# Speech Completion using GPT-3

## 1 – Goal

In a project related to the speech synthesis and cloning project, I also worked on a speech completion tool. Again, in a medical setting, some people will temporarily or permanently lose the ability to speak, despite being awake, conscious, and coherent. In some cases, loss of speech is caused by being intubated. In other cases, it is the result of surgery. Furthermore, some patients might have difficulties using mobile devices to type out what they want or need to communicate. But even for someone with excellent dexterity, communicating through typing individual words is a slow, laborious, and frustrating process. So, I started to think about augmenting this process using new technologies.

## 2 – Technology

In recent years, our smartphones and computers have become more helpful. Word processors, email clients, and texting apps are now kind enough to offer auto-completion to our writing. While sometimes helpful, it has three disadvantages. First, they offer up only one of the many possible ways to complete our sentence. Second, they have a short forecasting window. They will offer completions to often used phrases like 'sincerely yours' or 'thank you for your consideration'. Third, for the most part, they are meant to complete our sentences rather than help us along our writing process. Fourth, as far as I can tell, they are not adaptive. They do not learn the uniqueness of the person and the words they use. The name of a spouse, a particular location, or common events.

However, a new player has entered the game. Meet OpenAI's GPT-3 Model. Having been trained on massive amounts of text data, it has required the ability to carry on an impressively human-sounding conversation with an actual human being. It is a generative algorithm using all that has been said before to predict what is next. It can be used as a chatbot, power the dialogue between our voice assistants and us, and even write stories when given some initial first lines. In those scenarios, what the algorithm generates quickly becomes gibberish in content, but it still sounds reasonably coherent. But in the short term, GPT-3 predictions of what will be said next are remarkable. And not only that, it has many alternatives to choose from, all ranked by probability. For this reason, I decided to use it to build a simple web application that uses the GPT-3 API to help people speed up their communication process.

## 3 – Solution

The web application had a straightforward design principle. It showed the user a 4 by 4 grid of tiles. As the user starts typing, GPT-3 is queried for the 16 best candidates of what might be typed next, looking much further into the future than a conventional auto-completion algorithm. These 16 options are displayed on each tile. From that point, the user can stop typing and select the tile that perfectly or perhaps sufficiently reflects what they want to say next. This triggers another query to GPT-3, and the tiles are replaced with 16 new options based on all that has already been typed and selected. The user also has the option to ignore the offerings and continue typing or can request the next 16 best options GPT-3 has to offer. Finally, when the phrase is fully formed, the user submits it to whomever they are speaking to. If the response from the other person is available to GPT-3, it uses that to refine its predictions even further of what its user wants to say in reply.

## 4 – Adaptive

GPT-3 has been trained on massive amounts of textual data, requiring heavy resources and time. Therefore, it is impractical to retrain or tune it on the fly for each user. However, a second, much smaller custom-built network can work alongside it, learning some of the user's key characteristics. For example, it can learn frequently used words, reflecting entities unique to the user, such as names and locations. And with that knowledge, it can override one or more of GPT -3's suggestions.

## 5 – Impact

I designed it as a web application because that was the only way, at the time, to interface with the GPT-3 API. But I imagine it would be equally useful on a handheld device like a tablet or smartphone. Unfortunately, I had neither the time nor the resources to implement the secondary adaptive model. However, I did manage to do voice cloning, which I also used in the speech synthesis project, allowing the application to synthesize the phrase in the user's voice. I cannot share this web application as it was hosted on my previous employer's internal network, but below, you can see a mock demo I recreated.

## 6 – Resources Used

|  |  |  |
| --- | --- | --- |
|  |  |  |