June 14th Meeting

Intro

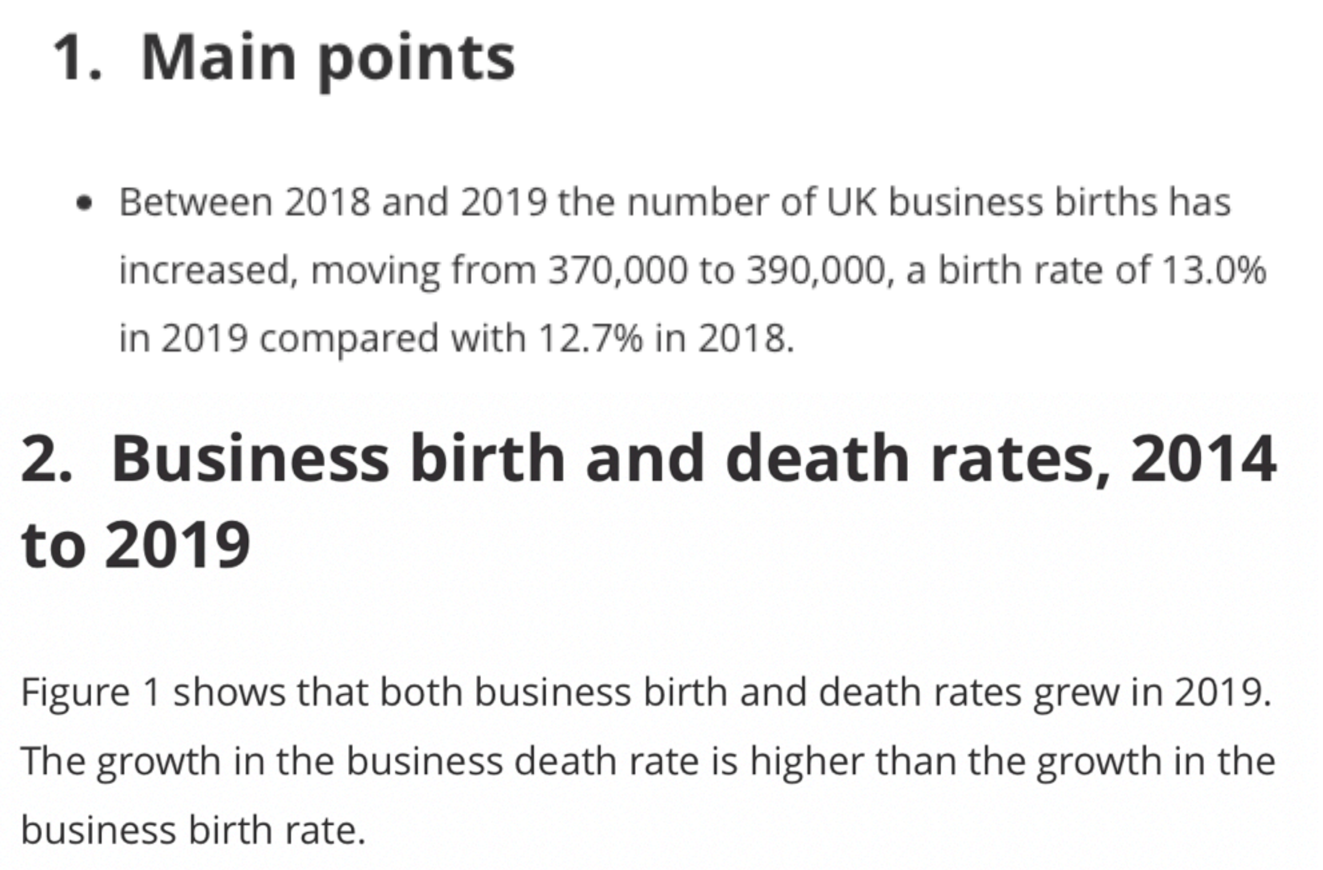
I completed my exams! I took a much needed break, celebrated my 25th birthday and have started working on my dissertation. I am applying for jobs too and that was somewhat a time sink... hopefully they'll like me!

Reading

1. Understanding tables with intermediate pre-training (Eisenchlos et al, https://arxiv.org/abs/2010.00571)
2. Joint verification and reranking for open fact checking over tables (Riedel et al, https://arxiv.org/abs/2012.15115)
3. Logical Natural Language Generation from Open-Domain Tables (Chen et al, https://arxiv.org/abs/2004.10404)
4. Logic2Text: High-Fidelity Natural Language Generation from Logical Forms (Chen et al, https://arxiv.org/abs/2004.14579)
5. TAPAS: Reasoning over tables with intermediate pre-training (Muller, Eisenchlos et al, https://arxiv.org/abs/2104.01099)
6. Open Question Answering over Tables and Text (Chen et al, https://openreview.net/forum?id=MmCRswl1UYl)
7. Open domain QA over tables via dense retrieval (Herzig, Eisenchlos et al, https://arxiv.org/abs/2103.12011)

Coding

1. I've been reading the API documentation and also contacted ONS
    - The documentation isn't particularly thorough; I am getting the hang of it by trial and error
    - I did obtain a contact email for questions but they are not responsive, as expected
2. I wrote a quick scraper that can parse bulletins
    - requests, beautifulsoup, parses and stores in a 'database' (dictionaries)
    - I extract: title, links, relevant publications, relevant datasets, metadata
    - I took the liberty to perform some preprocessing
        - Removed `|` character, artifact of how contact details are stored on the page
        - Removed references to Twitter or to the ONS (these were referrals for reader to check-out other socials)
        - **There are sentences about rounding errors, how averages are calculated. I started writing catches to remove them but I am not sure how it will scale**
        - Removed whitespaces and urls
3. Reading the *relevant publications* requires some further processing which I started writing
    - I am scraping text found in both `<li>` and `<p>` html tags



1. Main points

- Between 2018 and 2019 the number of UK business births has increased, moving from 370,000 to 390,000, a birth rate of 13.0% in 2019 compared with 12.7% in 2018.

2. Business birth and death rates, 2014 to 2019

Figure 1 shows that both business birth and death rates grew in 2019. The growth in the business death rate is higher than the growth in the business birth rate.

    - Those in list form, are sentences in a usable format without much preprocessing required
    - Those in paragraph form, require us to design a scalable solution to generate the queries
    - When (or if) should I focus on writing a solution to extract queries from the paragraph style text?
        - Difficulty would be in identifying what senteces we should drop?
4. Reading the *relevant datasets* you quickly observe that style and format is inconsistent
    - Initially I thought this was going to be a bigger problem
    - But reading the papers I see that **Table Linearization** should be sufficient, assuming we can identify WHERE the relevant data starts
    - Tables don't necessarily start on the same row number, so you need to identify the correct position.
    - (The API can be used to query specific observations/ variables so we can **possibly** utilise that approach?)
    - What if table has multi-level column-headers? Would Linearization be consistent for tables that don't have this?

Key Observations

1. The API does not provide a method to find publications from datasets. You need to read the website and scrape the publications, then you need to call the API and find the dataset with some particular ID.
    - I need to understand how to link the publications with the datasets.
    - If this isn't done right it will surely come back to bite me...
    - I have some experience building a retrieval indexer, maybe I should take that approach?
        - Retrieve a number of publications
        - Find the relevant datasets that are used
        - Find their IDs and store them in an index
        - You would then call the publication and it would provide a dataset ID you can pass to the API and progress?
    - Thoughts?
2. Would you argue that sentences from the "Main Points" list style (example above) are sufficient for our training data? We can use those to do our counterfactual generation.
    - This relies on the ONS website design purely. It does not scale to other websites, what happens if they decide to change it?
    - Are these issues I need to have in mind, in terms of how generalisable my research is?
    - If we want to use the paragraph style sentences, then I need to understand how I can remove sentences that do not refer to a dataset/ table. (see below, the publications are full of these)
        - *"Following a classification review, we concluded that the ERMAs placed TOCs under public sector control."*

1. Would it make sense to start by focusing on sentences that contain numerical statements? I can think of some 'hacky' ways to generate a training dataset fast. And we wouldn't need to work with counterfactual data generation
    - Eisenchlos et al. utilise other rows in the table to "scramble" and generate counterfactuals
    - Xiong et al. go the MASK approach and switch words of the same grammar family

Next Steps

1. Connect publications to datasets
2. Understand and implement preprocessing needed to get at a **clean** training set

In [ ]: