

VG | Part A: Hard cases for Speech Recognition: CASE STUDY

1. Reflect on the following:

- Can you think of any names for fictional places, people or objects that are not recognized? (Keep your final project in mind!)

```
"the idol": { person: "Menzi Idol"},
"menzi": { person: "Menzi Idol"}, // nickname: very difficulty
"menzi idol": { person: "Menzi Idol"},
```

```
"the kid": { person: "Kodi Grand Hotel"},
"kodi grand hotel": { person: "Kodi Grand Hotel"},
"kodi": { person: "Kodi Grand Hotel"},
"the magician": { person: "Kodi Grand Hotel"},
```

```
"dreamons": { person: "dreamons" }, // it is an invented word. it finds out something different
```

- Did you come across any real locations or people that are also just not picked up?

```
"but so, you are a bastard": { person: "Menzi Idol" }, // it doesn't work because it is censored *****
```

```
"so for the dessert we have tiramisu": { person: "Kodi Grand Hotel" }, // dessert close to research , tiramisù should be tiramisu
```

- Any specific accent you are using that makes words difficult to process?

```
"the hierophant": { person: "The Hierophant"}, // difficult for pronunciation
```

2. Write some sample code to test the confidence scores in speech recognition. Take a look at the confidence score with the help of XState's Visualizer (or you can log it). How good is it?

I will analyse three different cases:

- **Nickname/unusual word case: “menzi”**
- **Error in pronunciation/very marked pronunciation case: “the hierophant”**
- **New word case: “dreamons”**

3. Think about how this problem could be solved. Why do you think recognition falters for the examples that you tried? -> **Part A-VG. Azure Custom Speech**

To solve the problem you will use Custom Speech:

- You will basically have to provide data, either plain text or audio files, to help the recognition process.
- Train and deploy your model (enable content logging). Note the **Endpoint ID**.

To test your model:

- Create a file `dm3.ts` which implements a very basic ASR test (analogous to `dm.ts` in this repository). Add the following to your `settings` object:
 - `speechRecognitionEndpointId: "paste your Endpoint ID here",`
- Now you can test your new ASR model! You will be able to download the log files for your model in Custom Speech interface.

Extend the your report with the following information:

- Which new words are now supported and can be tested. Report should contain your Endpoint ID.

I used a txt file with 10 sentences for each of the critical words, in this report we analyse just the first 3:

- menzi
- hierophant
- dreamons
- dessert
- ASL goodbye
- Kodi

<https://speech.microsoft.com/portal/8032a881d1714f93a1097e4d35d8b4a6/customspeech/78d41433-40a9-488b-92b9-fd3818f63a9c/data>

Azure AI | Speech Studio

Speech Studio > Custom Speech > Speech datasets

Training and testing dataset Editor

Upload speech data for testing or training a speech recognition model.

Upload data Train Test Export to Editor Rename Delete Download as CSV

Name	Description	Type
dreamons_dataset	menzi, hierophant, dremons, dessert, ASL goodbye, Kodi	Plain text

Then:

Azure AI | Speech Studio

«

Speech Studio > Custom Speech > Train custom models

Overview

Prepare your data

Custom Speech projects

Marco Leali
English (United States)

Speech datasets

Train custom models

Test models

Deploy models

Use your speech data to train a custom speech model

Train a new modelDeploy modelsTest modelRenameDeleteCopy to

Name ▾	Description ▾	Baseline ▾
dreamons_total_model		20241218
hierophant_model		20241218
dreamons_model		20241218
menzi_model		20241218

And finally I got the endpoint copy and pasted inside the dm.ts file.

Azure AI | Speech Studio

«

Speech Studio > Custom Speech > Deploy models

Overview

Prepare your data

Custom Speech projects

Marco Leali
English (United States)

Speech datasets

Train custom models

Test models

Deploy models

Define an endpoint to deploy a custom speech model in your sol

Deploy modelChange modelRenameDeleteDownload as CSV

Name ▾	Description ▾	Model ▾	Logging ▾
dreamons		dreamons_total_model	Yes

```
const settings: Settings = {  
  azureCredentials: azureCredentials,  
  azureRegion: "northeurope",  
  asrDefaultCompleteTimeout: 0,  
  asrDefaultNoInputTimeout: 5000,  
  locale: "en-US",  
  ttsDefaultVoice: "en-US-DavisNeural",  
  speechRecognitionEndpointId: "c9cf0d8b-777e-479e-afdd-9d399383fe53",  
};
```

```
speechRecognitionEndpointId: "c9cf0d8b-777e-479e-afdd-9d399383fe53"
```

1) Nickname/unusual word case: “menzi”

before of custom speech: it misinterpreted the word with Manzi

[SpSt] All ready	exec — raise-1db27a82.development.esm.js:567
You just said: Manzi	params — dm3.ts:283
Full name: Manzi	params — dm3.ts:284
Speaking: "You just said: Manzi. It is not in the grammar."	spst.speak — dm3.ts:160
State update	dm3.ts:894
State value: — "CheckGrammarPerson"	dm3.ts:895
State context: — {spstRef: Actor, person: [{utterance: "Manzi", confidence: 0.07086755}], day: null, ...}	dm3.ts:896
State update	dm3.ts:894
State value: — "CheckGrammarPerson"	dm3.ts:895
State context: — {spstRef: Actor, person: [{utterance: "Manzi", confidence: 0.07086755}], day: null, ...}	dm3.ts:896
[SpSt→TTS] SPEAK — {utterance: "You just said: Manzi. It is not in the grammar."}	exec — raise-1db27a82.development.esm.js:567
[TTS.start] with input — {wsaTTS: \$0adf7fcdea541e7b\$export\$1268b12b5ca510be, wsaUtt: class \$fd1e2557ea351c52\$var\$SpeechSynthesisUtterance, ttsLexicon: undefined, ...}	speechstate.js:27544
[TTS] SPEAK: — "You just said: Manzi. It is not in the grammar."	speechstate.js:27549

before of custom speech: it misinterpreted the word with Menzie

[SpSt] All ready	exec — raise-1db27a82.development.esm.js:567
You just said: Menzie	params — dm3.ts:283
Full name: Menzie	params — dm3.ts:284
Speaking: "You just said: Menzie. It is not in the grammar."	spst.speak — dm3.ts:160
State update	dm3.ts:894
State value: — "CheckGrammarPerson"	dm3.ts:895
State context: — {spstRef: Actor, person: [{utterance: "Menzie", confidence: 0.14231506}], day: null, ...}	dm3.ts:896
State update	dm3.ts:894
State value: — "CheckGrammarPerson"	dm3.ts:895
State context: — {spstRef: Actor, person: [{utterance: "Menzie", confidence: 0.14231506}], day: null, ...}	dm3.ts:896
[SpSt→TTS] SPEAK — {utterance: "You just said: Menzie. It is not in the grammar."}	exec — raise-1db27a82.development.esm.js:567
[TTS.start] with input — {wsaTTS: \$0adf7fcdea541e7b\$export\$1268b12b5ca510be, wsaUtt: class \$fd1e2557ea351c52\$var\$SpeechSynthesisUtterance, ttsLexicon: undefined, ...}	speechstate.js:27544
[TTS] SPEAK: — "You just said: Menzie. It is not in the grammar."	speechstate.js:27549
[TTS→SpSt] TTS_STARTED	exec — raise-1db27a82.development.esm.js:567
State update	dm3.ts:894
State value: — "CheckGrammarPerson"	dm3.ts:895
State context: — {spstRef: Actor, person: [{utterance: "Menzie", confidence: 0.14231506}], day: null, ...}	dm3.ts:896
[TTS] TTS_STARTED	speechstate.js:27554
[TTS→SpSt] SPEAK_COMPLETE	exec — raise-1db27a82.development.esm.js:567

after of custom speech: It works!

It has been trained on a model based on the follwing dataset (txt file) (here you see just the part for “menzi” word):

Menzi is a great lawyer.
I just met Menzi at the concert.
Do you know Menzi Idol?
Menzi Idol is a good friend.
So, Menzi, do you want to come?
How are you Menzi?
Menzi is reading a book.
What if it happened that Menzi did it?
Kid was playing with Menzi Idol.
Really? Menzi Idol is back in the town?

🔍 You just said: Menzi	params — dm3.ts:290
👤 Full name: Menzi	params — dm3.ts:291
🗣️ Speaking: "You just said: Menzi. It corresponds to Menzi Idol."	spst.speak — dm3.ts:165
▼ State update	dm3.ts:901
🔍 State value: — "CheckGrammarPerson"	dm3.ts:902
🔍 ▶ State context: — {spstRef: Actor, person: [{utterance: "Menzi", confidence: 0.8120727}], day: null, ...}	dm3.ts:903
▼ State update	dm3.ts:901
🔍 State value: — "CheckGrammarPerson"	dm3.ts:902
🔍 ▶ State context: — {spstRef: Actor, person: [{utterance: "Menzi", confidence: 0.8120727}], day: null, ...}	dm3.ts:903
🔍 [SpSt→TTS] SPEAK — {utterance: "You just said: Menzi. It corresponds to Menzi Idol."}	exec — raise-1db27a82.development.esm.js:567
🔍 ▶ [TTS.start] with input — {wsaTTS: \$0adf7fcdea541e7b\$export\$1268b12b5ca510be, wsaUtt: class \$fd1e2557ea351c52\$var\$SpeechSynthesisUtterance, ttsLexicon: undefined, ...}	speechstate.js:27544
🔍 [TTS] SPEAK: — "You just said: Menzi. It corresponds to Menzi Idol."	speechstate.js:27549
🔍 [TTS→SpSt] TTS_STARTED	exec — raise-1db27a82.development.esm.js:567

2) Error in pronunciation/very marked pronunciation case: “the hierophant”

before

🔍 You just said: The Hierophant	params — dm3.ts:283
👤 Full name: The Hierophant	params — dm3.ts:284
🗣️ Speaking: "You just said: The Hierophant. It corresponds to The Hierophant."	spst.speak — dm3.ts:160
▼ State update	dm3.ts:894
🔍 State value: — "CheckGrammarPerson"	dm3.ts:895
🔍 ▶ State context: — {spstRef: Actor, person: [{utterance: "The Hierophant", confidence: 0.52203834}], day: null, ...}	dm3.ts:896
▼ State update	dm3.ts:894
🔍 State value: — "CheckGrammarPerson"	dm3.ts:895
🔍 ▶ State context: — {spstRef: Actor, person: [{utterance: "The Hierophant", confidence: 0.52203834}], day: null, ...}	dm3.ts:896
🔍 [SpSt→TTS] SPEAK — {utterance: "You just said: The Hierophant. It corresponds to The Hierophant."}	exec — raise-1db27a82.development.esm.js:567
🔍 ▶ [TTS.start] with input — {wsaTTS: \$0adf7fcdea541e7b\$export\$1268b12b5ca510be, wsaUtt: class \$fd1e2557ea351c52\$var\$SpeechSynthesisUtterance, ttsLexicon: undefined, ...}	speechstate.js:27544
🔍 [TTS] SPEAK: — "You just said: The Hierophant. It corresponds to The Hierophant."	speechstate.js:27549
🔍 [TTS→SpSt] TTS_STARTED	exec — raise-1db27a82.development.esm.js:567
▼ State update	dm3.ts:894
🔍 State value: — "CheckGrammarPerson"	dm3.ts:895
🔍 ▶ State context: — {spstRef: Actor, person: [{utterance: "The Hierophant", confidence: 0.52203834}], day: null, ...}	dm3.ts:896

after: slightly better. To really improve it I should upload my different audios when I say The hierophant with my particular pronunciation

🔍 You just said: The Hierophant	f params — dm3.ts:290
👤 Full name: The Hierophant	f params — dm3.ts:291
🗣️ Speaking: "You just said: The Hierophant. It corresponds to The Hierophant."	f spst.speak — dm3.ts:165
▼ State update	dm3.ts:901
📄 State value: — "CheckGrammarPerson"	dm3.ts:902
📄 ▶ State context: — {spstRef: Actor, person: [{utterance: "The Hierophant", confidence: 0.59529316}], day: null, ...}	dm3.ts:903
▼ State update	dm3.ts:901
📄 State value: — "CheckGrammarPerson"	dm3.ts:902
📄 ▶ State context: — {spstRef: Actor, person: [{utterance: "The Hierophant", confidence: 0.59529316}], day: null, ...}	dm3.ts:903
🔵 [SpSt→TTS] SPEAK — {utterance: "You just said: The Hierophant. It corresponds to The Hierophant."}	f exec — raise-1db27a82.development.esm.js:567
🔵 ▶ [TTS.start] with input — {wsaTTS: \$0adf7fcdea541e7b\$export\$1268b12b5ca510be, wsaUtt: class \$fd1e2557ea351c52\$var\$SpeechSynthesisUtterance, ttsLexicon: undefined, ...}	speechstate.js:27544
🔵 [TTS] SPEAK: — "You just said: The Hierophant. It corresponds to The Hierophant."	speechstate.js:27549
🔵 [TTS→SpSt] TTS_STARTED	f exec — raise-1db27a82.development.esm.js:567
▼ State update	dm3.ts:901
📄 State value: — "CheckGrammarPerson"	dm3.ts:902
📄 ▶ State context: — {spstRef: Actor, person: [{utterance: "The Hierophant", confidence: 0.59529316}], day: null, ...}	dm3.ts:903
🔵 [TTS] TTS_STARTED	speechstate.js:27554
🔵 [TTS→SpSt] SPEAK_COMPLETE	f exec — raise-1db27a82.development.esm.js:567

3) The new word case: dreamons

before

It could not recognise that word, since it does not exist.

after

🔍 You just said: Dreamons
👤 Full name: Dreamons
🗣️ Speaking: "You just said: Dreamons. It corresponds to dreamons."
▼ State update
📄 State value: — "CheckGrammarPerson"
📄 ▶ State context: — {spstRef: Actor, person: [{utterance: "Dreamons", confidence: 0.73826027}], day: null, ...}
▼ State update
📄 State value: — "CheckGrammarPerson"
📄 ▶ State context: — {spstRef: Actor, person: [{utterance: "Dreamons", confidence: 0.73826027}], day: null, ...}
🔵 [SpSt→TTS] SPEAK — {utterance: "You just said: Dreamons. It corresponds to dreamons."}

after

🔍 You just said: Dreamons	f params — dm3.ts:290
👤 Full name: Dreamons	f params — dm3.ts:291
🗣️ Speaking: "You just said: Dreamons. It corresponds to dreamons."	f spst.speak — dm3.ts:165
▼ State update	dm3.ts:901
📄 State value: — "CheckGrammarPerson"	dm3.ts:902
📄 ▶ State context: — {spstRef: Actor, person: [{utterance: "Dreamons", confidence: 0.17028476}], day: null, ...}	dm3.ts:903
▼ State update	dm3.ts:901
📄 State value: — "CheckGrammarPerson"	dm3.ts:902
📄 ▶ State context: — {spstRef: Actor, person: [{utterance: "Dreamons", confidence: 0.17028476}], day: null, ...}	dm3.ts:903
🔵 [SpSt→TTS] SPEAK — {utterance: "You just said: Dreamons. It corresponds to dreamons."}	f exec — raise-1db27a82.development.esm.js:567
🔵 ▶ [TTS.start] with input — {wsaTTS: \$0adf7fcdea541e7b\$export\$1268b12b5ca510be, wsaUtt: class \$fd1e2557ea351c52\$var\$SpeechSynthesisUtterance, ttsLexicon: undefined, ...}	speechstate.js:27544
🔵 [TTS] SPEAK: — "You just said: Dreamons. It corresponds to dreamons."	speechstate.js:27549
🔵 [TTS→SpSt] TTS_STARTED	f exec — raise-1db27a82.development.esm.js:567
▼ State update	dm3.ts:901
📄 State value: — "CheckGrammarPerson"	dm3.ts:902
📄 ▶ State context: — {spstRef: Actor, person: [{utterance: "Dreamons", confidence: 0.17028476}], day: null, ...}	dm3.ts:903
🔵 [TTS] TTS_STARTED	speechstate.js:27554
🔵 [TTS→SpSt] SPEAK_COMPLETE	f exec — raise-1db27a82.development.esm.js:567