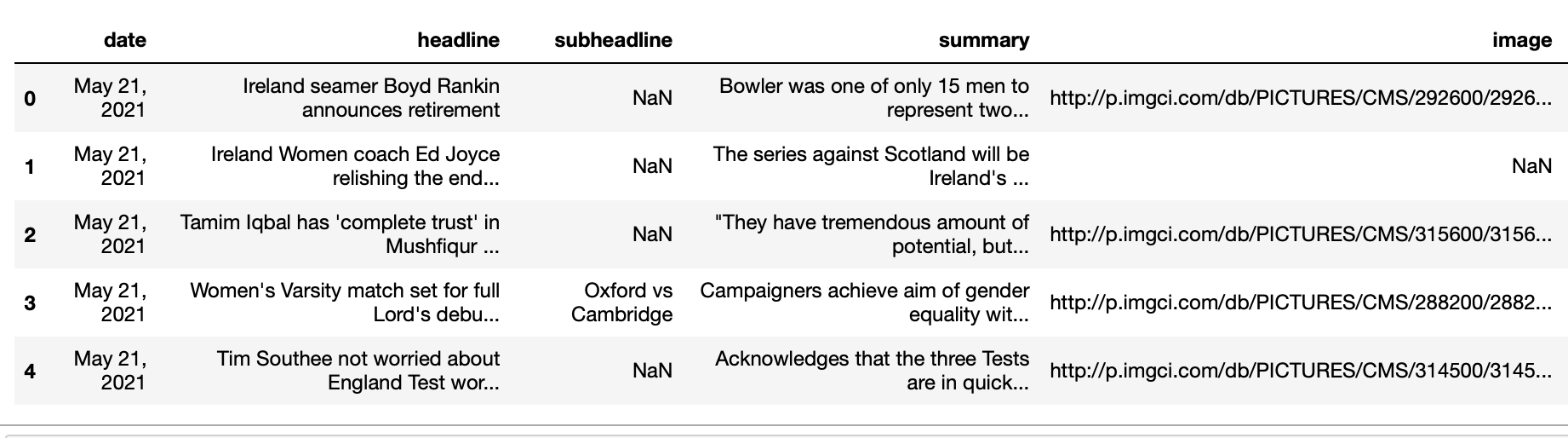
**Team Members: 2022H1030118P Tanmay Singh Aswal**

**2022H1030099P Mayank Gupta**

**Q2.a)Co-occurrence Matrix**

Dataset Used-> cricinfoNews.csv

Data Head

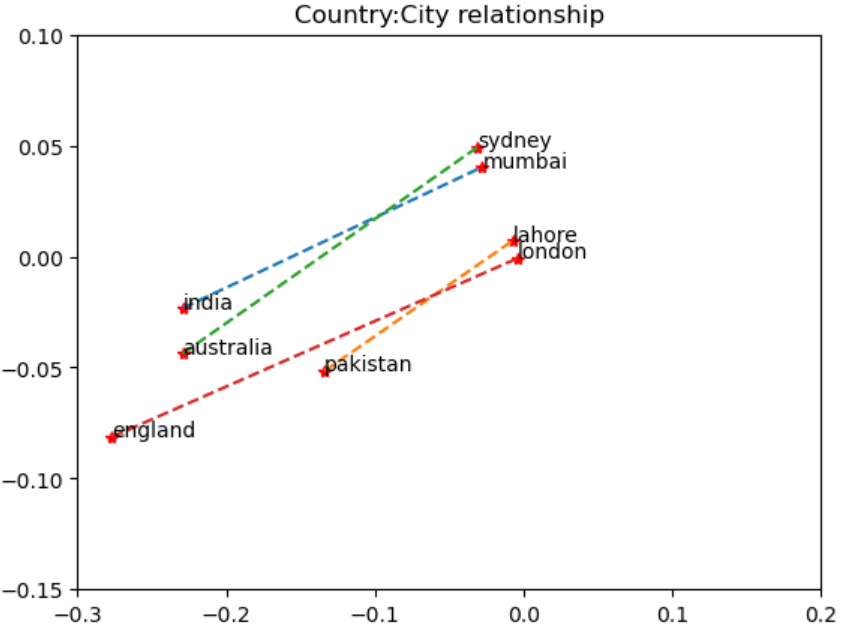
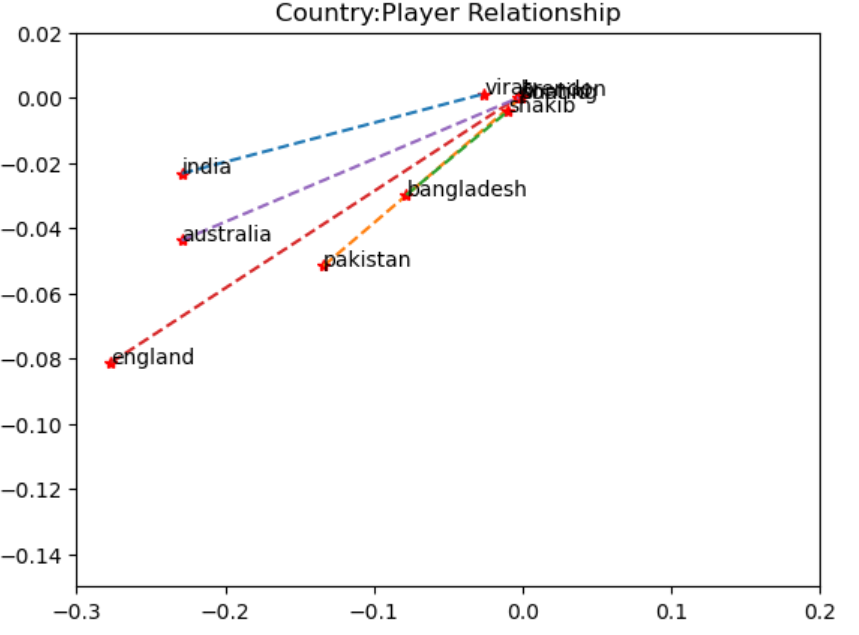
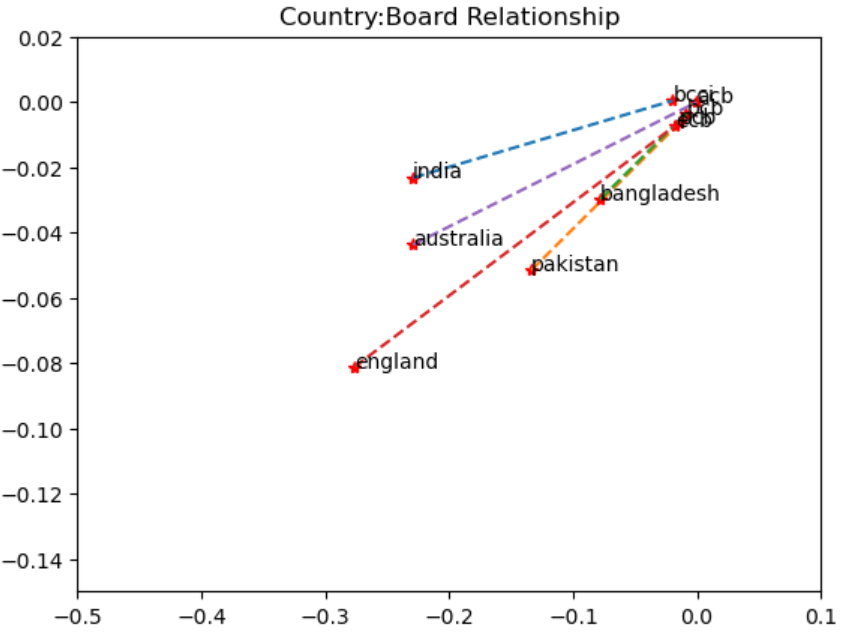
****

We are combining the headline and summary data, comprising our entire corpus.

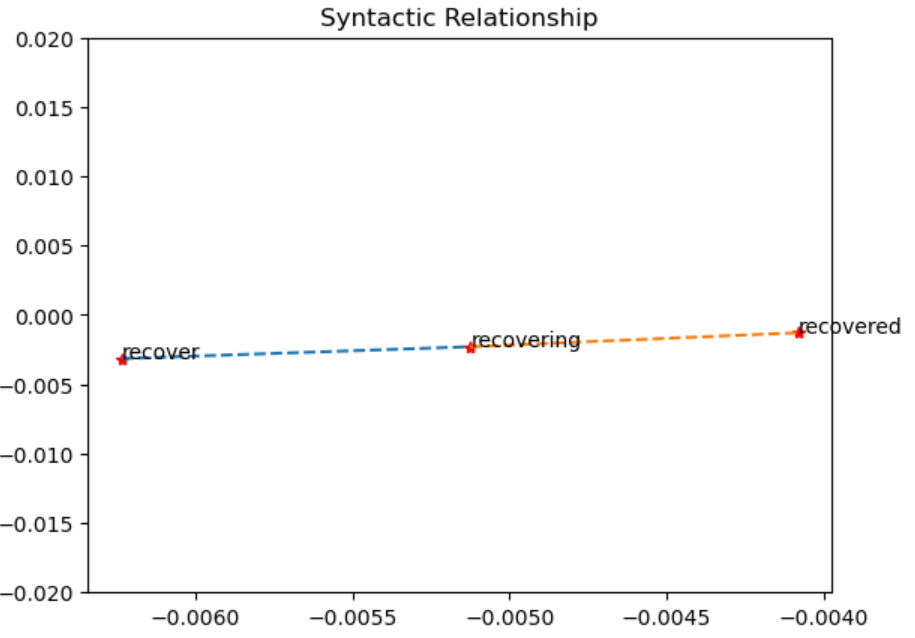
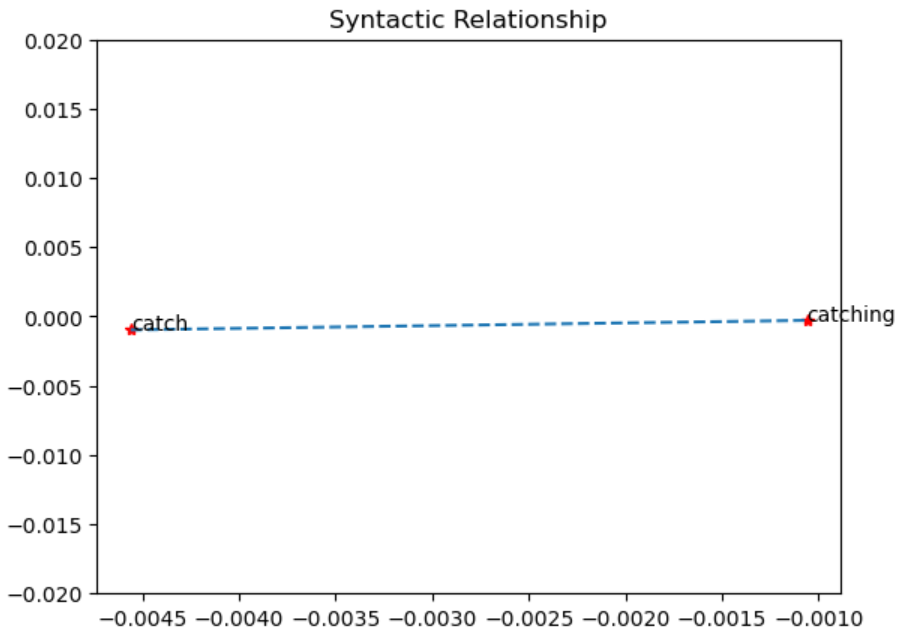
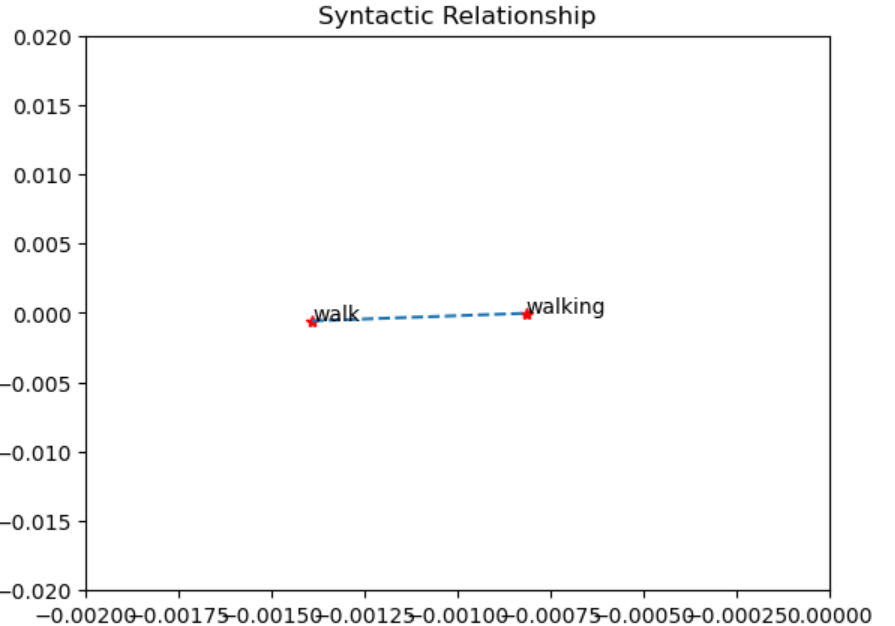
Cleaning the entire corpus by removing the stopwords, numbers, and special characters and converting them to lowercase.

We then created a co-occurrence matrix finding all the unique words from the corpus. The rows and columns in the matrix are the numbers of unique words. This resulted in a very big sparse matrix of 16760\*16760 size. if a word was in the context of the target word then increase the count of the cell at row of target word and column of context word. We then applied the SVD for dimensionality reduction to reduce the dimension to only 2 so that we can plot the graphs for semantic and syntactic relationships.

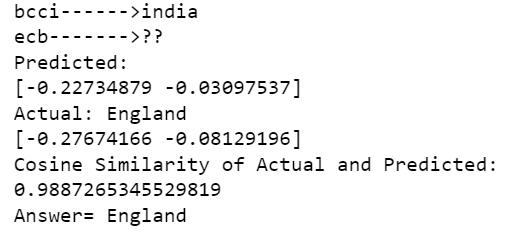
Semantic Relationship:

****

**Syntactic Relationship:**

****

**Analogy:**

****

**Q.2 b) Word2Vec Matrix**

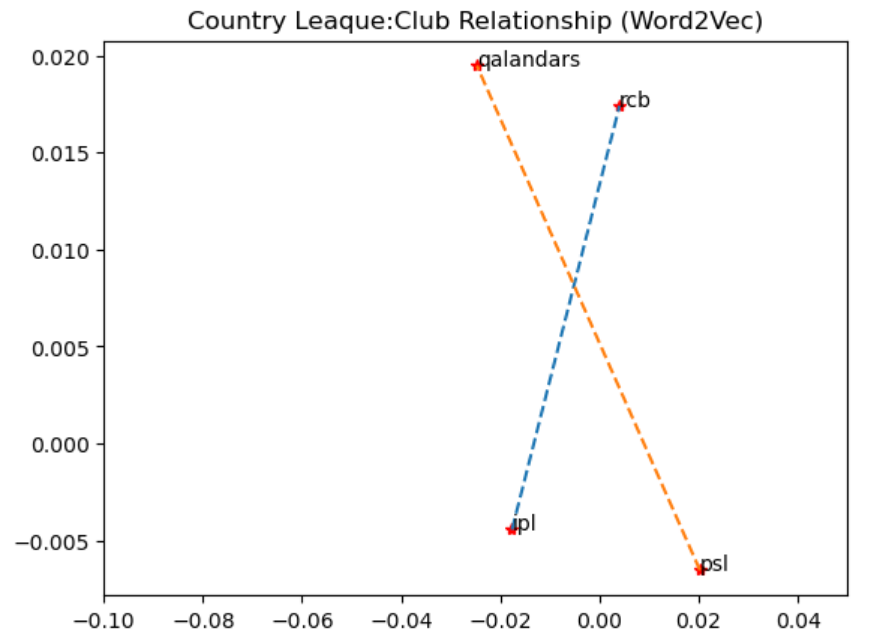
In Word2Vec, we take the same dataset as the above Dataset Used-> cricinfoNews.csv.

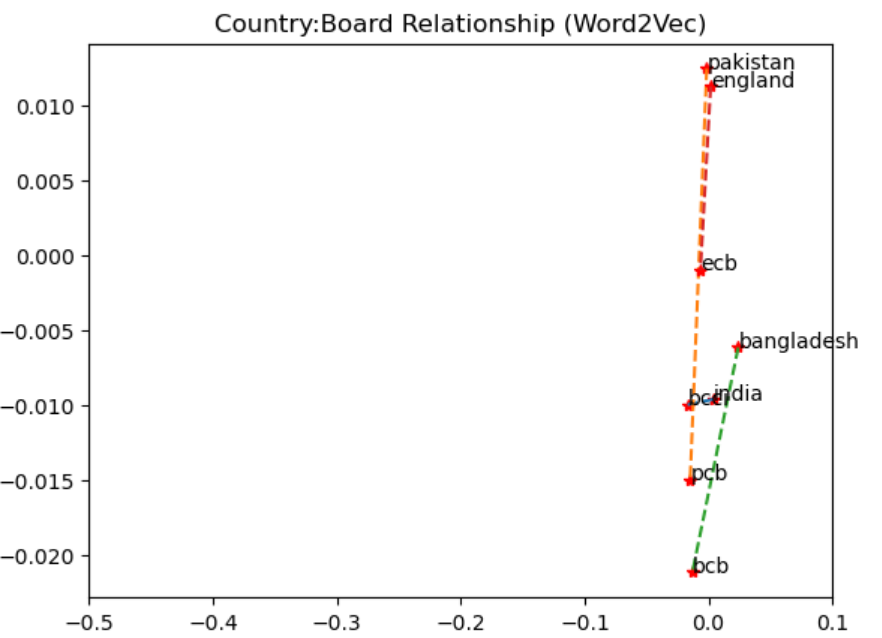
We then clean the data by removing stopwords, special characters, numbers, etc.

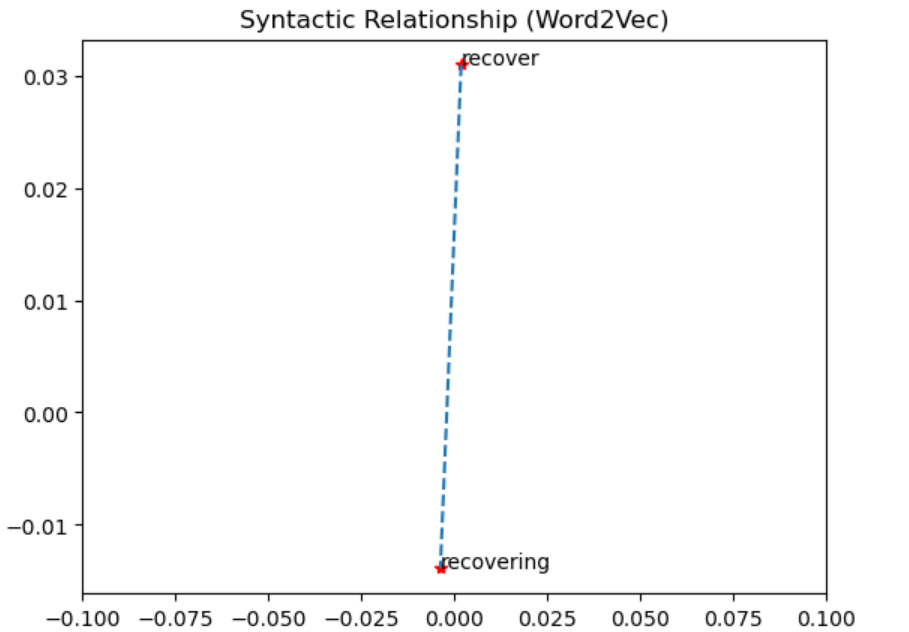
Word2Vec is a neural network structure to generate word embedding by training the model on supervised classification problems. Similar to most neural network models, the steps to train the word2vec model are **initializing weights (parameters that we want to train), propagating forward, calculating the cost, propagating backward, and updating the weights.** We take input as a 1-hot vector of the target word for forward propagation, and when backtracking, the 1-hot vector of the context words helps calculate the loss.

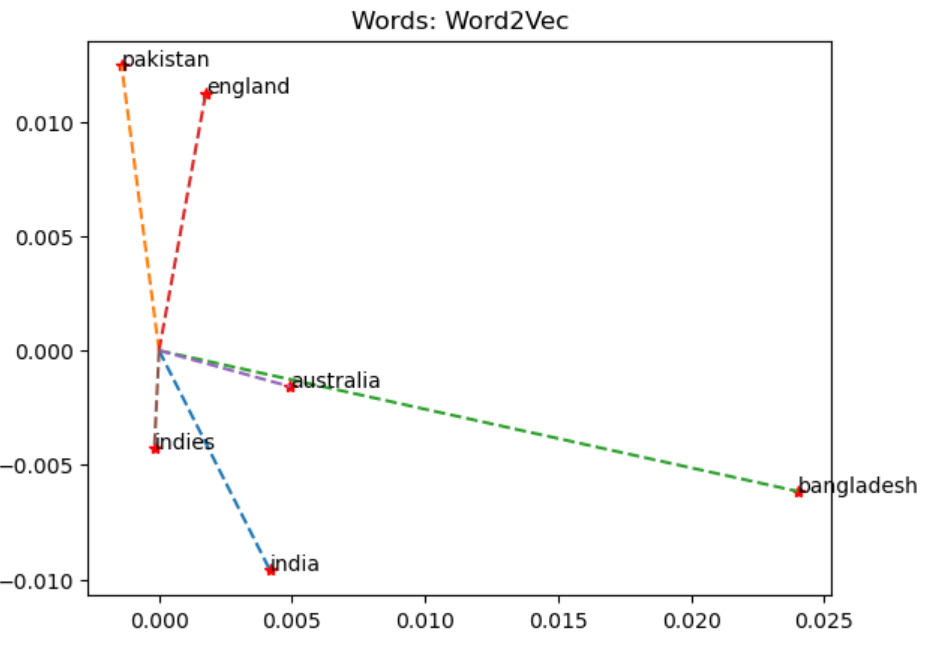
The model uses a linear activation function in the hidden layer and a softmax activation function at the output layer. The cost function we use is the negative log-likelihood function for the word2Vec model.

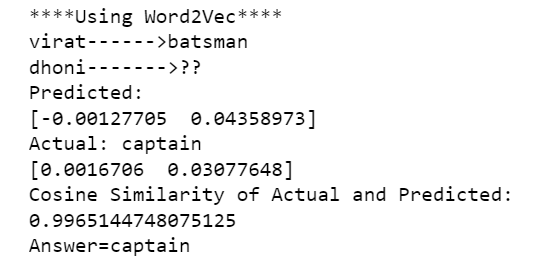
After the model was trained, we applied SVD to reduce the embedding dimension and brought it down to 2 so that we could plot it on the graph.











NOTE: Since the model of word2vec uses a neural network even with very less vocab (<4000), the amount of the weight that needs to be updated is 12 lakhs , which is very high to train in a normal system. But we need more vocab for the model to be good and give accurate results but since increasing the vocab is not a possible solution we only trained our model on (<4000) vocabulary. This resulted in a less accurate result.

4) Innovation: To make the Word2Vec model better, we can use negative sampling as a method that was given by the original author of word2vec, which can solve the problem that we have for not being able to train the model with more data as it will result in the high number of parameters to train. Using a technique called “Negative Sampling,” which causes each training sample to update only a small percentage of the model’s weights, helps take in more vocabulary as now we don’t have to update all the parameters.