Background

**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" (PUT REF HERE)*

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 day’s 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is expected that some of the older documents will be less useful for this project due to the likelihood of them using outdated speech that would no longer be relevant.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.

**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 text’s sentiment. However, this method is likely only useful for those pieces of text where it’s 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.

**Related Work**