

# Classification Challenges And Advances: Navigating API Limits And Precision In Legislative Sentiment Analysis

*AP Research & Senior Research Project*

Ojas Chaturvedi

<https://github.com/ojas-chaturvedi>

---

This week was entirely dedicated to developing a classification script that effectively assigns each piece of legislation a label indicating whether it supports gun rights or advocates for gun control. I originally introduced this concept in my fourth blog, '[Data Collection Progress: Legislative Text Acquisition And Preliminary Classification Efforts](#)', but to summarize, this step ensures the analysis accurately reflects the specific context and nuances of each stance, leading to more precise and balanced insights into the legislative discourse surrounding gun policy. For the classification task, I utilized \$2,500 in tokens available through the OpenAI Researcher Access Program, applying them within any Large Language Model framework. This funding enabled me to employ the OpenAI API—a cloud service providing access to OpenAI's advanced machine learning models—allowing for the integration of sophisticated AI functionalities into my project. Initially, I sought to streamline the model's output to either 'control' or 'rights' for uniformity, ensuring consistent responses to identical inputs. After still not getting the one-word responses, I dived into the extensive documentation provided by OpenAI and discovered the role of the 'temperature' parameter, which influences the predictability of the model's responses. A higher 'temperature' setting yields more varied and imaginative outputs, while a lower setting results in more predictable outcomes. By adjusting this parameter to the minimum value of 0.0, I was able to standardize the responses, achieving consistent classification for each piece of legislation upon reevaluation.

Now, my classification script is up and running smoothly, with just one significant hiccup: the API's rate limit. This limitation is in place to stop anyone from overburdening the API and servers, whether accidentally or intentionally. There are specific caps on how many requests and tokens I can send in a day. Here, requests are the commands I issue through the API, and tokens are essentially chunks of text, roughly equating to 4 characters each. There are two primary constraints impacting my work: a daily limit of 200 requests and a token usage cap of 40,000 per minute. The daily request limit means I can process 200 legislative texts each day. With 1,560 texts to go through, it'll take about 8 days to complete. However, I'm considering expanding my project's timeline to start from as early as 1799 instead of 2001, depending on how Congressional English has changed over time, which would increase the volume of texts

needing classification and analysis. To navigate the 40,000-token-per-minute limit, I've devised a couple of strategies. Firstly, I'm using a Python library called TikToken to gauge the token count of each text. If a text exceeds 40,000 tokens, I bypass it for manual segmentation and classification later. This strategy addresses the issue of large texts. To manage the influx of tokens in multiple requests per minute, I've introduced a script sleep timer. Should I hit the rate limit, the script pauses for 60 seconds minus the duration of the previous requests. For instance, if my initial requests took 45 seconds and triggered the rate limit error, the program would take a 15-second break before attempting to process the text again. Below, you'll find the code for my classification script and the solutions I've implemented for these challenges.

Legislation classification script code:

[https://github.com/ojas-chaturvedi/NLP-Gun-Legislation/blob/master/data\\_collection/classification.py](https://github.com/ojas-chaturvedi/NLP-Gun-Legislation/blob/master/data_collection/classification.py)

In the following week, I plan to continue (and hopefully finish) running the classification script for all legislative texts between 2001 and 2023. Alongside this, I'll be tackling three additional tasks. First off, I'll wrap up my research into how Congressional English has changed over time, aiming to pinpoint a specific period with a consistent level of English that will serve as the focus for my project. Also, I plan to update the literature review section of my paper, as I've come across a lot more sources that touch on firearm legislation and the sentiment analysis fields. Given the significant shifts in my project approach, I also plan to update my methodology section with the new steps I have taken, and I believe it's important to revise the overall paper sooner rather than later. Next, I'll dive into developing the sentiment analysis models, aiming to get at least one fully up and running by the end of next week. Setting up the first model will likely be the trickiest part since it involves applying what I've learned about Python libraries recently and starting to create graphs to check the model's performance. However, I expect the setup for subsequent models won't differ too much from the first. By my practice Presentation & Oral Defense on April 11th, I'm aiming to have preliminary results from at least three sentiment analysis models ready to share. Of course, I'll also keep up with my duties in my internship and AP Research class, as usual.