Final lab report

Nico Mayr

October 24, 2023

Technical University of Munich

Supportive Process Automation and Worker Assistance

Summer Semester 2023

1 Introduction

To enrich the Cocktail-Robot with the functionality of an easy-to-use and robust ordering mechanism, a process-oriented web-interface was used that allows the ordering of available cocktails via speech input. The system connects to an available speech recognition service of the browser, detects as soon as a cocktail is mentioned, and returns it to the process engine if a defined confirmation keyword is mentioned.

2 Approach

The main functionality is reflected within the first line of the file `transcript.js` (see lst. 1).

```
const recognition = new (window.SpeechRecognition ||
    window.webkitSpeechRecognition ||
    window.mozSpeechRecognition ||
    window.msSpeechRecognition ||
    window.oSpeechRecognition)();
```

Listing 1: Speech Recognition

It creates a Speech Recognition Object [3] implementing the Web Speech API, by selecting the available implementation from the respective browser. After the initialization (see fig. 1), every time a speech is recognized, the `onresult()` function of `recognition` Object is called with an `event` argument. From that, all containing text is extracted and given to the `normalizeText()` function to obtain text that is as simple as possible to make later name extraction easier, by for example removing any whitespace or dashes and converting everything to lowercase. The normalized text is then collated with the cocktail matrix, an object that has been initialized on page load, containing all the cocktail

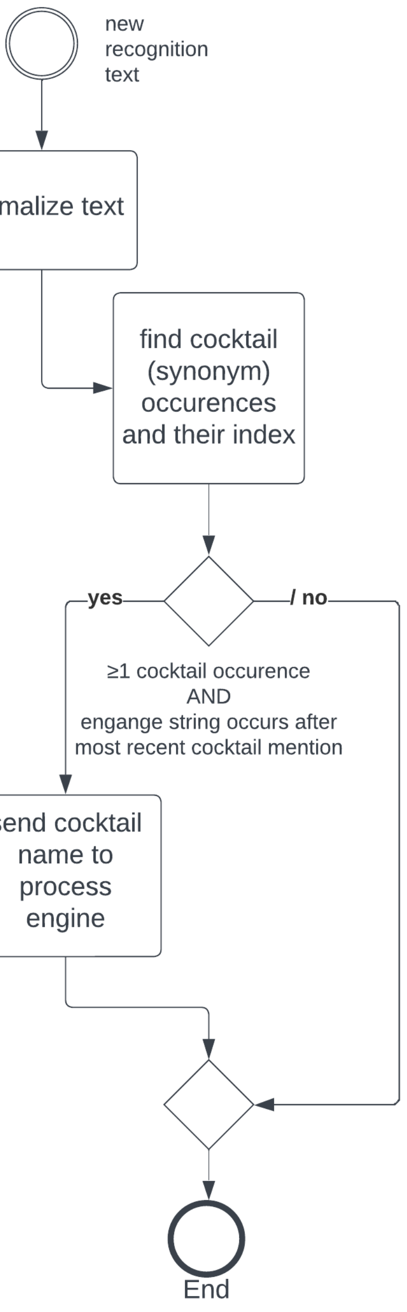

new recognition text
normalize text
find cocktail (synonym) occurences and their index
yes
/ no
≥1 cocktail occurence AND engange string occurs after most recent cocktail mention
send cocktail name to process engine
End

Figure 1: Speech Recognition and Cocktail Detection Workflow

names and their synonyms. From that, one gets a list of pairs consisting of the index of occurrence and the cocktail row (the name of a cocktail and all its synonyms), for every instance where any name or synonym occurs within the text. To get the most recently mentioned cocktail name, one simply returns the first element of the row that has the highest associated index. The detected cocktail is displayed on the page along with the request for confirmation. With that, one additionally listens for the confirmation keyword and returns the cocktail name to the process engine as soon as it is mentioned.

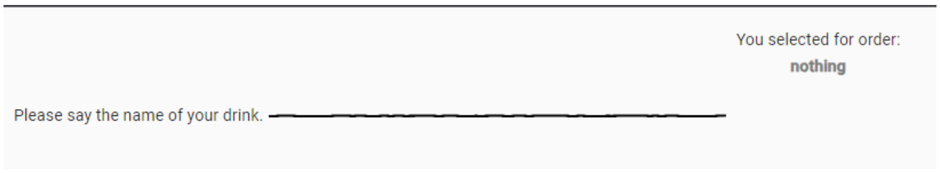
The webpage interface also includes a visual representation of the recorded waveform to ensure the user that the microphone is working, and the processing is underway. To complete the user experience, a cocktail browser has been added at the bottom for the user to see all cocktails. This browser is a modified version of a preexisting Web-App [1] to better fit the needs of the user and the layout of the page.

Lastly, not relevant to the user but still helpful, a `speak()` function is included to allow the testing of the detection functionality via text input.

3 Example process instance

3.1 Client

After the microphone permission has been granted to the site, the user is greeted with a waveform, visualizing the sound being picked up by the mic. Further, a simple instruction, pointing the user towards the purpose of the page, is visible in the middle left (see fig. 2).

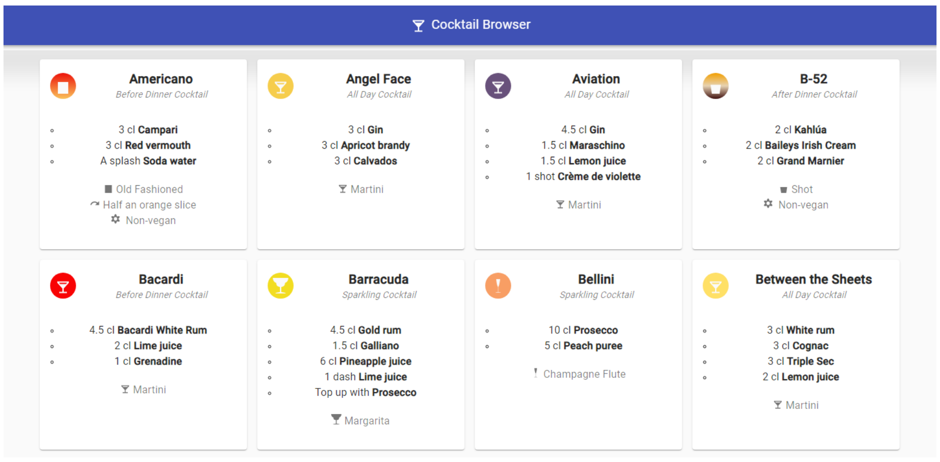


You selected for order:
nothing

Please say the name of your drink.

Figure 2: Initial view of the main part of the page

The user then can get an overview of all the available cocktails (see fig. 3) and see more details (see fig. 4) by clicking on them.



Americano
Before Dinner Cocktail
- 3 cl **Campari**
- 3 cl **Red vermouth**
- A splash **Soda water**

Old Fashioned
Half an orange slice
Non-vegan

Angel Face
All Day Cocktail
- 3 cl **Gin**
- 3 cl **Apricot brandy**
- 3 cl **Calvados**

Martini

Aviation
All Day Cocktail
- 4.5 cl **Gin**
- 1.5 cl **Maraschino**
- 1.5 cl **Lemon juice**
- 1 shot **Crème de violette**

Martini

B-52
After Dinner Cocktail
- 2 cl **Kahlúa**
- 2 cl **Baileys Irish Cream**
- 2 cl **Grand Marnier**

Shot
Non-vegan

Bacardi
Before Dinner Cocktail
- 4.5 cl **Bacardi White Rum**
- 2 cl **Lime juice**
- 1 cl **Grenadine**

Martini

Barracuda
Sparkling Cocktail
- 4.5 cl **Gold rum**
- 1.5 cl **Galliano**
- 6 cl **Pineapple juice**
- 1 dash **Lime juice**
- Top up with **Prosecco**

Margarita

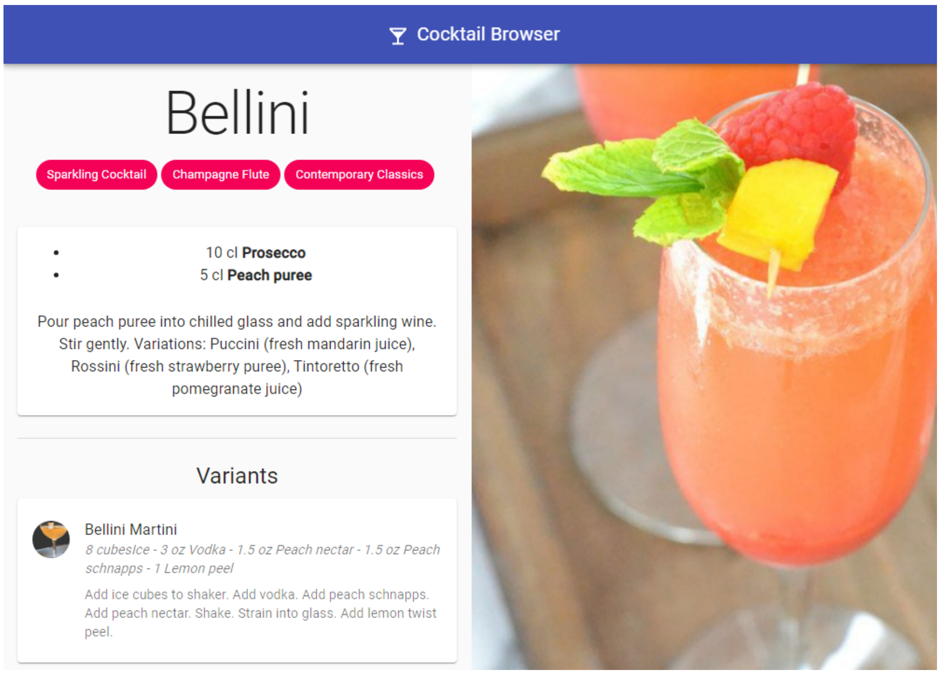
Bellini
Sparkling Cocktail
- 10 cl **Prosecco**
- 5 cl **Peach puree**

Champagne Flute

Between the Sheets
All Day Cocktail
- 3 cl **White rum**
- 3 cl **Cognac**
- 3 cl **Triple Sec**
- 2 cl **Lemon juice**

Martini

Figure 3: Cocktail browser



Bellini

- 10 cl **Prosecco**
- 5 cl **Peach puree**

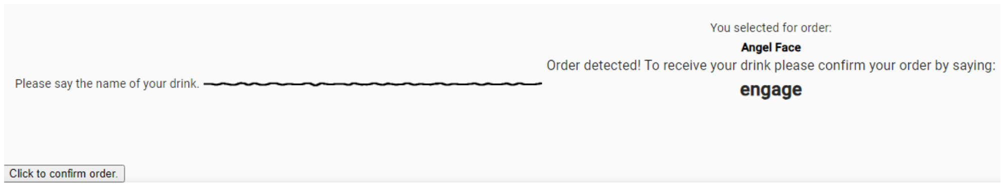
Pour peach puree into chilled glass and add sparkling wine. Stir gently. Variations: Puccini (fresh mandarin juice), Rossini (fresh strawberry puree), Tintoretto (fresh pomegranate juice)

Variants

Bellini Martini
8 cubesIce - 3 oz Vodka - 1.5 oz Peach nectar - 1.5 oz Peach schnapps - 1 Lemon peel
Add ice cubes to shaker. Add vodka. Add peach schnapps. Add peach nectar. Shake. Strain into glass. Add lemon twist peel.

Figure 4: Detailed cocktail view

Assuming now the user wants to order the cocktail "Angel face" and thus says its name. The system recognizes it and displays the current order status along with the request for confirmation (see fig. 5). This can not only be done verbally, via the engage-keyword, but also via a button in the bottom left corner.

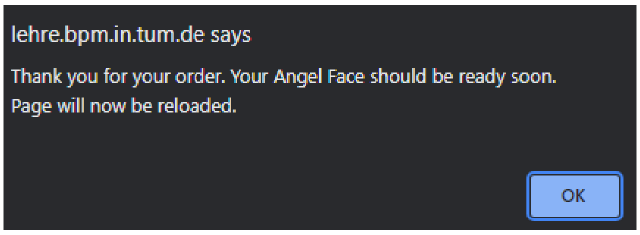


Please say the name of your drink.

You selected for order:
Angel Face
Order detected! To receive your drink please confirm your order by saying:
engage

Figure 5: Cocktail detection

Following that, the user confidently says "engage", resulting in the end of the ordering process and an alert to notify the user of the success (see fig. 6).



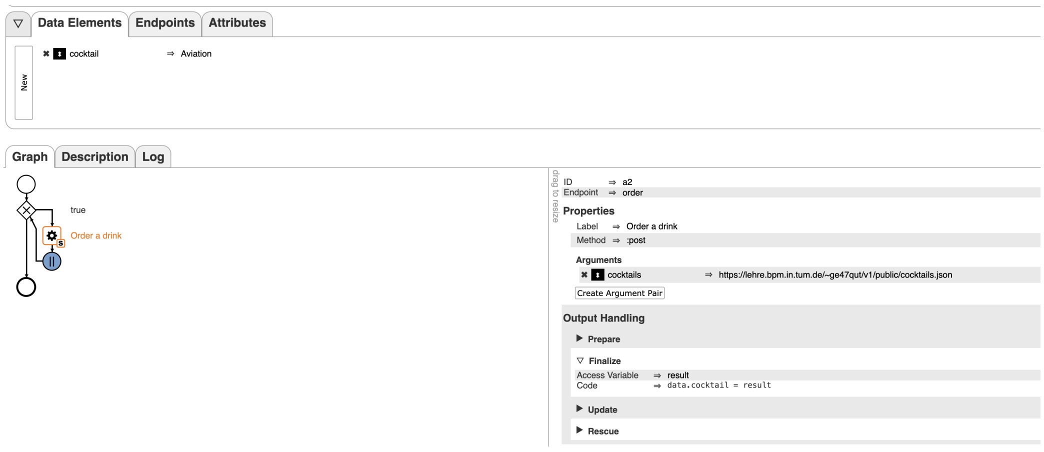
lehre.bpm.in.tum.de says

Thank you for your order. Your Angel Face should be ready soon.
Page will now be reloaded.

Figure 6: Order confirmation alert

3.2 Process engine

Receiving the cocktail name, the process engine now has all it needs to orchestrate the production of the cocktail the user desires (except in the screenshot the user seems to have ordered the "Aviation" cocktail). (see fig. 7).



cocktail ⇒ Aviation

ID ⇒ a2
Endpoint ⇒ order

Properties
Label ⇒ Order a drink
Method ⇒ :post

Arguments
cocktails ⇒ https://lehre.bpm.in.tum.de/~ge47qut/v1/public/cocktails.json

Output Handling
Prepare
Finalize
Access Variable ⇒ result
Code ⇒ `data.cocktail = result`
Update
Rescue

Figure 7: Process engine receiving the cocktail message

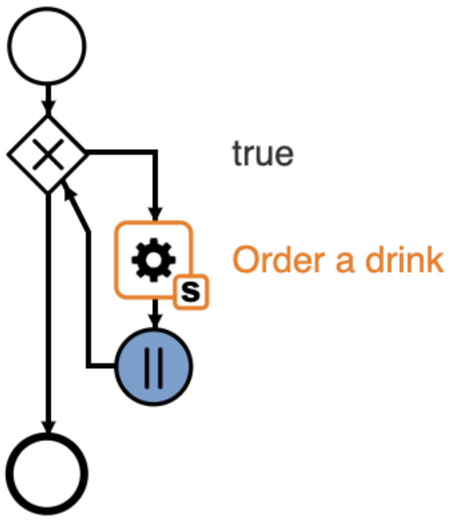


Figure 8: Process graph (enlarged)

4 Outlook

The software can be extended by providing more synonyms in the JSON-file such that the recognition preforms better. Further, one could include the functionality to convert symbolic numbers (like ”1”) to their respective alphabetical form (like ”one”). Moreover, one could also empirically experiment with increasing `maxAlternatives` attribute of the SpeechRecognition Object to possibly make recognition more tolerant to pronunciation.

Currently, a standard speech recognition model is used which may not always work optimally on highly specific cocktail names, despite the many considerations that have already been undergone. One could extend the system by one's own Model, by using the capability of Mozilla's *DeepSpeech* [2] to custom train their base model via deep learning. To start the recognition locally with the pre-trained base model, one simply executes the Python script with the additional flags `--model deepspeech-0.9.3-models.pbmm`

`--scorer deepspeech-0.9.3-models.scorer`.

References

[1] Mikey Hogarth. *Cocktails Browser*. original-date: 2019-04-28T00:28:44Z. July 7, 2023. URL: https://github.com/mikeyhogarth/cocktails (visited on 07/10/2023).

[2] Mozilla. *DeepSpeech*. 2023. URL: https://github.com/mozilla/DeepSpeech (visited on 07/03/2023).

[3] Mozilla. *SpeechRecognition*. 2023. URL: https://developer.mozilla.org/en-US/docs/Web/API/SpeechRecognition (visited on 07/03/2023).