**Creating a Network Graph from the Cairo Geniza:**

**1. Introduction to the Cairo Geniza**

[Brief introduction to the Cairo Geniza, its historical significance, and why creating a network graph from it is valuable.]

**2. Dataset Acquisition**

Over 35 years of research, S.D. Goitein created over 35,000 index cards detailing his notes on the Cairo Geniza. Of them, roughly 9000 were letters. Each index card describes a single fragment from the Geniza, forming the basis of the network graph I created.

[Insert image of index card example here]

These notecards were transcribed and made available online by the Princeton Geniza Lab. The dataset includes each document's PGPID (a unique sequential ID), written description, and document type (letter, legal, ketubah, list, etc.).

**3a. Data Cleaning and Name Extraction**

**Initial Approach: Named Entity Recognition**

I first attempted to extract names using named entity recognition (NER). However, standard NER tools struggled with patronymic-based names like "Nahray ben Nissim", often misidentifying parts of the name as separate entities.

**Refined Method: Regular Expressions**

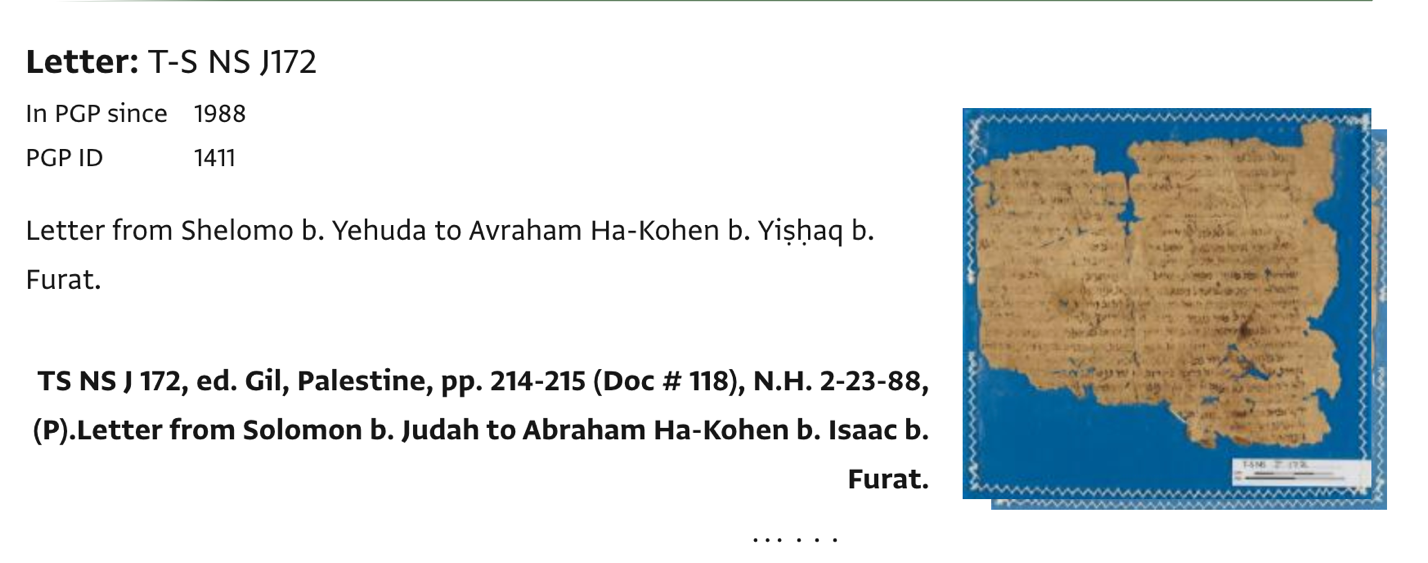
Fortunately, Goitein's consistent note-taking style for letters allowed for a simpler approach using regular expressions. The general format was:

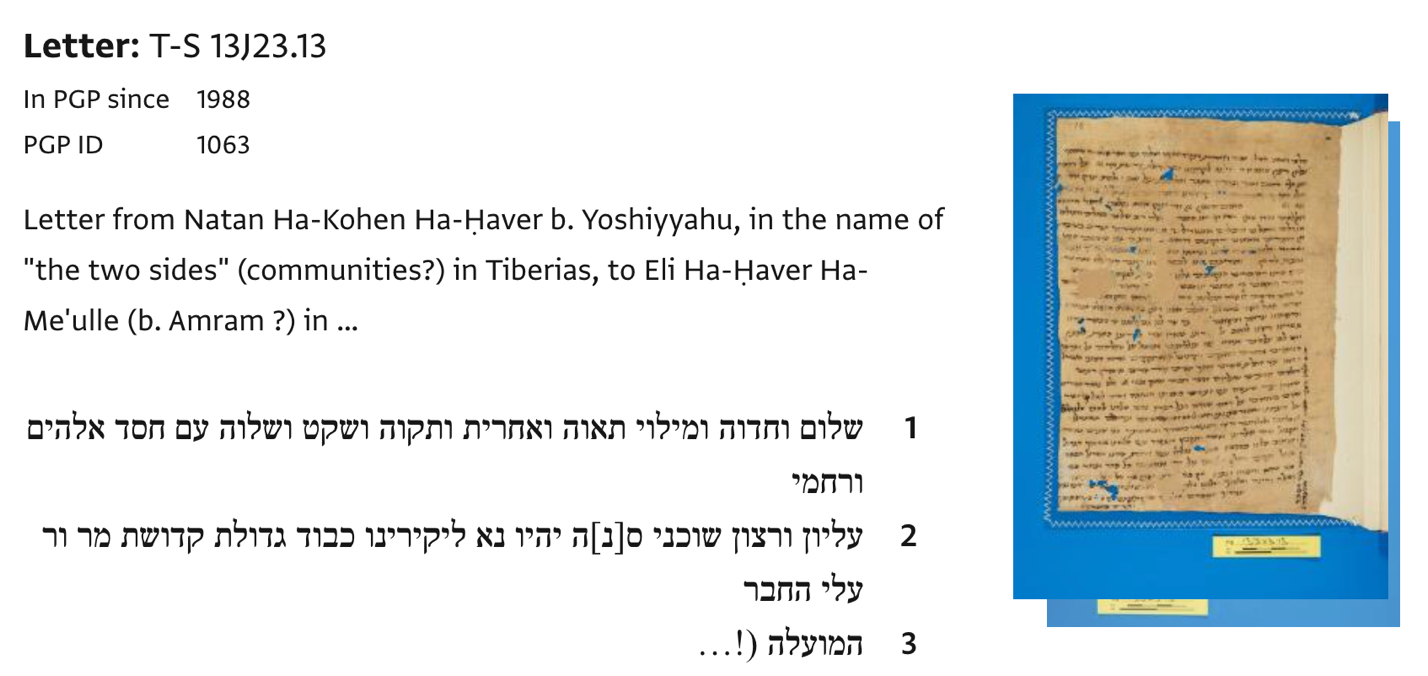
"Letter from [sender] to [recipient]"

This allowed me to extract senders and recipients using regex patterns:

* Sender: Text between "from" and "to"
* Recipient: Text after "to" until the next punctuation mark

Sometimes this worked perfectly without any issues.

PGPID 1411: “Shelomo b. Yehuda” “Avraham Ha-Kohen b. Yishaq b. Furat”

Sometimes, it required a little bit of cleaning by hand

PGPID 2588: “Yosef b

Sometimes, it did not work at all.

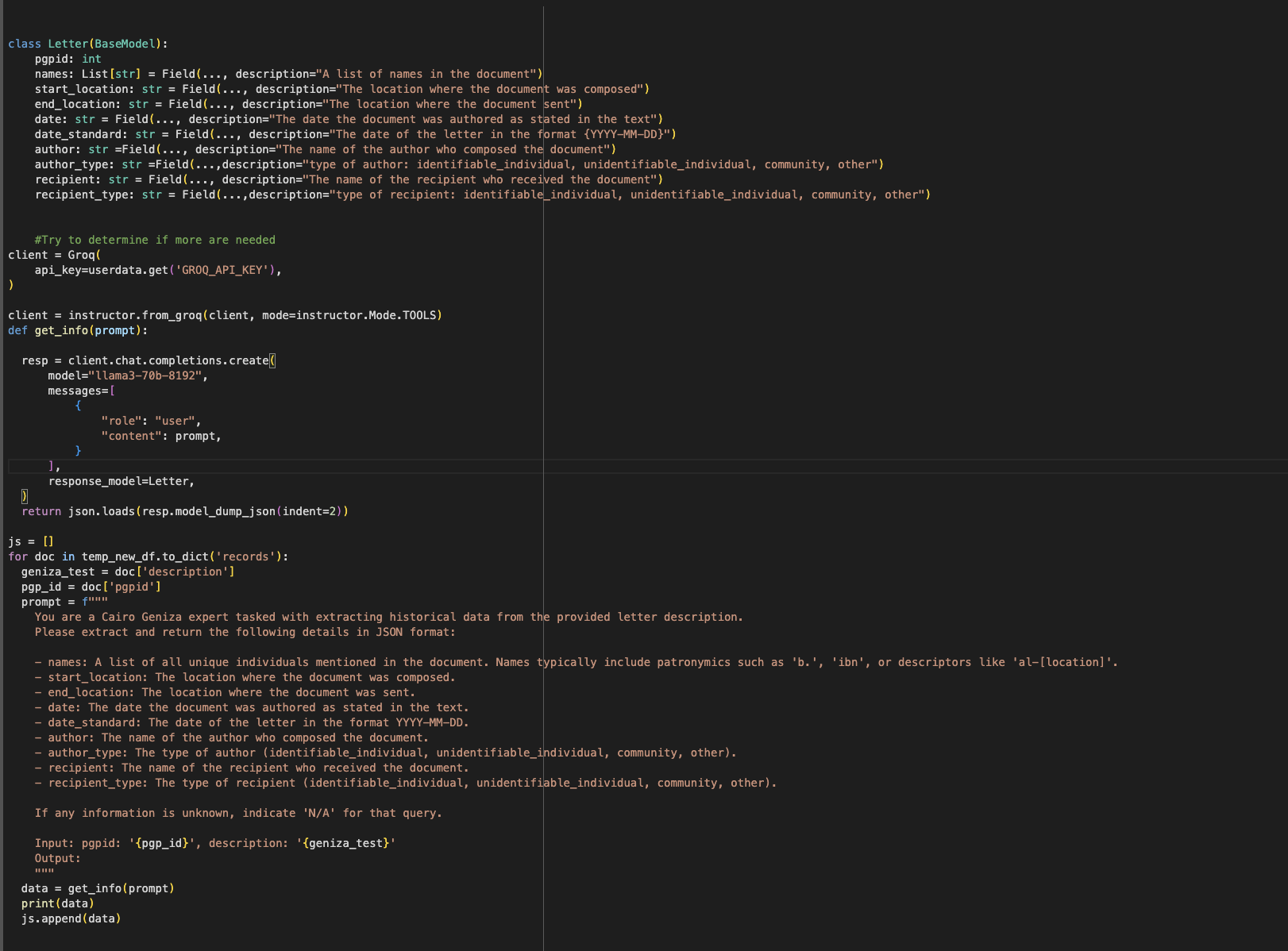


So no doubt, I missed out on letters which simply appeared to be completely empty from the point of view of the regex identifier.

**3b. Using a LLM to Get the Data**

I used a large language model to get the letter recipient alongside a bunch of other relevant metadata from the descriptions. It didn’t work so well with some other large language models and adding an example set actually hindered the finding of names because it got obsessed with adding the names of whoever wrote the text description (it kept on saying that Goitein wrote the letter. Luckily, this only appeared 8/4300 documents in my current version).

I did not trust the large language model to accurately determine which person was which so I had to use the same name correction and standardization process for both.



**Manual Cleaning and Standardization**

The regex approach required substantial manual cleaning:

1. Reviewing thousands of rows to ensure name consistency
2. Standardizing name spellings and honorifics
3. Identifying when different spellings referred to the same person

For example:

* "Musa b. Yishak b. Hisdai" and "Musa b. Yishak b. Hisda" likely refer to the same person
* "Yosef b. Ya'aqov b. Awkal", "Ibn Awkal", and "Abu l-Faraj Yusuf b. Yaaqov b. Awkal" all refer to one individual

This process was time-consuming and prone to errors, especially when dealing with similar names and varying honorifics.

There isn’t a very cool process. I first wrote a script to find all of the unique names in the output data. Then, I used a fuzzy name matching technique to determine what names were similar. This worked a bit for some very similar names (e.g. Ya’aqov vs Yacub) but was pretty unsuccessful in anything more complicated. For the most part, I had to go name by name and find similar names and then look at their original note to determine if they were the same person or a different person.

**4. Creating the Network Graph**

After standardizing the names, I used d3.js to build a network graph representing letter senders and recipients. With assistance from Ben Johnson at the Princeton Geniza Lab, I created a small website to host the graph. Later, I added functionality to view letters by clicking on nodes (individuals) or links (connections between individuals).

**5. What Did I Learn From All of This**

There, unfortunately, is not a real replacement for cleaning data by hand. As much as I would like to automate as much of this process as possible, it simply doesn’t really work. It took me a long time and I am certainly still missing things but there is no real alternative.