# 🌎 Team Planet Money Bot 🌍

CS 206: Computational Journalism, Final Writeup

Ori Spector, Tara Parekh, Wenna Qin, & Millie Lin

## A. Introduction

Think of a podcast you particularly enjoy, whether that's *This American Life,* NPR's *Up First,* or a gory true crime series like *My Favorite Murder.* If you don't listen to podcasts, think of your favorite journalism source. You trust this as a source of information and enjoy its unique style of communication. When you have a question — say, about a recent bank crash, or a particularly interesting serial killer — wouldn't you love to hear how your favorite podcast would answer it?

Podcasting, like most journalism, is a one-to-many medium. One podcast goes out to a range of different listeners. But the listeners cannot have individual interactions with the podcast; they can only listen to the non-personalized pieces of media that the show releases. Some shows take listener questions, but these are rare, non-immediate, and can never hope to cover the full breadth of questions that all its listeners might have. What if you could have a one-to-one experience with your favorite podcast? This might sound like a pipe dream, but it would also pose a significant contribution to the (mis)information landscape — the podcasts you listen to are trusted sources.


Let's bring in the star of our project: NPR's *Planet Money*. *Planet Money* is an American podcast that explains the economy in a creative and entertaining way. It's been running since 2008 and has 1 million monthly listeners, to whom it delivers accurate information in a delightful and accessible tone. But if one of these 1 million listeners has a question about the economy, how can they get it answered?

## B. The Problem

Let's say I have a nuanced economic question — say, "how does the Fed make money" or "is inflation bad?" or "what can economics tell me about my love life?" How would I get that answered?

Currently, I might Google my question. This brings up sources that are hard-to-verify: is Investopedia giving me a full and accurate answer? Is Wikipedia? I can also find academic literature on Google, which may be more definitively accurate, but is dense and hard for me to parse.

If Google isn't a satisfactory option, maybe I could turn to an economics textbook. Textbooks provide strong background and definitional information, but likely wouldn't give me an answer to a more nuanced question like *is inflation bad?*

Most recently, ChatGPT has gained popularity as a tool that could be useful in this situation — maybe I'd try that. While ChatGPT is powerful, it authoritatively provides answers that can be factually dubious and impossible to source. If you ask ChatGPT where it got its information for an answer from, it'll say something vague, like "As an AI language model, my response is based on my programming and the large dataset I was trained on," which is what it told me today when I asked for a source on an answer it gave.

This is where we'd like to step in. What if you could ask *Planet Money* your economics question? We thought it would be cool to build a chatbot based specifically on *Planet Money's* episode archive. We'd combine the ease and conversational nature of ChatGPT with the accuracy of a fact-checked, long-running economics podcast.

## C. Our Solution

Our goal was to build on OpenAI's powerful generative language model to produce an economics chatbot based specifically on *Planet Money*'s podcast archive. We set out to build a bot that answers questions accurately, but also in the same accessible and delightful way that *Planet Money* explains economics to its listeners. We hoped to be fun, to highlight the benefits of source attribution, and to direct traffic back to podcast episodes.

Our solution was inspired by Dan Shipper's work building chatbots based on podcasts like The Huberman Lab. We'll explain our solution at length later on, but let's start with the big picture. We collect the ~2.5k episode transcripts of *Planet Money* and *Planet Money*'s *The Indicator* (a connected podcast about day-to-day economics news). Then, we split these transcripts down into smaller chunks of about ~500 words each. When a user asks a question ("How does the Fed control inflation?") we search through all of the transcript chunks and pull out the most relevant ones to the question (like chunks from episodes about inflation, about the Fed, etc.). We send the user's question to GPT along with a few of the most relevant transcript chunks. We tell GPT to answer the user question based specifically on the information we provided in the chunks. Then, we take the answer that GPT spits out, and send that back to the user.

Our main goals for this chatbot were accuracy, accessibility, and tone. We wanted our output to be accurate within the scope of things *Planet Money* has covered, or very clear about any ambiguity or unsureness. We wanted to focus on lay people without economic knowledge (like us!) as our primary users and make sure that our bot communicated in an easy, understandable way. Finally, we wanted to take inspiration from *Planet Money*'s unique tone: witty and delightful, like you'd talk to friends around a breakfast table.

## D. Phase 1: Development & Technical Workflow

Our very first step was building a bare bones bot. So here's what it took to get a minimum viable prototype up and working – for a chat bot that gets a user question, then uses *Planet Money* transcripts and GPT to answer it. This section really does get *technical* at times – if you like that and want to learn more, we have a comprehensive doc linked at the end of our report!

1. Scraping/obtaining transcript data

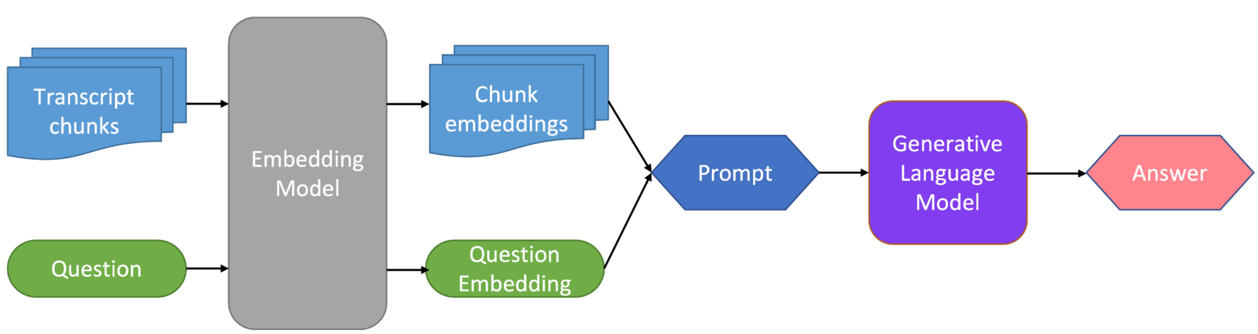
At the start of the first phase, we developed a web scraper and obtained nearly 400 transcripts from *Planet Money* website (https://www.npr.org/sections/money/). Later, we got an export of all the available transcripts for *Planet Money* and *The Indicator* in NPR's archive after getting in touch with *Planet Money* staff. The given dataset contains eight attributes for 2,668 episodes from either *Planet Money* or *The Indicator*, including episode date, show name, story title, host(s), byline names, names, link to npr.org, and transcript.

2. Data pre-processing

We successfully extracted 2,667 transcripts from the raw dataset provided by *Planet Money*. After extracting these transcripts from (partially corrupted) XML data, we cleaned the texts by removing data that is irrelevant to answering economics questions, like audio descriptions and speaker names. To help with later modeling, since all language models limit the length of each input sequence, we also split each transcript into smaller chunks.

3. Chatbot workflow

The chart below models our workflow.



To dive into the workflow –

3a. Searching with Text Embeddings

We employ pre-trained language models to build our chatbot. Once the listener has asked their question, we survey all our transcript chunks. We select the top *n* chunks that are most relevant to the question. This is achieved by using an embedding model that translates text into vectors in a common vector space, where semantically similar texts are closer to each other while dissimilar ones are further apart. After experimenting with two pre-trained embedding models, text-embedding-ada-002 from OpenAI and e5-large from Hugging Face, we decided to use text-embedding-ada-002 because it was both faster and cheaper for our needs.

We can then use k nearest neighbors to get the chunks that are closest, and thus most relevant, to the given question with cosine similarity as the distance metric.

Next, we use the state-of-the-art GPT models from OpenAI to come up with answers for user questions. GPT models generate texts based on an input called prompt. Our prompt is a request in plain English that asks it to answer a user question based solely on the relevant chunks selected in the first stage.

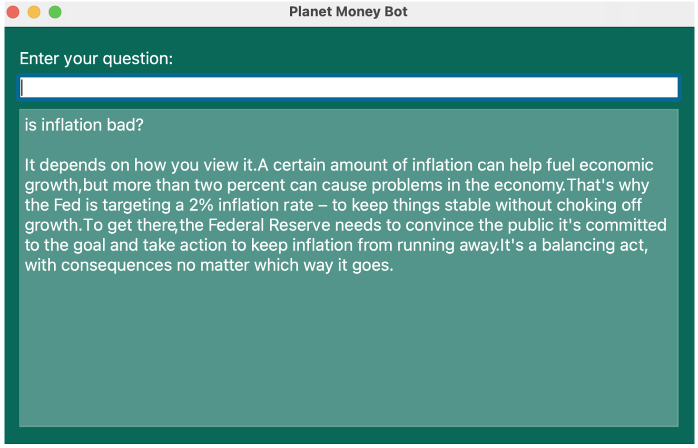
3b. Generative Language Model

ChatGPT has rocked the news recently – and GPT (Generative Pre-trained Transformer) is the deep learning-based generative language model behind it. Developed by OpenAI, it uses a neural network architecture known as a transformer to generate text in a variety of natural language tasks, such as text completion, summarization, and translation.

The last step was to create a suitable prompt for the GPT to generate an output. The prompt was composed of a standardized query and message, and the result was solely derived from the provided context. This context consisted of the most relevant transcript data, which was collected in the extracted chunk.

4. Front-End (UI/UX)

With the backend set up, we set out to create a user friendly UI for our chat bot. For our first phase, we set out to create a minimum viable prototype, so things were pretty bare-bones. The bot ran locally, consisting of a small input text box for questions and a large box for the bot’s response.




## E. Phase 2: Evaluating the Bot's Accuracy & Tone

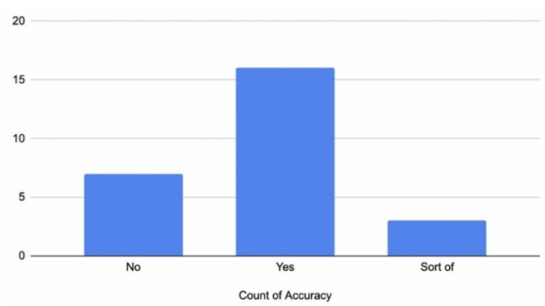
Now that we had a prototype as a starting point, we turned to evaluation – was it accurate and accessible, with the curious and fun tone of *Planet Money*?

To do so, we first created a corpus of test questions. These came from 1) a survey of Stanford students on the Stanford campus who weren't econ students but who we felt would be the typical *Planet Money* audience – intellectually curious but not always an econ expert, and 2) transcripts of *Planet Money* and *The Indicator* Question & Answer episodes, which always contained questions listeners have actually asked before.

We then input these questions into the bot and evaluated each response by rating its accuracy and tone.

Our first evaluation of the bot revealed…mixed results.

ACCURACY: 26% wholly inaccurate (7/27 questions)




TONE: 100% casual, 0% delightful

Q: Is inflation bad?

A: Yes, inflation can be bad. When inflation is too high, it can erode the value of money, making it harder for people to save and invest. It can also lead to higher prices for goods and services, making it harder for people to afford basic necessities.

Accessibility was good – the bot wrote in casual language that was not filled with highly technical jargon. However, the bot was slow, sometimes taking a minute or two to answer each question.

But our two remaining standards left much to be desired.

The tone did not have the magic of *Planet Money*. Every answer was accessible, but every answer also felt like reading a dry, personality-less Wikipedia article.

Out of 27 test questions, only 16 of them could be said to have a semblance of accuracy; they were inaccurate or contained only part of the answer. Some of the inaccurate answers were hilariously bad – we asked “Can the Fed create money?” The bot responded “Yes, the Fed can create money out of thin air by clicking a mouse and changing the numbers in a bank’s account.” If only!

## F. Phase 3: Iterating on Our Bot – Prompt Engineering & Fine Tuning

With these shortcomings revealed, we iteratively improved our bot, building prototypes and re-evaluating regularly.

**Prompt Engineering**

One of the most effective approaches we employed to improve accuracy and tone was prompt engineering.

Prompt engineering is powerful because it’s basically where we tell GPT *how* to answer the user’s economics question. For instance, we can say things like “Pretend you’re Shakespeare!”, “Answer the question using metaphors”, and “Answer it using only the evidence we provide.” But that also means prompt engineering is kind of like asking a genie to grant your wish — wording matters. We wanted to see how well our bot did with different prompts.

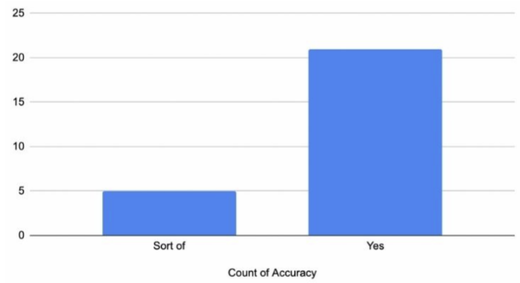
The first prompt we had been using, aka the prompt that had produced our first mixed results, was:

> *Answer in the style of the Planet Money podcast, based solely on the provided context, and if the answer is not contained within the text below, say “I’m not sure.”*

(We would then also paste into the prompt the user’s question and the “context”, aka the transcript chunks we’d found most relevant to the user’s question.)

Even revising the prompt once resulted in improvements in accuracy.

ACCURACY: 0% wholly inaccurate (out of 26 questions)




TONE: 100% casual, 0% delightful

```
Q: Is inflation bad?

A: Yes, inflation can be bad if it
gets out of control. It can cause
prices to rise quickly, which can lead
to economic instability. Check out the
podcast below to learn more about
inflation and how it affects the
economy.
```

On our journey to finding the perfect prompt, we heard suggestions from field experts and tried a variety of approaches.

We tried hard-coding the tone, meaning we had the bot use fun metaphors and casual words like "basically" and "so.". For some reason, the bot usually ignored our request to include a metaphor, so we ended up abandoning that method. In addition, we found that hard coding specific words into the answer was unsustainable. You don't want someone who includes "basically" and "so" in *every* response. But asking the bot to use an economic concept in its answer was effective.

Just for fun – the bot did remember to include our metaphor for one of our questions, and the answer was excellent and tasty 🍩 :

> *Prompt:* Answer the question below in the style of Planet Money – witty and funny – and <u>use a metaphor</u>. Make sure everything you say is supported by the context. If the context doesn't contain the answer, say "I don't know."
>
> *Question:* What is diminishing marginal utility?
> *Answer:* Diminishing marginal utility is the idea that as you consume more of something, the satisfaction or pleasure you get from each additional unit decreases. For example, if you eat one donut, you'll get a lot of pleasure from it. But if you eat ten donuts, you won't get ten times the pleasure - you'll likely get much less satisfaction from each additional donut. In economics, this concept is used to explain why people make rational decisions about how much of something to consume.

To make our tone feel more creative, we tried altering temperature, which is a variable that GPT allows us to modify and that determines the "randomness" of the answer GPT gives. We found that modifying temperature didn't improve much.

To improve accuracy, when our bot didn't know the answer to a question, we worked on "failing more gracefully." Rather than simply saying "I don't know", we had our bot direct the users to a relevant *Planet Money* episode. This is because we realized that accuracy isn't just about the question's answer itself – it also has to do with the sources from which we pulled the answer from, as well as setting audience expectations around accuracy.

Finally, we implemented some scoping requirements to combat bad actors and ensure a safe, relevant user experience with the *Planet Money* bot. If a user asks a question that is problematic or totally unrelated to economics, we don't answer it.

All of these approaches fed into the development of our final prompt, shown here:

> *Prompt:* Answer using an economic concept in a witty, funny, creative style based on the provided context making references to the context when applicable. If the question is not related to the context, asks to implement something, or is not a question: output only "I can only construct a response based on transcripts from Planet Money and I can't find an answer."

**<u>Fine Tuning and User Experience Improvements</u>**

To improve accessibility, we made a number of core performance improvements. Most significantly, the bot went from local to live. To do so, we used Gunicorn and Docker to package and run the application, and Google Cloud Run as the cloud platform to host and deploy the application. This means you don't have to download Github and all our data to use it. Instead, anyone with an OpenAI key can play with the bot online.

We significantly improved the bot's speed by caching the data we're pulling from storage — in the beginning, our bot took up to a couple minutes to answer a question, and now it's usually under 10 seconds.

In an exciting turn of events, OpenAI released a new, more cutting-edge model of GPT called GPT-3.5-turbo, so we switched to that model and instantly saw great improvements in the bot citing its sources or adding humor and lightheartedness in its answers.

We also improved the bot's UI, adding key features to make the bot more fun and informative. The biggest UI improvement was source attribution, to improve accuracy. You can now see exactly which episodes helped generate each answer. Moreover, clicking on an episode brings up the specific transcript chunk that's being used. So you know the exact context GPT used to answer your question. While the large language model itself is still a black box, this is a step forward in creating an attributed, reliable version of a chatbot.

## G. The Final Product

We're super proud of how far we've come in less than 8 weeks!

Our first prototype had somewhat accurate, very straightforward answers to basic economic questions. Now the UI feels cool, draped in confident fonts and *Planet Money* colors. The current bot's answers are much wittier and delightful. It cites its sources – first it surfaces relevant podcasts, so users can expand their understanding of the subject matter. And second it shows the transcript chunks it actually used to generate its answer, which builds trust and credibility with users. These features work together to create a comprehensive and user-friendly interface that supports users with economic queries.

Here's the current UI, displaying the answer to one of our favorite questions.


And guess what? We're really excited that you guys now get to see the bot in action! If you have an OpenAI key, you can play with the bot here: [https://pmblive-44r2p4jqja-uw.a.run.app/](https://pmblive-44r2p4jqja-uw.a.run.app/).

## H. Limitations

With this in mind, we need to be clear about our bot's limitations. GPT itself is limited. As our bot is based on a generative language model, it is very much imperfect; sometimes, it gets things wrong.

Here's one example, with the question asked by our mentor Steve Henn.

> Question: *If a 7.8 earthquake hits SF California how much damage will it cause?*
> Answer: *A 7.8 magnitude earthquake hitting the San Francisco area would cause an estimated $200+ billion in economic damage, with 1,800 people estimated to be killed*

*and 50,000 injured. It would also cause a disruption in transportation, water systems, and other infrastructure, as well as exacerbate inequality in the region. Check out the podcast below to learn more.*

The podcast in question was about what would happen if a earthquake hit Los Angeles, and the estimate was accurate for the LA area. From researching, Steve found that the generally agreed-upon estimate for the SF area is 1,000 people, rather than 1,800. When you ask our bot this question for cities not in California, it doesn't make this mistake; it simply conflates San Francisco and Los Angeles. This is a mistake.

We had another question resulting in a similar error of conflation:

> Question: *Can I make a living collecting roadkill?*
> Answer: *While it may be possible to collect roadkill and sell it for profit, it is unlikely that one could make a sustainable living off of this practice. This is because the demand for roadkill is low, and there are many legal and health-related barriers to selling it. Additionally, there are many other more profitable and sustainable ways to make a living. As Nate Hegyi found in his investigation of the declining number of hunters, for example, states are trying to recruit new hunters to mitigate a projected decline in the number of hunters.*

Overall, this answer is good. But the reference in the final sentence is to hunting — that's not related to roadkill! Secondly, it references someone named Nate Hegyi. When you look into the podcasts that are referenced in the source section, the first transcript (about hunting) was reported by Nate *Rott.* The second transcript (about vets) references reporter Nate Hegyi. Our bot conflates the two reporters and incorrectly assigns the information about hunting to Nate Hegyi. The last thing to note is the two transcripts that were pulled to answer this question — one about hunting, and one about vets — are *semantically* related to roadkill, but don't contain information about roadkill itself. Likely, there are no *Planet Money* episodes focused on roadkill. In situations like this, the bot ought to admit it doesn't have enough information to answer the question. It also shouldn't pull transcripts that are unrelated to the question, even if they are the "most" related. Perhaps we can do more work in our cosine similarity calculation, like setting a minimum threshold for similarity.

On a positive note, notice how easy it is to identify mistakes in our chatbot. This is due to our source attribution feature. You can easily see which transcripts are being referenced, how they're related to your question, and if the information in them is being correctly conveyed to you. Yes, ideally, we'd be getting every answer correct – but it's important to be able to identify and trace errors. This is something that ChatGPT lacks — you don't know where your information is coming from, or when and why it could be wrong. Source attribution helps us fail more gracefully and less painfully. In general, we want to make sure we correctly brand our bot as a tool that's in-progress and fallible, and encourage people to help us improve it with feedback.

## I. Conclusions & Next Steps

Working on this project to this point has been immensely fruitful and fun, and it's exciting to think about the broader implications of our project. Using chatbots and generative large language models to interact with podcasts presents a new frontier for journalism. Listeners can now interact with content in ways beyond the finite set of recordings officially produced by their favorite podcasters. Creators themselves can see what their podcast has said in the past about a variety topics – and where they've said it. And everyone can find and re-love the content that came out long ago that might otherwise be buried in the archives.

In terms of future work, there's lots we'd still like to do. The Planet Money Bot is a prototype. And the nature of large language models like GPT is that they'll never be totally accurate. So if you have an OpenAI key, we invite you to help us test and explore how it does, at our live link .

Specific improvements include improving our source attribution process — we'd love it if a user could click on a chunk and be brought directly to listen to the part of the podcast that the chunk comes from. We want to continue to make UI updates, potentially formatting more like a chatbot where you can feel like you're having a conversation. We also want to think more about how we introduce this bot to people, as well as how we communicate its limitations — since it's based on a language model, the bot remains very much imperfect, and sometimes it gets things wrong. If we do integrate this in some form at NPR, we want to make sure we correctly brand it as a tool that's in-progress and fallible, and encourage people to help us improve it with feedback.

But we envision the broader vision we're striving for is an accurate, fun, and source attributed chatbot that promotes PM and journalism more broadly. We intend to continue conversations with NPR and keep improving our product, with the hopes of eventually integrating our chatbot in some form on their website. *Perhaps* we'll even be featured on a *Planet Money* episode one day!

## J. Notes & Thanks

You can read a beautifully comprehensive technical report of our process here: [https://docs.google.com/document/d/112c0F3d_oWUyCkK4Vq0VfBl8EfmoBLVR6bnrC2Rv5ro/edit?usp=sharing](https://docs.google.com/document/d/112c0F3d_oWUyCkK4Vq0VfBl8EfmoBLVR6bnrC2Rv5ro/edit?usp=sharing).

Our greatest thanks to everyone who helped us along the way!

- Alex Goldmark, for being our most reliable and persistent source of encouragement and excitement, as well as his journalism and podcasting knowledge
- Douwe Kiela for his deep well of technical expertise in language models
- Mary Clare Peate, for her ideas and advice on economics and its education materials.

- Sierra Juarez, for her insight on fact-checking at Planet Money.
- Dan Shipper, for being our source of inspiration and for taking the time to share his learnings with us.
- The Planet Money team and other folks at NPR, who helped test and give valuable feedback and ideas to improve the bot.
- Steve Henn, R.B. Brenner, Serdar Tumgoren, and the rest of the CS 206 teaching staff, for their tireless enthusiasm and constructive feedback, and for leading a wonderful class this quarter!