For my semester project, I chose option 2. I like the idea of creating data from what most people would call “nothing”.  
The Wagner Group is a Russian paramilitary organization often referred to as “Putin’s private army.” Being able to document members of the organization and their activities could go a long way toward countering their objectives and maintaining peace in unstable regions.

To meet the requirements for the project, I mentally broke the task into 3 different sub-tasks. First, how to identify content discussing the Wanger Group. A private army does not advertise their existence. For this task, I used the open-source python library GNews. GNews supports Google News searches and can search for individual keywords or topics. At the moment, the library is limited to returning 100 results. It does support date ranges, so it would be possible to loop over individual time frames to get more than 100 results.

I stuck with this limitation for the project for the following reason. I look at this project as a proof-of-concept. If I can write code to read, parse, and extract important data from a single piece of text content, I can do it for 100, or 100000.   
  
The second step was to parse the identified articles and extract the names, places, and dates. The GNews results are just pointers to the articles. They contain a summary, but not the complete article. To get that, it has a method call get\_full\_article() which uses the newspaper3k Python library under the hood. I had problems with some sites returning 403 errors that could be fixed by changing the user-agent used by GNews, but I could only do that using the Article class from newspaper directly. So, I use the newspaper3k library to download the article text from the GNews results.  
  
To extract the required data from the text, I used spacy and SUTime. The Article class of newspaper3k supports an nlp method, but it does not recognize as many entities as spacy does. And spacy can recognize what is supposed to be a date, but not the date itself. For example, it will call ‘three weeks ago” a DATE but cannot determine the year/month/day format necessary.

SUTime can. The Python SUTime library is a wrapper around the Stanford Natural Language Processing group’s Java implementation of SUTime, a library for “recognizing and normalizing time expressions” [link](https://nlp.stanford.edu/software/sutime.html) SUTime can take the phrase “last Thursday” and identify it as a DATE as can spacy. But SUTime can make sure it has the correct parts. Dateparser confirms this and creates the date string to store in the entity collection. The required java jar files are too large to commit to GitHub, but the provided pom.xml can be used with Maven to download and install them.

Once all the entities were identified, I used spacy to identify all GPE (geo-political entity) and PERSON entities, if the text of the PERSON entity was at least 2 words (first and last name). Further references only include that last name, which is not very helpful. I used dateparser to confirm that the dates identified by SUTime were full dates, containing month, day, and year.

The last step is to take the data identified from the individual articles and combine them into one large dataset. I used Pandas for this. What was interesting here was to decide which of the more than 1500 entities I had identified should be included. A single place referenced a single time in a single document is probably not very significant. So, I set the threshold to be included at 2.5%, rounded down. I.e. an entity needed to be referenced by at least 2.5% of the documents to be included. For my 100-run testing, that meant the same person, place, or date needed to be referenced by at least 2 of the 100 articles.

I am happy with what I am submitting but know it could be more complete. I did not include a database to save prior results, and since the top 100 results change over time, resultsets can change slightly from day to day. I do believe the results are beneficial and can scale up to thousands and 10’s of thousands of documents. And outside of the dependence on GNews to identify the URLs, the rest of the code should be able to handle online text content from any source.