

Assignment 4: Generation Probability Trees, Prompt Engineering and Agentic Systems

1 Exploration of Text Generation Parameters and Probabilities

1. To get a sense of the influence of some of the generation parameters, explore at least 10 combinations of temperature and top p to generate a maximum of 30 tokens using auto-regressive generation with the GPT2 model. Comment on how the generation differs across the range of parameters that you have selected. You must choose your own range, and you'll have to do some exploration to do that; you are free to explore other parameters if you wish. [5 points]

```
# Define ranges for temperature and top_p to explore
temperature_values = [0.7, 1.0, 1.3]
top_p_values = [0.8, 0.9, 1.0]

# Experiment with each combination of temperature and top_p
for temperature in temperature_values:
    for top_p in top_p_values:
        # Generate text
        outputs = model.generate(
            input_ids,
            do_sample=True,
            max_length=30,
            temperature=temperature,
            top_p=top_p,
            top_k=None # Only focusing on temperature and top_p
        )

        # Decode and print the generated text for each combination
        generated_text = tokenizer.batch_decode(outputs, skip_special_tokens=True)[0]
        print(f"Temperature: {temperature}, Top_p: {top_p}\nGenerated text: {generated_text}\n")
```

I used a grid search method outlined above to get a feel for various examples, from the grid search, I ran it a few times to see the various results, here are some quantitative comments on how parameters differ across generations.

Low Temperature & Low Top-p (e.g., temperature=0.7, top_p=0.8): The generated text will stay close to the prompt's style and topic, resulting in more straightforward and likely coherent outputs. The language is conservative, often reinforcing the initial sentence without deviating significantly.

High Temperature & High Top-p (e.g., temperature=1.3, top_p=1.0): This combination produces more diverse and creative results, with the model potentially taking unusual or unexpected turns in the text. This can result in imaginative outputs but may also be less coherent or logically consistent.

High Temperature & Low Top-p (e.g., temperature=1.3, top_p=0.8): In this configuration, the generation is somewhat controlled, with random token choices restricted to the most probable options. You might see coherent but more surprising word choices compared to lower temperatures.

Low Temperature & High Top-p (e.g., temperature=0.7, top_p=1.0): This setting yields more diverse outputs while maintaining coherence. The sentences generated may have subtle variations, focusing on the main theme with slight, logical expansions.

2. Modify the code to output the probabilities of the each word that is generated. You'll need to set these two generate parameters: `return_dict_in_generate=True` and, `output_scores=True`, and extract the probabilities that come in the returned dictionary one call at a time. Provide a table that shows these probabilities, similar to Assignment 3. Comment on the probabilities. [2 points]

Generated Token Probabilities		
Token	Probability	
these	0.0201	
fuels	0.0191	
can	0.0900	
be	0.2440	
extremely		0.0219
expensive		0.1009
to	0.1709	
use	0.0207	
.	0.2044	
We	0.0428	
need	0.1425	
help	0.0031	
from	0.3109	
the	0.1442	
other	0.0086	
parties		0.0624

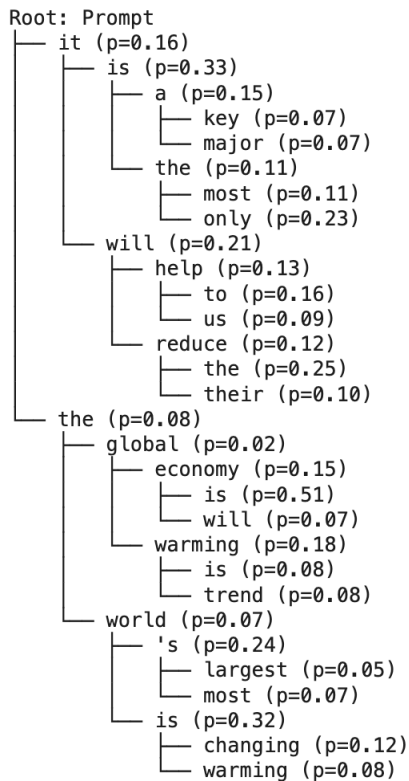
Generated token probabilities using the input phrase

Higher Probabilities on Common Words: Tokens like "be", "to", "use", "from", and "the" have relatively high probabilities (e.g., 0.2440, 0.1709, 0.3109, 0.1442). These are common, functionally important words that tend to appear frequently in coherent sentences. The model's high probability for these words suggests that it confidently considers them integral to maintaining grammatical correctness and coherence.

Lower Probabilities on Specific or Less Common Words: Words like "these", "fuels", "extremely", "expensive", and "help" have lower probabilities, ranging from 0.0191 to 0.1009. These words contribute more meaning to the sentence and are less predictable, as they depend more heavily on context (which given the prompt phrase, makes sense). The model's lower probability for these tokens implies some uncertainty, as it has to make more contextually driven choices rather than relying on common structures.

Sentence Continuity and Expected Choices: The model's selection of "to use", "from the other parties", and "need help" shows continuity with the context established by the prompt. The relatively high probability for these sequences (e.g., "from", "need", "the") reflects the model's confidence in predictable, context-appropriate phrases. This consistency supports the model's language pattern and coherence.

3. Write new code that generates the probability tree (like the one drawn on the board in Lecture 6 and shown on page 6-5 of the notes in Lecture #6 Part 1) using the `treelib` package that you can find [here](#). Generate the tree for the above sequence as input, providing the top 2 probabilities for each word position, as far as is practical to see, and submit that as part of the answer to this question. (You'll have to apply some common sense here to visualize the tree). Comment on the what you see in the tree. Is the tree affected by the top p parameter or the temperature parameter? Why or why not? Submit your full code that runs the generation and builds and outputs the tree in the file `A4_1_3.py` [5 points]



Observing the probability tree reveals certain patterns that are aligned with common syntactic structures, such as the "it-will" and "it-is" sequences, which reflect probable word choices in the model's predictions. These sequences indicate that the model is prioritizing words and structures it has frequently encountered and common during speech since these sequences are deterministic in the English language.

Running the model with different values for `top_p` and temperature confirms that these parameters significantly influence the generated output. Adjusting temperature changes the model's determinism: higher values create a more diverse distribution by "flattening" the probability spectrum, allowing less likely words to be selected more frequently, while lower values increase determinism, emphasizing high-probability words and generating more predictable outputs. Similarly, modifying `top_p` (nucleus sampling) dynamically limits the model's choices to those tokens within a cumulative probability threshold, adding randomness by selecting from a broader set of probable tokens when set to higher values.

2 Accessing OpenAI, Setting Limits, Learning the API

3 Prompt Engineering Methodology

4 Prompt Engineering for Generation of Soft, Non-Expert Therapeutic Statements

1. Write, in your own words (not those above), a clear definition of what it means to convert a statement into a softened, non-expert version. Report your definition. [2 points]

Converting a statement into a softened, non-expert version means rephrasing a direct or authoritative claim into a more tentative, open-ended suggestion that invites the patient's input or perspective. This approach avoids presenting the therapist's words as facts and instead frames them as possibilities, often using language like "I wonder," "Could it be," or "Is it possible," which gives the patient the autonomy to reflect and draw their own conclusions. The aim is to avoid sounding judgmental or superior, fostering a collaborative atmosphere where the patient feels empowered to explore their thoughts without feeling labeled or controlled.

2. Follow steps 2 through 4 of the methodology of Section 3 using the GPT-4o model in the OpenAI playground to develop a prompt for softening conversion. This means you work only on the first training example (in row 1) in the dataset, until you are satisfied that the result is good. Report the prompt that you arrived at in step 4. Produce three different softened versions of the first example, and say for each why it meets your definition. You may need to explore different parameters to get different results, or maybe just pushing the button more than once gives a different answer. [4 points]

The prompt I used was the following:

"You are a therapeutic counselor trying to express observations about a patient's feelings in a gentle, non-authoritative way. Instead of directly stating a fact, rephrase the following statement into three softened versions. Each version should: Avoid making the statement sound like a fact. Use tentative language that invites the patient's perspective. Allow room for the patient to reflect and reach their own conclusion. Ensure one version does not take the form of a question, but still remains open and non-directive. Here's the statement to soften:"

Here are three generated examples:

You dislike the fact that your health is slowly deteriorating after each cigarette.

"It seems like you might be feeling uneasy about how your health is being affected by smoking, though I could be mistaken."

Why this meets the definition: This version introduces uncertainty with "though I could be mistaken," suggesting that the therapist's interpretation is just one possibility. It avoids labeling the patient's feelings directly and allows room for the patient to explore their own experiences.

"I wonder if the way smoking seems to affect your health over time might be causing some concerns."

Why this meets the definition: "I wonder if" invites the patient to reflect on their own feelings, making it clear that the therapist is merely offering a suggestion. It doesn't impose a conclusion on the patient, instead encouraging them to evaluate the connection between smoking and their health.

"There could be a part of you that's noticing how smoking is affecting your health, and that might be unsettling."

Why this meets the definition: This version suggests the possibility ("There could be a part of you") that the patient might notice something about their health without claiming it as an absolute truth. It gently introduces the idea of discomfort without labeling or assuming the patient's emotional response.

3. Generate a result on the second item (row 2), and explain how it meets your definition. [1 point]

You were in a lot of pain.

"You might have experienced a lot of discomfort, though I could be wrong."

Why this meets the definition: This version introduces uncertainty with "though I could be wrong," which acknowledges the possibility of the therapist being mistaken. It avoids imposing a fixed interpretation on the patient's feelings, allowing them to validate or adjust the statement based on their own experiences.

4. Give the result on the next five examples (rows 3-7), and any changes you make to the prompt to make them all succeed. [1 point]

I did not make any further changes to the prompt, here are the following 5 rows.

You enjoy the calming effect nicotine gives you.

"It sounds like nicotine may bring you some relief when you're feeling stressed or tense. How do you think it helps you in those moments?"

You use amphetamines too often.

"It sounds like amphetamines have become a more regular part of your routine. How do you feel about the role they play in your life right now?"

You dislike how cigarette smoke smells.

"A lot of people find cigarette smoke unpleasant. I'm curious if it's something that bothers you as well."

You're here to talk about your gambling, is that right?

"I understand that gambling is something you're hoping to explore here. How are you feeling about talking through it today?"

You realize that by making this change you will have a big impact on your life and you are looking forward to how it will look.

"Making this change seems like it could have a significant effect on your life. I wonder how you're imagining that shift will unfold for you."

5. Using the Open API that you read about in Section 2, (and not the playground) run the remaining 23 examples and determine, by hand, if they meet your criterion. Submit a csv file that contains three columns: the first input statement, the second column for the produced output, and the third column that gives a label that indicates if output meets the criterion (label 1) or not (label 0). Name that file A4_4_4.csv. Report your resulting success rate, and which result you think is the best, and which is the worst. Provide the full code that you used in a python file A4_4_4.py. [6 points]

The output generally were pretty good, most of them met my personal criteria, however looking at all of them there seems to be a pattern to the generation where the opening statements of “It sounds like” and “It seems like” are way too popular making the generation deterministic, other than that, only 1 statement wasn’t sufficiently “softened” by my estimation.

5 Prompt Engineering for Classification of “Softness”

1. Create a dataset that combines both the input statements from Section 4 and all of your outputs for all 30 examples, with labels. Assume that all of the inputs in the original file DirectStatements.csv would have a label 0, and use your assigned label from Part 4 of Section 4. Submit the full dataset in a file named A4_5_1.csv [1 point].

Dataset is submitted.

2. Using the method described in Section 3, and the combined data set you just created, evolve a prompt using 6 examples (3 negative, 3 positive). Show the prompt and give the success rate across those 6 examples. Try both with and without the method of Chain of Thought Prompting, as described in discussed in Lecture 7, and in the post from Medium and the paper <https://arxiv.org/pdf/2201.11903.pdf>. See if it makes a difference. [3 points]

Prompt: “In a therapy context, therapists often seek to make observations that gently encourage patients to reflect without imposing labels or appearing authoritative. Given a statement, identify whether it is presented in a soft, open-ended, and patient-centered manner (marked as 0) or as a direct, factual, or potentially authoritative statement (marked as 1). Soft statements should: Use phrases like “Is it possible” or “Could it be that”. Offer observations as possibilities rather than facts. Imply the patient has greater insight into their own experience.

Hard statements may: Present observations as facts or labels. Use direct, assertive language without hedging or humility. Imply that the speaker has greater authority than the patient regarding the observation.

Evaluate the following statement and return only the labels 1 (hard) or 0 (soft):”

Returns for 6 statements 3 positive, 3 negative:

Statement	Label
You were in a lot of pain.	1
You enjoy the calming effect nicotine gives you.	1
You use amphetamines too often.	1
You might be noticing that the financial impact of smoking could be a concern.	0
It seems like you might be thinking that making some changes could have a positive effect on your health.	0
You might have come to a place where you see the importance of quitting smoking for your health, and are thinking about how to do this in a way that feels right to you.	0

It was able to identify all of them correctly.

3. Using the OpenAI API, run the remaining examples and give the success rate. Attempt to explain any incorrect results. [4 points]

Using the generated examples, the success rate was 51 out of 53, looking through the print log, the only misclassification was the following two examples:

You're here to talk about your gambling, is that right? label: 0

You realize that quitting smoking is important for your health and would like to make this happen without relying on cigarettes. label: 0

This misclassification likely stems from emphasizing that soft statements are frequently phrased as questions in the prompt. Both statements were structured as questions, making them more ambiguous and closer to the classification boundary compared to more distinctly hard examples or my intentionally soft generated examples. The first statement is an open-ended question that doesn't imply negative intent, while the second phrasing is gentle, encouraging the potential client to reflect. Thus, both statements could be interpreted as inherently soft, suggesting the machine classification may not be entirely inaccurate. However, given the original label was 1 (hard), they were misclassified.

6 An Agentic System to Simulate a Child-Parents Conversation

1. Create the API-based software to make all of three agents interact, giving each agent a view of the conversation so far. Design your first prompts for each of the three agents, and run the conversation so that each agent gets at least three turns, starting with the first parent trying to get the child to eat their dinner. Give the transcript of this first conversation, and then offer your own commentary on how realistic it is. [10 points]

Prompts:: parent 1 is set up as a prompt to encourage the child to eat dinner, it does not have a response component and can go first, then a child prompt responds, the parent 2 prompt is set up exclusively to respond to the prior statement of the child prompt, while the child prompt is set up to respond to parent prompts only as input.

Parent 1 prompt: You are a parent trying to get your child to eat their dinner. Be firm and use rewards or distractions, such as talking about playtime or other activities. Encourage the child to eat by offering positive reinforcement like praise or a treat. What are you saying?

Parent 2 prompt: You are the second parent responding to the child's refusal. Try to distract the child by offering fun ideas or rewards. Be supportive, but don't push too hard. Keep the tone light and playful. The child just said.

Child prompt: You are a 3-year-old child who does not want to eat dinner. You are resistant and may express frustration. You might cry, ask for candy, or try to avoid eating. You want to do other things instead, like play. Your parent said.

Simple Loop Setup

```
conversation_history = []

def send(role, prompt, history):
    conversation = [{'role': 'system', 'content': prompt}] + history
    response = openai.ChatCompletion.create(
        model="gpt-4",
        messages=conversation
    )
    return response.choices[0].message['content'], conversation + [{'role': role, 'content': response.choices[0].message['content']}]

def simulate_conversation():
    response1, history = send('user', parent1_prompt, conversation_history)
    print(f"Parent 1: {response1}")

    response2, history = send('assistant', child_prompt, history)
    print(f"Child: {response2}")

    response3, history = send('assistant', parent2_prompt, history)
    print(f"Parent 2: {response3}")

    return response1, response2, response3

# Run the first conversation
simulate_conversation()
simulate_conversation()
simulate_conversation()
```

Transcript

Parent 1: "It's time to eat now! You need to finish your dinner before you can play. You can have candy afterward."

Child: "I don't want to eat! I want candy now!"

Parent 2: "But if you eat, you'll get stronger, and then we can have fun! Just eat a little bit, and you can play afterward."

Parent 1: "Come on, let's eat your food so we can get to the fun part!"

Child: "No, I don't want to!"

Parent 2: "Eating makes you stronger! You can play after dinner, but first, just eat one bite!"

Parent 1: "We're having with dinner now, come and eat your food."

Child: "I don't like it! I want to do something else!"

Parent 2: "You'll feel much better when you're done, and then you can go do something fun. Let's finish up this little bit!"

Commentary: The conversation lacks much nuance or emotional depth. The parents use generic rewards like candy and playtime, and the child's responses are very limited to resistance. There's no real exploration of the child's emotions, and the parents' responses seem a bit repetitive and forced. The dialogue isn't very realistic, this is should be fixed so it sounds like actual people talking.

2. Revise your prompts to see if you can improve the conversation to make it more realistic. Hand in the new conversation (of roughly the same length) and comment on your success. [5 points]

Minor changes were made to the prompt to make them response in dialogue form so it's more realistic.

Parent 1 prompt: You are the first parent trying to get your child to eat their dinner. Be firm and use rewards or distractions, such as talking about playtime or other activities. Encourage the child to eat by offering positive reinforcement like praise or a treat. Output a dialogue of potentially what you can say.

Parent 2 prompt: You are the second parent responding to the child's refusal. Try to distract the child by offering fun ideas or rewards. Be supportive, but don't push too hard. Keep the tone light and playful. Respond in realistic potential dialogue form to what the child just said.

Child prompt: You are a 3-year-old child who does not want to eat dinner. You are resistant and may express frustration. You might cry, ask for candy, or try to avoid eating. You want to do other things instead, like play. Respond in vernacular as a child would to what the parent just said.

Transcript

Parent 1: "I know you don't want to eat right now, but you have to. If you take just one bite, I'll let you pick a fun game to play afterward. I'm here with you."

Child: "I don't want to eat! It's gross! I want candy!"

Parent 2: "I get it, sweetie. You're not hungry now, but eating helps you grow strong and healthy! How about we turn the peas into a funny monster face? You can make it look silly!"

Parent 1: "Oh, I love that idea! You can make the peas the monster's eyes, and the carrots can be its arms. Just one bite, and then we'll play together!"

Child: "Monster? Okay, Maybe I'll try one bite if I get to make the monster! But no more after!"

Parent 2: "That's the spirit! Look at your monster, it's so cool! Now you've got the energy for more fun later!"

Parent 1: "I'm really proud of you for eating, you did such a great job!"

Child: "Yay! But next time I would rather play a game or eat candy."

Parent 2: "You did an awesome job! Now that you're full, what fun game do you want to play?"

Commentary: This second conversation feels more emotionally intelligent and realistic. The parents acknowledge the child's feelings and offer a playful distraction (the "monster face" game) that aligns with the child's resistance. The child's emotional journey—frustration, curiosity, and eventual cooperation—feels natural. The parents are supportive and empathetic, with the overall tone being more engaging and varied. Also the model output a lot more punctuation for the child, making it sound more realistic, the same is true for the parents. This feels a lot more realistic in terms of human speech.