Question 1

For my web crawler, I started with the default Scrapy settings. This ensured that I was following any requests given in the robots.txt file for each site. However, I did not enable the autothrottle, as I was not aware of this setting at the time of crawling. In the future, I will aim to enable this feature for crawling in order to avoid hitting a web server too frequently with requests.

Question 2

My lexical extractor was rather manual compared to my syntactic extractor. For the birthplace, I wanted to find words surrounded by phrases that might hint at where an author was from. I searched for tokens with phrases like “born in”, “grew up in”, “birthplace is”, and more. Then, I confirmed that the following terms were titlecase, as a city or state would be, and extracted the following words until they were no longer titlecase. I previously tried search for sequences that followed the City, State pattern (i.e. Los Angeles, California), but found that I was picking up too many irrelevant proper nouns. Thus, I changed my rules to search for specific phrases. I also searched for nationalities that ended in common nationality suffixes. I then matched these to the relevant countries to pull out information if a city or state was not found. For genre, I also searched for specific words, such as fantasy, sci-fi, literature, fiction, etc. and pulled those words into a list of genres for the author. Given that most genres are rather specific words, I felt this accomplished the goal decently well. For notable books, I searched for terms such as “series” or “best-seller”, and then tried to find titlecase words surrounding them. This was definitely the hardest lexical extraction, as without entity naming, it was difficult to find titles. For awards, I searched for title case words with the words “award”, “prize”, or “medal” nearby. These words, combined with titlecase, were pretty dead on for this category. Finally, for education, I searched for titlecase words surrounding words like university or college, as these were indicative of a title of a school. After I extracted these words + phrases, I inputted several rules to trim the extracted info into the data I was search for, and then added it all back to the initial dataframe I was working with, which was later converted into a json format and outputed into the jsonl file.

Question 3

For the syntactic extractor, I relied heavily on the name entity recognition to pick out important data points. For birthplaces, I identified the first entity with the GPE label as the birthplace, as typically the first city or state mentioned tended to be the birthplace of the author. I tried briefly to combine the labels with some dependency parsing to see if I could tie the labeled places back to a verb phrase such as born in or grew up in, but was unsuccessful in implementing this strategy. Thus, I relied on the first entity with the label. For genre, I went to a token level instead of entities. I tagged each token with a POS, and then searched out for adjectives or nouns that also had a list of words in their head text. This list of words included terms such as “book”, “novel”, “works”, etc., and the goal was to find an adjective that was describing one of these words to find relevant genres. For notable books, I mainly relied on entities tagged with the “work_of_art” tag that did not include the words “award” or “prize”. For awards, I searched for entities with “award” and “prize”. This was much more successful than my lexical extractor because I was able to identify names of awards at an entity level. Finally, for education, I searched for entities with the label “org” that contained words such as “university”, “college”, or “school. I then outputted these into the json file in the same process as the lexical extractions. The syntactic extractor showed considerably better results.