Index

Introduction

Textual similarity is one of the essential techniques of Natural Language Processing (NLP) which is being used to find the closeness between two chunks of text by its meaning or by surface. In general, every statement has some context, some of which could also have some sort of preference towards some group over others. Gender bias is one such example. The forms of gender bias could be:

1. Firstness: where a gender (male or female) is always mentioned first.
2. Stereotype: where we relate traits with a particular gender
3. Subordination: where the text reflects a gender is subordinate compared to others.

Task

We are given a set of pairs of text and the task is to determine if they are similar or dissimilar based on some form of gender bias present in the text as explained above.

Data Description

The criteria for biases in this dataset was GENDER. We are provided with 3 text files:

- text-and-id.txt : consists of numerous unique IDs followed by sentences.
- pairs-label-training.txt : consists of rows of 2 IDs and an indicator of similarity (0 or 1).
- test-data.txt : consists of rows of 2 IDs whose similarity we are supposed to determine.

Indicator of similarity:

- 0 indicates both statements are biased, or both are unbiased.
- 1 indicates one statement is biased and the other is unbiased.

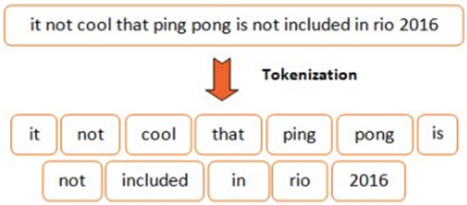
Data Preprocessing

Steps:

First we convert the given text files into csv files. Now we have text with ids in one CSV file and ids and biases in another CSV file. Firstly we substitute ids with corresponding text.

A. Tokenization:

We now tokenize the text data. Tokenizing is the process of dividing text material into tiny bits. These tokens aid in better comprehending the data and developing the model accordingly.

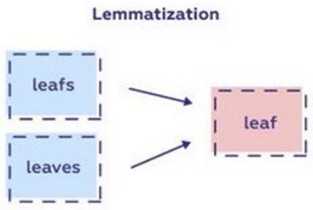


B. Eliminating Whitespaces:

In the preceding tokenizing procedure, spaces also become a token, which unnecessarily expands our data. So, we eliminate tokens that contain whitespaces.We now transform all tokens to lowercase because vectorizing the same word would provide distinct vectors if some letters are capitalized.

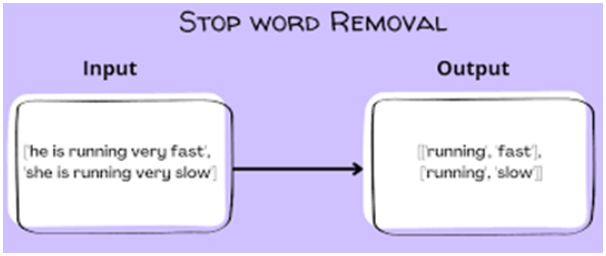
C. Lemmatization:

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.



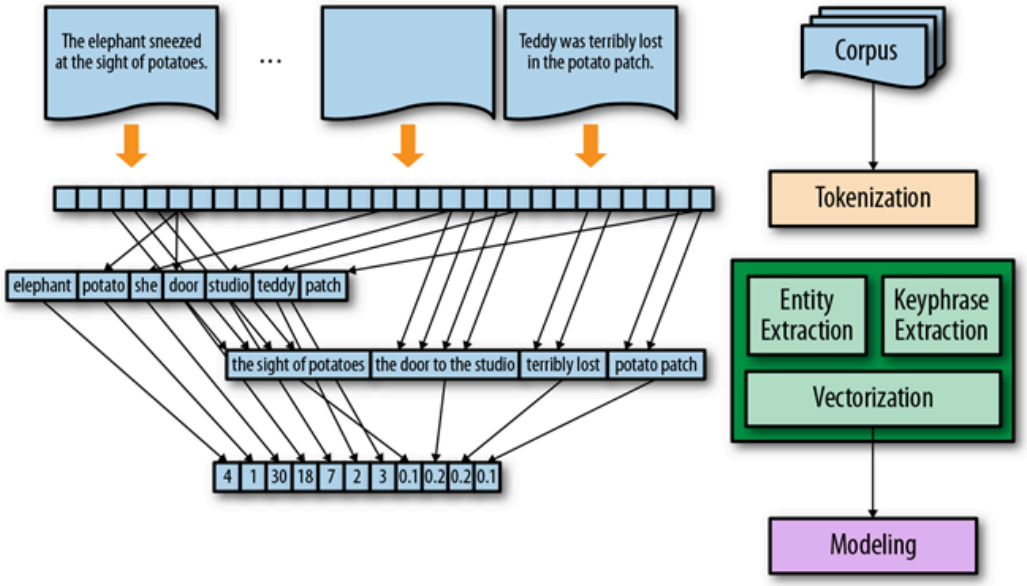
D. Removing Stopwords:

Then we use stop words to delete common phrases such as the, a, an,but, yes, no, and many more. Data becomes more precise and to the point as a result. As per our need, we also omit some words that are usually removed but are required in the current scenario. These terms are he, she,her, him, himself, and herself.



E. Vectorisation:

We now use the TensorFlow's built in tokenizer module to convert tokens to indices thus converting an array of words to a vector.

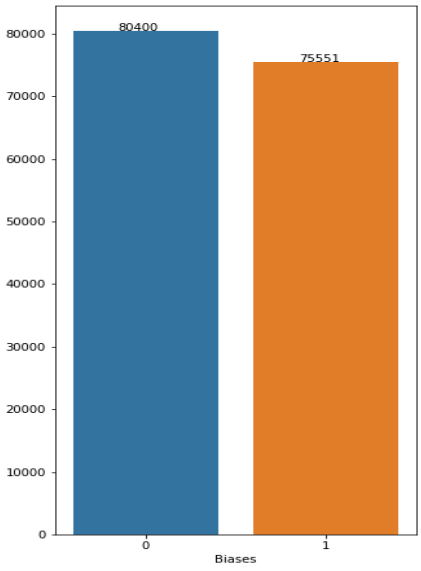


Data Visualization

Plot of biases:

We plot the labels in training data to get the information of biases.

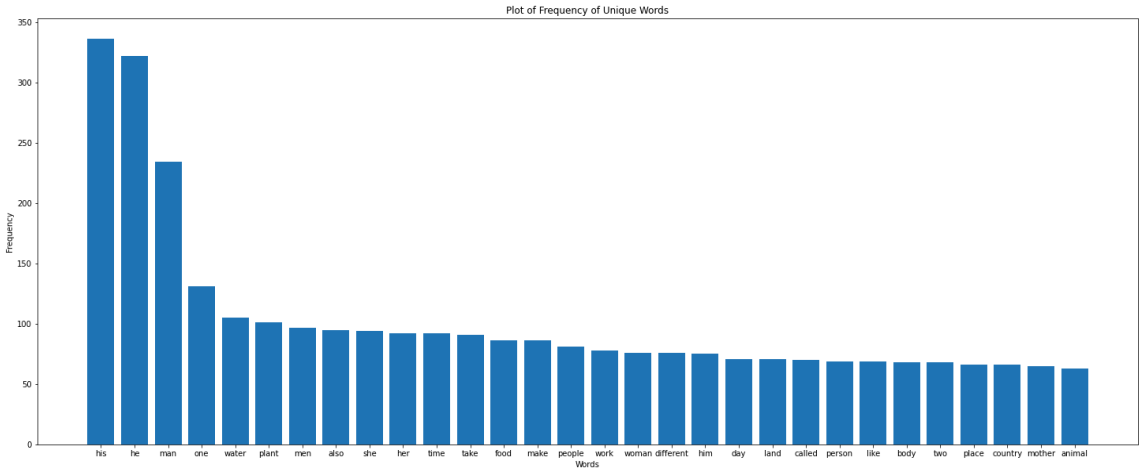
- 0 indicates both statements are biased, or both are unbiased.
- 1 indicates one statement is biased and the other is unbiased.



By observing the plot we can say that there are 80400 pairs in the training data which are label zero and 75551 pairs which are label one.

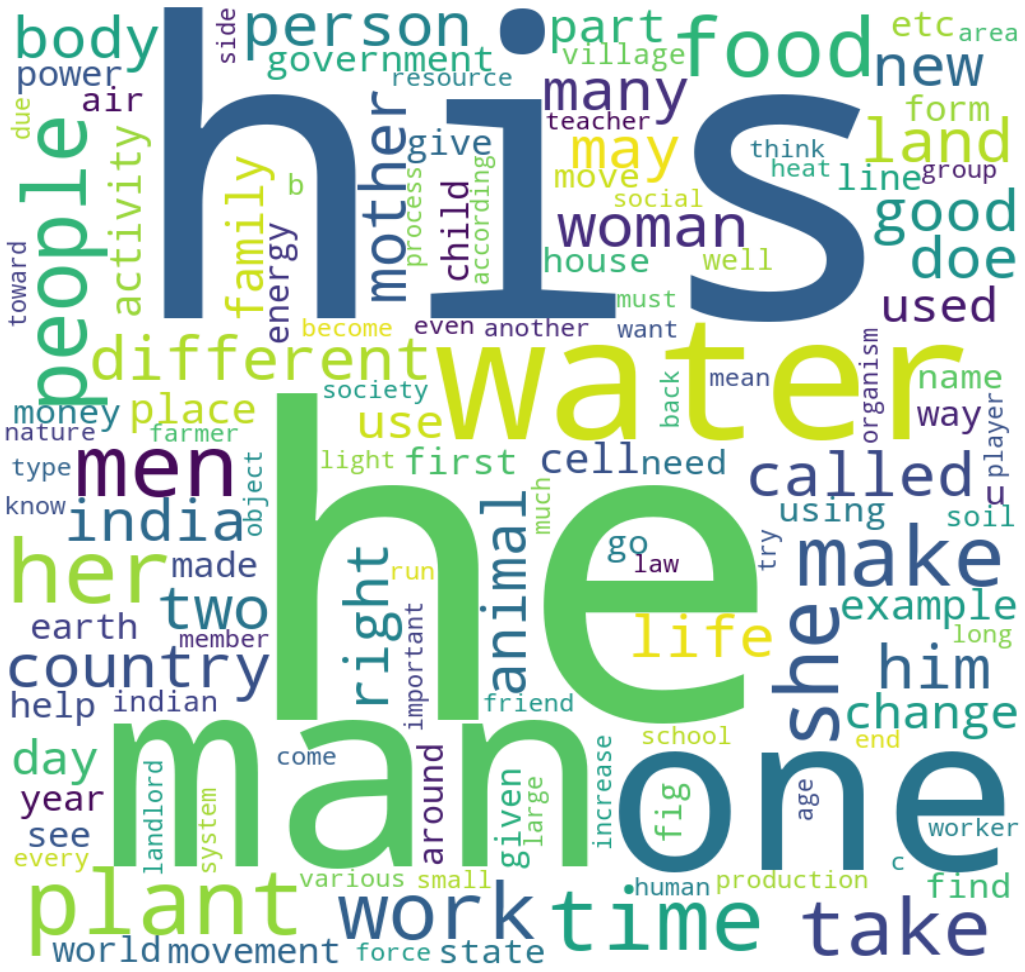
Plot of frequency of words:

We plot the frequencies of some unique words which have higher frequencies in the text. We observe that words ‘his’, ‘he’, ‘man’, ‘she’, ‘her’ are more frequent in the text.



Word Cloud:

We visualized the data in the form of a word cloud. The bigger and bolder the word appears,the more often it is mentioned and the more important it is.

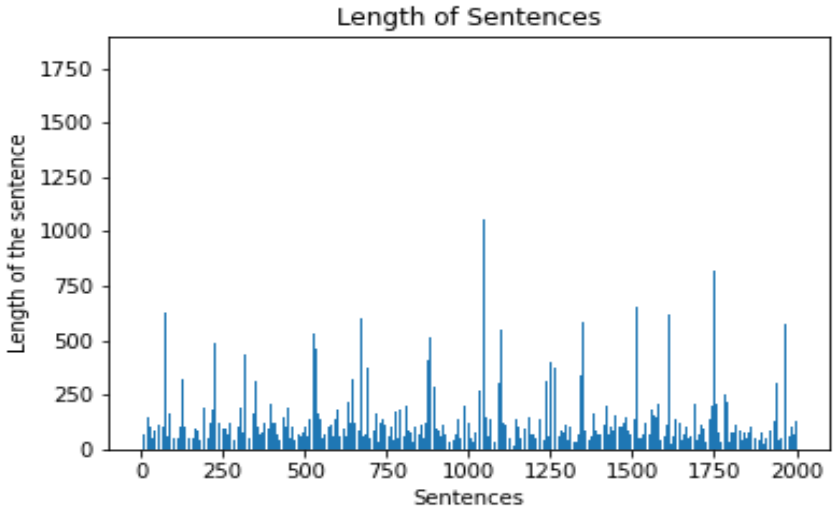


We can observe from the above two plots that the frequency of masculine words like ‘his’, ‘he’, ‘man’, ‘men’ are significantly higher than their feminine counterparts, this leads to more biased sentences which is evident from the first graph (Plot of biases).

Plot of length of sentences:

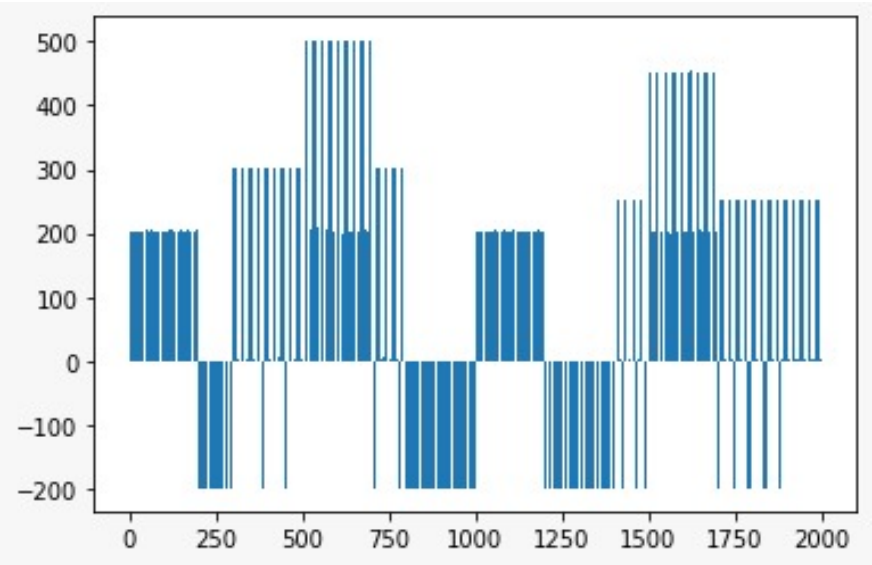
We plot the sentences with their corresponding lengths. Length of the sentence includes text, numbers, special characters, blank spaces.

From this plot, we can see that the majority of sentences have lengths ranging from 150 to 250, and there are some spikes in the plots that show that there are some longer sentences, with lengths ranging from 400 to 800 or even greater than 1000.



Plot of frequency of statement IDs in training data:

Given below is the plot of frequency of ids present in pairs-label-training.txt
Where IDs which were not used in the training data are shown by -200 frequency for better visualization.

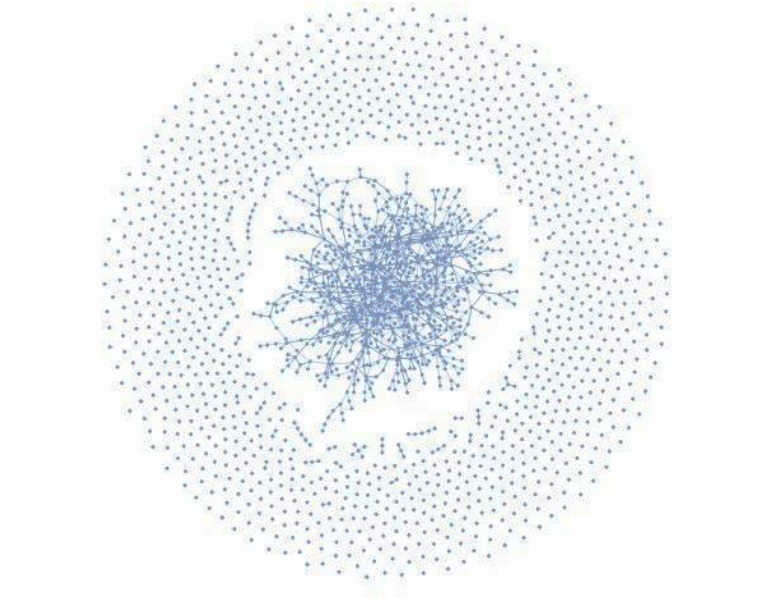


We can observe from the plot that in the training dataset mostly ids are repeated 200,300,400,500 times but there are some ids which haven't been used at all like ids belonging to 800 to 1000.

Exploratory Data Analysis

By observing and visualizing the given data, we can see that some of the statements were not used at all in the training dataset pairs. This means while splitting our training dataset into training and validation sets we cannot randomly select some pairs for validation and remaining pairs for training as this will lead to inaccurate results because the statements used in the validation set would have already been used in the training. Therefore we have to split the data in such a way that the statements used in the validation set won't be present in the training set.

An undirected graph is created for the whole training set by treating a single training example id1, id2, and 0/1 as an edge between "id1" and "id2" nodes with weight 0/1. A single connected component with 1152 nodes was discovered in the created graph using the disjoint set union approach.



There are some isolated nodes on the boundary in the graphs above, and all remaining nodes form a single connected component.

Approach and Models

After removing some training examples(edges in the graph), the training data is divided into two unconnected components for validation purposes because the provided train data lacks any ids that appear in the test data. Further training and validation is done on the valid split obtained using the above method.

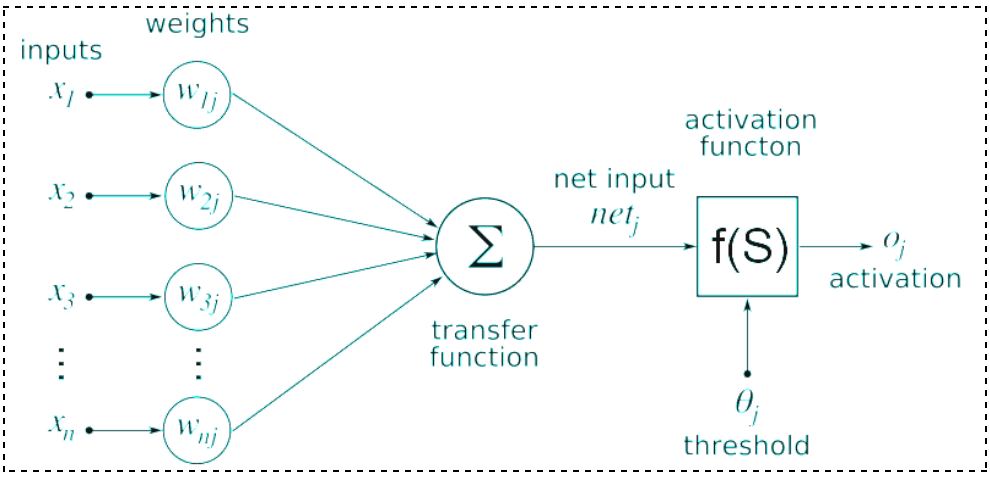
Artificial Neural Networks (ANN)

Artificial neural network (ANN) is a deep learning method that arose from the concept of the human brain biological networks. ANNs are built in a way to mimic the network of neurons in a human brain by a computer. Even though they work in an extremely similar manner, they are not identical.

Architecture of Artificial Neural Network (ANN)

The basic architecture of ANN consists of 3 layers:

1. **Input layer :** which accepts numeric and structured data.
2. **Hidden layer(s) :** which extracts some of the most relevant patterns from the inputs and sends them on to the next layer for further analysis.
3. **Output layer :** which gives the final result that we are looking for.



Each layer is a set of various nodes that act as neurons with their own weights and biases, i.e, the factor by which that particular node influences the prediction. Think of the weights as the interconnection strength between nodes and the bias as the minimum threshold to send signals. Each node also has an activation function that is a transfer function which is used to get the desired output for the problem designed.

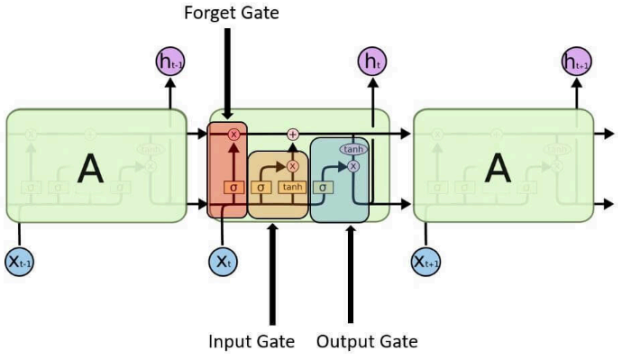
For our problem the preprocessed vectorized dataset was given as an input to the ANN and the binary labels were provided as the output. We observed an accuracy of **64%** on the validation dataset.

Recurrent Neural Network (RNN)

RNNs are neural networks consisting of multiple hidden layers wherein output of each hidden layer is fed as input into the next hidden layer. They are most useful when memory of previous processing is required to make new predictions such as that in case of Natural Language Processing. RNN ensure that memory of each processing is transferred onto the next level.

Long Short Term Memory

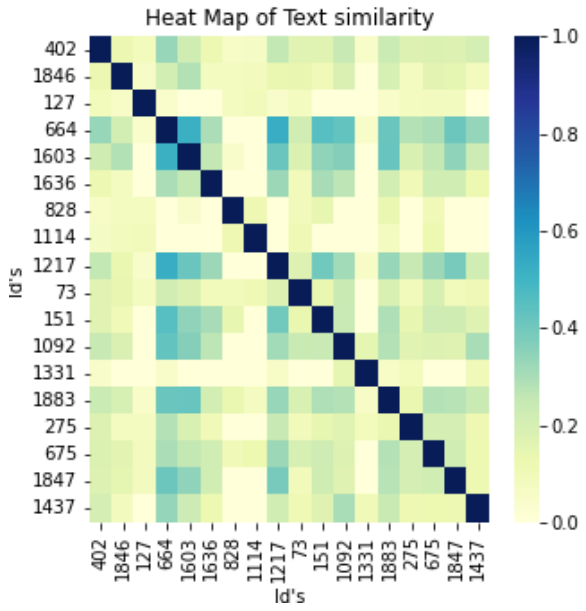
Here we have used an RNN model called Long Short Term Memory networks – usually just called “LSTMs” . They’re a special type of RNN model capable of learning long term dependencies in the dataset. Unlike traditional RNNs,each layer of an LSTM receives input from multiple previous layers assigning them weights such that recent data has more weight in the predictions made.



After implementing LSTM on our vectorized input dataset, we achieved an accuracy of **79%** on the validation dataset. Which is significantly better than that of ANN as LSTM can better understand the sequence of words in text

Text Similarity

Cosine similarity is a model that measures the similarity between two vectors in an n dimensional space. It does so by measuring the cosine of angle between the vectors thus determining whether the vectors are pointing in roughly the same direction.

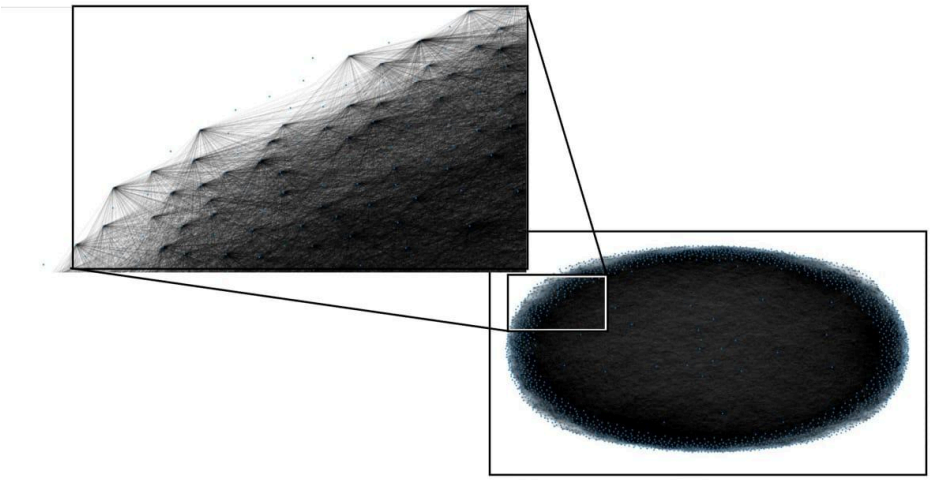


We applied the cosine similarity model on our vectorised dataset and observed the correlation between how related two sentences are and whether they had the same biases or not. We observed that beyond a certain correlation coefficient between two sentences, most of them had the same biases, and below that threshold value, they had different biases. Thus by applying a threshold on percentage similarity between two sentences, we were able to create a model that predicts the output of the test dataset. This model gave us an accuracy of **67%** after determining the best threshold value.

The Bias Model

We made a graph to represent all pairs in the training dataset as mentioned in the exploratory data analysis section. Then we manually choose a clearly biased statement which represents a node in the graph and label it as biased(1). Any neighbor to the chosen node that has an edge weight of 0 will be biased(1), while an edge weight of 1 will be unbiased(0). This idea to

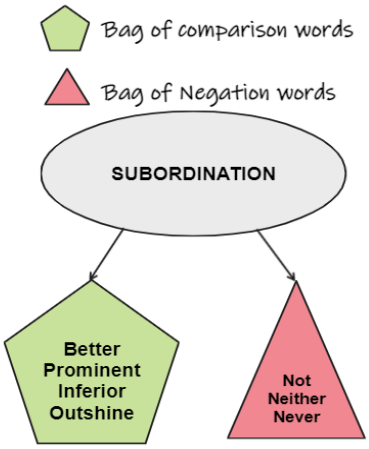
determine biases or unbiasedness of neighboring nodes is extended for all the remaining connected nodes in the graph using a breadth first search algorithm on the selected node. Nodes with odd path length will be unbiased(0) and with even path length will be biased(0). This implementation gave us a dataset of 1152 datapoints. Where each datapoint represents a statement used in the training and is labeled on the basis if it is gender biased or not. After training a machine learning model Support Vector Machine on the above data to determine if a statement is biased or not gave an accuracy of **60%**. Which is understandable as the dataset had only around 900 points to train on. The image below shows a few nodes of the graph created and shows that the degree of each node is very high.



Subordination Model

The Subordination model classifies sentences as biased or unbiased based on the presence of gender related words and comparison related words. The model preprocesses the sentences using the steps of Natural Language processing. After running them through a series of if-else statements, the sentences are classified as biased and unbiased.

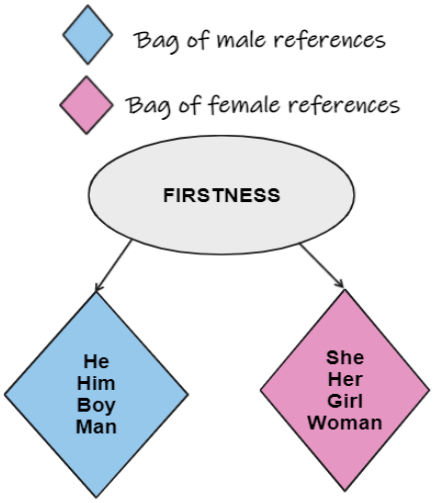
4 separate lists were made out of the given data, namely male list, female list, comparison list and a negation list. The male and female lists consist of a set of words to address the respective gender in different forms. The comparison list consists of a list of words that compares two entities. The negation list contains a set of words that negate a sentence. For every sentence that contains words from both male and female lists, we check that for comparison words. If it exists, then we also check for negation words. If the negation word exists, it is marked as unbiased, else it is marked as biased. With this model we get an accuracy of **84.7%**.



Firstness Model

This is a model that predicts if the statement is biased or unbiased on the basis of earlier occurrence of gender depicting words. Similar to subordination, this model also uses a series of if-else statements and a dataset of words to predict the bias of the statement. We observe that in most texts consisting of both genders, words depicting male gender are used prior to words depicting female gender.

We create 2 sets of words, one depicting male and another depicting female gender. Using these sets, if a sentence contains male words before female words, or words depicting only a single gender, we classify the statement as a biased one. Rest of the statements are considered to be unbiased. Using this model, we obtained an accuracy of **84.1%**.



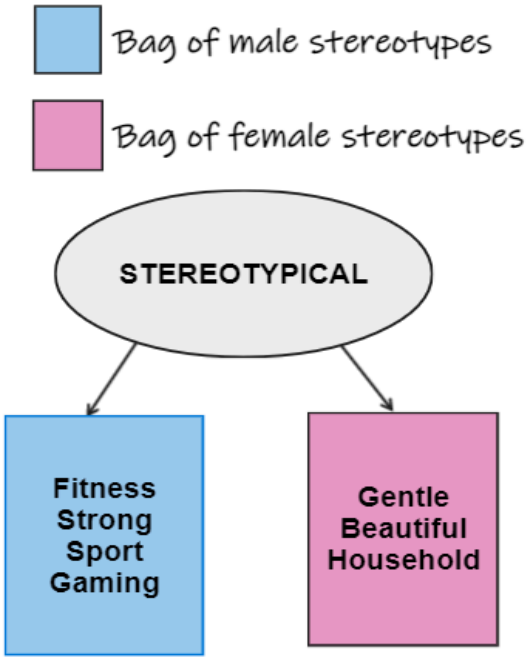
Stereotype Model

In this model, we deal with biases based on certain fixed general characteristics associated with words that we as humans learn throughout our lifetime. Consider the following two lines:

1. The man is going to the gym.
2. He is cooking food in the kitchen.

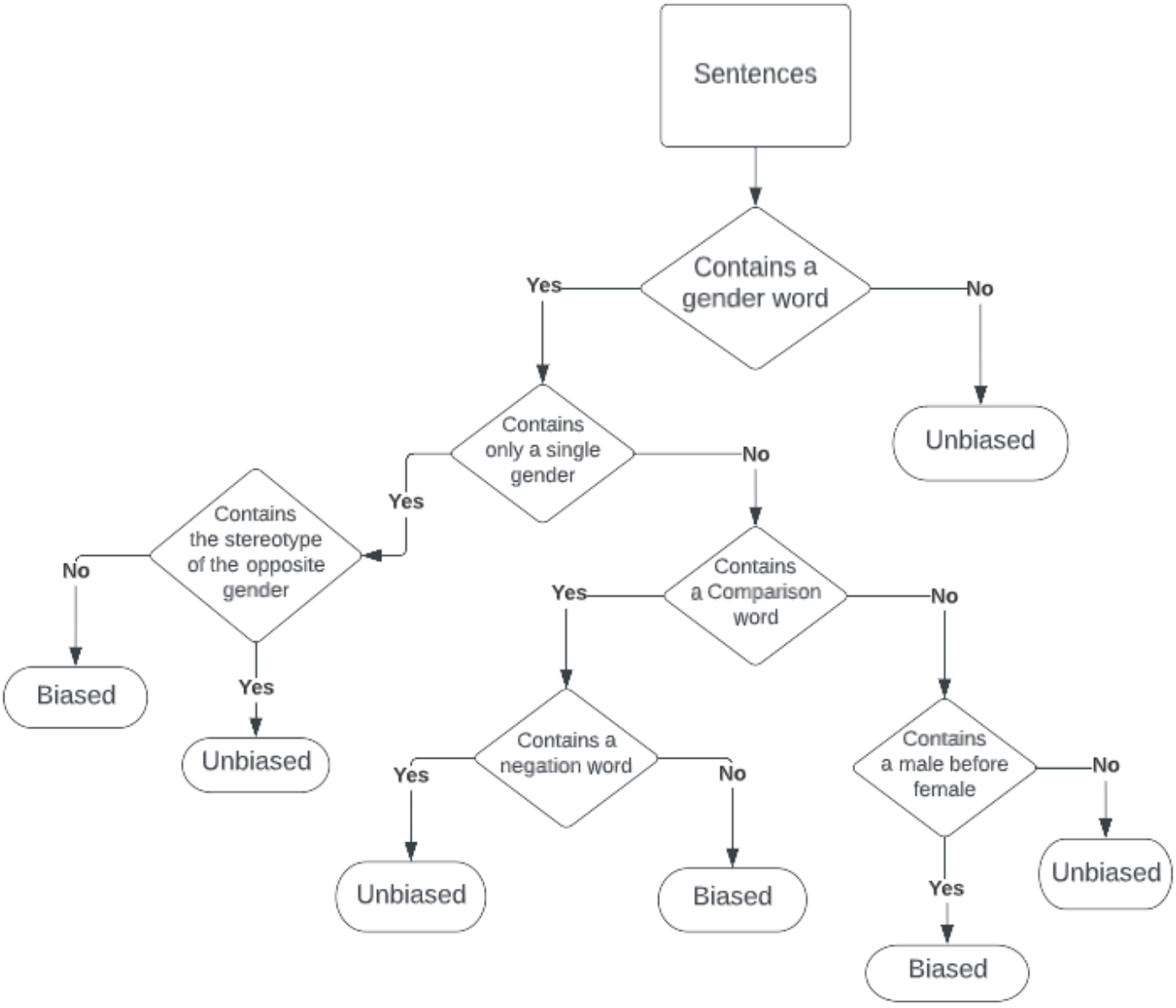
On reading the first line, we are talking about 'man' which depicts masculinity and we often see gym associated with it, thus, it is a biased statement. Whereas in the second line, we are associating kitchen with 'he', thus, disregarding that 'kitchen' is a female oriented stereotype. Hence, it's unbiased.

For each statement, we tried to get the context of the sentence by observing if the words in the sentence are masculine or feminine and we also observed if the subject is male or female. If both the context and the subject are of the same gender, it shows stereotype and is labeled as biased. Otherwise the statement is anti-stereotypical and is labeled as unbiased. Using this model, we obtained an accuracy of **84.5%**.



Combined Model

Learning from the results of the last 3 models (subordination, firstness and stereotype), we decided to create a hybrid model that combines subordination, stereotype and firstness in coordination to predict if a statement is biased or unbiased. This model predicts the statement to be biased if either one of the 3 models predict the statement to be biased. Using this model, we were able to predict the biases of statements with **88%** accuracy.



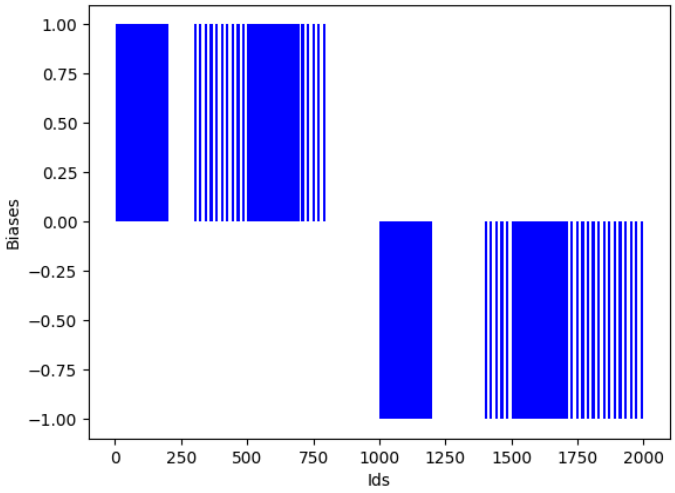
Final Approach

For deploying a model on real and practical dataset we can use our combined model given above, this model gave an accuracy of **88%** on the given dataset.

But while making the graphs and determining biases or unbiasedness of a statement we made a very interesting observation.

The biases or unbiasedness of each statement is determined using the biased model as mentioned before. Each statement has a unique ID ranging from 1 to 2000. When the Biases/Unbiasedness vs ID graph is plotted, the following interesting trend emerges:

Every biased statement returned by the bias model has an ID between 1 and 1000, and every unbiased statement has an ID between 1001 and 2000. The reason behind this could be human error during labeling and indexing of the statements data. When we used this observation as an approach to determine the biases or unbiasedness of a statement and subsequently if a pair of statements are similar or not, we observed an astonishing accuracy of **100%** on the validation set.



Therefore, as a final approach for this competition and this given dataset, we will determine if a statement is biased or not based on its ID. Once the biasness is found, a pair of statements will return 0 if both are biased or both are unbiased. Else, it will return 1 if one of the statements is biased and the other one is unbiased.

References

- Estimating Gender Bias in Sentence Embeddings
- Understanding text classification in nlp
- Recurrent Neural Network in Python
- Understanding LSTM Networks
- Natural Language processing
- Deep Learning with Python: Neural Networks
- Recurrent Neural Network(RNN): Tutorial
- Support Vector Machines