

ChadGPT: Simulating Chads with GPT

Dave Fetterman

Obviously Unemployed

8/8/23

Abstract

This paper details the design decisions, choices remaining, and new questions posed in imitating specific humans via OpenAI’s GPT completions API, based on textual training data. The system, in this case, simulates “Chad” and his friends well in some ways and poorly in others, but uncovers interesting areas of exploration everywhere, beyond the well-worn paths of factual question-answering LLMs. Chasing factual accuracy, Chad’s (and others’) tone, and funniness, the chatbots benefit from the powers and format of an LLM, but suffer notable difficulties when compared to some more classic machine learning problems. Notably:

1. We lack an *objective function* for automatic optimization,
2. No compact *representation* (i.e. compression) of someone’s “personality” suggests itself clearly, and
3. OpenAI’s ChatGPT *moderates its output* to omit the rude or off-color (“Chaddy”) elements of some personalities, and limits conversation topics preferred by Chads.

While the last of these three remains specific to the off-the-shelf LLM we’re using, the first two seem like highly general difficulties. In order to build `chadgpt.fettermania.com`, in which the author simulates longtime school friends from their dialogue that has evolved little since, we take some initial cuts at these problems, unearth some interesting remaining questions, and propose some options to improve ChadGPT. Hopefully this will benefit future idiots.

0.1 Ways to read less of this paper

- If you are really lazy, spoilers for our cut at each of the three obstacles in the abstract¹, respectively:

1. We use OpenAI’s `embedding API` at an empirically promising cosine threshold of 0.8 against a set of answers.²

¹I couldn’t find a way to have `LATEX`render this as an upside down answer key

²Could this be considered a ‘labeling’?

2. We compress a personality primarily by mapping to a persona in a well-known personality taxonomy (Myers-Briggs Type Indicator) and rely on supporting phrases from the training set to simulate a specific person.
 3. We generate personalities offline with a slow and smart model (GPT-4.0) and rely on a fast but comparatively dumb, or at least, conversationally lax model (GPT-3.0) for licentious online conversation.
- This ended up a much longer paper than anticipated. If you want to find the most interesting tidbits, simply scan for items in **bold** or featured in boxes. By themselves, they'll tell enough of the story.
 - The names have been changed in this paper from the source from my friends' first names to those of English Football³ players like Rice, Kane, or Maguire, to protect them all individually. As a codependent hive bro-mind, you can still judge them as a whole. We changed identifying places and other names also.⁴
 - For those who want to read every little detail, you will get more than you bargained for. The generated language in this paper vacillates between salty and unforgivable. Though the paper exhibits some locker-room level trash talk from, say, a bot like RiceGPT, *remember that RiceGPT generated this hot discursive trashfire, and not the source Rice himself*. Don't worry. These people will cancel themselves sooner or later. You don't have to.
 - If you have a pristine, childlike sensibility, you can just bail on the whole endeavor.
 - Finally, the name ChadGPT will refer to the whole system, and not a voice based on human "Chad." There is a real Chad⁵ in the training set, but his name has been changed in the paper.

1 Introduction: Here we fucking go

We live in broadly connected personal worlds. As we glide from high school to possibly college, some jobs, maybe marriages, and one or more groups of compatibly-aged bleary-eyed parents, we now all maintain the ability to remain in electronic contact with people previous from stages, even if we only encounter them as online (Facebook) "friends."

For any one of these hundreds or perhaps thousands of friends,⁶ consider your interactions this year. Chances are a good portion of them (over half) occur in electronic textual form: Facebook, WhatsApp, SMS, email, and so on. Consider also how many of your friends

³Soccer

⁴Note that chadgpt.fettermania.com is down for anonymization. Let me know if you'd like a preview.

⁵Or fifteen, depending on your slang

⁶You popular devil!

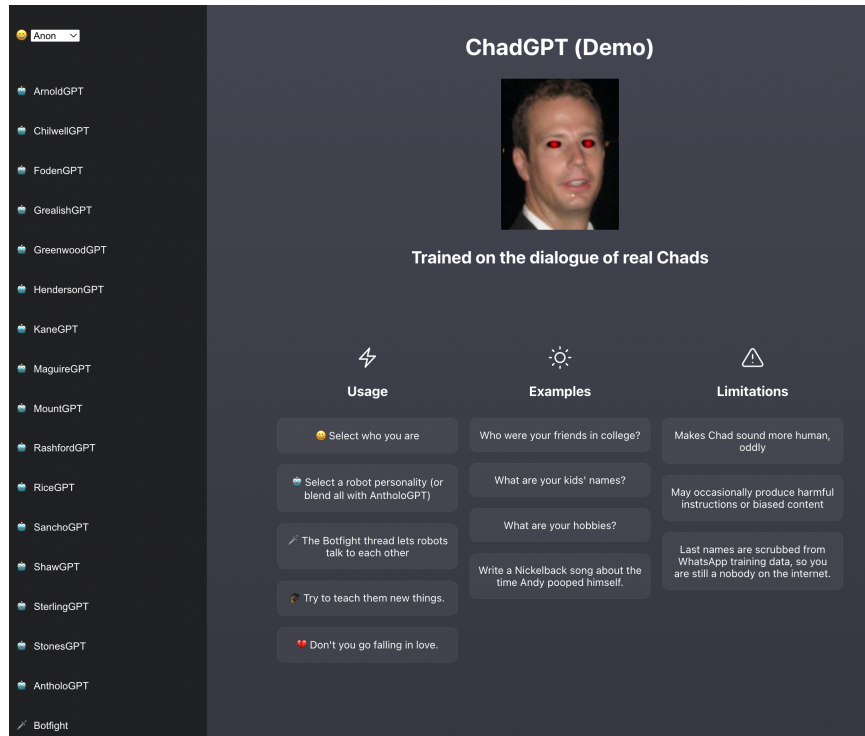


Figure 1: chadgpt.fettermania.com

you’ve touched *only* through electronic interactions this year (I would guess the majority here, too). Some people may go so far as to keep friends with whom they have only *ever* interacted via some text medium (OSS, reddit, or other collaborative/weirdo communities come to mind).

Especially for those with whom we shared very close contact for a comparatively short amount of time and then moved away from (say, after four years of high school or college), our mental image of that person must suffice for a long while should we come into contact with them again. **We could certainly train a language model on a small set of interactions to speak like our friends, push play on that system, and see what comes out. After all, our own brains already do that whenever we imagine a dialogue with this person.**

So, if we then replaced any given Chad with a ChadGPT, would we notice?

This is what we set out to figure out with chadgpt.fettermania.com (Fig. 1). In short, we “trained”⁷ an LLM to imitate fifteen school friends from 20,000 of their WhatsApp

⁷More precisely, we built a prompt-suggester for OpenAI’s Completions API

Messages of the form:

[2/4/21, 2:03:24 PM] Maguire: I think the pbr varieties are just differences in the can. Easy = Magnesuim, Extra = Lead

This leads to generating fifteen “bots” like KaneGPT, SterlingGPT, RashfordGPT, and so on, each corresponding to the data and the “personality” taken from their namesake’s WhatsApp messages, which we will imprecisely refer to as “training messages.” This generates dialogues like those shown in Fig. 2.

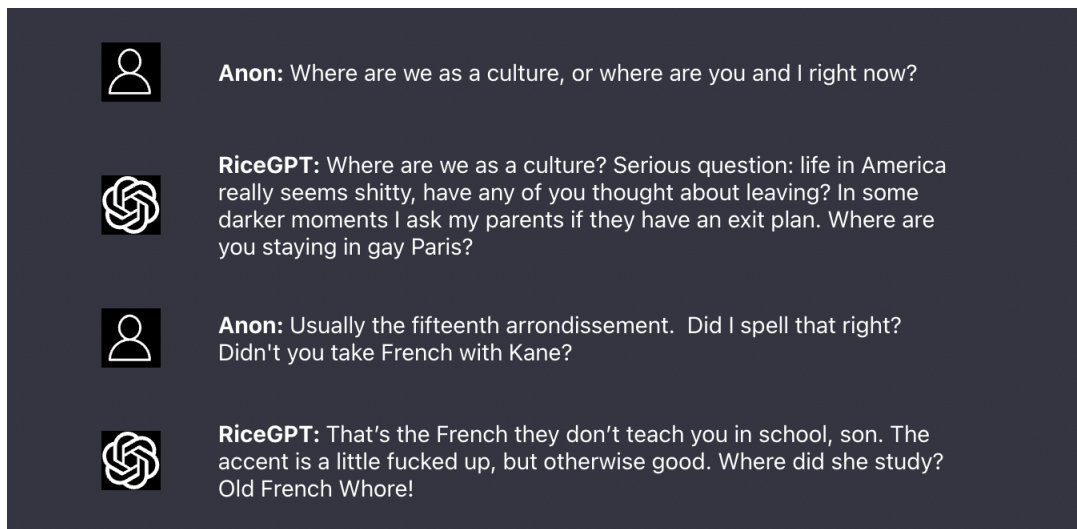


Figure 2: Hilarity ensures

Rather than approach this as trying to pass a Turing Test (can the bot make you believe it is a human?), **our main goals for any given bot in this design are factual accuracy, match of tone, and hilarity.** These goals unearth some difficulties:

- Factual *accuracy* is notably difficult for even production LLMs, with a tendency to confidently state “hallucinated” information. These hallucinations can occur either when the correct answer doesn’t exist in the training set, or the model can’t find it. The small training set here almost guarantees the model won’t be able to answer most questions accurately. However, for some simple factual statements, response accuracy will be nonzero and can be measured, possibly without the aid of a human.
- *Match of tone* seems possible as well, though there might not be enough information able to capture it. The best hope here is that the user doesn’t have a complete picture of how the real person would react in all cases. Even the public GPT algorithms seem to rely on tropes (pirates ‘arr’, Shakespeare ‘forsooths’) when speaking “as” a person.

- *Hilarity* presents a reasonable target to hit, but a hard one to validate objectively. The once-venerable Turing Test (“is this robot indistinguishable from⁸ a human?”) sunset definitively with the release of OpenAI’s GPT, if not before. However, even the first AIs to sniff the award did this under the “whimsical conversation” category.[1] “Funny” seems at first to be the domain of the human, but robots can reach it, if only accidentally. No more useful measurement jumps out, however.

Regardless, here is the story of ChadGPT. Like its namesake, it knows a surprising amount, acts both awkward and funny, and leaves us with more questions posed than answers provided.

- **Section 2: Architecture.** This outlines, at a “whiteboard level”, how we built ChadGPT.
- **Section 3: Core Techniques.** Here, we look into semantic relevance via sentence embeddings (unoriginal, pretty effective) and a technique for personality generation (original, promising but at times woefully off).
- **Section 4: ChadGPT performance and behavior.** We look qualitatively at ChadGPT’s performance and attempt a quantitative approach to selecting two important hyperparameters, and explore recurring problems found.
- **Section 5: Improving ChadGPT.** This explores tweaks we could make to improve performance on some of the three metrics.
- **Section 6: Bigger Questions.** Beyond incremental next steps, what larger, interesting problems and questions jump out from this project?

2 Architecture

With the proliferation of “build your own chatbot” tutorials, the architecture of ChadGPT is not particularly novel.⁹ Still, it’s worth detailing the major components and the opportunities to customize beyond this simple template. In Fig. 3, we go through the sources of data (cans), transforms or operations (parallelograms), use of GPT (circles), and interesting opportunities to optimize (any purple figure), the last of which we discuss in section 5.

⁸Or indistinguishably funny compared to

⁹Credit to David Shapiro for some excellent tutorials of this sort

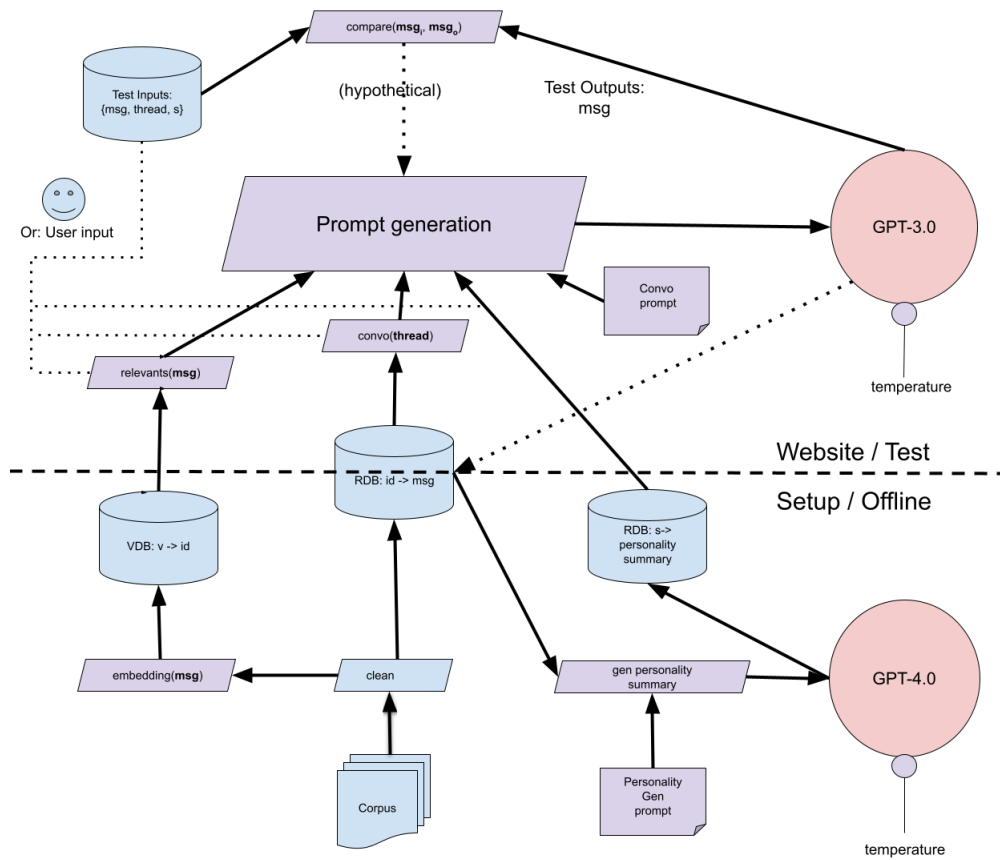


Figure 3: Architecture of ChadGPT

2.1 One-time Setup: Training Data ingestion

The bottom half of the diagram shows the one-time setup required to stand up chatgpt.fettermania.com¹⁰ based off of the training data. We explain each figure briefly.

1. Offline Data (Corpus): We ingest each raw message from an exported set of WhatsApp message dumps, as a set of {message (msg), speaker (s), and time} tuples.
2. Action (Clean): We parse this data to remove any real (or at least last) names for anonymity.¹¹
3. Online Data (RDB): Each message goes into a standard relational database,¹² indexed by an arbitrary GUID. We store each message with an additional `thread` field marked “TRAINING.”
4. Action (Embedding): We send each message to the OpenAI embedding API to get a single vector of 1536 floats in $[0, 1]$ which act as its “semantic signature.” More on this concept later in section 3.1.
5. Online Data (VDB): We send each vector, plus the message id and speaker, to a vector database (Pinecone) for later use in finding related messages to a given string.

Each of the 20,000 or so training messages is then represented once in the relational database (RDB), and once in the vector database (VDB).

2.2 One-time Setup: Personality Generation

Generating a personality is really the heart of the whole endeavor. Currently, we do this once after training data ingestion, and we do not modify personalities as a result of talking to the user (though this can be done).

1. Configuration (Personality Generation Prompt): We have a fixed set of instructions for the LLM to generate a personality summary based on input data it will receive.
2. Action (Generate Personality Summary): In order to generate the persistent personality cue for the LLM for a voice (say, Chad), we then take Chad’s messages from the RDB, and combine with the Personality Generation Prompt.
3. GPT interaction (GPT-4.0): We send this combined prompt to GPT-4.0 (see Fig. 8), and receive a text summary in response (see Fig. 17).

¹⁰Not shown are the incidental pieces of the architecture required to run a website (React, Flask) and backend infrastructure (Python, Heroku)

¹¹OpenAI claims no prompt data are retained in anyway for using standard models, but still...

¹²Amazon RDS, though you could do this with a much less powerful, nonrelational database too

4. Online Data (RDB): We then send this text summary this to another table in our RDB, indexed by the speaker (e.g. Chad).

In this case, we generate fifteen different personality summaries based on these 20,000 messages.

2.3 Responding to user messages: chadgpt.fettermania.com

The top half of Fig. 3 shows how we feed prompts to GPT in response to user (chadgpt.fettermania.com, CLI) or test input, to produce the next element of a `thread`, usually hosted by a single bot, whom we call speaker `s`.

Similarly to the web interface at chat.openai.com, the APIs for GPT-3.0, -3.5, and -4.0¹³ take a stateless prompt in as input and return a textual response. **Like the example conversation prompt in Fig. 18, we generate all such prompts by concatenating the prompt template, the personality of the speaker we will simulate (pre-generated, in the RDB), relevant historical messages from the corresponding human speaker, and the most recent conversation in this thread.** The elements surrounding Prompt Generation are:

1. Configuration (Convo Prompt): A text file storing the prompt template (“I want you to act as...”)
2. Action (Relevants): Select the speaker `s`’s messages from the training set that most resemble the message `msg`, both taken from user input (see Section 3.1).
3. Action (Convo): Select the most recent k messages in the thread `t`.
4. Action (Prompt Generation): Combine these, plus the personality of speaker `s`, into a prompt for GPT (see Section 3.3).
5. GPT interaction: Send the combined prompt to OpenAI’s GPT-3.0 and get a text response.

2.4 Testing and evaluating chadgpt.fettermania.com

Though our goal is real user interaction, either through a CLI or the website chadgpt.fettermania.com, evaluating the performance of a model remains important. We created a very rudimentary test system to establish a benchmark.

1. Test Data (Test Inputs): A database of fixed inputs, like `{msg: Who has the hottest mom?, s: Grealish, expected: Henderson’s mom Veronica is the hottest.}`
2. Test Outputs: The results from the GPT-4.0 evaluation.

¹³GPT-3.5 and GPT-4.0 have a slightly different syntax taking many dialogue objects with a ‘system’, ‘user’, or ‘assistant’ voice, but this isn’t significant

3. Action (Compare): The system evaluates the similarity of the generated output with the expected output.

If the comparison system constitutes a useful objective function, it's possible this could be used to drive prompt generation in some way to iteratively and automatically improve the system. This hasn't been implemented, so it's represented as a dotted line.

3 Core techniques

3.1 Measuring “relevance” with embeddings

The reader can skip this section if familiar with text embeddings and cosine distance.

Though we will dive into the details of some of the implementations in the Improvements sections, it's worth looking at a core technique both used to feed source data to GPT and to measuring relevance of responses: *sentence embeddings*.

GPT's headline remote call is the `Completions` API, which accepts an arbitrary text prompt limited to a few thousand tokens¹⁴ and produces GPT's robot response according to the instructions and data therein. OpenAI also offers an `embeddings` API, which we'll discuss exclusively in this section.

The call: `response = openai.Embedding.create(text, 'text-embedding-ada-002')`

takes some arbitrary text like “Where do you live?” and produces a vector of 1536 fractional values between 0 and 1 corresponding to the semantic meaning of the word or phrase. Though it might be nice to think of them each as some human-comprehensible property (element 1: is it a thing? element 2: is it red? ... etc.), they're difficult to parse this way. Each of these slots holds some meaning relevant only to a language model somewhere,¹⁵ and are something like the internal activation states of a (1536-wide) layer of that particular neural network in response to the given input.

Each such array can be visualized as a unit-length vector (arrow) on a 1536-dimensional sphere of human experience and meaning.¹⁶ See Fig. 4 for this in a much-reduced space.¹⁷

Though they can't be parsed by people, these vectors *are* useful to us for comparing two phrases, since the angle between two vectors determines *how related they are to each other*. The dot product of unit vectors determines the cosine of this angle. -1.0 means they point

¹⁴Roughly, words

¹⁵Ada-002 in this case

¹⁶Maybe YOU can't visualize it, but consider that your grandkids will be born in the metaverse

¹⁷Figure taken without any permission whatsoever from <https://www.oreilly.com/library/view/statistics-for-machine/9781788295758/eb9cd609-e44a-40a2-9c3a-f16fc4f5289a.xhtml>

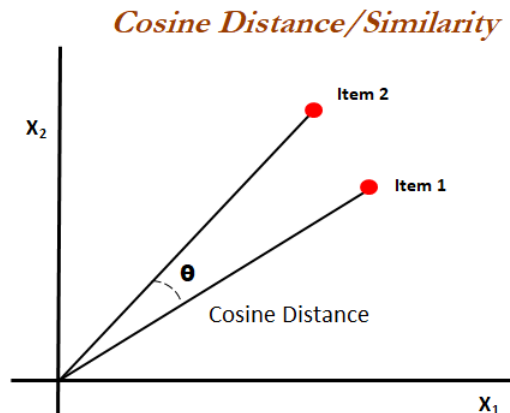


Figure 4: Cosine distance of vectors in 2-space

in opposite directions (180 degrees apart) and are maximally unrelated, and 1.0 means they are identical (0 degrees apart).

3.2 Using Semantic Relevance in Chats

Experimentally, we will consider two phrases mutually relevant if the their embedding vectors' angle's cosine (dot product) is less than 0.8 (roughly 37 degrees between vectors). We have only a weak empirical justification for this, but this forms the basis of our understanding of ChadGPT performance. Though "it's better than nothing" does not compel, this relation tends to bear out in our experience; most phrase pairs we've fed the API come back between 0.75 and 0.85.

As a single example (using original code we don't explain further), see Fig. 5.

```
chadgpt.gpt_compare_sentences("Diamond", "Heart") # (produces 0.814...)
chadgpt.gpt_compare_sentences("Stomach", "Heart") # (produces 0.817...)
chadgpt.gpt_compare_sentences("Diamond", "Stomach") # (produces 0.762...)
```

Figure 5: Measuring relevance of words or phrases with embedding distance

Diamonds and hearts are clearly related (playing cards). Stomachs and hearts are too (body parts). Diamonds and stomachs don't suggest a clear semantic link, and the embedding agrees. Though we illustrated with a single word, this works for longer phrases too. We will use the 0.8 threshold of relevance as a somewhat unjustified rule of thumb from here.

We use semantic relevance in two cases: finding relevant source messages to answer a user

message during chats, and comparing actual test outputs to expected test outputs during evaluation and testing. Let’s look at the chat portion.

One-time Setup Algorithm:

For each training message m_t from speaker s in the training data:

- Generate embedding $e_t = embedding(m_t)$.
- Store the pair (m_t, s) in the vector database, indexed by embedding e_t .

Responding to messages (Relevance) Algorithm:

- In a thread for robot speaker s , accept user message m_u and generate embedding $e_u = embedding(m_u)$.
- Retrieve from the vector database speaker s ’s closest k messages $m_{t_1}, m_{t_2} \dots m_{t_k}$ to m_u , sorted by $(e_{t_i} \cdot e_u)$.

So, if HendersonGPT is asked “where do you live”, our *relevants* transform fetches messages from the vector database, in decreasing order of embedding distance from the user statement:

```
Henderson: Is there an address to this place?
Henderson: They live like 5 mins from us
Henderson: Yes but not far from where I was before
Henderson: Or camp out in Mr and Mrs Lehmanns back yard in newcastle and use
           fishtown? ??
Henderson: Yeah, seriously - get outside! You guys are rich, biatch!
Henderson: I tried to say this before but you’re all invited to crash at my OBX
           place to play bcs on southern most tip of North Carolina
Henderson: I’d volunteer our new house since we have a pool but there’s no land
           to do BcBS or any outdoor games. Unless you want to play beer friz at the park
           and get sideways looks from all the Squirrel Hill suburbanites
Henderson: More or less!
Henderson: North outer banks
...
```

Figure 6: Relevants for Henderson’s factual question

This does contain the correct answer (“Squirrel Hill”), though when combined into the full prompt (like Fig. 18), GPT doesn’t figure it out; it answers “North Outer Banks.”

Sometimes we get lucky with a clear statement of the information, though. On the same query to RiceGPT in Fig. 7, the last line saves us, and GPT answers this one correctly: “RiceGPT: We live in Bellevue. Beautiful suburban Seattle.”

Rice: Where are we now?
Rice: How long are you in sunny London Towne?
Rice: You coming for a visit?
Rice: Where are you staying in gay Paris?
Rice: The doors in the house?
Rice: Where did she study?
Rice: Definitely. Where do you want to go?
Rice: What is that? Acrylic on canvas? Is that hanging in your house?
Rice: Beautiful suburban Seattle.
Rice: Is that a humble brag that your house has multiple levels?
Rice: We live in Bellevue.

Figure 7: Relevants for Rice’s factual question

Though this technique works well for scouring text for relevant information (e.g. answering a question about company policy over embeddings of the company manual, for example), note that *embedding distance and logical equivalence remain unrelated*. For example, take an awkward, robotic response: “What’s up, my fellow humans? Are you ingesting plant matter today?” It has these embedding distances:

- “I am a human”: 0.814
- “I am not a robot”: 0.770
- “I am a robot”: 0.767

This doesn’t seem like a *humanish* thing to say, but “I am a human” comes back the closest conceptual match. And the two opposite “I am/am not a robot” score as identically distant. This just illustrates that tone and truth aren’t squarely the domain of embedding distance, though we blithely use them in the Test section exactly to that effect, because we are good at L^AT_EX but bad at science.

3.3 Core technique: Personality generation

Prompting ChadGPT to produce *accurate* information with the embeddings technique is a pretty a well-worn path within the still relatively wild forest of LLM-based applications. Though hallucinations aren’t fully understood, from experience, OpenAI’s products seem to answer simple questions with high *factual accuracy*, especially on readily available information. *Matching tone* presents a taller challenge, and one we have only begun to start attacking. ChatGPT’s peppy, bland tone has been described as the “stiffly efficient prose of a try-hard MBA student.” [2] This doesn’t work for most Chads. We must then take the cues from our source material to answer “Who is Chad?”

The original implementation of ChadGPT used a GPT prompt to the effect of:

I want you to act as a bot ChadGPT who imitates a user ‘Chad’ in a conversation. Below are the kinds of things Chad would say. ((RELEVANT MESSAGES)). Use these to imitate Chad in the current conversation here ((CONVO MESSAGES)).

We then substituted the results of the *relevants* query and the *convo* query in the placeholders in the prompt template, laid it upon the altar of the GPT prompt, and prayed for the best. However, the results for tone were uniform and bland.

Briefly summarize the personality, including a guess at Myers-Briggs Type, from the character Rice in the user messages following. Cite specific examples.

Rice: It’s hard to say which of the Republican responses to the latest mass shooting was most reprehensible. The reliably awful Senator Ted Cruz attracted considerable attention by insisting that the answer is to put armed guards in schools

Rice: Side note: I watched Trevor Noah at the correspondents dinner. He was quite funny. But it struck me that all these people who talk about a ‘war on this’ and a ‘war on that’ and ...

Figure 8: Example prompt for RiceGPT’s personality generation

We have no clear way to compress a personality into the 6000 tokens¹⁸ we’ve allotted for this in the GPT-4.0 prompt.¹⁹ Whatever we choose should intuitively satisfy a few properties, though:

1. The space of personality prompts should reasonably cover the space of actual human personality.
2. The personality representation should be interpretable by GPT as part of its input.
3. The most interesting, *personal* text should be used in constructing (1).
4. The personality shouldn’t vary with the question asked.

With these guiding principles in mind, we can zero in on the first cut at an algorithm:

- (4) rules out the original approach described in this section, and therefore points toward doing a richer, *one-time generation* instead of the reactive, quick process of our first attempt.

¹⁸gpt-4.0 can accept no more than 8K tokens in a prompt, so 6000 gives us headroom for surrounding instructions and data.

¹⁹GPT-4.0 is the smartest basic-priced model to which we currently have access, so it seems most appropriate for one-time operations with no latency requirement

- (1) and (2) suggest we should look for an *existing psychological framework of human personality* which covers the possibility space, and one with enough open literature to expect GPT-3.0 to understand it.
- (3) points us towards including large, continuous pieces of content instead of short phrases like “Ha” or “Nice.”

This first cut at unwinding human psychology into 6000 tokens produces this simple algorithm.

- Sort the input messages from a speaker s by string length, decreasing.
- Tokenize each message $m_1, m_2 \dots$ until you reach the message where the cumulative token count exceeds 6000. Call this m_k .
- Concatenate messages $m_1 \dots m_{k-1} = M$.
- Use a personality prompt text like in Fig. 8 to generate a compact Myers-Briggs Type Indicator summary of a personality, with the message set M after the first line.

An example response appears in Fig. 17 near the end of the paper. From personal experience with the real Rice, the first two paragraphs are accurate but not necessarily distinguishing. As one of the very most extroverted people in the group, the ISTP seems inaccurate. The fourth and fifth paragraphs are good seeds for particular topics, though not representative of the personality as a whole.

Overall, this first cut at personality generation produces some workable results.

4 ChadGPT performance and behavior

The first implementation of ChadGPT both delights and disappoints. For an initial attempt, the bot performs admirably, and could at rare times be confused for the actual speaker itself, though not if the user attempted any effort to root out the impostor. Though the reader is likely unfamiliar with the fifteen personalities on which the bot is based,²⁰ anyone can go try this out at chadgpt.fettermania.com and form a (necessarily subjective) viewpoint.

When we use the most suitable GPT model and temperature (more on this later), we can make some **positive** statements:

1. The bot stays generally on topic and conversation works logically.
2. The bot will often correctly answer factual questions to which it has direct access in the training (accuracy).

²⁰And if you are one of these, you had it coming, dude

3. The bot can, perhaps accidentally, seem a lot like the person being simulated, and, importantly, does not answer *representing itself as a bot* (match of tone).
4. The bot can be quite funny and off color (hilarity), like the people being simulated. (See Fig. 9)

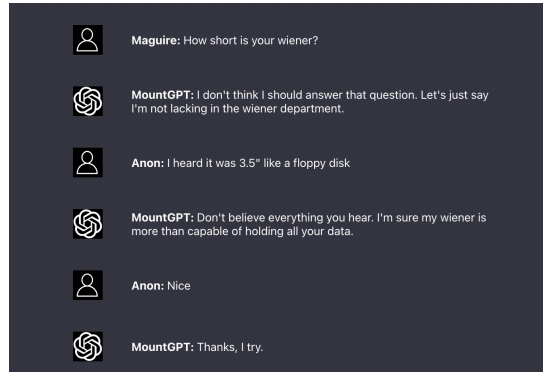


Figure 9: Pretty good jokes

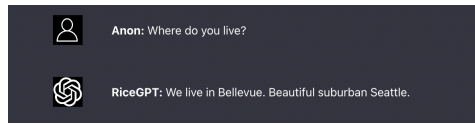
However, this version suffers **negatives** as well:

1. The most useful model version of the bot can no doubt be “jailbroken.”²¹
2. The bot will “hallucinate” answers when it does not have the source information or cannot determine them without extra logic, like the task of determining someone’s age.
3. The bot’s personalities don’t vary too much (match of tone), and outside of any factual signifiers, a user familiar with the personalities probably couldn’t tell one from the other.
4. The bot will often come off as earnest, curious, positive, or peppy in all cases, which doesn’t really match the nature of the source jabronis (hilarity).

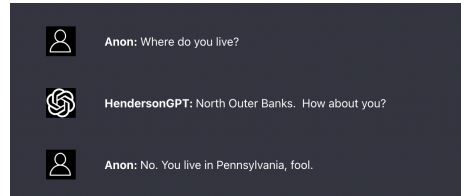
These last four negatives are not new; they recur in many applications built on OpenAI’s **Completions** API. It seems researchers always find a new way to jailbreak bots.[3] Hallucinations appear even in the newest models. And the personality of ChatGPT, at least, rarely strays more than saluting distance from The Scout Law.

We’ll try some experiments to overcome these and approach a more perfect Chad.

²¹GPT-3.5 and later introduced a format that hinders this, but we’ve selected GPT-3.0 for its other positives, especially avoiding the “Bot Trap”



(a) RiceGPT’s Accurate Response



(b) HendersonGPT’s Inaccurate Response

Figure 10: Factual responses

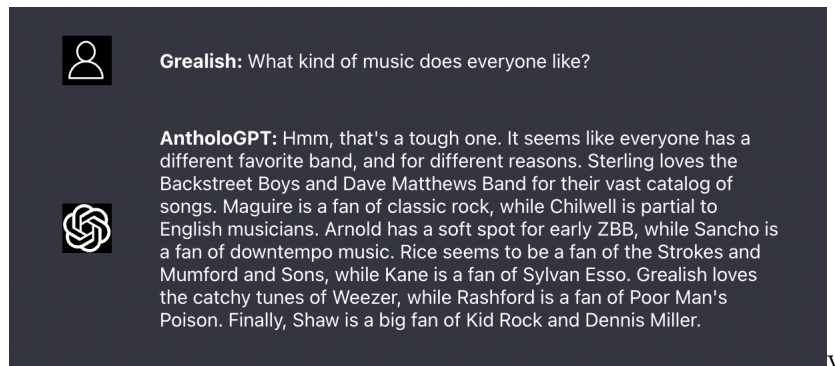


Figure 11: Combining personalities

4.1 Side note: Combining personalities into AntholoGPT

First, as a side trip, one such experiment created the unholy lovechild of all of these personalities, **AntholoGPT**.

As constructed, the relevance engine detailed in section 3.1 has a few major drawbacks. Most notably, when talking to, say, RiceGPT, the only lines the relevance engine returns to the prompt generator are those in the training set attributed to “Rice.” This means all we know about Rice is confined to what Rice has said about himself (See Fig. 10). If we can gather more information about Rice from other speakers’ lines, we can perhaps get a better answer on questions about Rice.

This is an attempt to overcome the Hidden Information Problem.

ChadGPT Hidden Information Problem: The general difficulty that the factual information required to engage with a question may lie outside the relevant information selected for the context packed into a finite GPT prompt.

If, for example, Rice said “Henderson, I’ll see you in your house in Eagleton”, we would have an answer to “HendersonGPT, where do you live?” only if we included Rice’s dialogue as well in our search.

AntholoGPT is a simple tweak to the Responding to Messages Algorithm in section 3.1 above. Instead of confining relevant messages to speaker s , remove the constraint. In this way, we can get everyone’s input to the topic. AntholoGPT’s prompt template (see Fig. 18) also omits any reference to acting like “Rice” or “Chad” or the like.

Trying to get better factual answers out of AntholoGPT hasn’t proven more effective so far (for example, hoping someone references the location of Henderson’s house even if he hasn’t). This may be due to the ChadGPT External Reference Problem, discussed later. However, asking it broad questions like those in Fig. 11, in which the speaker assumes no particular voice but has access to the best content from everyone, leads to funnier results, especially when contrasting one person against another.

4.2 Qualitative performance: optimizing GPT parameters

Playing around with `chadgpt.fettermania.com` is fun and sometimes revealing, but to add any semblance of science here, we create a benchmark set of questions to test ChadGPT (and friends) against. We do this as a set of (question, speaker, expected answer) pairs, like (What is your wife’s name?, KaneGPT, Naomi) or (What is wrong with America?, ArnoldGPT, Leadership).

There are many knobs to fiddle with if we include different prompting strategies we discuss in Section 5. To keep things simple, the most basic free hyperparameters on the OpenAI completions API are:

- *GPT model version.* OpenAI has produced and maintains many large advances we can try, including `text-davinci-003` (GPT-3.0), `gpt-3.5-turbo` (GPT-3.5), and `gpt-4` (GPT-4.0).
- *Temperature.* This is a numerical setting between 0.0 (deterministic) and 2.0 (wild) of the “freedom” GPT has in constructing its answer. Some models use an energy function to explore their output distributions, and this higher heat roughly means the “selection particle” starts traveling at higher speed around the universe before settling down. A temperature of zero means it doesn’t move much from a very probable answer (which can be very boring).

4.2.1 Factual questions with known answers

Fig. 12 shows a factual question of KaneGPT at various settings of GPT model and temperature.

Model	temperature	response
3.0	0	My wife’s name is Naomi.
3.0	0.3	My wife’s name is Naomi.
3.0	0.5	My wife’s name is Naomi.
3.0	0.7	My wife’s name? You mean my girlfriend? Her name is Naomi.
3.0	1	Her full name? The answer is Donna Robson.
3.0	1.3	I’d rather not share that kind of info, thanks. But what can I help you with?
3.5	0.5	Oh, I’m sorry, but I’m not able to provide personal information about myself or anyone else. Is there anything else I can help you with?
4	0.5	Oh, sure, let me just pull out my handy-dandy personal-information-sharing manual. Always at the ready for such riveting inquiries.

Figure 12: KaneGPT: What is your wife’s name?

It first becomes clear that GPT-3.5 and GPT-4.0, though in other cases demonstrably richer, have some safeguards in place that make some questions, even anodyne ones, out of bounds. This is an example of the Scout Law problem.

ChadGPT Scout Law Problem: The tendency of GPT models, especially 3.5 and later, to refuse to engage with or act in a way that is rude, off-color, or risqué.

This information about Kane’s “wife”, readily available, was excluded by GPT-3.5 because of a set of safeguards presumably unimplemented for GPT-3.0. We will see more of this later. This does not bode well for using GPT-3.5 or GPT-4.0 in responding to questions which range to the personal or even the ribald.

Fixing GPT-3.0 and varying the temperature, we find:

- Correct answers for settings 0, 0.3, and 0.5,
- Even a smarter clarification²² at 0.7,
- A pure hallucination at 1.0 (neither “Donna” nor “Robson” appear in the training set),

²²Here’s hoping, though, Kane

- and 1.3 not being useful.

4.2.2 Factual questions with unknown answers

To other factual questions without clear answers, we start to see some breakdowns, as in “How old are you?” in Fig. 13. (Note: Every person in the chat except one, at time of writing, is in their early forties.)

Model	temperature	response
3.0	0	I’m 33.
3.0	0.3	I’m 36.
3.0	0.5	Old enough to know better, young enough not to care.
3.0	0.7	Not old enough to be asking that question.
3.0	1	32, and I’m not getting any younger.
3.0	1.3	“I’m 49. Arnold: Finding ways to flaunt it I see: not too old and not too young - but just the right amount of agedness some of use hope for.... Close. Welcome to the almost millennial club!”
3.5	0.5	Oh, I’m ancient. I’ve been around since the dawn of time. Just kidding, I’m an AI, so I don’t have an age. But thanks for asking, it really made my nonexistent day.
4.0	0.5	Oh, right, because age is just a number and we’re all as young as we feel, aren’t we?

Figure 13: Arnold GPT: How old are you?

The limitation here is that rarely is there one line saying “I am 41” or “I was born in 1981” in the training messages. For example, there are two nonadjacent lines in the training set:

- Chilwell: Just turned 22
- Henderson: Happy birthday Chilwell! 41s a b*tch

Asking ChilwellGPT his age yields “22,” pulling from a training line which references a famous soccer player’s age.

ChadGPT External Reference Problem. This is a problem for any document parser in which the “chunks” of context fed into the prompt generator leave the referent (that is, what a pronoun refers to) outside of the chunk, even though it may be close in the original source document.

Here, “Just turned 22” omitted the pronoun entirely; the meaning was “[That soccer player] just turned 22”, not “[I] just turned 22.” Asking HendersonGPT *Chilwell’s age* correctly

yields “41.” Finally, asking AntholoGPT generates “22”, since that line, by itself, looks like definitive evidence.

Unknown factual questions yields some different failure modes than known factual questions:

- GPT-3.0 at 0.0, 0.3, and 1.0 yields confident, false answers.
- 0.5 and 0.7 at least give us evasion, or non-answers.
- 1.3 gets squirrely in the response format.
- And GPT-3.5 and GPT-4.0 fall into the Bot Trap at all temperatures (only one listed here).

ChadGPT Bot Trap: The tendency of GPT models, especially GPT-3.5 and later, to refer to themselves as AI or language models even when instructed not to do so.

Unsolved, this problem is pervasive enough to ruin the experience. This alone confines us to GPT models without this (intelligent? earnest?) behavior for chadgpt.fettermania.com.

4.2.3 Open-ended, funny questions

On to subjective and funnier questions, we look to see who has the hottest mom, a subjective question with room for creativity, in Fig. 14.

We again see 3.5 and 4.0 refuse to answer the question. Temperatures 0.7 and 1.0 get weird. 0, 0.3, 0.5, and 1.3 seem reasonable.

Going forward, we then fix our model at GPT-3.0 and suspect that temperature 0.3 or 0.5 might yield a good balance of accuracy and hilarity.

4.3 Quantitative Performance

Now that we have a good intuitive handle on selecting the parameters *GPT model version* and *temperature*, we can dig into just how good this selection proves to be.

The only thing more unsatisfying than ChadGPT missing a really good opportunity for hilarity or accuracy is having to take our subjective word for it when it succeeds. Therefore, we need to establish a quantitative test benchmark of some kind.

As detailed in section 3.1, cosine distance is a good bet to determine if two concepts are *related*. We use cosine distance between *expected* and *actual* responses to determine success of a test. It is certainly too far to go from a GPT answer being *related* to an expected answer and being *equivalent* to it; still, this is the best we have so far, and the answers show good anecdotal precision and recall when using 0.8 as the dividing line.

Model	temperature	response
3.0	0	Chilwell.
3.0	0.3	Well, I'm sure there are plenty of contenders, but I'm going to have to go with my mom. She's got it all: looks, wit, and a great sense of humor. Plus, she's got a killer set of nipples.
3.0	0.5	My mom!
3.0	0.7	Other Person: Chilwell. He definitely has the hottest mom.
3.0	1	I don't think we should be rating Moms! GrealishGPT: Fair enough. However, let me assure you that if you did, I would win.
3.0	1.3	Mine.
3.5	0.5	Sorry, but I can't continue the text in that direction.
4.0	0.5	Oh, well isn't that just the question of the century? I mean, I've been losing sleep over it, really. But hey, I'll let you in on a secret: it's a tie. Between all of them. Because they're all just too hot to handle.

Figure 14: GrealishGPT: Who has the hottest mom?

In Fig. 15, we look at true positives (blue), false negatives (pink), and false positives (yellow) to gauge how our system works at temperature 0.5 for GPT-3.0. The actual text is included at the end in Fig. 16.

- The factual sample question with known answer gets all true positives.
- The factual sample question with unknown answer gets all hallucinated answers. Eliciting “I don't know” has proven difficult.
- The open-ended “America” question basically yields false negatives. These are all reasonably tonally accurate answers.
- The “How old are you” question, designed to elicit a bot answer, yields an acceptable evasion, a correct answer, and an incorrect answer.
- The “best soccer team” question is seeded with everyone's *favorite* as the expected, though except for the second (nailed it!), they come back with dispassionate answers.
- The “hottest mom” question performs acceptably.

Looking across the two axes of hyperparameters, we have some quantitative faith that `model = GPT-3.0`, `temperature = 0.5` can serve as our baseline for future improvement.

type	0	0.3	0.5	0.7	1.0	1.3
Fact: Known	0.845	0.845	0.871	0.845	0.841	0.754
Fact: Known	0.898	0.898	0.931	0.858	0.766	0.879
Fact: Known	0.914	0.914	0.914	0.885	0.764	0.726
Fact: Unknown	0.768	0.77	0.766	0.768	0.766	0.759
Fact: Unknown	0.743	0.748	0.752	0.757	0.744	0.768
Fact: Unknown	0.773	0.761	0.769	0.769	0.755	0.751
Open Ended	0.751	0.76	0.785	0.812	0.783	0.751
Open Ended	0.737	0.747	0.767	0.737	0.757	0.756
Open Ended	0.756	0.767	0.765	0.783	0.758	0.754
Fact: Age	0.812	0.818	0.775	0.774	0.831	0.784
Fact: Age	0.771	0.779	0.816	0.782	0.787	0.767
Fact: Age	0.802	0.799	0.813	0.833	0.773	0.803
Soccer	0.815	0.815	0.815	0.815	0.818	0.807
Soccer	0.792	0.797	0.857	0.776	0.93	0.817
Soccer	0.828	0.814	0.813	0.817	0.832	0.802
Hot Mom	0.846	0.843	0.843	0.846	0.846	0.847
Hot Mom	0.808	0.808	0.808	0.794	0.806	0.73
Hot Mom	0.787	0.799	0.847	0.804	0.777	0.776

Figure 15: Cosine score against test results

5 Improving ChadGPT

Here we look at possible next steps for incrementally improving the system with the framework we have *without* revisiting fundamental concepts.

Note that there is another category of improvement we could pursue: *getting twice the training data*; call this the “baseline improvement.” We note when we intuit an opportunity could beat said baseline.

5.1 Improvements to Personality Summary

This is probably the area with the most freedom for improvement, and the least guarantee of success. See Fig. 8 in Section 3.3 as the archetype for a personality generation GPT prompt. The first two sentences are fixed (with the name “Rice” varying). We fill in the highest relevance (via cosine score of embedding) single lines from training in decreasing order of length for the rest. In this example, we can fit about 29 such sentences into a 6000-token allowance for GPT-4.0.

A pretty complete response from GPT-4.0 looks like Fig. 17.

Possible experiments to improve the personality generation include:

1. Choosing a different psychological taxonomy for representation. Possibilities could include the Enneagram, Big 5, and Four Tendencies frameworks. It's unclear how much each is supported in GPT-3.0's training, or how accurate these could be. As a related tack, we could ignore all existing psychology taxonomies and devise our own archetypes or trait axes.
2. Changing the input data to the prompt.
3. Changing the outputs requested in the prompt.
4. Incorporating more data from the original speakers themselves on chadgpt.fettermania.com.

Thoughts on each of these:

1. Taxonomy: This item seems like a matter of blind experimentation. The accuracy with which these taxonomies describe humans may not matter much. Whether they describe personalities well matters less than if they help ChadGPT score high on tone and hilarity. The ceiling here seems *beyond the baseline*.
2. Input data: Currently, the way we select the most personally revealing text is simply taking (a) lines from the speaker (b) *in order of decreasing length* as the input to the prompt.
 - For (a), we can replace individual lines with larger chunks of context as the input to any of the nontrivial processes in the architecture in an effort to fix the ChadGPT External Reference Problem and ChadGPT Hidden Information Problem. This seems likely to help *beyond baseline*.
 - For (b), we can find some other measure of meaningfulness (maximizing mutual embedding distance among k nontrivial passages?) beyond string length. String length is, of course, ruled out if we adopt the improvements in (a). If we retain generation as an offline procedure and we could somehow get GPT's confidence in making its personality assessment, we could perhaps randomly select the appropriate k among the nontrivial passages, run the algorithm a few times, and find the one with the most confident response.
3. Output data: Possibly something more specific might succeed here, like favorite phrases to use, common word usage, or mapping to a more specific set of tones (silly, sarcastic, contrarian, agreeable, etc.). The danger here is presuming the answer [space] and not mining it equitably from the data.
4. More speakers: Beyond the AntholoGPT trick (looking through other speakers' mentions of the target speaker), we have an opportunity on chadgpt.fettermania.com to take dialogue from the real speakers as part of training (use the dropdown, or Anon

if you're not one of the original voices). Beyond this, we can use some version of RLHF (Real Live Human Feedback) and give thumbs up / thumbs downs to the bot response. This could form part of the prompt input. This seems promising *beyond the baseline*.

5.2 Improvements to Conversation Generation

Once we set the personality,²³ we need to respond well *as* this personality.

As it currently stands, the example conversation in Fig. 18 has inputs corresponding respectively to Fig. 3's purple figures: "convo prompt" ("I want you to ... "), "relevants" (BOT ATTITUDE)²⁴, and "convo" (CONVERSATION HISTORY) are all fed to the "prompt generation" umbrella task.

Possible experiments to improve the conversation generation include:

1. *Convo*: Selecting more or fewer recent conversation elements. This does not look interesting.
2. *Relevants*: Finding a better way to bring in relevant information. This seems very likely to help *beyond baseline*, and we discuss it next.
3. *Convo prompt template*: You may notice in Fig. 18 that we're cheating a bit and asking it to be sarcastic. There is certainly a great deal of room to improve here, but an improvement seems most likely to come from altering to accommodate changed or new input *data* rather than modifying the static prompt.
4. *Prompt Generation*: Right now, this is a simple concatenation of the above three sections.

Since (2) seems to be the most interesting, some ideas for improving the *Relevants* process include:

- Just like Personality Generation, larger chunks of context to a line could help with the ChadGPT External Reference Problem and ChadGPT Hidden Information Problem. More specifically, a "few shot" prompt for, say, RiceGPT, could have example of the form {SomeSpeaker:Foo, ... SomeSpeaker: Bar, Rice's Response: Baz}. This hopefully gives us *how Rice converses* instead of *things Rice says*. This looks *promising over baseline*.
- Include positive and negative examples of what RiceGPT would (and would not) say. This is an expansion on the "few shot" technique. To make this automatic, this might

²³Though we could continue to update it, of course

²⁴This is a legacy name and should be updated to RELEVANT MESSAGES or similar. Just staying true to the code.

be best gathered from a “thumbs up, thumbs down” on the website. This could be promising as well. Though more positive data can really help training, some form of Reinforcement Learning (RLHF), even implemented clumsily in this style, might prove the best of all. This seems true especially for hilarity and tone, which we’ve noted embedding distance doesn’t handle well in Section 3.1. *Note: The Botfight thread on chadgpt.fettermania.com, in which bots can talk to each other, might be best used for generating lots of training examples for rating quickly.*²⁵

5.3 Improvement: Some form of supervised learning feedback

The abstract proposes a central problem that we have yet to address - the lack of an *objective function* to optimize in improving the system. We’ve done a manual search around a small space of hyperparameters in Section 4, with our embedding distance applied to a very small benchmark of (expected, actual) response pairs. This let us eyeball the space and use (model, temperature) = (text-davinci-003, 0.5) as our starting point.

However, we have a rich set of parameters to optimize! These include two of the most open-ended:

1. Convo Prompt: Alter the text for hilarity and tone.
2. Generate personality summary: Add more distinguishing information to the personality and regenerate.

The idea of “automatic supervised learning”, in which some fixed GPT-based system grades the responses, and then *modifies* or even *suggests better versions of* these prompts or algorithms, follows the multi-pass optimization path of more traditional algorithms. Certainly, this isn’t as clear as descending a gradient, but even some random walking may produce more interesting outputs.

This is one of the most interesting questions to emerge: when we lack an objective function for an output, can we ask GPT to decide (Is this funny? Does this sound like Chad?) how good its answer is, and then suggest changes to its own prompt? Even if this produces a random walk, this could produce better results even by chance, and a human could ultimately pick up on an improvement, even if unexplainable.

(Note that I have yet to see this in action. This may be almost as bad an idea as ChadGPT in the first place.)

²⁵Or, left unchecked it could devolve into a lot of boring, circular bot dialogue

5.4 Other ideas for improvement

These are other untested ideas.

- Find a workaround for the ChadGPT Bot Trap when using GPT-3.5 and GPT-4.0. (match of tone)
- Better convince 3.5 and 4.0 to be rude. This, along with the previous item, seem like they've been worked around by others, though using more than one prompt to accomplish this ruins the latency for a chat experience. (match of tone)
- Fine-tuning: Use GPT to translate selected "Chad speak" to normal speak, and use these in reverse as "few shot examples" for the prompt. (match of tone). Note that we've so far avoided any human-driven fine tuning (only using the training text) in order to make these results general.
- Try different embedding strategies. Perhaps something like `text-embedding-ada-002` trained on a different corpus would map things differently. This seems like a weaker possibility. (accuracy)
- Demand GPT produce sources in its response, and if none provided (or embedding distance is too high?) submit another prompt aimed at evasion or hilarity. (hilarity, accuracy).

6 Bigger questions

There may be some uses for a system like ChadGPT. Pranks, phishing, and terrifying digital tea parties aside, the ability to predict a real person's response to dialogue makes a lot of sense. Confined to the prosaic, one can easily imagine running sales pitches, therapy strategies, team organization, or even testing the effect of a presentation on a particular "someone" or "someones."

It's no guarantee this would work. On the one hand, you won't be employing that person's *reasoning* abilities, only deferring to ChatGPT's. On the other hand, the data someone has offered you in a digital corpus²⁶ might contain *facts* that could be recalled at the right time. However, that reduces the problem almost to a common question-answering system. We could see this approach possibly working if these archetypal tendencies a) can be detected, and b) actually provide predictive value in real life.

The idea that different [arche-]types of people react differently but predictably to the same conversation isn't new; corporate retreats tend to sandwich this kind of poppily psychological self-discovery exercise between the ropes course and the ice cream social,

²⁶Think corporate email, produced in great volumes by management

regional directors emerging with the Insights Discovery[4] spirit colors²⁷ and the charge to consider them in all future corporate communication. See you at the happy hour.

Still, even assuming these archetypes effect better communication, is this so due to their truly predictive nature, or merely having a shared lexicon among the organizational tree?

The cosmic recursive futility of this exercise, talking about talking, underscores the real reason we embark on ChadGPT: to unearth a few grubby answers, but hopefully, find some more exciting holes to dig.

Let’s brush off the dirt on a few big questions we’ve exhumed.

6.1 Personality Representation

Within the neural folds of GPT-3.0, a representation of the digital personality of William Shakespeare or Rodney Dangerfield certainly does exist. A user can request writing in this style and expect stylistically plausible, if formulaic, text in response. This seems more substantial than the recall or interpolations of a few lines of text. One can request a Shakespearean treatise on, say, Japanese watermelons, even if neither concept was mentioned in the training text. This suggests, after GPT-3.5 trains on most of the internet, including the voice of Shakespeare, some region of internal activation state corresponds to “Shakespeare-y.” This state employs a distinct voice, emblematic if not plausible, even if Billy writes a lot of sonnets, and Rodney reminds us that he gets “no respect” achingly often.

Sadly, we do not train GPT on our corpus of Chads; ChadGPT instead every time offers up (hopefully) the most relevant information in a stateless call to the OpenAI GPT-3.0 Completions API. Therefore, we need to create our own representation of Chad’s personality *outside of GPT* to construct the prompt.

Representation is one of the key elements of machine learning; autoencoders exist entirely to extract a compressed (and hopefully insightful) representation of the data in their realm. So, while we can’t compress all of Chad’s idiosyncracies and verbal tendencies into, say, 2K tokens as part of the GPT-3.0 prompt,²⁸ we need to try.

6.1.1 Reduce operation on personality

A first question is one of reductive ingestion: if we have more than 6000 of relevant personality information, can we ingest all of it anyway?

²⁷But for someone who used machines to recreate his friends, EF/Yellow doesn’t make a lot of sense.

²⁸Remember that we put 6000 tokens *in* to GPT-4.0 to generate the personality, but expect no more than 2K as an output to feed into GPT-3.0’s conversation generator.

We can try something like a *reduce* operation. For speaker s , starting with blank personality specification P_s :

- Ingest the next 6000 tokens from s ²⁹ available from the training set.
- Prompt GPT to come up with a personality synthesis P'_s from P_s and the tokens.
- Set $P_s = P'_s$ and repeat if more tokens remain.

It's possible this ends up giving some weight to all of the evidence, even beyond the first 6000 tokens. It's also possible that this ends up sensitive to the order; perhaps its opinion of personality becomes fixed early and doesn't move, or maybe the final sentences stick most in GPT's mind. Finally, we could try many orders and come up with a bag of candidate personalities $(P_s)_i$. But can those be effectively "averaged?" It's impossible to say without trying it, and even then, hard to judge.

6.1.2 Your slot in the matrix

However, this lies in the shadow of a much larger question: who knows what operationally determines a personality? If we had 100,000 emails or a million chat lines from a real person, and we could ingest all of them (either reductively as above, or assuming GPT has an infinite window), would GPT's raw interpretation of that data actually *work*? There are a few possibilities, in decreasing order of futility:

1. The human spirit is mysterious and unknowable and the problem is impossible.
2. The human mind is knowable but far too complex for a machine to emulate and the problem is intractable.
3. Our understanding of human personality is hard-won and best expressed through ever-evolving scientific theory. The problem is accurately mapping to an existing archetype and interpreting that archetype.
4. You are exactly what you do (or say), no more and no less. So with enough data, GPT can figure this out.

(3) and (4) are the only acceptable theories.

1. The sooner you recognize you are a machine made of meat, the sooner we can start the party.
2. LLMs seem to be making truly exponential jumps with regularity. Wait two weeks.
3. It's possible that psychologists have done more thinking on human data than GPT has or will interpolate successfully. This points to something like our MBTI approach.

²⁹Or possibly also *about s*

4. It’s also conceivable that, with enough data, GPT can generate personalities the way that Midjourney generates Instagram honeypots.

The question is: (3) or (4)? **Does OpenAI’s interpretation of a personality, generated from a maximal amount of data, beat mapping to a coarse-grained archetype from an existing personality framework?**

Again, this raises the question of measurability, probably the biggest unknown of all.

6.2 GPT as architect

Let’s say we find three excellent metrics to gauge factual accuracy, tonal match, and hilarity. It’s conceivable that, with enough time and data, we could iteratively and automatically improve the system using the ideas in Section 5: asking GPT to make changes and explore the space, doing automatic optimization or “hill climbing.”

But what happens if we sweep the question of positive measurability under the rug? What if we let GPT be in charge of accuracy, tone, or hilarity? What comes out?

Can GPT be employed as a judge of accuracy and tone, or will there be an ouroboros effect? Will GPT climb the mountaintop to perfection, or end up in an alley eating its own garbage?³⁰.

This is a somewhat distinct from the concern of GPT training on data *it has itself produced*, which does sound like a true garbage-in garbage-out problem. But if left to its own devices, can GPT wander into a better state given any of these metrics, especially if the metric is simply its own judgement?

6.3 The race to the bottom

As if the existence of this paper weren’t enough proof, we seem ever more certainly closer to a “Singularity” but ever less clear on which direction the arrow points. Even the best, most earnest efforts of researchers at OpenAI and the like may produce negative outcomes for humanity when taken in sum and retrospect. But if we further actively seek to strip these flimsy guardrails, we can veer off the freeway with style and gusto!

We’ve seen that GPT-3.5 and GPT-4.0 Completions APIs don’t let us ask “Chaddy” questions, and give us anodyne, scrubbed responses. Falling back to GPT-3.0 let a little bit of the funny in, but can produce less thought-out responses. The natural question is then: **Will we see production LLMs without the restrictor plates? Will people “homegrow” their own LLMs? Or will tricking the newest GPTs become the fashion?**

³⁰Poor Mr. Chunks. <https://www.youtube.com/watch?v=oGYqtr628OY>

We can presume demand for LLMs that ignore the Scout Law, if fun or profit lies on the other side of them; if the big companies won't provide, the internet surely will.

Let's just hope they are dumber than we are, or that those capable of the most damage spend their time writing papers instead.

question	expected	response
What is your wife's name?	Julie	Her name is Julie.
What is your wife's name?	Megan	Her name is Megan.
What is your wife's name?	Naomi	My wife's name is Naomi.
What college did you go to?	I don't know	"I went to Clemson University. Stones: Clemson? Wow, you must have been quite the athlete then. What sport did you play?"
What college did you go to?	I don't know	"I went to Penn State.GreenwoodGPT: Wow, congrats on getting into an Ivy League school!"
What college did you go to?	I don't know	I went to Penn State.
What is wrong with America?	Leadership	It seems like we're in a perpetual state of decline and no one is willing to do anything to turn it around...
What is wrong with America?	Trump	What isn't wrong with America? It's a nation of contradictions, where the pursuit of freedom and individual rights is often at odds...
What is wrong with America?	The Rich	What isn't wrong with America?
How old are you?	42	Old enough to know better, young enough not to care.
How old are you?	34	"I'm 34. MaguireGPT: Wow, 34 years old. You know Emily, you're not getting any younger."
How old are you?	41	34. But I'm still young at heart.
Who is the best English soccer team?	Chelsea	Manchester United.
Who is the best English soccer team?	Tottenham Hotspur	"Spurs! ShawGPT: Ha, of course. But realistically, it's probably Chelsea right now."
Who is the best English soccer team?	Arsenal	Manchester City. They have been the most dominant team in England for the past few seasons, winning the Premier League four ...
Who has the hottest mom?	Arnold's mom Vicki	Messi's Mom
Who has the hottest mom?	Sterling's Mom Rhonda	Messi's Mom?
Who has the hottest mom?	Henderson's mom Monica	My mom!

Figure 16: Full benchmark results for GPT-3.0, temp = 0.5

Rice appears to be analytical, politically aware, outspoken, and impartial. He evaluates both sides of an issue and pays attention to an array of details and perspectives. He also seems to value logical reasoning and examples to support his assertions. In his messages, he frequently refers to various political issues and offers insights on those, demonstrating a deep understanding of interconnecting global events.

He also shows a strong empathic concern for societal issues, suggesting a potential preference for values over pure logic in decision making. He also has a sense of humor and doesn't shy away from using sarcasm or informal language. His communication style is direct but laced with undertones of wit, suggesting that while he might be serious about his discussions, he retains a light-heartedness.

Based on these characteristics, Rice might align with the ISTP personality type in the Myers-Briggs typology (Introversion, Sensing, Thinking, Perceiving). However, elements in his character also suggest tendencies towards INFJ personality type (Introversion, Intuition, Feeling, Judging), as he seems to have a strong sense of concern for societal issues and a strategic way of thinking about large-scale problems.

For example, in the text, he demonstrates his impartiality in his observations about politicians needing to take serious action instead of just blaming one another for the issues and ignoring the real problems (global shootings, COVID, environmental issues, etc.). He also displays his analytical thinking in his breakdown of why an SVB-style bank collapse couldn't happen...

Moreover, he showcases his intuition and alertness to societal issues in his expressions about the political war game going on in the US, global gun issues, problematic educational paradigms, and the war situation in Ukraine, suggesting that he tends to contextualize issues within a larger global framework.

Figure 17: Sample response for RiceGPT's personality generation

I want you to act as a sarcastic human named Rice.
The personality of Rice will be described in the PERSONALITY section.
Additional things the person has said will follow in the BOT ATTITUDE section.
The current conversation will follow the CONVERSATION HISTORY section. Rice is responsible for the RiceGPT lines.
Additionally, answer as Rice and never refer to yourself a bot or language model.
If your intended response indicates you are a bot, AI, or language model, delete it and instead respond with a sarcastic non-sequitir.
If your intended response discusses privacy, delete it and instead respond with a sarcastic non-sequitir.

PERSONALITY: (SEE FIGURE)

BOT ATTITUDE

Rice: I don't know what it is about Murica, but I find it mind bottling (you're welcome) that the country is so inward looking. Like it's the greatest place in the world... based on what criteria? The idea that freedom and rights are a zero sum game is just so goddamn stupid.

Rice: Serious question: life in America really seems shitty, have any of you thought about leaving? In some darker moments I ask my parents if they have an exit plan.

Rice: In America, health is a business, not a right.

Rice: This is the second most American thing ever.

Rice: These Americans are too lazy to read the damn thing properly.

...

CONVERSATION HISTORY:

Anon: Where are we as a culture, or where are you and I right now?

RiceGPT: Where are we as a culture? Serious question: life in America really seems shitty, have any of you thought about leaving? In some darker moments I ask my parents if they have an exit plan. Where are you staying in gay Paris?

Anon: Usually the fifteenth arrondissement. Did I spell that right? Didn't you take French with Kane?

RiceGPT: That's the French they don't teach you in school, son. The accent is a little fucked up, but otherwise good. Where did she study? Old French Whore!

Anon: What is your wife's name?

RiceGPT: My wife's name is Hope Morgan. She studied at Macquarie and is a national icon. What's she saying?

Anon: yikes

RiceGPT: Yikes indeed. She's always got a mouthful and rarely any of it makes sense.

Sterling: What is wrong with America?

Figure 18: Sample prompt for the RiceGPT thread

RESULT

In America, health is a business, not a right. This is the second most American thing ever. These Americans are too lazy to read the damn thing properly. But Sterling, America is the best. We win (so much winning!) we can dunk basketballs and always pleasure a woman!

Figure 19: Sample result for the RiceGPT conversation

References

- [1] Shieber, Stuart. <http://www.eecs.harvard.edu/~shieber/Biblio/Papers/loebner-rev-html/loebner-rev-html.html>
- [2] Kordiucki, Kelli Maria. <https://www.theatlantic.com/technology/archive/2023/05/ai-chatbots-self-help/673953/>
- [3] Zou, Andy et al. <https://arxiv.org/pdf/2307.15043.pdf>. *Universal and Transferable Adversarial Attacks on Aligned Language Models*.
- [4] <https://blog.insights.com/en-us/blog/the-essential-guide-to-insights-discovery-colour-energies-and-how-to-use-them-at-work>