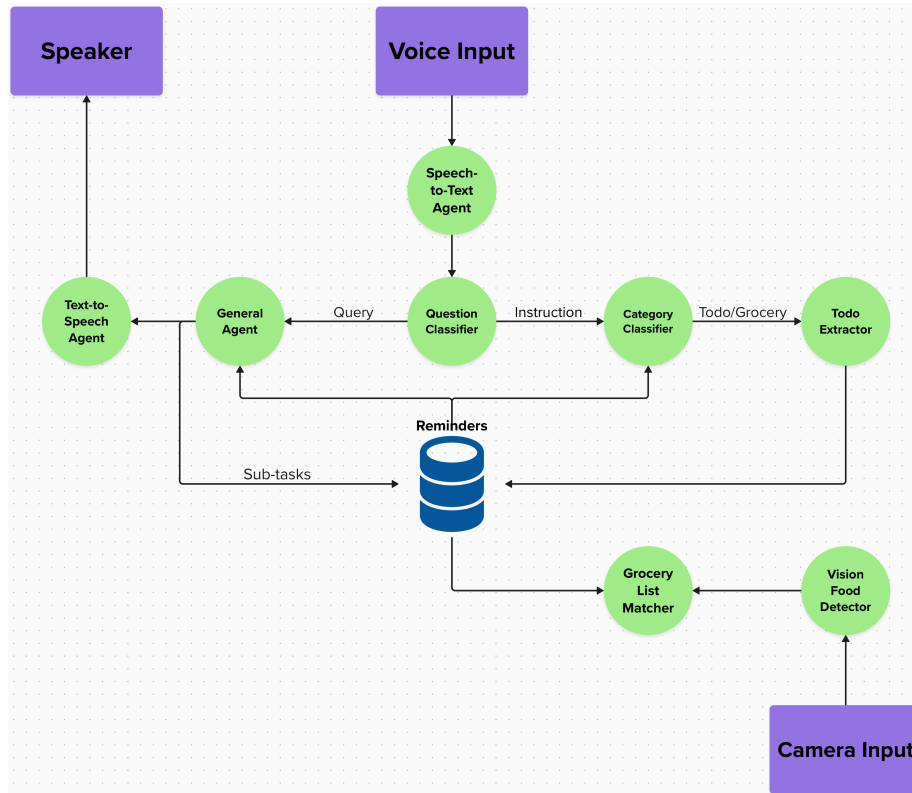# Aldone

Fausto Zamparelli, Daniel Falbo

May 30, 2024

Introduction

Aldone as in "AI-Done" is a tool for personal reminders and grocery list management. This is a proof of concept of a product that could save you time every day by allowing you to smoothly interact with it on the fly.

Aldone is a react web app that interacts with a node.js and a python server doing most of the magic. After a log-in you will be able to speak to it to add tasks to your grocery list, to-do list, or make questions about reminders you saved earlier. Furthermore, you could ask to split a task in sub-tasks or approximate how long a task will take. Aldone is able to detect fully by himself weather you are asking him to add something to your grocery list or to-do list.

The coolest thing is that once you will come home from the supermarket with tired legs and arms from carrying bags, you won't have to manually remove everything from this digital grocery list but you can use your phone camera to detect what foods you actually bought and then Aldone will remove them for you. Other use cases for this include checking what's already in your kitchen or marking products as bought after delegating your groceries to someone else.

Method

Visual representation of the agents that make up Aldone

As we can see Aldone was created by splitting its behind-the-scenes functionalities into multiple specialized AI agents that cooperate with each others. In order to fully understand how Aldone works under the hood it is best to understand the functionality of each of these agents alone. Let's work our way thorough the graph from top to bottom:

Speech-to-Text Agent

For the Speech-to-Text task, being a web app, Aldone uses the browser's built-in Web Speech API service for convenience. The implementation can be found by searching for "webkitSpeechRecognition" in the 'src/app/[username]/page.tsx' file.

Question Classifier

The task of the Question Classifier is to determine whether the user's input is a question/query or a statement/instruction. Anything that involves looking for information in the web or from the existing data in the database, is a query. Anything that provides new data within the input, is not a query.

Input	Label
"What's the capital of Italy?"	Query
"Add milk to my grocery list"	Not Query
"Split task X into subtasks"	Query

For example, the input "What's the capital of Italy?" is a query, while "Add milk to my grocery list" is an instruction. The input "Split task X into subtasks" is also a query because the data is not being provided by the input.

The Question Classifier is implemented using a neural network trained on a dataset of 500 examples. The question classifier is a pytorch neural network and its code can be found in the subdirectory "src/py/question_classifier".

In the subdirectory,

- `nndata.json`: is the dataset used to train the model

- `nntraining.py`: is the training script that generates the model. 80% of the data is used for training and 20% for testing. Of the training data, 25% is used for validation.

- `nnlabeler.py`: uses the trained model to label new input data.

```
❯ py nntraining.py
Epoch [1/10], Training Loss: 0.6730, Validation Loss: 0.6708
Epoch [2/10], Training Loss: 0.6405, Validation Loss: 0.6452
Epoch [3/10], Training Loss: 0.6141, Validation Loss: 0.6097
Epoch [4/10], Training Loss: 0.5305, Validation Loss: 0.5626
Epoch [5/10], Training Loss: 0.4792, Validation Loss: 0.5070
Epoch [6/10], Training Loss: 0.3936, Validation Loss: 0.4475
Epoch [7/10], Training Loss: 0.3198, Validation Loss: 0.3905
Epoch [8/10], Training Loss: 0.2877, Validation Loss: 0.3388
Epoch [9/10], Training Loss: 0.1875, Validation Loss: 0.2945
Epoch [10/10], Training Loss: 0.2605, Validation Loss: 0.2580
Accuracy on test set: 97.92%
```

We achieved 97% accuracy on the test set. Of course it's still a small dataset and more data would make it more solid.

Category Classifier

Todo Extractor

[Confusion Matrix here]

Vision Food Detector

We first asked the LLM model to give us the entire updated grocery list object, but empirically it often made errors. What we ended up doing is asking the LLM model for indices of the grocery list that were updated, and then we updated the grocery list object ourselves. This way, we could ensure that the grocery list object was updated correctly.

Grocery List Matcher

General Agent

Text-to-Speech Agent

Results

Glue Orchestration

Source of inspiration

[1] Cosmos: An Operating System for the AI Era
[2] aiXplain