

# C:/> ./detecting\_political\_bias\_in\_text.bat

~ Richard Jones

## > mkdir ./Introduction

- > In recent years, the atmosphere of UK politics has become a hotbed of unrest. Many have begun to challenge news reports and statistics as misleading or simply untrue. From this, more and more people are starting to realise they are not as good at detecting political bias as they once believed.
- > This project aims to develop a system that uses deep learning techniques to detect political bias in text, and provide an interface for a member of the general public to analyse their own text for this sentiment.
- > The aim of this project is to provide a platform where users can receive a generalised, unbiased opinion of the skew of text. By using a trained Artificial Intelligence, it is believed that this risk of a biased opinion can be reduced.

## > Objectives

- > Develop a deep learning system, trained on speeches from the UK parliament, that is capable of identifying whether a piece of variable-length text is more left-leaning or right-leaning ideologically.
- > Provide an Interface for a user to interact with the system, allowing them to upload and analyse their own texts.
- > Design and Build a Database to store the relative inputs and outputs of the system for the user to query separately.

## > Project Methodology

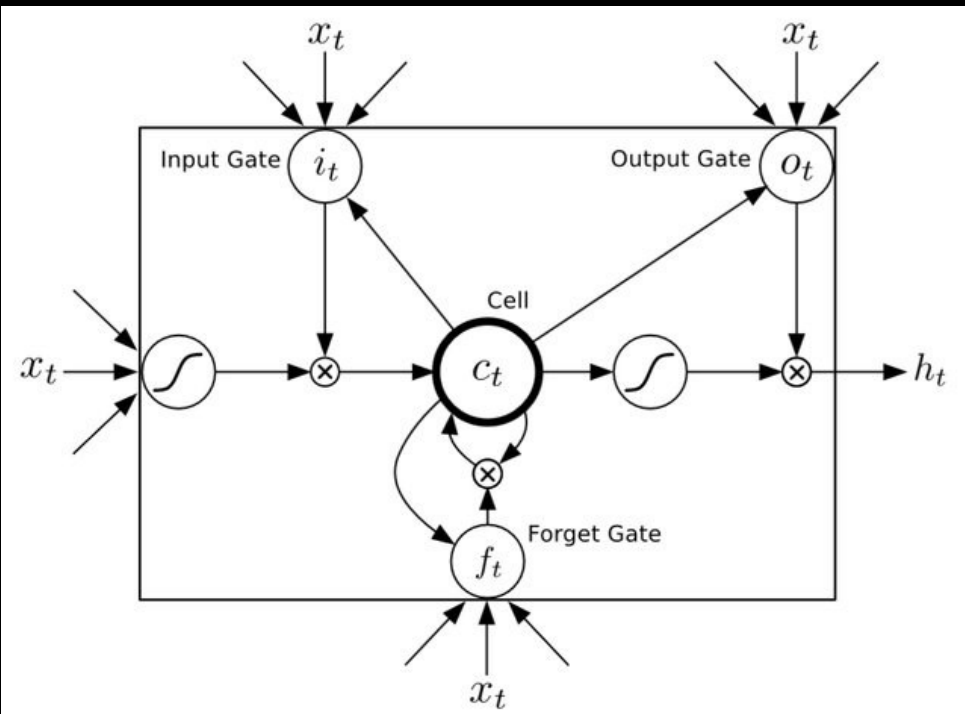
- > This project will incorporate an AGILE methodology. All of the objectives will be completed through iterative development .
- > A backlog of the tasks that need to be completed for the projects success has been created, with necessary deadlines set for each of them.
- > *The Gantt Chart below outlines the content and deadline of this product backlog.*

Section	Week																	
	Semester 1									Christmas			Semester 2					
Project Plan	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6
Literature Review																		
Analysis and Design																		
Neural Network Design																		
NN Training/Validating																		
NN Testing																		
Database																		
UI Development																		
Final Testing																		
Evaluation																		
Project Report																		
Documentation																		

## > vim Project\_Design.txt

## > python ./NeuralNetwork.py

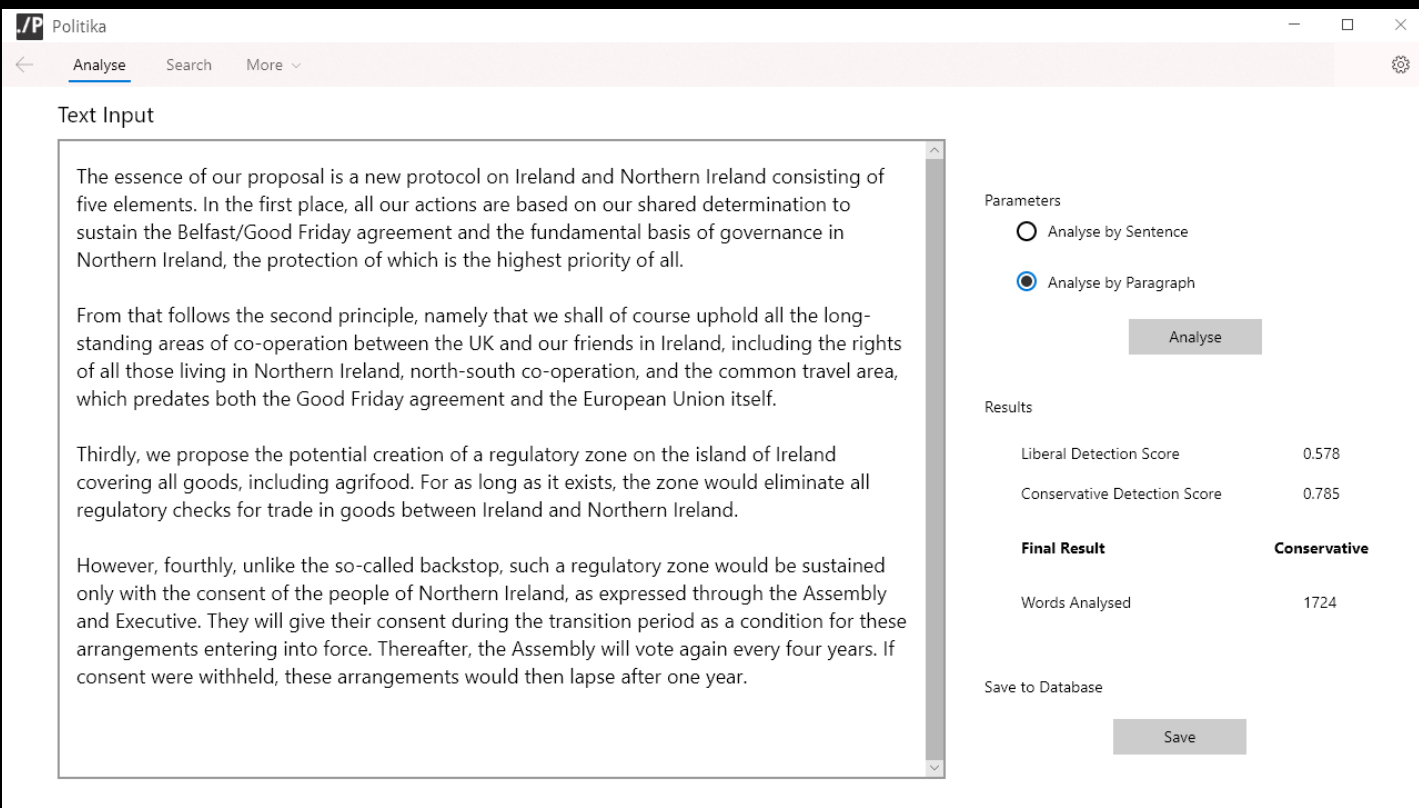
- > As this project will be focussed around analysing the political sentiment of text, using a standard Multi-Layer Perceptron model for the Neural Network will not be viable. Instead, an **LSTM (Long Short-Term Memory)** Model will be utilised (*pictured below*).
- > Every LSTM cell contains an input gate, an output gate and a forget gate. Which allows relevant or important information and characteristics of the text to be remembered, and irrelevant content to be forgotten. This has been proven to be effective in analysing sentiment across several scenarios



The Neural Network has been developed in python using Google's TensorFlow Library.

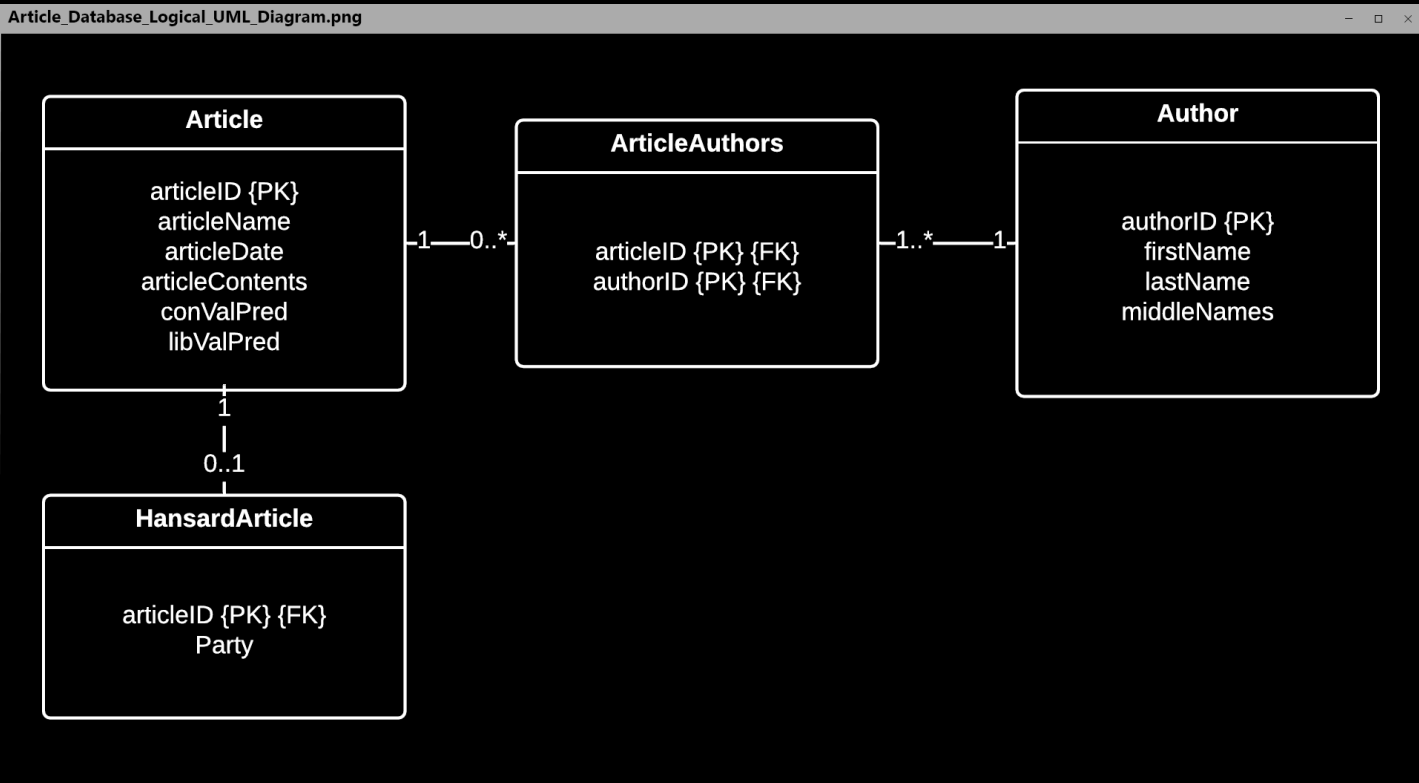
## > javac ./UserInterface.java

- > The UI will be programmed in Java using the Swing GUI Library.
- > It will provide a interface for the user to analyse their own texts, as well as query previously analysed texts through a search function.
- > It will also be responsible for passing information between the relevant python scrips that will analyse all inputted text.
- > A wireframe (*pictured to the right*) was created visualise the expected design of the UI.



## SQL> CREATE DATABASE articleDB

- > The Article Database will be developed using the SQLite Database Management system following the schema of this Logical UML Diagram (*pictured to the right*).

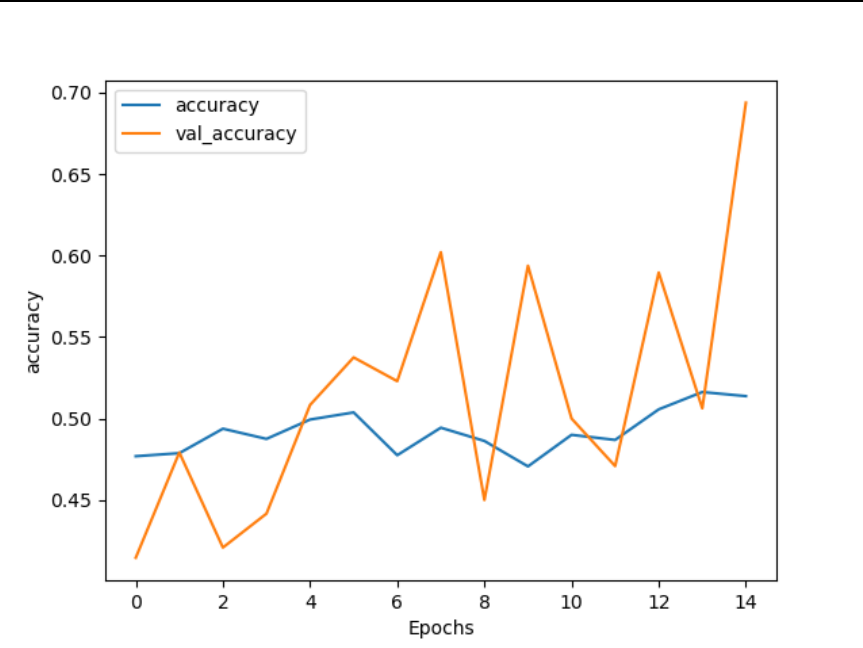


## > head Current\_Progress

- > Retrieved and Processed Training, Validation and Testing Datasets from the Hansard Parliamentary Debates Archives.
- > Created a TensorFlow Input Pipeline that: Retrieves the collected debate data, loads it into a TensorFlow dataset, and builds a text-encoder vocabulary file for use in the training of the Neural Network.
- > Developed and have begun training/optimising the Neural Network.
- > Coded the analysis section of the User-Interface and developed a method to communicate with the Neural Network.
- > Created and populated the Article Database based on the designed UML Schema.

## > echo Neural\_Network\_Performance

- > Currently, attempts at training the Neural Network have not been as optimal as anticipated.
- > As the diagram to the right shows, though there is a slow increase in training accuracy, the validation accuracy is constantly shifting or “exploding”.
- > The cause of this is possible due to too large of an initial LSTM layer size, which could’ve immediately over-trained the Neural Network, leading to the sub-prime results here.
- > Currently, work is being carried out using much smaller LSTM-Layer sizes to see if performance is improved.



## > TaskKill ./Remaining\_Tasks

- > Develop the search functionality between the User-Interface and the Article Database.
- > Finish optimising the Neural Network to a sufficient standard.
- > Test the Program's functionality and usability through feedback sessions.
- > Finalise Project in a Report.