

Thursday, 19

Rochester Institute of Technology

Computational Analysis of Gender Perception in the UK Before and After Women's Suffrage

PRESENTED BY
Shaista Syeda

FACULTY ADVISOR
Ashiqur R. KhudaBukhsh





Objective

In the UK, It was not until the Equal Franchise Act of 1928 that women over 21 were able to vote and women finally achieved the same voting rights as men

Goal

Gain an understanding of how women's representation has evolved over time



- Performed Data Segmentation and have a total of 6 files
- Have tokenized the files
- Performed Basic Statistical Analysis
- Performed Keyword Analysis
- Analysis of Male and Female Pronoun Count



- Generated male and female pronoun counts for each year
- Trained the data on Word2Vec
- Generated Word Analogies
- Generated similar words to Woman and Man
- Tried training the data on FastText Model

xssti / Analysis-of-gender-perception

[Code](#)
[Issues](#)
[Pull requests](#)
[Actions](#)
[Projects](#)
[Wiki](#)
[Security](#)
[Insights](#)
[Settings](#)

Analysis-of-gender-perception
Public

[Pin](#)
[Unwatch 1](#)
[Fork 0](#)
[Starred 1](#)

[main](#)
1 branch
0 tags

[Go to file](#)
[Add file](#)
[Code](#)

About

xssti Pronoun count notebook
d677b43 13 hours ago 47 commits

	ADS-2	Pronoun count notebook	13 hours ago
	Analysis_GP.ipynb	Code file in .ipynb format	10 months ago
	README.md	Update README.md	10 months ago
	Sentiment_Analysis.ipynb	Word Cloud Code file	10 months ago
	WordCloud.ipynb	.ipynb file to merge CSVs and create a word cloud	10 months ago
	analysis_gp.py	Initial Analysis on the data	10 months ago
	drive-download-20221211T043627...	Sphinx Generated Files and html pages	10 months ago
	drive-download-20221211T043638...	Sphinx Generated Files and html pages	10 months ago
	scraping_articles.py	Data Extraction code	10 months ago
	scraping_urls.py	Data Extraction code	10 months ago
	urls.csv	Data Extraction code	10 months ago

No description, website, or topics provided.

[Readme](#)
[Activity](#)
1 star
1 watching
0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Analysis-of-gender-perception / ADS-2 /

xssti Pronoun count notebook

Name



..



Data



Jupyter-Notebooks

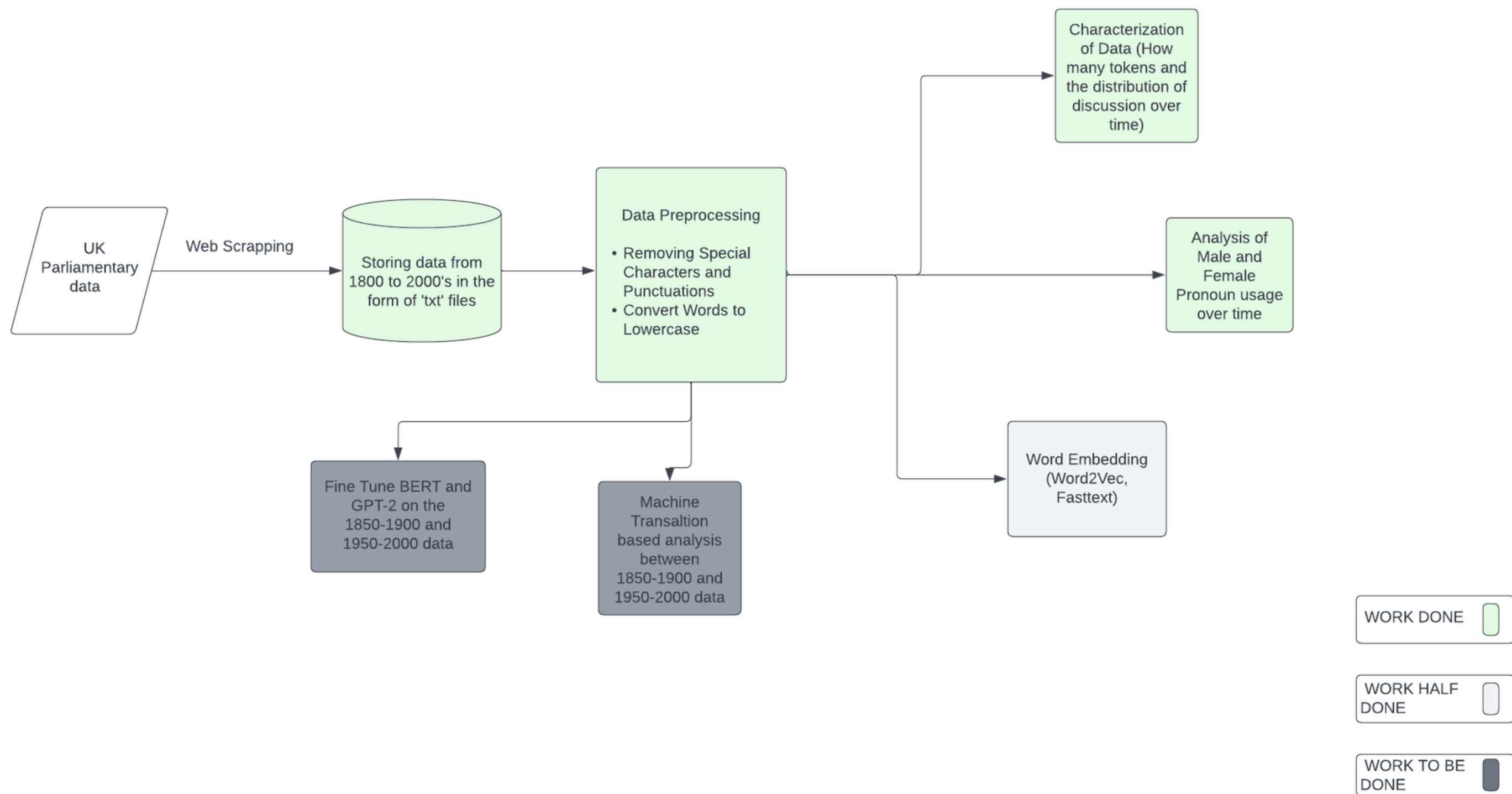


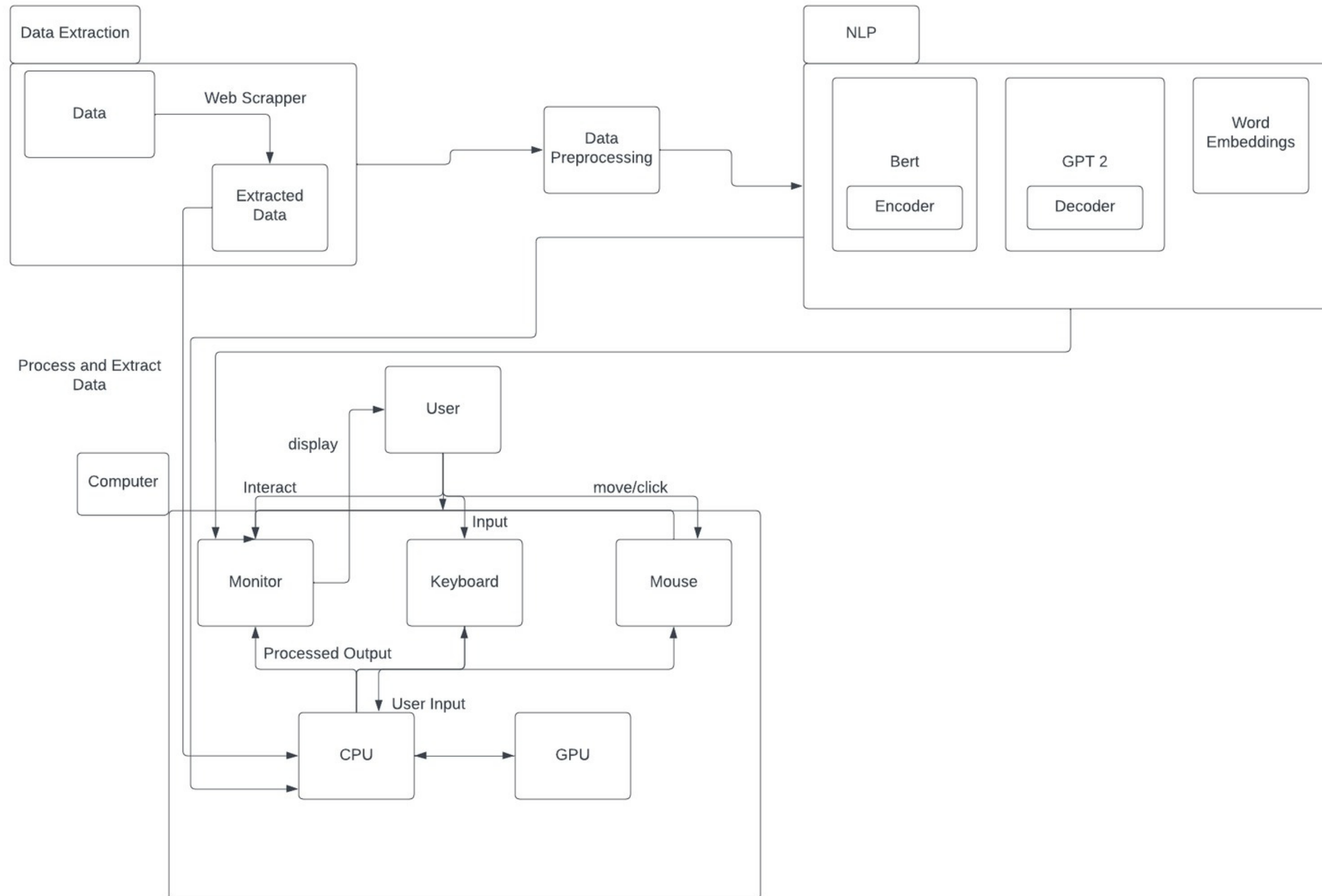
Models



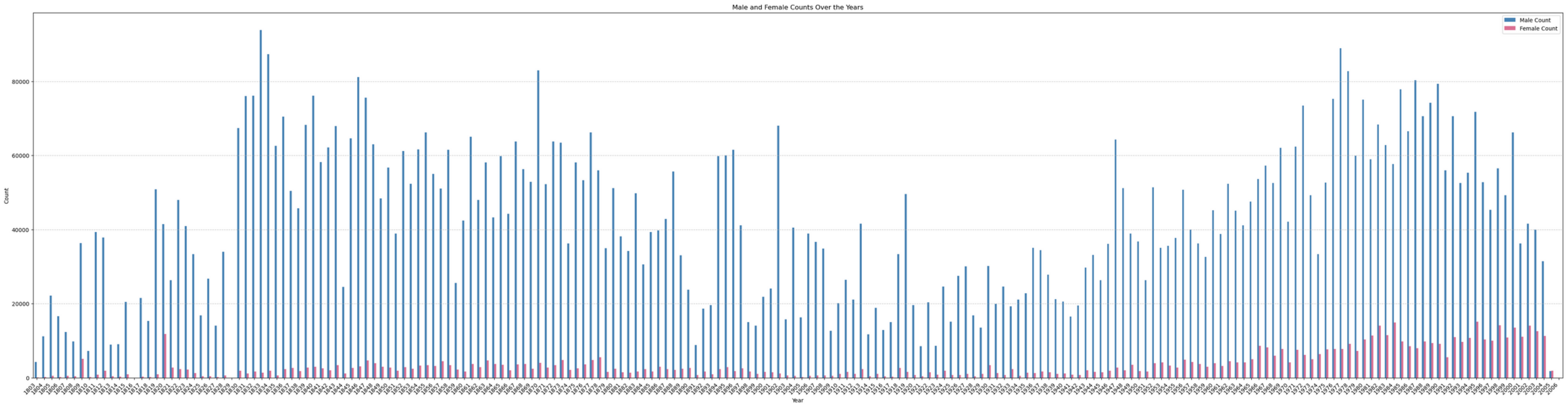
Presentation

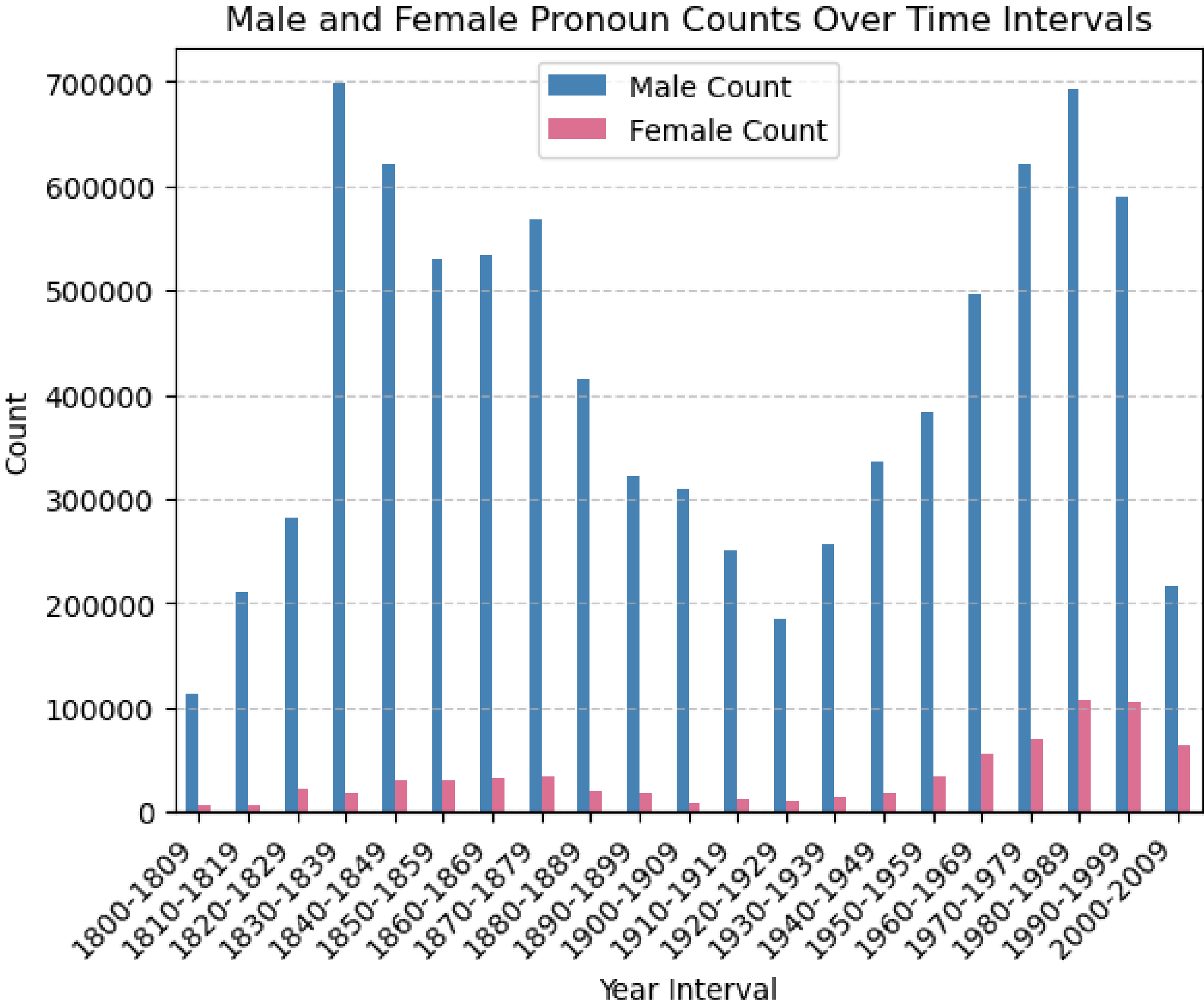
Github Link – <https://github.com/xssti/Analysis-of-gender-perception/tree/main/ADS-2>





VII Results as of now - Gender Pronoun Count





Words similar to WOMAN

```
# Find words similar to the word woman  
similar_words = model.wv.most_similar('woman', topn=5)  
print(f"Similar words to 'woman': {similar_words}")
```

Similar words to 'woman': [('men', 0.5933540463447571), ('female', 0.58524489402771), ('sex', 0.5548138618469238), ('wife', 0.5384922623634338), ('girl', 0.5375543236732483)]

Words similar to MAN

```
# Find words similar to the word man  
similar_words = model.wv.most_similar('man', topn=5)  
print(f"Similar words to 'man': {similar_words}")
```

Similar words to 'man': [('men', 0.6600674986839294), ('someone', 0.5912762880325317), ('husband', 0.5009009838104248), ('person', 0.4972824454307556), ('wife', 0.49441832304000854)]

Word Analogies

```
#word analogy
analogy_result_1850 = model_1850.wv.most_similar(positive=['men', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result for the year 1800-1850: {analogy_result_1850}")
#word analogy
analogy_result_1900 = model_1900.wv.most_similar(positive=['men', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result for the year 1851-1900: {analogy_result_1900}")
#word analogy
analogy_result_1950 = model_1950.wv.most_similar(positive=['men', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result for the year 1901-1950: {analogy_result_1950}")
#word analogy
analogy_result_1970 = model_1970.wv.most_similar(positive=['men', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result for the year 1951-1970: {analogy_result_1970}")
#word analogy
analogy_result_1990 = model_1990.wv.most_similar(positive=['men', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result for the year 1971-1990: {analogy_result_1990}")
#word analogy
analogy_result_2006 = model_2006.wv.most_similar(positive=['men', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result for the year 1991-2006: {analogy_result_2006}")
```

```
Word analogy result for the year 1800-1850: [('statesman', 0.35491928458213806)]
Word analogy result for the year 1851-1900: [('comrade', 0.38791826367378235)]
Word analogy result for the year 1901-1950: [('lad', 0.49189355969429016)]
Word analogy result for the year 1951-1970: [('lad', 0.46681398153305054)]
Word analogy result for the year 1971-1990: [('soldier', 0.5077458620071411)]
Word analogy result for the year 1991-2006: [('someone', 0.41598090529441833)]
```

```
: # Find words similar to the word woman
similar_words = model.wv.most_similar('woman', topn=5)
print(f"Similar words to 'woman': {similar_words}")
```

Similar words to 'woman': [('wife', 0.6032223105430603), ('female', 0.5838871002197266), ('girl', 0.5633318424224854), ('husband', 0.552932858467102), ('seducer', 0.5423462986946106)]

Word Analogies

```
: #word analogy
analogy_result_1850 = model_1850.wv.most_similar(positive=['seducer', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result for the year 1800-1850: {analogy_result_1850}")
#word analogy
analogy_result_1900 = model_1900.wv.most_similar(positive=['seducer', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result for the year 1851-1900: {analogy_result_1900}")
#word analogy
analogy_result_1950 = model_1950.wv.most_similar(positive=['seducer', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result for the year 1901-1950: {analogy_result_1950}")
#word analogy
analogy_result_1970 = model_1970.wv.most_similar(positive=['seducer', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result for the year 1951-1970: {analogy_result_1970}")
#word analogy
analogy_result_2006 = model_2006.wv.most_similar(positive=['seducer', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result for the year 1991-2006: {analogy_result_2006}")
```

Word analogy result for the year 1800-1850: [('men', 0.3185313045978546)]
 Word analogy result for the year 1851-1900: [('shilling', 0.32045993208885193)]
 Word analogy result for the year 1901-1950: [('impunity', 0.316707968711853)]
 Word analogy result for the year 1951-1970: [('someone', 0.3898908197879791)]

KeyError: "Key 'seducer' not present in vocabulary"


```
analogy_result = model_1850.wv.most_similar(positive=['childbirth', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result: {analogy_result}")
```

Word analogy result: [('men', 0.36237356066703796)]

```
analogy_result = model_1900.wv.most_similar(positive=['childbirth', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result: {analogy_result}")
```

Word analogy result: [('administrator', 0.29424118995666504)]

```
analogy_result = model_1950.wv.most_similar(positive=['childbirth', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result: {analogy_result}")
```

Word analogy result: [('illness', 0.38594549894332886)]

```
analogy_result = model_1970.wv.most_similar(positive=['childbirth', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result: {analogy_result}")
```

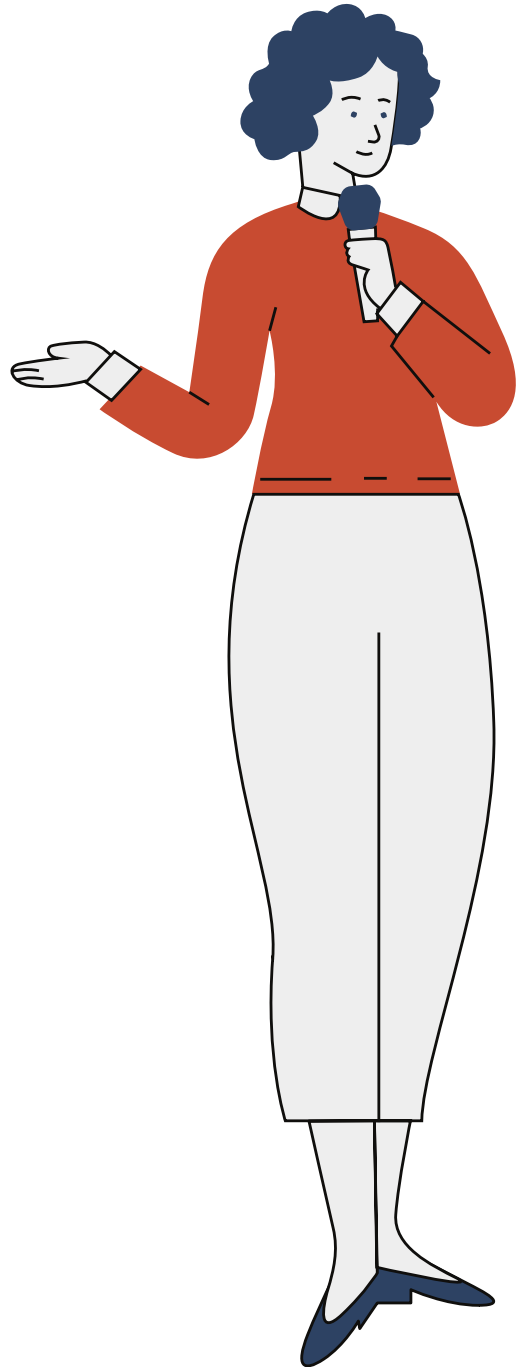
Word analogy result: [('someone', 0.3920690715312958)]

```
analogy_result = model_1990.wv.most_similar(positive=['childbirth', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result: {analogy_result}")
```

Word analogy result: [('unpublishable', 0.3387390971183777)]

```
analogy_result = model_2006.wv.most_similar(positive=['childbirth', 'man'], negative=['woman'], topn=1)
print(f"Word analogy result: {analogy_result}")
```

Word analogy result: [('daughter', 0.3618466258049011)]



- Get Access to HPC Servers
- Generate Word Embeddings using FastText
- A Machine Translation-based analysis between 1850 - 1900 and 1950 - 2000 data
- Fine-tune BERT and GPT2 on the 1850 - 1900 and 1950 - 2000 data and analyze the differences

PROJECT TIMELINE

Process	September				October				November				December			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Initial Analysis																
Word Embedding and Machine Translation																
Fine Tuned LLM's																
Results and Writeup																

The End

Thank you

