The Use of Structural Topic Modelling to Analyse Survey Text Data

Anna-Grace Linton

April 2022

Abstract

1 Executive Summary

This report summarises the work carried out as part of the NHSX PhD data science internship project on structural topic modelling for survey text data. The main aim of the project was to evaluate the use of machine learning methods to gain insight from free text responses from NHS surveys readily. Structural topic models are advantageous in identifying topics discussed in the text and the effect of information associated with this text. This project i) evaluates how the text analytic methods and structural topic modelling can be used together to gain insight from large bodies of text, and ii) provides a method to analyse survey text data demonstrated using openly available data.

2 Introduction

The use of free text data is essential in making better use of the data collected. Specifically, we consider the free text in NHS survey data, which often provides context to responses to closed questions within a survey and provides valuable information otherwise missing from the rest of the survey [10], [11]. However, these surveys can be extensive. For example, 648,594 participants responded to the NHS Staff survey in 2020 [25] and the annual Cancer Patient Experience Survey generates more than 70,000 comments each year [14]. Whereas the analysis of the closed questions in these surveys can be done efficiently using statistical software, analysis of free text responses is often conducted manually, making it a resource and time-demanding task. As a result, these responses are often under-explored, and when they are, there is a prolonged time between collection and gaining actionable insight from the analysis.

This project evaluates the use of machine learning techniques as tools for rapid analysis of free-text responses from NHS survey data for actionable recommendations. In particular, this project looks at the extent to which structural topic modelling (STM), an unsupervised text classification technique, can be used alongside other text analysis methods to gain insight into topics discussed in the responses. This is explored in this project by focusing on i) preprocessing methods to facilitate data preparation, ii) sentiment analysis on responses, iii) accessible method for model selection, labelling topics and sentiment and comparing models using qualitative and quantitative measures, and iv) visualising and utilising results of stm for further analysis of the data. In this report, Section 3 provides an overview of STM and its value in this project. Section5 outlines the methodology of experiments conducted in this project, while Section 6 details the results of the experiments. Finally, section 7 discusses this approach for analysing survey text data and considerations for a continuation of this project.

3 Structural Topic Modelling

Topic modelling and word clustering are common natural language processing (NLP) approaches to obtaining insight into text [19]. They have been used to facilitate qualitative text analysis through automating topic extraction, and grouping semantically similar text [11], [21]. Structural topic models (STM) are a generative models that are an extension of topic models such as Latent Dirichlet Allocation (LDA) [1] and Correlated

Question	Frequency	%
1	6242	60.4
2	4092	39.6

Organisation	Frequency	%
Trust A	2625	25.4
Trust B	3307	32.0
Trust C	4402	42.6

Table 1: Description of data

Topic Model (CTM) [2]. These models identify latent topics in text. A topic is a set of words where each word has a probability of belonging to that topic. A document is a mixture of topics that can also be correlated. Unlike LDA and CTM, STM enables the covariates (metadata) to be associated with a document of interest [7]. The metadata covariates may influence the topic mentioned or during data generation, such as the date or trust the survey is collected. For example, feedback during the winter may include details about longer wait times and demand on services. Essentially, STM enables context to be added when generating a topic model. The details of the STM are described in full in [7]. In general, the model iterates through each word in a document. Based on a prior distribution of the topic proportions, the model assigns the word to a topic. The metadata covariates can influence the prevalence of the topics. In this way, documents with similar covariates will tend to mention similar topics and use more similar words to discuss them.

The value of STM lies in its ability to discover topics in a corpus and estimate the effect of the associated metadata. The analyst is then able to look at the relationship between variables and topics in the text, which enables model interpretability and hypothesis testing [13]. Additionally, the inclusion of covariates has been shown to improve the accuracy of topic inference [4]. Some applications of STM have included examining the public opinion of the UK government throughout the COVID-19 pandemic [24]; understanding causes of user dissatisfaction from complaints [20] and topics present in aviation incident reports [12]. STM has also been used alongside other text analytic methods such as sentiment predictions and hierarchical clustering to improve the usability of Intelligent Personal Assistants [22].

4 Data

The data used for this work is open-source data from the Nottinghamshire Healthcare NHS Foundation Trust patient feedback text data. This data has been labelled manually by a team at the Trust (cite). The data consisted of 10, 344 textual responses and associated metadata.

The metadata associated with the feedback was criticality (-4 to 4), organisation (Trust A, Trust B, or Trust C), question (1 or 2), label (e.g., Staff, Access, Environment/Facilities), code (e.g., xn, ap, mi) and subcategory (e.g., Staff: general, Provision of Services and Facilities/equipment). Although label, code and subcategory did not affect the STM strongly, there were not used as metadata in the experiments. The distribution of the criticality, question and organisation is shown in Table 1

5 Methods

5.1 Overview of approach

The project was set up using R programming due to the availability of the stm, an R package for structural topic model [15] and the availability of several open-source libraries for text analytics.

5.2 Preprocessing

The data were cleaned to remove rows with null values and declare the data types of each variable, a process dependent on the dataset. The responses were prepared for topic modelling by expanding contractions (e.g., don't -¿ do not) , removing punctuation and digits, making words lowercase and removing stopwords. Stopwords are words which convey little to no additional meaning to a sentence such as "a" "and" "am". These stop words are able to be customised to include other non-informative words from the corpus as determined by the user such as "nothing" and "nope". The words were also normalised to their root using

stemming or lemmatisation, tokenised and converted into a document feature matrix for STM. All tokens are unigrams, unless otherwise stated. Responses that had fewer than 3 tokens were removed.

In stemming a part of the word is removed to reduce the word it stem. In lemmatisation, a word is mapped to its lemma, or canonical form. Lemmatisation can been seen as more interpretable as the root derived are real words and words such as "happy" and "happily" map to "happy" where as with stemming they would map to "happy" and "happi". "run", "running" and "ran" all map to "run" .

5.3 N-gram analysis

Tidytext library was used to extract unigrams, bigrams and trigrams from the text.The frequency of these n-grams was calculated, which provided an overview of the types of words and phrases in the data. This was used to highlight features that needed to be considered in the preprocessing. Unigrams and bigrams were used as the input for stm in an experiment. Otherwise, only unigrams were used.

5.4 Sentiment Analysis

Sentiment analysis calculates the affective state of text. Commonly, polarity is calculated, in which a score is given stating how positive, negative or neutral a statement is. Several sentiment analysis tools available in R perform well on text. On the raw data, we compared the performance of VADER [6], Sentiment-Analysis [9], Affin [3] and NRC emotion lexicon [5]. The libraries operate using different methods and are trained with different corpora, for example, VADER is a rule based dictionary trained on Twitter data and SentimentAnalysis: Dictionary GI is a general purpose dictionary using Havard-IV dictionary.

5.5 Structural Topic Modelling

SSTM was implemented using stm R package [15], to identify topics in the responses. We ran stm on 5 to 65 topics using searchK to select several models to inspect in detail. Semantic coherence, exclusivity score, heldout log-likelihood and lower bound were used to determine the number of topics for the model (K). Once a smaller number of models (K=20, 25, 30) were selected, quantitative metrics and qualitative measures were used to evaluate model suitability. The average semantic coherence, exclusivity score, heldout log-likelihood of topics in a model and the semantic coherence and exclusivity score of each topic within a model were compared. A well performing model would seek to maximise these metrics (finding a balance between them). The models were evaluated qualitatively by manually looking at representative text (higher proportion of text estimated for a given topic) and the most associated words, such as words ranked highest by FREX score and those with the highest probability. FREX score is a weighted mean of the probability of a word appearing in a topic (frequency) and its exclusivity to a topic.

To visualise and interpret the topic models we examined the range of outputs in the stm package, such as word clouds, plot the estimated effect of the metadata, and print most associated words. ToLDAvis was used to visualise the topic-word distributions in an interactive pop-out window. This provided an overview of topic quality by looking at topic content and similarity. stminsights package [23] was used to produced an interactive dashboard for a detailed inspection of the model and topics.

5.6 Text search

Text search feature enables the user to input a list of terms to search for in the text. Using the svDialogs library in R markdown, a pop-up box prompts the user for a list of terms to search. These terms can be single or multi-word phrases, separated by a comma. The synonyms function from the WordNet package is used to expand the search terms to similar terms that are also withi the the data's vocabulary. The user is prompted to consider expanding the search terms to include similar terms with another pop-up box. The search results in a data frame filtered to contain text with the searched terms and their associated data. The data is summarised in a series of graphs.

Text	VADER	SA-GI	SA-HE	NRC	Affin
Food was not good, lack of choice and poor quality. Meals did not cater to a wide variety of diets. As an autistic person, there are certain foods I don't eat due to sensory issues and the lack of choice made it difficult to get a healthy balanced diet with enough calories.	-0.549	-0.0667 0	0.033	0.625091	-0.14286
Do not judge people (like myself) for how they look (looks can be deceiving, like the says 'Do not judge a book by its cover !!!) but judge them by how they feel (especially me). As a Christian lady and my faith keeps me going and the person I am, honest, gentle, caring and loving, but I am very slow for my age.	0.975	0.230769	0	0.65596	1

Table 2: Example feedback comments with the sentiment score from each sentiment analysis library. SA-GI: SentimentAnalysis - General Inquirer, SA-HE: SentimentAnalysis - Henry's Financial Dictionary, NRC: NRC emotion lexicon

6 Results

This approach generated data labelled with the two most likely topics mentioned in the feedback and sentiment score and developed means to visualise the results of the analysis. This provided a mechanism to rapidly evaluate and select the model most suitable for this data. The results reported here detail the experiments that contributed to designing an approach for this project.

We compared how different sentiment analysis libraries performed on this dataset. We used the sentiment libraries as described in 5. Table 2 shows the score on two example feedback comments. There were some differences between the scores given by the sentiment analysis tools. For example, SentimentAnalysis - HE is based on Henry's Financial Dictionary and, although it was able to identify opinionated words, was more likely to score the text as neutral. VADER was selected as the sentiment analysis tool for this data. The selected library, VADER was selected as was used to label each feedback response with a sentiment score. VADER was better able to capture feedback with a negative sentiment than the other libraries. Feedback responses tended towards a neutral or strongly positive sentiment for this dataset. The sentiment score was used as an additional variable of the metadata in the subsequent experiment or as an option to filter the dataset to responses with positive or negative sentiment only.

We evaluated the impact of the various preprocessing methods on the performance of STM. We used stm package preprocessing as a base and compared the impact of using n-grams and sentiment as input variables and lemmatisation, stemming in for preprocessing. Figure 1 shows the performance of a model with 25 topics. There is a great similarity in the overall performance of the model between preprocessing methods. Expectantly, we see that the use of n-grams as an input into STM increased the semantic coherence of topics compared to base (mean = -156.97, difference = +23.5%) and negatively impacted the exclusivity (mean = 9.811, difference = - 0.5%). Consequently, n-grams were not considered for input into stm. Instead, n-grams were used only to understand the vocabulary used in the text, such as domain-specific terminology, and to inform the preprocessing. Overall, including lemmatisation in the preprocessing improved the quality of the topics marginally based on exclusivity and semantic coherence scores.

We looked to produce an interactive visualisation of the analysis. This is to visualise the output of the models and search the data to find all responses with their associated data relevant to a set of search terms. Examples of the visualisations are shown in figure 2. These were beneficial in alternating between models to compare their performance. toLDAvis enabled the most relevant words to a topic to be observed alongside the frequency relative to the word's frequency in the entire corpus.

Further, it allowed the prevalence of the topic (the size of the circle) and the similarity of the topics represented in 2D space, with a substantial overlap representing a considerable similarity in the topic context.

	Heldout log likelihood	Exclusivity	Semantic Coherence
Base	-5.772	9.863	-205.244
Stemmed	-5.632	9.870	-195.782
Lemmatised	-5.920	9.873	-216.889
N-gram	-7.476	9.811	-156.974
Sentiment	-5.837	9.862	211.609

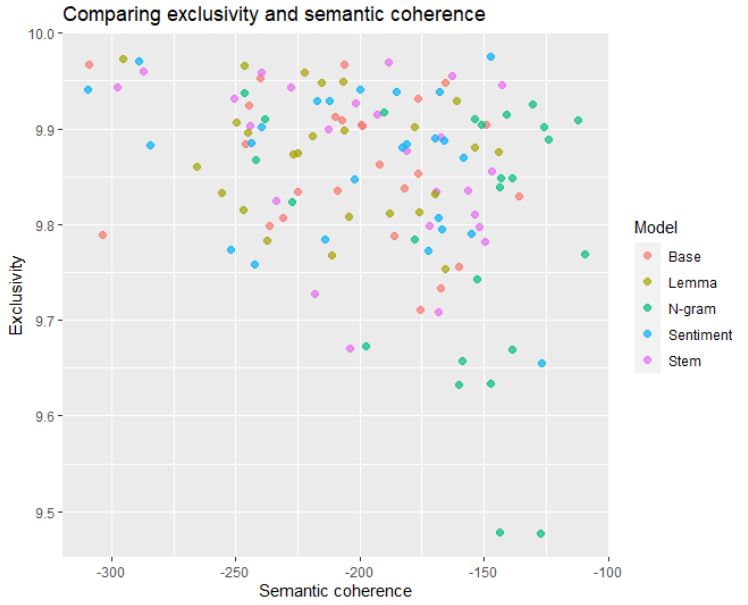

Figure 1: Exclusivity, Semantic coherence and held out log likelihood of topics for STM K=25. Stem - stemming, lemma - lemmatisation

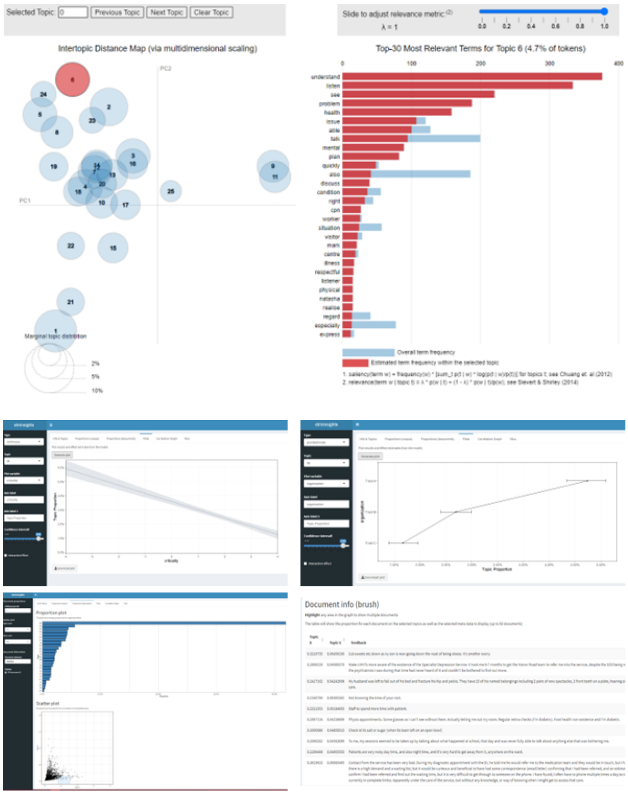

Figure 2: Example visualisations using toLDAvis (Top) and stminsights (Bottom).

stminsights enabled closer inspection of topics in models in a dashboard. The models, topics and effect estimates of interest could be selected quickly, and related documents and terms presented. It also allowed for input by the user through manual topic labelling and adjustment of metric parameters. stminsight proved helpful in comparing models during model selection and visualising a series of outputs for a selected model.

The text search feature of this project enabled users to input a set of search terms such as "staff, doctor, nurse" and produce a table of the data containing these and similar terms as well as summary visualisations as in figure 3. This part of the project utilises the data labelled with topics and sentiment in previous analyses. This approach allowed relevant data to be presented to the user. However, plurals of the words, such as "doctors" and "nurses", were not considered synonyms of the singular noun "doctor" and "nurse", and so texts containing these terms were not returned.

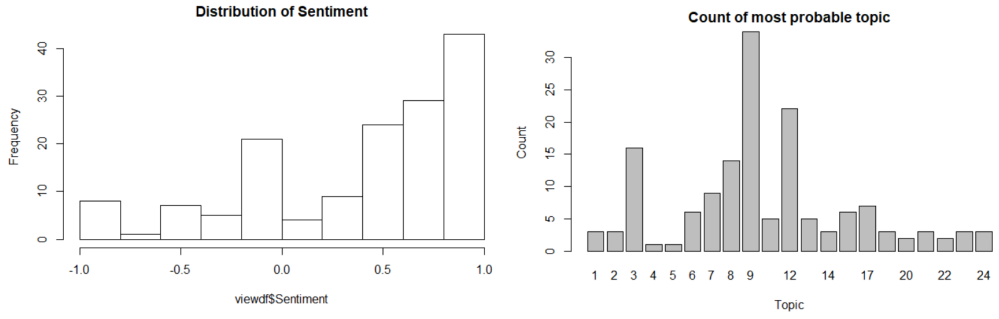

Figure 3: Example summary graph from result of text search.

7 Discussion

This project evaluated the use of STM and other text analytic methods to gain insight from survey text data in an accessible and readily available manner. Text survey data and accompanying metadata were analysed, interpreted and used for deeper exploration of the data. In this project, using open-source survey text data, we were able to identify topics mentioned in the feedback comments and provide a sentiment score. Also, this model enabled the analysis and reporting of the topics mentioned in the text, the effect of the metadata on them and "zoom in" to select information relating to specific content. This approach has the potential to be used as a tool to readily extract and summarise information from survey text data for tasks such as gathering the main information considering a specific question or concept. Alternatively, it has the potential to reduce feedback data to responses relating to specific issues of interest (topics) so that other manual or automated analyses can be conducted on a reduced dataset [18].

In this project, we initially focused on the preprocessing of the data as it has been shown to improve the performance of text classification models [8], [17]. While we explored some of the available tools, other methods and libraries could also be explored, such as contextual language models and spaCy, a suite of natural language processing pipelines. Moreover, further standardisation of the text in preprocessing can be explored in subsequent work, such as the impact of grouping similar concepts into one entity on the model's ability to learn more nuanced topics. For example, "spouse", "children", and "sister" could be grouped as "family".

This project sought a user-focused approach to visualise, interpret, and use STM analysis, combining unsupervised topic discovery (STM) with user-driven information extract (text search). By identifying occurrences of terms the user has specified, they can further examine specific concepts and gain relevant information. This feature alongside the visualisation is especially useful to those who do not use R programming as the interactive interface allows the user to generate a wide range of outputs easily. The current text search implementation does not perform wildcard searches or uses Boolean operators. Immediate continuation of this project would include these features to capture relevant text better. WordNet was used to find the synonyms of the search terms. There is scope to explore the degree of similarity to expand the search by moving further up the hypernyms and subsequent hyponyms of WordNet. Once improved, the text search function also has the potential to be used following analysis or during the preprocessing to run STM on feedback relating to a specific area.

As an indicator of quality, the balance between semantic coherence, exclusivity and held out log-likelihood served as a valuable way to evaluate a suitable number of topics. Using this method, a suitable small range of topics was identified to be evaluated in more depth. The balance of these metrics is a useful guide to the performance of the topic model. However, evaluation of the models still required qualitative evaluation. The interactive visualisation aided this evaluation by presenting the representative text and words for the parameters chosen. Regardless, there is still a fair amount of human effort required to determine the labels of the topics. Often, the words with the highest FREX score or highest probability are insufficient to label the topics with a human readable label. Further work could investigate automatic labels such as using neural based approaches [16].

As stated earlier, a key benefit of STM is the ability to incorporate metadata into the model and to estimate its effect for model interpretation and hypothesis testing. Although not it was not explored in this project extensively, we are able to see the effect of organisation, criticality score and question on the topics. STM can readily model topics in the context of more data such as time and regions that may influence the prevalence of a topic or its content [24]. This idea can be further investigated in other datasets and is beneficial in looking at additional metadata and aiding with subgroup analysis.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[2] J. Lafferty and D. Blei, "Correlated topic models," *Advances in neural information processing systems*, vol. 18, 2005.

[3] F. Å. Nielsen, "A new anew: Evaluation of a word list for sentiment analysis in microblogs," *arXiv preprint arXiv:1103.2903*, 2011.

[4] L. Du, W. Buntine, and M. Johnson, "Topic segmentation with a structured topic model," in *Proceedings of the 2013 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies*, 2013, pp. 190–200.

[5] S. M. Mohammad and P. D. Turney, "Nrc emotion lexicon," *National Research Council, Canada*, vol. 2, 2013.

[6] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the international AAAI conference on web and social media*, vol. 8, 2014, pp. 216–225.

[7] M. E. Roberts, B. M. Stewart, D. Tingley, *et al.*, "Structural topic models for open-ended survey responses," *American journal of political science*, vol. 58, no. 4, pp. 1064–1082, 2014.

[8] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Information processing & management*, vol. 50, no. 1, pp. 104–112, 2014.

[9] S. Feuerriegel and N. Proellochs, *Dictionary-based sentiment analysis*, Performs a sentiment analysis of textual contents in R. This implementation utilizes various existing dictionaries, such as General Inquirer, Harvard IV or Loughran-McDonald. Furthermore, it can also create customized dictionaries. The latter uses LASSO regularization as a statistical approach to select relevant terms based on an exogeneous response variable, 2016. [Online]. Available: `https://github.com/sfeuerriegel/SentimentAnalysis`.

[10] E. Harrop, F. Morgan, A. Byrne, and A. Nelson, ""it still haunts me whether we did the right thing": A qualitative analysis of free text survey data on the bereavement experiences and support needs of family caregivers," *BMC palliative care*, vol. 15, no. 1, pp. 1–8, 2016.

[11] T. C. Guetterman, T. Chang, M. DeJonckheere, T. Basu, E. Scruggs, and V. V. Vydiswaran, "Augmenting qualitative text analysis with natural language processing: Methodological study," *Journal of medical Internet research*, vol. 20, no. 6, e9702, 2018.

[12] K. D. Kuhn, "Using structural topic modeling to identify latent topics and trends in aviation incident reports," *Transportation Research Part C: Emerging Technologies*, vol. 87, pp. 105–122, 2018.

[13] R. Wesslen, "Computer-assisted text analysis for social science: Topic models and beyond," *arXiv preprint arXiv:1803.11045*, 2018.

[14] C. Rivas, D. Tkacz, L. Antao, *et al.*, "Automated analysis of free-text comments and dashboard representations in patient experience surveys: A multimethod co-design study," 2019.

[15] M. E. Roberts, B. M. Stewart, and D. Tingley, "Stm: An r package for structural topic models," *Journal of Statistical Software*, vol. 91, no. 2, pp. 1–40, 2019. DOI: `10.18637/jss.v091.i02`. [Online]. Available: `http://statistik-jstat.uibk.ac.at/index.php/jss/article/view/v091i02`.

[16] A. Alokaili, N. Aletras, and M. Stevenson, "Automatic generation of topic labels," Jul. 2020, pp. 1965–1968. DOI: `10.1145/3397271.3401185`.

[17] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, "The influence of preprocessing on text classification using a bag-of-words representation," *PloS one*, vol. 15, no. 5, e0232525, 2020.

[18] A. Soteriades, *Cdu data science team blog: Classification of patient feedback*, 2020. [Online]. Available: `https://cdu-data-science-team.github.io/team-blog/posts/2020-12-14-classification-of-patient-feedback/`.

[19] T. Chang, M. DeJonckheere, V. V. Vydiswaran, J. Li, L. R. Buis, and T. C. Guetterman, "Accelerating mixed methods research with natural language processing of big text data," *Journal of Mixed Methods Research*, vol. 15, no. 3, pp. 398–412, 2021.

[20] X. Chen, G. Cheng, H. Xie, G. Chen, and D. Zou, "Understanding mooc reviews: Text mining using structural topic model," *Human-Centric Intelligent Systems*, vol. 1, no. 3-4, pp. 55–65, 2021.

[21] R. P. Lennon, R. Fraleigh, L. J. Van Scoy, *et al.*, "Developing and testing an automated qualitative assistant (aqua) to support qualitative analysis," *Family Medicine and Community Health*, vol. 9, no. Suppl 1, 2021.

[22] M. J. Sánchez-Franco, F. J. Arenas-Márquez, and M. Alonso-Dos-Santos, "Using structural topic modelling to predict users' sentiment towards intelligent personal agents. an application for amazon's echo and google home," *Journal of Retailing and Consumer Services*, vol. 63, p. 102 658, 2021.

[23] C. Schwemmer, *Stminsights: A shiny application for inspecting structural topic models*, R package version 0.4.1, 2021. [Online]. Available: `https://github.com/cschwem2er/stminsights`.

[24] L. Wright, A. Burton, A. McKinlay, A. Steptoe, and D. Fancourt, "Public opinion about the uk government during covid-19 and implications for public health: A topic modeling analysis of open-ended survey response data," *PloS one*, vol. 17, no. 4, e0264134, 2022.

[25] N. England. "2020 nhs national staff survey free text commentary." (), [Online]. Available: `https://www.england.nhs.uk/statistics/2021/05/27/2020-nhs-national-staff-survey-free-text-commentary/`.

A Set up

R studios was used to run the code in R version 3.6.1 on Windows 10. In Rstudios, run the environment by opening the R project (stmpreprocessing.rproj). The required libraries are contained in the file libraries.r and will install packages required if they are not already installed. The model sources the data from the folder data

B Installing WordNet in R Studios

When installing WordNet, rJava is required for installation. To install rJava on Windows, download and install Java for Windows Offline and Java JDK for Windows. In R, install rJava using install.packages("rJava"). Point the JAVA_HOME environment to Finally, run library(rJava). [taken from https://cimentadaj.github.io/blog/2018-05-25-installing-rjava-on-windows-10/installing-rjava-on-windows-10/]

To install WordNet, install it using install.packages("wordnet"). The dictionary for WordNet can be downloaded from `http://wordnetcode.princeton.edu/2.1/WordNet-2.1.exe`. Set the WordNet environemnt with Sys.setenv(WNHOME = "C:/Program Files(x86)/WordNet/2.1"). Run library(wordnet), which will throw an error. Set the dictionary for Wordnet with setDict("C:/Program Files(x86)/WordNet/2.1/dict") followed by getDict(). WordNet should now be fully installed in R.

C Visualisation

C.1 stmInsights

Stminsights is an interactive application that uses R Shiny. An .Rdata file containing the data, models to be visualised and the effect estimates are loaded in the browser. Multiple models can be saved in one file allowing for easy comparison between model outputs.