Detecting Political Bias in Text

Richard Jones

Computer Science (BSc Hon.)

08/03/2020

SCHOOL OF COMPUTING AND MATHEMATICS

Word Count: 7966

Keele University
Keele
Staffordshire
ST5 5BG

Word Count: 7966

1 Initial Pages

1.1 Acknowledgements

I would like to take this section to thank a selection of people who have helped throughout the development of this project. Most importantly, my Project Supervisor Dr. Charles R. Day for guiding through the whole process of this dissertation and providing me with helpful suggestions when I felt that my project was unachievable. I would also like to thank my family and Esther Carter for providing me with support through some of the more stressful sides of this project's development. Finally, I would also like to thank the five participants who gave up their spare time to help evaluate this project's deliverables.

1.2 Abstract

This project focuses around the concept of developing a deep learning model capable of detecting either left or right political bias, specific to the UK, in variable length text. The model was trained on UK political data from the Hansard Parliamentary archive from both the Labour and Conservative parties from the year 2015 to the end of 2019. Overall, the model performed with some relative success, reaching 72% accuracy for some of the experiments.

The project also developed a system to communicate with this model for users to be able to analyse their own inputted text for political bias. The system provided an interface that allowed users to analyse and save their articles with supplementary information as well providing an interface for users to search for them.

2 Table of Contents

3 Introduction

3.1 Problem Context

3.1.1 Background

The past few years of UK politics have become a hotbed of unsurety among the general public. Increasingly, news reports are being criticised for publishing misleading or in some cases untrue events and statistics, giving rise to popular phrases like ‘fake news’ as well as an increase in alternative theories and conspiracies of events that have taken place. This new atmosphere has led the public to believe they may not have been as adept at detecting political bias as they had once thought.

Overall, it appears that people would benefit from a method of detecting political bias and ideology in what they read that, in itself, is unbiased and neutral. However, as bias is a major part of the human condition, an attempt to develop this method through computational efforts may be more fruitful.

This project will aim to use computational methods to design and create a model capable of detecting political bias in text to a certain degree of accuracy (e.g. 70%). The developed model will be specialised specifically for UK politics. This project will also aim to provide a User-Interface platform, called Politika, for an end-user (any member of the public) to detect bias in texts they find.

3.1.2 Academical and Practical Significance

The art of detecting political bias through computational methods falls under the discipline of sentiment analysis through natural language processing (NLP). Some NLP techniques are capable of tracking words carrying strong sentiment and linking them to the subject and objects of phrases that they are referring to. In terms of feasibility, applying these processes to the problem at hand would definitely be achievable.

Furthermore, though there have been attempts at political bias detection from large companies, very few of these products have been released for use by the general public. By developing a platform that facilitates a user to interface with these models, it may allow the general public to take back some control over their political opinions and beliefs.

3.2 Literature Review

3.2.1 Sentiment Analysis

Similar attempts to use sentiment analysis to predict political ideology have already been conducted in research, varying in degrees of success.

As highlighted in a paper from Iyyer et al. at the University of Maryland (Iyyer, et al., 2014), an initial baseline in this form of natural language processing (NLP) was set using simpler models such as Bag-Of-Words, a method that compares only the contents and frequency of words used in a portion of text (Zhang, et al., 2010). As Iyyer et al. (2014) also argue, while the Bag-Of-Words model did yield some success in detecting political bias, political ideology detection could be greatly improved by monitoring other points of information such as syntactical order. It was then a logical conclusion that, to successfully analyse the complexity of modern language, a deep learning NLP technique could be employed.

Iyyer et al. (Iyyer, et al., 2014) proposed and developed a solution for political ideology detection using a Recursive Neural Network (RNN). RNNs were designed with the ability to process multiple points of related data (e.g. time-series data or text) in one iteration (Irsoy & Cardie, 2014). This means that, in theory, an RNN would be capable of processing every word in context to the words preceding and succeeding them by recursively feeding in portions of the sentence.

The developed RNNs were compared against several baselines, including Bag-Of-Words models (which in some cases had added features such as the phrase-level annotations or pseudo-word features) and a baseline that randomly selects *liberal* or *conservative* (Iyyer, et al., 2014). Overall, the RNNs were able to train to an acceptable accuracy, and outperformed all of the provided baselines (Iyyer, et al., 2014).

From this paper's finding, it is clear that deep learning techniques, such as RNNs, are a strong and powerful tool that could be utilised in this project for several reasons. However, the most compelling argument for its adoption in this project is that the art of processing and understanding language and its behaviour is complex to model using simple algorithms. Though more linear models such as Bag-Of-Words are of use in the study of NLP; it would be near impossible to hard-code a similar program or algorithm that could understand and take account of all laws and rules in the English Language, as well as others. By implementing a Neural Network that is capable of: processing text, drawing its own conclusions on what it processes, and creating its own rules to understand the future inputs, it would appear that a Neural Network may perform more optimally with a problem as complex as this. Furthermore, it may even be able to identify less discussed or conventional rules of language, that an expert in the domain may not know or be able to describe.

However, though the performance of RNNs to detect political bias have so far been successful, Iyyer et al. found that the RNN performed sub-optimally on longer form sentences (Iyyer, et al., 2014). The problem that this may be referring is most likely the Vanishing/Exploding Gradients problem outlined Bengio, Simard and Frasconi (Bengio, et al., 1994). Bengio et al. (1994) state that when an RNN is set with a task of an extensive length, the RNN is incapable of propagating useful information between long-term dependencies (which in this case would be the meaning of words in context to the words around it). Though Iyyer et al. (2014) tried to counteract this using phrase-level annotations (3.2.2), it would be more useful to use a similar Recursive Neural Network that is capable of processing longer form text for this project.

Luckily, Revisions of the RNN model have been developed that address this problem, such as the Long Short-Term Memory Network (LSTM). LSTM cells work similarly to RNN cells except for the introduction of *constant error carousels* (and in some iterations forget gates (Gers, et al., 2000)), reducing the problems with vanishing gradients over longer form inputs (Hochreiter & Schmidhuber, 1997). An LSTM model has also been used to address the problem of detecting political ideology in text by Arkajyoti Misra and Sanjib Basak at the University of Stanford (Misra & Basak, 2016).

Similar to the work done at the University of Maryland (Iyyer, et al., 2014), the models developed by Misra and Basak (2016) were compared against a baseline using a Naïve Bag-Of-Words approach. This baseline performed poorly on the IBC dataset (Iyyer, et al., 2014), obtaining an F1-score of 0.3. However, when evaluated using the *On The Issue* (OTI) dataset, which followed labelled US political speeches and dialogues, the model achieved an F1-score of 0.7.

For this research, Misra et al. developed several models, each comprising of a single LSTM layer and a varying number of LSTM units for each model.

An Adam optimiser function was implemented for each model, which would allow the learning rate of the model to adaptively change given the state of the neural networks activity and performance (Kingma & Lei Ba, 2015).

The model developed by Misra et al. uses binary cross-entropy loss function which, like all cross-entropy loss functions, measures the negative log-likelihood between the labels outlined by the training data and the probabilities calculated by the neural network in order to calculate the error in the predictions of the Neural Network (Goodfellow & Bengio, 2016). As the loss function used is binary cross-entropy, the LSTM model has likely been developed for binary classification (Godoy, 2018). With this in mind, it is likely that Misra et al. would have developed their LSTM network to have one output node, which if activated closer to 0 or 1 indicates a separate political ideology (e.g. 0 is left-leaning and 1 is right-leaning). While this is a viable solution to the task at hand, as the objectives only need to predict two eventualities (left or right), binary classification is more well suited to measuring the degree of one variable (e.g. measuring the degree of political bias, without specifying left or right-leaning bias). This project will aim to improve on this paper by using categorical classification, and therefore a categorical cross-entropy loss function, to model two output nodes for the magnitude of both a left and right-leaning bias.

The models developed by Misra and Basak (Misra & Basak, 2016) were run for an unlimited number of epochs with a 10 step early stop procedure in place to avoid overtraining/overfitting. The model also used a static batch-size of 32. Except for the style of classification, Misra and Basak created a robust and compelling argument to adopt an LSTM network as the backbone of the sentiment analysis in this project. Not only was the argument for using an LSTM (given the vanishing gradient problem over lengthier permutations) solid and would allow this project to process longer-form text, but the architecture and hyper-parameters of the model were also described in depth; providing a detailed foundation to improve on their solution. Furthermore, in terms of performance, the model was able to obtain an F1-score of 0.9 (0.1 away from the maximum) on their On The Issue dataset (though only scoring a .568 on the IBC Dataset, which they argue was due to the difference between the training and validation data), making the LSTM network a major possibility for use in this political ideology detection system.

3.2.2 Data and Processing

In order to train their RNN model (mentioned in 3.2.1), Iyyer et al. used two main datasets. The Convote dataset was a collation of transcripts of US Congress Debates, taken from 2005 onwards. The dataset also had the political inclination (Democratic or Republican) of the speaker labelled alongside the text of each datapoint (Iyyer, et al., 2014). All of the debates were separated into sentences and filtered using a tool to assess the existence and level of explicit bias through detection of trigger-words (Yano, et al., 2010). The sentences were then further filtered using other statistical methods and balanced so an equal amount of Democratic and Republican articles were present in the dataset (totalling at 7,816 articles).

Overall, the use of political datasets similar to Convote could be a potential solution for this project. Collating data from a widespread view of the overall political atmosphere at that point in time could be achieved and therefore train the developed network to generalise of a wider scope of political topics. However, in order to ensure that the data used for training is of an appropriate relevancy, it would be wise to use a smaller time-frame than Iyyer et al., as by using a time-frame as big as 9 years (2005-2014) could potentially reduce the relevancy of the political topics in question. For example, in 2011 the UK's political conversations were very different. Debates that are now mostly dominated by topics such as *Brexit* would have been related more to other events of the time such as the *War in Afghanistan*. If this project is to adopt any political dataset, it should contain a much smaller time-frame (e.g. 4-5 years).

Iyyer et al. also used data from the Ideological Book Corpus (IBC) (Iyyer, et al., 2014) which consisted of articles by authors infamous for their political leanings. The IBC underwent the same filtering procedure as Convote, producing a dataset of 55,932 sentences. However, the data was filtered further using the DUALIST network to reduce the number of neutral sentences, ensuring the remaining sentences had implicit bias.

The IBC was annotated at the sentence level (like Convote) but was also further annotated at the phrase level by a crowdsourcing application, which ultimately raised the accuracy achieved by the RNN when compared to an RNN trained on sentence level annotations.

Though the results from using these datasets are positive, there could be complications in using such finely filtered data for this project. As stated in the problem context of this report (3.1), The models used in the final artefact of this project will need to be capable of being retrained for more recent political events and therefore will need more recent data to be retrieved. By replicating the levels of filtering performed on the discussed datasets, the process of updating models would become tedious and ineffective. Furthermore, by reducing the level of filtering performed on this project’s chosen data, it may be possible to investigate the performance of more generic data points (which could potentially be beneficial to the project). Arguably, the RNN’s performance by Iyyer et al. benefitted from the phrase level annotations created using crowdfunding. However, this issue is lessened by the LSTM cell’s ability to propagate over a larger number of inputs (Hochreiter & Schmidhuber, 1997).

3.2.3 Conclusion

In conclusion, the proposed problem could be solved using a number of different methods. However, the introduction of a deep learning type system would appear to be the most accurate and fruitful. Though Iyyer et al. (2014) produced an RNN capable of detecting political bias, the fact that the model performed worse on longer text is great concern. As the project will require the model to be constantly retrained to keep up with more modern political trends, performing extra processing such as the crowdfunding performed by Iyyer et al. (2014) on the data to get a reasonable result will not be as viable in this project. Employing a LSTM, similar to the paper written by Misra & Basak (2016), would allow this project to achieve similar or better results without this arduous text processing. Furthermore, it may be possible to improve on previous work performed with an LSTM (Misra & Basak, 2016) by providing data labelled over a longer range (paragraph-level over sentence-level), By doing so, an LSTM may be able to provide an even more in-depth analysis of sentiment, increasing accuracy in bias detection.

Though all of the sources discussed used a wide variety of datasets, it would appear that using a political data source would be the most beneficial to this project. This is because these data sources are usually easier to label (as the writers/speakers belong to a political party), plentiful in data and constantly mimic the current political atmosphere.

Another key point to be made is that the discussed papers also only focus on data relevant to US Politics. As this paper will focus on UK politics, a more appropriate dataset will need to be used. Luckily, the Hansard Archive, which holds all debates from the UK parliament has been made publicly available. Hansard would appear to be a great source to create the dataset for this project.

3.3 Main Objectives/Requirements

1. **Design and create a LSTM model capable of detecting political bias in text.**
   1.1. The model shall be capable of taking text of variable length as an input.
   1.2. The model shall be capable of processing the input in a necessary manner to ensure it can be interpreted by the model in the most appropriate way.
   1.3. The model shall be trained on processed, labelled data from the Hansard UK Parliamentary Archive, from the 5th January 2015 to the 5th November 2019
   1.4. The model shall indicate its results in the form of two probability scores (between 0 and 1), indicating the likeliness of either a left or right bias.
2. **The project shall provide an interface for users to analyse their own texts for political bias.**
   2.1. The user shall be provided with a screen facilitating the inputting of their chosen text to be analysed.
   2.2. The interface shall provide the user with a choice to analyse their chosen text either at a sentence level or at a passage level.

2.3. The interface shall provide an analytics panel, displaying: the results of the model after processing the user's chosen text as well as the number of words analysed by the model.

2.4. The interface shall provide a message area to log relevant information to the program, including updates on the analysis process.

2.5. The interface shall provide relevant information on the model to give the user added understanding of the program's functionality.

2.6. The interface shall provide relevant information on any text processing produced by the model to give the user added understanding of the program's functionality.

3. **The interface shall provide a way to save analysed text alongside relevant metadata, forming an article.**

3.1. Only input data that has been analysed as one passage (Requirement 2.2) can be saved.

3.2. Input data that has been changed since its analysis will not be permitted for saving.

3.3. The interface shall request a name for the article, validating that the inputted name is more than 5 characters.

3.4. The interface shall request a date for when the article was created.

3.5. The interface shall request an input for the author(s) of the text/article, in the form

3.6. The interface shall request any political affiliation (E.g. Party) of the author at the time of the article's creation.

3.7. The interface shall only save articles that have inputs for the afore-mentioned fields.

4. **The project shall provide a database to store data analysed by the users.**
    4.1. The database will store the analysed text alongside the results of the model's analysis, the article name, the article date, the author and the author's political party.
5. **The project shall provide an interface for the user to search for records in the database.**
    5.1. The interface will allow users to search for records by their name, date, author or the political party of the author.
    5.2. The interface will retrieve all results relevant to the user's search query.
    5.3. The interface will allow users to select any retrieved records to view a detailed report on them.
6. **The project shall be developed in a manner to allow the model to be interchangeable with other models for increased longevity.**

4 Analysis & Design

4.1 Development Methodology

This project will use an Agile Methodology (Agile Alliance, n.d.) to develop the deliverables outlined in the problem definition of this report (3.3). This decision was made to ensure that the deliverables could be developed as quickly as possible, allowing more time for evaluation of each iteration while reducing the risk of potential roadblocks to the development process while also ensuring the maximum quality possible of the project's deliverables.

4.1.1 Discussion of Development Approach

During this analysis, a list of requirements will be created to assess the success of the development stage and outline all deliverables as a result of the project. These requirements will then be translated into a product backlog during the design phase, where the practical aspects of the project will be further solidified (including the technologies and tools that will be used to develop the deliverables). The product backlog and requirements will be separated roughly by each feature of the program and will be developed in these orders. In the event that any mission-critical issues arise during the development (such as the LSTM network not performing to a sufficient level), work on the next feature will commence to ensure that the project is fully completed in the correct time-frame. Time has been allocated at the end of the projects development to retrospectively evaluate and improve on each feature of the program. On top of this, feature tests will also be performed during the development stage to ensure that each section of the project is working to an acceptable level. The project will then be evaluated by a selection of potential end-users to assess the usability of the user-interface created and the performance of the program over all.

After this point, the project will have reached the final evaluation, where potential extensions and changes to the program will be discussed to provide a better deliverable in the next iteration.

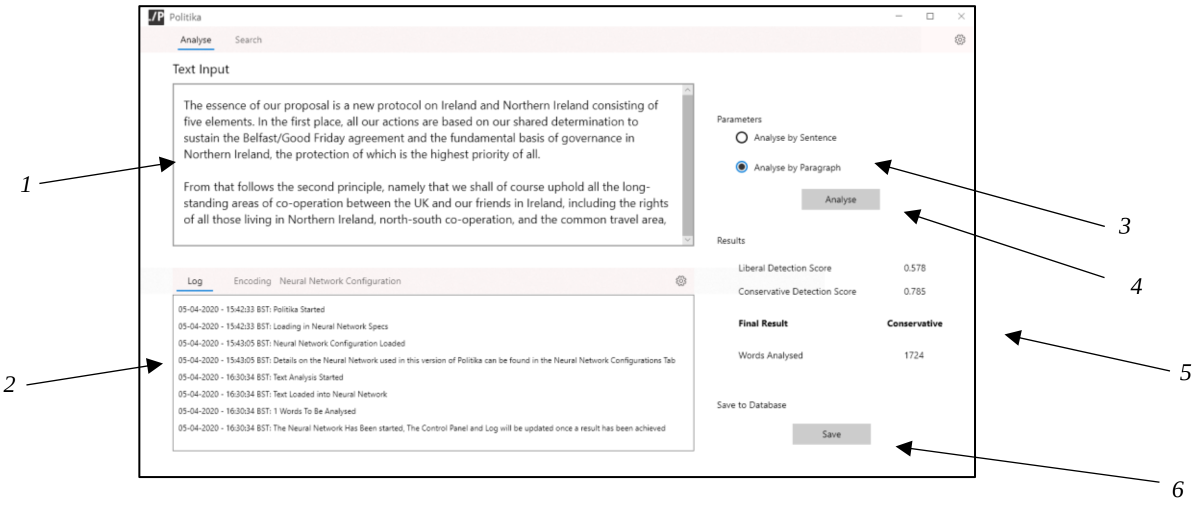
This framework has been put in place to guarantee that the project is finished in the allotted time-frame, while also ensuring the project is developed to the best quality possible.

4.2 Conceptual Design

4.2.1 User-Interface Wireframes

The following diagrams were designed to model the planned User-Interface for the program. They will

4.2.1.1 Analysis Panel



The essence of our proposal is a new protocol on Ireland and Northern Ireland consisting of five elements. In the first place, all our actions are based on our shared determination to sustain the Belfast/Good Friday agreement and the fundamental basis of governance in Northern Ireland, the protection of which is the highest priority of all.

From that follows the second principle, namely that we shall of course uphold all the long-standing areas of co-operation between the UK and our friends in Ireland, including the rights of all those living in Northern Ireland, north-south co-operation, and the common travel area,

Analyse by Sentence [✓ Analyse by Paragraph]

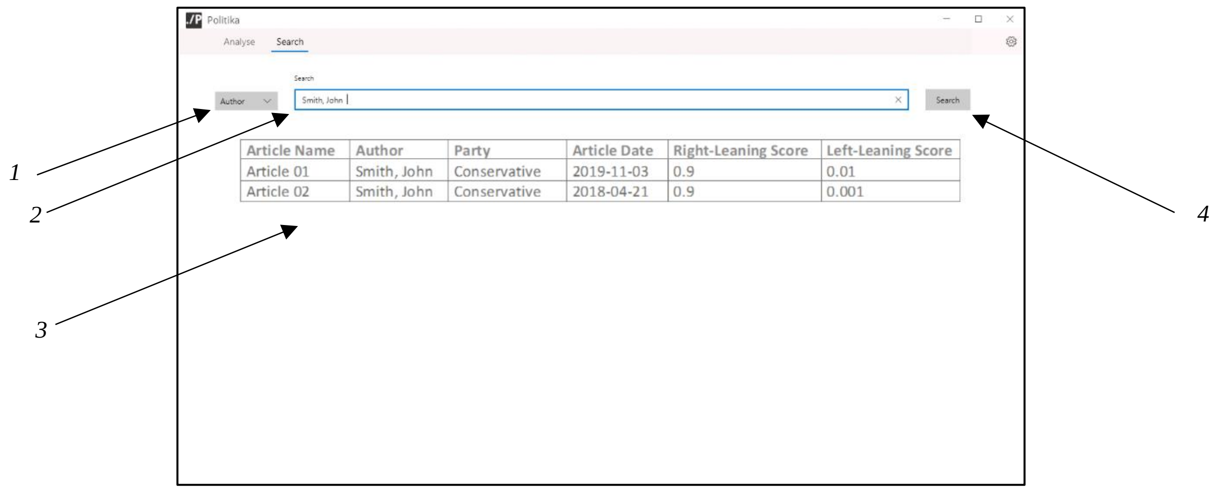
Liberal Detection Score	0.578
Conservative Detection Score	0.785
Final Result	**Conservative**
Words Analysed	1724

```
05-04-2020 - 15:42:33 BST: Politika Started
05-04-2020 - 15:42:33 BST: Loading in Neural Network Specs
05-04-2020 - 15:43:05 BST: Neural Network Configuration Loaded
05-04-2020 - 15:43:05 BST: Details on the Neural Network used in this version of Politika can be found in the Neural Network Configurations Tab
05-04-2020 - 16:30:34 BST: Text Analysis Started
05-04-2020 - 16:30:34 BST: Text Loaded into Neural Network
05-04-2020 - 16:30:34 BST: 1 Words To Be Analysed
05-04-2020 - 16:30:34 BST: The Neural Network Has Been started, The Control Panel and Log will be updated once a result has been achieved
```

1. The analysis panel provides a text area for the user to input their chosen text for analysis.
2. The analysis panel provides a log panel to notify the user of information related to the program (including steps of analysis).

3. A selection interface allowing the user to select between analysis by sentence or passage.
4. A button to initiate the analysis process.
5. A results panel to provide information on the analysis that has taken place priorly, including: the output of the model, the final results and the number of words analysed.
6. A button to save the analysis results.

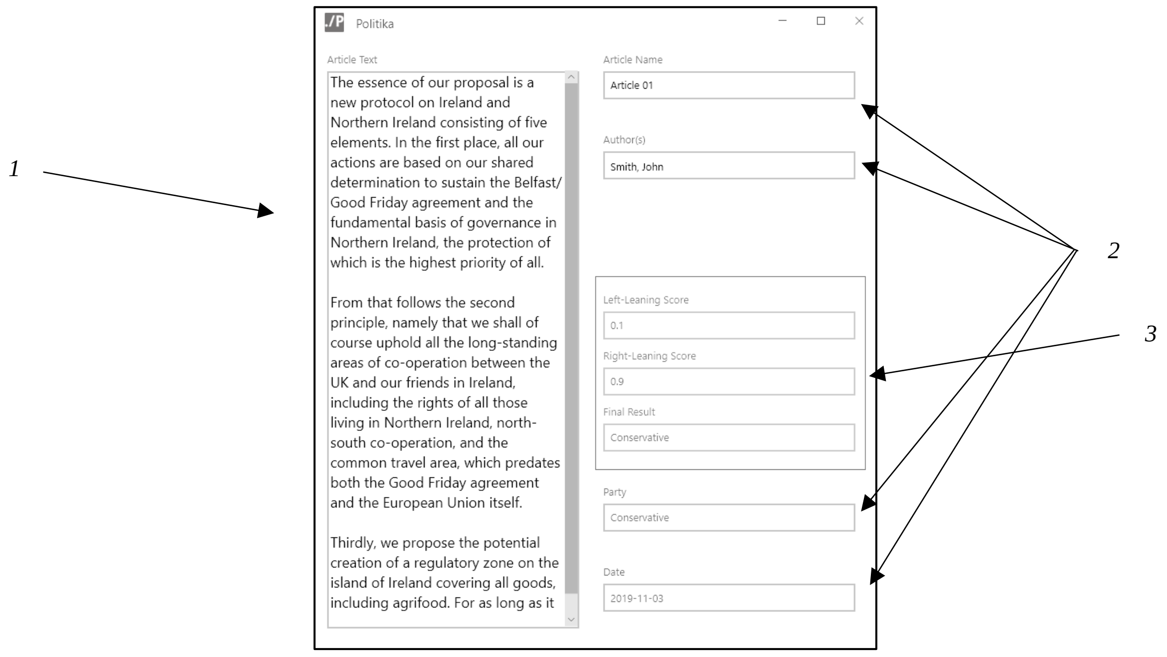
4.2.1.2 Search Panel



Article Name	Author	Party	Article Date	Right-Leaning Score	Left-Leaning Score
Article 01	Smith, John	Conservative	2019-11-03	0.9	0.01
Article 02	Smith, John	Conservative	2018-04-21	0.9	0.001

1. A search term drop-down box, allowing the user to select what field of the record they want to search for.
2. A search box, for the user to input the term they want to search the database for.
3. A results table to display any results of the user's search.
4. A search button for the user to initiate a search.

4.2.1.3 Article Report Panel



Politika

Article Text

The essence of our proposal is a new protocol on Ireland and Northern Ireland consisting of five elements. In the first place, all our actions are based on our shared determination to sustain the Belfast/ Good Friday agreement and the fundamental basis of governance in Northern Ireland, the protection of which is the highest priority of all.

From that follows the second principle, namely that we shall of course uphold all the long-standing areas of co-operation between the UK and our friends in Ireland, including the rights of all those living in Northern Ireland, north-south co-operation, and the common travel area, which predates both the Good Friday agreement and the European Union itself.

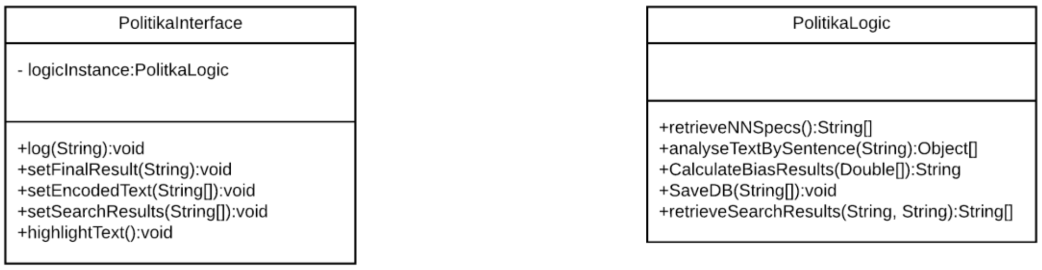
Thirdly, we propose the potential creation of a regulatory zone on the island of Ireland covering all goods, including agrifood. For as long as it

Article Name	Article 01
Author(s)	Smith, John
Left-Leaning Score	0.1
Right-Leaning Score	0.9
Final Result	Conservative
Party	Conservative
Date	2019-11-03

1. An area displaying the article text, equipped with a scroll bar to accommodate variable length text.
2. Text Areas displaying the relevant on the article.
3. Text areas displaying the information on the article's analysis.

4.2.2 Conceptual UML Diagram

The following diagram shows the relationships between the two main classes to be implemented for the User-Interface:

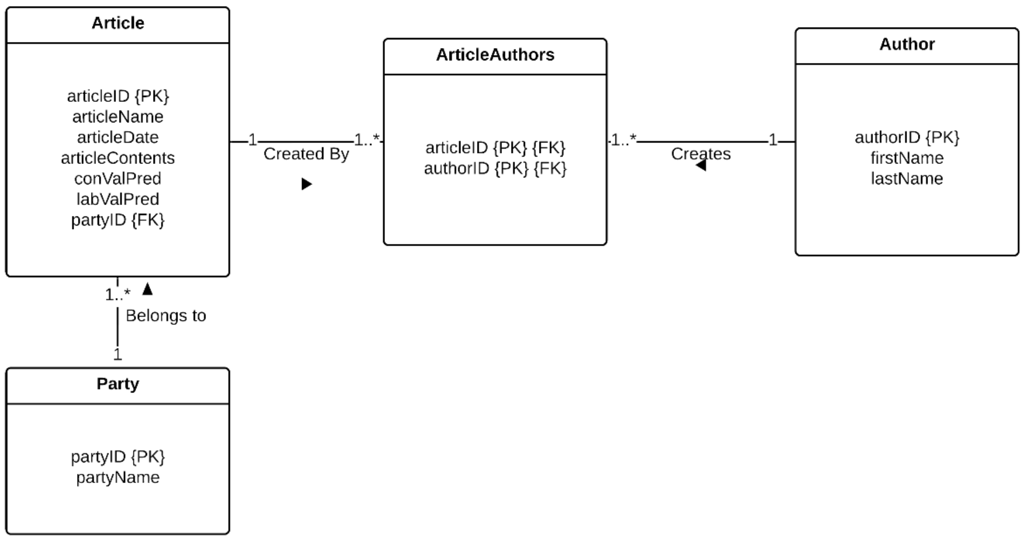

PolitikaInterface
- logicInstance:PolitkaLogic
+log(String):void
+setFinalResult(String):void
+setEncodedText(String[]):void
+setSearchResults(String[]):void
+highlightText():void
PolitikaLogic
+retrieveNNSpecs():String[]
+analyseTextBySentence(String):Object[]
+CalculateBiasResults(Double[]):String
+SaveDB(String[]):void
+retrieveSearchResults(String, String):String[]

During the main initialisation of the program, two objects from the described classes will be created: PolitikaInterface, which handles necessary user inputs and outputs for the program, and PolitikaLogic, which handles all necessary logical operations.

The PolitikaInterface class holds a PolitikaLogic attribute, passed to it through its constructors, in order to execute its logic-methods then the appropriate scenario arise. The communications between these two classes will be further described in the logical design of the project.

4.2.3 Conceptual ERD Diagram

The diagram below describes the Database system to be implemented during the projects development. The ERD diagram aims to represent the data that the program is required to store in a method efficient relational method, keeping to at least 3rd Normal Form where possible.



4.3 Discussion of Technologies

4.3.1 Data Retrieval

As discussed in the literature review of this report (3.2.2), the Hansard parliamentary archive will be used to build the training, validation and test datasets. The debates will be retrieved as XML files from the ParlParse database (mySociety, 2019).

4.3.2 XML Parser

In order to create the full dataset to be used in this project, an XML parser will need to be produced to retrieve all of the relevant debates, and write them, alongside their corresponding party labels into a single csv file.

The parser will be written in Java to make error handling and debugging as straightforward as possible, reducing the time spent coding and fixing the parser. In addition to retrieving debates, the parser will also need to retrieve the necessary data on the speakers of the debates, which is kept in a json format. To parse through this json file, the json-simple java library will be used (fangyidong, 2014).

4.3.3 Neural Network

To achieve the functionality necessary to model an LSTM neural network, the TensorFlow library for the python programming language will be used (Abadi, et al., 2015). Furthermore, the TensorFlow-Datasets library will also be used to model the input pipeline for the dataset correctly.

4.3.4 Database Solution

In order to facilitate the rapid development of this project, SQLite will be used to develop the article database described in the main requirements for this project (**Error! Reference source not found.**, Requirement 4).

4.3.5 User Interface

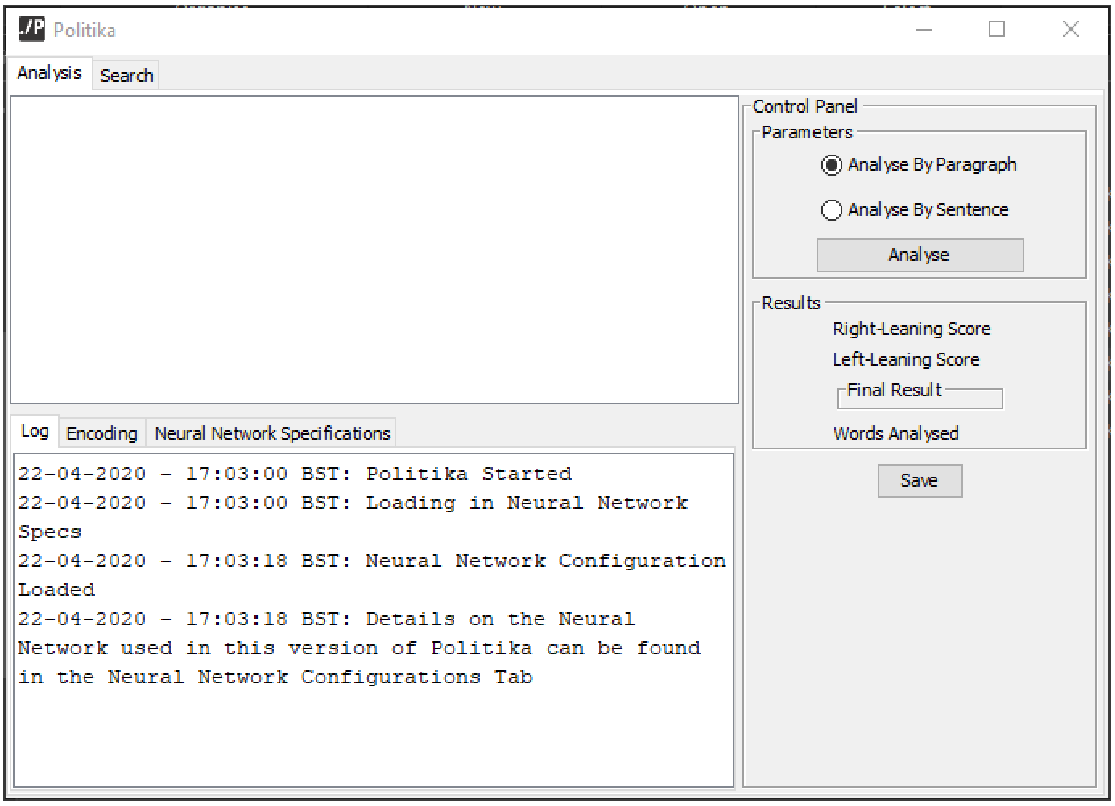
To provide a secure and responsive application, the User-Interface for the project will be developed using the Java programming language alongside the Swing standard library for the relevant GUI components and respective listeners.

4.4 Logical Design

4.4.1 User-Interface Forms

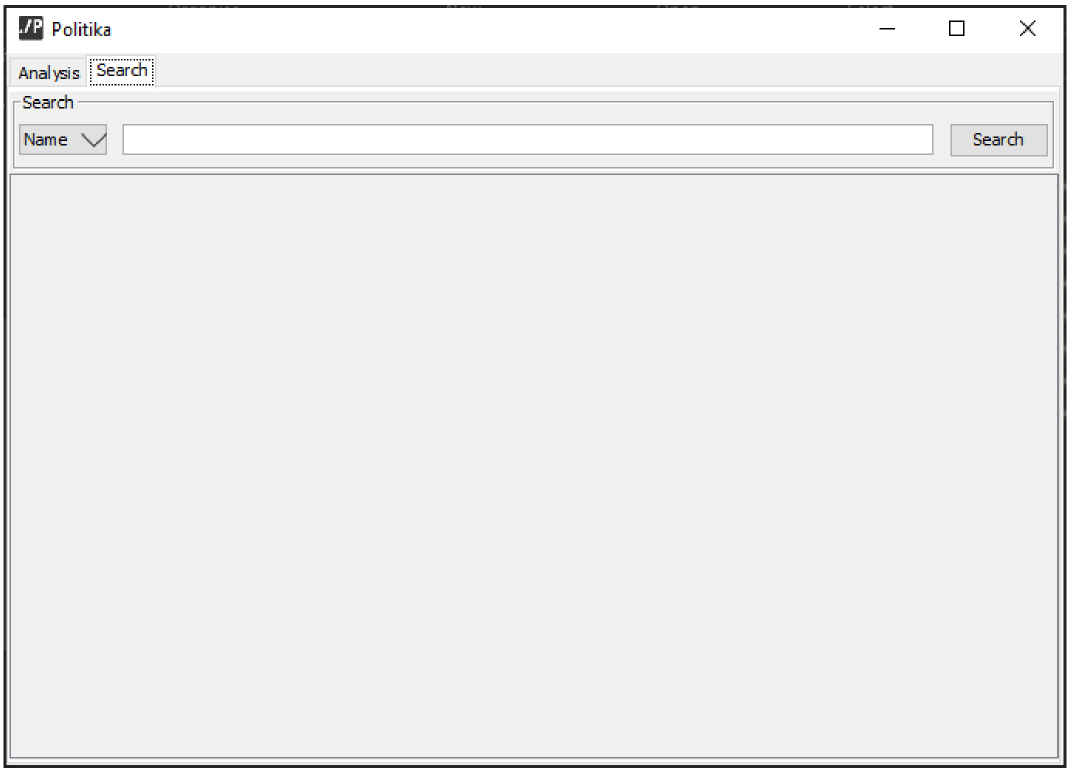
As a further development to the wireframes designed in 4.2.1, the following User-Interfaces were developed using IntelliJ's Form Designer (JetBrains, 2001). These forms will also be used as the final User-Interface for the project.

4.4.1.1 Analysis Panel

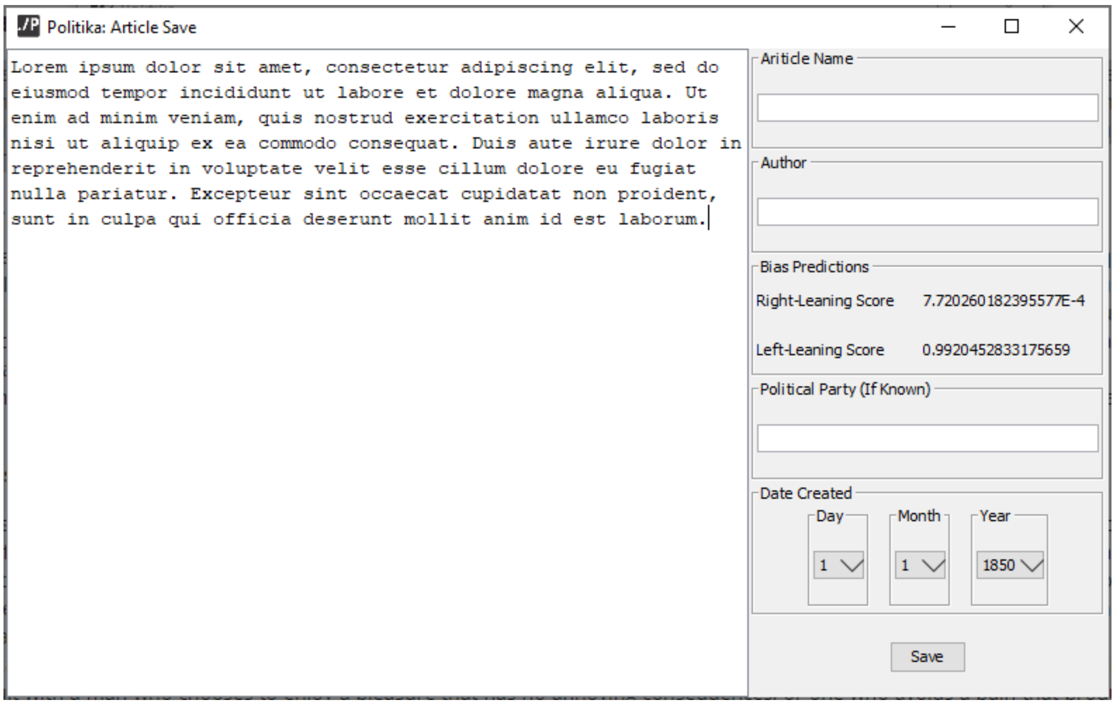


```
22-04-2020 - 17:03:00 BST: Politika Started
22-04-2020 - 17:03:00 BST: Loading in Neural Network
Specs
22-04-2020 - 17:03:18 BST: Neural Network Configuration
Loaded
22-04-2020 - 17:03:18 BST: Details on the Neural
Network used in this version of Politika can be found
in the Neural Network Configurations Tab
```

4.4.1.2 Search Panel



4.4.1.3 Save Panel

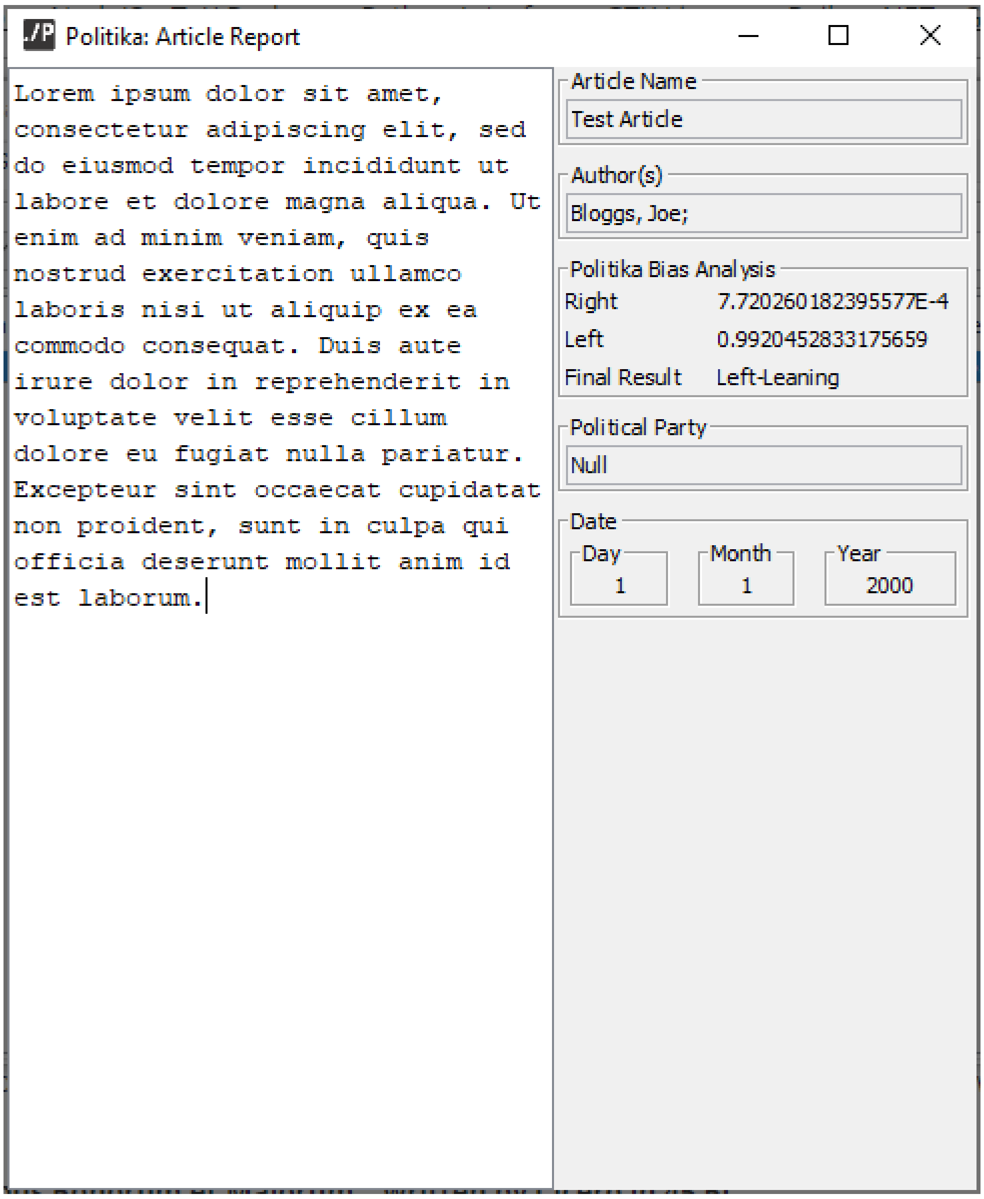


```
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat
nulla pariatur. Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum.
```

Right-Leaning Score	7.720260182395577E-4
Left-Leaning Score	0.9920452833175659

Day: 1 Month: 1 Year: 1850

4.4.1.4 Report Panel


Politika: Article Report

```
Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed
do eiusmod tempor incididunt ut
labore et dolore magna aliqua. Ut
enim ad minim veniam, quis
nostrud exercitation ullamco
laboris nisi ut aliquip ex ea
commodo consequat. Duis aute
irure dolor in reprehenderit in
voluptate velit esse cillum
dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat
non proident, sunt in culpa qui
officia deserunt mollit anim id
est laborum.
```

Article Name: Test Article

Author(s): Bloggs, Joe;

Politika Bias Analysis

Right	7.720260182395577E-4
Left	0.9920452833175659
Final Result	Left-Leaning

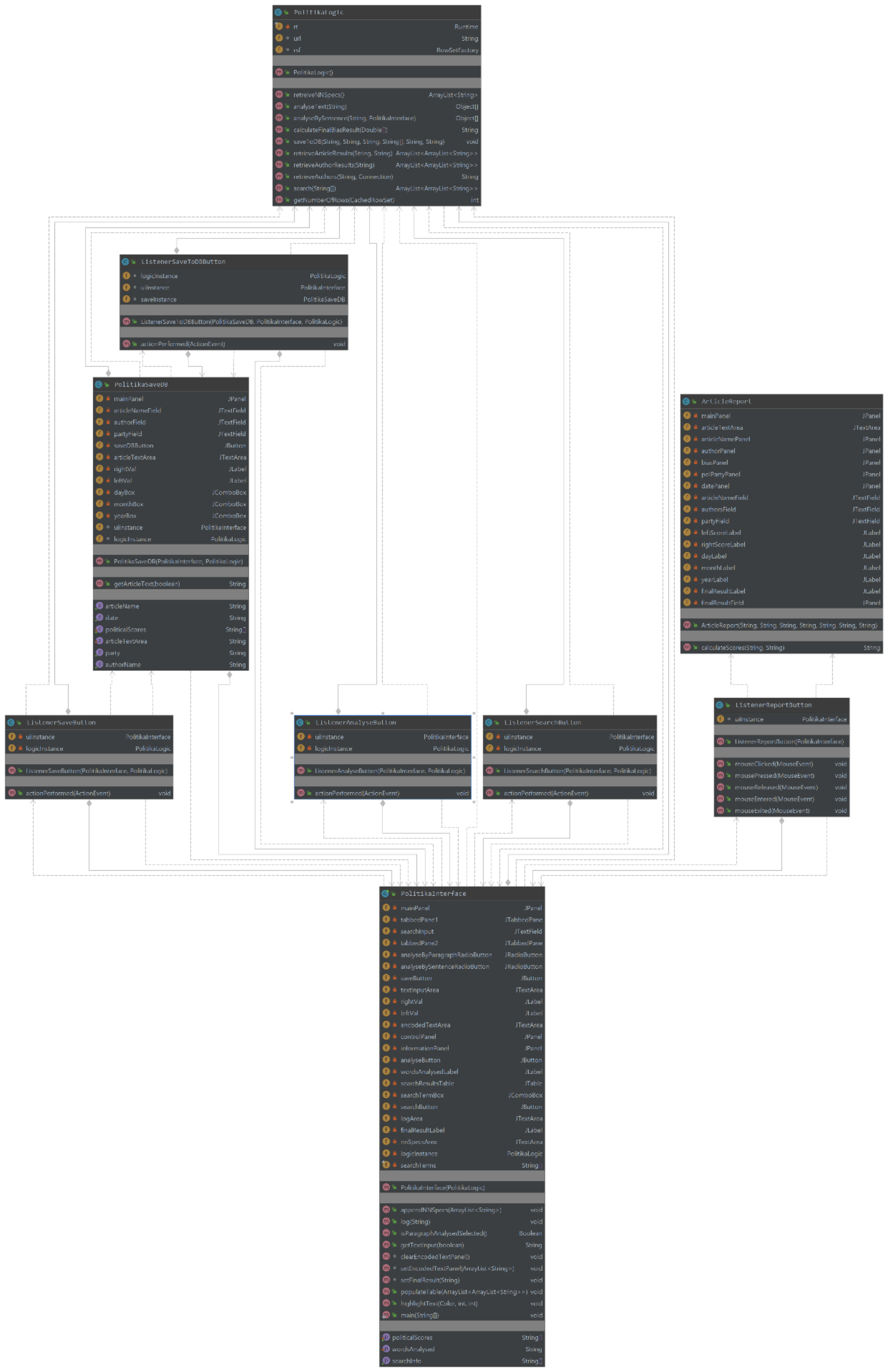
Political Party: Null

Date

Day	Month	Year
1	1	2000

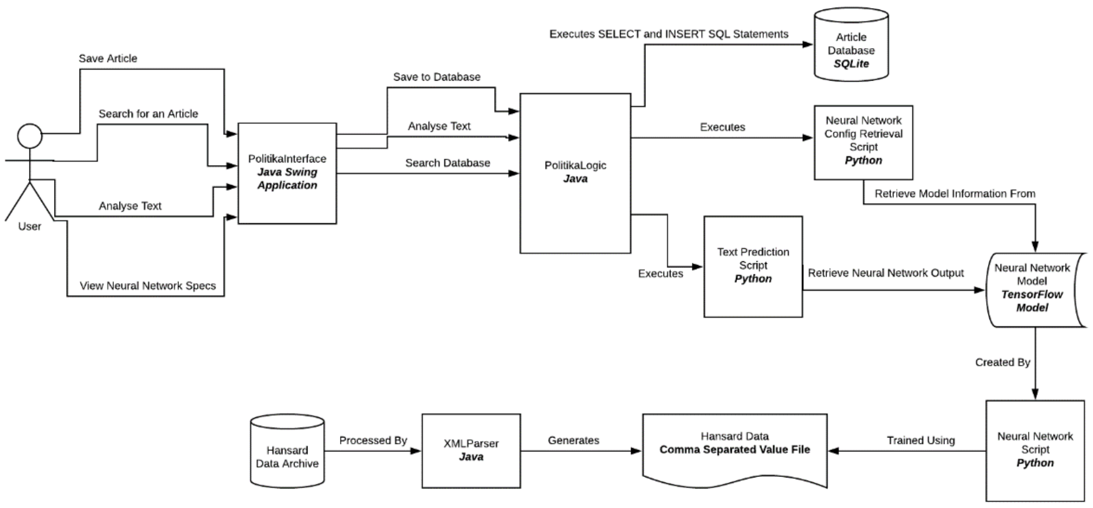
4.4.2 Logical UML Class Diagrams

This diagram displays a more in-depth view of the User-Interface classes, including the addition of Swing UI components and Swing Listener classes that link the front-end class with the logic classes.


PolitikaLogic
rt Runtime
url String
rsf RowSetFactory
PolitikaLogic()
retreiveNNSpecs() ArrayList<String>
analyseText(String) Object[]
analyseBySentence(String, PolitikaInterface) Object[]
calculateFinalBiasResult(Double[]) String
saveToDB(String, String, String, String[], String, String) void
retrieveArticleResults(String, String) ArrayList<ArrayList<String>>
retrieveAuthorResults(String) ArrayList<ArrayList<String>>
retrieveAuthors(String, Connection) String
search(String[]) ArrayList<ArrayList<String>>
getNumberOfRows(CachedRowSet) int
ListenerSaveToDBButton
logicInstance PolitikaLogic
uiInstance PolitikaInterface
saveInstance PolitikaSaveDB
ListenerSaveToDBButton(PolitikaSaveDB, PolitikaInterface, PolitikaLogic)
actionPerformed(ActionEvent) void
PolitikaSaveDB
mainPanel JPanel
articleNameField JTextField
authorField JTextField
partyField JTextField
saveDBButton JButton
articleTextArea JTextArea
rightVal JLabel
leftVal JLabel
dayBox JComboBox
monthBox JComboBox
yearBox JComboBox
uiInstance PolitikaInterface
logicInstance PolitikaLogic
PolitikaSaveDB(PolitikaInterface, PolitikaLogic)
getArticleText(boolean) String
articleName String
date String
politicalScores String[]
articleTextArea String
party String
authorName String
ArticleReport
mainPanel JPanel
articleTextArea JTextArea
articleNamePanel JPanel
authorPanel JPanel
biasPanel JPanel
polPartyPanel JPanel
datePanel JPanel
articleNameField JTextField
authorsField JTextField
partyField JTextField
leftScoreLabel JLabel
rightScoreLabel JLabel
dayLabel JLabel
monthLabel JLabel
yearLabel JLabel
finalResultLabel JLabel
finalResultField JPanel
ArticleReport(String, String, String, String, String, String, String)
calculateScores(String, String) String
ListenerReportButton
uiInstance PolitikaInterface
ListenerReportButton(PolitikaInterface)
mouseClicked(MouseEvent) void
mousePressed(MouseEvent) void
mouseReleased(MouseEvent) void
mouseEntered(MouseEvent) void
mouseExited(MouseEvent) void
ListenerSaveButton
uiInstance PolitikaInterface
logicInstance PolitikaLogic
ListenerSaveButton(PolitikaInterface, PolitikaLogic)
actionPerformed(ActionEvent) void
ListenerAnalyseButton
uiInstance PolitikaInterface
logicInstance PolitikaLogic
ListenerAnalyseButton(PolitikaInterface, PolitikaLogic)
actionPerformed(ActionEvent) void
ListenerSearchButton
uiInstance PolitikaInterface
logicInstance PolitikaLogic
ListenerSearchButton(PolitikaInterface, PolitikaLogic)
actionPerformed(ActionEvent) void
PolitikaInterface
mainPanel JPanel
tabbedPane1 JTabbedPane
searchInput JTextField
tabbedPane2 JTabbedPane
analyseByParagraphRadioButton JRadioButton
analyseBySentenceRadioButton JRadioButton
saveButton JButton
textInputArea JTextArea
rightVal JLabel
leftVal JLabel
encodedTextArea JTextArea
controlPanel JPanel
informationPanel JPanel
analyseButton JButton
wordsAnalysedLabel JLabel
searchResultsTable JTable
searchTermBox JComboBox
searchButton JButton
logArea JTextArea
finalResultLabel JLabel
nnSpecsArea JTextArea
logicInstance PolitikaLogic
searchTerms String[]
PolitikaInterface(PolitikaLogic)
appendNNSpecs(ArrayList<String>) void
log(String) void
isParagraphAnalysedSelected() Boolean
getTextInput(boolean) String
clearEncodedTextPanel() void
setEncodedTextPanel(ArrayList<String>) void
setFinalResult(String) void
populateTable(ArrayList<ArrayList<String>>) void
highlightText(Color, int, int) void
main(String[]) void
politicalScores String[]
wordsAnalysed String
searchInfo String[]

4.4.3 Project Flow Diagram

The following diagram outlines every component of the practical side of the project, alongside any possible decision made in the project, based on both the Requirements and Design discussion to this point.

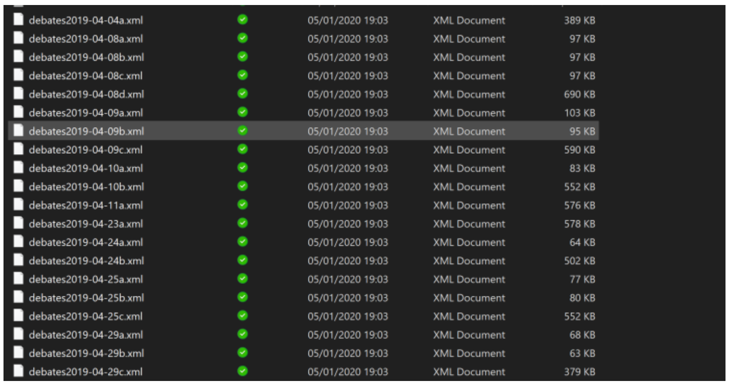


5 Development

5.1 Data Retrieval and Initial Processing

5.1.1 Retrieving Hansard Data

As described in the Design section of the report, the Hansard data that will comprise the training dataset will be retrieved as XML files from the ParlParse Text Parser. The data was retrieved from the ParlParse website using rsync and filtered to only pull debates in the inclusive range of 5th January 2015 to the 5th November 2019 (the most recent date at the time of development). The data was stored on a local drive to be accessed by XML Parser in the next stage of development.



```
debates2019-04-04a.xml    05/01/2020 19:03   XML Document   389 KB
debates2019-04-08a.xml    05/01/2020 19:03   XML Document    97 KB
debates2019-04-08b.xml    05/01/2020 19:03   XML Document    97 KB
debates2019-04-08c.xml    05/01/2020 19:03   XML Document    97 KB
debates2019-04-08d.xml    05/01/2020 19:03   XML Document   690 KB
debates2019-04-09a.xml    05/01/2020 19:03   XML Document   103 KB
debates2019-04-09b.xml    05/01/2020 19:03   XML Document    95 KB
debates2019-04-09c.xml    05/01/2020 19:03   XML Document   590 KB
debates2019-04-10a.xml    05/01/2020 19:03   XML Document    83 KB
debates2019-04-10b.xml    05/01/2020 19:03   XML Document   552 KB
debates2019-04-11a.xml    05/01/2020 19:03   XML Document   576 KB
debates2019-04-23a.xml    05/01/2020 19:03   XML Document   578 KB
debates2019-04-24a.xml    05/01/2020 19:03   XML Document    64 KB
debates2019-04-24b.xml    05/01/2020 19:03   XML Document   502 KB
debates2019-04-25a.xml    05/01/2020 19:03   XML Document    77 KB
debates2019-04-25b.xml    05/01/2020 19:03   XML Document    80 KB
debates2019-04-25c.xml    05/01/2020 19:03   XML Document   552 KB
debates2019-04-29a.xml    05/01/2020 19:03   XML Document    68 KB
debates2019-04-29b.xml    05/01/2020 19:03   XML Document    63 KB
debates2019-04-29c.xml    05/01/2020 19:03   XML Document   379 KB
```

Figure 5-1: A sample of the retrieved Hansard XML files

The ParlParse People JSON file, which contains details on all Members of Parliament (past and present) was also downloaded to provide the relevant parties for all of the speeches retrieved. The XML files provided did not contain the identity of the speeches but instead represented them as a number. This could then be queried using the People JSON file to retrieve information on the speaker, including their political party.

5.1.2 XML Parser Procedure

The parsing-procedure iteratively scans every debate-file in the directory, retrieving elements that coincide with speeches. For each, the speaker's unique ID is retrieved and compared to the JSON-file discussed earlier; their respective party is retrieved. The party is encoded (0.5 for right/conservative, 1.0 for left/labour) and the collected data is then pushed into a stack dependent on their party. Each stack is then shuffled and equally popped and written into a csv file. Producing a shuffled and equal dataset for each political standing.

5.2 Neural Network Development and Training

5.2.1 Text Encoder

In order to represent the text of the dataset appropriately to ensure it is understood by the neural network, a text encoder – trained on the words of the dataset – was necessary. The TensorFlow library includes several different methods of text representation. However, the Sub-Word Text Encoder was the best choice for this project as it allows words outside of the trained encoder-vocabulary to be encoded; allowing the model to predict new passages inputted by the user.

To train the encoder, the contents of the dataset were loaded into a python script using a data frame from the Pandas python library. From this data-frame, a TensorFlow dataset was constructed.

The constructed dataset was then iterated through, tokenising the text found in each dataset record and adding it to a vocab Set; which was then used to initialise the encoder. The encoder is then saved to a file to be used in the Neural Network model training.

5.2.2 Neural Network Input Pipeline

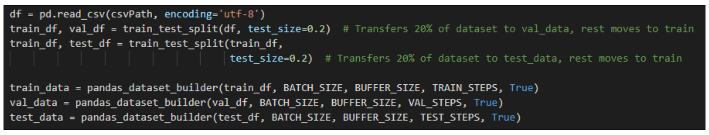
5.2.2.1 CSV to TensorFlow Dataset Translation

In order for the training data to be used with the TensorFlow model, it was necessary for the data to be translated into a TensorFlow dataset model.

Initially, the development of the input pipeline tried to use TensorFlow's own make_csv_dataset method to perform this task. This method was able to successfully place all of the data into a special CSV Dataset variable. However, further complications were met when trying to batch and encode the dataset. This is most likely because the CSV Dataset variable was experimental and did not have the functionality of more generic TensorFlow dataset models; as well as the model's layout being slightly different.

An alternative method was then used, incorporating the Pandas Python Library (pandas, 2020). Using panda's dataset manipulation functionality, it was possible to correctly manipulate the csv data directly into a TensorFlow dataset.

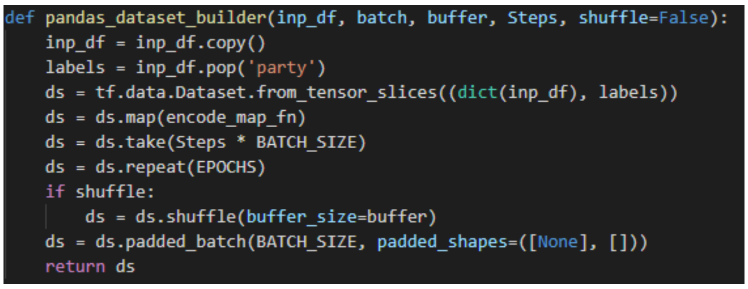
To start the Pandas dataset build, the csv data was read into a pandas data frame. The dataset was then split using the train_test_split function, moving 20% of the original data frame to a validation data frame and 20% of the remaining data to a test data frame. The remaining data was kept for training.



```
df = pd.read_csv(csvPath, encoding='utf-8')
train_df, val_df = train_test_split(df, test_size=0.2)  # Transfers 20% of dataset to val_data, rest moves to train
train_df, test_df = train_test_split(train_df,
                                     test_size=0.2)  # Transfers 20% of dataset to test_data, rest moves to train

train_data = pandas_dataset_builder(train_df, BATCH_SIZE, BUFFER_SIZE, TRAIN_STEPS, True)
val_data = pandas_dataset_builder(val_df, BATCH_SIZE, BUFFER_SIZE, VAL_STEPS, True)
test_data = pandas_dataset_builder(test_df, BATCH_SIZE, BUFFER_SIZE, TEST_STEPS, True)
```

Figure 5-2 Pandas Data Frame Creation Code Snippet

Each data frame was then passed into the pandas_dataset_builder function, along with the necessary hyper-parameters such as: the dataset batch size, the shuffle buffer size, the number of steps per epoch and whether the dataset should be shuffled.



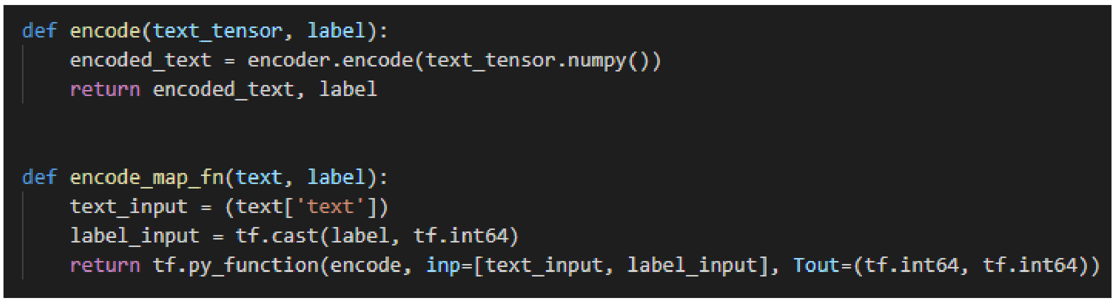
```
def pandas_dataset_builder(inp_df, batch, buffer, Steps, shuffle=False):
    inp_df = inp_df.copy()
    labels = inp_df.pop('party')
    ds = tf.data.Dataset.from_tensor_slices((dict(inp_df), labels))
    ds = ds.map(encode_map_fn)
    ds = ds.take(Steps * BATCH_SIZE)
    ds = ds.repeat(EPOCHS)
    if shuffle:
        ds = ds.shuffle(buffer_size=buffer)
    ds = ds.padded_batch(BATCH_SIZE, padded_shapes=([None], []))
    return ds
```

Figure 5-3 Function Used to Create The TensorFlow Datasets Used in Training

The function is responsible for converting each data frame into a TensorFlow dataset. The party labels are popped from the data frame and the dataset object is created using the from_tensor_slices function. The function is passed a dictionary object representing the text from the data frame, as well as the previously popped labels.

During this function, the dataset is trimmed to only contain enough to process one epoch. The dataset is then repeated for 30 epochs, or until the Early-Stopping mechanism is triggered, ensuring the same dataset is used for each epoch. Finally, the data is batched – based on the batch size specified – and padded with zeroes to ensure that each data point in the batch is of equal size.

The text encoder is also mapped to the dataset using TensorFlow’s map function to map the following methods to each item of the dataset:



```
def encode(text_tensor, label):
    encoded_text = encoder.encode(text_tensor.numpy())
    return encoded_text, label


def encode_map_fn(text, label):
    text_input = (text['text'])
    label_input = tf.cast(label, tf.int64)
    return tf.py_function(encode, inp=[text_input, label_input], Tout=(tf.int64, tf.int64))
```

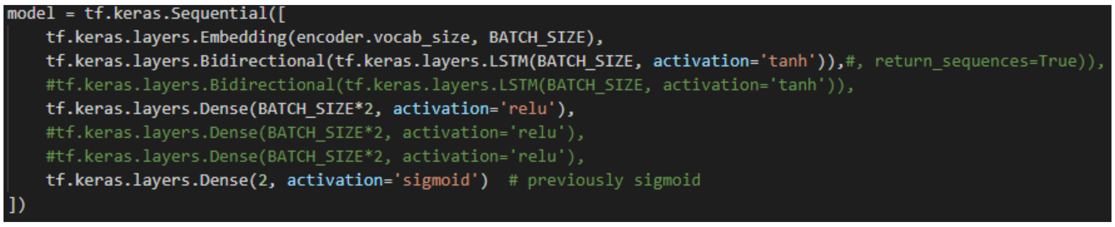
Figure 5-4 Encoding Functions used to Map the Text Encoder to the TensorFlow Dataset

The encode_map_fn receives the speech text and the party label from each item and retrieves the pure primitive value of each. The label is then cast to a TensorFlow integer and the processed data and label are then passed into the encode function The text is then encoded into an integer format and returned with the relevant label.

5.2.3 Neural Network Configuration and Experiment Domain

The development strategy for the network followed an iterative method, where training tests were performed with slight differences on several hyper-parameters. These include the batch size (5, 10 and 15 respectively) and the number of hidden layers in the model (1 or 2). Work was also undertaken using a stacked LSTM model (two LSTM layers) to explore how performance could be improved using a deeper LSTM model.

5.2.3.1 Neural Network Design



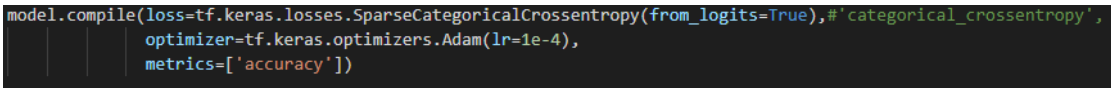
```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(encoder.vocab_size, BATCH_SIZE),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(BATCH_SIZE, activation='tanh')),#, return_sequences=True)),
    #tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(BATCH_SIZE, activation='tanh')),
    tf.keras.layers.Dense(BATCH_SIZE*2, activation='relu'),
    #tf.keras.layers.Dense(BATCH_SIZE*2, activation='relu'),
    #tf.keras.layers.Dense(BATCH_SIZE*2, activation='relu'),
    tf.keras.layers.Dense(2, activation='sigmoid')  # previously sigmoid
])
```

Figure 5-5 The Overall Layout of the Neural Network used in the Project

The developed model starts with an Embedding layer, responsible for embedding the encoded text into fixed-sized vectors. The embedding layer uses the vocab size of the text encoder, to correctly embed the full set of words, and initialises the layer size accordance to the batch size. The output of the embedding layer is then fed into a bidirectional LSTM layer. All LSTM layers used in this project's models will employ a tanh activation function, the usual standard for LSTM models. The output of the LSTM layer is then fed into the hidden Dense layer(s), employing a relu activation function. Finally, another dense layer is constructed to provide the output nodes of the network. This layer consists of two nodes (representing the likelihood of either a right or left bias) and uses a sigmoid function to generate the outputs.

5.2.4 Model Training & Evaluation Methodology

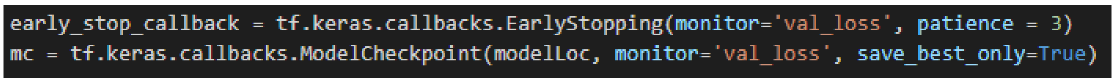
The model was trained using an Adam Optimiser, with an initial Learning Rate of *0.0001*, to allow the training to adaptively change and improve against previous performance. As the model has been configured to detect bias against two categories (left & right), a Categorical Cross-Entropy Loss Function will also be employed.



```
model.compile(loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),#'categorical_crossentropy',
              optimizer=tf.keras.optimizers.Adam(lr=1e-4),
              metrics=['accuracy'])
```

Figure 5-6 Section of the Neural Network Code Describing the Loss and Optimiser Functions Employed

The model was trained using a fixed batch size of 15, for an unlimited number of epochs, with a 3-step early-stopping mechanism that monitored the validation loss of the model. When the model was stopped, the highest performing checkpoint in the model was saved using the ModelCheckpoint call-back in TensorFlow.



```
early_stop_callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience = 3)
mc = tf.keras.callbacks.ModelCheckpoint(modelLoc, monitor='val_loss', save_best_only=True)
```

Figure 5-7 Call-backs Employed to Develop the Early-Stopping Mechanism

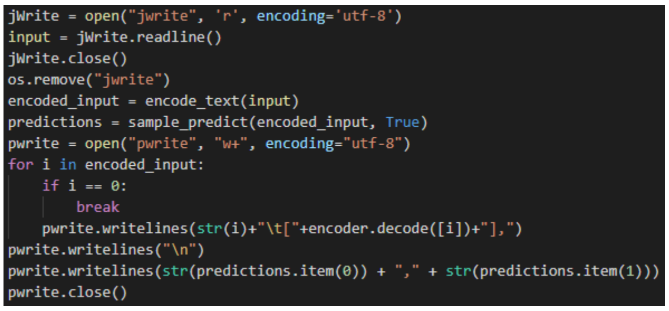
The model was trained using several different methodologies, differing in the amount of dense-hidden layers in the model as well as using a stacked LSTM-layer (more than one LSTM-layer). The results of these models will be reported during the evaluation of this project, and the highest performing will be used in the end artifact.

5.3 Python Neural Network Service Scripts

5.3.1 Prediction Script for User Interface

The predictor script was developed to provide the java program with a method of analysing text using the TensorFlow model. The script starts by loading the saved model and encoder.

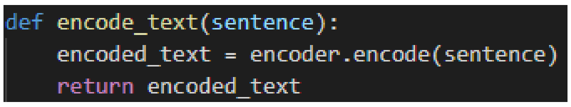
To communicate with the main Java program, two files were used: one for the Java program to write the text to be analysed and one for the python program to write the analysis results and encoded text (*jwrite and pwrite* respectively).The script verifies that that no pwrite file currently exists and deletesit if it is found. A similar check is carried out for the jwrite file, if the file exists then the main body of the script is executed.



```
jWrite = open("jwrite", 'r', encoding='utf-8')
input = jWrite.readline()
jWrite.close()
os.remove("jwrite")
encoded_input = encode_text(input)
predictions = sample_predict(encoded_input, True)
pwrite = open("pwrite", "w+", encoding="utf-8")
for i in encoded_input:
    if i == 0:
        break
    pwrite.writelines(str(i)+"\t["+encoder.decode([i])+"],")
pwrite.writelines("\n")
pwrite.writelines(str(predictions.item(0)) + "," + str(predictions.item(1)))
pwrite.close()
```

Figure 5-8 Code Segment Displaying the full function of the Prediction Script used for analysing input-text

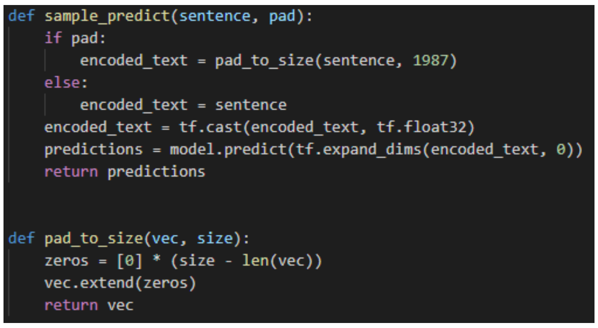
The main body of the script is responsible for processing the input data, loading the data into the model and organising the model's output for retrieval by the main Java program. The main body begins by retrieving the contents of the jwrite file and deleting it. The retrieved text is then encoded through the encode_text function, performing the same task as the mapping method used in the Input Pipeline (5.2.2):



```python
def encode_text(sentence):
    encoded_text = encoder.encode(sentence)
    return encoded_text
```

Figure 5-9 Function used by the prediction script to encode input-text

The script then calls the sample_predict function, which analyses the input text using the LSTM model:



```python
def sample_predict(sentence, pad):
    if pad:
        encoded_text = pad_to_size(sentence, 1987)
    else:
        encoded_text = sentence
    encoded_text = tf.cast(encoded_text, tf.float32)
    predictions = model.predict(tf.expand_dims(encoded_text, 0))
    return predictions


def pad_to_size(vec, size):
    zeros = [0] * (size - len(vec))
    vec.extend(zeros)
    return vec
```

Figure 5-10 Function used by the prediction script to process and analyse the input text

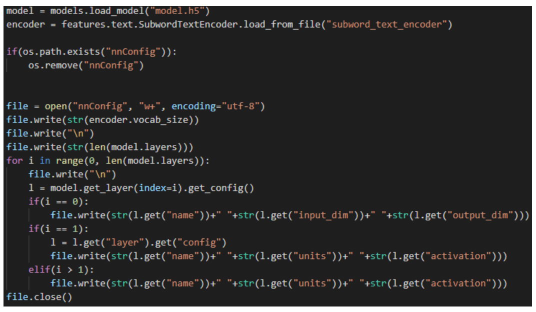
The sample_predict function, takes the encoded text as an input and pads it (if necessary) before analysis. As the model is trained on padded data, it is recommended to pad input data to reduce the risk of skew. The encoded text is cast to a TensorFlow float variable and passed into the model through the predict function. The model's analysis is then retrieved assigned to the prediction variable.

The main body of the program then retrieves the encoded properties of the inputted text which is appended to the pwrite file with the model's predictions.

5.3.2 Neural Network Configuration Retrieval Script

The Neural Network Configuration scripts is responsible to retrieving necessary information about the LSTM model, including the breakdown of the layers and the vocab size of the sub-word encoder.

To start off the script, the LSTM model and sub-word text encoder are loaded and assigned to variables. Then, the existence of the nnConfig file (which will hold the information for reading by the Java program) is checked and removed if it is; the file is then created again. The number of layers in the model and the vocab size of the encoder is then obtained and written to the file. The layers of the model are then iterated through, retrieving information such as the name, input size and other information. The nnConfig file is then closed, ending the execution of the script.

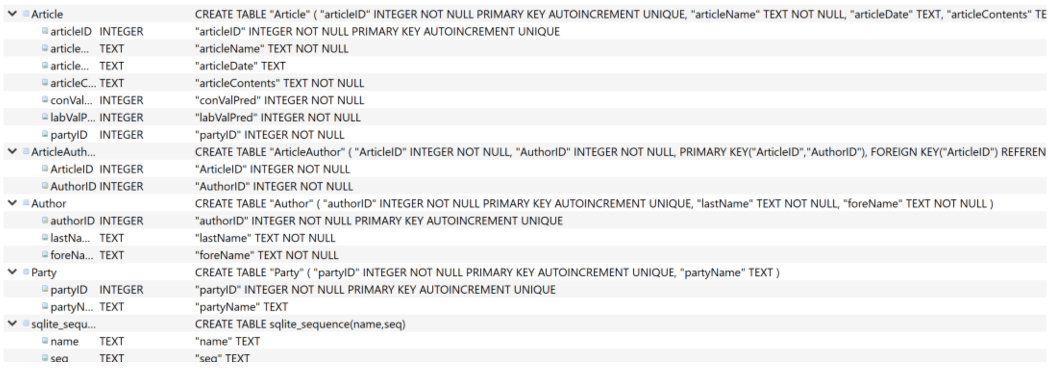


```
model = models.load_model("model.h5")
encoder = features.text.SubwordTextEncoder.load_from_file("subword_text_encoder")

if(os.path.exists("nnConfig")):
    os.remove("nnConfig")


file = open("nnConfig", "w+", encoding="utf-8")
file.write(str(encoder.vocab_size))
file.write("\n")
file.write(str(len(model.layers)))
for i in range(0, len(model.layers)):
    file.write("\n")
    l = model.get_layer(index=i).get_config()
    if(i == 0):
        file.write(str(l.get("name"))+" "+str(l.get("input_dim"))+" "+str(l.get("output_dim")))
    if(i == 1):
        l = l.get("layer").get("config")
        file.write(str(l.get("name"))+" "+str(l.get("units"))+" "+str(l.get("activation")))
    elif(i > 1):
        file.write(str(l.get("name"))+" "+str(l.get("units"))+" "+str(l.get("activation")))
file.close()
```

Figure 5-11 Functionality of the NNConfig script, used to retrieve the layout of the current Neural Network used by the system

5.4 Article Database

The Article Database was designed in accordance to the Entity-Relationship Diagrams outlined in the conceptual design of this project (4.2.3) Using the SQLite DB Browser program. All foreign keys were configured to perform a cascaded-delete using the ON DELETE syntax.



```
Article          CREATE TABLE "Article" ( "articleID" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE, "articleName" TEXT NOT NULL, "articleDate" TEXT, "articleContents" TE
  articleID  INTEGER   "articleID" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE
  article... TEXT      "articleName" TEXT NOT NULL
  article... TEXT      "articleDate" TEXT
  articleC... TEXT     "articleContents" TEXT NOT NULL
  conVal...  INTEGER   "conValPred" INTEGER NOT NULL
  labValP... INTEGER   "labValPred" INTEGER NOT NULL
  partyID    INTEGER   "partyID" INTEGER NOT NULL
ArticleAuth...   CREATE TABLE "ArticleAuthor" ( "ArticleID" INTEGER NOT NULL, "AuthorID" INTEGER NOT NULL, PRIMARY KEY("ArticleID","AuthorID"), FOREIGN KEY("ArticleID") REFEREN
  ArticleID  INTEGER   "ArticleID" INTEGER NOT NULL
  AuthorID   INTEGER   "AuthorID" INTEGER NOT NULL
Author           CREATE TABLE "Author" ( "authorID" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE, "lastName" TEXT NOT NULL, "foreName" TEXT NOT NULL )
  authorID   INTEGER   "authorID" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE
  lastNa...  TEXT      "lastName" TEXT NOT NULL
  foreNa...  TEXT      "foreName" TEXT NOT NULL
Party            CREATE TABLE "Party" ( "partyID" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE, "partyName" TEXT )
  partyID    INTEGER   "partyID" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE
  partyN...  TEXT      "partyName" TEXT
sqlite_sequ...   CREATE TABLE sqlite_sequence(name,seq)
  name       TEXT      "name" TEXT
  seq        TEXT      "seq" TEXT
```

Figure 5-12 Final DataBase Configuration, Developed using SQLite DbBrowser

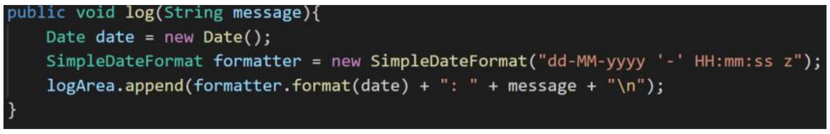
5.5 Politika: Main Program

5.5.1 UI Implementation

As the forms constructed in 4.4.1 were fully implementable, no more work was needed to develop the interactive section of the UI. Furthermore, by using the Form-Designer, the skeletons of the classes to implement were automatically constructed.

5.5.2 Log

To fulfil Requirement 2.4, the interface required a method to log information.



```java
public void log(String message){
    Date date = new Date();
    SimpleDateFormat formatter = new SimpleDateFormat("dd-MM-yyyy '-' HH:mm:ss z");
    logArea.append(formatter.format(date) + ": " + message + "\n");
}
```

Figure 5-13 Log Method Used in Politika

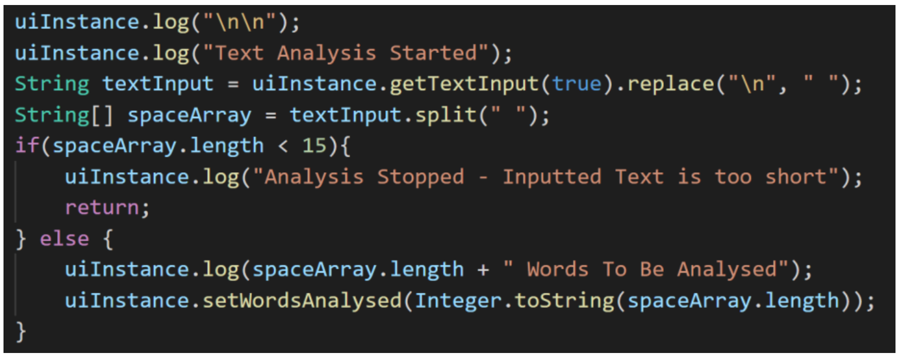
The log method takes the desired message as an input and outputs it to the logging text-area, appended with a data and time of its execution.

5.5.3 Text Analysis

5.5.3.1 Analyse Button Listener

The main entry point for the bias analysis process is the Analyse button, linked to an implemented Action Listener.

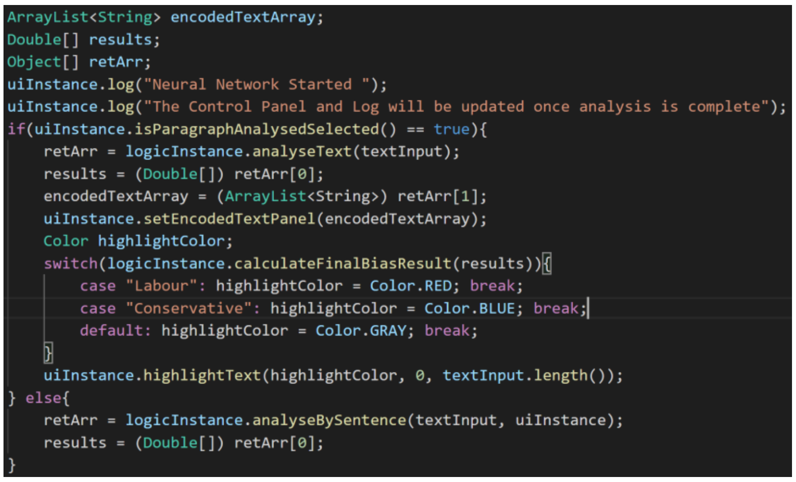
The method starts by logging the initiation of the bias analysis process. The text input is then retrieved from the input and any line-breaks (which could compromise the program when reading the jwrite file) are removed. The number of words is calculated by counting the number of spaces in the text, which is used to assert that the input is greater than the minimum length (15) and set the words-analysed label in the UI.



```
uiInstance.log("\n\n");
uiInstance.log("Text Analysis Started");
String textInput = uiInstance.getTextInput(true).replace("\n", " ");
String[] spaceArray = textInput.split(" ");
if(spaceArray.length < 15){
    uiInstance.log("Analysis Stopped - Inputted Text is too short");
    return;
} else {
    uiInstance.log(spaceArray.length + " Words To Be Analysed");
    uiInstance.setWordsAnalysed(Integer.toString(spaceArray.length));
}
```

Figure 5-14 Validation used in the ActionPerformed Method of the Analyse Button Listener

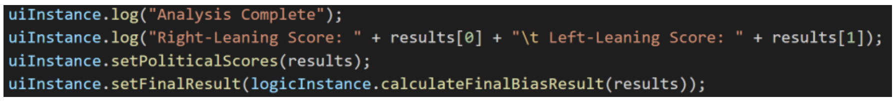
After the input has been fully prepared and validated, a new thread is started to run the rest of the method; ensuring the process doesn’t freeze the rest of the program while it is run. The method then declares variables to hold the results of the program, including an Array-List of strings to hold the encoded text and an array of doubles to hold the results of the bias analysis. The text is then analysed, either by passage or sentence, based on the selection of the radio-button in the UI.



```java
ArrayList<String> encodedTextArray;
Double[] results;
Object[] retArr;
uiInstance.log("Neural Network Started ");
uiInstance.log("The Control Panel and Log will be updated once analysis is complete");
if(uiInstance.isParagraphAnalysedSelected() == true){
    retArr = logicInstance.analyseText(textInput);
    results = (Double[]) retArr[0];
    encodedTextArray = (ArrayList<String>) retArr[1];
    uiInstance.setEncodedTextPanel(encodedTextArray);
    Color highlightColor;
    switch(logicInstance.calculateFinalBiasResult(results)){
        case "Labour": highlightColor = Color.RED; break;
        case "Conservative": highlightColor = Color.BLUE; break;
        default: highlightColor = Color.GRAY; break;
    }
    uiInstance.highlightText(highlightColor, 0, textInput.length());
} else{
    retArr = logicInstance.analyseBySentence(textInput, uiInstance);
    results = (Double[]) retArr[0];
}
```

Figure 5-15 Input Data Pre-Processing & Post-Processing in the Analyse Listener Action Listener

After processing, the analysis's completion is logged again and the results are past to the UI.

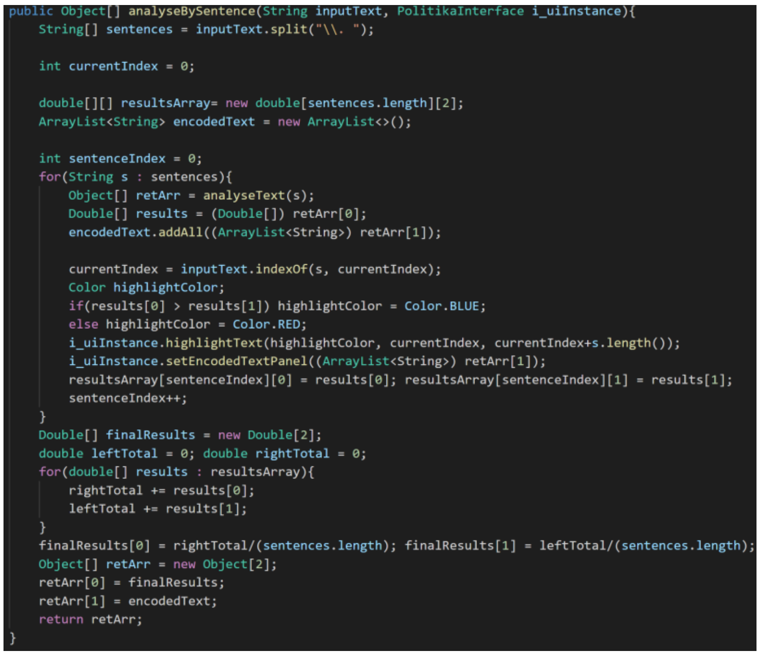


```java
uiInstance.log("Analysis Complete");
uiInstance.log("Right-Leaning Score: " + results[0] + "\t Left-Leaning Score: " + results[1]);
uiInstance.setPoliticalScores(results);
uiInstance.setFinalResult(logicInstance.calculateFinalBiasResult(results));
```

Figure 5-16 Logging Performed to display the user with the results of analysis

5.5.3.2 Analyse by Passage

The method starts by retrieving the text input and writing it to the jwrite file (discussed in 5.3). The method then executes the predictor script from a runtime instance, waiting until the script has finished executing. The pwrite file is then opened and read, assigning the encoded text and model results to respective variables. The output data is then further translated into an object array that is returned by the method. The text in the input area is then highlighted reflecting the results of the analysis.



```java
public Object[] analyseBySentence(String inputText, PolitikaInterface i_uiInstance){
    String[] sentences = inputText.split("\\. ");

    int currentIndex = 0;

    double[][] resultsArray= new double[sentences.length][2];
    ArrayList<String> encodedText = new ArrayList<>();

    int sentenceIndex = 0;
    for(String s : sentences){
        Object[] retArr = analyseText(s);
        Double[] results = (Double[]) retArr[0];
        encodedText.addAll((ArrayList<String>) retArr[1]);

        currentIndex = inputText.indexOf(s, currentIndex);
        Color highlightColor;
        if(results[0] > results[1]) highlightColor = Color.BLUE;
        else highlightColor = Color.RED;
        i_uiInstance.highlightText(highlightColor, currentIndex, currentIndex+s.length());
        i_uiInstance.setEncodedTextPanel((ArrayList<String>) retArr[1]);
        resultsArray[sentenceIndex][0] = results[0]; resultsArray[sentenceIndex][1] = results[1];
        sentenceIndex++;
    }
    Double[] finalResults = new Double[2];
    double leftTotal = 0; double rightTotal = 0;
    for(double[] results : resultsArray){
        rightTotal += results[0];
        leftTotal += results[1];
    }
    finalResults[0] = rightTotal/(sentences.length); finalResults[1] = leftTotal/(sentences.length);
    Object[] retArr = new Object[2];
    retArr[0] = finalResults;
    retArr[1] = encodedText;
    return retArr;
}
```

Figure 5-17 Method used to analyse by sentence

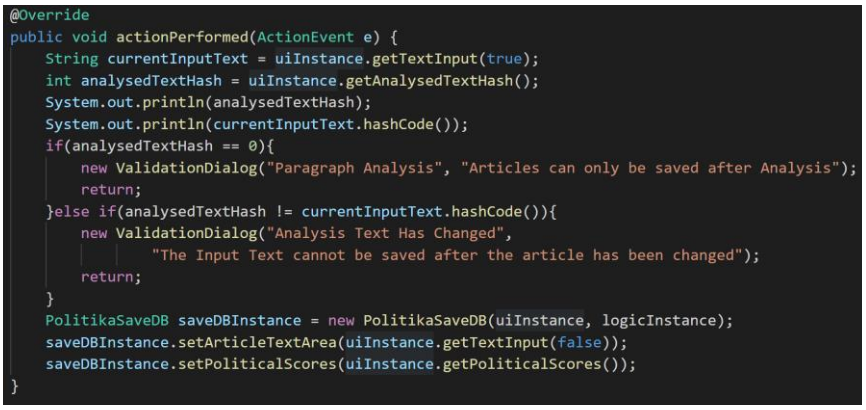
5.5.3.3 Analyse by Sentence

This method works very similarly to the passage method, with slight differences. The input text is split into an array by its sentences and sequentially processed using the method from 5.5.3.2, the results of which are appended to an array. The encoded text is also appended to the encoded-text-area. Once all of the sentences have been processed, the average of the sentence results are computed, on both the left and right nodes, and returned alongside the encoded text array. Each sentence is also highlighted in its own respective colour.

5.5.4 Saving Results to the Article Database

The save feature implements two Listeners similar to the one implemented in 5.5.3.1. The first listener is linked to the save button on the analysis panel, which validates the necessary validation has been performed, and opens up the Save form, copying over the input text and article results.

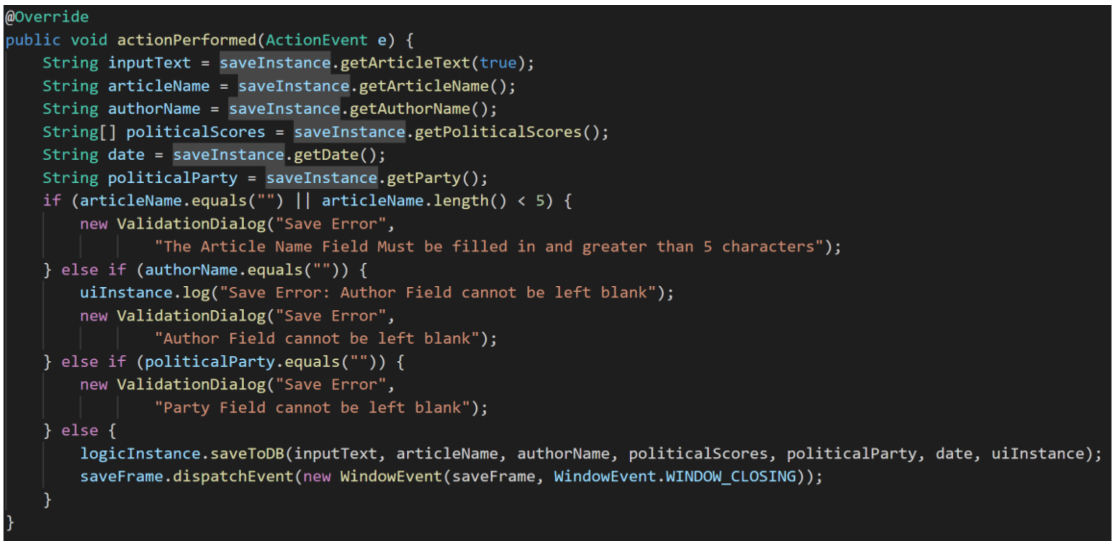
To detect whether the article text had been changed since it was last analysed, the String class's hash-code method was called. A hash of the input data is saved whenever the analysis is successful, and the two values are compared, allowing the integrity of the article to be fully validated (Requirement 3.2).



```java
@Override
public void actionPerformed(ActionEvent e) {
    String currentInputText = uiInstance.getTextInput(true);
    int analysedTextHash = uiInstance.getAnalysedTextHash();
    System.out.println(analysedTextHash);
    System.out.println(currentInputText.hashCode());
    if(analysedTextHash == 0){
        new ValidationDialog("Paragraph Analysis", "Articles can only be saved after Analysis");
        return;
    }else if(analysedTextHash != currentInputText.hashCode()){
        new ValidationDialog("Analysis Text Has Changed",
                "The Input Text cannot be saved after the article has been changed");
        return;
    }
    PolitikaSaveDB saveDBInstance = new PolitikaSaveDB(uiInstance, logicInstance);
    saveDBInstance.setArticleTextArea(uiInstance.getTextInput(false));
    saveDBInstance.setPoliticalScores(uiInstance.getPoliticalScores());
}
```

Figure 5-18 Action Method Called by the Save Button listener

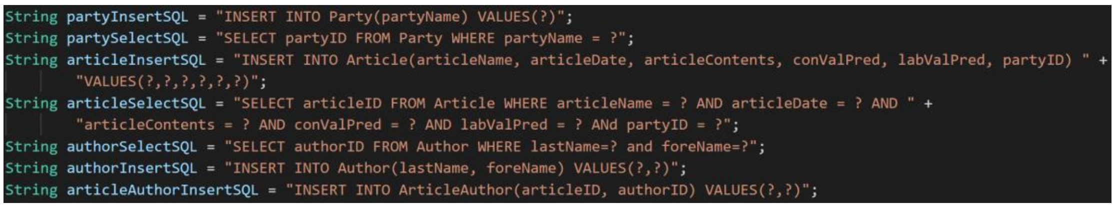
The second listener is linked to the save to database button on the save form and retrieves all of the necessary inputs from the form; passing them to the saveToDB method.



```java
@Override
public void actionPerformed(ActionEvent e) {
    String inputText = saveInstance.getArticleText(true);
    String articleName = saveInstance.getArticleName();
    String authorName = saveInstance.getAuthorName();
    String[] politicalScores = saveInstance.getPoliticalScores();
    String date = saveInstance.getDate();
    String politicalParty = saveInstance.getParty();
    if (articleName.equals("") || articleName.length() < 5) {
        new ValidationDialog("Save Error",
                "The Article Name Field Must be filled in and greater than 5 characters");
    } else if (authorName.equals("")) {
        uiInstance.log("Save Error: Author Field cannot be left blank");
        new ValidationDialog("Save Error",
                "Author Field cannot be left blank");
    } else if (politicalParty.equals("")) {
        new ValidationDialog("Save Error",
                "Party Field cannot be left blank");
    } else {
        logicInstance.saveToDB(inputText, articleName, authorName, politicalScores, politicalParty, date, uiInstance);
        saveFrame.dispatchEvent(new WindowEvent(saveFrame, WindowEvent.WINDOW_CLOSING));
    }
}
```

Figure 5-19 Action method for the Save To Database Button, with validation for the user's input.

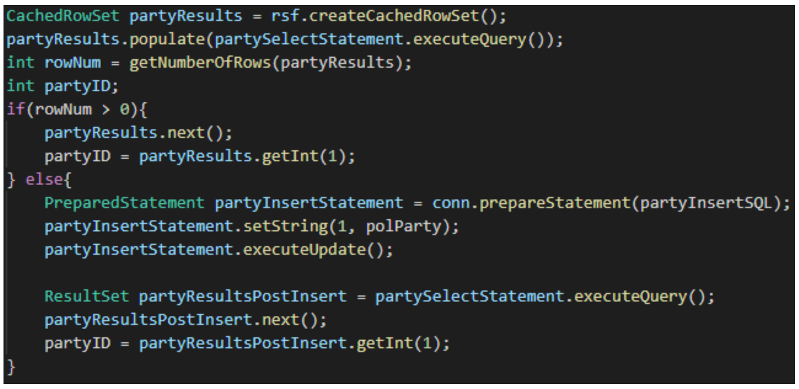
The method uses several prepared statements to securely query the database. First, a connection is established to the article database.



```java
String partyInsertSQL = "INSERT INTO Party(partyName) VALUES(?)";
String partySelectSQL = "SELECT partyID FROM Party WHERE partyName = ?";
String articleInsertSQL = "INSERT INTO Article(articleName, articleDate, articleContents, conValPred, labValPred, partyID) " +
        "VALUES(?,?,?,?,?,?)";
String articleSelectSQL = "SELECT articleID FROM Article WHERE articleName = ? AND articleDate = ? AND " +
        "articleContents = ? AND conValPred = ? AND labValPred = ? ANd partyID = ?";
String authorSelectSQL = "SELECT authorID FROM Author WHERE lastName=? and foreName=?";
String authorInsertSQL = "INSERT INTO Author(lastName, foreName) VALUES(?,?)";
String articleAuthorInsertSQL = "INSERT INTO ArticleAuthor(articleID, authorID) VALUES(?,?)";
```

Figure 5-20 Prepared Statements used to save articles to the Article Database

The method then executes a select SQL statement for any parties with names similar to the input of the save form. If the query returns a result, its partyID is stored. Otherwise, the party is new to the program and is inserted into the database.



```java
CachedRowSet partyResults = rsf.createCachedRowSet();
partyResults.populate(partySelectStatement.executeQuery());
int rowNum = getNumberOfRows(partyResults);
int partyID;
if(rowNum > 0){
    partyResults.next();
    partyID = partyResults.getInt(1);
} else{
    PreparedStatement partyInsertStatement = conn.prepareStatement(partyInsertSQL);
    partyInsertStatement.setString(1, polParty);
    partyInsertStatement.executeUpdate();

    ResultSet partyResultsPostInsert = partySelectStatement.executeQuery();
    partyResultsPostInsert.next();
    partyID = partyResultsPostInsert.getInt(1);
}
```

Figure 5-21 INSERT Statement with Validation

The article insert statement is prepared and executed using the inputted data, another query is then executed to retrieve the articleID of the inserted article record.

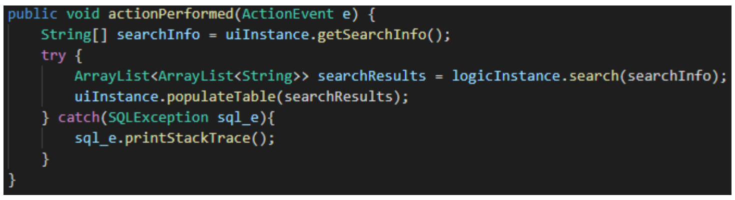
The author input is split into an array for each name (semicolons, in case multiple names are used) which is then iterated through. For each name, another split is performed by comma so that the surname and forenames are separated. The data is then used to construct a select statement to assert whether the author already exists in the table. If so the authorID is retrieved, otherwise, the author details are inserted into the database and the authorID is retrieved again.

Another operation is performed on the ArticleAuthor link-table. For each authorID obtained in the previous step, a database insert is performed alongside the articleID retrieved earlier; ensuring that all of the authors are linked to the inserted article. This concludes all of the necessary operations to save the article into the database.

5.5.5 Article Search

5.5.5.1 Search Function

The entry point of the article search function is located in the search button Listener, similarly to the previous listener functions. The listener retrieves both the search input and the search term (which denote which part of the article the search input refers to, chosen using a drop-down menu).



```
public void actionPerformed(ActionEvent e) {
    String[] searchInfo = uiInstance.getSearchInfo();
    try {
        ArrayList<ArrayList<String>> searchResults = logicInstance.search(searchInfo);
        uiInstance.populateTable(searchResults);
    } catch(SQLException sql_e){
        sql_e.printStackTrace();
    }
}
```

Figure 5-22 Action Method for the Search Button Listener

The listener calls the search method in the logic instance, which performs different operations depending on the search term. For all search terms other than the author, the retrieveArticleResults method is used, which performs a select statement on the Article table, linking it to the appropriate authors through the retrieveAuthors method.

The retrieveAuthorResults method works similarly, but retrieves the appropriate articleIDs through an inner-joined select statement and then proceeds to retrieve the appropriate articles.

These methods all result in the populateTable method, which iteratively adds all of the results to the search table in the user-interface.

5.5.5.2 Article Report

The article report functionality is activated by double clicking any record in the search table. A listener function retrieves the necessary data and passes it to the ArticleReport UI class, which implements the form created in the logical design, filling it with the necessary information on the selected record.

6 Evaluation

6.1 Neural Network Performance

Neural Network Configuration	Average Validation Accuracy
1 Hidden Layer	0.7039
2 Hidden Layers	0.7339
3 Hidden Layers	0.7092
4 Hidden Layers	0.7087
5 Hidden Layers	0.7074
1 Hidden Layer & 2-Stacked LSTM Layers	0.7123

Neural Network Configuration	Average Test Accuracy
1 Hidden Layer	0.7234
2 Hidden Layers	0.7195
3 Hidden Layers	0.7056
4 Hidden Layers	0.7116
5 Hidden Layers	0.7023
1 Hidden Layer & 2-Stacked LSTM Layers	0.701

The data above was collected throughout the exploration of the network domain, based on the average accuracy of the highest performing validation step (when the model was saved) and the accuracy of the testing stage after the model's training. The average of both of these datasets of results were then collated into a composite accuracy table, shown below:

Neural Network Configuration	Average Composite Accuracy	Ranking
1 Hidden Layer	0.71365	2
2 Hidden Layers	0.7267	1
3 Hidden Layers	0.7074	4
4 Hidden Layers	0.71015	3
5 Hidden Layers	0.70485	6
1 Hidden Layer & 2-Stacked LSTM Layers	0.70665	5

As the results tables show, there was very little change in accuracy throughout the experimentation, ranging from around 0.704 to 0.727. However, it appears that the best performance was achieved by using two dense hidden-layer, producing a model that was around 73% accurate on the test data.

Interestingly, though the average validation accuracy of this configuration was high, the average test accuracy was outperformed by the *one hidden-layer configuration*

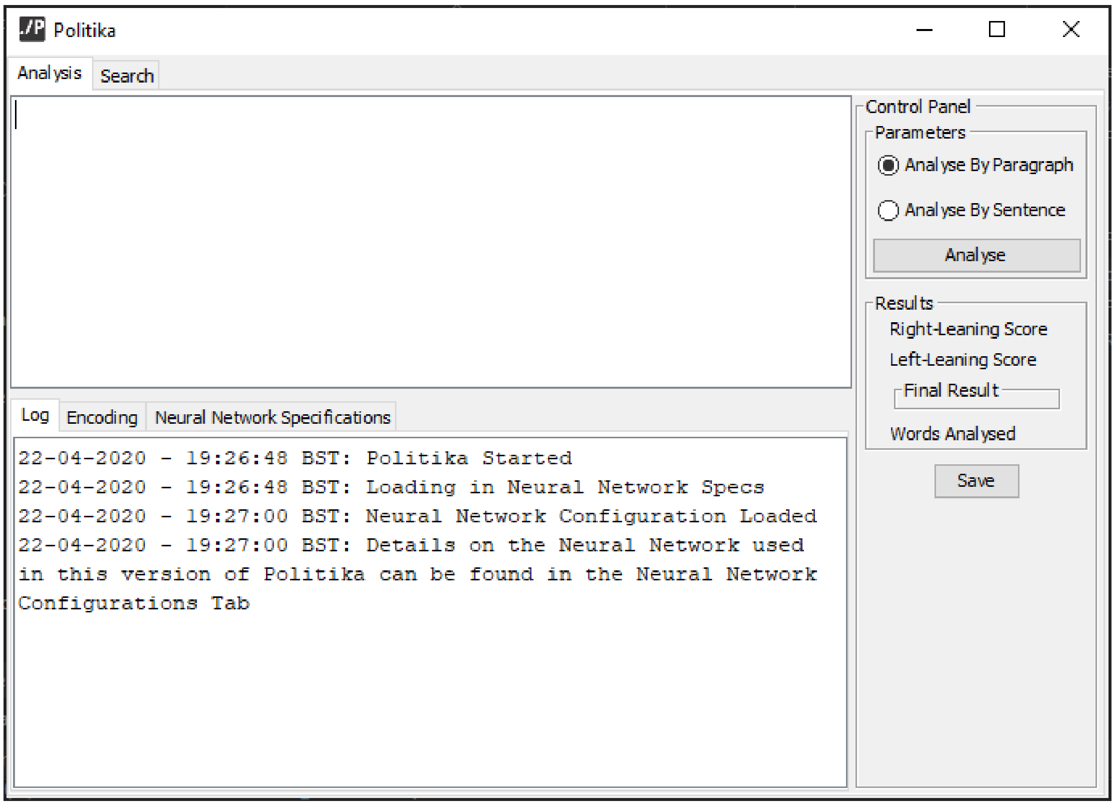
(0.7234 against 0.7195), which suggests that using an even smaller number of hidden-layers could produce better results. However, it is clear to show that for this problem, a more compact network is key to more optimal results.

Interestingly enough, increasing the complexity of the recursive section of the network by stacking an extra LSTM layer did not achieve better results at similar parameter settings, but however did increase the time to train greatly.

6.2 User-Interface Feature Testing

6.2.1 Message Logging

The logging method is triggered several times, including at the start of the program. This means that viewing the success of its implementation can be seen instantly:

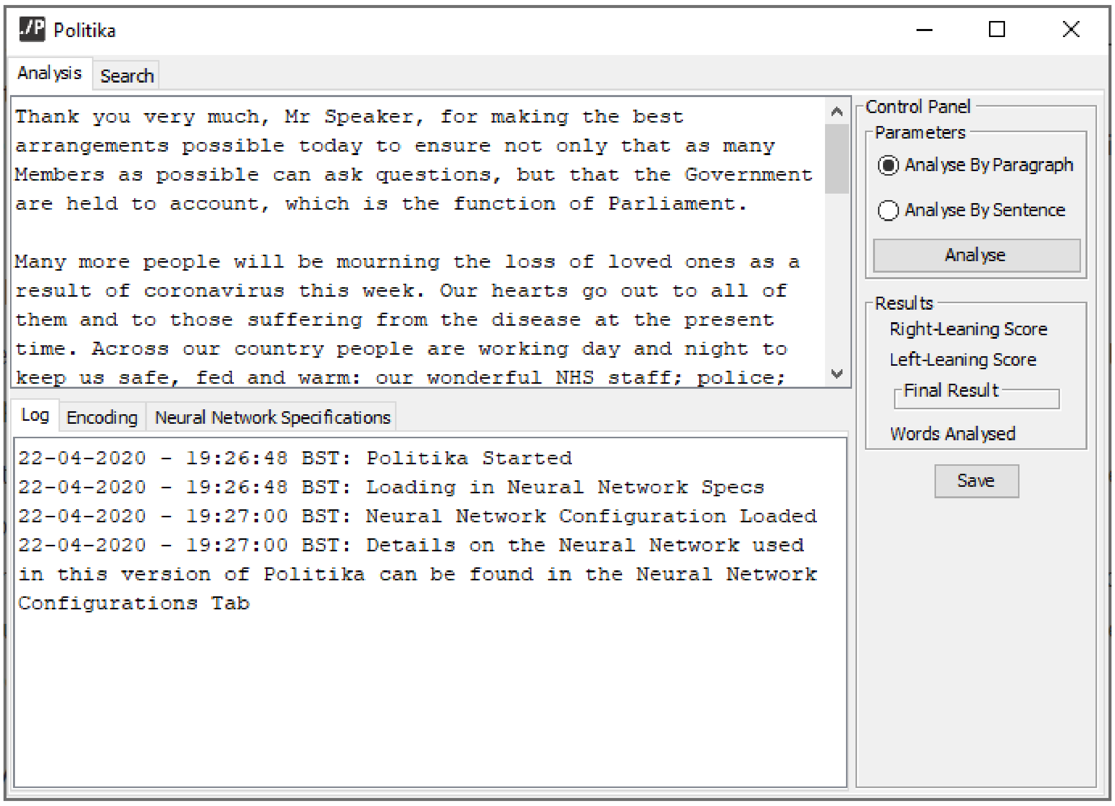


```
22-04-2020 - 19:26:48 BST: Politika Started
22-04-2020 - 19:26:48 BST: Loading in Neural Network Specs
22-04-2020 - 19:27:00 BST: Neural Network Configuration Loaded
22-04-2020 - 19:27:00 BST: Details on the Neural Network used
in this version of Politika can be found in the Neural Network
Configurations Tab
```

As the screenshot shows, as soon as the program is started, the log panel is written to, proving that the logging feature has been implemented successfully, proving Requirement 2.4 has been achieved..

6.2.2 Text Analysis

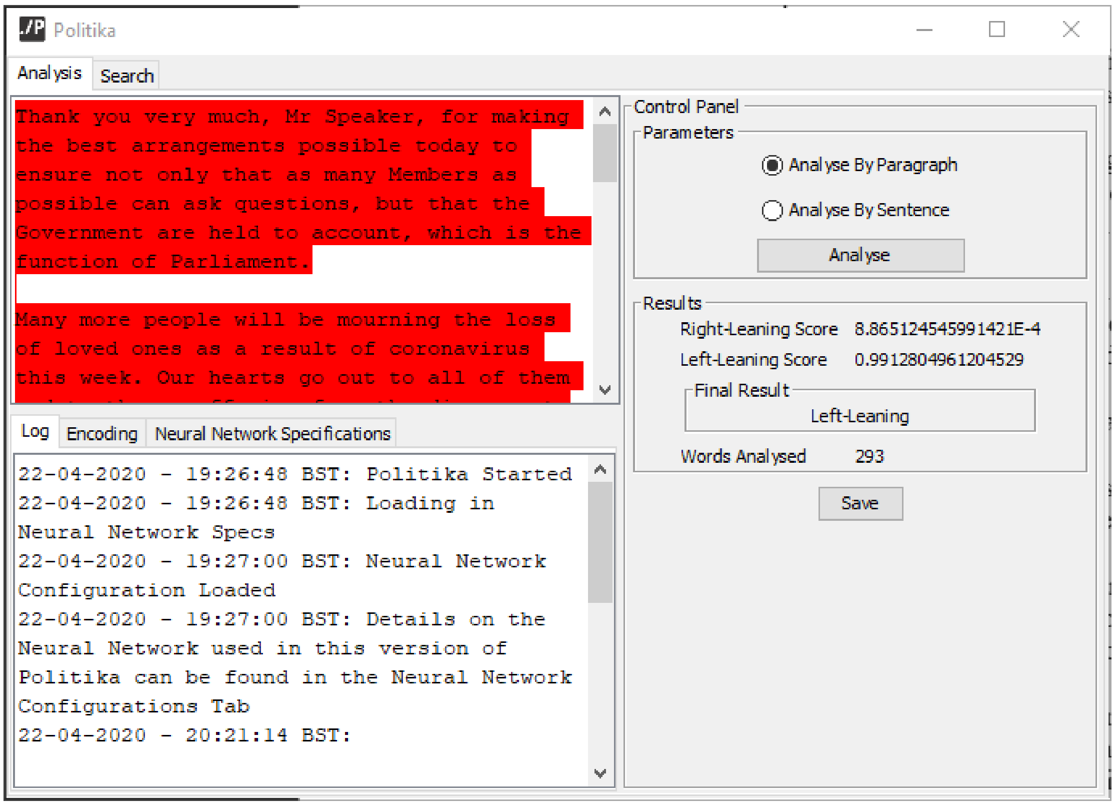
The program should be capable of taking in an input and displaying an appropriate output, breaking down the analysis that has taken place. Below is a screenshot demonstrating the input of the program, with the analyse by paragraph setting on, before the analyse button has been clicked:



```
Thank you very much, Mr Speaker, for making the best
arrangements possible today to ensure not only that as many
Members as possible can ask questions, but that the Government
are held to account, which is the function of Parliament.

Many more people will be mourning the loss of loved ones as a
result of coronavirus this week. Our hearts go out to all of
them and to those suffering from the disease at the present
time. Across our country people are working day and night to
keep us safe, fed and warm: our wonderful NHS staff; police;
```

```
22-04-2020 - 19:26:48 BST: Politika Started
22-04-2020 - 19:26:48 BST: Loading in Neural Network Specs
22-04-2020 - 19:27:00 BST: Neural Network Configuration Loaded
22-04-2020 - 19:27:00 BST: Details on the Neural Network used
in this version of Politika can be found in the Neural Network
Configurations Tab
```

Once the analysis button was activated, the log was updated and after a few seconds the results section of the control panel was updated. The text was also highlighted red, denoting the bias of the text:



Thank you very much, Mr Speaker, for making the best arrangements possible today to ensure not only that as many Members as possible can ask questions, but that the Government are held to account, which is the function of Parliament.

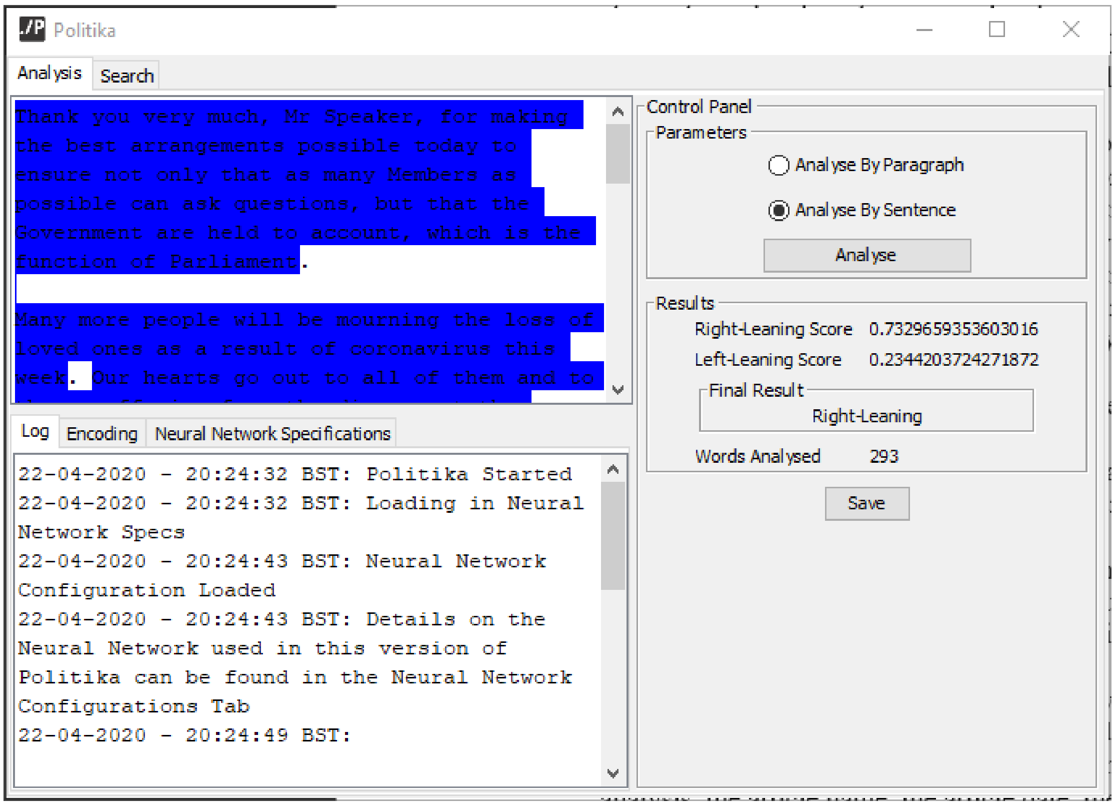
Many more people will be mourning the loss of loved ones as a result of coronavirus this week. Our hearts go out to all of them

Right-Leaning Score	8.865124545991421E-4
Left-Leaning Score	0.9912804961204529
Final Result	Left-Leaning
Words Analysed	293

```
22-04-2020 - 19:26:48 BST: Politika Started
22-04-2020 - 19:26:48 BST: Loading in
Neural Network Specs
22-04-2020 - 19:27:00 BST: Neural Network
Configuration Loaded
22-04-2020 - 19:27:00 BST: Details on the
Neural Network used in this version of
Politika can be found in the Neural Network
Configurations Tab
22-04-2020 - 20:21:14 BST:
```

6.2.3 Article Saving

The saving feature is activated by pressing the save button after text analysis has taken place. Once the save button is pressed, Politika validates that the article text is has been analysed specifically by passage, if it is not then a dialog-box is displayed and stops the save panel from opening (achieving Requirement 3.1):



Thank you very much, Mr Speaker, for making the best arrangements possible today to ensure not only that as many Members as possible can ask questions, but that the Government are held to account, which is the function of Parliament.

Many more people will be mourning the loss of loved ones as a result of coronavirus this week. Our hearts go out to all of them and to

```
22-04-2020 - 20:24:32 BST: Politika Started
22-04-2020 - 20:24:32 BST: Loading in Neural
Network Specs
22-04-2020 - 20:24:43 BST: Neural Network
Configuration Loaded
22-04-2020 - 20:24:43 BST: Details on the
Neural Network used in this version of
Politika can be found in the Neural Network
Configurations Tab
22-04-2020 - 20:24:49 BST:
```

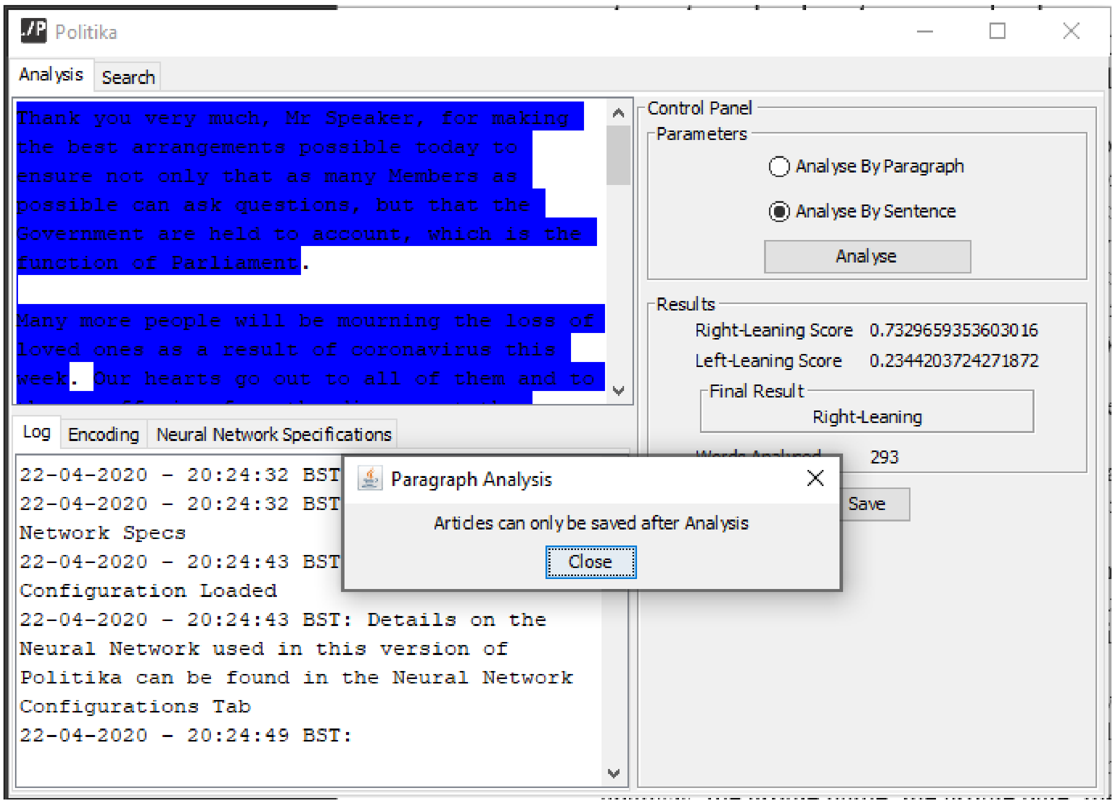
Control Panel

Parameters

( ) Analyse By Paragraph [✓ Analyse By Sentence]

Results

Right-Leaning Score	0.7329659353603016
Left-Leaning Score	0.2344203724271872
Final Result	Right-Leaning
Words Analysed	293



Thank you very much, Mr Speaker, for making the best arrangements possible today to ensure not only that as many Members as possible can ask questions, but that the Government are held to account, which is the function of Parliament.

Many more people will be mourning the loss of loved ones as a result of coronavirus this week. Our hearts go out to all of them and to

```
22-04-2020 - 20:24:32 BST
22-04-2020 - 20:24:32 BST
Network Specs
22-04-2020 - 20:24:43 BST
Configuration Loaded
22-04-2020 - 20:24:43 BST: Details on the
Neural Network used in this version of
Politika can be found in the Neural Network
Configurations Tab
22-04-2020 - 20:24:49 BST:
```

Control Panel

Parameters

( ) Analyse By Paragraph [✓ Analyse By Sentence]

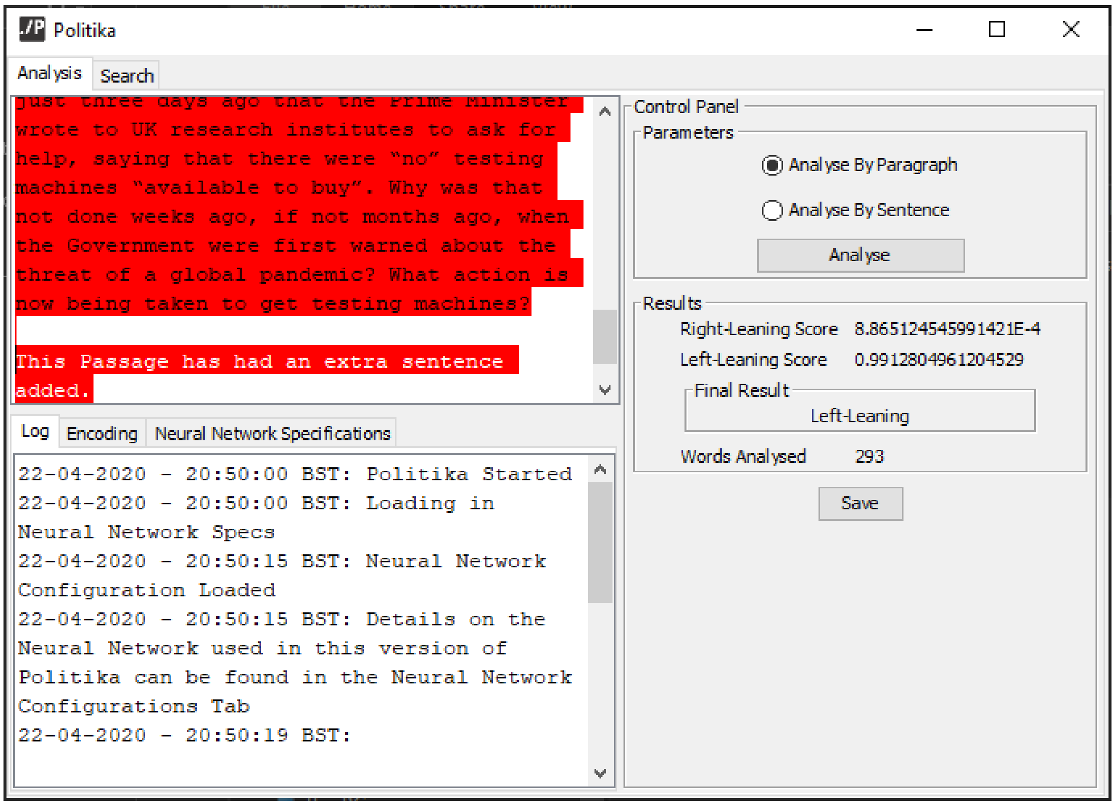
Results

Right-Leaning Score	0.7329659353603016
Left-Leaning Score	0.2344203724271872
Final Result	Right-Leaning
Words Analysed	293

Paragraph Analysis

Articles can only be saved after Analysis

Furthermore, if information is added or removed from the text before saving, a similar dialog will appear, fulfilling Requirement 3.2:



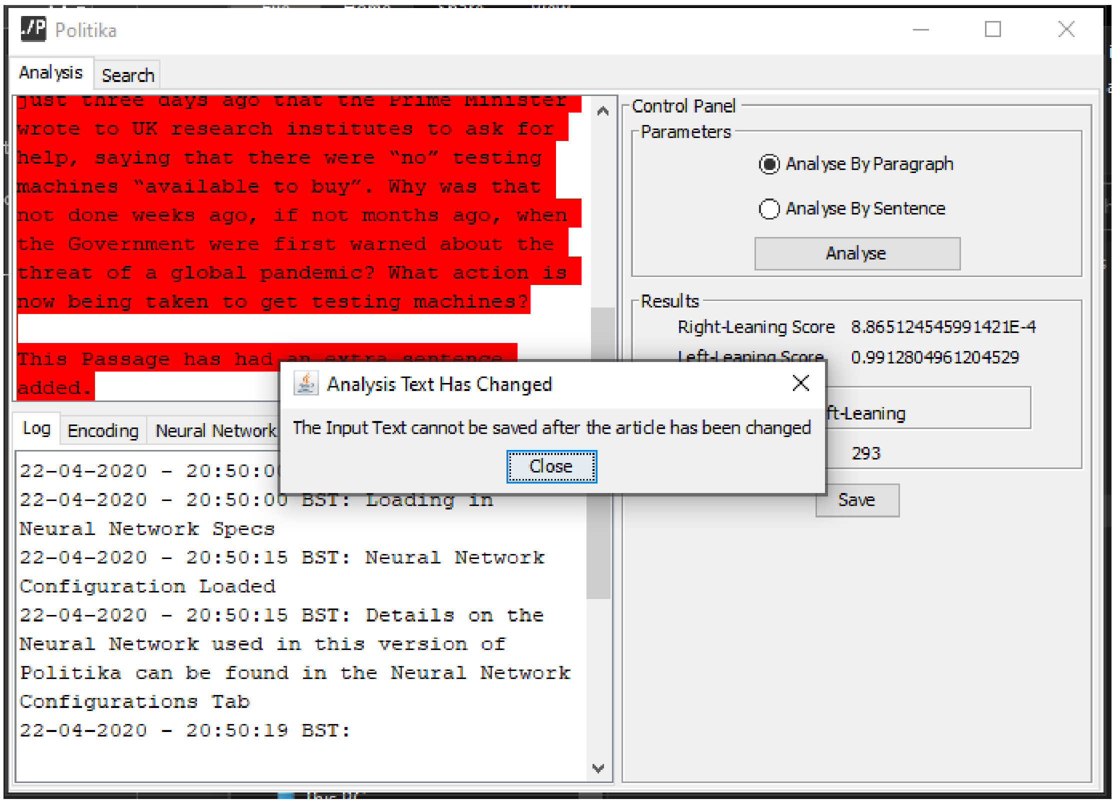
just three days ago that the Prime Minister wrote to UK research institutes to ask for help, saying that there were "no" testing machines "available to buy". Why was that not done weeks ago, if not months ago, when the Government were first warned about the threat of a global pandemic? What action is now being taken to get testing machines?

This Passage has had an extra sentence added.

(● Analyse By Paragraph) ○ Analyse By Sentence

Right-Leaning Score	8.865124545991421E-4
Left-Leaning Score	0.9912804961204529
Final Result	Left-Leaning
Words Analysed	293

```
22-04-2020 - 20:50:00 BST: Politika Started
22-04-2020 - 20:50:00 BST: Loading in
Neural Network Specs
22-04-2020 - 20:50:15 BST: Neural Network
Configuration Loaded
22-04-2020 - 20:50:15 BST: Details on the
Neural Network used in this version of
Politika can be found in the Neural Network
Configurations Tab
22-04-2020 - 20:50:19 BST:
```


Politika

```
just three days ago that the Prime Minister
wrote to UK research institutes to ask for
help, saying that there were "no" testing
machines "available to buy". Why was that
not done weeks ago, if not months ago, when
the Government were first warned about the
threat of a global pandemic? What action is
now being taken to get testing machines?

This Passage has had an extra sentence
added.
```

```
22-04-2020 - 20:50:0
22-04-2020 - 20:50:00 BST: Loading in
Neural Network Specs
22-04-2020 - 20:50:15 BST: Neural Network
Configuration Loaded
22-04-2020 - 20:50:15 BST: Details on the
Neural Network used in this version of
Politika can be found in the Neural Network
Configurations Tab
22-04-2020 - 20:50:19 BST:
```

Control Panel

Parameters

(●) Analyse By Paragraph

( ) Analyse By Sentence

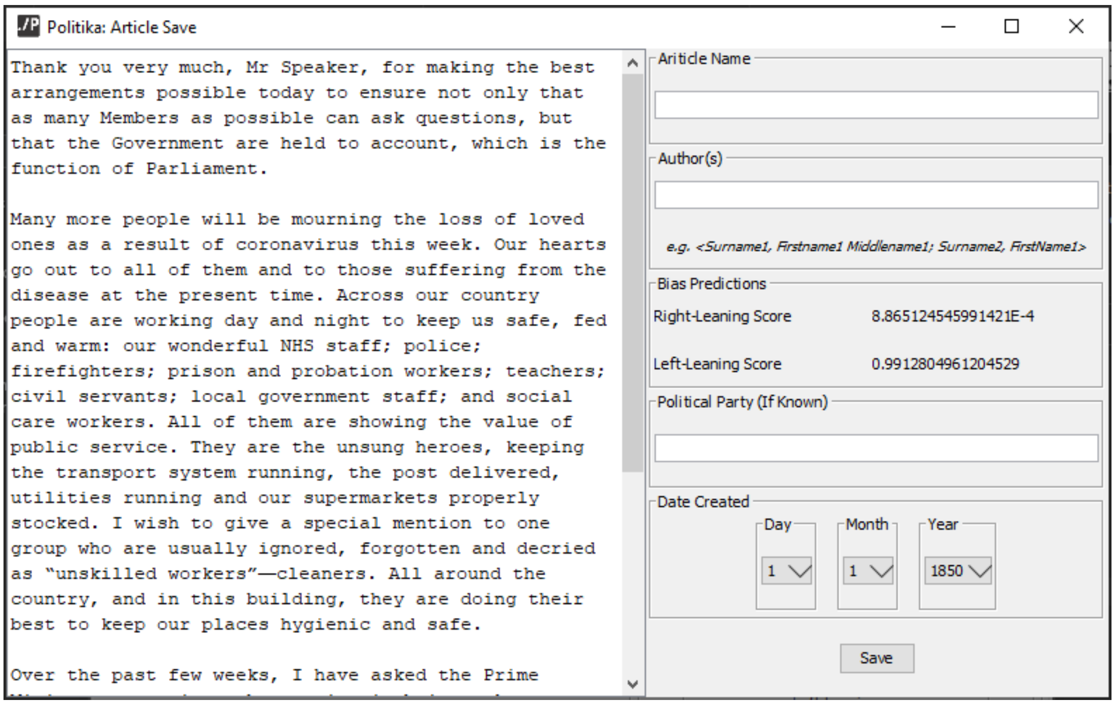
Results

Right-Leaning Score	8.865124545991421E-4
Left-Leaning Score	0.9912804961204529
	ft-Leaning
	293

Analysis Text Has Changed

The Input Text cannot be saved after the article has been changed

If both checks pass, the *Save Article* form is opened with the inputted text from the *Analysis Panel* present, fulfilling Requirement 3:


Politika: Article Save

```
Thank you very much, Mr Speaker, for making the best
arrangements possible today to ensure not only that
as many Members as possible can ask questions, but
that the Government are held to account, which is the
function of Parliament.

Many more people will be mourning the loss of loved
ones as a result of coronavirus this week. Our hearts
go out to all of them and to those suffering from the
disease at the present time. Across our country
people are working day and night to keep us safe, fed
and warm: our wonderful NHS staff; police;
firefighters; prison and probation workers; teachers;
civil servants; local government staff; and social
care workers. All of them are showing the value of
public service. They are the unsung heroes, keeping
the transport system running, the post delivered,
utilities running and our supermarkets properly
stocked. I wish to give a special mention to one
group who are usually ignored, forgotten and decried
as "unskilled workers"—cleaners. All around the
country, and in this building, they are doing their
best to keep our places hygienic and safe.

Over the past few weeks, I have asked the Prime
```

Article Name

Author(s)

e.g. <Surname1, Firstname1 Middlename1; Surname2, FirstName1>

Bias Predictions

Right-Leaning Score	8.865124545991421E-4
Left-Leaning Score	0.9912804961204529

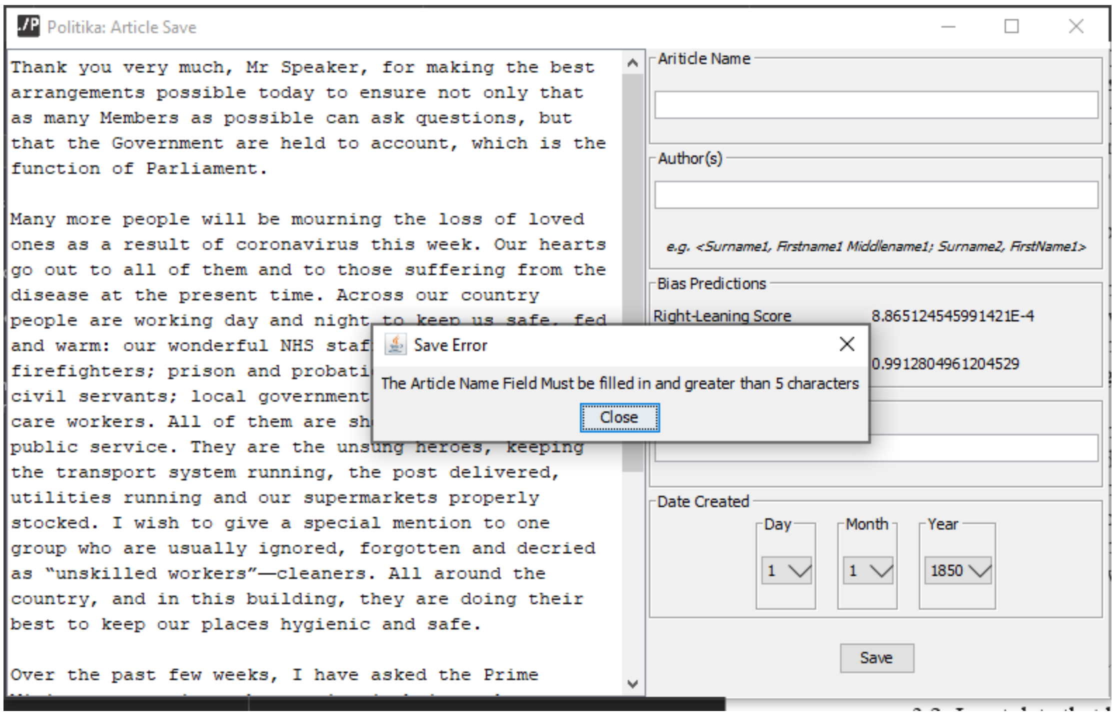
Political Party (If Known)

Date Created

Day	Month	Year
1	1	1850

When the save button on this panel is pressed, the program will go through several validation checks requested by Requirements 3.3-3.7:

6.2.3.1 Validation: Article Name Left Blank:



Politika: Article Save

```
Thank you very much, Mr Speaker, for making the best
arrangements possible today to ensure not only that
as many Members as possible can ask questions, but
that the Government are held to account, which is the
function of Parliament.

Many more people will be mourning the loss of loved
ones as a result of coronavirus this week. Our hearts
go out to all of them and to those suffering from the
disease at the present time. Across our country
people are working day and night to keep us safe, fed
and warm: our wonderful NHS staf
firefighters; prison and probati
civil servants; local government
care workers. All of them are sh
public service. They are the unsung heroes, keeping
the transport system running, the post delivered,
utilities running and our supermarkets properly
stocked. I wish to give a special mention to one
group who are usually ignored, forgotten and decried
as "unskilled workers"—cleaners. All around the
country, and in this building, they are doing their
best to keep our places hygienic and safe.

Over the past few weeks, I have asked the Prime
```

Article Name:

Author(s):

e.g. <Surname1, Firstname1 Middlename1; Surname2, FirstName1>

Bias Predictions

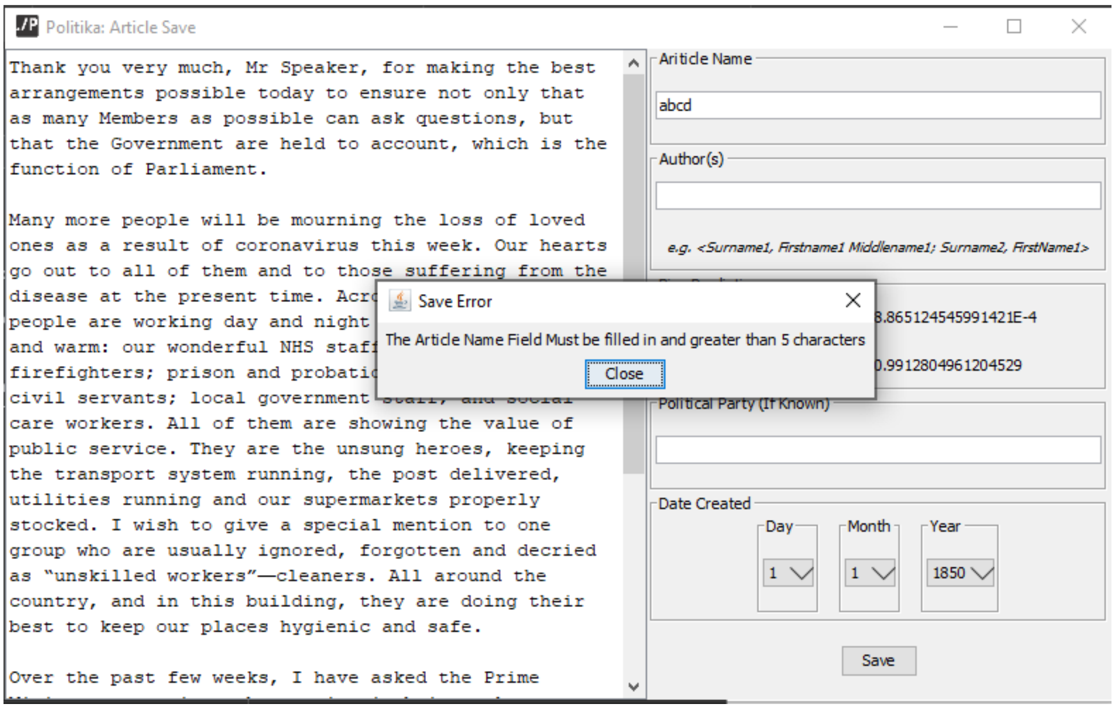
Right-Leaning Score	8.865124545991421E-4
	0.9912804961204529

Date Created — Day: 1, Month: 1, Year: 1850

Save Error

The Article Name Field Must be filled in and greater than 5 characters

6.2.3.2 Validation: Article Name (<5 Characters):



Politika: Article Save

```
Thank you very much, Mr Speaker, for making the best
arrangements possible today to ensure not only that
as many Members as possible can ask questions, but
that the Government are held to account, which is the
function of Parliament.

Many more people will be mourning the loss of loved
ones as a result of coronavirus this week. Our hearts
go out to all of them and to those suffering from the
disease at the present time. Acro
people are working day and night
and warm: our wonderful NHS staf
firefighters; prison and probati
civil servants; local government
care workers. All of them are showing the value of
public service. They are the unsung heroes, keeping
the transport system running, the post delivered,
utilities running and our supermarkets properly
stocked. I wish to give a special mention to one
group who are usually ignored, forgotten and decried
as "unskilled workers"—cleaners. All around the
country, and in this building, they are doing their
best to keep our places hygienic and safe.

Over the past few weeks, I have asked the Prime
```

Article Name: abcd

Author(s):

e.g. <Surname1, Firstname1 Middlename1; Surname2, FirstName1>

	8.865124545991421E-4
	0.9912804961204529

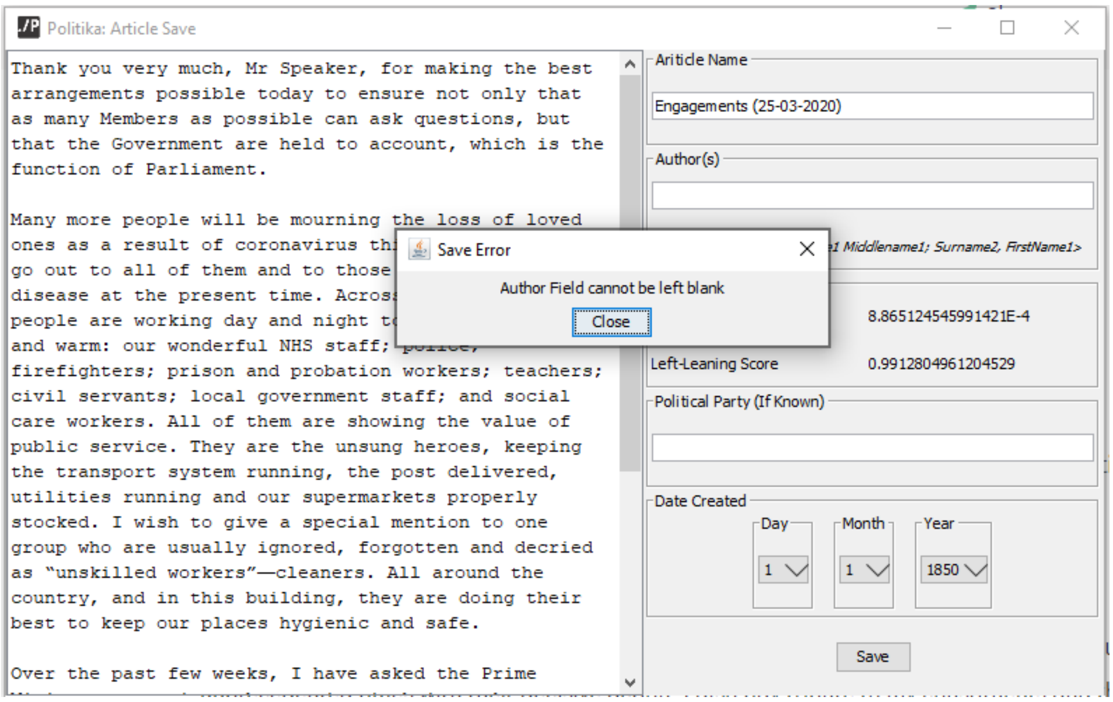
Political Party (If Known):

Date Created — Day: 1, Month: 1, Year: 1850

Save Error

The Article Name Field Must be filled in and greater than 5 characters

6.2.3.3 Validation: Author(s) Field Left Blank:



Politika: Article Save

```
Thank you very much, Mr Speaker, for making the best
arrangements possible today to ensure not only that
as many Members as possible can ask questions, but
that the Government are held to account, which is the
function of Parliament.

Many more people will be mourning the loss of loved
ones as a result of coronavirus thi
go out to all of them and to those
disease at the present time. Acros
people are working day and night t
and warm: our wonderful NHS staff;
firefighters; prison and probation workers; teachers;
civil servants; local government staff; and social
care workers. All of them are showing the value of
public service. They are the unsung heroes, keeping
the transport system running, the post delivered,
utilities running and our supermarkets properly
stocked. I wish to give a special mention to one
group who are usually ignored, forgotten and decried
as "unskilled workers"—cleaners. All around the
country, and in this building, they are doing their
best to keep our places hygienic and safe.

Over the past few weeks, I have asked the Prime
```

Article Name: Engagements (25-03-2020)

Author(s):

1 Middlename1; Surname2, FirstName1>

8.865124545991421E-4

Left-Leaning Score 0.9912804961204529

Political Party (If Known):

Date Created: Day 1, Month 1, Year 1850

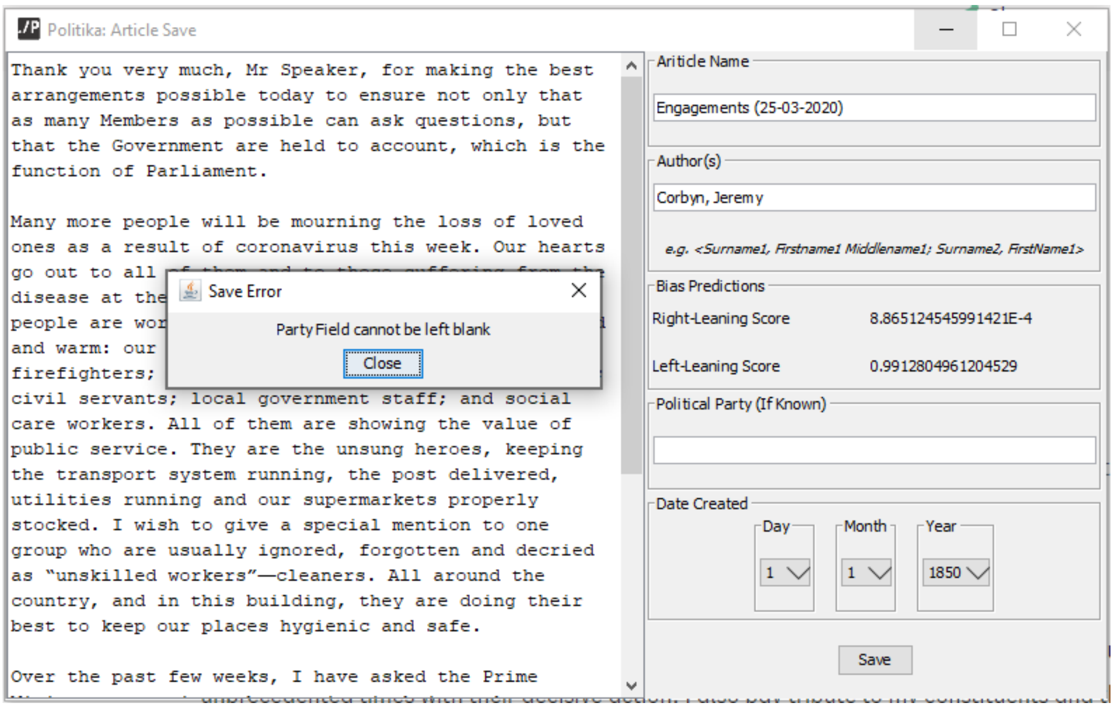
Save

Save Error

Author Field cannot be left blank

Close

6.2.3.4 Validation: Party Field Left Blank:



Politika: Article Save

```
Thank you very much, Mr Speaker, for making the best
arrangements possible today to ensure not only that
as many Members as possible can ask questions, but
that the Government are held to account, which is the
function of Parliament.

Many more people will be mourning the loss of loved
ones as a result of coronavirus this week. Our hearts
go out to all
disease at the
people are wor
and warm: our
firefighters;
civil servants; local government staff; and social
care workers. All of them are showing the value of
public service. They are the unsung heroes, keeping
the transport system running, the post delivered,
utilities running and our supermarkets properly
stocked. I wish to give a special mention to one
group who are usually ignored, forgotten and decried
as "unskilled workers"—cleaners. All around the
country, and in this building, they are doing their
best to keep our places hygienic and safe.

Over the past few weeks, I have asked the Prime
```

Article Name: Engagements (25-03-2020)

Author(s): Corbyn, Jeremy

e.g. <Surname1, Firstname1 Middlename1; Surname2, FirstName1>

Bias Predictions

Right-Leaning Score 8.865124545991421E-4

Left-Leaning Score 0.9912804961204529

Political Party (If Known):

Date Created: Day 1, Month 1, Year 1850

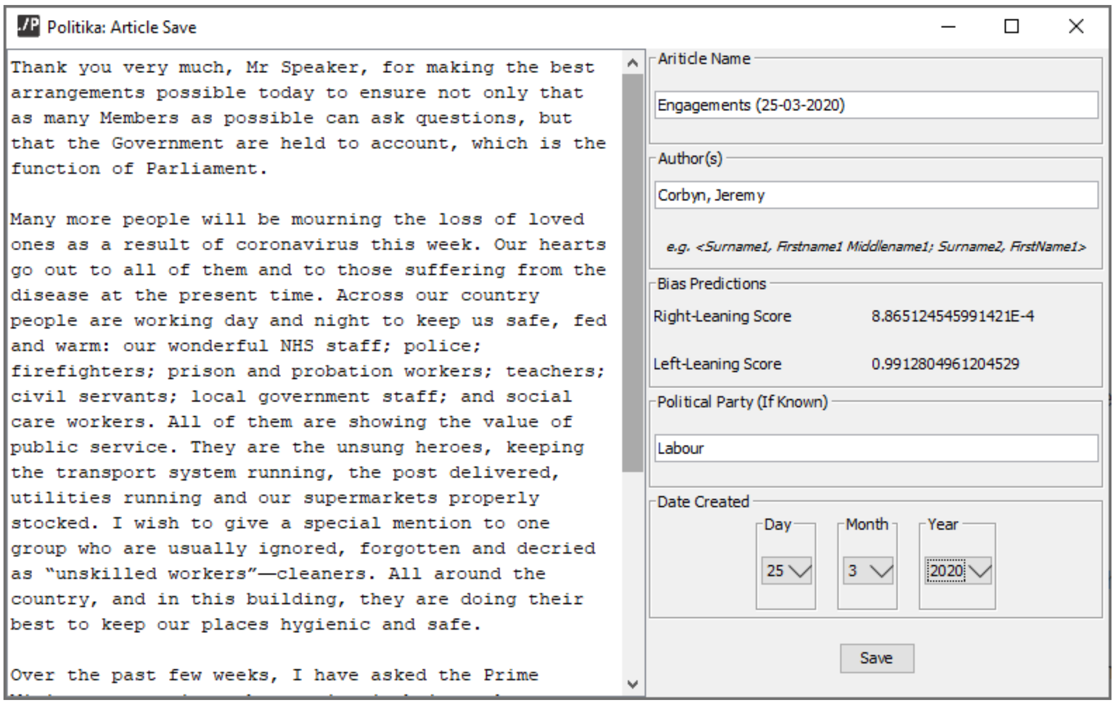
Save

Save Error

Party Field cannot be left blank

Close

If all information has been filled in and the save button has been pressed, the panel is closed and the article is written to the database:



```
Thank you very much, Mr Speaker, for making the best
arrangements possible today to ensure not only that
as many Members as possible can ask questions, but
that the Government are held to account, which is the
function of Parliament.

Many more people will be mourning the loss of loved
ones as a result of coronavirus this week. Our hearts
go out to all of them and to those suffering from the
disease at the present time. Across our country
people are working day and night to keep us safe, fed
and warm: our wonderful NHS staff; police;
firefighters; prison and probation workers; teachers;
civil servants; local government staff; and social
care workers. All of them are showing the value of
public service. They are the unsung heroes, keeping
the transport system running, the post delivered,
utilities running and our supermarkets properly
stocked. I wish to give a special mention to one
group who are usually ignored, forgotten and decried
as "unskilled workers"—cleaners. All around the
country, and in this building, they are doing their
best to keep our places hygienic and safe.

Over the past few weeks, I have asked the Prime
```

Article Name: Engagements (25-03-2020)

Author(s): Corbyn, Jeremy

e.g. <Surname1, Firstname1 Middlename1; Surname2, FirstName1>

Bias Predictions

Right-Leaning Score	8.865124545991421E-4
Left-Leaning Score	0.9912804961204529

Political Party (If Known): Labour

Date Created: Day 25, Month 3, Year 2020

The saved article can be found in the database:

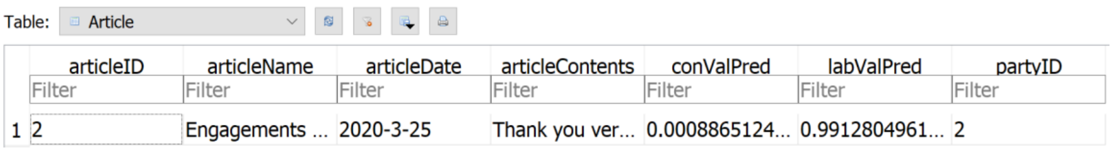


Table: Article

	articleID	articleName	articleDate	articleContents	conValPred	labValPred	partyID
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	2	Engagements ...	2020-3-25	Thank you ver...	0.0008865124...	0.9912804961...	2

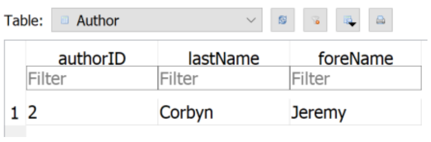


Table: Author

	authorID	lastName	foreName
	Filter	Filter	Filter
1	2	Corbyn	Jeremy

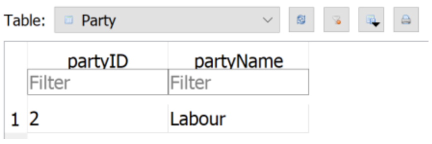
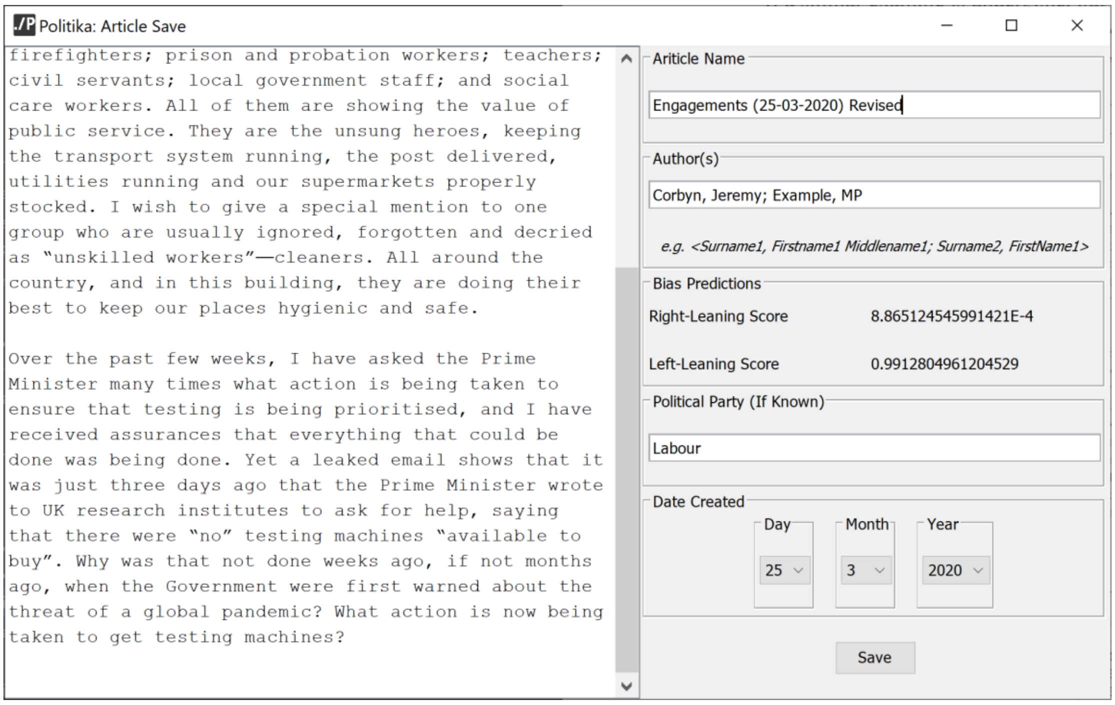


Table: Party

	partyID	partyName
	Filter	Filter
1	2	Labour

6.2.3.5 Validation: Multiple Authors

If a similar example is undergone, but specifying two authors in the recommended syntax (*LastName1*, *FirstName1*; *LastName2*, *FirstName2*), the database is updated to hold the extra author and the ArticleAuthor linking table is updated for both authors on one article.



```
firefighters; prison and probation workers; teachers;
civil servants; local government staff; and social
care workers. All of them are showing the value of
public service. They are the unsung heroes, keeping
the transport system running, the post delivered,
utilities running and our supermarkets properly
stocked. I wish to give a special mention to one
group who are usually ignored, forgotten and decried
as "unskilled workers"—cleaners. All around the
country, and in this building, they are doing their
best to keep our places hygienic and safe.

Over the past few weeks, I have asked the Prime
Minister many times what action is being taken to
ensure that testing is being prioritised, and I have
received assurances that everything that could be
done was being done. Yet a leaked email shows that it
was just three days ago that the Prime Minister wrote
to UK research institutes to ask for help, saying
that there were "no" testing machines "available to
buy". Why was that not done weeks ago, if not months
ago, when the Government were first warned about the
threat of a global pandemic? What action is now being
taken to get testing machines?
```

Ariticle Name: Engagements (25-03-2020) Revised

Author(s): Corbyn, Jeremy; Example, MP

e.g. <Surname1, Firstname1 Middlename1; Surname2, FirstName1>

Bias Predictions

Right-Leaning Score	8.865124545991421E-4
Left-Leaning Score	0.9912804961204529

Political Party (If Known): Labour

Date Created: Day 25, Month 3, Year 2020

The two new authors added can also be seen in the database:

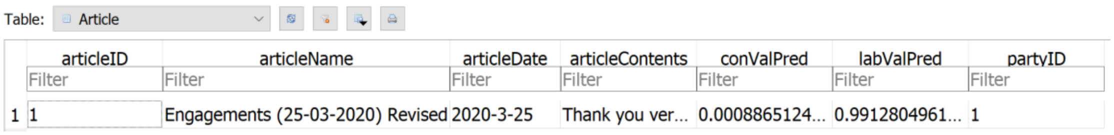
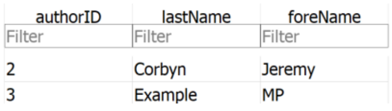


Table: Article

	articleID	articleName	articleDate	articleContents	conValPred	labValPred	partyID
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Engagements (25-03-2020) Revised	2020-3-25	Thank you ver...	0.0008865124...	0.9912804961...	1



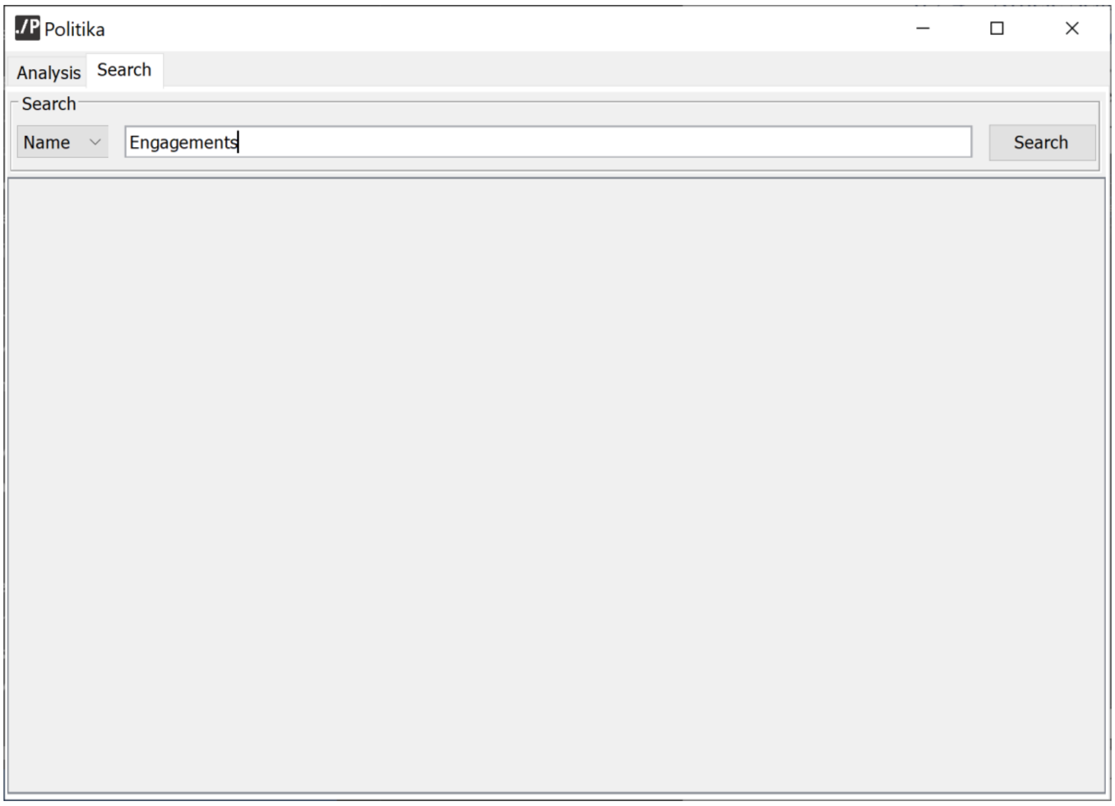
authorID	lastName	foreName
Filter	Filter	Filter
2	Corbyn	Jeremy
3	Example	MP

This concludes the testing for the Article Save section of the project, as the tests show all of Requirement 3.

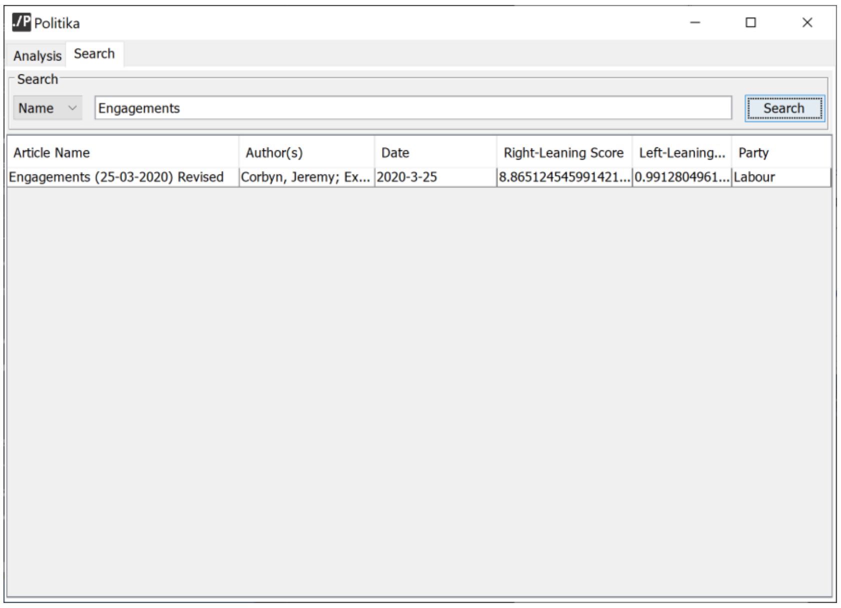
6.2.4 Article Search

6.2.4.1 Article Name Search

The name search is performed by selecting *Name* in the search term drop-down menu, inputting the desired search term and pressing the search button.



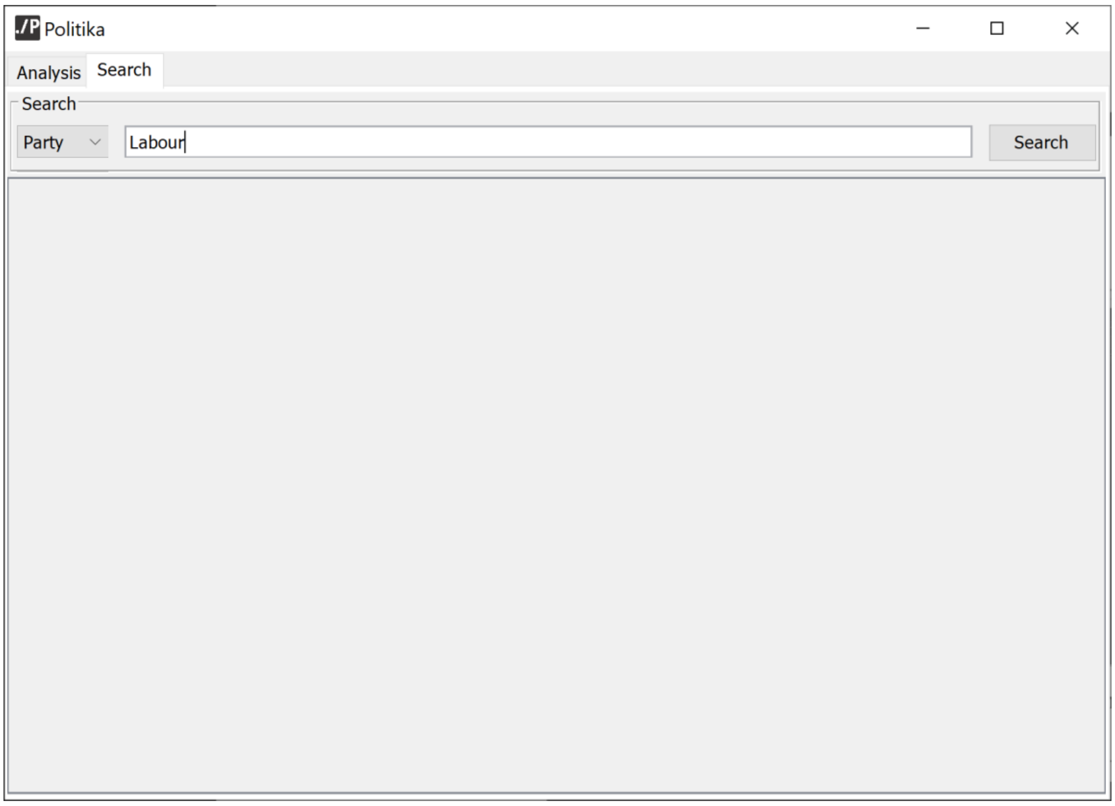
Once the Search button is pressed, all results with an Article Name that matches the input will be displayed in the results table below the search bar, :



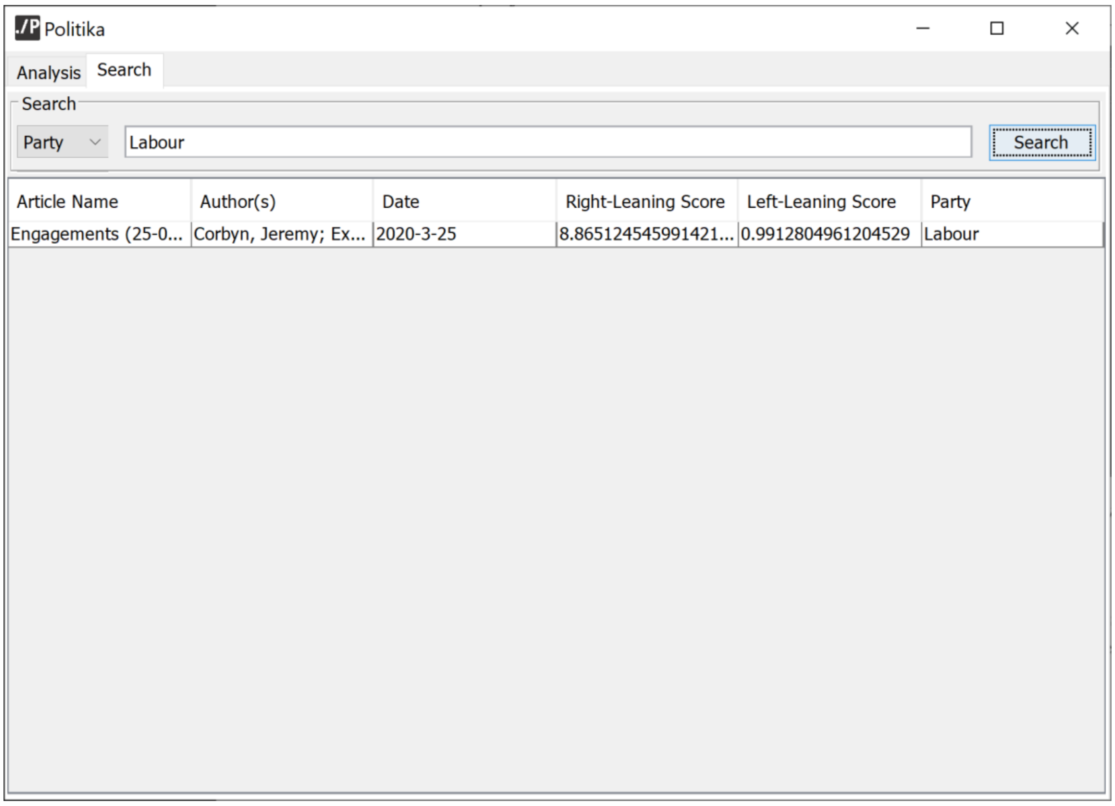
Article Name	Author(s)	Date	Right-Leaning Score	Left-Leaning...	Party
Engagements (25-03-2020) Revised	Corbyn, Jeremy; Ex...	2020-3-25	8.865124545991421...	0.9912804961...	Labour

6.2.4.2 Party Search

The party search is performed similarly to the article name search, except with *Party* selected as the search term in the drop-down menu:



Once the search button is clicked, all articles that have a matching Party label will be displayed in the results table:

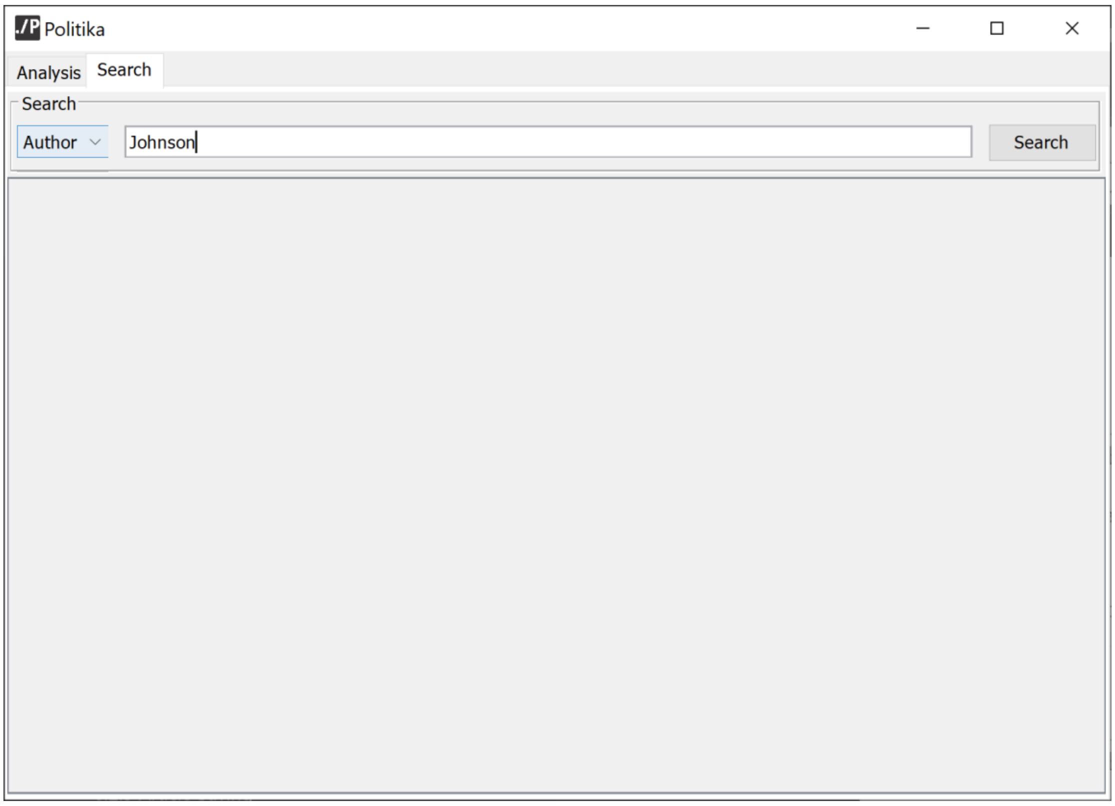


Article Name	Author(s)	Date	Right-Leaning Score	Left-Leaning Score	Party
Engagements (25-0...	Corbyn, Jeremy; Ex...	2020-3-25	8.865124545991421...	0.9912804961204529	Labour

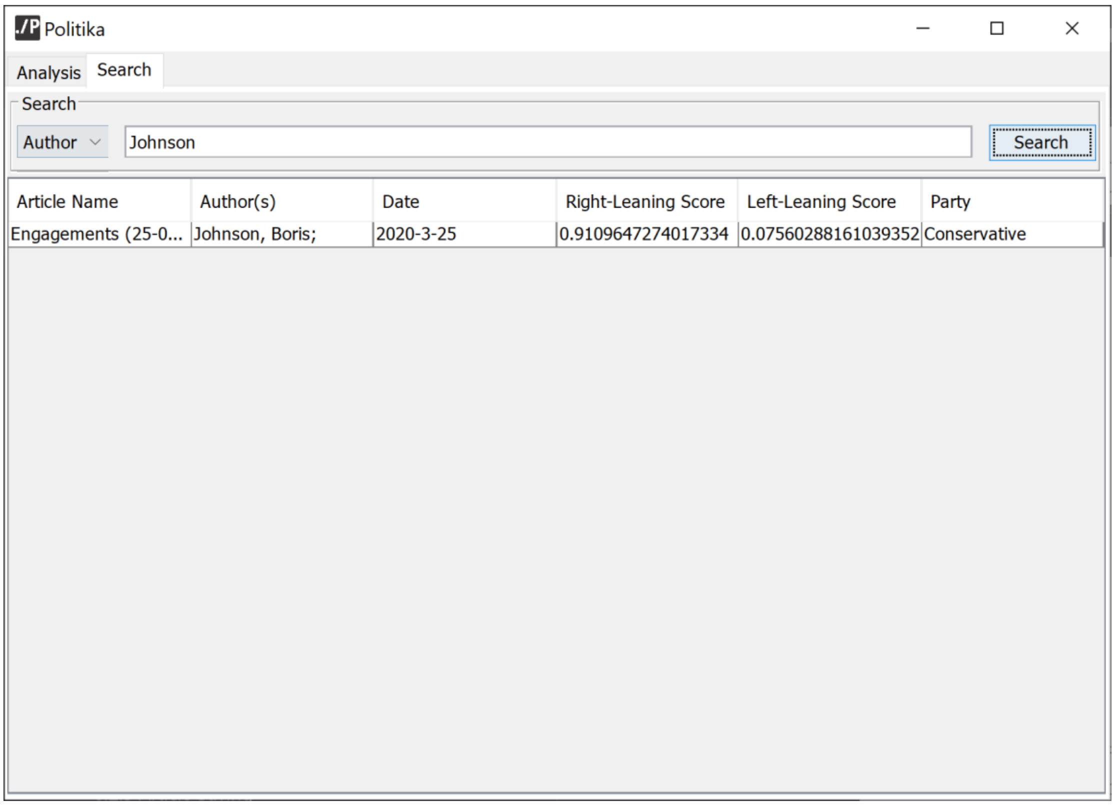
As this screenshot shows, if the article has multiple authors, they will all be displayed in the results table.

6.2.4.3 Authors Search

The authors search also works similarly to the previous examples. However, in order to search for authors effectively, the usual syntax will need to be used (last name, first name):



Once the search button is clicked, all articles with authors matching the input are displayed:

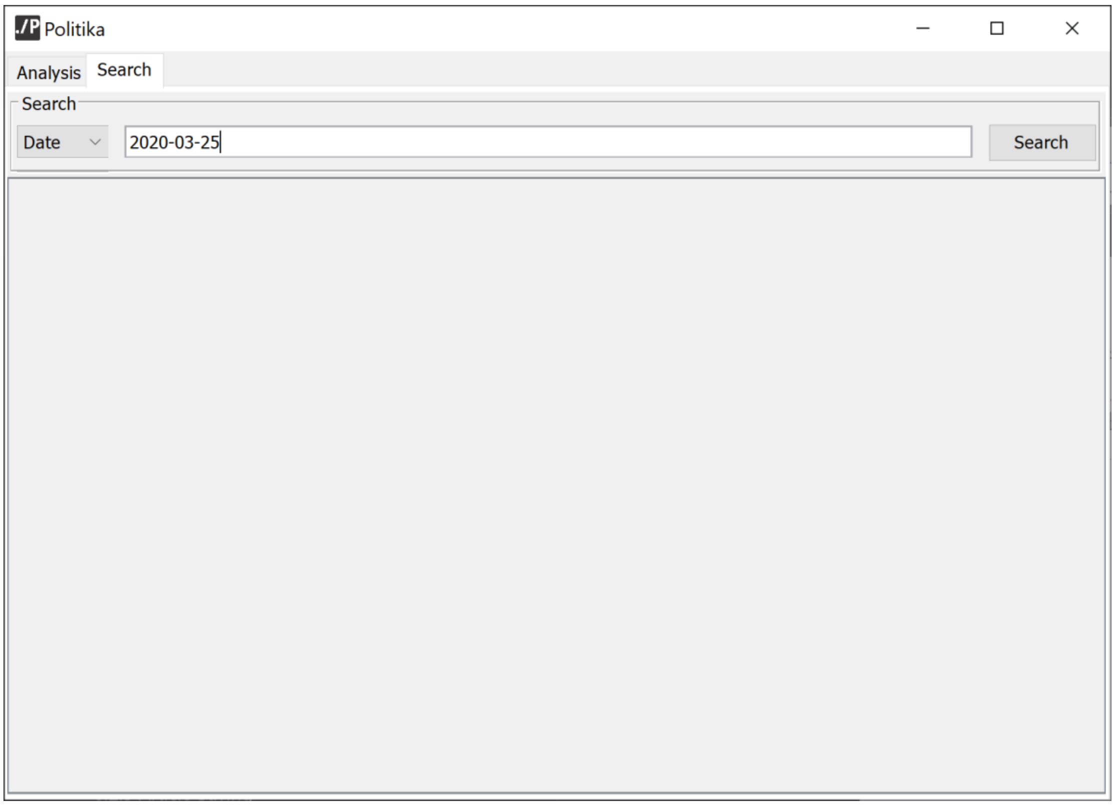


Johnson

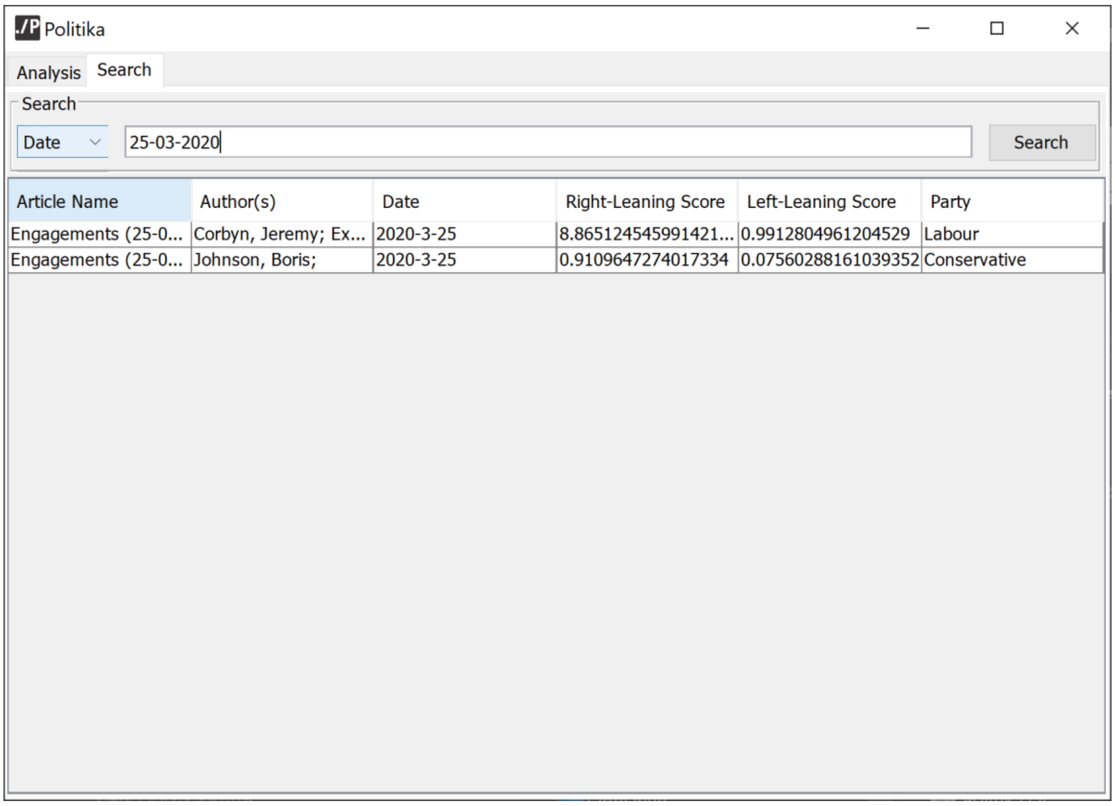
Article Name	Author(s)	Date	Right-Leaning Score	Left-Leaning Score	Party
Engagements (25-0...	Johnson, Boris;	2020-3-25	0.9109647274017334	0.07560288161039352	Conservative

6.2.4.4 Date Search

The data search can be achieved by selecting *Date* in the search term drop-down menu and inputting the selected date:



When the search button is clicked, all articles with a matching date to the input date are displayed:

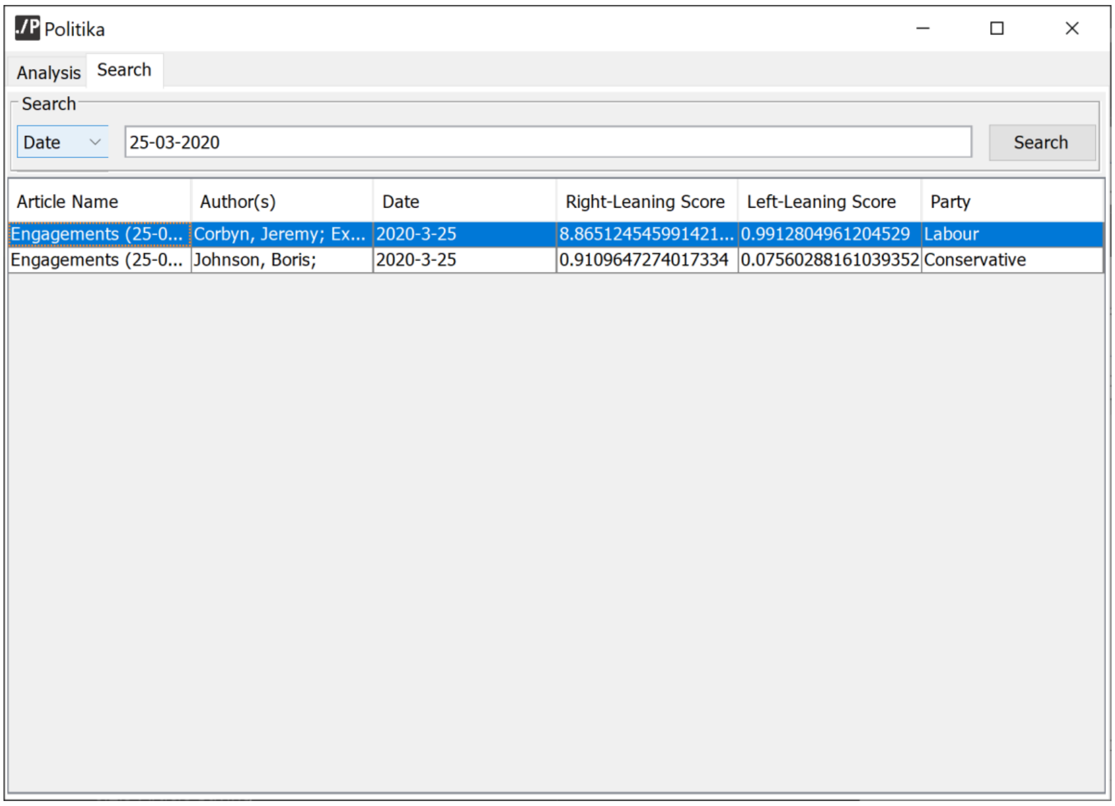


25-03-2020

Article Name	Author(s)	Date	Right-Leaning Score	Left-Leaning Score	Party
Engagements (25-0...	Corbyn, Jeremy; Ex...	2020-3-25	8.865124545991421...	0.9912804961204529	Labour
Engagements (25-0...	Johnson, Boris;	2020-3-25	0.9109647274017334	0.07560288161039352	Conservative

6.2.4.5 Article Reports

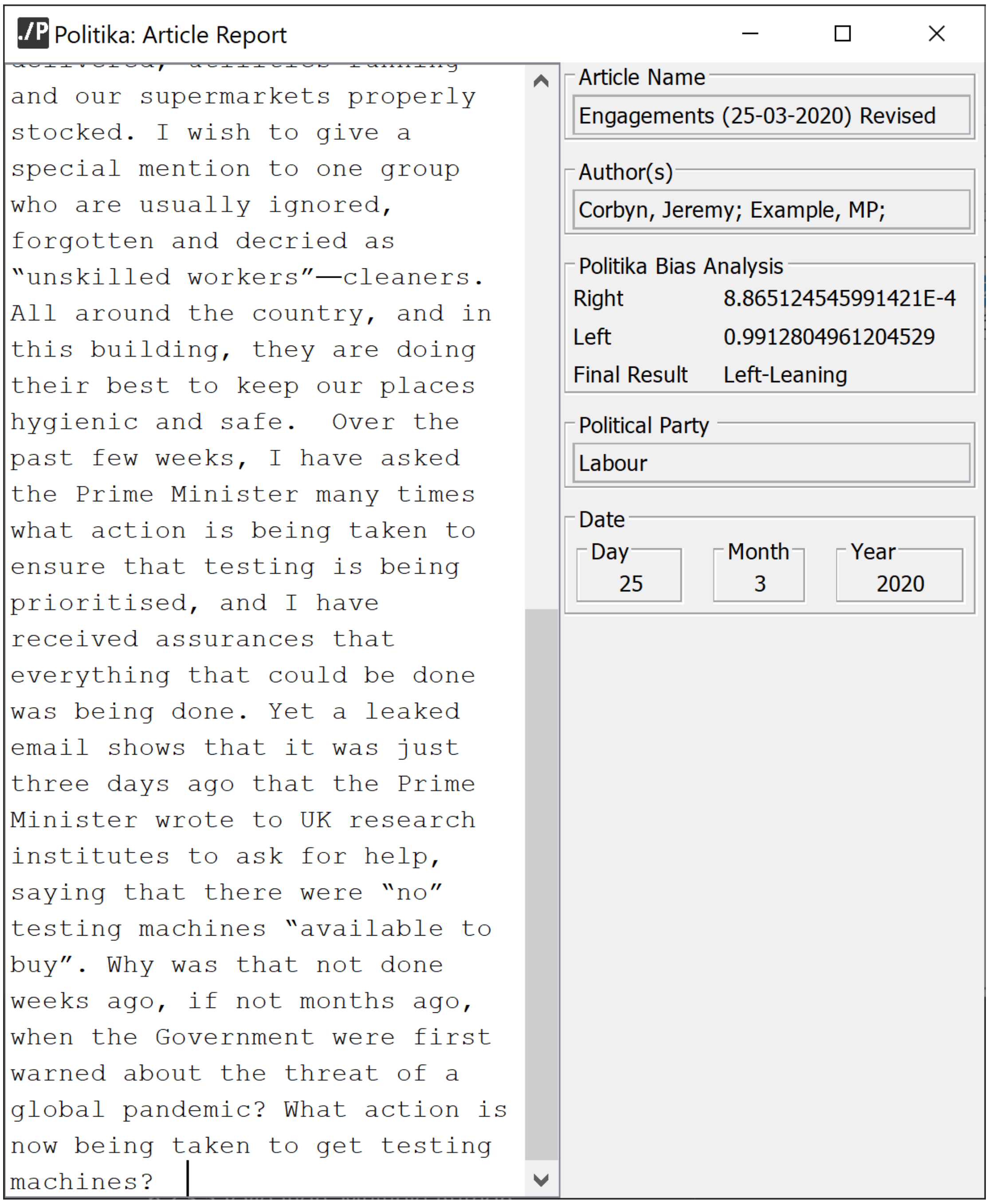
Article reports can be obtained by double clicking one of the articles in the search table:



25-03-2020

Article Name	Author(s)	Date	Right-Leaning Score	Left-Leaning Score	Party
Engagements (25-0...	Corbyn, Jeremy; Ex...	2020-3-25	8.865124545991421...	0.9912804961204529	Labour
Engagements (25-0...	Johnson, Boris;	2020-3-25	0.9109647274017334	0.07560288161039352	Conservative

Once the result has been double-clicked, an Article Report form is opened, displaying the relevant information for the selected article:


Politika: Article Report

and our supermarkets properly stocked. I wish to give a special mention to one group who are usually ignored, forgotten and decried as "unskilled workers"—cleaners. All around the country, and in this building, they are doing their best to keep our places hygienic and safe. Over the past few weeks, I have asked the Prime Minister many times what action is being taken to ensure that testing is being prioritised, and I have received assurances that everything that could be done was being done. Yet a leaked email shows that it was just three days ago that the Prime Minister wrote to UK research institutes to ask for help, saying that there were "no" testing machines "available to buy". Why was that not done weeks ago, if not months ago, when the Government were first warned about the threat of a global pandemic? What action is now being taken to get testing machines?

Article Name: Engagements (25-03-2020) Revised

Author(s): Corbyn, Jeremy; Example, MP;

Politika Bias Analysis

Right	8.865124545991421E-4
Left	0.9912804961204529
Final Result	Left-Leaning

Political Party: Labour

Date

Day	Month	Year
25	3	2020

6.3 Artifact Usability Evaluation

6.3.1 Introduction

For the artifact's evaluation, five members of the public were asked to test the finished program, guided by a set of exercises that spanned the full functionality of the program, and then complete a questionnaire outlining their experiences with it.

		Strongly Agree	Agree	Indifferent	Disagree	Strongly Disagree
		1	2	3	4	5
1	I found the application easy to navigate					
2	I felt confident using features of the application					
3	I felt the system's design was inconsistent					
4	Some features of the application were confusing to me					
5	I found the overall aesthetic of the program inviting (e.g. colour scheme and design)					
6	I was happy with the outputs of the application (e.g. were the results of the application organised in a way that you expected/approved of?).					
7	The information produced by this application is useful/valuable					
8	I would recommend this application to other users					
9	There have been times in my past where I would have found this application useful					
10	I would use this application again.					

Figure 6-1 The Usability survey presented to each participant

6.3.2 Results

Questions	1	2	3	4	5	Average
1	3	3	2	3	2	2.6
2	2	4	3	2	3	2.8
3	5	5	5	4	5	4.8
4	2	3	4	3	1	2.6
5	3	3	4	3	2	3
6	1	2	1	1	3	1.6
7	2	2	1	2	1	1.6
8	2	3	4	2	3	2.8
9	1	1	2	1	3	1.6
10	2	2	1	3	2	2

Figure 6-2 The results of each participant's survey, with the average response recorded.

As the results table above shows, the overall feedback to the usability of the program was very positive, but not without its criticisms.

Overall, people agreed that the application was clearly laid out and easy to navigate, and that they felt confident using all features of the program. As a

Arguably the most important point, most of the users agreed that the application was of great use and value to them, and would even consider using the program in the future and recommending it to a friend. This opinion is of great value to the project as, while the design of the system can be improved, the overall concept cannot. People's approval of this fact holds a strong inclination that the project's main aim has been achieved.

However, the majority of participants also agreed that some features of the application were confusing to them as well as the aesthetic of the program being fairly uninviting. These factors could be improved in later iterations of the project by implementing a different, more personalised Look-And-Feel to the interface and providing a more detailed guidance through the interface.

7 Conclusion

7.1 Requirements Attainment

Requirements	**Sections That Achieve Requirements**
1	5.2; 6.1
2	4.2.1.1; 4.4.1.1; 5.5.3; 6.2.2
3	4.4.1.3; 5.5.4; 6.2.3
4	4.2.3; 5.4
5	4.2.1.2; 4.4.1.2; 4.4.1.4; 6.2.4.4; 6.2.4.5
6	*By saving the models to a h5 file, it is possible to retrain the network and replace the model without incurring any issues with the Project.*

7.2 Reflection and Potential Improvements

In conclusion, the deliverables outlined in the Project definition were able to be successfully implemented to an effective standard.

An LSTM model was trained to detect political-bias in variable-length text to an accuracy of 72%. The model was trained on processed data from the Hansard Parliamentary debates archive from members of both the conservative and labour party, that was annotated at a passage level. Though the experiments around the neural network's configuration provided little fluctuation into its performance, later iterations of the system could be improved by increasing the depth of the dense hidden layers.

An interface was also successfully constructed to allow the general public to analyse their own texts and receive processed feedback, along with an implemented database to store their articles with supplementary information. Functionality was also added to the interface to facilitate the searching of these saved articles and view them in the form of a report. After testing the usability of the developed interface with a selection of participants, feedback was generally positive. The participants believed that the main concept of the program was of use to themselves and their peers and that the program was fairly easy to navigate and understand. However, the overall aesthetic and functionality of the program could be improved upon in greater iterations.

Though the deliverables outlined in the project were produced to a fairly acceptable standard, there is still a lot of room for improvement on their development. Because of the wide-scope of this project, in having to train a neural network and produce an interface to communicate with it, it was difficult to achieve a polished and exceptional final product in the short time-frame of this project. Given more time, more experiments could have been performed on the neural network to find an even more optimal configuration. Furthermore, a more in-depth usability enquiry could have been performed, to ensure that the delivered program would achieve would be more easily understood and used by its intended audience. These factors could be improved upon in later iterations of the project.

On a personal reflection, while the program did work as intended, its performance was slowed through the use of python scripts to perform the necessary text analysis. This performance-cap was mainly due to the import times for TensorFlow. By using a different method to perform this analysis, such as a server-client relationship (where the server has a constantly running python instance), it may be possible to reduce this time. This change would also benefit the security of the project, as the necessary model files would not need to be kept on user machines.

8 References

Abadi, M. et al., 2015. *TensorFlow: Large-scale machine learning on heterogeneous systems.* [Online]

Available at: www.tensorflow.org

[Accessed 19 March 2020].

Agile Alliance, n.d. *What is Agile Software Development?.* [Online]

Available at: https://www.agilealliance.org/agile101/

[Accessed 24 March 2020].

Bengio, Y., Simard, P. & Frasconi, P., 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions On Neural Networks,* March, 5(2), pp. 157-166.

bulenkov, 2017. *Darcula: Darcula Look and Feel.* [Online]

Available at: https://github.com/bulenkov/Darcula

[Accessed 15 March 2020].

fangyidong, 2014. *json-simple: A simple Java toolkit for JSON. You can use json-simple to encode or decode JSON text..* [Online]

Available at: https://github.com/fangyidong/json-simple

[Accessed 5 October 2019].

Gers, F., Schmidhuber, J. & Cummins, F., 2000. Learning to Forget: Continual Prediction with LSTM. *Neural Computation,* 12(10), p. 2451–2471.

Godoy, D., 2018. *Understanding binary cross-entropy / log loss: a visual explanation.* [Online]

Available at: https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

[Accessed 12 March 2020].

Goodfellow, I. & Bengio, Y. C. A., 2016. *Deep Learning.* Cambridge, Massachussetts: MIT Press.

Hochreiter, S. & Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation,* 9(8), pp. 1-32.

Irsoy, O. & Cardie, C., 2014. Deep Recursive Neural Networks for Compositionality in Language. *Advances in Neural Information Processing Systems,* Issue 27, pp. 1-9.

Iyyer, M., Enns, P., Boyd-Graber, J. & Resnik, P., 2014. *Political Ideology Detection Using Recursive Neural Networks,* Maryland: University of Maryland.

JetBrains, 2001. *IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains.* [Online]

Available at: https://www.jetbrains.com/idea/

[Accessed 1 October 2019].

Kingma, D. P. & Lei Ba, J., 2015. *Adam: A Method For Stochastic Optimization.* San Diego, ICLR.

Kulkarni, V., Ye, J., Skiena , S. & Wang, W. Y., 2018. *Multi-view Models for Political Ideology Detection of News Articles,* Brussels: Association for Computational Linguistics.

Misra, A. & Basak, S., 2016. *Political Bias Analysis,* Stanford: Stanford University.

mySociety, 2019. *Overview | ParlParse.* [Online]

Available at: https://parser.theyworkforyou.com

[Accessed 10 October 2019].

pandas, 2020. *pandas - Python Data Analysis Library.* [Online]

Available at: https://pandas.pydata.org

[Accessed 18 March 2020].

Yano, T., Resnik, P. & Smith, N. A., 2010. Shedding (a thousand points of) light on biased language. *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk,* pp. 152-158.

Zhang, Y., Jin, R. & Zhou, Z.-H., 2010. Understanding bag-of-words model: a statistical framework. *Int. J. Mach. Learn. & Cyber.*, Issue 1, pp. 43-52.