Blog

Technical Overview · 1 min read

Analytics for LLM Apps: The Metrics Developers Need to Track with the ChatGPT API

Learn the key metrics LLM app developers need to track for the ChatGPT API to optimize user experience, latency, and token costs. Discover how powerful analytics tools can help you make informed decisions.

Nico Acosta

• Published: May 3, 2023
• Last updated: May 9, 2023

Photo: Propel

As an LLM app developer, you're likely aware of the importance of tracking and monitoring the performance of your apps. But do you know which metrics are crucial to optimizing user experience, latency, and token costs? In this blog post focusing on OpenAI's ChatGPT API, we'll dive into the key metrics you need to track, how to derive valuable insights from these metrics, and the importance of powerful analytics tools to make informed decisions for your ChatGPT-powered LLM apps.

What's different about LLM apps?

When it comes to LLM apps and ChatGPT, there are two crucial differences that set them apart from other software applications. These differences have a significant impact on their costs and user experience.

LLMs consume tokens

A defining characteristic of large language models (LLMs) is that they consume tokens, and token consumption directly affects the cost of LLM apps. Token usage needs to be tracked to ensure that costs are managed and allocated appropriately.

Tradeoffs between token usage, latency, and user experience

In LLM apps, there are tradeoffs between token usage, latency, and user experience. By understanding these tradeoffs and monitoring key metrics, you can optimize your LLM app's performance, providing a better experience for your users.

Key metrics for LLM apps

To optimize your LLM or ChatGPT-powered app, focus on these five key metrics: token usage, latency, user feedback, conversation length, and “finished reason.” These metrics will be important, no matter if you build with OpenAI or another LLM provider.

Token Usage

Token usage is crucial because both input and output tokens are consumed during app usage. Different models have varying token costs, so it's essential to monitor and optimize token usage to manage expenses effectively. Token usage is a great proxy for the length of the prompt and the response.

Latency

More complex inputs can lead to increased response times, which can negatively affect user experience. Additionally, different models exhibit different latencies. By tracking latency, you can ensure a smooth and responsive user experience.

Feedback

Capturing user feedback is vital to gauging whether your app is providing value to its users. Feedback will vary depending on the app and its use case, so it's essential to tailor your feedback capture methods accordingly.

Conversation length

You'll want to track the number of turns in a conversation. This allows you to monitor and display to the user any trends in conversation length over time as you make changes to the experience, switch models, or tune parameters.

Finished reason

Finally, you'll want a metric that counts the occurrences of each possible value for the `finished_reason` property. This will help you identify how often users experience incomplete responses. Tuning the prompt and `max_tokens` parameters can help reduce the frequency of incomplete responses for your users.

From Metrics to Insights

To gain valuable insights from the metrics discussed above, you'll need a powerful analytics backend that allows you to slice and dice the data efficiently.

Key Drill-Down Dimensions

To better understand the metrics above, ensure that you capture the following dimensions from OpenAI's APIs as well as your app's metadata:

Dimension	Description	
prompt_tokens	The tokens consumed by the prompt.	
completion_tokens	The tokens consumed by the completion.	
total_tokens	The total tokens consumed.	
index	The `index` of the conversation.	
model	The name of the model used, e.g., `gpt-3.5-turbo`.	
prompt	The `prompt` that was used.	
temperature	The `temperature` value used.	
top_p	The `top_p` value that was used.	
max_tokens	The `max_tokens` value that was used.	

Dimension	Description
presence_penalty	The `presence_penalty` value that was used.
frequency_penalty	The `frequency_penalty` value that was used.
logit_bias	The `logit_bias` value that was used.
app_metadata	All your application metadata helps contextualize the ChatGPT request/response.
feedback	The user's feedback on the model's response.

By leveraging these dimensions, LLM app developers leveraging the ChatGPT API can fine-tune requests to improve user experience, latency, and token costs for their apps.

Leveraging Propel for ChatGPT-powered LLM apps

ChatGPT developers can greatly benefit from utilizing Propel, a powerful analytics backend with a GraphQL API and UI component library to collect and query their metrics data. Propel enables engineering teams to deliver high-performance customer-facing analytics without the need to scale or manage infrastructure.

Benefits

- **Metrics definition:** Developers can define the token usage, latency, and feedback metrics with all the dimensions described above.
- **Aggregations:** Propel will automatically aggregate data to provide developers with the insights they need.
- **Filtering and grouping:** Developers can filter and group metrics by any of the dimensions defined, giving them powerful analytical capabilities.
- **GraphQL API**: Propel's GraphQL API enables internal monitoring as well as customer-facing analytics features.
- **UI Component Library**: Propel's UI component library provides a set of pre-built data visualization and React components, equipping developers to create visually appealing and informative analytics dashboards for their ChatGPT-powered apps.

- **No Infrastructure Scaling or Management**: With Propel, developers can focus on building and optimizing their ChatGPT apps without worrying about the complexities of scaling and managing their analytics infrastructure.

By leveraging Propel, ChatGPT developers can access powerful analytics tools that help them track essential metrics, optimize app performance, and provide an improved user experience without the burden of infrastructure management. To learn more, you can read the docs or get started with a free Propel account.

Conclusion

By focusing on key metrics like token usage, latency, and feedback, ChatGPT developers can optimize their apps, ensuring a better user experience while managing costs effectively. Powerful analytics tools like Propel, with the ability to slice and dice data across dimensions, can further aid in making informed decisions both in internal tools and customer-facing apps, that lead to shorter time to market and better user experiences for ChatGPT apps.

Further Reading

For more insights and advanced techniques, check out these related articles:

- **Introducing the new Metric Report API: Powerful reports for any app with a single GraphQL request**
- **Propel UI Kit: Data visualization and dashboard React components**
- **Announcing the Propel Terraform Provider**
- **Introducing the new Webhook Data Source**

If you don't have a Propel account yet, you can **try Propel for free** and start building data apps.

Nico Acosta
Co-founder and CEO at Propel

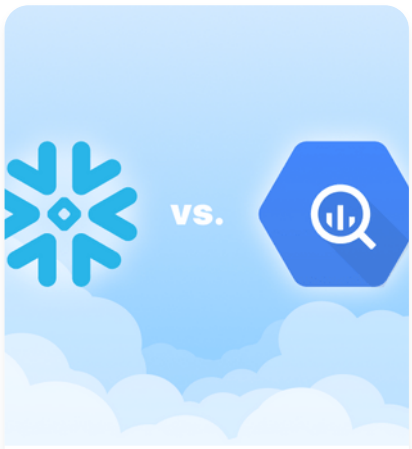
Related posts

Technical Overview

Managing Timezone Settings in Snowflake

Technical Overview

How to reduce Snowflake costs: A five-point checklist



Technical Overview

Snowflake vs. BigQuery: Key Differences and Similarities

Nico Acosta

May 25, 2023

Nico Acosta

May 16, 2023

Nico Acosta

April 20, 2023

Return to blog

You could be building more

Get a product demo to see how Propel helps your product dev team build more with less.

Try for free | Book Demo

Propel

Company

- Product
- Pricing
- About
- Contact
- Legal
- Snowflake

Resources

- Docs
- Changelog
- Blog
- Case Studies
- Podcasts

Follow us

- Twitter
- LinkedIn
- Github
- Reddit