Hansard Historical Sentiment Analysis and Comparison

Report Name	Outline Project Specification
Author (User Id)	Adam Neaves (adn2)
Supervisor (User Id)	Amanda Clare (afc)
Module	CS39440
Degree Scheme	GH7JP (Artificial Intelligence and Robotics)
Date	February 11, 2018
Revision	1.0
Status	Release

---

1 Project description

The Hansard Sentiment Analysis will develop a Python based Sentiment Analysis tool trained on Parliamentary debates, and aims to provide an output comparing the sentiment of modern Parliament with sentiment found in the historical Hansard Dataset.

The Hansard Dataset provides a written record of all written and oral debates held in the House of Commons and the House of Lords from between 1803 and 2004, in XML. Using this archive of data, I hope to be able to extract the debates themselves from the dataset and produce a record of sentiment on a number of topics, which can then be compared to the modern debates, as they are released. In doing this, I should be able to detect any patterns in the change of opinion, hopefully showing trends in how political opinion has changed over the past 200 years.

Python 3 has been selected as the language of choice, due to its accessibility, and the availability of multiple packages that may prove useful for this project, such as BeautifulSoup for XML parsing, and spaCy or NLTK for the Natural Language Processing.

2 Proposed tasks

2.1 Investigate Hansard Data

The Hansard Dataset provides full records of debates in an XML format. However, because these are historical documents going back over 200 years, the formatting of the files can be disorganised. Examples of the data will have to be looked at, to evaluate how the system can confidently extract the relevant data from these files, and which parts are going to be useful to the project. From preliminary investigations, it appears that each series of Hansard Data is formatted slightly differently from each other, which is something the system will have to be able to handle.

2.2 Investigation of XML Parsers for Python

The provided Hansard Dataset is a series of XML files. These files need to be parsed and tidied by the system before they can be used for Sentiment Analysis. I need to investigate the XML parsers available for Python, and work out which will work best for me. BeautifulSoup is a popular parser, but there is also XBase, which would allow me to query the data in a similar fashion to using SQL.

2.3 Investigate Sentiment Analysis Tools

There are a number of Natural Language Processing packages available for Python. The options will need examining, and a package selected based on chosen factors, which will also need to be selected and prioritized. Additionally, the type of sentiments assigned will have to be chosen, as there are options for either assigning a score for positive or negative, or possibly assign an emotional label to the speaker, such as ”happy”, ”sad” or ”angry”.

2.4 Create Training Dataset

The sentiment analysis tool will be a form of machine learning. In order to efficiently train the system, a set of labelled training data will have to be provided, which will have to be created by hand. This may involve the creation of a tool to aid in labelling the data. The creation of this will save time over the course of the project.

2.5 Train Sentiment Analysis Model

Once the training and testing datasets have been created, the Sentiment Analysis model can be trained on the data. This should produce the model that will assign sentiment to speeches made, so that they may be compared.

2.6 Investigate Potential Data Sources

Each document has a list of attending members of Parliament. It may be possible to find additional data sources that provide more information about the members, such as political party, or date they started at Parliament. If this data is available, it can be used to compare more about the sentiment, perhaps allowing the system to compare sentiment of a particular individual at the start of their Parliamentary position compared to the end, or comparing sentiment between political parties.

3 Project deliverables

3.1 Sentiment Analysis Model

A model trained and tested on the Hansard Dataset, capable of detecting the sentiment from the questions and answers provided in the dataset.

3.2 Training and Testing Datasets

The datasets used to train and test the model, which will be provided as an appendix to the report written to show what the model was trained on.

3.3 Data Sentiment labelling Tool

If it is deemed appropriate, a tool designed to help manually label sentiment in order to produce the training dataset will be created, and delivered alongside the core software.

Annotated Bibliography

[1] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st ed., 2009, vol. 43, p. 479.

A guide to the Natural Language Toolkit for Python, a specific package that may be used for the project.

[2] H. M. Noble, *Natural Language Processing*, 1st ed., T. Addis, B. DuBoulay, and A. Tate, Eds. Oxford: Blackwell Scientific Publications, 1988.

This book provides a decent overview on Natural Language Processing for computer systems. It does not, however, mention any specifics for Sentiment Analysis.

[3] J. Perkins, "Text Classification For Sentiment Analysis - Naive Bayes Classifier," 2010. [Online]. Available: https://streamhacker.com/2010/05/10/text-classification-sentiment-analysis-naive-bayes-classifier/

An online article that describes how the author trained a Naive Bayes classifier on movie reviews using the NLTK for Python. It should serve as a good reference when looking to do something similar.

[4] L. Richardson, "Beautiful Soup Documentation." [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Documentation for BeautifulSoup4, a HTML/XML parser for Python, which could be useful for dealing with the large badly formatted XML documents.