Arman Keshavarz

8/15/25

Prof Bari

Predictive Analytics

Voice Controlled Chrome Navigator

The purpose of this dataset I created was to be able to encompass almost all commands that any chrome user would be able to do. I.E. I wanted the dataset to span all functionalities that chrome offers while also putting an emphasis on common commands such as searching, clicking and scrolling. Putting every single text input that users would want to do into the dataset is impossible but the corresponding rpc commands is far less so focusing on increasing the breadth of the rpc commands was an important part of the dataset creation. I originally started by just typing up around 50 common commands that I thought would be vital to the functionality of the chrome navigator and then figuring out the correct rpc command for said text input. A good example is this: {"utterance": "scroll down by 300 pixels", "rpc": {"method": "scroll", "params": {"direction": "down", "amount": 300}}} The best way to look at the dataset is the first column label is the key "utterance" and column we are trying to predict is the second key "params." In order to expand the dataset I had already created, I wrote a Python script that takes each of my seed commands and automatically generates a set number of brand-new commands using the OpenAI API. The script reads each seed from my original jsonl file and sends it to the model along with a system prompt that forces the output to only use the allowed RPC methods—navigate, click, type, and search—while also making sure there's variety in tone, websites, queries, and typing fields. For each seed, it generates exactly the number of new commands I specify (in my case 20), parses them, removes any duplicates, and appends them to a new jsonl file. Essentially, this script multiplies my original dataset by a fixed factor, keeping the structure consistent while greatly increasing the diversity and coverage of possible Chrome voice commands. The biggest challenge to creating this dataset was just being able to create the most diverse dataset possible using the OpenAI API. I think it would have been better if I hand wrote 200 commands myself instead of 50. In the future I would like to dedicate more time and money to

creating a better and larger dataset that would not only be able to capture single commands, but handle inputs that require a chain of commands to execute properly. This would require a completely new dataset but I believe it would absolutely make this a better tool.