Ambadipudi, Chaitra (223009560) **Sharma, Aastikta (823007872)**

Detecting gender bias in news articles

Introduction:

In this project, we have chosen to gather articles published over a decade by the New York Times and perform experiments to identify the gender biasness of the articles. Task 1 aimed at crawling data from this website. As a part of task 2, pre-processing of the crawled data along with appropriate natural language processing followed by indexing has been carried out. This report aims at highlighting the various milestones achieved as a part of task 2.

Pre-processing of Data:

The data crawled from website contained a lot of undesired information. Hence, we wrote a script to extract useful parts of it, including, but not limited to:

- Headline
- Snippet
- Persons
- Keywords like subjects/organizations

This information will be used in name entity recognition followed by gender labeling.

Natural Language Processing:

After getting the desired fields, name entity recognition was performed using the Stanford NER server. Every document is processed to extract the names from the articles. The first names are then used for gender labeling. We use the Python-SM library that consists of a huge database of names with over 40000 names and genders.

The first names found in a document are extracted and then passed on to this dictionary of words to correctly identify the gender of the person. The output can be either “male”, “female” or “anonymous” (for the names not found in the dictionary).

After getting the gender labeling done, all desired fields are written into JSON format files which are required by Solr for indexing. The field names that make up the JSON files are “news_desk”, “lead_paragraph”, “word_count”, “_id”, “snippet”, “subsection_name”, “pub_date”, “persons”, “subject”, “organizations”, “male”, “female” and “anonymous”.

Solr Indexing:

Before indexing the files onto Solr, various configuration files were created and/or updated to accommodate the files. A new workspace has been created on Solr with a new core to store the files. The basic schema was configured to suit the data from the JSON files. Using the command line options, we were then able to push (using `bin/post`) all the data files to Solr.

We checked if the indexing has been done properly by querying a couple of tasks like querying all the documents containing name "Ken", etc.

Challenges and Solutions:

1. We faced a couple of challenges while completing this task. The first challenge was to identify a good exhaustive dictionary of names and gender that accommodates a large number of names with their genders. We finally managed to find a good dictionary called SexMachine(SM) library to suit our needs.
2. The second challenge was to merge the entities provided by default by the NYT API with the entities extracted by the Stanford NER server. We managed to do a basic consistency check by looking for pattern matches in the first and last names of a person, and then merging similar ones.
3. Next, while running the python code to extract names and then perform gender labeling, initially took almost 6 hours for 6 months of data for a year. To resolve this, we created a single instance of the dictionary and called that instance over all the years. This reduced our run time drastically from hours to a couple of minutes.
4. Another challenge, while indexing documents to Solr, was with the schema setup. Initially, there was a managed schema which Solr was using by default to index the data. Since the managed schema could not properly capture all the indexable fields during our JSON data, we were unable to query the documents successfully. Once we created our own schema and configured Solr to use it, we were able to push the data files and query them.

Future Tasks:

Since we are done with indexing, we will be querying the appropriate data based on male-female count, organizations biased towards male or female, etc. After getting the results from the query, we will process the results to be displayed on screen with a good visualization environment. We plan to display a couple of information like trend over a period of 10 years, most popular names in a particular category, etc.