Hansard Historical Sentiment Analysis and Comparison

Final Report for CS39440 Major Project

Author: Adam Neaves (adn2@aber.ac.uk)
Supervisor: Dr. Amanda Clare (afc@aber.ac.uk)

4th March 2018
Version: 1.0 (Draft)

This report was submitted as partial fulfilment of a BSc degree in Artificial Intelligence and Robotics (GH7P)

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, UK

Declaration of originality

I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the regulations on Unacceptable Academic Practice from the University's Academic Quality and Records Office (AQRO) and the relevant sections of the current Student Handbook of the Department of Computer Science.
- In submitting this work I understand and agree to abide by the University's regulations governing these issues.

Name ....

Date ....

Consent to share this work

By including my name below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name ....

Date ....

Acknowledgements

I am grateful to...

I’d like to thank...

Abstract

Politics affects all aspects of a persons life. The results of debates in Parliament may have a profound influence on an individuals life, and knowing how your local MP speaks and the opinions they express during these debates could prove useful. Websites, such as https://www.theyworkforyou.com, show a user how their local MP votes, and how often they attend debates, ask questions, and other information that may allow the user to make informed decisions when voting during elections.

The aim of the Hansard Sentiment Analysis Tool is to provide a tool to detect the sentiment of statements made during political debates, and to compare it with sentiment expressed about the topic previously, or compare it with sentiment expressed by the same MP. The sentiment analysis will use a machine learning approach, where a model will be trained on labelled datasets generated from the source data.

Being able to view how sentiment changed over time, expecially about certain topics, could prove useful for a user who wants to not only know how an MP votes, but also how they represent themselves and their constituency in parliament. If an MP is seen to change opinion on a topic as time passes, this information could be used to keep voters informed.

This report will document the process of designing and developing the Hansard Sentiment Analysis Tool, highlighting any challenges encountered during development, and also the results of the technical work.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Background & Objectives

1.1 Background

1.1.1 Hansard Dataset

The Hansard Dataset is a set of documents produced by the British Parliament, which began in the 18th and 19th century. These documents contain reports and details of debates in the House of Commons, going back to the year 1803. Eventually, in 1907, these reports were made official and started being produced by Parliament itself, becoming The Official Report, though still unofficially known as Hansard. Along with becoming official, a report was officially defined as being one:

> which, though not strictly verbatim, is substantially the verbatim report, with repetitions and redundancies omitted and with obvious mistakes corrected, but which on the other hand leaves out nothing that adds to the meaning of the speech or illustrates the argument" [1]

Hansard is available in a variety of versions. The most commonly used and best known version is the Daily Hansard, which appears each morning and reports of the previous days proceedings. However, access to this is via an API that only provides the most recent 7 days. For this project, most training and processing will be done on the Historical Hansard dataset, which is a dataset containing all 6 series of Hansard, between 1803 to 2004, though it is expected that some of the older documents will be less useful for this project due to the likelihood of them using outdated speech that would no longer be relevant.

The Historical Hansard Dataset is available online in an XML format. Multiple documents per series are available, each document covering a few days debates at most. It is a large dataset, reaching around 10Gb in size in total. Most of the documents available are scanned from hard copies, rather than typed up directly, meaning there is a possibility of small errors from the scanning process that may have to be dealt with. Additionally, from preliminary looks, the data itself appears to be only loosely formatted, and each of the six series appear to be formatted in a slightly different way, so any system designed to read this data will have to be capable of dealing with any changes to formatting.

1.1.2 Natural language Processing

Natural Language Processing (NLP) is the process of getting a computer to read and understand written text. Computers are very good at dealing with numbers and performing complex calculations at high speed but are not as good at understand spoken or written language. Because of this, a large part of NLP is the act of processing the data, or text, to make it easier for the computer to understand and work with. NLP covers multiple topics, such as Named Entity Recognition, part of Speech Tagging, and Sentence Boundary Disambiguation. However, the part this project is mainly interested is the act of Sentiment Analysis.

Sentiment Analysis, also known as Opinion Mining, is the process of identifying and extracting the opinions expressed in a piece of text. It aims to determine the attitude of a speaker or writer towards a topic, or the overall polarity of a piece of text. This can be a judgement made by the writer or speaker, in the case of reviews, or the emotional state of the speaker or writer.

A basic version of Sentiment Analysis classifies the polarity of a piece of text, classifying it as either positive, negative or neutral. A more advanced version would be, for example, looking at emotions expressed in the text, classifying it as angry, happy, or sad, as some examples. A basic method used can be to compare a piece to two lists of words, one a list of words that usually denote a positive polarity, and one that usually denotes a negative polarity. A system can then simply count the number of positive and negative words in a piece of text, account for any negation (Saying not great would change the word great from a positive to a negative word, for instance) and whichever type of word was most common would denote the piece of texts sentiment. However, this method is likely only useful for those pieces of text where its known that strong sentiment is likely to be expressed in a simple enough manner, in text such as a review.

Stance Detection is another aspect of Natural Language Processing, similar to Sentiment Analysis. However, the difference here is that Stance Detection sets out to classify the relative stance of two pieces of text, classifying whether the texts agrees with, disagrees with, discusses, or are unrelated to each other. An example would be detecting the stance of a news article compared to its headline. This may be more applicable to the Hansard Dataset than Sentiment Analysis as the members of parliament are likely to be expressing some form of stance on a topic that they are debating, but also somewhat more complicated to do, due to the additional classes involved.

1.1.3 Related Work

In reasearching the potential design of project, a few relevant pieces of work done by others were discovered, some of which had a useful impact on the design of this project.

Towards sentiment analysis on parliamentary debates in Hansard [2] is a paper which discussed the progress made by a group of PHD researchers towards applying classic sentiment analysis techniques to the hansard dataset, such as word association. TODO FINISH THIS WHEN CAN READ PAPER

They Work For You (CITE HERE) is a website that allowed the user to search for their local MP via post code, and the site can then display the voting patterns for that MP, along with information about how often their votes align with their parties votes, and shows examples of appearences made by that MP and what they said. The source code is publicly available on Github (CITE HERE) and uses python for a large part of their code base. Whilst it does not appear that they use any form of sentiment analysis, its still a good example of the sort of thing that can be done using the parlimentary data, and would likely be well suplimented by my project, allowing them to also show how an MP might speak in debates, as well as how they vote.

The Fake News Challenge (CITE HERE) is a challenge set up to explore how artificial intelligence technologies could be leveraged to combat fake news. and aims to eventually produce a tool that can help human fact checkers tell if a news story is a hoax, or intentionally misleading. The first part of the challenge involved the use of Stance Analaysis on a series of news articles, comparing the contents of the article with the headline, to tell if the article contents agree with the headline or not. As the project is set up as a competition, multiple teams submitted solutions to the problem, showing a variety of techniques in solving this problem.

1.2 Analysis

Taking into account the problem and what you learned from the background work, what was your analysis of the problem? How did your analysis help to decompose the problem into the main tasks that you would undertake? Were there alternative approaches? Why did you choose one approach compared to the alternatives?

There should be a clear statement of the objectives of the work, which you will evaluate at the end of the work.

In most cases, the agreed objectives or requirements will be the result of a compromise between what would ideally have been produced and what was determined to be possible in the time available. A discussion of the process of arriving at the final list is usually appropriate.

As mentioned in the lectures, think about possible security issues for the project topic. Whilst these might not be relevant for all projects, do consider if there are relevant for your project. Where there are relevant security issues, discuss how they will this affect the work that you are doing. Carry forward this discussion into relevant areas for design, implementation and testing.

1.3 Process

You need to describe briefly the life cycle model or research method that you used. You do not need to write about all of the different process models that you are aware of. Focus on the process model that you have used. It is possible that you needed to adapt an existing process model to suit your project; clearly identify what you used and how you adapted it for your needs.

Appendices

The appendices are for additional content that is useful to support the discussion in the report. It is material that is not necessarily needed in the body of the report, but its inclusion in the appendices makes it easy to access.

For example, if you have developed a Design Specification document as part of a plan-driven approach for the project, then it would be appropriate to include that document as an appendix. In the body of your report you would highlight the most interesting aspects of the design, referring your reader to the full specification for further detail.

If you have taken an agile approach to developing the project, then you may be less likely to have developed a full requirements specification. Perhaps you use stories to keep track of the functionality and the 'future conversations'. It might not be relevant to include all of those in the body of your report. Instead, you might include those in an appendix.

There is a balance to be struck between what is relevant to include in the body of your report and whether additional supporting evidence is appropriate in the appendices. Speak to your supervisor or the module coordinator if you have questions about this.

Appendix A

Third-Party Code and Libraries

If you have made use of any third party code or software libraries, i.e. any code that you have not designed and written yourself, then you must include this appendix.

As has been said in lectures, it is acceptable and likely that you will make use of third-party code and software libraries. If third party code or libraries are used, your work will build on that to produce notable new work. The key requirement is that we understand what is your original work and what work is based on that of other people.

Therefore, you need to clearly state what you have used and where the original material can be found. Also, if you have made any changes to the original versions, you must explain what you have changed.

As an example, you might include a definition such as:

Apache POI library - The project has been used to read and write Microsoft Excel files (XLS) as part of the interaction with the client's existing system for processing data. Version 3.10-FINAL was used. The library is open source and it is available from the Apache Software Foundation [**?**]. The library is released using the Apache License [**?**]. This library was used without modification.

Appendix B

Ethics Submission

This appendix includes a copy of the ethics submission for the project. After you have completed your Ethics submission, you will receive a PDF with a summary of the comments. That document should be embedded in this report, either as images, an embedded PDF or as copied text. The content should also include the Ethics Application Number that you receive.

Appendix C

Code Examples

For some projects, it might be relevant to include some code extracts in an appendix. You are not expected to put all of your code here - the correct place for all of your code is in the technical submission that is made in addition to the Final Report. However, if there are some notable aspects of the code that you discuss, including that in an appendix might be useful to make it easier for your readers to access.

As a general guide, if you are discussing short extracts of code then you are advised to include such code in the body of the report. If there is a longer extract that is relevant, then you might include it as shown in the following section.

Only include code in the appendix if that code is discussed and referred to in the body of the report.

Annotated Bibliography

[1] "Factsheet G17 General Series The Official Report," 2010. [Online]. Available: https://www.parliament.uk/documents/commons-information-office/g17.pdf

[2] O. Onyimadu, K. Nakata, T. Wilson, D. Macken, and K. Liu, "Towards sentiment analysis on parliamentary debates in Hansard," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8388 LNCS, 2014, pp. 48–50.

BLOODY CANT READ AT THE MOMENT CAUSE PAYWALLS

[3] "NLTK with Python 3 for Natural Language Processing - YouTube - YouTube." [Online]. Available: https://www.youtube.com/playlist?list=PLQVvvaa0QuDf2JswnfiGkliBInZnIC4HL

[4] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text." [Online]. Available: http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

[5] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, "A simple but tough-to-beat baseline for the Fake News Challenge stance detection task," jul 2017. [Online]. Available: http://arxiv.org/abs/1707.03264

[6] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, "Stance Detection with Bidirectional Conditional Encoding," jun 2016. [Online]. Available: http://arxiv.org/abs/1606.05464

[7] S. Dori-Hacohen, "Controversy Detection and Stance Analysis," *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1057–1057, 2015.

[8] W. Ferreira and A. Vlachos, "Emergent: a novel data-set for stance classification," pp. 1163–1168. [Online]. Available: http://aclweb.org/anthology/N/N16/N16-1138.pdf

[9] A. Aker, L. Derczynski, and K. Bontcheva, "Simple Open Stance Classification for Rumour Analysis," aug 2017. [Online]. Available: http://arxiv.org/abs/1708.05286

[10] P. Nagy, "Python NLTK Sentiment Analysis," 2017. [Online]. Available: https://www.kaggle.com/ngyptr/python-nltk-sentiment-analysis/notebook

[11] O. Onyimadu, K. Nakata, Y. Wang, T. Wilson, and K. Liu, "Entity-Based Semantic Search on Conversational Transcripts Semantic." Springer, Berlin, Heidelberg, 2013, pp. 344–349. [Online]. Available: http://link.springer.com/10.1007/978-3-642-37996-3{_}27

[12] 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.

[13] 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.

[14] H. M. Noble, *Natural Language Processing*, 1st ed., T. Addis, B. DuBoulay, and A. Tate, Eds. Oxford: Blackwell Scientific Publications, 1988, 0632015020.

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

[15] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st ed., 2009, vol. 43, p. 479. [Online]. Available: http://www.amazon.com/dp/0596516495 9780596516499.

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