often possible for the performance of an AI system to vary across groups based on factors such as race, ethnicity, region, gender and age.

Speech-to-Text models are trained and tuned with voice audio with variations including:

- Microphones and device specifications
- Speech environment
- Speech scenarios
- Languages and accents
- Age and genders
- Ethnic background

Each version of the Speech-to-Text model is tested and evaluated against various test sets to make sure the model can perform without a large gap in each of the evaluation criteria.

# Evaluating Text-to-Speech in your applications

The performance of Text-to-Speech will vary depending on the real-world uses and conditions that customers implement. In order to ensure optimal performance in their scenarios, customers should conduct their own evaluations of the solutions they implement using Text-to-Speech.

A test voice dataset should consist of actual voice inputs collected in your applications in production. Customers should randomly sample data to reflect real user variations over a certain period of time. Additionally, the test dataset should be refreshed periodically to reflect changes in the variations.

# Next steps

- Guidance for integration and responsible use with Speech-to-Text