Step 1: Extract entities, axiomatic statements, and non-axiomatic claims from the document

1.1. Extract entities using spaCy

* Install the library: pip install spacy
* Download a language model: python -m spacy download en\_core\_web\_sm
* Write a Python script to read the document, load the model, and extract entities.

1.2. Identify axiomatic statements and non-axiomatic claims using GPT-4

* Obtain access to the OpenAI API.
* Write a Python script to send the document's text to the GPT-4 API and process the results to identify axiomatic statements and non-axiomatic claims.

Step 2: Organize and format the extracted data

2.1. Structure the extracted data

* Create a Python dictionary or a similar data structure to store the entities, axiomatic statements, and non-axiomatic claims.
* Organize the data according to your desired hierarchy and relationships.

2.2. Format the data with markdown classifiers

* Write a Python function to convert the structured data into markdown format, with the appropriate classifiers for each element.

Step 3: Add extracted data to a spreadsheet

3.1. Use the Google Sheets API to add entities, axiomatic statements, and non-axiomatic claims to a spreadsheet

* Set up a Google Cloud Platform project and enable the Google Sheets API.
* Install the required libraries: pip install google-auth google-auth-oauthlib google-auth-httplib2 google-api-python-client
* Write a Python script to authenticate and interact with the Google Sheets API to create and fill the spreadsheet with the extracted data.

Step 4: Create a wiki from the extracted data

4.1. Choose a wiki platform

* Research available wiki platforms, such as Notion, MediaWiki, or DokuWiki, and choose the one that best suits your needs.

4.2. Set up the wiki API

* Obtain the necessary API keys and install the appropriate libraries for the chosen wiki platform.

4.3. Create pages programmatically

* Write a Python script to interact with the wiki API and create pages for each entity, filling them with the markdown-formatted data from Step 2.

Step 5: Automate the process with Zapier

5.1. Create a new Zap in Zapier

* Set the trigger to be when a new document is added to a specific folder in Google Drive or Dropbox.

5.2. Set up the "Code by Zapier" action

* Use the "Code by Zapier" action to run the Python script from Steps 1-4, or set up a webhook to trigger the script on your server.

5.3. Configure the trigger and action

* Adjust the trigger and action settings according to your needs.

While this revised guide provides more granularity, it is still an overview of the process. As you work on the project, you may need to consult documentation or tutorials for specific tools and APIs to address any challenges that arise.