**Investigating How COP26 Reflected Global Demand, Linguistically**

**Abstract**

The goal of this project was to leverage unsupervised learning methods to determine major topics of attention at the COP26 Glasgow Climate Conference, so that these topics could be compared to the results of the 2021 UNDP People’s Climate Vote. I worked with data scraped directly from the UNFCCC’s website, and applied a number of different NLP models to discover topics, which ultimately suggested little linguistic overlap between the texts of the UNFCCC and the People’s Climate Vote. After refining models, I built an interactive ScatterText plot to visualize this effect and communicate my results.

**Design**

This project originates from my own curiosity regarding the democratic processes of the UN, particularly as they related to the efforts to mitigate and adapt to global climate change. The data is provided by the UNFCCC, and presents in two major categories: “speeches” and “reports”. All texts were in English and produced directly from the Glasgow Climate Conference. Identifying granular topics within this corpus would allow comparison between discussions and reports from Glasgow against the wishes of the global public concerning climate change, which are (likely) most obviously identified by the People’s Climate Vote. It has long been suggested that the UN is not very well connected to public sentiment; if such a disconnect exists, an analysis such as this could help identify how to close the communication gap.

**Data**

The dataset contains 422 [UN documents,](https://unfccc.int/documents?f%5B0%5D=conference%3A4301&f%5B1%5D=conference%3A4301&f%5B2%5D=conference%3A4301&f%5B3%5D=conference%3A4301&f%5B4%5D=language%3AEnglish) plus one more for the People’s Climate Vote, for a total of 423 source documents. These texts collectively contain roughly 1.2 million words across 4,000 pages, which were split into 23,024 lemmatized documents. The data is entirely textual in nature, though due to the large amount of report-type documents from the UNFCCC, there are a large amount of digits and roman numerals present.

**Algorithms**

*Web Scraping*

1. Create a list of all webpages to be scraped

2. Initialize dual webdrivers with Selenium

3. Access a given page of pdfs to download (webdriver1), and create a list of those links

4. Loop through the list of page links to download each pdf to the local machine (webdriver2)

*Text Processing*

1. Create a list of strings to identify ‘report’ files in the complete directory so that reports and speeches may be separated.

2. Loop through all files in the directory, sorting documents based on their format

3. Tokenize reports and speeches with separate functions.

*Topic Modeling*

1. Combine the two tokenized corpora into one.

2. Vectorize the corpus with CountVectorizer and TfidfVectorizer, using initial-guess parameters

3. Initialize LDA model to discover “rough topics” and use these to fine-tune vectorizers.

4. Apply a CorEx model to determine unanchored topics within the corpus.

5. Apply a second CorEx model with anchor words derived from the People’s Climate Vote text.

6. Use these outputs to continually refine the vectorizers, *especially the stop words.*

7. Parse the People’s Climate Vote and create a single dataframe with a ‘source’ column for ScatterText comparison.

8. Create ScatterText visualization with ‘source’ as the categorizer.

9. Apply PCA with 3 components to graphically verify ScatterText results.

*Scores*

Ultimately, there was very little overlap between the topics of UN documents and those contained in the People’s Climate Vote. Most notably, CorEx with anchor words indicated that fully half of the generalized words I selected were not present in the UN corpus. Additionally, reducing the dimensions of the UN corpus failed to provide any meaningful explanations of the variance within the corpus: three components explained ~11.25% of the variance, and even 30 components could only explain ~36%.

**Communication**

In addition to the slides and visuals presented, this project will be available on my personal GitHub. This will include all slides and all ScatterText visualizations generated at this point in the process.