



TEXTTUNNEL: EFFICIENT TEXT PROCESSING WITH OPENAI

We are excited to introduce you to our new Python package, `texttunnel`! The package is designed to enhance the process of interacting with OpenAI's API, saving time and money. By reducing inference runtime, it also reduces the CO2 footprint.

The package is fully open source and available on [PyPi](#) and [GitHub](#).

Use case

We developed `texttunnel` in response to challenges we faced when using the OpenAI API. It targets the following use case:

Suppose you possess a corpus of text data that you want to analyze using the GPT-3.5 or GPT-4 models. The goal is to perform extractive NLP tasks such as classification, named entity recognition, translation, summarization, question answering, or sentiment analysis. In this context, the package prioritizes efficiency and tidiness to provide you streamlined results. Our philosophy with the package is to make this particular use case efficient and easy. If your use case involves chaining requests, vector databases or using other models, we suggest [langchain](#) instead.

texttunnel features

Output Schema: Utilizes [JSON Schema](#) alongside OpenAI's function calling schema to define the output data structure.

Input Validation: Ensures well-structured and error-free API requests by validating input data.

Output Validation: Checks the response data from OpenAI's API against the expected schema to maintain data integrity.

Efficient Batching: Supports bulk processing by packing multiple input texts into a single request for the OpenAI's API.

Asynchronous Requests: Facilitates speedy data processing by sending simultaneous requests to OpenAI's API, while maintaining API rate limits.

Cost Estimation: Aims for transparency in API utilization cost by providing cost estimates before sending API requests.

Caching: Uses [aiohttp-client-cache](#) to avoid redundant requests and reduce cost by caching previous requests. Supports SQLite, MongoDB, DynamoDB and Redis cache backends.

Request Logging: Implements Python's native logging framework for tracking and logging all API requests.

Real world example: Classifying social media posts about cancer.

A client in the pharmaceutical industry was looking for insights about the challenges that cancer patients face in their treatment. We built a data pipeline to collect thousands of social media posts by cancer patients, caregivers and healthcare professionals.

To reduce time to insight, we opted to use GPT-4 instead of training a custom model for analyzing the texts.

First, we developed a categorization schema, telling the model which categories there are and how to discern them. This description fitted into the system message and output JSON schema. `Texttunnel`'s input validation ensured the inputs fit into the model's context window and the output validation guaranteed that the results neatly fitted into our database schema.

GPT-4 can become expensive to run, so we used `texttunnel`'s cost estimation throughout the project. We could directly see the cost improvements as we cut unnecessary words from the prompt. Caching also came in handy, as it allowed us to rerun requests without worry of having to pay twice.

Finally, we arrived at a concise prompt that guided the model to an accuracy of 93% when compared to a gold dataset labeled by human analysts.

At inference time, `texttunnel` enabled us to make the requests to the OpenAI API in parallel rather than sequentially. This reduced the wait time between the publish time of a new post and an update in the client's dashboard.

If your use cases are similar, we encourage you to try `texttunnel` and let us know what you think!

Open Source at TeamQ

At Q, we benefit greatly from the open-source community. Without the contributions of countless volunteers providing code and documentation, the capabilities and productivity in data science that we enjoy today wouldn't have been possible. We are thrilled to contribute `texttunnel` as our first open project.

We welcome and encourage pull requests, questions and bug reports on [GitHub](#). If `texttunnel` is useful for your work, please let us know on X (Twitter): [Q_InsightAgency](#) or on [LinkedIn](#).

Author: Paul Simmering