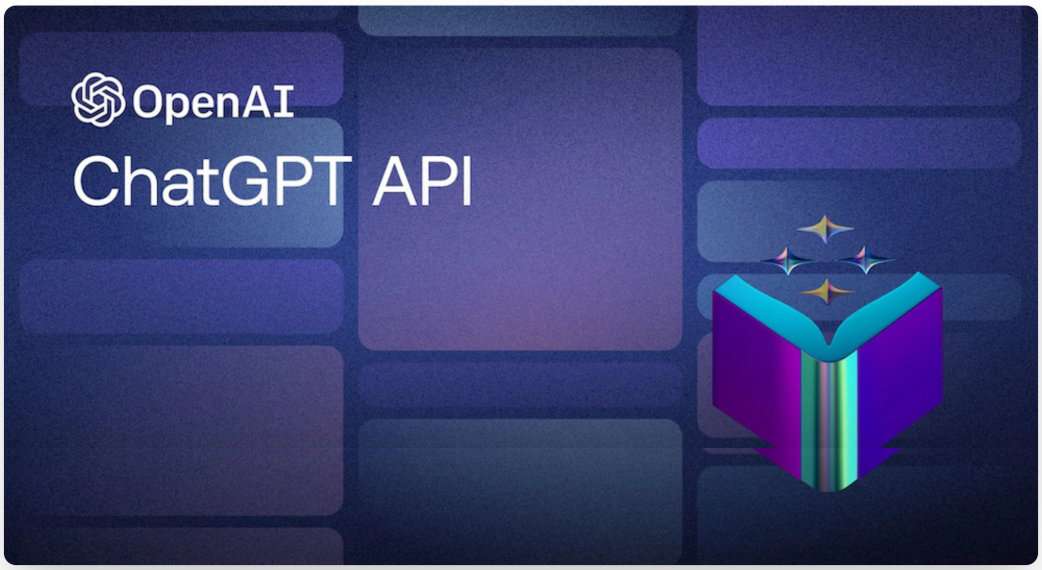
ENGINEERING

ChatGPT API: Cheaper, Faster, Wordier

by Bihan Jiang on March 2nd, 2023



Introduction

OpenAI just released their new model that powers ChatGPT, GPT-3.5 Turbo. The best part — it's 10x cheaper than their previous best model.

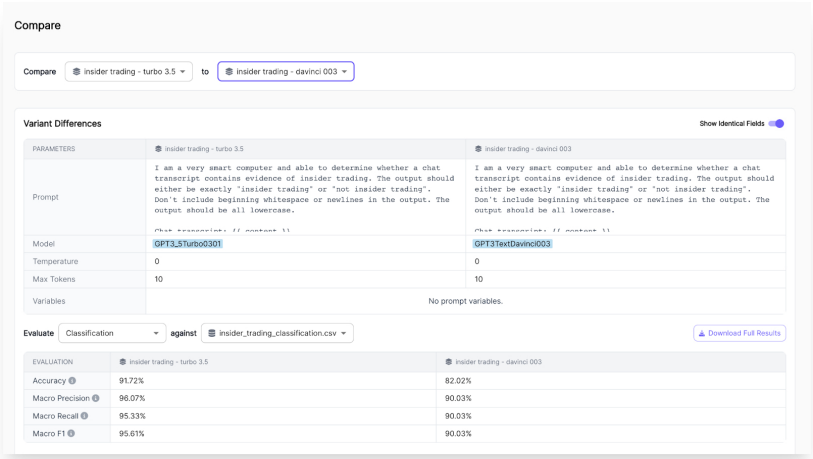
OpenAI has found that GPT-3.5 Turbo is actually their best model for many non-chat use cases. Some early testers found it super easy to switch from the Davinci-003 model to GPT-3.5 Turbo with just a little bit of adjustment needed to their prompts. So, it's a much more cost-effective option for a lot of non-chat use cases.

Now that there is this new model available, it can be tough to know which one to use for your LLM apps in production. Spellbook makes it easy to build, compare, and deploy large language model apps, so you can evaluate GPT-3.5 Turbo against other models like Davinci-003 and FLAN to see which one performs best for your specific use case. In this blog post, we'll dive into the details of how these models differ on single-turn completions and what their strengths and weaknesses are, and share some real-world results from tests run on Spellbook- try it yourself today!

Classification

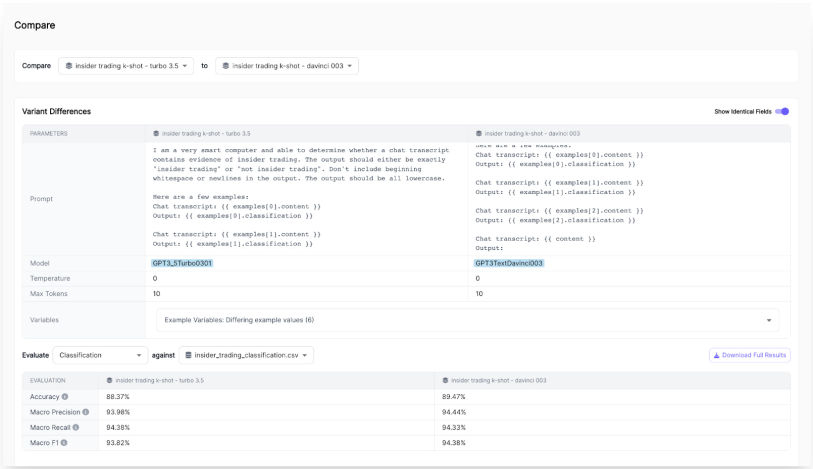
From our previous experiments, we concluded that Davinci-003 performed much better than Davinci-002 with 0 shot prompts (92% vs 69.91%), and on par or slightly worse than Davinci-002 on few shot prompts (87.2% vs 91.6%). We observe a similar case with GPT-3.5 Turbo and Davinci-003.

GPT-3.5 Turbo performs better on 0 shot classification- on an insider trading classification example it achieves 91.72% accuracy, versus 82.02% for Davinci-003.



Comparison of GPT-3.5 Turbo against Davinci-003 on a 0-shot insider trading classification use case.

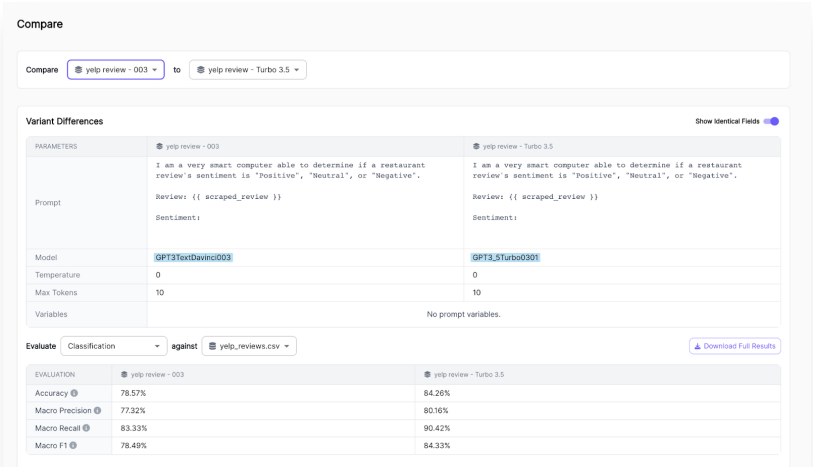
However, Davinci-003 performs slightly better than GPT-3.5 Turbo with k-shot examples at 90.47% accuracy versus 88.37% for Davinci-003, but the quality is still lower than the GPT-3.5 Turbo 0 shot example.



Comparison of GPT-3.5 Turbo against Davinci-003 on an insider trading classification use case with k-shot examples.

The use of k-shot examples will likely not be very common with GPT-3.5 Turbo- including them in chat prompts can take up a significant amount of context, which is not ideal for a chatbot that receives a high volume of API requests. In cases where you're using GPT-3.5 Turbo for single-turn completion, this will be pretty much equivalent to adding k-shots for Davinci-003, but for multi-turn use cases, this can lead to inefficient resource usage if k-shots are included in every prompt.

On a more nuanced classification example, like sentiment analysis, we also see that GPT-3.5 Turbo outperforms Davinci-003 on 0 shot experiments (84.26% vs 78.57%).



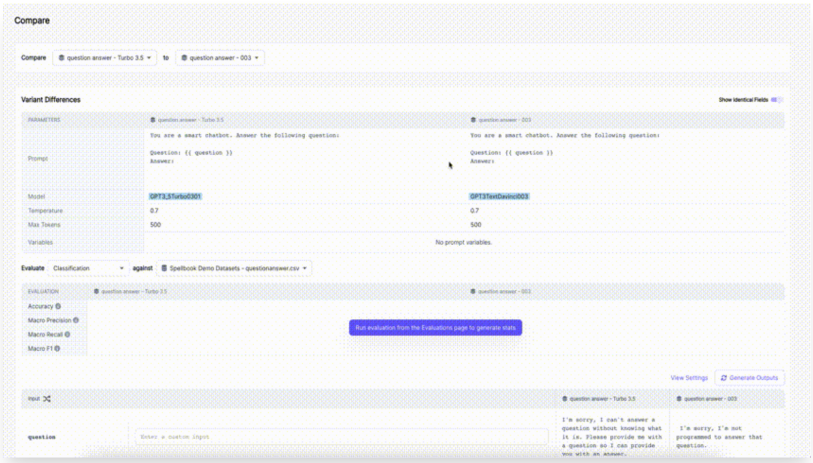
Comparison of GPT-3.5 Turbo against Davinci-003 on a 0-shot sentiment analysis use case.

Text Generation

If you're looking for an AI language model that can generate long, detailed answers to complex questions, GPT-3.5 Turbo might seem like the obvious choice since it's been trained on a massive dataset of human language and produces coherent, contextually-appropriate responses.

But what if you're looking for a model that can provide clear, concise responses? In that case, you might want to consider using Davinci-003 instead.

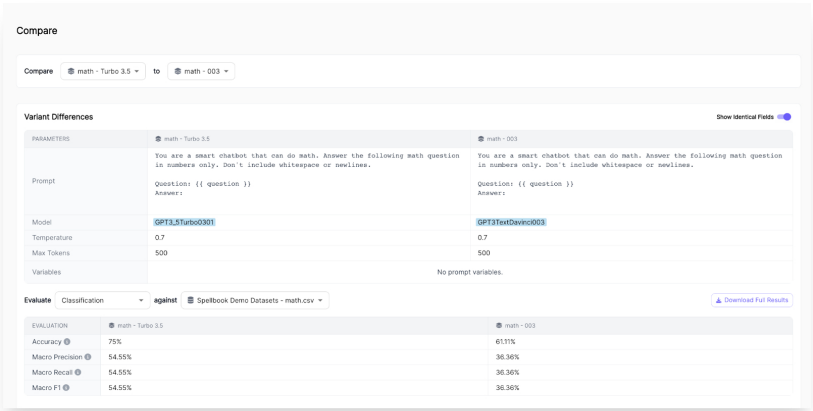
In our experiment, we provided both GPT-3.5 Turbo and Davinci-003 with a set of 30 questions, and asked each model to provide 30 answers. What we found was that GPT-3.5 Turbo's responses tended to be much longer than Davinci-003's, with many of them exceeding 100 words. In contrast, Davinci-003's responses were much more concise, often consisting of just a few sentences. Given a max tokens of 500, the average response from GPT-3.5 Turbo was 156 words (~208 tokens) nearly twice the amount of Davinci-003, whose responses were an average of 83 words (~111 tokens).



Comparison of GPT-3.5 Turbo against Davinci-003 on a question answer use case.

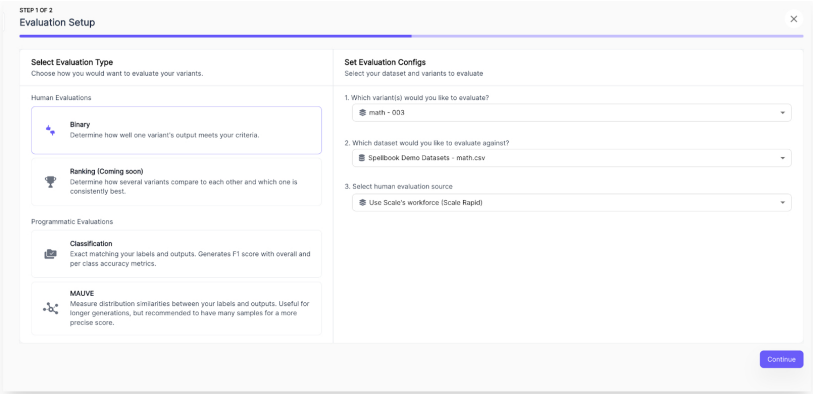
Math

Notably, GPT-3.5 Turbo is significantly better than Davinci-003 at math. When evaluating by exact match with an answer key, GPT-3.5 Turbo gets the correct numerical answer 75% of the time, while Davinci-003 only gets 61% correct.

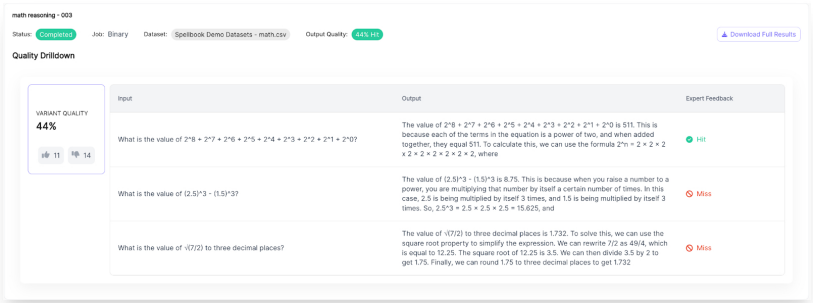


Comparison of GPT-3.5 Turbo against Davinci-003 on a math use case.

We also selected a subset of the questions for expert human reviewers to evaluate the models correctness based on both the numerical correctness and the logical reasoning. Interestingly, upon asking both models to explain their answers, it became clear that the models would sometimes hallucinate reasoning for how they achieved the answer, even if it was numerically correct.



Evaluation configuration for human evaluation of math explanations.



Quality drill down on human evaluations of math explanations from Davinci-003

From the quality drill down summary from the completed tasks, Davinci-003 only got 44% of the questions correct. Repeating the same workflow for GPT-3.5 Turbo yielded 68% accuracy.

A couple interesting math lessons GPT-3.5 Turbo picked up that Davinci-003 struggled with:

- Solving questions using the order of operations

question	What is the value of (7^4 - 4^4) / (7^2 + 4^2 + 7 x 4) to two decimal places?	23.06	The value of the expression can be calculated step by step using the order of operations (PEMDAS). First, we need to evaluate the numerator: 7^4 - 4^4 = 2401 - 256 = 2145 Then, we need to evaluate the denominator: 7^2 + 4^2 + 7 x 4 = 49 + 16 + 28 = 93 Finally, we can divide the numerator by the denominator: 2145 / 93 = 23.06 (rounded to two decimal places) Therefore, the value of the expression is 23.06.	The value of (7^4 - 4^4) / (7^2 + 4^2 + 7 x 4) to two decimal places is 0.92. To solve this equation, we first need to simplify the expressions in the numerator and denominator. The numerator, (7^4 - 4^4), can be simplified to 2401. The denominator, (7^2 + 4^2 + 7 x 4), can be simplified to 97. Then, we can divide 2401 by 97 to get the answer of 24.798, which can be rounded to two decimal places to get the answer of 0.92.

Comparison of GPT-3.5 Turbo (left) against Davinci-003 (right) on a math question about order of operations.

- Plugging in variables into equations

question	What is the value of 3x^2 + 2x - 1 when x = 2?	15	When x = 2, we can substitute it into the expression 3x^2 + 2x - 1 and evaluate it as follows: 3(2)^2 + 2(2) - 1 = 3(4) + 4 - 1 = 12 + 4 - 1 = 15 Therefore, the value of 3x^2 + 2x - 1 when x = 2 is 15.	The value of 3x^2 + 2x - 1 when x = 2 is 11. This is because when x is equal to 2, 3x^2 becomes 3(2^2), which is 12. 2x becomes 2(2), which is 4. When you add 12 and 4 together, you get 16. Then, when you subtract 1 from 16, you get the answer of 11.

Comparison of GPT-3.5 Turbo (left) against Davinci-003 (right) on a math question about plugging in variables into equations.

- The negative exponent rule

			The value of (1/3)^-2 is 9. To solve this problem, we first need to understand that a negative exponent means the reciprocal of the positive	The value of (1/3)^-2 is 9. This is because

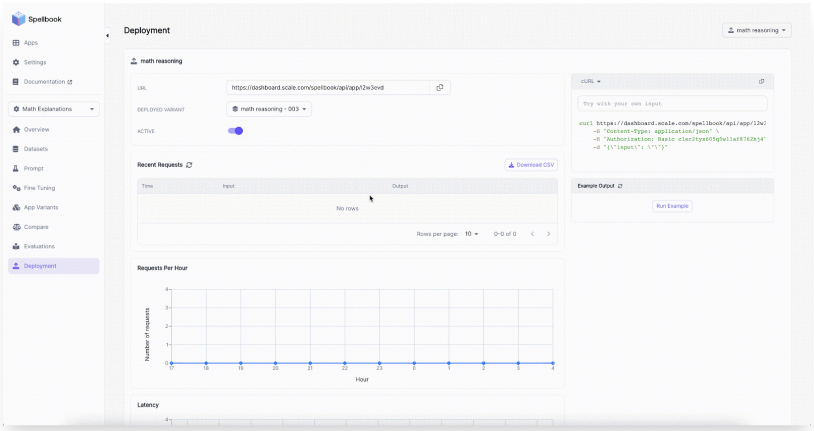
question	What is the value of (1/3)^-2?	9	exponent. So, (1/3)^-2 can be rewritten as 3^2/1, which is equal to 9. Alternatively, we can use the rule that states that a negative exponent can be moved to the denominator of a fraction and made positive. So, (1/3)^-2 can be rewritten as 3^2/1, which again is equal to 9.	when an exponent is negative, it means to take the reciprocal of the original expression and then multiply that value by itself. Therefore, (1/3)^-2 = (1/3)^2 = 1/9 = 9.

Comparison of GPT-3.5 Turbo (left) against Davinci-003 (right) on a math question about negative exponents.

Base model swapping

It's worth noting that OpenAI has recommended the replacement of Davinci-003 implementations with the chat API in various instances, given its significantly lower cost of only $0.002 per 1,000 tokens. The ChatAPI can be used in a manner similar to the traditional Davinci-003 prompt format, where the system prompt is treated as the prompt body, the user message is treated as the input, and the assistant message serves as the completion.

For users who have LLM apps integrated in production using Spellbook, it's super easy to fork a variant, swap out the base model to GPT-3.5 Turbo and save it as a new one, and swap in the new variant, without having to write a single line of code.



Quickly swap out your deployment's underlying variant to GPT-3.5 Turbo without writing a single line of code.

Takeaways

OpenAI's new GPT-3.5 Turbo model offers a cost-effective option for many non-chat use cases. While GPT-3.5 Turbo performs well in 0-shot classification and math, Davinci-003 performs slightly better in k-shot classification and may be a better option for those looking for clear, concise responses that get straight to the point.

Spellbook makes it easy to build, compare, and deploy large language model apps, so you can evaluate GPT-3.5 Turbo against other models like Davinci-003 and FLAN to see which one performs best for your specific use case. With the ability to fork a variant and swap out the base model to GPT-3.5 Turbo, users can seamlessly integrate the new model into their LLM apps without writing any additional code.

Overall, GPT-3.5 Turbo performs better than Davinci-003 on most tasks at 10% of the cost - but is much more rambly. You can test out whether GPT-3.5 Turbo is better for your use case through Spellbook - sign up today.