How We Use Speech Transcription and Info Retrieval

Project Goal
Create a Question and Answering (QA) system that can answer a driver’s questions accurately and efficiently using the information provided from the car’s user manual.

How Speech Transcription and Info Retrieval fit into the QA Pipeline
• Speech-to-text transcribes the spoken question
• IR narrows down the full document to paragraphs which most likely contain the answer

Speech to Text
Experimented with Deep Speech, Microsoft Azure and Google’s API to select the best possible one

Information Retrieval

How the IR System Works
• Prepare the manual (done only once)
• Sentence tokenization (NLTK Punkt)
• Encoding sentences (Univ. Sent. Encoder QA)
• Indexing embeddings (Spotify’s Annoy)
• Retrieve candidates (each time a question is asked)
• Encode user question (Univ. Sent. Encoder QA)
• Find its nearest-neighbor sentences using the prebuilt index
• Paragraphs are weighted and ranked higher based on paragraph mean distance

Results
• 99.5% accuracy when retrieving 400 paragraphs
• Median ranking of 4 for the correct answer among candidate paragraphs
• 63% reduction in paragraphs sent to downstream BERT model

Spotify Annoy
• Builds data structure to efficiently find most similar vectors to any given vector
• Data Structure can be saved for later use

Why use IR?
In order to not overload the BERT QA System with too many paragraphs

User Interface

How It Works
1. Record and transcribe question
2. Run IR with the transcribed question
3. Displays top 10 answers

What We Used
• Frontend: HTML5/CSS3/JavaScript/Jquery
• Recording Audio: Recorder.js
• Backend: Python Flask