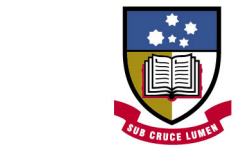

THE UNIVERSITY
of ADELAIDE

Something about NLP

by

Irulan Murphy

In fulfilment of the requirements for the degree of

Master of Philosophy

December 2023

Discipline of Mathematics

School of Computer and Mathematical Sciences

The University of Adelaide

Contents

List of Tables

List of Figures

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Irulan Murphy 20/12/2023

Acknowledgements

Firstly, I would like to thank my supervisors, Lewis Mitchell and John Maclean. Your support and guidance has been invaluable. You have made this difficult journey much, much easier.

To the whole of the Level 7 office, thank you for the laughs, tea, biscuits, and Dutch Blitz. You are the reason that I come in to uni every day, and the reason that I have made it this far. Even when things look bleak, I believe there is nothing that a good Dutch Blitz session can't fix.

Thank you also to my parents who "taught me everything I know". Mum, your constant encouragement pushes me to keep going and to try new things. Dad, your enthusiasm for maths is almost unmatchable and reminds me that I really do love maths.

I extend further gratitude to my parkrun family. Saturday mornings are a welcome distraction from life at university and I am grateful for such a wonderful group of people who support me unconditionally.

Finally, I would like to thank my partner, Matthew. Your love and support throughout this project has meant so much to me.

Abstract

Access to unbiased information is crucial in forming an objective view of events. While news media plays a large role in the public's understanding of events, it often exhibits a bias impacting the impartiality of these sources. Detecting media bias can be challenging, and so we seek to develop an automated method to quantify it. By breaking the bias down into two components—coverage and statement bias—we can more easily examine its different aspects. Furthermore, this approach allows a deeper understanding of the reasons behind potential biases, particularly in terms of differences in language used.

We analyse bias in the Tveeder data set of Australian television captions. This is the first wide-scale analysis of this data set and allows us to gain a clearer picture of how television media reflects the sentiment of the general public at a given time. This comprehensive analysis paves the way for future investigations into Australian television captions.

While firstly developing a measure for coverage bias, we explore CorEx topic modelling in detail. We provide a complete derivation of the technique, as well as real-world applications showcasing some of its capabilities. These include semi-supervised and hierarchical topic modelling, as well as making predictions on unseen documents. Collating these aspects of CorEx into a single document increases accessibility.

We introduce several methods to choose the number of topics, as well as to compare topic models. Additionally, since the Tveeder corpus does not have well-defined documents, we treat the document length as a parameter. We find that the length of document should be taken into consideration when training due to its significant impact on the quality of topic models. We also note the need to be cautious when applying CorEx topic modelling due to the large effect of randomness in the models, proposing to use the mean of topic probabilities to minimise the effect of this randomness.

Applying CorEx topic modelling to the data set, we qualitatively show that the topic probabilities provide a good estimate of media attention. From the media attention, we propose two measures to quantify coverage bias.

We then apply a sentiment analysis to determine statement bias. Due to the domain-specific nature of the language in this corpus, traditional sentiment lexicons struggle

to capture the sentiment within. We build our own sentiment lexicon from word embeddings trained on our corpus. Using a word shift analysis, we show this lexicon to be more effective at calculating sentiment in this corpus and present two methods to measure statement bias.

We then develop a bias measure that incorporates both coverage and statement bias, forming a comprehensive quantitative measure for media bias in general. We focus on two-party political bias as a simple and somewhat measurable application, noting that the technique can be easily extended to the analysis of bias towards any topic. Although a quantitative analysis of the measure's quality is beyond the scope of this thesis, we show that it broadly aligns with public polling data over time, demonstrating that the sentiment of the population can be reasonably inferred from television captions.

Introduction

0.1 Motivation

Access to unbiased information is crucial in forming an objective view of events. News media plays a large role in the public's understanding of events, however it often exhibits a bias, referred to as *media bias*.

Media bias is often defined from its opposite: neutrality. Neutral reporting is not strongly slanted in favour of any particular political view or topic. In this case, all views or topics are equally represented according to a benchmark for neutrality. Media bias measures the extent to which a media outlet differs from this benchmark [14].

It can, however, be difficult to difficult to determine a benchmark for neutrality without searching hundreds of articles and forming comparisons between different providers or channels. This is both time consuming and monotonous to do by hand. In addition, we may incorporate our own biases into an analysis of this bias! As such, we seek to develop an automatic method to identify media bias.

The definition of media bias can be refined further. Williams [64] requires that a bias must be intentional for it to be 'widely interesting', however intention can be difficult to determine and we don't require this.

Many common definitions also split media bias into three categories: coverage, gatekeeping, and statement [11, 14, 19, 42]. *Coverage bias* is concerned with the amount of attention that various topics or entities receive. *Gatekeeping bias* refers to the selection of particular articles or stories to be broadcast. *Statement bias* describes the way that topics are presented, whether in a positive or negative tone [19].

In this thesis, we merge both coverage and gatekeeping bias since they are directly related. Both forms can be measured by analysing the amount of attention received and it can be difficult to discern between the two without additional background knowledge. We refer to the bias measured by the amount of attention as simply *coverage bias*.

The bias measure that we develop will consider both coverage and statement bias, forming a comprehensive quantitative measure for media bias in general. We focus on two-party political bias as a simple and somewhat measurable application. Our bias measures generally focus on the difference between values corresponding with the two

major Australian political parties: Liberal and Labor. Although we focus on political bias, this can be easily extended to the analysis of bias towards any topic, most easily those with binary views.

0.2 Contributions

Our primary contribution is an analysis of the Tveeder data set [54]. This data set consists of a range of Australian television captions dating back to 2015. This thesis provides a comprehensive analysis of the captions encompassing both topic and sentiment modelling. We further draw conclusions regarding the correlations between television captions and real-world events. Finally, we present a widescale analysis of bias in the data. To our knowledge, this is the first detailed anlaysis of this data set.

We explore the application of Correlation Explanation (CorEx) topic modelling [16, 37, 46, 47] to quantify coverage bias. The CorEx topic modelling framework is relatively rarely used and has not been explored in a wide variety of applications. It has broad pratical applications utilising its abilities to 'anchor' words within a topic and to form a hierarchical structure. These techniques have been applied to testing data [16, 37, 46, 47], however they have not been implemented on real-world data outside of this. We make several contributions investigating this framework further while developing a quantitative bias measure.

We firstly provide a thorough derivation of the CorEx topic modelling framework in Section 1.2. This derivation includes consistent notation to increase clarity of the method. We also discuss extensions of the method to both hierarchical and semi-supervised topic modelling, again with consistent notation. Section 1.2.5 presents a method to make predictions for topic probabilities in previously unseen documents which has not yet been described in the literature. We apply this to real-life data in Section 3.3, comparing two methods of implementation. By consolidating CorEx and its extensions into a single document, we improve accessibility.

In Section 1.3, we suggest a novel technique for comparing topic models and how this can be extended to compare the texts being modelled. The comparison of two topic models trained on the same corpus has not yet been undertaken. We believe this could be done using the Kullback-Liebler divergence of word-topic probabilties for generative models such as Latent Dirichlet Allocation (LDA). In the case of CorEx, however, we suggest a similarity measure which exploits the probabilties of topics given the documents.

In Section 3.2, we methodically discuss the parameters required in the CorEx topic model. The number of topics has not been explored in great detail in the literature, nor has any indication been given as to how one may determine an appropriate number of topics to choose. We aim to address this gap, presenting three different methods to

determine this value and discussing the consequences of choosing an incorrect number of topics.

We further discuss how we may choose the length of documents in a corpus where a document is ill-defined and examine the effect of choosing the length to be too short or too long. The document length has not previously been discussed in detail and considered a parameter, as documents are often acquired from corpuses with distinct partitions. We find that the document length is indeed a parameter which should be taken into consideration when training due to its significant impact on the quality of topic models.

We also discuss how randomness within the model affects its outcomes when using different seeds. Again, this has not been discussed at length in the literature.

The hierarchical structure attainable through CorEx topic modelling proves effective at separating large topics into several smaller sub-topics. In Section 3.6, we demostrate this for the 'sport' topic within each television channel, clarifying, with a simple example, how this can be applied to real-world data.

As part of our development of a comprehensive bias measure, we seek to explore semi-supervised CorEx as a method for quantifying media attention surrounding a particular pre-chosen topic. We use the media attention to determine coverage bias which is the first part of the extensive measure for overall media bias.

Chapter 4 focusses on sentiment analysis and how it can be employed to determine statement bias in text. Within this chapter, we investigate the need for a domain-specific sentiment lexicon to accurately gauge the sentiment of text in particular fields. Our data contains a large proportion of news text which is more neutral and contains a vastly different distribution of terms to generic text. Thus, a sentiment lexicon trained on general text, or one built without the particular area in-mind, will struggle to perform well. We analyse the effectiveness of a domain-specific lexicon against a traditional lexicon. We use the Mittens [12] technique to fine-tune GloVe word embeddings [35] and create a new lexicon using methods from Dingwall and Potts [12]. We explore the effect of randomness within the embeddings and how this affects the sentiment lexicon, which has not been covered in the literature.

We expand on ideas from word shift graphs [15] to analyse sentiment lexicons. This allows us to easily compare the words with the greatest sentiment shifts, and the words with the greatest effect on the calculated sentiment of the text after a change in sentiment lexicon.

Finally, we extend the basic bias measure presented in Chapter 3 to one that also accounts for the sentiment of text. This new measure allows us to see both the prevalence of a political party in the media and whether their coverage is positive or negative. This is a novel approach for detecting bias in one television channel relative to other stations. This can be extended to other news formats such as print, radio, or social

media. It does, however, require a comparison to find a benchmark for neutrality and cannot be used as an explicit bias measure without context.

In summary, the contributions of this thesis are:

- To our knowledge, the first in-depth analysis of the Tveeder data set.
- A thorough derivation of the CorEx topic modelling framework with consistent notation, and discussion of its extensions. This improves accessibility of the method.
- Description of methods to predict topic probabilities from previously unseen documents.
- A novel technique for comparing two CorEx topic models which can be asymmetrically extended to topic models trained on different documents. We present cases for which this may be useful in a comparison of text.
- Three methods to determine the appropriate number of topics in a CorEx topic model. This increases the range of methods available to find the number of topics, allowing for a different choice depending on the requirements of the model.
- Discussion of the choice of document length in a corpus where this is ill-defined. We emphasize the importance of considering document lengths as an additional parameter in a topic model.
- Discussion of randomness within the CorEx topic model and its effect on results. We conclude that the CorEx method is heavily influenced by its random seed and discuss approaches to mitigate this.
- Methods exploiting the properties of CorEx to model media attention. We show these to correspond with major events and programs shown on each channel.
- Discussion of randomness within Mittens word embeddings and how this affects automatically calculated sentiment values.
- A method for analysing differences in sentiment lexicons.
- Methods to quantify both coverage and statement bias, and a single metric for media bias that combines these ideas.

Chapter 1

Background

This chapter presents a broad mathematical background for the thesis, beginning with an overview of information theory in Section 1.1. Section 1.2 introduces Correlation Explanation (CorEx) topic modelling which is the foundation of Chapter 3. This method of topic modelling serves as the foundation for a proposed coverage bias measure. In Section 1.3, we also discuss topic model similarity measures which are used to compare topic models.

Chapter 4 is concerned with sentiment analysis, which we introduce in Section 1.4. We discuss both manual and automatic methods to generate sentiment dictionaries. We present two specific methods, one of which requires the generation of high-quality word embeddings which we discuss in Sections 1.5 and 1.6.

Section 1.7 introduces three methods of analysing pairwise differences between texts. We exploit these techniques several times in Chapter 4 to contrast text from different channels, understand differences in word usage, and compare sentiment lexicons.

Section 1.8 details how we can estimate the significance of results from empirical data. This provides us with strong evidence to back up results where the distribution of data is unknown.

1.1 Information theory background

CorEx topic modelling is fundamentally based on information theory. Here we define estimates of uncertainty in random variables and the associations between them that describe their correlation. Definitions in this background were introduced by Shannon [44], Watanabe [59], and Han [20].

The *entropy* of a random variable measures the amount of uncertainty in its outcomes. A random variable with high entropy will have a high level of uncertainty in its outcomes and will therefore be difficult to predict.

Definition 1.1.1 (Entropy) *Let X be a discrete random variable with realisations $x \in \mathcal{X}$ distributed according to $p : \mathcal{X} \to [0, 1]$. The entropy of X is given by*

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x).$$

The choice of base for the logarithm varies across different fields. We use the natural logarithm.

The *joint entropy* extends this concept to a set of variables, measuring their joint uncertainty. For two random variables, X_1 and X_2, this is given by

$$H(X_1, X_2) = \sum_{x_1 \in \mathcal{X}_1} \sum_{x_2 \in \mathcal{X}_2} p(x_1, x_2) \log p(x_1, x_2),$$

where $p(x_1, x_2)$ is the joint probability of X_1 and X_2 occurring together. The joint entropy is defined to be 0 if $p(x_1, x_2) = 0$.

The *conditional entropy* is the amount of information needed to describe the outcome of one random variable after observing the outcome of a second random variable. The conditional entropy of X_1 given X_2 is

$$H(X_1|X_2) = -\sum_{x_1 \in \mathcal{X}_1} \sum_{x_2 \in \mathcal{X}_2} p(x_1, x_2) \log \left(\frac{p(x_1, x_2)}{p(x_2)} \right).$$

The *mutual information* between two random variables is an estimate of their mutual dependence. This is a way of quantifying the amount of information obtained about one random variable by observing another random variable.

Definition 1.1.2 (Mutual Information) *Let X_1 and X_2 be random variables with possible values $x_1 \in \mathcal{X}_1$ and $x_2 \in \mathcal{X}_2$ respectively. The mutual information between X_1 and X_2 is*

$$I(X_1 : X_2) = H(X_1) + H(X_2) - H(X_1, X_2).$$

Re-written in terms of probability, this definition is

$$I(X_1 : X_2) = \sum_{x_1 \in \mathcal{X}_1} \sum_{x_2 \in \mathcal{X}_2} p(x_1, x_2) \log \left(\frac{p(x_1, x_2)}{p(x_1)p(x_2)} \right). \tag{1.1}$$

Note that the mutual information of a random variable with itself is simply the entropy of that random variable.

The *total correlation* of a set of random variables is a generalisation of the mutual information. This quantifies the dependency among a set of random variables. We denote a set of random variables $\{X_1, X_2, \ldots, X_n\}$ as $\bar{X}$. This notation should not be confused with the sample mean.

Definition 1.1.3 (Total Correlation) *Let $\bar{X}$ be a set of n random variables, $\{X_1, X_2, \ldots, X_n\}$. The total correlation of $\bar{X}$ is given by*

$$TC(\bar{X}) = \sum_{i=1}^{n} H(X_i) - H(\bar{X}).$$

We clarify here that the probability of an observation $\bar{X} = \bar{x}$ is the joint probability of all X_i, and so $p(\bar{x}) = p(x_1, x_2, \ldots, x_n)$. The entropy of $\bar{X}$ is then the joint entropy of all X_i, that is $H(\bar{X}) = H(X_1, X_2, \ldots, X_n)$. Note that if $\bar{X}$ consists of just two elements, the total correlation is simply the mutual information of the two random variables X_1 and X_2. The total correlation can also be written in terms of the Kullback-Leibler divergence as

$$TC(\bar{X}) = D_{KL}\left(p(\bar{x}) \middle\| \prod_{i=1}^{n} p(x_i)\right),$$

where the Kullback-Leibler divergence of probability distributions P and Q is

$$D_{KL}\left(P \,\|\, Q\right) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right).$$

The Kullback-Leibler divergence is a measure of how different one probability distribution, P, is from a second distribution, Q. The total correlation is therefore a measure of how much the joint probability distribution differs from the product of the individual distributions. A low total correlation indicates that there is little information shared among variables, or that the joint probability of the random variables can be modelled well with an assumption of independence. A high total correlation indicates that a large amount of information can be learnt about one variable by observing the others. The maximum possible value occurs when one variable in $\bar{X}$ determines all other variables. In this case, the maximum total correlation is

$$TC_{\max} = \sum_{i=1}^{n} H(X_i) - \max_{X_i} H(X_i).$$

Conditioning on another random variable can provide the amount of information obtained about a set of random variables by observing just one other random variable.

Definition 1.1.4 (Conditional Total Correlation) *Let $\bar{X}$ be a set of n random variables $\{X_1, X_2, \ldots, X_n\}$. The total correlation of $\bar{X}$ conditioned on another random variable Y is*

$$TC(\bar{X}|Y) = \sum_{i=1}^{n} H(X_i|Y) - H(\bar{X}|Y).$$

Written in terms of the Kullback-Leibler divergence, this definition becomes

$$TC(\bar{X}|Y) = D_{KL}\left(p(\bar{x}|y) \,\middle\|\, \prod_{i=1}^{n} p(x_i|y)\right).$$

It is clear from its definition in terms of the Kullback-Leibler divergence that the conditional total correlation, $TC(\bar{X}|Y)$, is zero if and only if the distribution of $\bar{X}$'s conditioned on Y factorises.

The extent to which a random variable Y explains the correlations in a set of random variables $\bar{X}$ can be measured by analysing the amount by which the total correlation of $\bar{X}$ is reduced when conditioned on Y.

Definition 1.1.5 (Joint Total Correlation) *Let $\bar{X}$ be a set of n random variables $\{X_1, X_2, \ldots, X_n\}$. The joint total correlation of $\bar{X}$ and another random variable Y is*

$$TC(\bar{X}; Y) = TC(\bar{X}) - TC(\bar{X}|Y).$$

This definition can be re-written in terms of mutual information as

$$TC(\bar{X}; Y) = \sum_{i=1}^{n} I(X_i : Y) - I(\bar{X} : Y). \tag{1.2}$$

Note that unlike the mutual information, the joint total correlation is not symmetric. A semicolon is used in the notation as a reminder of this. If Y explains all of the correlation in $\bar{X}$, the joint total correlation, $TC(\bar{X}; Y)$, is maximised. We see in Section 1.2 that the total correlation plays an important role in CorEx topic modelling, as the quantity that we aim to maximise.

1.2 Correlation Explanation

We begin with a review of CorEx originally presented by Ver Steeg and Galstyan in 2014 [46]. The technique was futher developed by Ver Steeg and Galstyan in 2015 [47] to allow for hierarchical topic models, and by Reing et al. in 2016 [37] and Gallagher et al. in 2017 [16], to facilitate the anchoring of words to a topic. While other topic models such as LDA work by finding parameters for a generative probability distribution for words throughout the topics, CorEx instead aims to maximise the total correlation between the documents and topics by separating words into appropriate groups. This non-generative method reduces the need for many assumptions and limits the hyperparameters required. It also outperforms other topic modelling frameworks in several assessments [16, 46, 47].

Section 1.2.1 thoroughly derives the CorEx topic modelling approach with clear and consistent notation. In Section 1.2.2, we discuss how to implement the algorithm on

real data. Sections 1.2.3 and 1.2.4 introduce anchoring and hierarchical topic modelling which are used throughout Chapter 3 to gain an in-depth insight into the subtleties of the language used by each television station. Finally, Section 1.2.5 introduces a method for predicting topic probabilities for unseen documents.

1.2.1 Derivation of the CorEx framework

This subsection derives the method of CorEx topic modelling. We first introduce the concept of words, documents, and topics. We then derive the CorEx method by maximising the joint total correlation between documents and topics.

In the context of topic modelling, let X_i be a discrete random variable with observations of *words*, x_i. Let there be n total words in the corpus, and let $G \subseteq \{1, \ldots, n\}$ be a set of indices. We then define $\bar{X}_G$ as the multivariate random variable which is the subset of all X_i's corresponding with the indices G. An observation of this random variable is a list of words, which we will call a *document*.

We now introduce $\bar{X}$ as the random variable which can have any document as an outcome. The probability of the observation $\bar{X} = \bar{x}$ is written as $p(\bar{x})$. The proportion of documents containing word x_i is given by $p(x_i)$. Note that the probabilities $p(\bar{x})$ sum to 1, as $\bar{x}$ partitions the sample space. This is not true, however, for x_i, since the probability $p(x_i)$ is not the direct proportion of words which are of word type x_i.

Let Y be a discrete random variable with realisations y which we will call *topics*. The number of topics, denoted by m, is typically greater than 1 and is a fixed hyper-parameter which is determined before training. We aim to find m different topics, $Y_1, Y_2, \ldots, Y_m$, that explain the correlations in all of the documents. To do so, we separate all words into groups G_j each associated with topic Y_j. The set of word types in a group G_j is denoted by $\bar{X}_{G_j}$. Note that $\bar{X}_{G_j}$ is not a document but the set of word types associated with Y_j.

We pause here to introduce an example for clarity. Consider a sample of five small documents from our corpus:

> Get your Woolies worth with weekly specials like Australian-grown Croc Egg plums, just $4.90 a kilo at Woolworths.
>
> You don't see him put many catches down, Nathan Lyon. Got very good hands.
>
> Well inside Melbourne territory on the last tackle. Spiralling bomb. They let it bounce. Tuipulotu gets there. Try. Manly goes back-to-back.
>
> The Government's dedicated COVID response team is being disbanded tonight, amid Victoria's ongoing health crisis.
>
> Welcome back to the live grand finale of I'm A Celebrity... Get Me Out Of Here!

The words, X_i, include 'Woolworths', 'Nathan', and 'tackle'. The probability of these words is the number of documents containing each term. In the case of 'Woolworths',

this would be 0.2. A more common term such as 'the' has probability 0.6 in this example. Note that we remove capital letters and punctuation and thus 'The' would be treated the same as 'the'. Each of these documents has a probability of 0.2 of occurring.

The aim of topic modelling is to separate words into topics. Words commonly used in the same document are likely to be grouped together. In this case, the words 'Woolies' and 'Woolworths' would be grouped in the same topic, as would 'bomb' and 'Melbourne'. If we had more documents, 'bomb' and 'Melbourne' would likley be separated into different topics as they cooccur relatively infrequently. 'Woolworths' and 'Woolies' would likely remain in the same topic and may be joined by other advertising terms such as 'Coles'.

We now introduce the conditional probability of a topic given a word as

$$\begin{aligned} p(y|x_i) &= \frac{p(y, x_i)}{p(x_i)} \\ &= \sum_{\bar{x}} \frac{p(y|\bar{x})p(\bar{x})\delta_{\bar{x}_k,x_i}}{p(x_i)}, \end{aligned} \tag{1.3}$$

where $\delta_{\bar{x}_k,x_i}$ is an indicator variable which is 1 when at least one of the random variables in $\bar{x}$ is equal to x_i. Since $\bar{x}$ forms a partition of the sample space, from the law of total probablility, we also know that

$$\sum_{\bar{x}} p(\bar{x})p(y|\bar{x}) = p(y). \tag{1.4}$$

The CorEx model seeks to explain the dependencies of words in documents through latent topics by maximising the joint total correlation of all X_{G_j} and Y_j. To do so, we search over all G_j and Y_j that may explain the correlations in the group. We also impose the condition that the sets G_j are disjoint, which ensures that each word belongs to only one topic. The problem is posed as an optimisation problem; we search for the optimum

$$J = \max_{G_j, p(y_j|\bar{x}_{G_j})} \sum_{j=1}^{m} TC(\bar{X}_{G_j}; Y_j), \tag{1.5}$$

where $p(y_j|\bar{x}_{G_j})$ is the likelihood of seeing y_j given the words in $\bar{X}_{G_j}$. We are maximising over all G_j and $p(y_j|\bar{x}_{G_j})$, however write the maximum as above for simplicity.

In order to find a solution to this maximisation, we first rewrite (1.5) in terms of mutual information using (1.2). The optimisation then becomes

$$J = \max_{G_j, p(y_j|\bar{x}_{G_j})} \sum_{j=1}^{m} \left[\sum_{i \in G_j} I(X_i : Y_j) - I(\bar{X}_{G_j} : Y_j) \right]$$
$$= \max_{G_j, p(y_j|\bar{x}_{G_j})} \sum_{j=1}^{m} \sum_{i \in G_j} I(Y_j : X_i) - \sum_{j=1}^{m} I(Y_j : \bar{X}_{G_j}).$$

Now, replace G_j with a set indicator variable, $\alpha_{i,j} = \mathbb{I}[X_i \in \bar{X}_{G_j}]$. The variable $\alpha_{i,j}$ is equal to 1 if word X_i is in group $\bar{X}_{G_j}$, and 0 otherwise. We replace the G_j in the sum so that we are now taking the sum over all words. Only terms where the word X_i is in the group, however, are non-zero. The non-overlapping group constraint is enforced by demanding that $\sum_j \alpha_{i,j} = 1$ for all words (which are indexed by i). The objective is now

$$J = \max_{\alpha_{i,j}, p(y_j|\bar{x})} \sum_{j=1}^{m} \sum_{i=1}^{n} \alpha_{i,j} I(Y_j : X_i) - \sum_{j=1}^{m} I(Y_j : \bar{X}) \tag{1.6}$$

Note also that we have dropped the G_j from the $\bar{X}$ in the second term, where $\bar{X}$ alone is the random variable which can take on any possible set of words. Dropping the G_j has no effect since the solutions must satisfy $I(Y_j : \bar{X}) = I(Y_j : \bar{X}_{G_j})$ because the joint probability of Y_j and X_i, $p(y_j, x_i) = 0$, when $i \notin G_j$.

We rewrite the objective by expanding the mutual information using (1.1) and take just one instance of j within the summation. The objective is then

$$J = \max_{\alpha_i, p(y|\bar{x})} \sum_{i=1}^{n} \alpha_i \left[\sum_{x_i, y} p(x_i, y) \log \left(\frac{p(x_i, y)}{p(x_i)p(y)} \right) \right] - \sum_{\bar{x}, y} p(\bar{x}, y) \log \left(\frac{p(\bar{x}, y)}{p(\bar{x})p(y)} \right).$$

To clarify, the summation over x_i indicates a sum over the binary outcomes, x_i, and its complement, x_i^c. An obeservation of x_i implies we have seen the word, while an observation x_i^c indicates we have not. Expanding the joint probabilities as conditional probabilities and then expanding the logarithms, we have

$$
\begin{aligned}
J = \max_{\alpha_i, p(y|\bar{x})} & \sum_{i=1}^{n} \alpha_i \left[\sum_{x_i, y} p(x_i) p(y|x_i) \log \left(\frac{p(y|x_i) p(x_i)}{p(x_i) p(y)} \right) \right] \\
& - \sum_{\bar{x}, y} p(\bar{x}) p(y|\bar{x}) \log \left(\frac{p(y|\bar{x}) p(\bar{x})}{p(\bar{x}) p(y)} \right) \\
= \max_{\alpha_i, p(y|\bar{x})} & \sum_{i=1}^{n} \alpha_i \left[\sum_{x_i, y} p(x_i) p(y|x_i) \left(\log p(y|x_i) - \log p(y) \right) \right] \\
& - \sum_{\bar{x}, y} p(\bar{x}) p(y|\bar{x}) \left(\log p(y|\bar{x}) - \log p(y) \right).
\end{aligned}
$$

We now subsitute in the value of $p(y|x_i)$ from (1.3) to get

$$
\begin{aligned}
J = \max_{\alpha_i, p(y|\bar{x})} & \sum_{i=1}^{n} \alpha_i \left[\sum_{x_i, y} p(x_i) \left(\sum_{\bar{x}} \frac{p(y|\bar{x}) p(\bar{x}) \delta_{\bar{x}_k, x_i}}{p(x_i)} \right) \left(\log p(y|x_i) - \log p(y) \right) \right] \\
& - \sum_{\bar{x}, y} p(\bar{x}) p(y|\bar{x}) \left(\log p(y|\bar{x}) - \log p(y) \right) \\
= \max_{\alpha_i, p(y|\bar{x})} & \sum_{i=1}^{n} \alpha_i \sum_{\bar{x}, y} p(y|\bar{x}) p(\bar{x}) \left[\sum_{x_i} \delta_{\bar{x}_k, x_i} \left(\log p(y|x_i) - \log p(y) \right) \right] \\
& - \sum_{\bar{x}, y} p(\bar{x}) p(y|\bar{x}) \left(\log p(y|\bar{x}) - \log p(y) \right).
\end{aligned}
$$

The indicator variable $\delta_{\bar{x}_k, x_i}$ is 1 for exactly one of the x_i or x_i^c for each document. We can therefore remove the summation and write the objective as

$$
J = \max_{\alpha_i, p(y|\bar{x})} \sum_{\bar{x}, y} p(\bar{x}) p(y|\bar{x}) \left[\sum_{i=1}^{n} \alpha_i (\log p(y|x_i) - \log p(y)) - (\log p(y|\bar{x}) - \log p(y)) \right].
$$

We begin by solving for each α_i fixed. We introduce a Lagrange multiplier $\lambda_j(\bar{x})$ for each value of $\bar{x}$ and each j to enforce the normalisation constraint. We then optimise over $p(y|\bar{x})$. Note that for a fixed value of α_i, the optimisation for different j decouple and the overall solution to the maximisation is the sum of the solutions for each j. We have already dropped the j index for notational simplicity. The Lagrange function is

$$\mathcal{L} = \sum_{\bar{x},y} p(\bar{x})p(y|\bar{x}) \left[\sum_{i=1}^{n} \alpha_i (\log p(y|x_i) - \log p(y)) - (\log p(y|\bar{x}) - \log p(y)) \right] + \sum_{\bar{x}} \lambda(\bar{x}) \left(\sum_{y} p(y|\bar{x}) - 1 \right).$$

Expanding the sums and using (1.4), we get

$$\mathcal{L} = \sum_{i=1}^{n} \alpha_i \left[\sum_{\bar{x},y} p(\bar{x})p(y|\bar{x}) \log p(y|x_i) - \sum_{y} p(y) \log p(y) \right] - \left[\sum_{\bar{x},y} p(\bar{x})p(y|\bar{x}) \log p(y|\bar{x}) - \sum_{y} p(y) \log p(y) \right] + \sum_{\bar{x},y} \lambda(\bar{x}) \left(p(y|\bar{x}) - 1 \right).$$

We can take the partial derivative of (1.3) and (1.4) with respect to a particular $p(y|\bar{x})$, which we call $p(y^*|\bar{x}^*)$. We introduce an indicator variable δ_{y,y^*} which is 1 when $y = y^*$ and 0 otherwise. We also introduce similar variables for $\bar{x}$ and x_i. The partial derivatives are given by

$$\frac{\partial p(y|\bar{x})}{\partial p(y^*|\bar{x}^*)} = \delta_{y,y^*}\delta_{\bar{x},\bar{x}^*} \tag{1.7}$$

$$\frac{\partial p(y)}{\partial p(y^*|\bar{x}^*)} = \delta_{y,y^*}p(\bar{x}^*) \tag{1.8}$$

$$\frac{\partial p(y|x_i)}{\partial p(y^*|\bar{x}^*)} = \frac{\delta_{y,y^*}\delta_{x_i,x_i^*}p(\bar{x}^*)}{p(x_i^*)}. \tag{1.9}$$

Here, we only use the case where at least one of the random variables in $\bar{x}$ is equal to x_i. Recall that $\alpha_{i,j}$ is 1 precisely when word x_i is in the set of words $\bar{x}$, so the other case can be disregarded.

We differentiate the Lagrangian equation with respect to a particular $p(y|\bar{x})$, which we call $p(y^*|\bar{x}^*)$, and set the partial derivative equal to zero. Thus, the sums over $\bar{x}$ and y disappear, since the derivative with respect to $p(y|\bar{x})$ for any other term in the sum not containing the chosen $\bar{x}$ and y is zero. The optimisation reduces to equating the expression

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial p(y^*|\bar{x}^*)} =& \sum_{i=1}^{n} \alpha_i \Bigg[\sum_{\bar{x},y} p(\bar{x}) \frac{\partial p(y|\bar{x})}{\partial p(y^*|\bar{x}^*)} \log p(y|x_i) + \sum_{\bar{x},y} p(\bar{x}) p(y|\bar{x}) \frac{\partial \log p(y|x_i)}{\partial p(y^*|\bar{x}^*)} \\ & - \sum_{y} \frac{\partial p(y)}{\partial p(y^*|\bar{x}^*)} (\log p(y) + 1) \Bigg] \\ & - \left[\sum_{\bar{x},y} p(\bar{x}) \frac{\partial p(y|\bar{x})}{\partial p(y^*|\bar{x}^*)} (\log p(y|\bar{x}) + 1) - \sum_{y} \frac{\partial p(y)}{\partial p(y^*|\bar{x}^*)} (\log p(y) + 1) \right] \\ & + \sum_{\bar{x},y} \lambda(\bar{x}) \frac{\partial p(y|\bar{x})}{\partial p(y^*|\bar{x}^*)}\end{aligned}$$

to zero. Here we have made use of the differentiation rule

$$\frac{d}{dx}(f(x) \log f(x)) = \frac{df(x)}{dx}(\log f(x) + 1)$$

for all but the first term. By substituting in the partial derivatives from (1.7) to (1.9), we get

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial p(y^*|\bar{x}^*)} =& \sum_{i=1}^{n} \alpha_i \Bigg[\sum_{\bar{x},y} p(\bar{x}) \delta_{\bar{x},\bar{x}^*} \delta_{y,y^*} \log p(y|x_i) + \sum_{\bar{x},y} p(\bar{x}) p(y|\bar{x}) \frac{\delta_{y,y^*} \delta_{x_i,x_i^*} p(\bar{x}^*)}{p(y|x_i) p(x_i)} \\ & - \sum_{y} \delta_{y,y^*} p(\bar{x}^*) (\log p(y) + 1) \Bigg] \\ & - \left[\sum_{\bar{x},y} \delta_{\bar{x},\bar{x}^*} \delta_{y,y^*} p(\bar{x}^*) (\log p(y|\bar{x}) + 1) - \sum_{y} \delta_{y,y^*} p(\bar{x}^*) (\log p(y) + 1) \right] \\ & + \lambda(\bar{x})\end{aligned}$$

Then, by substituting the values of the indicator variables, using (1.4), and rearranging this becomes

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial p(y^*|\bar{x}^*)} &= \sum_{i=1}^{n} \alpha_i \Bigg[p(\bar{x}^*) \log p(y^*|x_i) + \sum_{\bar{x}} p(\bar{x}) p(y^*|\bar{x}) \frac{p(\bar{x}^*)}{p(y^*|x_i^*) p(x_i^*)} \\
&\qquad - p(\bar{x}^*)(\log p(y^*) + 1) \Bigg] \\
&\qquad - [p(\bar{x}^*)(\log p(y^*|\bar{x}) + 1) - p(\bar{x}^*)(\log p(y^*) + 1)] \\
&\qquad + \sum_{\bar{x},y} \delta_{y,y^*} \delta_{\bar{x},\bar{x}^*} \lambda(\bar{x}) \\
&= \sum_{i=1}^{n} \alpha_i \left[p(\bar{x}^*)(\log p(y^*|x_i) + 1) - p(\bar{x}^*)(\log p(y^*) + 1) \right] \\
&\qquad - [p(\bar{x}^*)(\log p(y^*|\bar{x}) + 1) - p(\bar{x}^*)(\log p(y^*) + 1)] + \lambda(\bar{x}^*) \\
&= p(\bar{x}^*) \Bigg[\sum_{i=1}^{n} \alpha_i \left((\log p(y^*|x_i) + 1) - (\log p(y^*) + 1) \right) \\
&\qquad - \left((\log p(y^*|\bar{x}^*) + 1) - (\log p(y^*) + 1) \right) \Bigg] + \lambda(\bar{x}^*) \\
&= p(\bar{x}^*) \left[\sum_{i=1}^{n} \alpha_i \left(\log p(y^*|x_i) - \log p(y^*) \right) - \left(\log p(y^*|\bar{x}^*) - \log p(y^*) \right) \right] \\
&\qquad + \lambda(\bar{x}^*).
\end{aligned}
$$

Equating this to zero, we have

$$
\log p(y^*|\bar{x}^*) = \sum_{i=1}^{n} \alpha_i \left(\log p(y^*|x_i) - \log p(y^*) \right) + \log p(y^*) + \lambda(\bar{x}^*).
$$

Applying the exponential to both sides and introducing a constant $Z(\bar{x})$, this becomes

$$
\begin{aligned}
p(y|\bar{x}) &= p(y) \prod_i \left(\frac{p(y|x_i)}{p(y)} \right)^{\alpha_i} e^{\lambda(\bar{x})} \\
&= \frac{p(y)}{Z(\bar{x})} \prod_i \left(\frac{p(y|x_i)}{p(y)} \right)^{\alpha_i}.
\end{aligned}
$$

Finally, by replacing the subscript on the y, we have

$$p(y_j|\bar{x}) = \frac{p(y_j)}{Z_j(\bar{x})} \prod_i \left(\frac{p(y_j|x_i)}{p(y_j)} \right)^{\alpha_{i,j}}. \tag{1.10}$$

The $Z_j(\bar{x})$ that we have introduced is a normalisation constant which can be found by taking the sum over the latent factor, Y_j, which is binary.

$$Z_j(\bar{x}) = \sum_{y_j \in \mathcal{Y}_j} p(y_j) \prod_i \left(\frac{p(y_j|x_i)}{p(y_j)} \right)^{\alpha_{i,j}} \tag{1.11}$$

The rule for $p(y_j|\bar{x})$ increases the value of the objective. We now simply iterate until we achieve convergence.

For fixed values of $p(y_j|x_i)$, the solution is $\alpha_{i,j} = \mathbb{I}[j = \text{argmax}_{\bar{j}} I(X_i : Y_{\bar{j}})]$. In other words, $\alpha_{i,j}$ is equal to 1 if Y_j is the topic which maximises the mutual information between X_i and $Y_{\bar{j}}$. This, however, leads to a rough optimisation space. Each iteration, $\alpha_{i,j}$ is instead updated according to the softmax function

$$\alpha_{i,j} = \exp\left(\lambda (I(X_i : Y_j) - \max_{\bar{j}} I(X_i : Y_{\bar{j}})) \right). \tag{1.12}$$

This update function removes the hard constraint and leads to a smooth and continuous optimisation.

1.2.2 Implementation

We now consider implementing CorEx on real-world data. It is impractical for $p(\bar{x})$ to be known for any set of words. We instead look only at N observed documents which we will denote by $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N$. We can use these to approximate the distributions $p(\bar{x})$ and $p(y|\bar{x})$ with $\hat{p}(\bar{x})$ and $\hat{p}(y|\bar{x})$ respectively.

The value of $\hat{p}(\bar{x})$ can be found from

$$\hat{p}(\bar{x}) = \sum_{l=1}^{N} \frac{\delta_{\bar{x}_l, \bar{x}}}{N}.$$

This is simply the proportion of documents matching $\bar{x}$. The value of $p(x_i)$ can also be approximated as

$$\hat{p}(x_i) = \sum_{l=1}^{N} \frac{\delta_{\bar{x}_l, x_i}}{N}.$$

This is the proportion of documents containing each word type.

To implement the algorithm, we begin with a list of documents containing a corpus of words and extract the binary counts of each word in each document. These counts form a sparse matrix of size $n \times N$ containing a 1 at index i, j if word i appears in document j, and a 0 otherwise.

Pseudo-code for implementing the CorEx topic modelling framework is presented in Algorithm 1.

Algorithm 1: CorEx Implementation

input : A binary matrix of size $n \times N$ representing the word counts in N documents

output: A matrix, α, of size $n \times m$
A matrix, $p(y|x_i)$, of size $n \times m$
A matrix, $p(y|\bar{x})$, of size $N \times m$
A vector, $p(y)$, of length m

Estimate $\hat{p}(x_i)$ and $\hat{p}(\bar{x})$;

Randomly initialise $\alpha_{i,j} \sim U(1/2, 1)$ and $p(y_j|\bar{x}) \sim U(0, 1)$;

repeat

  Estimate the marginal probabilities, $p(y_j)$ and $p(y_j|x_i)$ according to (1.3) and (1.4) respectively using $\hat{p}(\bar{x})$ in place of $p(\bar{x})$;

  Calculate $Z_j(\bar{x}_l)$ using (1.11);

  Update $\alpha_{i,j}$ using (1.12);

  Update $\hat{p}(y_j|\bar{x})$ according to (1.10);

until *convergence*;

1.2.3 Anchoring

This subsection introduces *anchoring*, a method of ensuring that chosen words appear in the topic model and are *anchored* to a particular topic.

We begin with an overview of the *information bottleneck*. This is a method of formulating a balance between compression of data $\bar{X}$ into a representation Y_j, and preserving information in $\bar{X}$ that is relevant to X_i [53]. We would like our topics Y_j to compress documents $\bar{X}$ as much as possible, however we also desire the topics to capture as much information about individual words X_i as possible. The information bottleneck is generally expressed as a minimisation problem, however, by changing this to a maximisation problem and using the notation introduced in this thesis, we have

$$\max_{p(y_j|\bar{x})} \beta I(X_i : Y_j) - I(\bar{X} : Y_j).$$

The parameter β controls the trade-off between compression and preservation. A higher value of β puts a greater emphasis on the mutual information between words and topics.

Compare this maximisation with (1.6). The optimisation problems are almost identical. We are already invoking the information bottleneck, but to make full use of its potential, we can set $\alpha_{i,j} = \beta$ where $\beta \geq 1$ controls the anchor strength. Recall that $\alpha_{i,j} \in [0, 1]$. A value of $\beta \geq 1$ ensures that words in a topic not only need to be consistent with each other, but must also be consistent with the chosen anchor word. When anchors are used, β is not updated each iteration but instead remains fixed at the value input by the user.

Anchoring allows for multiple words to be anchored to a single topic, or for one word to be anchored to multiple topics since we have removed the constraint that $0 \leq \alpha_{i,j} \leq 1$.

Anchoring in CorEx topic modelling has several uses [16]:

- **Topic separability**: This allows larger topics to be split into subtopics or to separate co-occurring words if desired. This is done by anchoring words that may appear in the same topic to different topics resulting in their separation. For example, a large political topic may be split into smaller Liberal or Labor topics by anchoring the terms 'liberal' and 'labor' to different topics.

- **Topic representation**: Anchoring can be used to obtain more subtle or underrepresented topics that may not otherwise emerge. This is particularly useful when there is prior domain knowledge and we wish to investigate a topic that doesn't appear in the original unsupervised model. For example, to examine the media attention of a known event, the funeral of Queen Elizabeth II, we must anchor the term 'queen', since this does not appear in an unsupervised model.

- **Topic aspects**: We can find multiple topics that revolve around a single word. By anchoring the same word to several different topics, this can bring about different contexts in which the word is spoken about to increase understanding of how the subject has been used. For example, a term with multiple aspects such as 'sport' may be anchored to multiple topics to create smaller topics each centering around a different sport.

We use anchoring for topic representation, to ensure that terms that we would like to investigate appear in a topic model. This allows us to model the media attention of any topic, not just those which appear in an unsupervised model. Also note that anchoring for topic separability and topic aspects is similar to the hierarchical structure that we discuss in Section 1.2.4 although it cannot be used to create a hierarchical model with more than two layers. We also find that it does not perform as well as the hierarchical structure at separating larger topics.

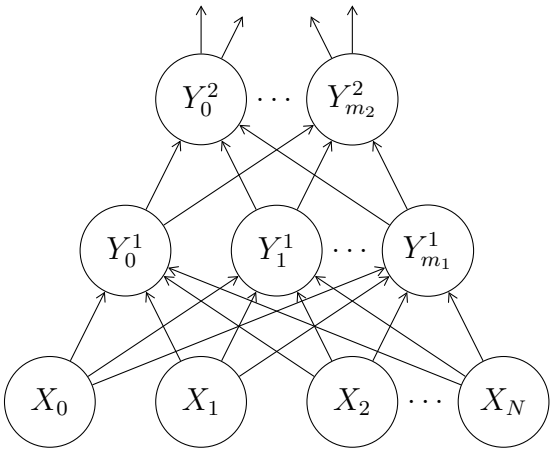

Figure 1.1: A simple visualisation of the hierarchical structure that is possible with CorEx.

1.2.4 Hierarchical Topic Modelling

To add further structure to the topic model, we can include hierarchical layers [37, 47]. This allows us to separate larger overarching topics into several smaller sub-topics.

To do so, we begin with a single topic model with m_1 topics. The value of m_1 is generally larger than that which would be found in Section 3.2.1, as the aim of this is to separate larger topics into much smaller but broader topics.

One output of the CorEx topic model is an $N \times m_1$ matrix of $p(y_j^1|\bar{x}_l)$ terms for each topic, y_j^1, and document, $\bar{x}_l$. Here we have denoted the first layer of topics with a superscript 1. The outputs can be easily converted to binary values by

$$f(p(y_j^1|\bar{x}_l)) = \begin{cases} 1 & \text{if } p(y_j^1|\bar{x}_l) > 0.5 \\ 0 & \text{otherwise.} \end{cases}$$

We can feed this binary matrix back into a second topic model with m_2 topics. The number of topics, m_2, should be less than m_1 so that we are able to form groups of several topics. We then estimate probabilities of the second layer of topics, $p(y_j^2)$, by taking the sum of each column of the matrix and dividing by n. This replaces the $\hat{p}(x_i)$ terms.

Instead of the second layer of the topic model grouping words into topics, it now groups topics into larger topics. We can also think of these as fewer larger topics each

containing more words. This structure can be repeated and several layers of topics and sub-topics can be found found.

This technique is useful for collecting similar topics. We implement this approach to collect different sports into one overarching 'sport' topic in Section 3.5. A hierarchical topic model with two layers allows us to quantitatively acknowledge the similarity of these topics and consider them as one if desired.

1.2.5 Prediction

Once we have a trained topic model, an obvious next step is to use that model to infer the topics of new, previously unseen documents. After training the topic model, we have the probabilities of the topics given each document in the training set, $p(y_j|\bar{x}_l)$. From here, we can calculate the probabilities of each topic given each word, $p(y_j|x_i)$, as well as the probabilities of the topics themselves, $p(y_j)$, using (1.3) and (1.4) respectively. We then calculate $p(y_j|\bar{x}_l)$ for the new document using (1.10). If a document contains unseen words, these are not included in the count matrix; they are simply ignored and have no effect on the calculated probability.

This method allows us to predict the probabilities of topics in unseen documents with reasonable accuracy. This can be useful if a new document is introduced after the topic model was trained or to infer probabilities if the model is only trained on a subsample of the data. We explore methods to construct appropriate training sets for prediction in Section 1.2.5.

1.3 Topic model similarity

Here we introduce a novel technique to compare the similarity of CorEx topic models. This technique is derived from $p(y|\bar{x})$, the primary use of the topic models in this thesis. We cannot compare these terms using a Kullback-Leibler divergence, since they do not form a valid probability distribution across topics. We thus treat these as ordinary values rather than probabilities and compare them with *Pearson correlation*,

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}.$$

The Pearson correlation measures the linear relationship between two variables. If the $p(y|\bar{x})$ terms from two topic models are similar, they will have a Pearson correlation close to 1.

We have two topic models, T_1 and T_2, with m topics which we denote as y_{1i} and y_{2j} for $1 \leq i, j \leq m$. The topic models T_1 and T_2 have probability functions p_1 and p_2 respectively. The *Pearson similarity* between the two topics is

$$PS_{T_1,T_2} = \frac{1}{m}\sum_i \sum_j \left[\delta_{y_{1i},y_{2j}} \left(\frac{\text{cov}(p_1(y_{1i}|\bar{x}), p_2(y_{2j}|\bar{x}))}{\sigma_{p_1(y_{1i}|\bar{x})}\sigma_{p_2(y_{2j}|\bar{x})}} \right) \right], \tag{1.13}$$

where $\delta_{y_{1i},y_{2j}}$ is the indicator function,

$$\delta_{y_{1i},y_{2j}} = \begin{cases} 1 & \text{if } |w_{y_{1i}} \cap w_{y_{2j}}| \geq k, \\ 0 & \text{otherwise.} \end{cases}$$

Here, $w_{y_{1i}}$ and $w_{y_{2j}}$ represent the set of words belonging to topics y_{1i} and y_{2j} respectively. The threshold value k can be adjusted according to the desired level of similarity. In this thesis, we set $k = 5$ so that 'similar' topics are those that share at least 5 words. The covariance and standard deviation in (1.13) are approximated by the sample covariance and sample standard deviation of the observations $p(y_{1_i}|\bar{x})$ and $p(y_{2_j}|\bar{x})$.

We include the indicator function rather than a plain weighted average to ensure that we are comparing just the similarity between topic models, and not within. If, for example, we had compared identical topic models using this method without the indicator function, the similarity score would not be perfect as there is variation between topics. Topic models with greater variation between topics would score lower on this measure, while topic models with similar topics would score higher. This could be used to compare the similarity of topics *within* a topic model, however we include the indicator function to compare just the similarities *between* topic models. Comparing identical topic models with this model will then yield a value of 1.

This method measures the similarity of the $p(y|\bar{x})$ terms of like topics. Due to the nature of our analysis, the $p(y|\bar{x})$ term is what we are primarily interested in and gives a reasonable estimate for the similarity of topics. Note that we must compare values across the same documents. We can predict values of $p(y|\bar{x})$ using the method detailed in Section 1.2.5 if the two topic models are trained on different documents.

This similarity measure is useful when wanting to compare two topic models and quantify their similarity. Although similarity measures have been introduced for LDA, these rely on the probability distributions of the words, and hence cannot be used to compare CorEx models. This measure is a reasonable way to make these comparisons on CorEx topic models.

Most topic models, even those run with the same parameters and data, will have slight variation. A good similarity measure is one above 0.8, since this denotes a reasonable Pearson correlation.

Since we are just comparing like topics, the Pearson correlation is unlikely to be negative. We will, however, let zero be the minimum value and any lower values we will treat as zero.

A similar method can be implemented by instead using the *Spearman correlation* to compare the order of the words in a given topic. The *Spearman similarity* is given by

$$SS_{T_1,T_2} = \frac{1}{m} \sum_i \sum_j \delta_{y_{1_i},y_{2_j}} \left(\rho_{R(w_{y_{1_i}}),R(w_{y_{2_j}})} \right). \tag{1.14}$$

The ranks, $R(w_{y_{1_i}})$ and $R(w_{y_{2_j}})$, indicate the order of words in a topic ranked by mutual information. The word with the highest mutual information will have a rank of 1, and this increases with each word. The Spearman correlation is then the Pearson correlation of the ranks, $\rho_{R(w_{y_{1_i}}),R(w_{y_{2_j}})}$. This similarity is useful when we are more concerned with the words contained in the topics and less about their performance in an application context. Note that this measure does not require the two topic models to be trained on the same documents, or for values of $p(y|\bar{x})$ to be predicted.

1.4 Sentiment analysis

Sentiment analyses are used to understand the positivity of text using automatic methods. They are commonly used for the analysis of online expression to improve advertisement relevance, or to classify reviews [28, 57]. We will make use of a sentiment analysis to determine the positivity of text. This will aid us in developing a statement bias measure that considers the sentiment of text surrounding a topic.

This thesis focusses on dictionary-based sentiment analyses which make use of a dictionary of words corresponding with sentiment scores. Both methods have strengths and weaknesses. In Chapter 4, we aim to compare these methods on our data set and discuss the relative strengths of each for assessing the sentiment of television news text and consequently the statement bias within. Two methods of creating sentiment dictionaries, or lexicons, are generally used: manual annotation (supervised), and automatic methods (semi-supervised or unsupervised). In a manual annotation, respondents may be shown a range of words and asked to rate them on a positivity scale. A mix of words is generally given and the respondent asked to choose the most/least positive from the word list. Automatic methods aim to generate a comprehensive positivity lexicon automatically using techniques such as machine learning. While manually annotated lexicons provide high precision, they lack the desired scope due to both time and financial constraints. Automatic methods to develop sentiment lexicons sacrifice precision for a wider coverage of words [17]. Both methods generally perform similarly when tested [10]. Many sentiment lexicons are readily available [4, 23, 32, 67], however these are very generic and many not be applicable to highly specific data sets [10].

Section 1.4.1 introduces dictionary-based sentiment analyses. Sections 1.4.2 and 1.4.3 expand on this by discussing both manual and automatic sentiment analysis methods in detail. Section 1.4.2 includes details on the NRC-VAD lexicon [32], a widely used

lexicon with high coverage. Section 1.4.3 discusses the method introduced by Cochrane et al. [10] to automatically generate a sentiment lexicon with word embeddings. We discuss these particular methods in detail as they form the basis for the sentiment analysis approaches we will employ throughout this thesis.

1.4.1 Dictionary-based sentiment analysis

Many sentiment analysis approaches make use of a sentiment lexicon: a list of features, such as words and punctuation, which are labelled according to their sentiment [23]. Generally, the sentiment is given by a score, such as a rating from 0 to 1 [32]. These ratings are often crowd-sourced to receive true human ratings. Several lexicons have also been developed using automatic methods. This is usually done in order to create domain-specific lexicons which are more accurate at analysing text from the domain on which they have been trained.

Once a lexicon has been created, it is used to calculate the sentiment of a text, often with a simple linear equation. The sentiment of a document is

$$S = \frac{1}{n} \sum_{i=1}^{n} s_i, \tag{1.15}$$

where n is the number of features in a document, and s_i is the sentiment of the i-th feature in the document.

1.4.2 Manual annotation: NRC-VAD

The NRC-VAD lexicon [32] is one of the most comprehensive manually generated sentiment lexicons, with over 20,000 sentiment ratings, including terms such as *happee*, *#lonely*, and *loveumom* which appear relatively commonly on Twitter. Note that although the lexicon has a very high coverage, many of the terms are unique to social media and do not appear in our data set. The 20,007 terms in the lexicon were selected from the following sources:

- the 14,182 words in the NRC Emotion Lexicon [33]
- the 4,206 terms in the General Enquirer [18]
- the 1,061 terms in Affective Norms for English Words [8]
- the 13,915 terms in the lexicon by Warriner et al. [58]
- the 520 terms from Roget's Thesaurus corresponding to the eight basic Plutchik emotions [36][38]
- the 11,418 high-frequency content terms (including emoticons) from the Hashtag Emotion Corpus [31].

The NRC-VAD lexicon contains scores in three dimensions:

- **valence:** positivity-negativity/pleasure-displeasure
- **arousal:** active-passive
- **dominance:** dominant-submissive.

The dimension that most corresponds with traditional sentiment ratings that measure the positivity of a feature is valence. This is the dimension that we will be using to indicate sentiment.

The lexicon was created using best-worst scaling which Mohammad [32] has shown provides more reliable results than previous VAD lexicons. Annotators are given four words at a time and asked to rate the words with the highest and lowest of a particular dimension. The questions used to obtain ratings for valence are shown below.

> *Q1. Which of the four words below is associated with the MOST happiness/pleasure/positiveness/satisfaction/contentedness/hopefulness OR LEAST unhappiness/annoyance/negativeness/dissatisfaction/ melancholy/despair? (Four words listed as options.)*
>
> *Q2. Which of the four words below is associated with the LEAST happiness/pleasure/positiveness/satisfaction/contentedness/hopefulness OR MOST unhappiness/annoyance/negativeness/dissatisfaction/ melancholy/despair? (Four words listed as options.)*

The results from the best-worst scaling are then converted into values between 0 and 1, where 0 indicates a high value of the dimension, and 1 indicates a low value of the dimension. We will be using the NRC-VAD lexicon to perform sentiment analyses due to its extensive coverage and high-quality.

1.4.3 Automatic method: word embeddings

Cochrane et al. [10] use the word2vec algorithm [29][30] to automatically generate a sentiment lexicon which is specific to political text. They make use of the semantic relations between word vectors found by Mikolov et al. [29][30] to create a sentiment for any word in their corpus. The nature of the semantic relations mean that a positive word should be clustered closer to a set of other positive words, while words with negative sentiment should be located near other negative words. Cochrane et al. [10] use cosine similarity as their measure of 'closeness', and take the difference between the cosine similarity of each word to a positive word set, and each word to a negative word set. The sentiment, s, of word w is then given by

$$s(\vec{w}) = \sum_{p=1}^{a} \frac{\vec{w} \cdot \vec{v}_p}{\|\vec{w}\|\|\vec{v}_p\|} - \sum_{q=1}^{b} \frac{\vec{w} \cdot \vec{u}_q}{\|\vec{w}\|\|\vec{u}_q\|}, \tag{1.16}$$

where $\vec{w}$ is the embedding of the word w, and the vectors $\vec{v}_p$ and $\vec{u}_q$ are the word embeddings of the terms in the positive and negative word sets respectively.

Since word embeddings are based on co-occurrences from a bag-of-words model, it is often found that words such as 'good' may be situated close to antonyms that are used in the same context. Like Turney and Pantel [56], and Cochrane et al. [10], we rely on seed-word pairs to resolve this issue. By identifying words that should theoretically be opposite on our dimension of interest, the displacement vector between the pair of antonyms measures the difference between two points on that dimension.

Using just a single pair of seed-words will not be able to perfectly capture the sentiment due to corpus noise. Cochrane et al. [10] have shown that using seven pairs of opposite seed words outperforms most other dictionary-based sentiment analysis models and performs much better than using any of the pairs on their own. We will therefore use their list of semantically opposite words which is taken from Turney and Litman [55]. These words are:

good	bad
excellent	terrible
correct	wrong
best	worst
happy	disappointed
positive	negative
fortunate	unfortunate

Any number of words from the corpus can be assigned a sentiment using this method. Generally, only words which appear a certain number of times are used to ensure that the quality of the word embeddings is reasonable. Cochrane et al. [10] only give sentiments to words that appear a minimum of ten times.

Using this method, the sentiment values for thousands of words can be calculated in a matter of minutes. This greatly reduces both the time and resources required to produce a lexicon, while increasing the coverage. Although the quality of the ratings may be somewhat reduced, this is a fair compromise for the amount of coverage that can be achieved.

1.5 GloVe word embeddings

The automatic method for generating a sentiment lexicon presented in Section 1.4.3 requires high-quality word embeddings trained on our text. This is both computationally expensive and requires a vast amount of data. To ensure that we obtain high-quality embeddings, we first begin with pre-trained Global Vector (GloVe) embeddings.

These embeddings were introduced in 2014 by Pennington et al. [35]. They propose a specific weighted least squared model that trains on global word-word co-occurrence

counts. This section derives the cost function used to calculate GloVe embeddings.

Pennington et al. [35] give an example showcasing how meaning can be extracted from co-occurrence probabilities, $P_{ij} = p(j|i)$, the probability that word j appears in the context of word i. We reproduce this example with commonly occurring terms in our corpus.

Take two words from our corpus, $i =$ *football*, and $j =$ *cycling*. The relationship between these two words can be analaysed by examining the ratio of their co-occurrence probabilities with a third probe word. For words related to *football* but not *cycling*, such as $k =$ *ball*, we expect the ratio P_{ik}/P_{jk} to be large. Similarly, for words related to *cycling* but not *football*, for example $k =$ *bicycle*, the ratio should be small. For words related to both such as $k =$ *sport*, or related to neither such as $k =$ *cake*, the ratio should be close to one.

Thus, it is concluded that an appropriate starting point for learning word embeddings should be with the ratio of co-occurrence probabilities. From this point, Pennington et al. [35] derive a cost function, J, and cast this as a least square regression problem. For the full derivation, see their paper. A weighting function $f(X_{ij})$ is included in the model to ensure that rare co-occurrences are weighted lower. The bias terms, b_i and $\tilde{b}_j$, absorb the overall occurrences of i and j so that the relative abundance of a word within the corpus will not affect its embedding. The cost function is

$$J = \sum_{i=1}^{n} \sum_{j=1}^{n} f\left(X_{ij}\right) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{ij}) \right)^2, \tag{1.17}$$

where w_i and $\tilde{w}_j$ are word vectors representing word i and context word j respectively, and X_{ij} is the number of times i and j co-occur. There are n unique words.

Pennington et al. [35] recommend the function

$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases} \tag{1.18}$$

to use as the weighting function. During experiments, they find that $x_{\max} = 100$ and $\alpha = 3/4$ give the best results. We use the same values in this thesis.

Finally, the word embeddings are found by minimizing the cost function J with an adaptive gradient descent.

1.6 Mittens: an extension of GloVe

We may wish to obtain domain-specific word embeddings rather than the generic embeddings from GloVe. Dingwall and Potts [12] extend the GloVe algorithm with a

method which they call *Mittens*. This extension allows for the input of a set of existing vectors which give the optimisation a 'warm start'. The vectors can then be trained with a cost for any deviation away from the warm start. This section presents the new cost function and how Mittens can be implemented on real-world data.

Dingwall and Potts begin by fixing a shortcoming in the GloVe cost function: that $\log(X_{ij})$ is only defined for $X_{ij} > 0$. They propose a new function,

$$g(X_{ij}) = \begin{cases} k & \text{for } X_{ij} = 0 \\ \log(X_{ij}), & \text{otherwise,} \end{cases}$$

for any k. This function replaces $\log(X_{ij})$ and is defined at $X_{ij} = 0$. Since $f(0) = 0$, this does not change the value of the objective. They introduce

$$M = W^T\tilde{W} + b\mathbf{1}^T + \mathbf{1}\tilde{b}^T - g(X), \tag{1.19}$$

where W and $\tilde{W}$ are matrices whose columns contain the word and context embeddings respectively. In this case, g is applied elementwise, resulting in a vector with the number of rows equal to the dimension of the word embeddings. The cost function is then

$$J = \sum_{i=1}^{n}\sum_{j=1}^{n} f(X_{ij})(M_{i,j})^2.$$

Vectorising the GloVe cost function has enormous impacts on the speed of the implementation and makes it apparent that GloVe can be extended to a retrofitting model. In order to retrofit, we simply add a term to the objective that penalises the Euclidean distance from the learned embedding $\hat{w}_i = w_i + \tilde{w}_i$ to an existing one, r_i. The new objective function is

$$J_{\text{mittens}} = J + \mu\sum_{i=1}^{k}\|\hat{w}_i - r_i\|^2. \tag{1.20}$$

The k existing embeddings, r_i, are words in the new vocabulary for which pre-trained embeddings are available. The parameter μ acts as a weighting term, where higher values of μ give greater weight to existing embeddings. When $\mu = 0$ or when there are no original embeddings, the objective reduces to the original GloVe cost function.

1.6.1 Implementing mittens for fine-tuning GloVe

In this section, we suppose that a generic word embedding is available from GloVe or elsewhere. We use (1.20) to fine-tune that word embedding to a particular corpus.

We begin fine-tuning the GloVe embeddings by initialising the new vectors. Words with a prior embedding are initialised as this embedding. Words that do not have GloVe embeddings are given a randomly initialised vector with values sampled from the uniform distribution $U(-\sqrt{6/(n+d)}, \sqrt{6/(n+d)})$, where there are n unique words (and therefore n vectors), and each vector is d-dimensional.

The word embeddings are separated into a word component, w_i, and a context component, $\tilde{w}_i$. Some noise is added to break the symmetry and ensure that gradient updates are not identical for the two. This separation is

$$\begin{aligned} w_i &= 0.5r_i + \epsilon_{i,1} \\ \tilde{w}_i &= 0.5r_i + \epsilon_{i,2}, \end{aligned}$$

where $\epsilon_{i,j} \sim N(0, 0.01)$. The biases for each component of the vector, b_i and $\tilde{b}_i$ are also initialised according to the uniform distribution $U(-\sqrt{6/(1+n)}, \sqrt{6/(1+n)})$.

Written as a summation, the cost function for the original GloVe embeddings is then

$$J = \sum_{i=1}^{n} \sum_{j=1}^{n} 0.5 f(X_{ij}) (M_{i,j})^2.$$

The 0.5 term is introduced to account for the two separate components of each vector. We use $W = [w_1, w_2, \ldots, w_m]$ and $\tilde{W} = [\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_m]$ to calculate M according to (1.19). If $\mu = 0$, we calculate the gradients

$$\begin{aligned} \nabla w_i &= \left(\sum_{j=1}^{m} f(X_{ij}) M_{i,j} \right) \cdot \tilde{w}_j \\ \nabla \tilde{w}_j &= \left(\sum_{i=1}^{m} f(X_{ij}) M_{i,j} \right) \cdot w_i \\ \nabla b_i &= \sum_{j=1}^{m} f(X_{ij}) M_{i,j} \\ \nabla \tilde{b}_j &= \sum_{i=1}^{m} f(X_{ij}) M_{i,j} \end{aligned}$$

and perform a gradient descent.

If $\mu > 0$, we add to ∇w_i and $\nabla \tilde{w}_j$ for words with existing embeddings. These become

$$\nabla w_i = \left(\sum_{j=1}^{m} f(X_{ij}) M_{i,j} \right) \cdot \tilde{w}_j + 2\mu(w_i + \tilde{w}_i - r_i)$$
$$\nabla \tilde{w}_j = \left(\sum_{i=1}^{m} f(X_{ij}) M_{i,j} \right) \cdot w_i + 2\mu(w_i + \tilde{w}_i - r_i),$$

noting that ∇b_i and $\nabla \tilde{b}_j$ are identical to the $\mu = 0$ case. Finally, we optimise using a gradient descent with cost function given in (1.20). Values are updated each iteration in the direction of $-\nabla w_i$ and $\tilde{w}_j$ respectively according to

$$w_i^{(t+1)} = w_i^{(t)} - \nabla w_i^{(t)} \times \left(0.05 / \sqrt{\beta^{(t)}} \right)$$
$$\tilde{w}_j^{(t+1)} = \tilde{w}_j^{(t)} - \nabla \tilde{w}_j^{(t)} \times \left(0.05 / \sqrt{\beta^{(t)}} \right),$$

where $\beta^{(t)} = \beta^{(t-1)} + \left(\nabla w_i^{(t)} \right)^2$ and $\beta^{(t=0)} = 0.1$. Optimising until convergence generates vectors fine-tuned on a domain-specific corpus. This is useful for the generation of quality domain-specific word embeddings with a limited corpus size [12]. We use embeddings generated this way in Section 4.4 to develop a new sentiment lexicon specifically for news text.

1.7 Word shifts

Word shift analysis [15] provides a useful way to investigate pairwise differences between texts. Sections 1.7.1 to 1.7.3 present three methods for quantifying pairwise differences: relative frequency, Tsallis entropy, and dictionary-based scores. Section 1.7.4 extends the dictionary-based comparison to compare lexicons rather than the text.

The analysis consists of two texts, $T^{(1)}$ and $T^{(2)}$, each with a vocabulary $\mathcal{T}^{(i)}$. We denote a word type by τ which occurs with some frequency $f_\tau^{(i)}$ in each of the texts. Note that either $f_\tau^{(1)}$ or $f_\tau^{(2)}$ may be zero. Each word's normalised relative frequency is $p_\tau^{(i)} = f_\tau^{(i)} / \sum_{\tau' \in \mathcal{T}} f_{\tau'}^{(i)}$.

1.7.1 Relative frequency

One of the simplest ways of analysing the difference between texts is by looking at the raw difference in word usage between the two. The *proportion word shift* shows a ranking of the difference in relative frequencies of words between texts calculated by

$$p_\tau^{(2)} - p_\tau^{(1)}.$$

Positive terms are those which are relatively more common in $T^{(2)}$, while negative terms are relatively more common in $T^{(1)}$. A problem with this method is that the highest ranked terms are often common words that have high probabilities. This is unlikely to provide useful or interesting results that would provide clues as to what characterises the particular texts.

1.7.2 Entropy

Recall from Section 1.1 that the entropy is a measure of the surprise of a random variable, accounting for both a word's frequency and its uncertainty. The *Tsallis entropy* is a generalization of entropy in which we can emphasise common or rare words by altering a parameter $\alpha > 0$ [1, 15, 21].

Definition 1.7.1 (Tsallis Entropy) *Given a probability distribution P with probabilities p_i, and any positive real number α, the Tsallis entropy is given by*

$$S_\alpha(P) = \frac{1}{1-\alpha}\left(1 - \sum_i p_i^\alpha\right).$$

Consider the difference in entropy between the word-frequency distributions, $P^{(1)}$ and $P^{(2)}$,

$$\begin{aligned}\delta S_\alpha &= S_\alpha\left(P^{(1)}\right) - S_\alpha\left(P^{(2)}\right) \\ &= \frac{\sum_i \left(p_i^{(1)}\right)^\alpha - \sum_i \left(p_i^{(2)}\right)^\alpha}{\alpha - 1}.\end{aligned}$$

Now consider the contribution from each word τ,

$$\delta S_{\alpha\tau} = \frac{\left(p_i^{(1)}\right)^\alpha - \left(p_i^{(2)}\right)^\alpha}{\alpha - 1}. \tag{1.21}$$

In the limit as $\alpha \to 1$ with $k = 1$, common and uncommon words are weighted equally and the Tsallis entropy simplifies to the Shannon entropy. When $\alpha > 1$, common words are weighted more heavily, and when $\alpha < 1$, uncommon words are weighted more heavily. Decreasing α allows us to investigate rare words which generally contain more information about the text. A *Tsallis entropy word shift* represents these differences in entropy contributions from each word.

1.7.3 Dictionary-based scores

Using a similar method, we can also analyse the cause of a difference in sentiment between two texts [15]. Given a dictionary that gives a score ϕ_τ to each word τ in the vocabulary, the *weighted average word shift* shows the difference between word probabilities weighted by their sentiment. This is given by

$$\delta\Phi = \sum_{\tau\in\mathcal{T}} \phi_\tau \left(p_\tau^{(2)} - p_\tau^{(1)}\right).$$

If a word does not appear in the dictionary, it can be ignored for the purposes of this particular analysis.

To determine the causes behind the difference in sentiment of one text relative to another, we introduce a refrence score, $\Phi^{(\text{ref})}$. We can then determine whether a word is considered positive or negative relative to this reference score. The reference score can take any value, but is often the mean of the sentiment values in the dictionary or set to a known neutral value such as 0. The sum of contributions is then

$$\delta\Phi = \sum_{\tau\in\mathcal{T}} \left(\phi_\tau - \Phi^{(\text{ref})}\right)\left(p_\tau^{(2)} - p_\tau^{(1)}\right). \tag{1.22}$$

The equation for each word τ can be split into two components, giving us four ways that the word can affect the sentiment of the text.

$$\delta\Phi_\tau = \underbrace{\left(\phi_\tau - \Phi^{(\text{ref})}\right)}_{\uparrow/\downarrow}\underbrace{\left(p_\tau^{(2)} - p_\tau^{(1)}\right)}_{+/-}.$$

The ways that $T^{(2)}$ can have a higher score than $T^{(1)}$ are:

1. A relatively positive word $(+)$ is used more often $(\uparrow)$ in $T^{(2)}$ than in $T^{(1)}$
2. A relatively negative word $(-)$ is used less often $(\downarrow)$ in $T^{(2)}$ than in $T^{(1)}$

We define a 'relatively positive' score as one where $\phi_\tau > \Phi^{(\text{ref})}$, and a 'relatively negative' score as one where $\phi_\tau < \Phi^{(\text{ref})}$. The ways that $T^{(1)}$ can have a higher score than $T^{(2)}$ are:

1. A relatively positive word $(+)$ is used less often $(\downarrow)$ in $T^{(2)}$ than in $T^{(1)}$
2. A relatively negative word $(-)$ is used more often $(\uparrow)$ in $T^{(2)}$ than in $T^{(1)}$

This analysis allows us to compare the terms which contribute the most to differences in sentiment between texts.

1.7.4 Comparison of dictionaries

These word shifts can also be used to compare the impact that different lexicons have on the calculated sentiment of a text. We now have two scores, $\phi_\tau^{(1)}$ and $\phi_\tau^{(2)}$, for each word depending on which sentiment lexicon is used.

The difference in weighted averages is

$$\delta\Phi = \sum_{\tau\in\mathcal{T}} \phi_\tau^{(2)} p_\tau^{(2)} - \phi_\tau^{(1)} p_\tau^{(1)}.$$

In the case where we compare sentiment lexicons, the two comparison texts are identical and it is only the sentiment scores that are different. Taking $p_\tau^{(1)} = p_\tau^{(2)} = p_\tau$, we have

$$\delta\Phi = \sum_{\tau} p_\tau \left(\phi_\tau^{(2)} - \phi_\tau^{(1)}\right). \tag{1.23}$$

Note that we no longer require a reference value as we are just comparing between the two sentiment dictionaries. In this case, the scores from one lexicon $\phi_\tau^{(2)}$ are either positive or negative relative to the scores from the other lexicon $\phi_\tau^{(1)}$.

This novel method of comparing sentiment dictionaries can be useful to find the amount by which a change in sentiment lexicon affects the overall sentiment of the text. It easily allows us to see the words that have the greatest impact on the sentiment. These may not necessarily be the terms which have the greatest difference in sentiment, since they are often uncommon words with a small value of p_τ. Common words with a low difference in sentiment would also not rate very highly here.

1.8 Empirical significance test

It is often useful to understand the significance of certain values that have been obtained. To do so, we can use significance testing. Often, however, the underlying null distribution may be unknown. In these cases, although it is impossible to perform a theoretical significance test, an empirical test can be used. This involves estimating the null distribution by shuffling the labels on data and repeating calculations with this new data set. The technique is implemented as follows.

1. Randomly shuffle labels on the data set.
2. Estimate a value from this new data set.
3. Repeat (1) and (2) until adequate samples have been obtained.
4. Calculate an empirical p-value.

The empirical p-value is calculated as

$$\hat{p} = \frac{r+1}{n+1}, \tag{1.24}$$

where n is the number of samples and r is the number of observations that are more extreme than the value calculated from the true data [34]. We add one to account for the fact that the test statistic is also an observation of the same random variable. As such, this should be included in a count of the extreme values.

Chapter 2

Exploratory data analysis

2.1 Data background

This section provides a comprehensive overview of all accessible data. We state our reasons for discarding some attributes. Finally, we discuss the cleaning that is undertaken before the data is ready to analyse.

Since 2014, following a 2012 amendment to the Australian Broadcasting Services Act 1992, primary channels have been required to provide captioning for all programs from 6am to midnight. News and current affairs programs must also have captions at all times [2]. Tveeder [54] is a service that reads these captions and uploads them to the website, www.tveeder.com where they can be read in real-time. This website contains Australian television captions for all primary channels (ABC1, Seven, Nine, Ten, and SBS), as well as ABC News 24. Using the Tveeder API, we are also able to obtain historical captions dating back to January 1, 2015 which have been stored in a database.

We collect caption data in the form of JSON files which contain several key pieces of information about a line of text:

- ID: A unique identification number for each line of text in the database.
- Channel: The channel that the text is from, given as ID numbers.
- Text: The text from one line of captions.
- Date: The date given in epoch time (seconds since 00:00:00 UTC on the 1st of January 1970).
- CID: The character ID. This is an identification number for each of the characters who are speaking in a particular program. Due to the lack of a law requiring this information, often there will a single CID for an entire program.

A separate request allows us to access data on the television programs of each day. This JSON file contains many details about the content of each program:

- ID: A unique identification number for each instance of a program.

text	date	program	genre
He put you up on a pedestal,	1641168856.988	The Heights	reserved
and now you've done	1641168857.232	The Heights	reserved
the exact same thing his father did.	1641168859.864	The Heights	reserved
Grab that box,	1641168862.347	The Heights	reserved

Table 2.1: An example of the head of one data frame displaying data extracted from Tveeder. This contains just the text, date, program, and genre, which were selected as key pieces of information to retain.

- Channel: The channel that the programs have been shown on. These are given as ID numbers that correspond with those from the caption data.
- EventID: A second unique identification number for each instance of a program.
- Title: The title of the program.
- Description: A brief description of the program.
- Start time: The time that the program begins (in epoch time).
- End time: The time that each program finishes (in epoch time).
- Content info: A genre describing the content of the program.
- Rating info: The classification category given to each program.
- Country: The country that each program is broadcast in. For our data, this is always Australia.
- Language: The language that each program is broadcast in. For our data, this is most often English.

By cross-referencing the date with the start and end times of the programs, we can match the text to the program that it came from. The key information – the text, date, program, and genre – was extracted and placed into a data frame for simplicity. An example of the head of one data frame is shown in Table 2.1.

All channels are missing data in March and April 2022 and Channel 10 is missing a much larger portion of data between August 2018 and July 2019. This means that we cannot make any conclusions for the respective channels during these times. We ignore the missing data and this has no effect on other results.

The caption data is first cleaned by removing punctutation and capitalisation. It is then split into the desired document length by cross-referencing with the program data. This allows us to break the data up first into programs, and then further into smaller documents with length corresponding with a given time interval.

	Total words	Unique words	Entropy	Sentiment (NRC)	Sentiment (Mittens)
ABC1	63,446,138	150,024	6.9925	0.6112	0.5586
Channel 7	46,571,024	151,511	7.1006	0.6218	0.5881
Channel 9	46,146,327	156,568	7.0651	0.6213	0.5889
Channel 10	48,778,397	124,688	6.9360	0.6280	0.6014
SBS	43,271,585	171,335	7.1314	0.6124	0.5555
ABC24	54,606,560	120,039	6.9290	0.6086	0.5105

Table 2.2: The number of total and unique words, the entropy, and the sentiment of each channel. See text for discussion.

2.2 Exploratory data analysis

We now explore the data in detail to understand the characteristics of each channel and provide insight into any differences we may find in later analyses. Section 2.2.1 examines the words in detail, comparing the most common words on each channel and the number of words spoken. Section 2.2.2 is the first step to understanding the reason behind differences in the words that appear on each channel. Here, we discuss the different programs broadcast on each channel and how these impact the words. We look at data from only 2022 and, assuming uniformity over time, this will give us an insight into the characteristics of each channel relative to the others.

2.2.1 Words

We begin our analysis with an exploration into the number of words in each channel and the composition of each corpus. We provide reasoning behind our findings where possible.

The total words and the number of unique words are shown in Table 2.2. Note that most of the unique words do not appear in the dictionary, but are instead terms such as people or place names, or may be misspellings. ABC1, despite having by far the highest total words, contains fewer unique words than Channel 7 and Channel 9. SBS contains the highest number of unique words. This can be attributed to its multicultural focus; it occasionally shows programs in other languages and discusses a wider range of people and place names.

The entropy of each channel is given in Table 2.2. Recall that this is a measure of the amount of 'surprisal' in the channels. ABC24 exhibits the lowest entropy, and therefore the lowest uncertainty in its text. The lower entropy is attributable to its high level of news and current affairs content, containing less variation in genre.

We now wish to look at the words which make up the text of each channel. To do

ABC1	Channel 7	Channel 9	Channel 10	SBS	ABC24
repeats	sunrise	drop	judy	sbs	government
tenable	chaser	pounds	judge	riders	abc
garden	7news	zone	you	peloton	minister
fuse	reporter	counters	phil	breakaway	south
plants	ball	pointless	flavour	france	wales

Table 2.3: The five words from each channel that have the greatest difference in Tsallis entropy from the original text. These can all be explained by programs which are broadcast often on the respective channels.

so, we use the Tsallis entropy with a value of $\alpha = 0.3$ to give more weight to less common words. This allows us to see the words that truly characterise the text. Word clouds showing the words with the greatest Tsallis entropy difference between each channel's text, and the entire text are given in Figure 2.1. The words are coloured by their sentiment in the Mittens lexicon, generated in Section 4.4. Words with sentiment greater than 0.5 (positive) are coloured in orange, while words with sentiment less than 0.1 (negative) are coloured in blue. All other words appear in grey. The word size roughly corresponds with the difference in Tsallis entropy where a larger word size indicates a higher difference. To summarise, the five words that have the greatest difference in Tsallis entropy from the entire 2022 corpus are shown in Table 2.3.

Each channel contains a high proportion of words that can be easily attributed to the programs that they broadcast. For instance, Channel 9 contains a high proportion of the terms 'pounds', 'drop', 'zone', and 'counters'. These are all commonly used in the program *Tipping Point* and rarely occur elsewhere. As we show in the next section, this is the third most common program broadcast on Channel 9.

Most words featured in the ABC24 word cloud have a negative sentiment. We will show that this channel contains by far the greatest proportion of programs from the news genre, a genre which has a very negative sentiment. These terms contribute to the negativity of the channel.

The overall mean sentiment of each channel is shown in Table 2.2. This was again calculated from both the NRC and Mittens lexicons. The two lexicons produce similar results for this score. We see that ABC24 clearly has the lowest sentiment. Again, this is because it is a news channel, and is therefore inherently negative.

The total number of words for each hour is shown in Figure 2.2. This is simply the sum over each day in 2022 of the number of words occurring within a given hour. The dip between midnight and 6am is expected as channels are not required to have captions during this time. The second dip in the SBS word count appears unusual, however is likely the result of foreign news programs which were often shown during this time at the beginning of the year. These programs do not require captions as they are in a

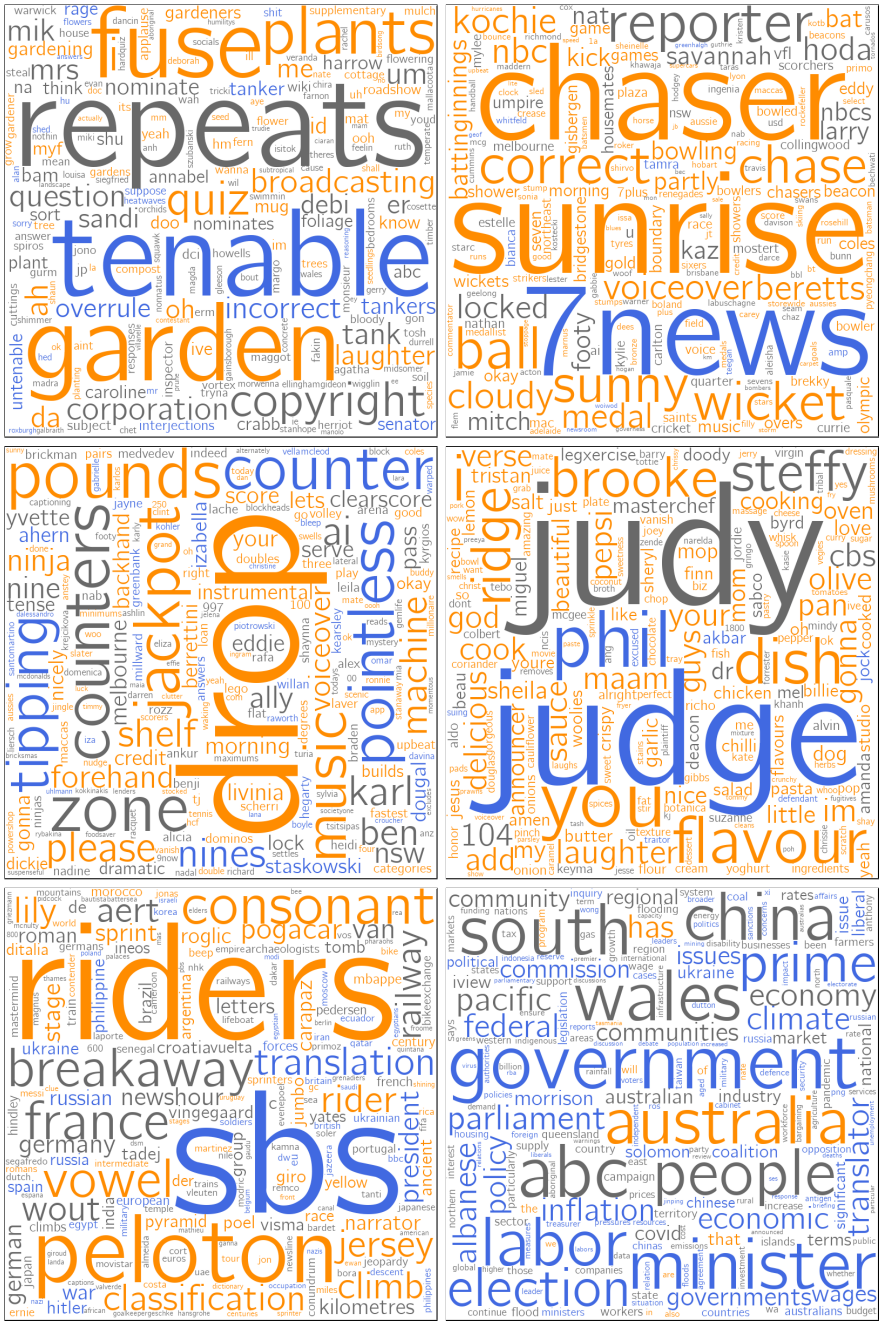

Figure 2.1: Word clouds indicating the words with the greatest Tsallis entropy difference between the 2022 text from each channel, and the entire 2022 text. From the top left, the channels are ABC1, Channel 7, Channel 9, Channel 10, SBS, and ABC24. Larger words have a greater Tsallis entropy difference. Words with sentiment greater than 0.5 (positive) are shown in orange, while words with sentiment less than 0.1 (negative) are shown in blue. Neutral terms appear in grey.

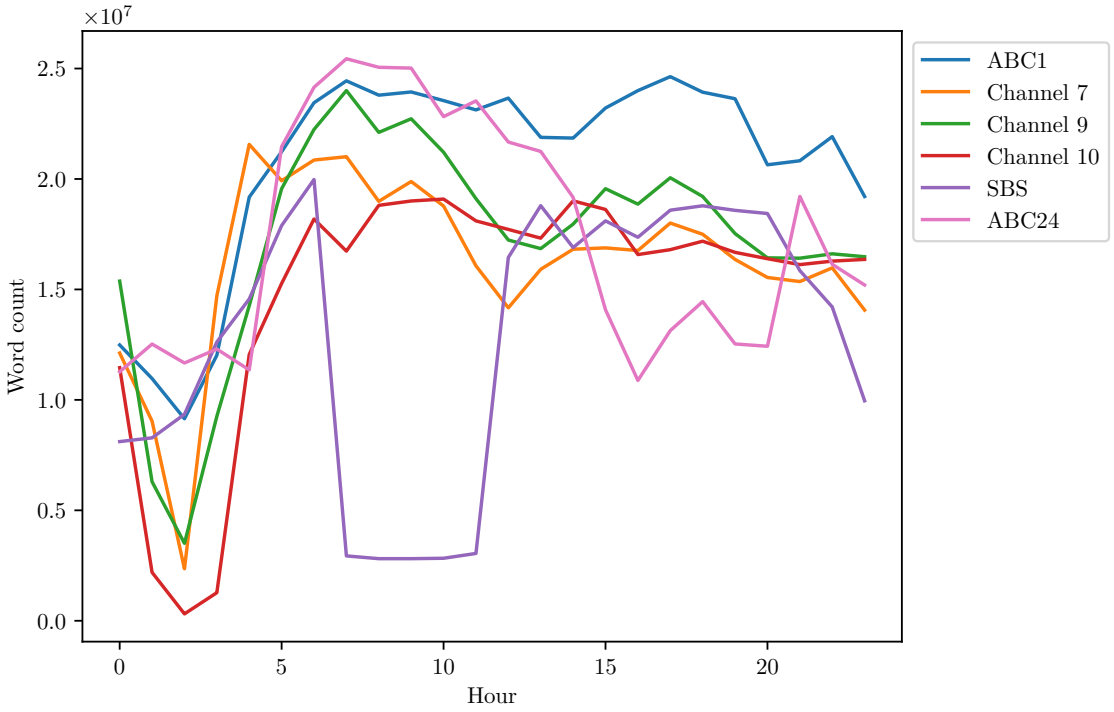

Figure 2.2: The number of words in each hour throughout 2022. There are clearly dips in the number of words in all channels between midnight and 6am when captions are not required. SBS has another significant dip between 7am and 11am, when news in languages other than English is commonly shown.

language other than English.

The total number of words for each day is shown in Figure 2.3. This is the sum of words in each day in 2022. There are between about 100,000 and 200,000 words each day. ABC1 has the highest word count by far, while SBS has the lowest, again because of its non-English programs.

We also see a significant drop in the number of words at the end of March and the beginning of April. Here, we are missing a large portion of data which causes the word counts to be much lower during this time. Data is also missing from the first three days of the year.

Most channels also appear to exhibit a weekly periodicity. This is most prominent on ABC1 which broadcasts the program *Rage* on weekends for 7 to 12 hours per day. This program has fewer captions since it is primarily music which does not require captioning and contributes to a significantly lower word count on weekends.

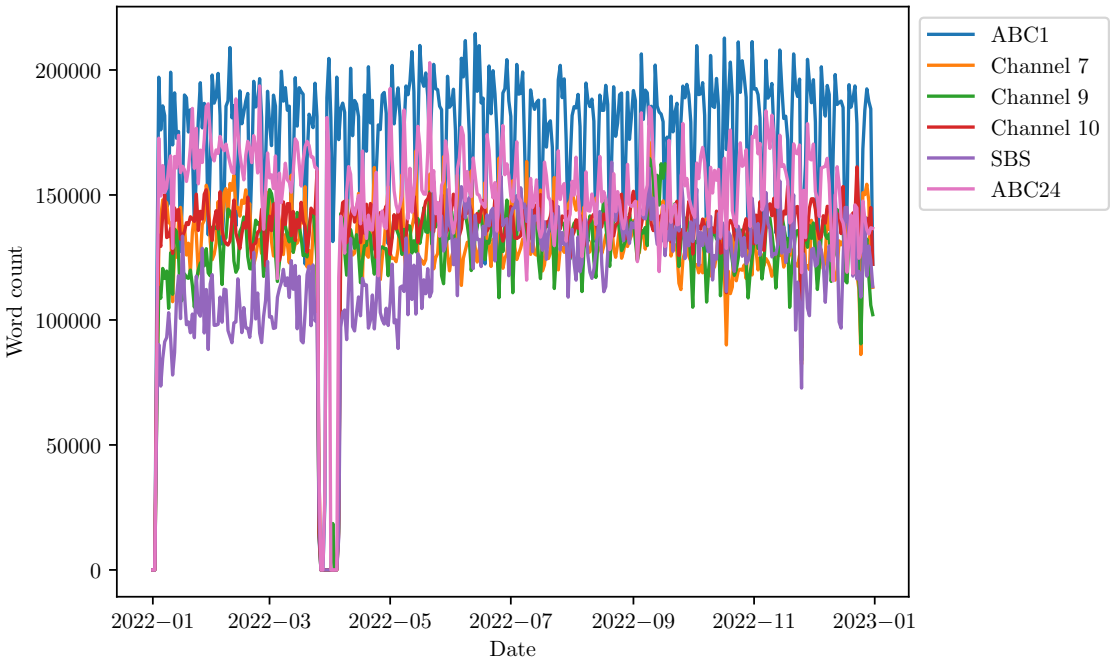

Figure 2.3: The number of words in each day throughout 2022. There is a significant drop at the end of March and the beginning of April due to missing data. The first three days of the year are also missing data. Most channels exhibit a weekly periodicity.

2.2.2 Programs

The programs televised on each channel form the basis for the text that appears. A look at the programs, and especially their genres, is another step to understanding differences in the content.

The number of unique programs and the genres that these belong to are shown in Table 2.4. SBS has by far the greatest number of both unique programs and genres. The high number of genres is the result of differences in the way that the programs are recorded. Many of these genres could be considered a 'subgenre' of the genres given on other channels. For example, SBS has 'performing arts' and 'soccer' categories which may just be labelled as 'arts' or 'sport' on Channel 10. The high number of programs indicates that SBS has a high program turnover.

The five most common programs on each channel are shown in Table 2.4, most being news programs. Words specific to other programs can be clearly spotted in the word clouds in Figure 2.1. We do not see words from the news genre, except in the ABC24 word cloud, since these words are not unique to each channel and do not appear relatively often enough.

The five most common genres in each channel are also shown in Table 2.4. ABC24 has by far the highest proportion of news programs. As mentioned before, SBS has a 'soccer' genre, which may be considered a subgenre of 'sport'. Differences between the way that each channel records data results in more genres for some channels. We also notice that many programs are categorised as 'other', or 'general', which does not provide us with meaningful insights into their content.

The sentiment of five selected genres are given in Table 2.5. The news genre has the lowest sentiment when calculated using the Mittens lexicon, which is built specifically to analyse news text. This explains why the sentiment of ABC24 is so low. The NRC and Mittens lexicons differ here. We explain the reason behind this in Chapter 4. The domain-specific Mittens lexicon should be used to analyse the news text, however the NRC lexicon is likely more reliable for other genres.

2.2.3 Summary

This data analysis has provided us with a comprehensive overview of each channel. Understanding the differences between them is key to interpreting results throughtout this thesis.

We understand that SBS, being a multicultural channel, has significant sections of time with no text due to programs in other languages. It also has a wider vocabulary as a result of its multiculturalism and greater number of unique programs. The main themes of the channel are cycling and international news.

ABC24 is predominantly a news channel; 68% of its programs are classified in the

	Unique programs	Unique Genres	Most common programs	Most common genres
ABC1	675	12	News Breakfast The Drum ABC News at Noon ABS News Mornings Rage	Other (24.61%) News (23.83%) Political (16.43%) General (14.31%) Education (7.39%)
Ch 7	526	16	Sunrise NBC Today The Morning Show Seven News Weekend Sunrise	News (40.79%) Sport (16.21%) Music (13.74%) Other (7.35%) Game show (5.15%)
Ch 9	551	9	Today Today Extra Tipping Point Weekend Today Nine News	News (31.92%) General (24.65%) Other (20.21%) Sport (5.72%) Political (4.37%)
Ch 10	260	9	Studio 10 10 News First CBS Mornings Judge Judy Dr Phil	General (67.29%) Other (12.69%) News (11.93%) Arts (1.84%) Sport (1.77%)
SBS	1263	25	SBS World News PBS Newshour The Cook Up DD India Prime Time News Mastermind	News (35.93%) Other (34.23%) Game show (6.35%) Soccer (4.33%) Cooking (3.47%)
ABC24	217	8	News Breakfast ABC News Mornings ABC News at Noon The World Weekend Breakfast	News (66.74%) Political (23.41%) Other (3.06%) Education (2.32%) Sport (1.85%)

Table 2.4: The number of unique programs and unique genres, as well as the five most common programs and genres on each channel. See text for discussion.

	News	Sport	Game show	Movie/Drama	Nature/Environment
NRC	0.5313	0.5337	0.5050	0.4915	0.4991
Mittens	0.5072	0.6044	0.6169	0.6031	0.6039

Table 2.5: The average sentiment of five selected genres. News has by far the lowest sentiment when calculated using the domain-specific Mittens lexicon. This explains why the sentiment of ABC24 is so low. The NRC and Mittens lexicon differ somewhat here. This is explained in more detail in Chapter 4.

news genre. Many of the terms with higher entropy are related to news and all of the most common programs are from the news genre. This results in the channel generally having a lower sentiment.

ABC1 and Channels 7, 9, and 10 have a wide range of programs. The most common programs are news-related, however they also feature many other programs from a variety of genres. The key terms used in each channel are summarised in the word clouds in Figure 2.1.

Chapter 3

Topic modelling

3.1 Introduction

This chapter aims to develop an understanding of topics in the media and form a measure for coverage bias. We use Correlation Explanation (CorEx) topic modelling [46] discussed in Section 1.2 to obtain the top topics from each channel, compare each channel and investigate how these change globally throughout time.

In Section 3.2, we explore the parameters used in CorEx in detail. We discuss the document length and number of topics that produce the best topic models. We also explore random variation within the models caused by the random initialisation of variables. In Section 3.3, we look at scaling up the model to handle enormous amounts of data by training our model on subsamples of our data and inferring probabilities from this model.

Applying the technique to our data in Section 3.4, we first fit an unsupervised topic model, allowing us to understand the topics that appear the most in each channel and form comparisons between them. We further compare the channels in Sections 3.5 and 3.6 with topic model similarity measures and by training a hierarchical topic model. In Section 3.7, we make use of the $p(y|\bar{x})$ term to model the media attention received by each topic which allows us to see how the relevance of these topics changes over time.

We then fit a supervised model, anchoring words that we know correspond to notable events that occurred in 2022. This allows use to see which channels focus more on each event and how the attention surrounding these changes over time.

Finally, in Section 3.8, we take our first look at bias, proposing a measure for coverage bias by analysing the probability that each topic is discussed in documents from each channel. This allows us to see possible coverage biases as some channels cover particular topics more than others. We extend this concept to political bias in Section 3.8.1. We

conclude by discussing improvements that can be made in order for us to investigate statement bias by analysing the content of the text.

In summary, the contributions of this chapter are:

- To our knowledge, the first topic modelling analysis of the Tveeder data set.
- The application of hierarchical and semi-supervised topic models to real-world data.
- The description of methods to determine the appropriate number of topics in a CorEx topic model.
- Discussion of the choice of document size in a corpus where this is ill-defined.
- Discussion of randomness in CorEx topic models and approaches to mitigate its effect on results.
- The application of CorEx to model media attention, and a measure extending this to coverage bias.

3.2 Model parameters

To apply CorEx topic modelling, we first need to understand its parameters to ensure that our topic models are the best possible. The only parameter required by CorEx is the number of topics. In this section we also discuss the document length. This is not necessarily a parameter of the CorEx model itself, but rather a way that we can adjust our data to help our topic model perform better. Due to the nature of our data, a document is imprecisely defined. We could simply take a document to be a single program, however we show that this does not produce good topics. We therefore manipulate our data by altering the size of a document to produce the highest quality topics. Further, we discuss the effect of random variation on CorEx topic models and how we can reduce its impact on our results.

3.2.1 Document length

Before conducting any analysis, our television caption text must be partitioned into documents on which to train the topic model. We have chosen to break the text down by time intervals, not by the number of words. Due to the spoken nature of the text, time is the natural unit with which to partition the captions.

The text that we have is first broken down into programs to ensure that the text within each document is consistent and does not cover multiple programs. The text is then separated further into 5-minute intervals so that we obtain a higher level of precision than we can achieve with hourly or half-hourly programs. These 5-minute intervals form the documents that we feed into our topic model.

10s	60s	120s	300s	600s	1200s	1800s	program split
sport	sport	sport	sport	sport	sport	sport	sport
bee	game	game	players	players	athletes	dani	swimmer
red	win	players	win	match	athlete	scarf	ski
win	players	match	game	afl	intili		
captions	match	final	match	games	daniela		
broadcasting	final	team	player	league	elite		
copyright	team	league	games	player	dani		
corporation	cup	cup	team	coach	wheelchair		
australian	league	games	league	sports	hockey		
final	games	player	final	victory	gym		

Table 3.1: The top words for topics anchored on the word 'sport'. By inspection, these topics appear to be reasonably coherent for lengths of document from 60–600 seconds. The 1800 second and program split documents only consist of three words and are not informative at all. We conclude that any length of document shorter than 10 minutes can be used.

The reasoning behind a choice of 5-minute intervals is as follows. Take words such as 'sport', and 'weather' for example. These are very common topics, appearing in almost every single news program. If we separate text into just one document for each program, we will likely see the 'sport' and 'weather' topics appearing in every single one of the news documents. This won't provide us with much information about the actual content of the news. If we instead separate the text into documents that are 5 minutes in length, we will have the precision to see if these topics appear more than once. This provides us with more information about how the topics are being discussed in the media. Through testing, we find that this greatly improves the performance of the topic model.

The length of the documents also has an effect on the actual topics themselves. Although we want to maximise the total correlation, it is unreasonable to compare these for different numbers of documents. This is because as the length of documents is increased, there is a higher probability of each topic belonging to each document. This increases the joint probability between the two, in turn increasing the total correlation. We are also unable to compare the topic models using coherence as this measure will also increase with the length of the documents due to the co-occurrence of terms being relatively higher.

We therefore qualitatively compare the topic anchored by the word 'sport' to determine the effectiveness of topic models trained on documents of different lengths. We do this 30 times for each document length and obtain subjectively consistent results. The top 10 words for an example 'sport' topic for varying lengths of document are shown in Table 3.4.

The topics are quite similar and appear reasonably coherent for lengths of document from 60–600 seconds. The 1200 second (20 minute) document length topic model has found the name of an ABC sports reporter, Daniela Intili, however this is not the kind of sports-related term that we are after, and the topic in general is much less coherent. The 1800 second (30 minute) document lengths, and the program split topic models were only able to form topics containing three words. These topics are not very informative or coherent. From these results, we conclude that any length of document shorter than 10 minutes is reasonable to use for CorEx topic modelling.

Splitting documents into a shorter length of time, however, means that we are working with more documents overall. This significantly increases the computation time required. In worst case, the scaling for the CorEx algorithm is $O(Nn)$. As such, reducing the document length from 300 to 120 seconds will approximately increase the run time by a factor of 2.5, while reducing again to 60 seconds will increase the run time approximately 5 times. It is this balance between computation time and precision which pushes us to choose intervals of 5 minutes rather than 1 or 2 minutes.

Due to the computational simplicity and increased precision, we will split our television programs into documents containing 5 minute intervals of text. These intervals contain an average of 700 words with a standard deviation of 255 words. A histogram of the number of words contained in each 5-minute interval is shown in Figure 3.2. In Figure 3.1, the number of words in each program is plotted for comparison. Although there is still significant variation, by limiting the interval size to 5-minutes we have removed much of the variation and improved the quality of the topics.

The optimum value of this parameter will not change as we alter the corpus size. Hence, we will split our text into 5-minute intervals throughout this chapter.

3.2.2 Number of topics

The number of topics is a critical parameter in the CorEx topic model, having an impact on the total correlation of each topic, and hence that of the topic model in general. The quality of the topics is affected as the number of topics is adjusted: the precision of each topic varies, as well as the number of words, and of course the actual words themselves. We present three methods to determine an appropriate number of topics to choose.

To find an appropriate number of topics, we investigate how the total correlation of the topic model changes as we increase the number of topics. This is the sum of the individual total correlations for each topic. We start at 10 topics and increase by values of 10 looking at the total correlation each time. We perform this for 30 iterations with random seeds on all data from 2022. A box plot showing how the number of topics relates to the total correlation is shown in Figure 3.3. It is evident from this plot that values of 30 and 40 topics provide the greatest total correlation and as such will

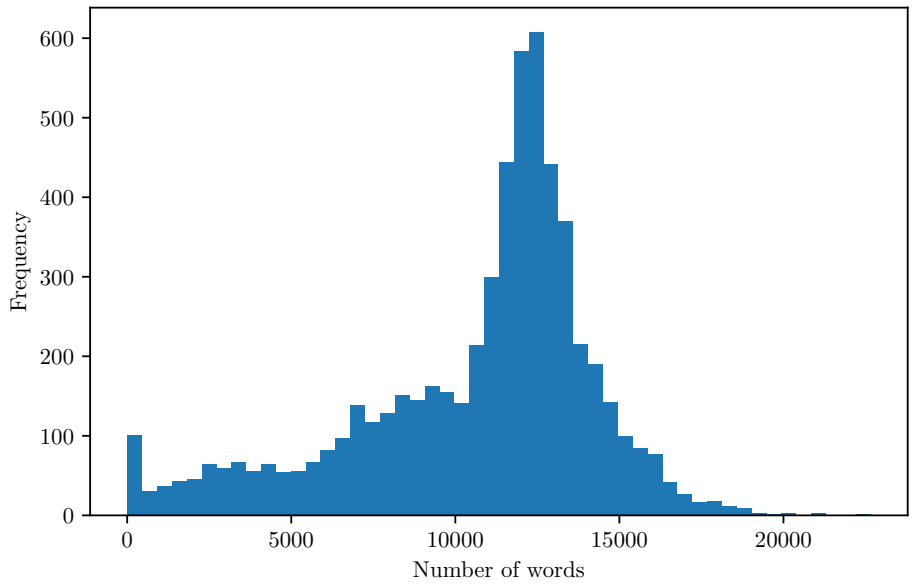

Figure 3.1: The number of words contained in each program. There is significant variation with documents ranging from 0 to over 20,000 words.

theoretically produce the best topics.

We note here that the total correlation decreases as the number of topics is increased past 30. This decrease seems counter-intuitive since this total correlation is simply the sum of the (non-negative) total correlations of individual topics. If we add another topic, this new topic could simply be empty (and thus have 0 total correlation), and when added to the total correlation of the original topics, the overall total correlation would remain the same. If the algorithm has truly maximised this total correlation, there should be no decrease as more topics are added.

The reason for this drop in total correlation is that the algorithm is not perfect; it may not necessarily find the global maximum but instead converges to a local maximum. As such, the calculated total correlation may find a different local maximum and decrease despite the global maximum actually increasing. Since we are applying the technique in real-life, we focus on the real data that we have obtained, and the high values of total correlation at 30 and 40 topics, rather than this theoretical maximum.

To understand why we obtain these values, we study the political topics obtained from topic models containing 10, 40, and 100 topics in Table 3.2. This allows us to look at the actual content of these topic models and conduct a quantitative analysis. All of the topic models produce reasonable topics. The 10-topic model's political topic is somewhat more general than the others, while the 100-topic model contains two highly

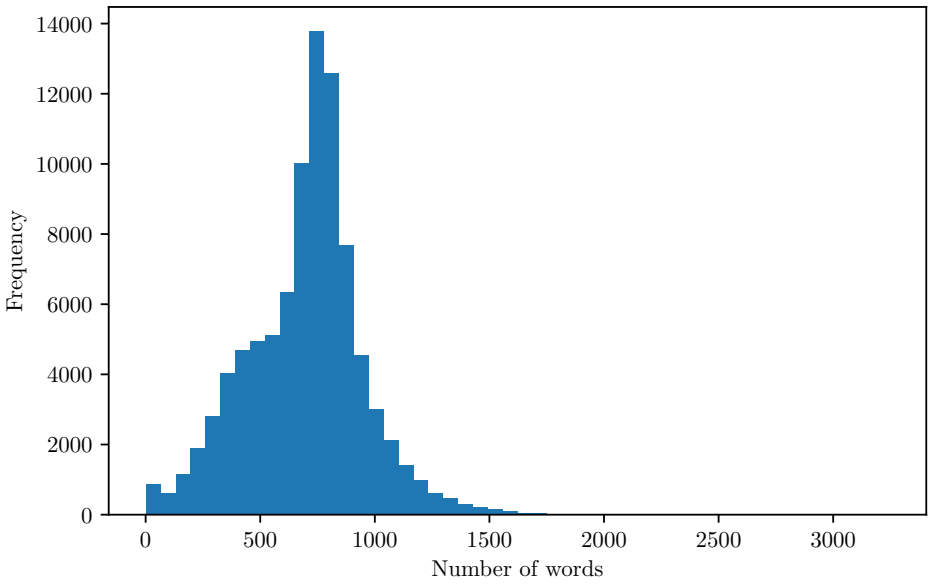

Figure 3.2: The number of words contained in each 5-minute interval. Limiting documents to 5-minutes in length has greatly reduced the variation. Values now only lie between about 0 and 1,600 words.

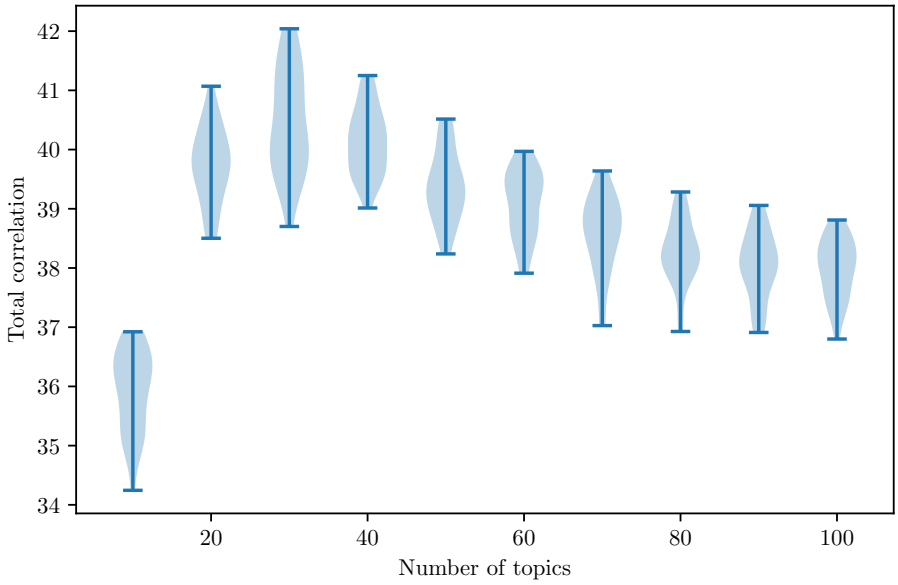

Figure 3.3: The total correlation of the topic model for number of topics 10 to 100. Note that this is the total correlation of all topics (the sum of the individual total correlations) for topic models containing n topics. Values of 30 and 40 topics have the highest total correlation indicating that these would produce the best topics.

10 topics	40 topics	100 topics	
government	government	election	minister
minister	election	labor	prime
prime	labor	albanese	government
election	federal	political	federal
federal	political	parliament	ministers
says	economy	party	governments
state	governments	morrison	leaders
political	economic	liberal	cabinet
labor	policy	anthony	meeting
country	liberal	coalition	boris

Table 3.2: An example of the words contained in the manually labelled political topic for topic models containing 10, 40, and 100 topics. The topic from the model containing 10 topics is very general. The 100 topic model produces two political topics which can make it difficult to compare with other topics. The topic from the 40 topic model appears to be the most coherent and useful in this case.

specific political topics. For our analysis, such a level of detail is not necessary and it would be difficult to compare these two topics with other areas of focus consisting of just one topic.

In addition to simply maximising the total correlation, we would also like our topic model to be relatively stable so that a small change in the number of topics does not have a dramatic impact on the topics themselves. We therefore plot a heat map of the similarity between like topics in topic models with number of topics differing by 10. We use the Pearson and Spearman similarity measures in (1.13) and (1.14) to compare the ranked lists of the words contained in each topic. The heat maps showing these similarities are shown in Figure 3.4. Higher values around 30 to 50 topics indicate that adjusting the number of topics from 30 to 40, or 40 to 50 would not have as much of an effect on their content as adjusting from 90 to 100 topics. Choosing these values would ensure that our topic model is more robust to a change in the number of topics.

From all results presented, we conclude that 40 topics is the most appropriate number for a topic model trained on all data from 2022 due to the high total correlation, topic stability, and reasonable topics. We also consider values of 30 or 50 topics to be reasonable and find that a small change in the number of topics would not have a large adverse effect on the quality of our topic models.

Note that this data set containing all words from 2022 has just over 300 million words and 289 thousand unique words. We expect that as we increase the number of unique words in the corpus, the number of topics required in a topic model should also increase. Furthermore, the distribution of words within text will also change with time

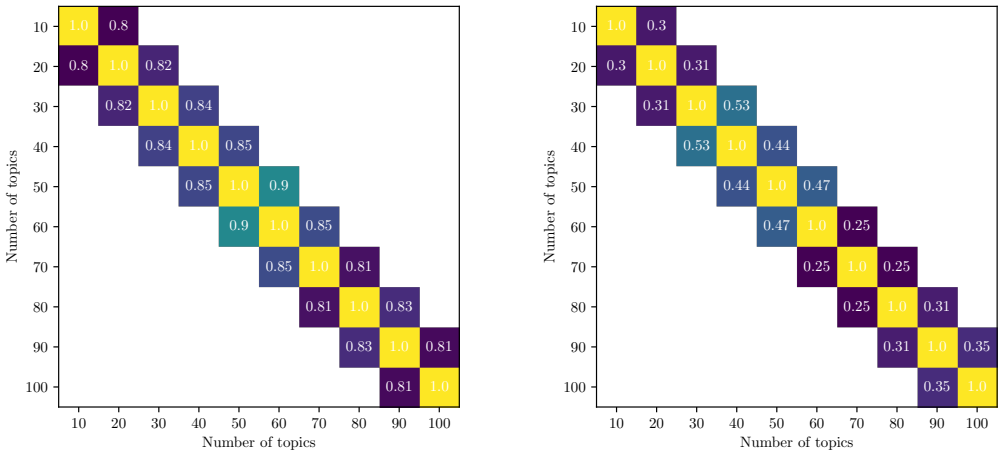

Figure 3.4: A comparison of the Pearson and Spearman similarities between topic models with an adjacent number of topics. We only look at topics which differ by 10 to avoid clutter since the other comparisons are meaningless here. Topic models with 40 and 50 topics have the highest Pearson similarity, while topic models with 30, 40, and 50 topics have the greatest Spearman similarity. Choosing one of these values would ensure that the topic model is robust to small changes in this parameter.

as different events are discussed, again requiring more topics in our topic model to pick up these different subjects.

Due to the nature of the data that we have, the increase in topics is more likely due to an increase in the time period of the data, rather than an increase in just the words alone. Therefore, we expect to see an increase in the number of topics required if we expand the time period while leaving the number of words roughly constant. We will take random subsamples of our text over the entire time period that we have data for and determine whether a topic model trained on this corpus will require a larger number of topics.

We randomly sample one seventh of the documents across the entire corpus. For one realisation, this leaves us with 339 million total words and we now have 540 thousand unique words. We expect that it is the number of unique words and variation in their distribution rather than the total words which has a greater impact on the number of topics required. This would mean that we would need more topics as the length of time is increased to encompass the entire corpus.

A violin plot displaying the results of 30 iterations of topic models with a range of topics from 10 to 100 is plotted in Figure 3.5. Again, we see the total correlation peak at 30 topics. Interestingly, this is followed by 20, 40, and 50 topics. The increase in the length of time has clearly not increased the number of topics required; if anything, this has decreased!

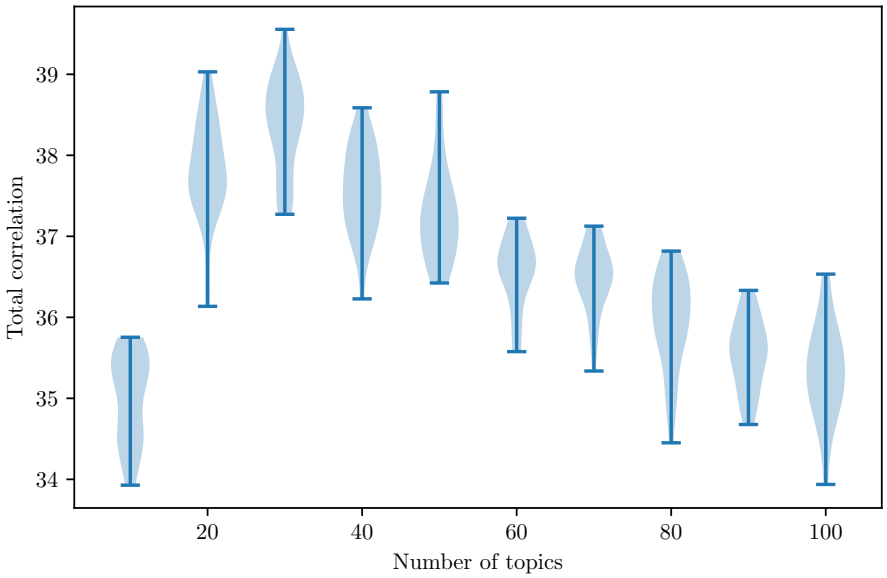

Figure 3.5: A comparison of topic models with an adjacent number of topics. These topic models are trained on a random sample of one seventh of the data so that there is roughly the same number of words but a greater time span. We see similarities with Figure 3.3. Values of 20, 30, and 40 topics perform the best in this case. This indicates that the greater time span is not significant enough to warrant the need for more topics.

1	2	3	4	5
sport (general)	ukraine	sport (general)	sport (general)	sport (general)
common words	cooking	ukraine	politics	ukraine
ukraine	politics	politics	cooking	cooking
short words	weather	cooking	ukraine	economics
cooking	sport (general)	economics	news (general)	short words

Table 3.3: The top 5 topics for unsupervised topic models trained on all data from 2022 with random seeds 1 to 5. All iterations contain 'sport', 'ukraine' and 'cooking' topics indicating a high level of similarity when using a different random seed.

Due to the setup of our topic model, to reduce computation time and disregard unnecessary information we use only the 10,000 most common terms. For a corpus containing a larger number of raw words or unique words, the topic model will still only be using 10,000 of these words. Although these unique words may contain more variation, we have shown that the number of topics required remains at approximately 40. We therefore choose to use 40 topics for all topic models throughout this thesis, even when training on larger data sets.

3.2.3 Random variation

Due to the random initialisation of α and $p(y|\bar{x})$ in the training of the CorEx topic model, there are small variations in the outcome of the topic modelling. Here, we look at the amount of variation that the topic models have when given a different starting seed.

We train 30 topic models on ABC1 data from 2022 using the random seeds 0–29 in training. The top 5 topics sorted by total correlation from the first five iterations are shown in Table 3.3. These topics have been manually labelled for clarity.

The top 5 topics are very similar for each iteration. Topics manually labelled as 'sport', 'cooking', and 'ukraine' all appear in the top 5 for each iteration, indicating a high level of consistency. There is also high consistency within the topics. The top 10 words ordered by their mutual information with the 'sport' topic for the first five iterations are shown in Table 3.4.

We plot the $p(y|\bar{x})$ terms for the 'sport' topic from the first five topic models in Figure 3.6. These values represent the probabilities of each topic given each document in the data set. The $p(y|\bar{x})$ terms are pushed towards either 0 or 1 and so we must apply a moving average to make sense of the results. A moving average window of 5000 documents is used in all cases where data from just 2022 is used. We notice some variation even in the topics containing an identical top 10 words. As shown in Figure 3.7, the $p(y|\bar{x})$ terms of all topics have a high Pearson correlation. Noticeably,

1	2	3	4	5
game	win	game	game	game
players	sport	players	players	players
win	games	win	win	win
match	team	match	match	match
sport	won	sport	sport	sport
team	coach	team	team	team
final	afl	final	final	final
player	football	player	player	player
games	cup	games	games	games
ball	finals	ball	ball	ball

Table 3.4: A comparison of the top 10 words in the 'sport' topics for each random seed. All topics are identical except for the second. Due to the random initialisation, this topic model has reached a different local maximum to the others.

the second topic containing a slightly different set of terms has the lowest correlation, however this is still high and reasonable.

We plot the $p(y|\bar{x})$ terms of the sport topic for all 30 topic models and see a different picture. Figure 3.8 clearly shows that the vast majority of probabilities lie within a relatively small area, however there are some outliers. The lines seem to form at least three distinct groups, each of which may correspond to a different local maximum that has been found. Just one set of values seems to not follow the same trend. This topic model contains very different terms and as such should be treated as an outlier.

It is clear from these results that the seed does indeed have a great impact on the results of the topic model. Clearly, there is enough random initialisation to push the topic model to a different local maximum. As such, it is important that we train each topic model multiple times to mitigate the influence of potential outliers. By taking the mean of multiple runs, we can reduce the effect of random variation.

3.3 Inference

We are unable to train our topic model on all of the 7 years of data that we have due to computational limitations. Hence, we devise a method for conducting inference on our model. We are able to predict the $p(y|\bar{x})$ terms for each document that we havewhile training on a much smaller subset of documents. We have two methods of subsampling to choose from.

Method 1: Train the topic model on all data from one shorter time period.

Method 2: Train the topic model on a randomly selected subsample from all of

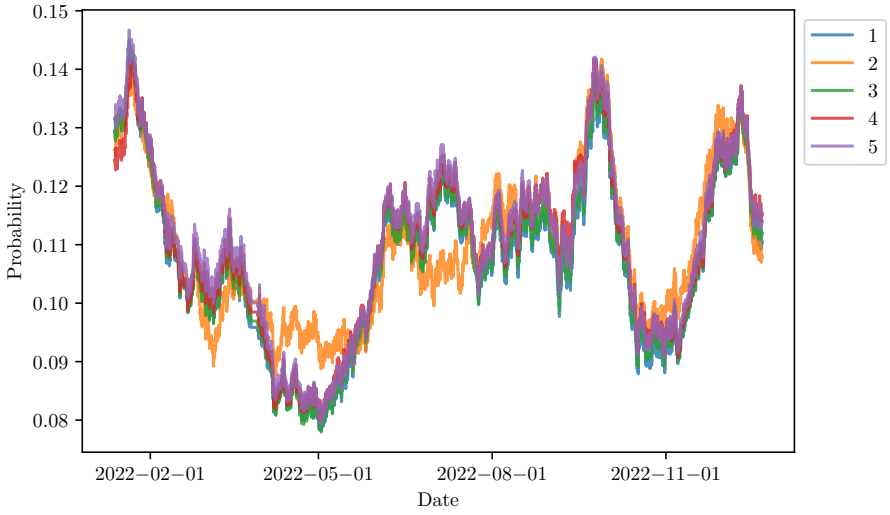

Figure 3.6: The probability of the sport topic in ABC1 captions from 2022 for 5 topic models with different seeds. Notice that the peaks and troughs line up very well despite the topic from the second topic model containing very different words. It has, however, reached a different local maximum.

our data.

In both cases, we make predictions for the probabilities of the remaining documents using (1.10). This allows us to compare them with each other, and with the ground truth.

We train two models: Model 1 using Method 1, and Model 2 using Method 2. We use data from ABC1 from 2015–2022. Training on just one channel removes any variation caused by differences between channels, reducing noise. The restriction to a smaller data set also allows us to train a ground-truth topic model for comparison. This model was trained on all 7 years of data from ABC1.

We train Model 1 on all data from 2022. Model 2 is trained on a random subsample of one seventh of the data to ensure that the two datasets are similar in size. We anchor the terms ‘covid’, ‘election’, ‘sport’, and ‘gardening’ to different topics for a wide variety of topics to compare. These topics all appear frequently in unsupervised topic models of ABC1.

The anchored topics in the three topic models consisted of very similar terms, however they were not perfectly identical due to differences in the subsampling and randomness in the model. To improve this subsampling, we could simply anchor every single word found in the ground-truth topics to both Models 1 and 2. This would ensure that our models end up being as similar as possible, however we must remember the reason for our subsampling in the first place. Without having a ground-truth to compare from,

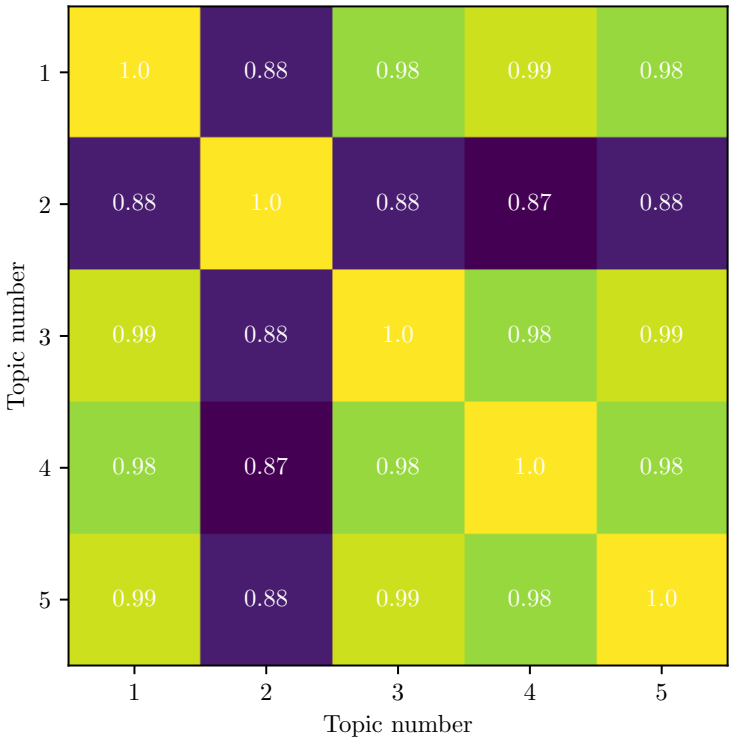

Figure 3.7: The correlation between each set of values of $p(y|\bar{x})$. The second topic model has the lowest Pearson correlation, the result of differences in its words due to the different local maximum it has reached.

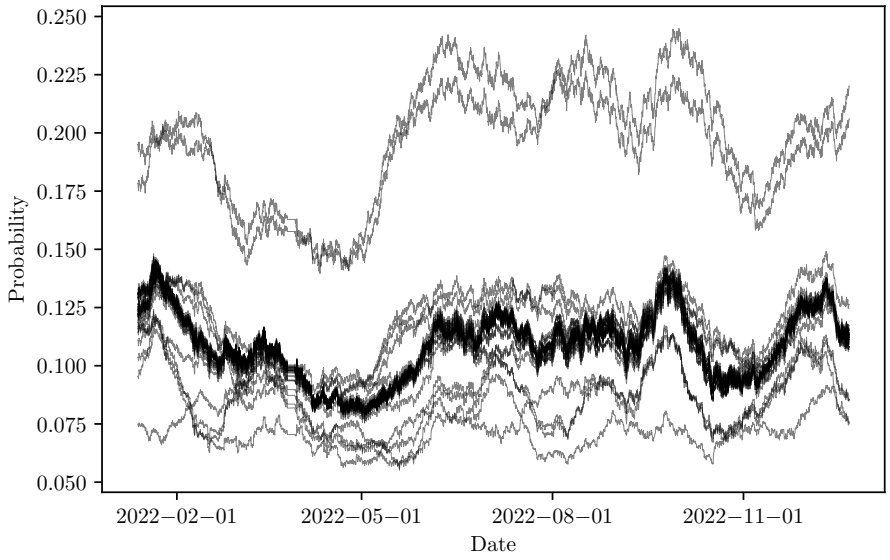

Figure 3.8: The probability of the 'sport' topic in ABC1 captions from 2022 for 30 topic models with different seeds. The distribution of topic probabilities given a random seed is not unimodal but may have 3 or even 4 clear stable points. These likely correspond to the various local maxima that our topic model can find. Despite differences in the probabilities, values from each topic model follow a similar trend with peaks and troughs occurring on almost identical dates.

	covid	sport	gardening	election
Method 1	1.0000	0.9753	0.8802	1.0000
Method 2	1.0000	0.9749	0.9510	1.0000

Table 3.5: The Pearson correlation coefficients between the inferred values and the ground truth. The 'covid' and 'election' topics are both almost identical to the ground truth. The 20% subsample performs better for the 'gardening' topic.

it will not be possible for us to know the terms that should occur in the topic model. As such, we anchor just one term.

The probabilities of the given topics in each document, $p(y|\bar{x})$, are shown in Figure 3.9. A moving average with a window size of 20,000 is applied to this data and all cases where we use data from 2015–2022. Let us pause here to discuss these figures in detail. The models have almost identical $p(y|\bar{x})$ terms for the 'covid' and 'election' topics. Despite being trained on different sets of data, they have reached the same local maximum and produce incredibly similar results for all time periods.

For the other two topics, we had expected that the probabilities from Model 1 would converge to the ground truth as the date gets closer to 2022. This is the data that the model is trained on, and we therefore expect it to follow the ground truth closely during this period. Although we don't see complete convergence in Figure 3.9, the model certainly nears the ground truth and follows the shape of the curve more closely during this time. Model 2 does, however, appear to follow the ground truth more closely in general.

The Pearson correlation of the two models with the ground-truth is shown in Table 3.5. This confirms that Model 2 indeed follows the ground truth more closely for the 'gardening' topic. Both models, however, perform well and have strong correlation with the ground truth. In addition, the Pearson correlation over 30 runs with different seeds is shown in Figure 3.10. As we found in Section 3.2, there is a fair degree of variation between runs, although Method 2 generally had higher correlation with the ground truth. Method 1 worked well for major events that occurred during this time period that it was trained on: the COVID-19 pandemic and the 2022 Australian federal election.

Although it works quite well for these particular topics, there are several reasons we may not want to make inference from just 2022 alone. Each year contains many major events that are somewhat unique to it. 2022, for example, contained major events such as 'ukraine', 'covid', and 'flood' which were picked up by the topic model. Contrasting this with 2015, we see no significant mention of these topics, however other topics, such as 'syria', appear strongly. Programs relating to 'sport' and 'gardening' in 2015 may also contain vastly different sports and content compared with 2022 and as such the topics should contain different terms. Making inference about the past or future from a topic model can lead to incorrect assumptions being made due to subtle differences in

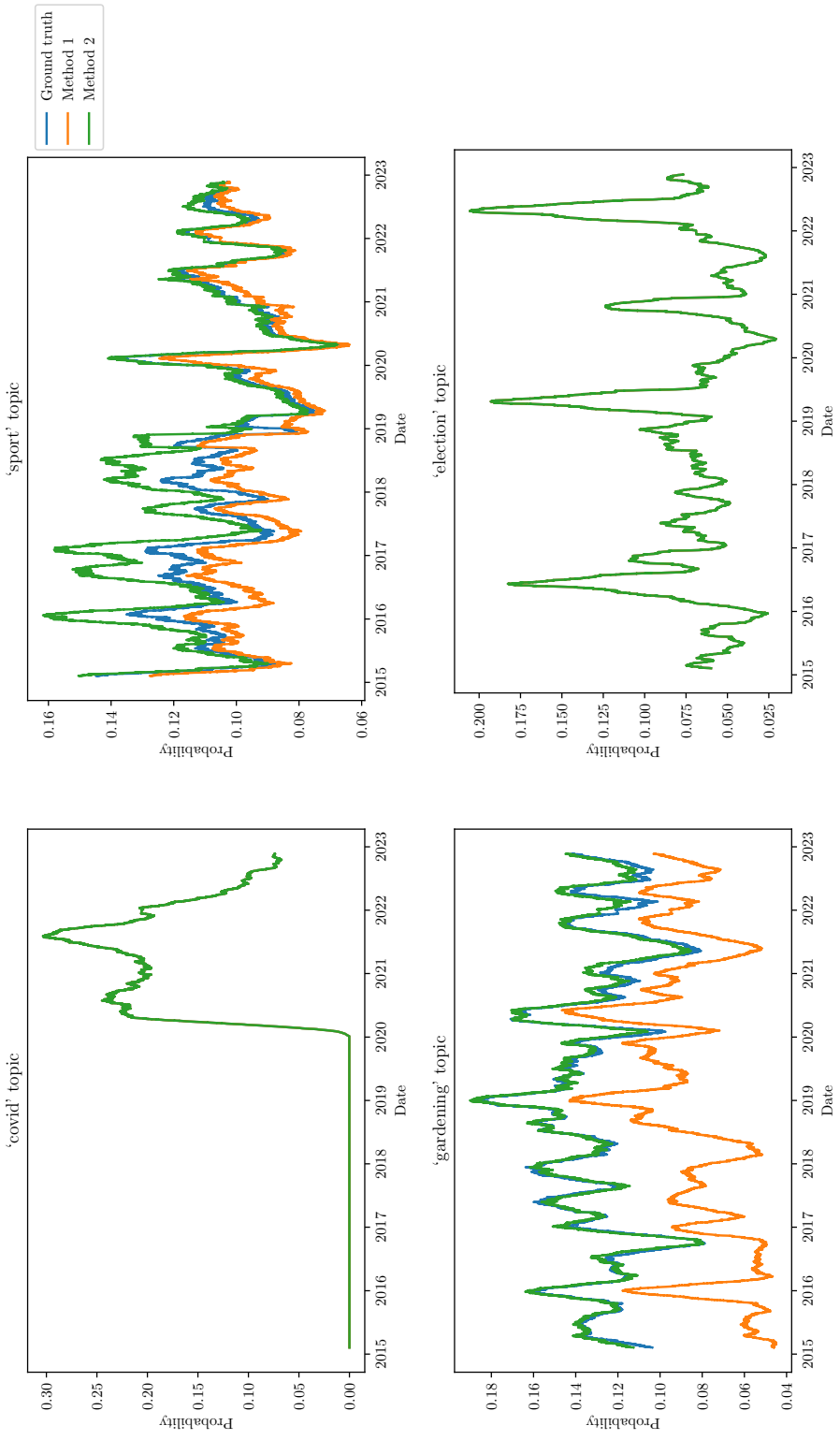

Figure 3.9: A comparison of the probabilities found using different measures of inference. Values for the 'covid' and 'election' topics are almost identical. The 20% subsample (Method 2) appears to provide a better approximation of the ground-truth in the other cases.

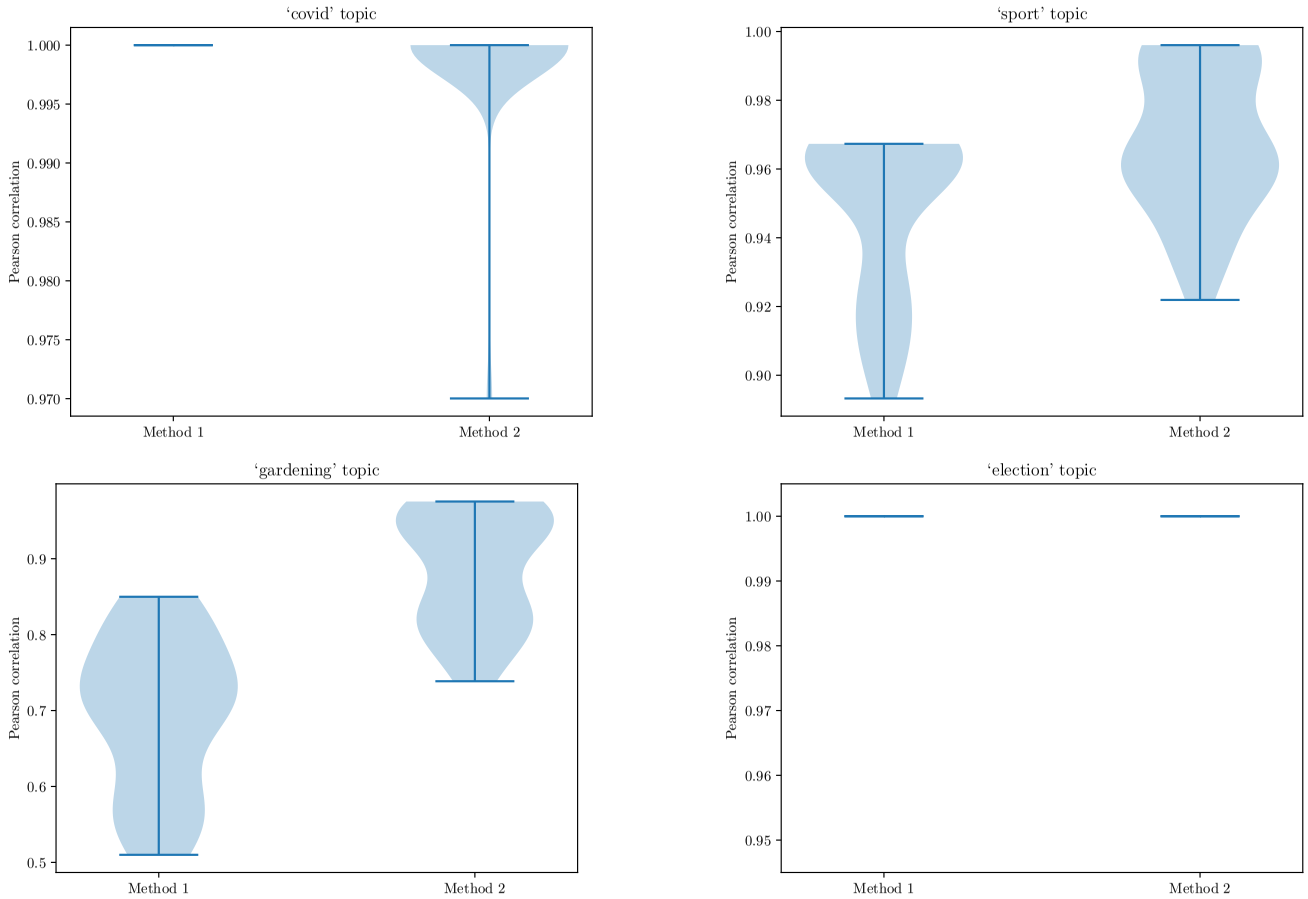

Figure 3.10: Violin plots showing the Pearson correlation between both methods of inference and the ground truth from 30 runs of topic models with different seeds. The covid and election topics have nearly identical probabilities for both methods over every run. There is much more variation in the gardening and sport topics, and Method 2 performs much better on these topics. We would expect to see greater variation in the content of these topics over time.

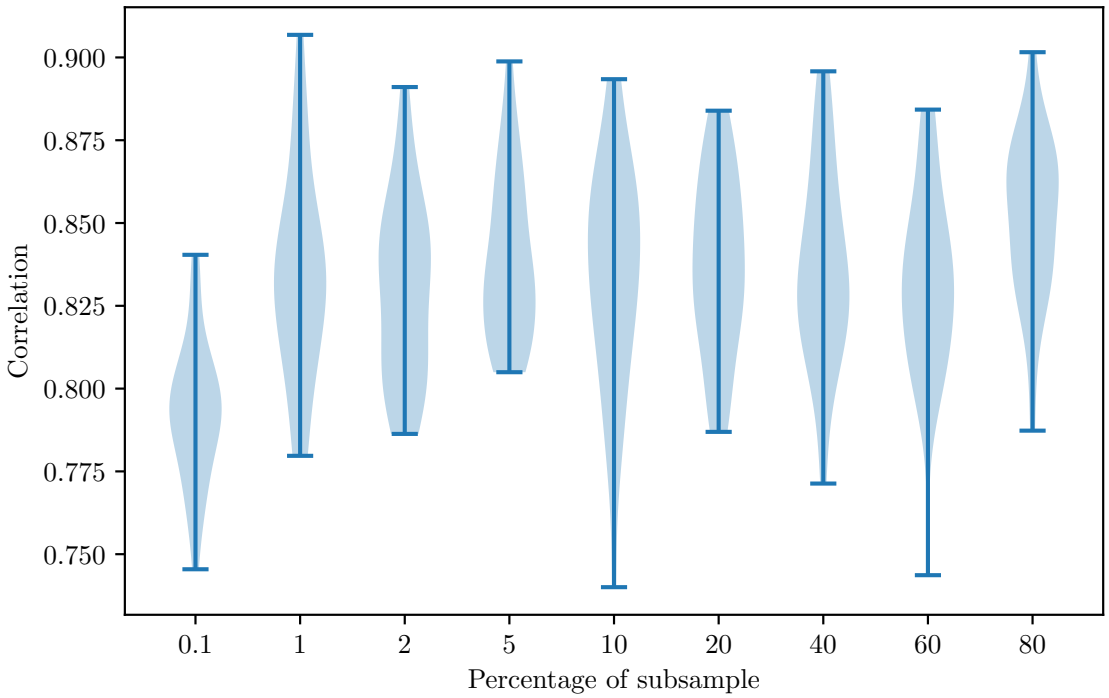

Figure 3.11: The Pearson similarity between 30 topic models trained on the full ABC corpus and subsamples of the corpus. The 80% sampled model performs the best, however taking such a large subsample of the data defeats the purpose of the exercise. Any other subsample size provides topics that are reasonably close to the entire model, even subsamples of just 0.1%.

topics. We will therefore be using Method 2 to infer probabilities from large quantities of data throughout this thesis. If required to make predictions for documents which were not available at the time of training, however, Method 1 is a reasonable alternative.

To find the amount of data at which the topic model is relatively stable, we take 30 subsamples of 0.1, 1, 2, 5, 10, 20, 40, 60, and 80 percent of the data using NumPy seeds 0–29. We then calculate the Pearson similarity, introduced in Section 1.3, between models using (1.13). We do not anchor any words in this topic model, as this also allows us to see how the subjects of the topics also differ when unsupervised. A violin plot of the Pearson similarity is shown in Figure 3.11.

The 80% sampled model performs the best. Taking such a large subsample of the data is a somewhat pointless exercise, however, since the computation time is not greatly reduced. Other subsample sizes also provide topics that are reasonably close to the entire model, even subsamples of just 0.1%.

In practice, we will take a subsample of 10% of our data to ensure that we are obtaining

ABC1	Channel 7	Channel 9
sport (general)	sport (cricket)	game show
politics	sport (football)	sport (general)
gardening	weather	weather
ukraine	floods	crime
interjections	politics/covid	weather
Channel 10	**SBS**	**ABC24**
cooking	sport (cycling)	sport (general)
advertisements	politics	ukraine
news (general)	cooking	politics
politics	war	inflation
sport (general)	ukraine	covid

Table 3.6: The top 5 topics for each television channel. All channels contain a 'sport' topic. When this is a specific sport, it corresponds with sports that most commonly appear on the channel and whose words appear in the word clouds in Figure 2.1. Other topics generally align with our findings from Section 2.2.

reliable results while also drastically reducing the computation required. This allows us to generate meaningful results using just one tenth of our data. This subsampling technique can be incredibly effective at reducing both the time and memory required while also providing good results.

3.4 Unsupervised topic modelling

In order to understand each channel's characteristics, we first run an unsupervised topic model on their individual captions from 2022. This approach enables us to identify the prominent topics in each channel, forming the basis for a comparative analysis. The top words sorted by mutual information are reviewed and the topics are then manually labelled. The top 5 topics, sorted by total correlation, for each channel are shown in Table 3.6. The corresponding probabilities of these topics over time are shown in Figures 3.12 and 3.13.

We find that sport is a very common topic. Each channel contains at least one sport topic in their top 5 and, when specific, this corresponds to the sport which is played most often on that particular channel. The other topics that we see align with our findings from Section 2.2.

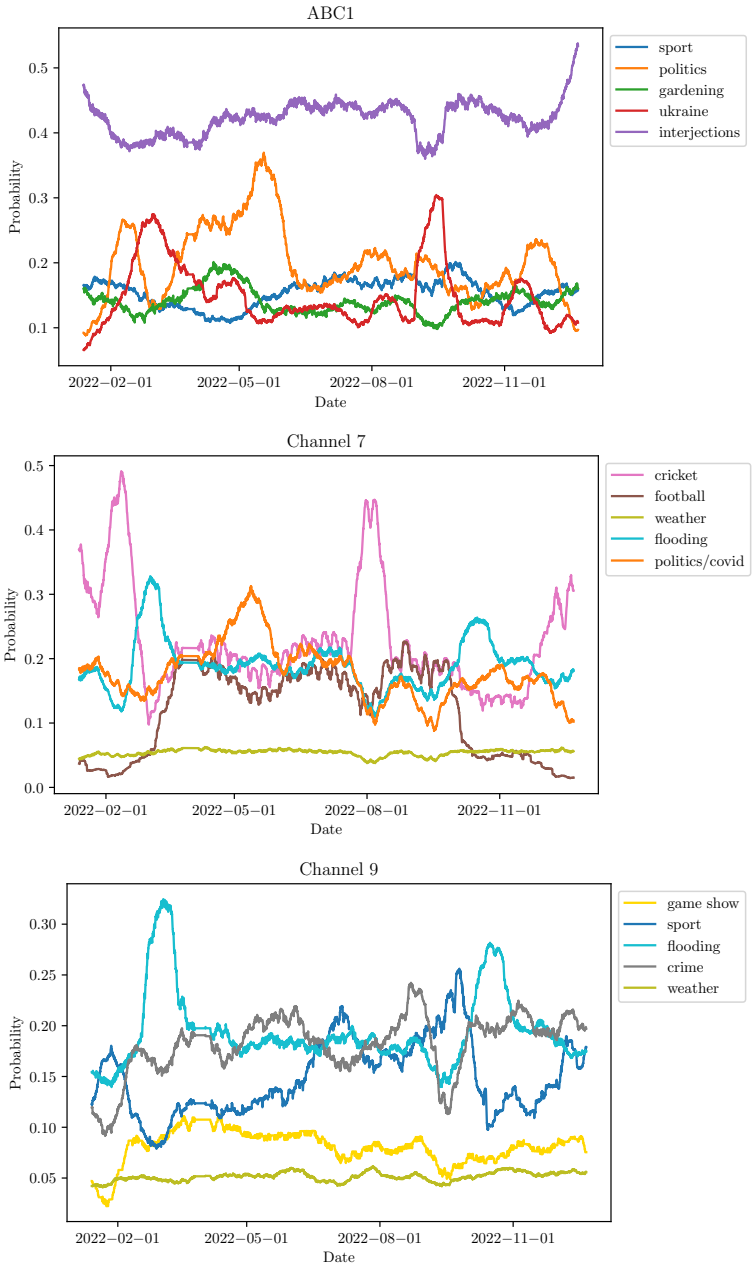

Figure 3.12: The probability of seeing the top topics for each channel over 2022. There are clear peaks in the 'sport' topics surrounding major sporting events. Some weekly periodicity is also seen here representing weekly sporting cycles. We also see 'politics' topics peak around the election, while 'ukraine' topics peak around the start of the Russia/Ukraine war. These are good signs that the $p(y|\bar{x})$ term is a good indicator of media attention.

Figure 3.13: The probability of seeing the top topics for each channel over 2022. There are clear peaks in the 'sport' topics surrounding major sporting events. Some weekly periodicity is also seen here representing weekly sporting cycles. We also see 'politics' topics peak around the election, while 'ukraine' topics peak around the start of the Russia/Ukraine war. These are good signs that the $p(y|\bar{x})$ term is a good indicator of media attention.

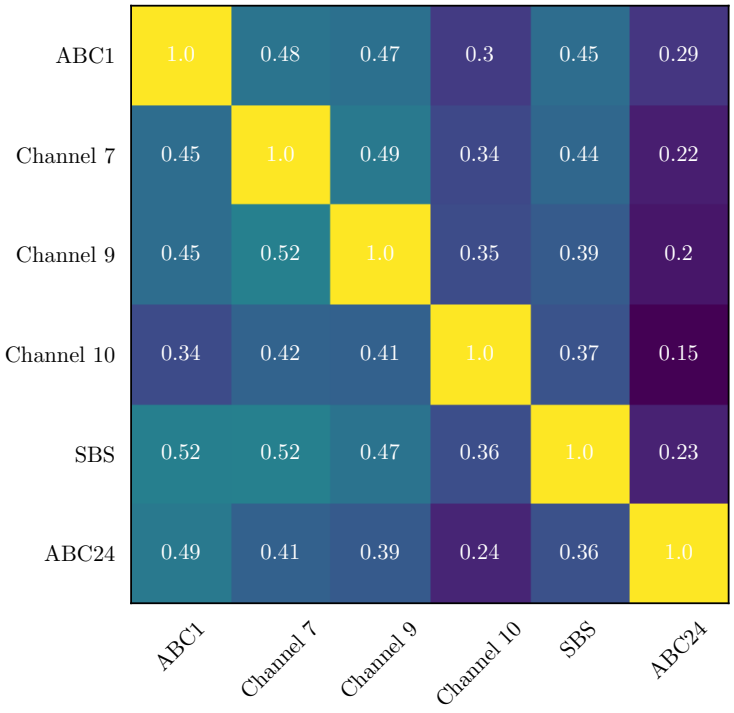

Figure 3.14: The mean Pearson similarity of 30 repeats of topic models trained on text from each channel. ABC24 has very different topics to the other channels. As we have seen, this is a news channel and shows a different variety of programs.

3.5 Channel comparison

Another way to understand the differences in channels is to examine the differences in topic models trained on each channel. We first compare unsupervised topic models. This allows us to compare the number of topics that each topic model has in common, and thus the similarity in the topics between each channel. Plotted in Figure 3.14 is the Pearson similarity of topic models from different channels found using (1.13). This is the mean after running this 30 times.

Recall that this similarity measure requires us to compare over identical documents. In order to do this, we use one topic model to make predictions on the text that the other topic model was trained on. We compare the $p(y|\bar{x})$ terms for this set of documents. We then reverse this process. The heatmap is not symmetric, as the calculations are not identical.

ABC24 has relatively low Pearson similarities. This indicates that the topic models, and therefore the text that they are trained on, are dissimilar. This is the result of a combination of the difference in the number of like-topics and the similarity of

these like-topics. The Pearson similarity gives more weight to the quantity of shared topics rather than the similarity between these topics and so it is likely that ABC24 has a greater difference in the topics which are present. ABC24 generally contains a high proportion of news-related topics, some of which may not be present or may be combined together on another channel.

To analyse just the content and similarity of given topics, we can anchor the terms

covid	ukraine	election	flood	queen	sport

to topic models trained on each of the channels. These words correspond to major events in 2022 and we know that each corpus should contain content relating to each of these topics. This allows us to compare how the differences between channels affect the words that appear in topics with identical anchors. Plotted in Figure 3.15 is the mean Pearson correlation of the $p(y|\bar{x})$ terms for identically anchored topics between each channel. This is similar to the Pearson similarity measure used previously, however this time we only look at the 6 anchored topics and take the mean over just these 6 topics.

SBS and Channel 7 have the lowest mean Pearson similarity for these topics. We split this value into the individual topics to investigate where the dissimilarities stem from. A heatmap for each topic is shown in Figure 3.16.

It is clear from this that the dissimilarity of Channel 7 primarily stems from the 'sport' topic. The Channel 7 sport topics are predominantly focussed on the Commonwealth Games while other channels cover more football and popular sports. Almost one third of the Channel 7 sport topics are nonsense, containing terms such as 'beretts', 'optus', and '5g', and 'ethereal'. These terms pertain to Channel 7 sports reporters and advertisements and have low similarity with the other channels.

SBS has relatively different 'election' and 'flood' topics. It is an international channel which primarily broadcasts news from overseas. These events occurred in Australia and featured less in the SBS news because of its less Australian-centric orientation. The SBS election topic instead contains terms such as 'president', 'biden', and 'united', indicating a topic which is more focussed on American politics. Its flood topics generally consist of more transport-related terms such as 'train', 'passengers', and 'ship' but also contain the names of overseas cities.

Also note the low similarity of the ABC1 Queen topics. Approximately half of these topics are mixed in with words such as 'contestant', 'incorrect', and 'answer', which are used in quiz shows. This is likely from the high proportion of quiz programs on ABC1, some of which may feature questions on Queen Elizabeth II.

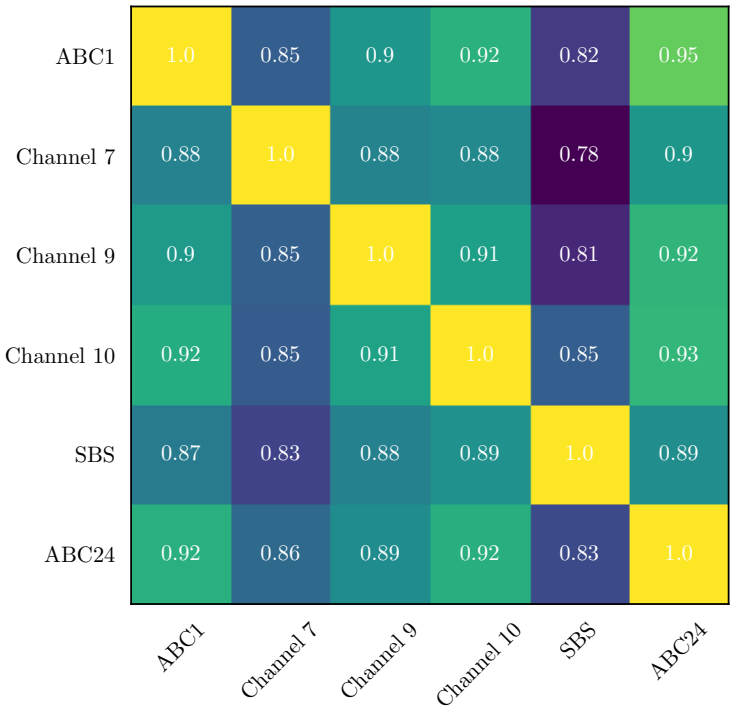

Figure 3.15: The mean Pearson similarity of 30 repeats of topic models trained on text from each channel with anchored words. SBS and Channel 7 are relatively different to the other channels. To investigate why this is the case, we break this result down into the individual topics.

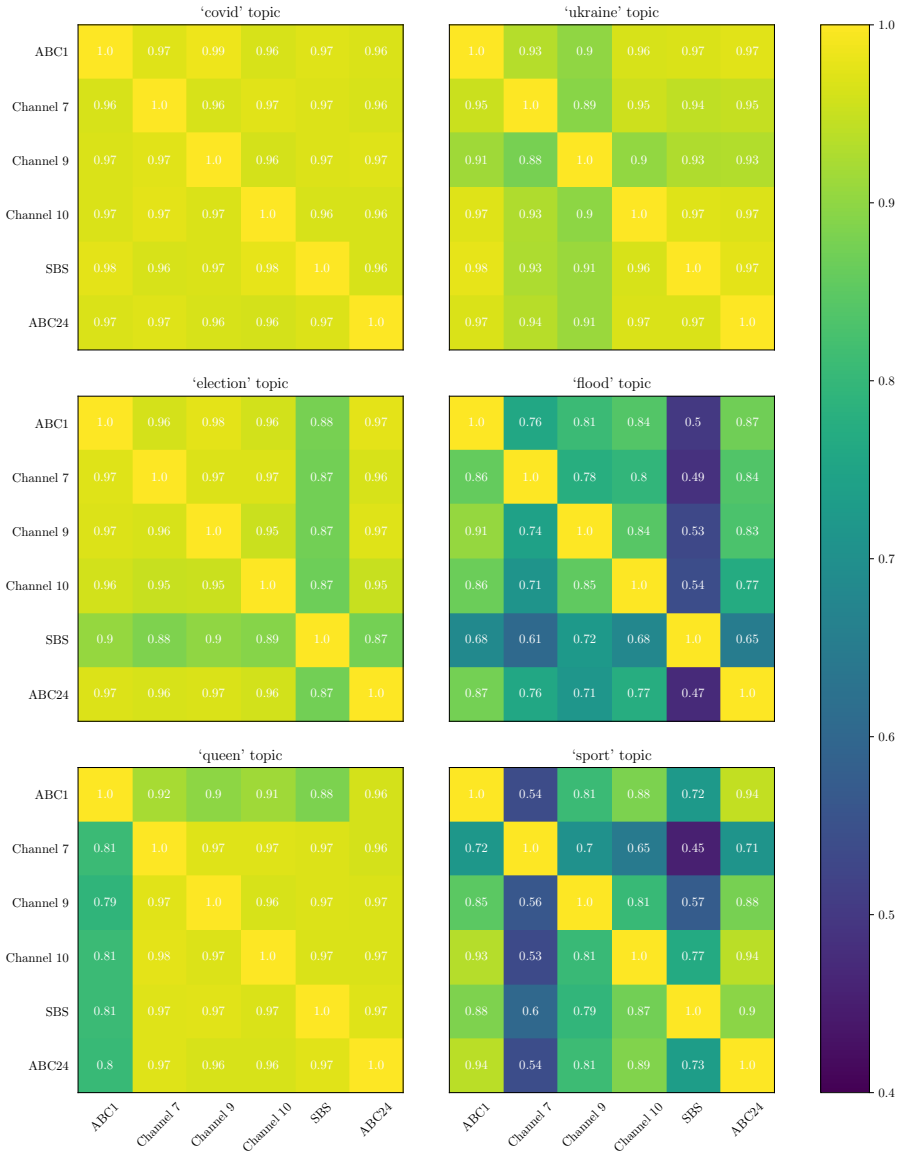

Figure 3.16: The mean Pearson similarity of 30 repeats of each topic from supervised topic models trained on each channel. All 'covid' topics are incredibly similar. SBS has relatively different 'election' and 'flood' topics. This is the result of it being an international channel which broadcasts these events from across the world and is not limited to primarily showing Australian news. Channel 7 has a very different sport topic as it is very sport-oriented and shows a wider variety of sports more often than other channels.

3.6 Hierarchical topic modelling

We can delve deeper into each channel's unsupervised 'sport' topic to determine the sports which are commonly broadcast and find a wider range of terms relating to this topic. The top 40 words ordered by mutual information are shown in Table 3.7. ABC1 and ABC24 have quite vague topics since they don't show any sports in particular and these topics simply come from the news programs. Other channels' sport topics clearly indicate the sports that are shown on those channels. Channel 7 broadcast both the Commonwealth Games and the Winter Olympic Games, both of which feature strongly in the topic. Channel 9 shows the Sunday Footy Show and broadcasts NRL matches. Channel 10 plays A-league football matches, while SBS broadcast the 2022 FIFA World Cup. Simply looking at the content of each topic can give us a great deal of insight into what appears on each channel.

Similarly, we can explore the range of sport topics by exploiting the hierarchical aspect of the CorEx topic modelling framework. This adds more structure to the results previosuly obtained. We choose our first topic model to contain 200 topics. Recall that this means the topics will be shorter, more precise, and there may be multiple topics of the same theme. After training a topic model on text from each channel, we see several different sport topics emerge. Generally, there is one topic for each different sport that regularly appears on a channel.

Instead of manually searching for other sport topics, we can train a second topic model on the *topics* obtained from the first topic model. This creates a hierarchical structure where we are able to group the topics obtained from the first topic model into larger overarching topics. Searching for the larger topic containing the term 'sport', 'ball', 'game', or 'player' allows us to automatically find a wide variety of sport topics.

The second topic model contains 20 topics so that on average there are 10 sub-topics per topic. This works well for half of the channels; we clearly find several sport topics on Channel 7, Channel 9, and SBS. The top 10 words from these topics are shown in Section 3.6. The topics generally relate to an individual sport which is broadcast by the channel. For example, we see obvious football, cricket, and horse-racing topics in the Channel 7 topic model.

We were unable to obtain meaningful results from the other channels. In these cases, several seemingly random sub-topics are mixed together in the larger overarching topic. We see topics referencing things such as Elon Musk and NASA appear alongside sports topics.

Although this technique hasn't worked for three channels, we have produced good results for the channels with the highest proportion of sport programs. If there are many topics that need sorting through, this can be a reasonable way to approach the task, however should be used with caution and the results verified.

We can perform this analysis with any topic which we may naturally obtain multiple

ABC1	Channel 7	Channel 9	Channel 10	SBS	ABC24
sport	sport	sport	sport	sport	sport
players	race	game	players	cup	game
win	medal	coach	game	game	match
game	olympic	footy	league	goal	win
match	gold	players	football	ball	players
player	racing	afl	goal	players	cup
games	bronze	club	player	tournament	games
team	medals	football	coach	football	league
league	olympics	against	win	player	team
final	athletes	season	games	match	coach
won	womens	player	cup	goals	won
coach	champion	win	ball	play	player
afl	medallist	finals	against	scored	final
cup	beretts	games	team	games	afl
tournament	silver	team	final	fifa	play
sports	track	final	victory	argentina	grand
season	mens	league	play	penalty	tournament
victory	races	nrl	match	referee	scored
finals	beijing	premiership	club	goalkeeper	finals
play	horse	collingwood	socceroos	fans	mens
cricket	podium	field	wizards	played	sports
mens	fastest	stadium	commentator	stadium	season
womens	championships	goal	euphoric	qatar	cricket
fans	athlete	rugby	goals	kick	football
tennis	lap	panthers	season	league	womens
football	pyeongchang	cup	arrows	midfield	fans
scored	commonwealth	bulldogs	finals	messi	victory
played	horses	clubs	packets	matches	played
tony	gisbergen	cameron	played	coach	playing
nrl	qualifying	teams	afl	brazil	nrl
grand	skiing	clash	grand	score	rugby
rugby	laps	victory	scored	playing	winning
winning	skating	tigers	spell	croatia	beat
goal	freestyle	eels	field	finals	tennis
playing	mostert	mcg	playing	champions	matches
teams	birmingham	sports	champions	mbappe	captain
goals	lizzie	blues	teams	final	round
sporting	davison	goals	won	morocco	goals
athletes	sprint	captain	stadium	half	teams
matches	filly	penrith	fc	scorer	goal
aflw	snowboard	carlton	clash	socceroos	club

Table 3.7: The top 40 words in the 'sport' topic for each channel ordered by mutual information. The content of the topics reflect the sports broadcast by each channel.

Channel 7
game, ball, players, play, player, match, played, playing, team, boundary
forward, side, mark, opportunity, chance, middle, hasnt, looked, form, momentum
off, front, line, third, top, far, half, wide, foot, running
footy, goal, kick, quarter, goals, afl, football, collingwood, kicked, season
against, ground, defence, shot, short, square, shots, opposition, sides, decisions
position, gets, score, straight, goes, spot, nicely, hand, quick, fast
wicket, cricket, wickets, bowling, innings, batting, bat, test, overs, bowled
pressure, hit, high, field, hitting, mitchell, eight, highest, pushing, zone
big, performance, watching, watch, hopefully, unbelievable, beautifully, absolute, terrific, james
race, racing, win, won, second, horse, races, run, horses, cup
Channel 9
game, match, players, ball, player, coach, sport, finals, final, footy
court, against, music, set, novak, djokovic, return, open, opening, action
point, last, second, third, left, forward, another, line, close, far
three, two, four, one, five, top, first, six, down, up
win, play, played, won, playing, film, winning, prize, winner, series
cup, cricket, t20, socceroos, england, wickets, sri, wicket, lanka, pakistan
defence, defend, attacking, tackled, conceded, restart, gillard, offensive, murray, 10m
loan, online, apply, loans, anz, insurance, save, switched, direct, excludes
swimming, medal, freestyle, championships, swim, medals, swimmer, relay, race, athletes
supporters, reminding, mob
SBS
win, team, final, second, won, winner, third, champion, teams, winning
riders, race, rider, front, bike, finish, seconds, riding, gap, position
game, players, football, player, play, played, games, playing, league, fans
peloton, jersey, stage, breakaway, climb, tour, classification, van, sprint, de
cup, goal, ball, tournament, goals, match, scored, argentina, fifa, penalty
minutes, chance, minute, plenty, opportunity, shoulder, box, injury, hasnt, screen

Table 3.8: The top 10 words in each sport sub-topic within the larger sport topic for channels in which we were able to automatically sort these. We see several similar sub-topics within the larger overarching sport topic. These generally relate to just one sport which is played on the channel. The entire topic represents the wide range of sports which appear on a channel.

aspects of, such as politics.

3.7 Media attention

We now aim to model media attention as a first step towards a measure for coverage bias. We start by naïvely looking just at the counts of the words used as anchors in the previous section. We test whether this is a reasonable estimate of media attention by plotting these values for topics corresponding to major events which occurred during 2022. We find these topics by anchoring the terms:

covid ukraine election flood queen sport

A moving average of the counts over time is plotted in Figures 3.17 and 3.18. We then compare the count values with values obtained from a CorEx topic model.

During optimisation, the CorEx topic modelling framework generates probabilities of each document belonging to each topic. We hypothesise that these $p(y|\bar{x})$ terms will also allow us to visualise the media attention that each topic receives at a particular time. During optimisation, the value of $p(y|\bar{x})$ is pushed to either 0 or 1. This gives us approximately binary information on each document and so plotting these raw values will not give us a good indicator of the media attention over a long period of time. Hence, we will take the moving average of these probabilities to visualise the relevance of a topic during a particular time window. The probability of each of these topics over 2022 is plotted in Figures 3.19 and 3.20.

We find that ABC24 generally has both a higher probability and count of these topics. This is because it is a specialised news channel and so only shows programs of the news genre. Thus, we expect to see a higher proportion of news-related topics such as these.

Where ABC24 does not have higher probabilities or counts, there are clear explanations. Both Channels 7 and 9 score highly on the 'queen' topic, especially in September following the death of Queen Elizabeth II. Similarly, Channel 7 has a slightly higher probability during her Platinum Jubilee celebrations in May and June. This corresponds to Channels 7 and 9 both broadcasting the funeral, while Channel 7 had exclusive rights to show the Platinum Jubilee.

This topic is also not consistent between the topic probabilities and the counts. This discrepancy lies with the fact that in most sport programs, the word 'sport' is rarely mentioned, however it is a common term to feature in news broadcasts. The topic model has picked up actual sporting programs, while the word count of 'sport' is unable to do so.

Within the topic probabilities, each channel's 'sport' topic peaks at a different time. While the probability of the ABC1, ABC24, and Channel 10 'sport' topics remain relatively steady, the other channels have peaks which correspond to the seasons of the

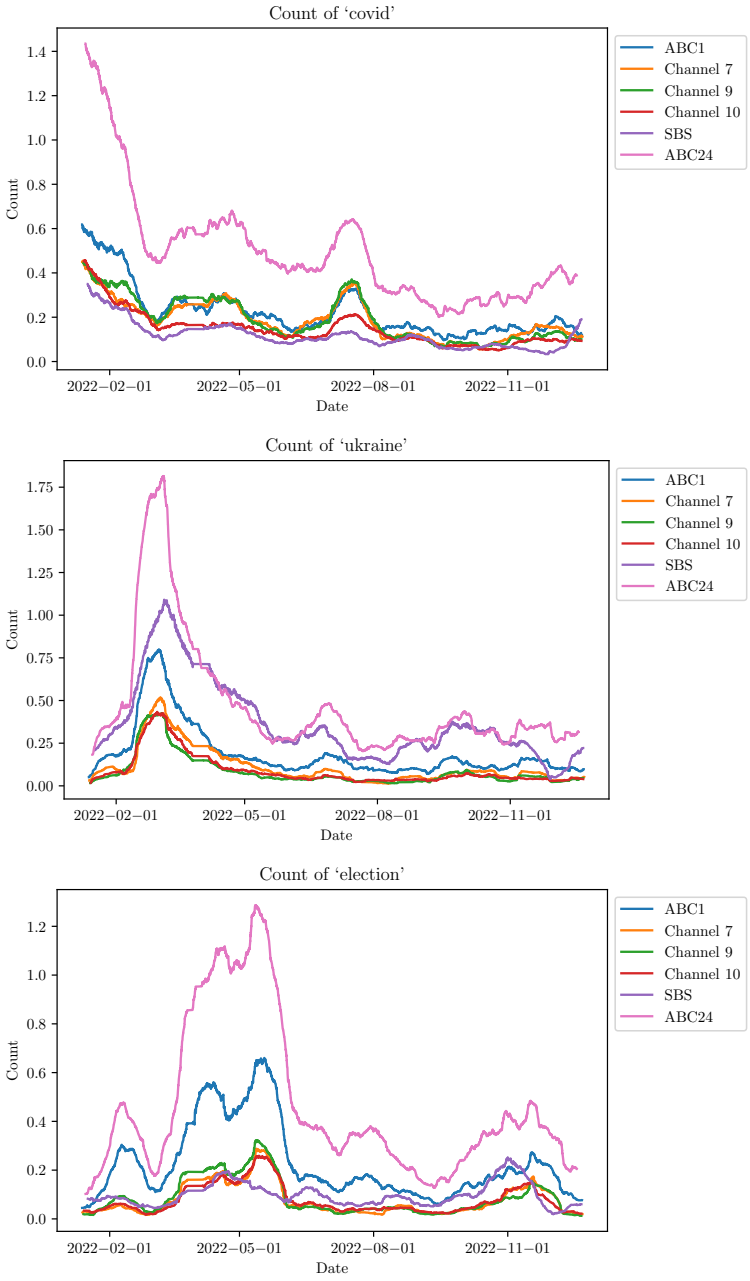

Figure 3.17: A moving average of the word counts over 2022. ABC24 generally has the highest word counts. These values are very similar to those in Figures 3.19 and 3.20.

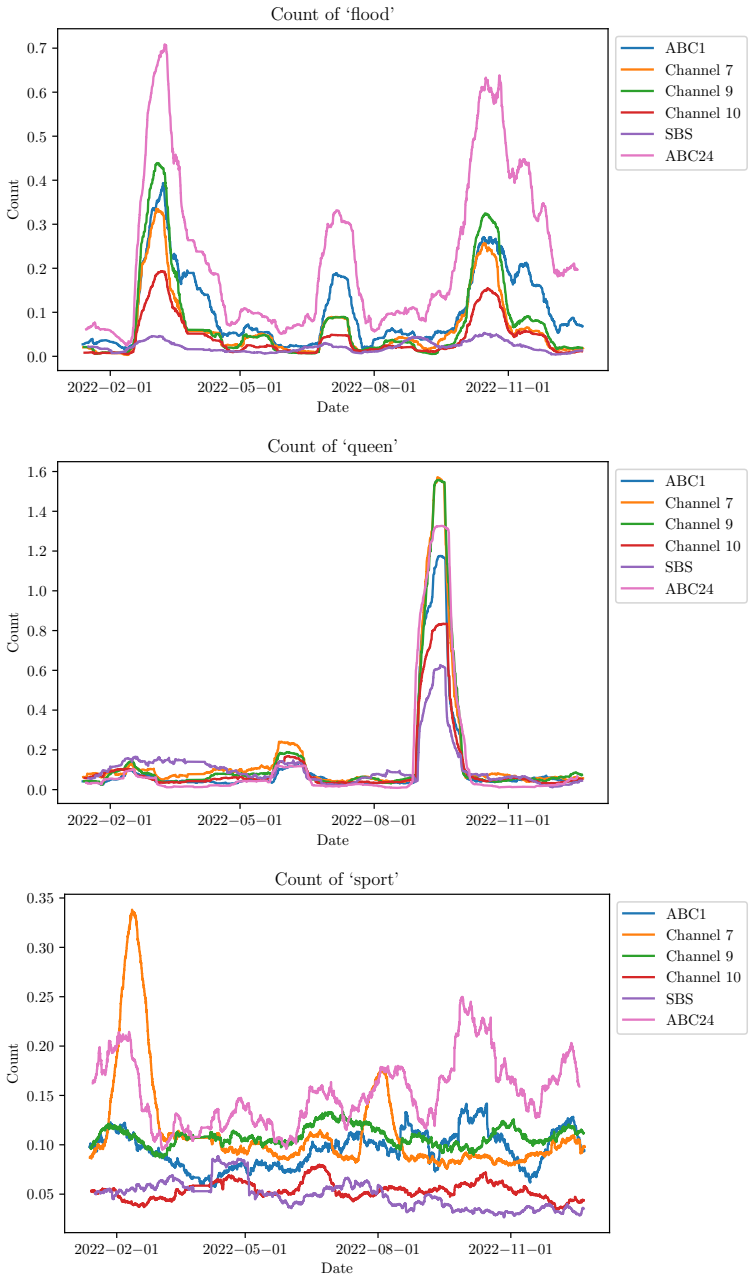

Figure 3.18: A moving average of the word counts over 2022. ABC24 generally has the highest word counts. Notice here that in particular the 'sport' topic is very different to that in Figure 3.20

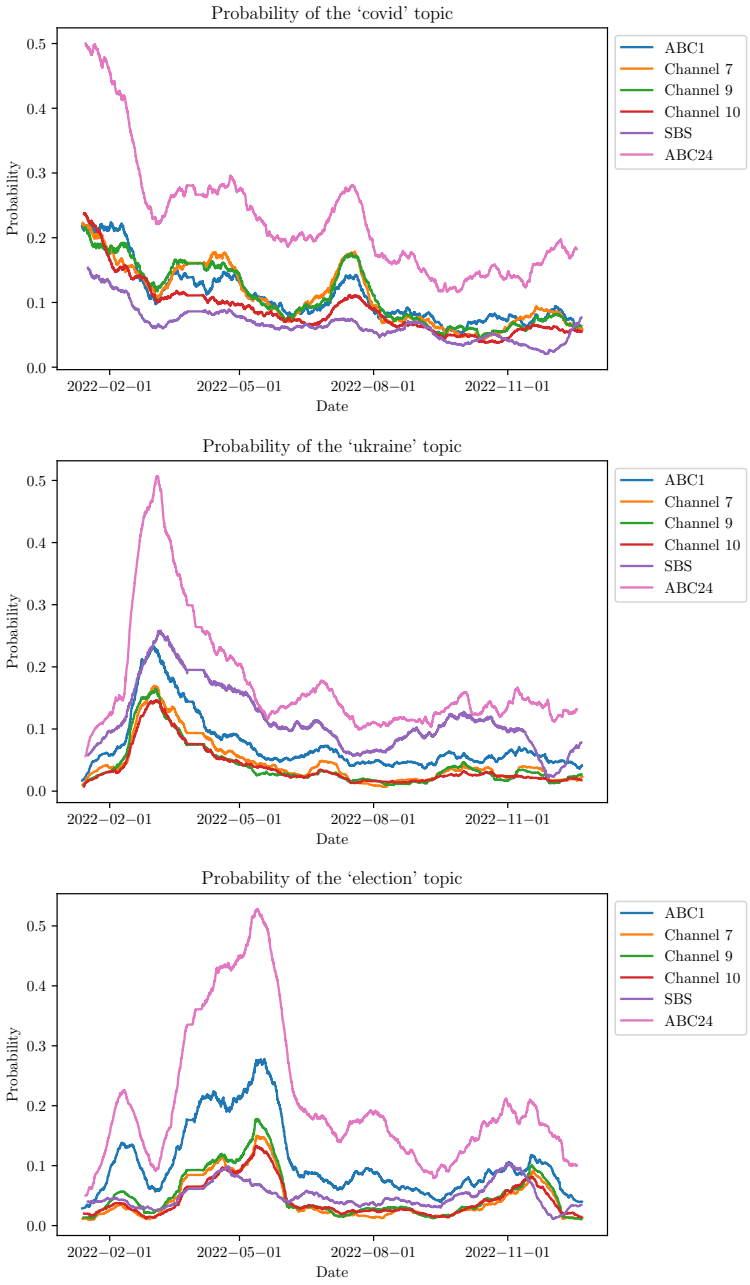

Figure 3.19: The probability of seeing various topics over 2022. ABC24 generally has the highest probability of showing news-related topics.

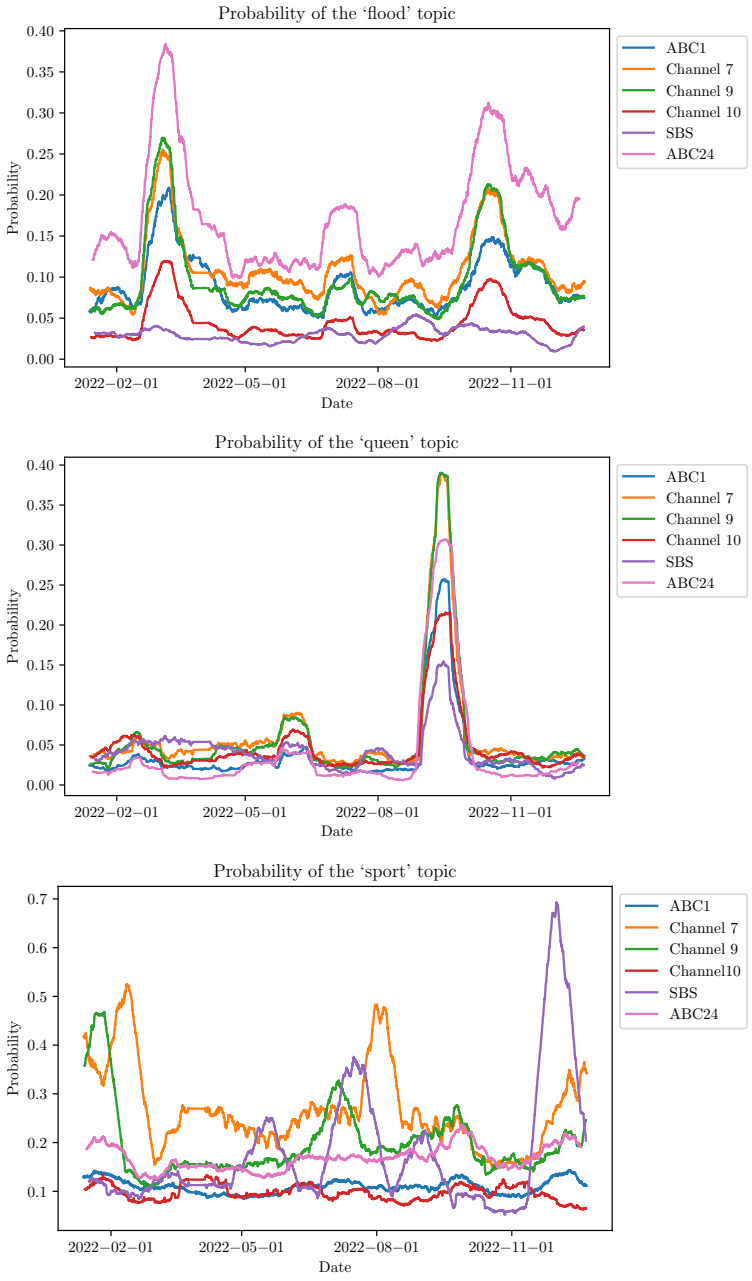

Figure 3.20: The probability of seeing various topics over 2022. ABC24 generally has the highest probability of showing news-related topics. Of interest is the 'sport' topic. This peaks at a different time for each channel, corresponding with the major sporting events shown. Channel 7, for example, has a weekly cycle throughout the AFL season, with a much larger peak around August corresponding with the Commonwealth Games.

sports broadcast on their channel. For example, SBS primarily featured major cycling events and the FIFA world cup. These align with with the three distinct peaks that we see: one each for the Giro D'Italia, the Tour de France, and the FIFA World Cup 2022.

Through this qualitative analysis, we conclude that the $p(y|\bar{x})$ term is a reasonable proxy for media attention and provides better estimates than a simple word count. We see distinct spikes in the probabilities around major events, as well as small peaks surrounding sporting events, and a rough correlation with COVID case numbers which are all good indicators of the quality of our measure.

3.8 Coverage bias

We now extend the analysis of media attention into a measure to quantify coverage bias. This gives an insight into how often each topic is mentioned in the media. We expect to see each channel have a coverage bias towards the topics that they show the most. This may be a sport that they have the rights to, or a topic that a certain channel is particularly interested in. For example, Channel 7 has the sole rights to broadcast AFL and so we expect it to have a higher coverage bias towards this topic. SBS is a multicultural channel, so we expect that they would have a coverage bias towards major events occurring overseas such as the war in Ukraine. Interestingly, each channel's 'sport' topic peaks at a different time. We start by again looking at the counts of the words used as anchors in the previous section.

The mean counts per document are given in Table 3.9. ABC24 has the highest counts for the majority of these terms. This is because it is a news channel and features more content on current events. Despite featuring very few programs from the sport genre, ABC24 has a high coverage bias towards sport due to its coverage of sporting events on news programs. Channel 7 has the highest count of the word 'queen'. This is because they had the rights to broadcast both the Platinum Jubilee and funeral of Queen Elizabeth II.

We therefore use the topic model to form another measure for coverage bias. Using (3.1), where N is the number of documents, we average the probabilities for various topics across all documents from 2022 to give us a coverage measure for 2022,

$$\beta_C = \sum_{\bar{x}} \frac{p(y|\bar{x})}{N}. \tag{3.1}$$

Table 3.10 shows the coverage measure for each channel and each topic that our model was trained on in Section 3.7.

We see that ABC24 has the highest coverage bias towards most of these topics. This is almost certainly because they are a dedicated news channel and as such would tend

	covid	ukraine	election	flood	queen	sport
ABC1	0.2312	0.1818	0.2153	0.1022	0.1129	0.0954
Channel 7	0.1906	0.1041	0.0723	0.0623	**0.1614**	0.1124
Channel 9	0.1922	0.0786	0.0792	0.0739	0.1557	0.1079
Channel 10	0.1527	0.0831	0.0736	0.0399	0.1066	0.0532
SBS	0.1223	0.1057	0.3287	0.0954	0.1081	0.0473
ABC24	**0.5280**	**0.4612**	**0.4312**	**0.2240**	0.1276	**0.1536**

Table 3.9: The mean word count per document. This is a very basic indicator of coverage bias on each channel. ABC24 has the highest counts for all words except 'queen'. In this case, Channel 7 had the rights to broadcast the Platinum Jubilee and funeral of Queen Elizabeth II and as such has a higher count.

	covid	ukraine	election	flood	queen	sport
ABC1	0.1100	0.0712	0.1020	0.0879	0.0386	0.1097
Channel 7	0.1100	0.0429	0.0406	0.1073	**0.0612**	**0.2754**
Channel 9	0.1102	0.0386	0.0478	0.0956	0.0596	0.2041
Channel 10	0.0915	0.0358	0.0412	0.0435	0.0477	0.0957
SBS	0.0667	0.1057	0.0485	0.0307	0.0405	0.1843
ABC24	**0.2350**	**0.1691**	**0.1978**	**0.1735**	0.0385	0.1735

Table 3.10: The average probability of each topic in each channel. This is an indicator of the coverage bias of each topic from each channel. ABC24 has the highest probabilities for the topics which commonly feature on the news, while Channel 7 has the highest probability of the 'queen' and 'sport' topics. Analysis in Section 2.2 can explain the differences in these probabilities.

to have a higher probability of each document containing a topic that is in the news. As before, Channel 7 and Channel 9 both have a higher coverage bias towards the queen topic, likely because they owned the rights to broadcast her funeral. Finally, we see that Channel 7 has the highest probability of sport. As we have seen before, Channel 7 has by far the most programs in the 'sport' genre. Behind ABC24, we also see that SBS has the second-highest coverage bias towards the 'ukraine' topic, while ABC1 has the second-highest coverage bias towards the election. This corresponds to the multicultural view of SBS and the dedicated federal and state election broadcasts on ABC1.

Note that this coverage measure can only be used as a comparative measure between different channels. This is because each topic naturally appears in different amounts throughout the corpus. If we simply state that any value over 0.1 indicates a coverage bias, we would conclude that almost all channels are biased towards sport, while this is not the case — it is simply a very common topic. We must therefore look at all of the channels as we have done and form comparisons between them to determine a relative coverage bias.

3.8.1 Political case study

An obvious form of coverage bias to now focus on is political coverage bias. We look at the two major Australian political parties so that we can easily compare between just two topics. These parties are the right-wing Liberal Party, and the left-wing Labor Party. To begin with, we naïvely look at the counts of words related to each party. We make a Liberal and a Labor list from the name of each party, as well as their two most recent leaders: Scott Morrison and Malcolm Turnbull, and Anthony Albanese and Bill Shorten respectively. These political lists are:

liberal	labor
scott	anthony
morrison	albanese
malcolm	bill
turnbull	shorten

We then count the number of words from each set in each document, c_i, and take the mean over the N documents,

$$\frac{1}{N}\sum_{i=1}^{N} c_i. \tag{3.2}$$

The mean counts for each channel are summarised in Table 3.11. We notice that all channels use words from the Labor set more than the Liberal set. This could simply be

Channel	Liberal	Labor
ABC1	0.3728	0.5834
Channel 7	0.1844	0.2803
Channel 9	0.2191	0.3376
Channel 10	0.1374	0.2774
SBS	0.1000	0.2399
ABC24	0.7425	1.1167

Table 3.11: The mean Liberal and Labor words per document for each channel. This very crude estimation of coverage bias indicates that all channels have a Labor 'bias'. This is the result of the Labor word list containing the common term 'bill' which can be used outside of a political context.

to do with how the words are being used. The word 'bill', for example, is not always used in a political context referring to Bill Shorten, but can also mean a document proposing a planned new law, or could also be referring to another person called 'Bill'. Our results could be skewed simply because 'bill' is commonly used out of the context that we expect.

We therefore instead choose to use a topic modelling framework for a metric which is less sensitive to the chosen seed words. We train a topic model on a 10% subsample of our data and anchor the terms 'liberal' and 'labor', as well as the names of each party's most recent leaders to separate topics to increase robustness. This is the list of words that we have counted previously.

Firstly, we plot the probability of seeing both of these topics in all channels from 2015–2022 in Figures 3.21 and 3.22. The difference between these values is plotted in Figure 3.23. These difference values correlate with the party in power. Until 2022 there was a strong Liberal coverage bias for all channels, mostly likely because they led the nation throughout this time. After May 2022, this strongly shifts to Labor, corresponding with their win at the federal election.

We now propose two new coverage bias metrics based on probabilities obtained from our topic model. For convenience, we write that the probability of topic 1 given document $\bar{x}_i$ is p_{1i}, and similarly for topic 2. Again, there are N total documents. The first metric looks at the mean of the differences (MOD) between the probabilities. This is given by

$$MOD = \frac{1}{N} \sum_{i=1}^{N} p_{1i} - p_{2i} \tag{3.3}$$

The second metric looks at the ratio of the sums (ROS) of the two probabilities. This is

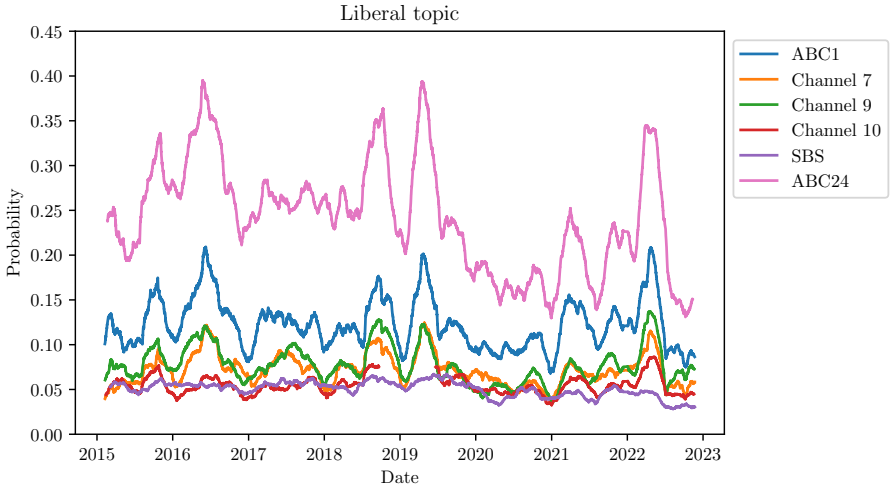

Figure 3.21: The probability of the Liberal topic for each channel from 2015 to 2022. ABC24 has by far the highest probabilities here because they are a news channel and feature a larger proportion of news content.

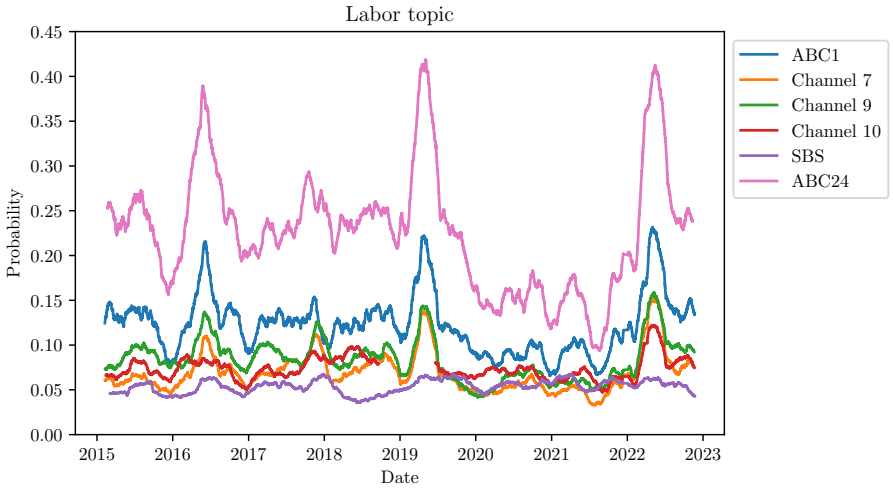

Figure 3.22: The probability of the Labor topic for each channel from 2015 to 2022. ABC24 has by far the highest probabilities here because they are a news channel and feature a larger proportion of news content.

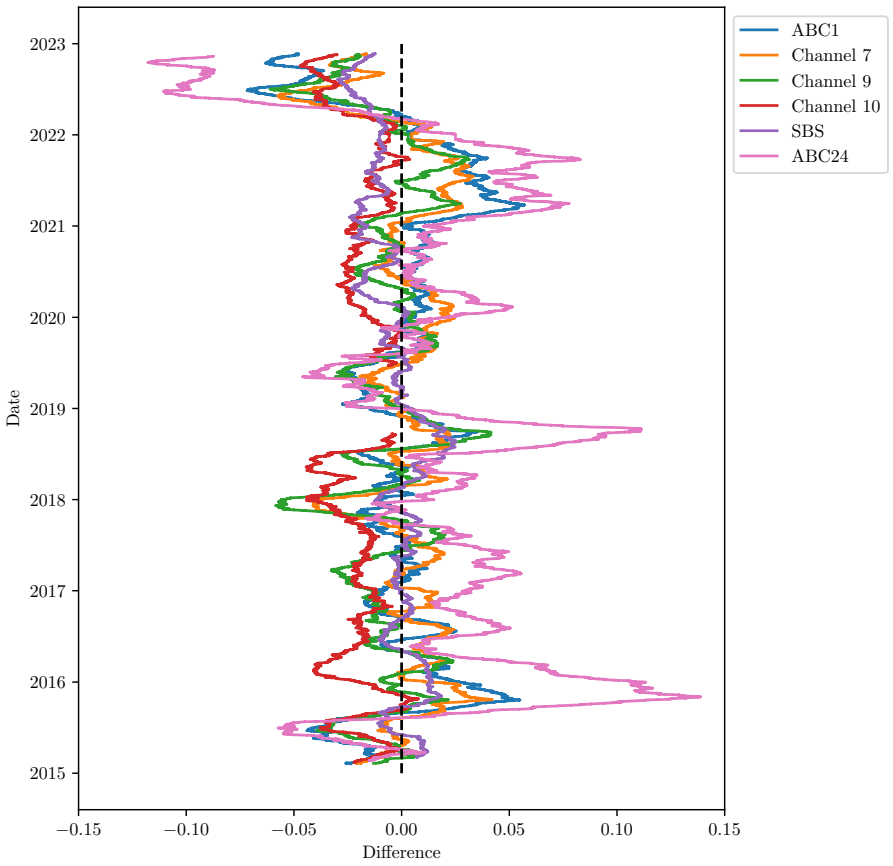

Figure 3.23: The differences in the probabilities of the Liberal and Labor topics from 2015 to 2022. A positive value indicates a Liberal slant, while a negative value indicates a Labor slant. The plot has been rotated for clarity, and to align with the left-wing and right-wing stances of the Labor and Liberal parties respectively. There is a clear shift towards Labor in May 2022. This corresponds with Labor's win in the 2022 federal election. Also note the spikes towards the Liberal party in 2015 and 2018. These correspond with a change in leadership and the media attention surrounding this.

Channel	Word count bias	MOD	p-value MOD	ROS	p-value ROS
ABC1	−0.2106	−0.0003	0.0010	0.9636	0.0010
Channel 7	−0.0960	0.0021	0.0010	0.9975	0.0010
Channel 9	−0.1185	−0.0062	0.0010	0.8922	0.0010
Channel 10	−0.1400	−0.0191	0.0010	0.7065	0.0010
SBS	−0.1399	−0.0036	0.0010	0.8999	0.0010
ABC24	−0.3742	0.0133	0.0010	1.0258	0.0010

Table 3.12: A comparison of each bias metric for the length of time from 2015 to 2022. A positive MOD value indicates a Liberal coverage bias, while a negative MOD value indicates a Labor coverage bias. An ROS value greater than 1 indicates a Liberal coverage bias, while an ROS value below 1 indicates a Labor coverage bias. Note that this is still a naïve estimator for bias, as it only reports the focus of attention and not whether this attention is positive or negative.

$$ROS = \frac{\sum_{i=1}^{N} p_{1i}}{\sum_{i=1}^{N} p_{2i}}. \tag{3.4}$$

In each case, a higher value indicates a more Liberal-leaning coverage bias, while a lower value indicates a more Labor-leaning coverage bias. In the first metric, a Liberal coverage bias is indicated by a number greater than 0, whereas in the second metric, this is a number greater than 1. These metrics should be less dependent on the words that were chosen than the previous count metric, improving the reliability of our results.

We use the topic model trained on the Liberal and Labor word sets given previously. We train the topic model 30 times using different seeds to obtain a reliable estimate for the $p(y|\bar{x})$ values. The values of the MOD and ROS are given in Table 3.12, along with p-values. Empirical p-values were found using the method described in Section 1.8, by randomly selecting one sixth of the documents and computing the MOD and ROS for these. This was repeated 10,000 times and p-value is the proportion of values that were more extreme than the values found.

We also compare the MOD and ROS with the difference between the count values in Table 3.11. The topic modelling metric gives quite different results to that of the original count metric. This is because it does not take into account just the count of a strict set of terms; the probability is instead calculated with a more complicated expression including marginal probabilities. It is therefore less dependent on the seed words than the previous method and should theoretically provide a better estimate of bias than the counts alone.

We cannot, however, determine true biases from these values. They do tell us how prevalent a topic is within a channel and we can compare this between channels. This is a good estimate for the coverage bias in a channel. It would, however, would be

unreasonable to use this as a true estimator for bias since this measure doesn't actually look at the sentiment of the text. A channel may be speaking a lot about a certain topic but in a negative context, which our current measure would interpret as a positive bias. In order to form a reasonable measure for bias, we must further examine the content of the text. This investigation is the topic of the next chapter.

3.9 Conclusion

This chapter has provided an in-depth analysis of CorEx topic modelling. We have shown that for our dataset, CorEx outperforms LDA. CorEx also has the advantage of easily anchoring selected words which has proven to be incredibly useful in this analysis. We explored the parameters used in the CorEx implementation, their effect on the topic model, and the optimal parameters for our particular use. In our case, we have shown that 5-minute intervals perform the best for the media coverage method that we have used. Additionally, we investigated the number of topics that optimises the total correlation of the model. We conclude that this does not have a large impact on the results, however we chose to use a value of 40 topics throughout.

We first found a reliable method for generating topic models with limited memory and computation constraints. Training our model on a subsample of just 1 percent of our data was shown to produce results incredibly similar to those obtained from a model trained on all data. This provides us with more flexibility on the amount of data that we are able to analyse and allows us to produce topic models across all of our data using a subsample.

We have compared channels to once again understand differences in their broadcasting. To do this, we compared topic models trained on each channel using a topic similarity measure. We then separated each topic into sub topics with a hierarchical structure providing us with a clearer view of the various sports featured on each channel.

We have shown that the $p(y|\bar{x})$ term obtained from a topic model is a reasonable proxy for media attention and were able to plot this over time. We see that media attention closely follows major events, and from a close analysis of the COVID topic, we see a general drop in attention over time when the topic remains constant.

Finally, we investigated a measure for bias towards particular topics by looking at the average probability of the topics for each channel across a time interval. We extend this concept to form a measure of political bias which compares Liberal and Labor topics. Our measure is naïve and in its current form simply determines which party appears more. In order to establish a rigorous measure for political bias, we should examine the actual content of the text to determine whether each party is being discussed in a positive or negative way. We will further investigate the text using a sentiment analysis to develop a more robust measure for bias.

Chapter 4

Sentiment analysis

4.1 Introduction

In this chapter, we extend the coverage bias measure presented in Section 3.8 to one which analyses both the content and sentiment of text. By performing a sentiment analysis, we can determine the positivity of text from a particular topic which allows us to measure statement bias. When combined with the topic modelling from Chapter 3, this forms a measure that considers two different aspects of bias.

Sentiment analysis has been widely applied to news media in the form of both news articles and social media [43, 48, 68]. Television, however, has been very rarely studied. This chapter aims to analyse Australian television captions, and specifically news programs, to gauge the sentiment of the general population as people respond to major events.

Section 4.2 presents a general sentiment analysis using the NRC-VAD lexicon to provide a summary of the change in sentiment of the entire corpus over time. In Section 4.3, we narrow our analysis to just text from the news genre. This allows us to understand the positivity of real-life events without noise from irrelevant programs. Throughout this chapter we implement word shift analyses [15] discussed in Section 1.7 to understand why any differences in sentiment between channels or across different time periods may occur. This reveals several contextual problems with the NRC-VAD analysis which we propose can be rectified with a sentiment lexicon trained on our corpus of news text.

In Section 4.4, we develop a domain-specific sentiment lexicon by fine-tuning GloVe word embeddings on our text and using the method from Cochrane et al. [10] to generate sentiment scores from these embeddings. Fine-tuning on our specific corpus resolves some issues encountered with the NRC-VAD lexicon. The scores in the new lexicon are adjusted according to the positive or negative context surrounding them in this corpus. In addition, words specific to the corpus such as 'covid', and 'morrison' that are not found in the NRC-VAD lexicon are given a sentiment. Section 4.5 details

the outcomes of a new sentiment analysis on news text usng this lexicon.

We further constrain the sentiment analysis to just political text. In Section 4.7, we discuss the selection of text corresponding to the two major political parties. We perform a sentiment analysis, and then explore methods for comparing the sentiment from news programs discussing the two parties. The difference between Liberal and Labor sentiment provides a reasonable estimate for statement bias over time.

Another simple measure for statement bias can be found by fine-tuning the GloVe embeddings on text from each channel individually. Section 4.8 investigates the sentiment assigned to political terms by each channel's lexicon as an alternative measure for statement bias.

Finally, Section 4.9 presents a bias measure that combines both the coverage and statement bias of topics. In addition to the amount of attention that each topic receives, we now analyse the positivity of this attention, enabling us to capture a larger portion of the bias present. We find that this is somewhat limited by the low weighting of sentiment in the measure. Future research should focus on assessing this bias measure on a labelled testing set to quantitatively determine its accuracy.

Section 4.10 compares the bias with opinion polling data over time. Our findings indicate that a weighted sentiment of the two political parties roughly aligns with public voting preference at a given time. This demonstrates the close relationship between television media and public sentiment.

In summary, the contributions of this chapter are:

- To our knowledge, the first sentiment analysis of the Tveeder data set.
- A sentiment lexicon developed specifically for the analysis of Australian television news text.
- A method to analyse statement bias using a sentiment lexicon.
- A measure of general media bias combining both coverage and statement bias.
- A comparison of television media and polling data to determine their relationship.

4.2 Preliminary sentiment analysis

We begin by performing a sentiment analysis on all text from 2015 to 2022 using the valence scores from the NRC-VAD lexicon [32] discussed in Section 1.4. Recall that the NRC-VAD valence scores lie between 0 and 1, where 0 indicates a very negative valence, and 1 indicates a very positive valence. We use the NRC-VAD lexicon with no additional rules due to its simplicity, large lexicon size, and fast computation time. As the document size is increased, additional rules such as negation have less impact on the calculated sentiment of each document. For our relatively large documents, this is not worth the added complexity and computation time.

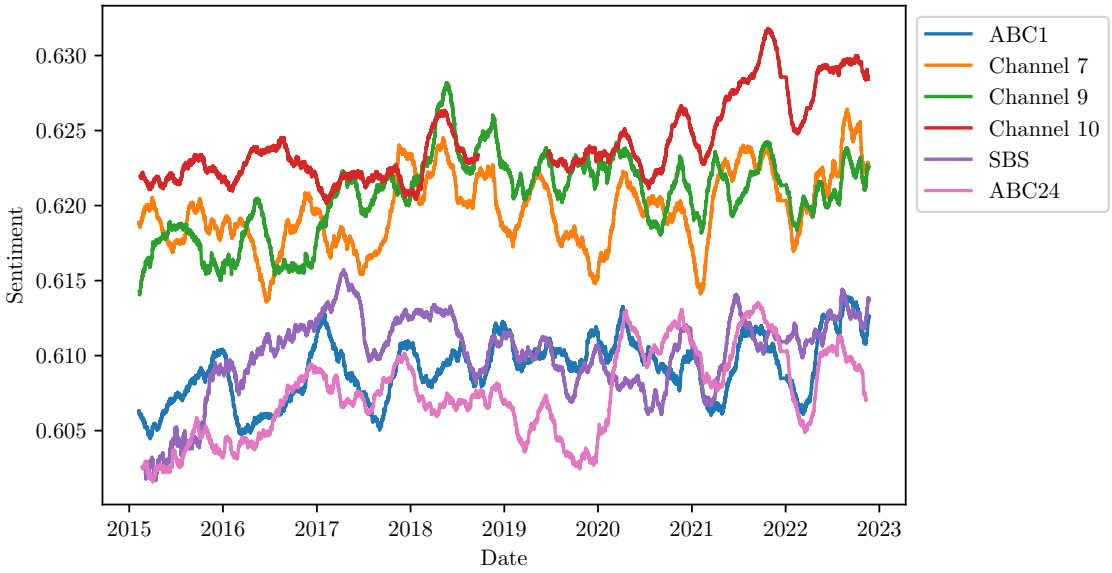

Figure 4.1: A sentiment analysis of all channels from 2015 to 2022. Channels 7, 9, and 10 have a significantly higher sentiment than the others. There is little change in sentiment over time.

The text is split into documents of 5 minute intervals for consistency with the topic modelling in Chapter 3. This allows us to easily form comparisons with the topic modelling and apply results found previously. The sentiment of a document is calculated from (1.15), by taking the mean of all sentiment scores of the individual words in the document.

The sentiments of documents from all stations are plotted in Figure 4.1 to compare their positivity. Channels 7, 9, and 10 have a much higher sentiment than the other three. ABC24, the news channel, has the lowest sentiment, especially in 2018 and 2019. This is what we expect, as the majority of programs shown on the channel are in the news genre. Section 2.2 shows that this genre typically has a more negative sentiment.

Using Gallagher's Shifterator Python package [15] which implements the word shifts discussed in Section 1.7, we explore the terms that cause the significant difference in sentiment between the most positive and negative channels, Channel 10 and ABC24. The words with the greatest difference in Tsallis entropy are given in Table 4.1. This allows us to compare the text usage between channels. Recall that this can give a higher weight to infrequent words by reducing the value of α. In this instance, we set α to 0.3 to emphasise rare words which generally carry more meaning and characterise each channel. The difference in Tsallis entropy is calculated using (1.21).

Many word shifts can be easily explained by differences between the programs that we have analysed in Section 2.2 The terms 'judge' and 'judy' both appear significantly

Channel 10	judy, you, oh, gonna, okay, yeah, laughter, your, ok, im, applause, dish, sauce, alright, nice, judge, i
ABC24	government, abc, minister
Channel 10	good, love, like, know, get, laughter, nice, applause, do, right, great, beautiful, can, laughs, want, thank, perfect, amazing, look, life
ABC24	

Table 4.1: The top 20 words with the highest difference in Tsallis entropy between Channel 10 and ABC24 text are shown in the top two rows. Words used on Channel 10 align with common programs, such as Judge Judy, and genres, such as cooking, which we found in Section 2.2. ABC24 uses political terms which commonly appear in news programs. The bottom two rows show the top 20 words with the greatest impact on the higher sentiment in Channel 10 than ABC24 text. Positive terms are indicated in orange. The difference in positivity is primarily the result of a higher number of positive terms on Channel 10, rather than more negative terms on ABC24. Words used more on Channel 10 such as 'laughter' and 'applause' are the result of programs with live audiences and a greater number of comedy programs contributing to the significantly higher sentiment.

more frequently on Channel 10. This stems from the program *Judge Judy* which is broadcast on Channel 10. The terms 'government', and 'minister' appear significantly more on ABC24, the result of a higher proportion of news programs which commonly reference politics. Also note that 'laughter' and 'applause' are used more on Channel 10 where many programs have a live audience or contain a laugh track. As ABC24 is a news channel, it does not broadcast these types of programs. Channel 10 is also characterised by more informal language such as 'oh', 'gonna', and 'yeah', as well as more first-person language, using words such as 'im', and 'i'. We also see the cooking terms 'dish', and 'sauce'. Again, these terms do not appear on ABC24 as the channel is more formal and impersonal and does not show cooking programs.

The words with the greatest impact on the higher sentiment of Channel 10 compared with ABC24 are also shown in Table 4.1. Recall from Section 1.7 that the sentiment shift considers both the frequency of each word and its individual sentiment. The difference in weighted sentiment is calculated using (1.22). The higher sentiment of Channel 10 compared with ABC24 can be attributed to both a higher use of positive terms on Channel 10, and a higher use of negative terms on ABC24. The top 20 terms contributing most to the difference in sentiment are higher valence words on Channel 10. Positive terms such as 'good', 'love', and 'like' are given a high valence (0.938, 0.927, and 0.719 respectively) and as such even a small difference in their usage can have a large impact on the sentiment. Additionally, we observe that the terms 'laughter' and

February 2018	canada, olympics, scotty, ski, barnaby, skating, medal, sochi, pyeongchang, olympic
Used more	gold, medal, top, champion, sports, sport, speed, skiing, medallist, athlete, skating, team, event, music
Used less	ill, kick, pandemic, police, ground, sorry
January 2021	bowling, innings, scorchers, wickets, overs, bat, capitol, inauguration, wicket, boop
Used more	quarantine, bat, shot, hit, test, short, rob, pandemic, cricket, down, lost, impeachment, wicket, vaccine, end
Used less	love, thank, know, christmas, family
September 2022	rsv, fetterman, sunak, halloween, maribyrnong, echuca, inflation, flood, verse, pepsi
Used more	music, thank, million, live, applause, laughter, water, worth, cheering, correct, value, welcome, fresh, living, money
Used less	final, shot, kick, end, little

Table 4.2: The top 10 words used more often during periods of extreme sentiment on Channel 7 relative to text from the same channel in the time period from January 2015 to December 2022. A sentiment breakdown is also included, listing the top 10 positive and negative-swinging words. Positive words are shown in orange, while negative words appear in blue. The terms roughly correspond with events at a given time, although there is a lot of noise and it is difficult to discern exactly what is going on. January 2021, for example, contains many terms relating to cricket, a sport which is shown often on Channel 7. This obscures important events occurring at this time, such as the January 6 Capitol riots.

'applause' that we previously identified as being more prevalent in Channel 10 text play a large role in increasing the sentiment.

We further analyse the change in sentiment over time. Restricting to just text from Channel 7 to remove noise caused by different channels, we analyse three months of extreme sentiment in Table 4.2. We show the top 20 words which have the greatest impact on the positive or negative sentiment in each month. A month can be positive due to either a relative increase in positive terms or a relative decrease in negative terms.

Several major events are visible here, although they are obscured by generic text from irrelevant programs. The terms appearing relatively more often in February 2018 clearly point to the winter olympic games broadcast on Channel 7 at this time and allow us to understand the positive sentiment here. Most words that appear more frequently in January 2021 are related to cricket, a sport shown often on Channel 7. This has obscured negative sentiment from other events such as the January 6 capitol riots

in the United States of America. The terms ‘applause’, and ‘laughter’ also contribute greatly to a high sentiment in September 2022 despite devastating floods occurring on the Murray river. Words such as these obscure real-world data and reduce our ability to detect major events occurring outside of the television.

4.3 Restricting to news text

To gauge the sentiment of real-life current events without interference from fictional television programs, we filter out non-news programs. We treat these programs as noise; text from them is irrelevant if we wish to anlayse the sentiment of real-world events. Hence, we limit the scope of our sentiment analysis to focus solely on text from news programs. This approach enables us to track changes in the sentiment of real-world events and, by extension, the sentiment of television news media organisations, without noise from other programs. Note that Channel 10 has fewer programs in the news genre because of the way the programs were labelled prior to 2018. We are, however, able to find a reasonable number of news programs by filtering for titles containing ‘news’.

A sentiment analysis performed on news text using the NRC lexicon is shown in Figure 4.2. We see a clear increase in sentiment, especially in ABC1 and ABC24 text, at the beginning of 2020. We contrast the sentiment of news text from ABC1 with the sentiment of all text in Figure 4.3. The news text generally has a slightly more negative sentiment and contains more significant peaks and troughs, especially in 2020 and 2021. To determine the cause of these spikes, we make another word shift comparison.

The Tsallis entropy difference between ABC24 text from 2020 and 2021 and all other text with a value of $\alpha = 0.3$ is shown in Table 4.3. Most terms with the greatest relative difference in usage are related to the COVID-19 pandemic. Words such as ‘coronavirus’, ‘quarantine’, ‘lockdown’, and ‘vaccine’ are used much more in 2020 and 2021. With prior knowledge, we would generally associate these with a negative sentiment. Of these terms, only ‘quarantine’ and ‘vaccine’ are given a valence score in the NRC-VAD lexicon (0.16 and 0.5 respectively).

To understand the reason behind the shift in sentiment between the two dates, terms with the greatest impact on positive sentiment in 2020 and 2019 are also given in Table 4.3. The word ‘health’ has the second highest impact on the positive sentiment of the 2020/2021 text with a sentiment score of 0.935. Similarly, ‘positive’ also has a large impact, despite being primarily used to indicate COVID-positive patients during this time. These words would generally appear negatively in the context of COVID-19 despite its high valence score and as such the sentiment may be incorrectly positively skewed here. Many other terms with a high difference in entropy relating to COVID-19 are absent here as they do not appear in the sentiment lexicon. The neutral terms ‘new’, ‘have’, and ‘victoria’ also have a large impact on the sentiment. These are given

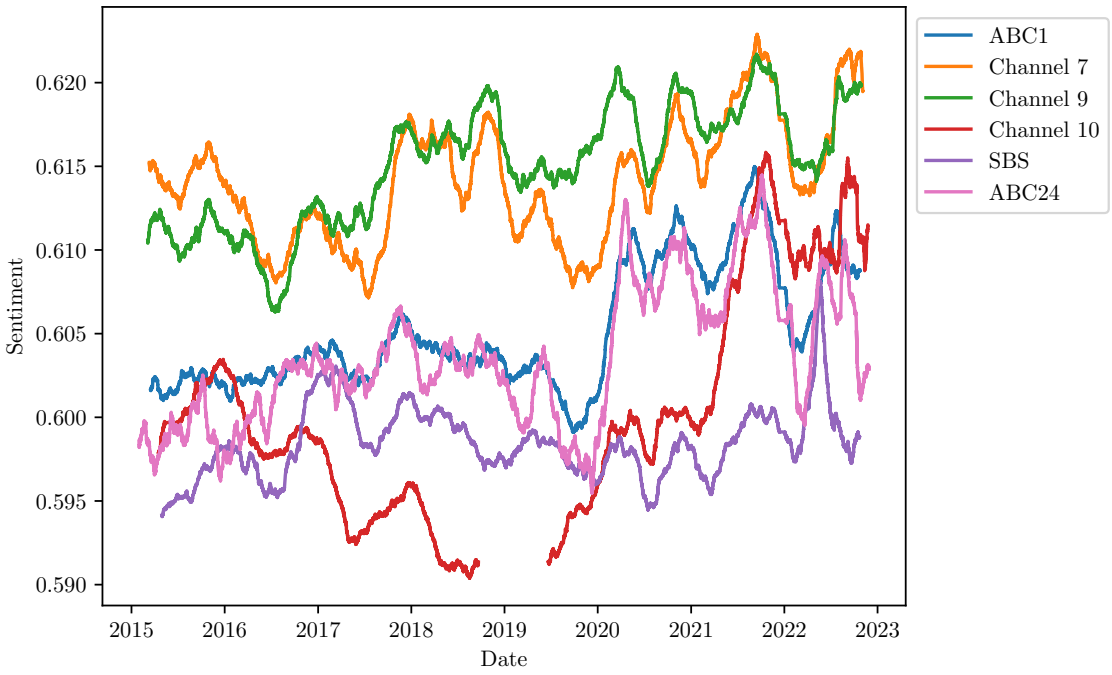

Figure 4.2: The sentiment of text from each channel pertaining to the news genre. Note the increase in sentiment of ABC1 and ABC24 text in 2020 and 2021 due to COVID-19 and health-related terms.

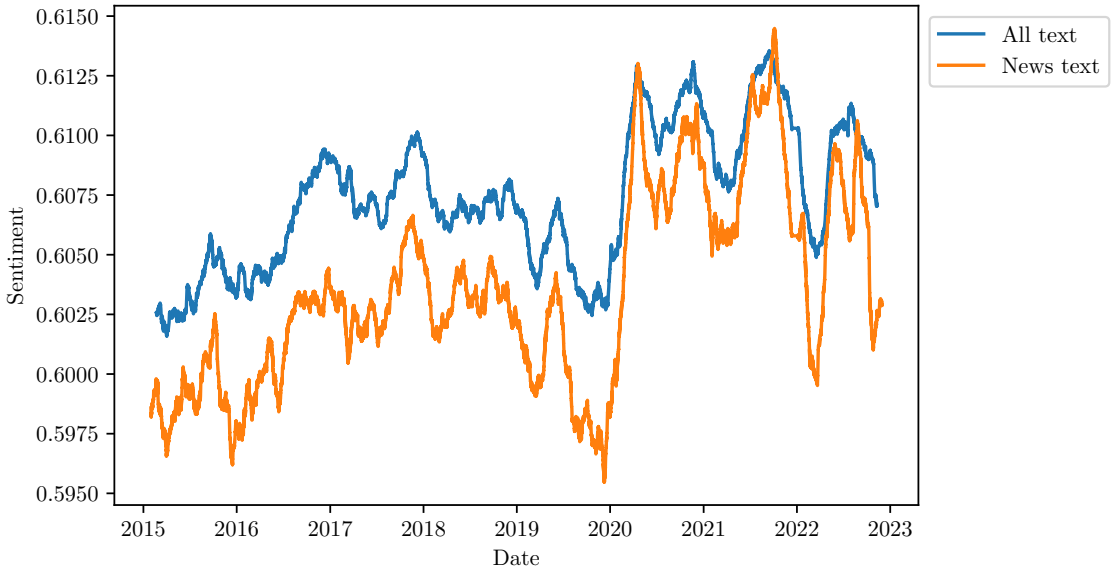

Figure 4.3: A comparison of the sentiment of ABC1 news text, and all ABC1 text. The sentiment restricted to news text is generally more negative and has more significant peaks and troughs.

valence scores of 0.917, 0.757, and 0.896 respectively despite their typically neutral usage.

The term 'party' also has a large impact on the sentiment, although in the other direction. It is a positive word which is used significantly less during this time period. Outside of a political context, the term 'party' may be considered extremely positive as it is used to indicate a social gathering. In the context of politics, however, it is generally a neutral word indicating a political organisation. The NRC-VAD lexicon rates 'party' as 0.948 on a valence scale from 0 to 1 which is overly positive for the political context that we intend to eventually analyse. Note that we have investigated peaks on other channels and found similar issues although in-depth analyses of these are not included here.

To rectify these problems, we should develop a sentiment lexicon that is tailored to our specific corpus. Doing so will allow new terms such as 'coronavirus' to be assigned a sentiment score, and the sentiment of other health-related terms will be reduced accordingly. Other corpus-specific words such as 'labor', and 'morrison' will also be assigned sentiment scores which will greatly benefit the analysis of political text. We also hope to see neutral terms such as 'party' and others that appeared in our word shift analysis obtain a more neutral score reflective of the context that we are analysing. Positive terms such as 'happy' should remain with a similar sentiment since their usage does not change in our corpus.

2020/2021	coronavirus, quarantine, lockdown, covid, vaccine, cases, restrictions, virus, astrazeneca, pandemic, vaccinated, pfizer, jobkeeper, tested, distancing, vaccination, cluster, outbreak, vaccines
Other dates	turnbull
2020/2021	new, health, have, victoria, be, positive, will, get, home, people, travel, premier, community, able, thank, care, hotel, today, know
Other dates	war

Table 4.3: The first two rows show the top 20 words with the highest difference in Tsallis entropy between ABC24 in 2020/2021 and other dates. In the top cell are words used relatively more often during 2020 and 2021, while the lower cell contains words used relatively more often at other times. Most terms used relatively more in 2020 and 2021 are related to the COVID-19 pandemic, however many of these terms do not appear in the sentiment lexicon. The bottom two rows show the top 20 words with the greatest impact on the difference in sentiment between 2020/2021 and other dates. Positive terms are shown in orange, while negative terms are shown in blue. Many terms are words which we may consider to be neutral, but have a very positive valence score in the NRC-VAD lexicon.

We do this using the process outlined by Cochrane et al. [10] and presented in Section 1.4.3. Recall that this involves obtaining word embeddings from text and using (1.16) to build a sentiment lexicon.

4.4 Creating a domain-specific sentiment lexicon

Previously, we demonstrated how our sentiment analysis was skewed by some very positive words such as ‘health’ and ‘positive’ which were often used in a neutral or negative context. Other terms generally considered to be neutral were also given excessively high sentiments. We also observed that important terms such as ‘coronavirus’ and ‘morrison’ were not given sentiment scores in the NRC-VAD lexicon. We now aim to rectify this by creating our own lexicon specific to news text. With this lexicon, we aim to:

- adjust the sentiment of terms to accurately reflect the context that they are generally used in;
- generate sentiment scores for important terms in this corpus which may not appear in generic sentiment lexicons.

4.4.1 Calculating sentiment scores

To ensure that we obtain high-quality word embeddings, we first begin with high-quality word embeddings and then fine-tune them on our corpus. This approach ensures a good standard of embeddings without the need for a corpus containing billions of words and extensive computational resources.

GloVe embeddings [35] are word vectors that have been pre-trained on an enormous corpus. These provide a good place to start. Several GloVe embeddings have been trained, each on different types of data. We will be using 300-dimensional GloVe embeddings trained on 6 billion tokens from Wikipedia. Wikipedia has a neutral point of view policy [63] and as such is an appropriate starting point for our more neutral data.

Given the GloVe embeddings, we then use the Mittens approach [12] to fine-tune on our data. Recall from Section 1.6 that this is a method of producing word embeddings specific to a particular corpus without needing to train new embeddings from scratch.

The GloVe embeddings are fine-tuned on the subset of our corpus consisting of news programs. We generate embeddings for the most common 30,000 words which all appear at least 150 times in the corpus, enough to produce reasonable embeddings. Many of the words are domain-specifc, such as 'covid', 'morrison', and 'albanese', and do not appear in NRC-VAD or most other traditional sentiment lexicons. The inclusion of these new terms is critical in the analysis of the text.

We implement the method for generating sentiment lexicons from Cochrane et al. [10]. Using (1.16), we calculate a sentiment value for the 30,000 words with word embeddings. We will refer to this new lexicon calculated from the Mittens embeddings as the *Mittens lexicon*.

4.4.2 Robustness

We first check the effect of the randomness within the model to ensure that the embeddings are robust. Recall that words that are not in the original GloVe vocabulary are given a random initialisation. A gradient descent is then performed as part of the embedding process. As the loss landscape is non-convex, there is no guarantee that different initialisations will give identical embeddings, and thus sentiment scores. A different random initialisation affects not only these words but also words that already have a GloVe initialisation. This is because the embeddings are trained relative to one another and their location may be manipulated by other words which have this random initialisation. Since there are almost 800 words in the Mittens lexicon that don't appear in the original GloVe model, this could have a large impact on the embeddings of all words.

Models with different seeds will have an identical vocabulary as the 30,000 most com-

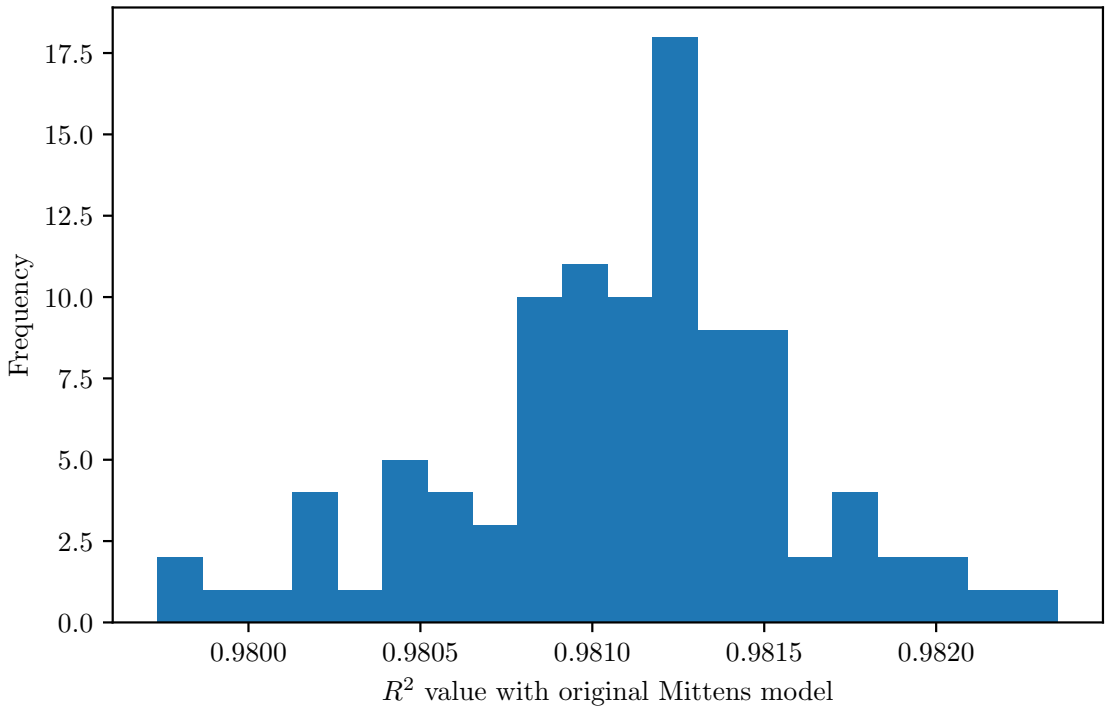

Figure 4.4: The Pearson correlation of Mittens models with seeds from 1 to 100, with the original model with seed 0. The Mittens models have low variation between seeds.

mon words do not change. These terms may appear in different locations in the vector space, however, due to the randomness.

The Pearson correlation values between the sentiment from models with different seeds, and the sentiment from the original Mittens model (with seed 0) are shown in Figure 4.4. The Pearson correlations lie between 0.97 and 0.99. The low spread and high correlation indicate that the random initialisation does not have a large effect on the outcome of the sentiment. It may be that the locations within $\mathbb{R}^{300}$ are different, however they are almost identical with respect to the positive and negative seed word embeddings.

4.4.3 Comparison with NRC-VAD and other sentiment lexicons

We now compare the Mittens sentiment lexicon with the NRC-VAD valence lexicon to determine whether we have rectified the issues from Section 4.3. Recall that the valence scores lie between 0 and 1. The positivity scores that we have calculated theoretically lie between -14 and 14, however we find that in practice they lie between -1 and 1.5.

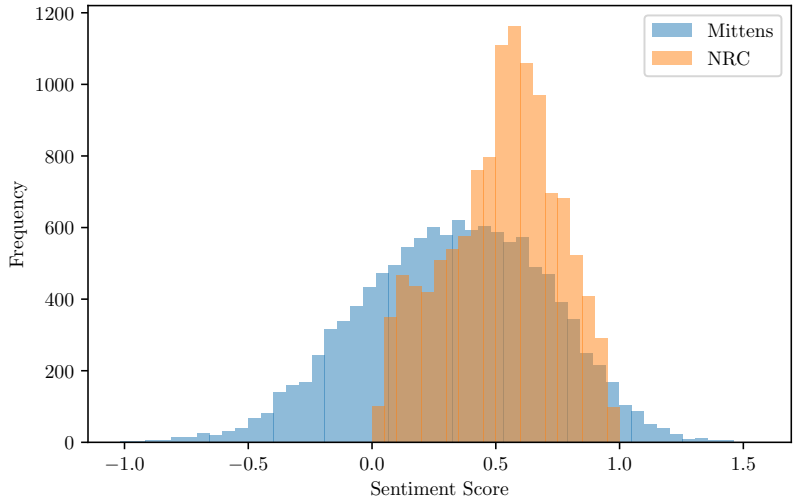

Figure 4.5: A histogram comparing the distributions of the NRC and Mittens sentiment scores. The NRC scores are left-skewed, whereas the Mittens sentiments appear approximately normal. The Mittens sentiments range from approximately -1 to 1.5 while the NRC sentiments are between 0 and 1.

We compare the two lexicons by first looking at histograms of the distribution of scores in Figure 4.5. Since the two lexicons contain different sets of words, we filter to just the words that the lexicons have in common. The union of these sets contains 12,690 words and doesn't include newer terms or proper nouns such as 'covid', or 'anthony' that our fine-tuning has picked up.

The distributions of the fine-tuned GloVe sentiment values and the NRC valence scores are similar. The NRC sentiments are slightly left-skewed and could almost be considered bimodal, whereas the Mittens sentiment scores, as the result of how they were generated, are much more symmetric. The mean of the Mittens lexicon is 0.3311, indicating a slight positive bias in the lexicon that is consistent with other research [13, 24]. This, however, could also be due to the choice of seed words, so that words generally lie closer to the positive seed words than negative seed words. The mean of the NRC lexicon is 0.5214.

The two sentiment scores are also directly plotted against one another in Figure 4.6 to analyse their correlation. We are not necessarily looking for a high correlation here, since the point of this exercise is to have a difference in scores. A small correlation would, however, indicate that our sentiment values align with human-rated scores to an extent.

The Pearson correlation between the two sets of sentiment scores is 0.4700 indicating

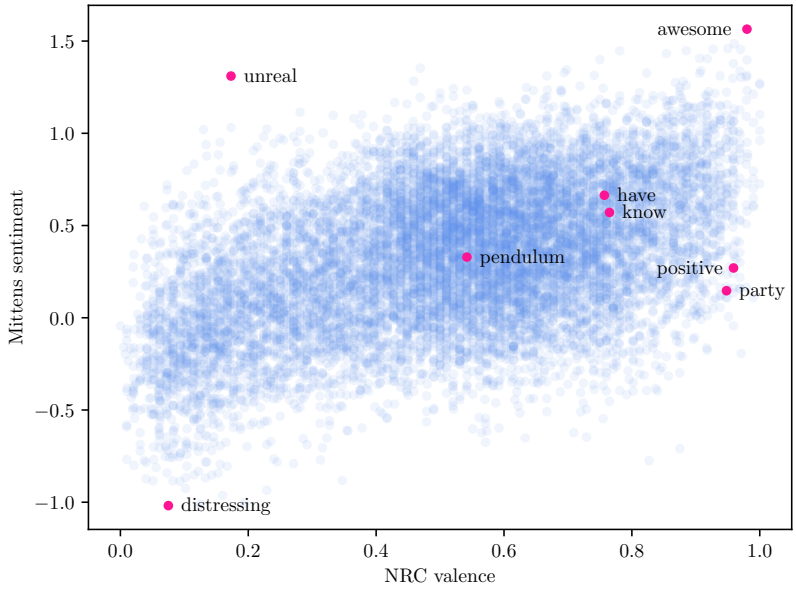

Figure 4.6: A comparison of the sentiment scores given by Mitten embeddings and NRC. The location of selected words are shown in magenta. The Pearson correlation between the values of the two lexicons is 0.4700 suggesting a weak correlation. Although there is some similarity, this shows that Mittens has adjusted the sentiment for our particular corpus.

Higher in Mittens	one, back, down, little, go, ill, way, bit, can, two, come
Lower in Mittens	have, think, know, new, family, people, will, good, thank

Table 4.4: The top 10 words with a higher and lower sentiment in the Mittens lexicon than in the NRC lexicon. The sentiments of some previously identified overly positive terms have been reduced in the Mittens lexicon.

a very weak correlation. This indicates that there is some similarity between the Mittens lexicon and the NRC-VAD lexicon although, as we had hoped, the new lexicon has adjusted the scores to be more appropriate for the Tveeder corpus.

We use a word shift analysis to investigate the sentiment changes that have the greatest effect on the calculated sentiment of the text. The weighted average shift from the NRC-VAD lexicon to the Mittens lexicon is shown in Table 4.4. We normalise the terms to form an accurate comparison of where words lie relative to others on the same sentiment axis. The terms 'have' and 'know' that we discussed earlier have had their sentiment shifted down to relatively more neutral values. The sentiment of these terms have some of the greatest impacts on the calculated sentiment of the overall text.

One of the words with the highest impact on the sentiment calculated previously was the term 'party'. The positivity of this word in the NRC-VAD lexicon was one of the reasons for creating a new domain-specific lexicon. The sentiment has shifted down, and it now has a sentiment of 0.1465, slightly negative for this lexicon. The sentiment of the term 'positive' itself has also been decreased to 0.2699 which is slightly negative. This reflects its common use in a negative COVID-19 context.

New terms such as 'covid' and 'coronavirus' have been given sentiments of 0.1664 and 0.0016 respectively. The creation of a domain-specific lexicon has given us the ability to interpret these previously unknown words and the negative sentiment associated with them.

It is worth noting that this automatically generated sentiment lexicon may not provide accurate values for the sentiment of individual words. It is better suited to analyse the sentiment of larger documents where terms are used in context. The sentiment of the term 'ukraine', for example is -0.1553. This does not necessarily mean that 'ukraine' is a negative word, rather it is commonly used in a negative context. When news reporters mention 'ukraine', this will likely be in a negative context, and so the sentiment of a document containing the term will be brought down accordingly. Terms such as this have more of a contextual sentiment which is useful in the analysis of news reports despite the sentiment of individual words potentially appearing incorrect.

4.5 Another sentiment analysis

We now perform another sentiment anlaysis on the news text only, this time using the Mittens lexicon. The sentiment over time is plotted in Figure 4.7. Again, we see that Channels 7, 9, and 10 have a higher sentiment than the others. The sentiment analysis no longer has a positive skew in 2020 and 2021 due to COVID-19-related terms in the previous analysis. We see a roughly yearly cycle on most channels; the broadcasting tends to be more positive during summer and more negative during winter. This corresponds with a positivity during the festive season due to terms such as 'christmas'. Also note the spike in the sentiment of SBS in May of 2022. We cannot find a single reason for this positivity; the terms used during this time are generic and do not align with any major events. The terms occurring more often in May 2022, and the specific date of this peak, the 15th of May 2022, are given in Table 4.5. Terms contributing the most to the high sentiment during this time are also shown in this table.

Although they are roughly correlated, each channel has a reasonably different sentiment. By combining all channels together, we can begin to form an understanding of the overall sentiment of the population with the effect of the channel minimised. Figure 4.8 shows the general media sentiment aggregated over all channels.

We observe correlation with events by performing a word shift analysis. Terms with the greatest entropy difference in specific months are shown in Table 4.5, alongside words contributing the most to extreme sentiment in these months.

April 2020 coincides with one of the first few months of the COVID-19 pandemic. Terms such as 'coronavirus', and 'cases' contribute to the very negative sentiment here. The significant dip in positivity at the beginning of 2022 corresponds with the start of the Ukraine/Russia war. Here, we see many words associated with this event such as 'ukraine', 'russian', and 'kyiv' contributing to the negative sentiment.

Text from January 2018 contains many tennis-related terms. These, however, do not appear in the terms contributing the most to a high sentiment due to their relatively neutral sentiment. Instead, the increase in sentiment is caused by a decrease in negative terms such as 'coronavirus' and 'government'. This was a period of relative calm, with no major negative news stories to bring the sentiment down. October 2021 is also a very positive month, although no major events seem to occur. Again, the primary reason for its relatively more positive sentiment is a decrease in negative terms. We find that relatively positive periods of time are often not those with positive news stories, but rather those without any significantly negative events.

Using this method, we can understand the impact of some major events on the sentiment of news captions. We are also able to see periods that lack negative events and are therefore given a relatively positive sentiment.

SBS May 2022	got, uh, gonna, you, carapaz, hindley, ok, laughs, oh, yeah
Used more	correct, do, beautiful, nice, applause, laughs, get, thank, laughter, great, like, love, know, good
Used less	virus, police, death, died, opposition, government
SBS 15th May 2022	meena, paddington, leemreize, oh, piers, allyson, 333rd, krystal, bolognese, yeah
Used more	amazing, family, profit, school, think, big, right, correct, get, laughs, love, nice, great, thank, like, know, good
Used less	death, died, government
January 2018	dimitrov, wolff, seaplane, oprah, mclachlan, wozniacki, federer, cilic, expedia, bannon
Used more	you, hot, day, big, book, your, it, cool, great
Used less	cases, minister, covid, prime, government, coronavirus, parliament, election, ukraine, virus, morrison
April 2020	curve, cases, easter, restrictions, 19, virgin, virus, ruby, distancing, coronavirus
Used more	coronavirus, cases, virus, toll, covid, crisis death, people, measures, distancing, restrictions, health, outbreak, government, pandemic, 19, workers, hospital
Used less	magenta
October 2021	icac, 2050, berejiklian, perrottet, glasgow, cleo, fully, dose, vaccination, vaccinated
Used more	get, new, and, we, to, so, for, you, your, be, first
Used less	police, trump, donald, president, attack, court, election, turnbull, security
March 2022	shane, flood, warne, lismore, mariupol, russia, kyiv, ukrainian, russian, ukraine
Used more	ukraine, russian, russia, war, ukrainian, kyiv, putin, russias, mariupol, russians, nato, flood, civilians, invasion, floods, ukrainians, lismore, forces
Used less	the

Table 4.5: The words which contribute the most to a difference in sentiment of general news media text during specific months. Positive terms are shown in orange, while negative terms are shown in blue.

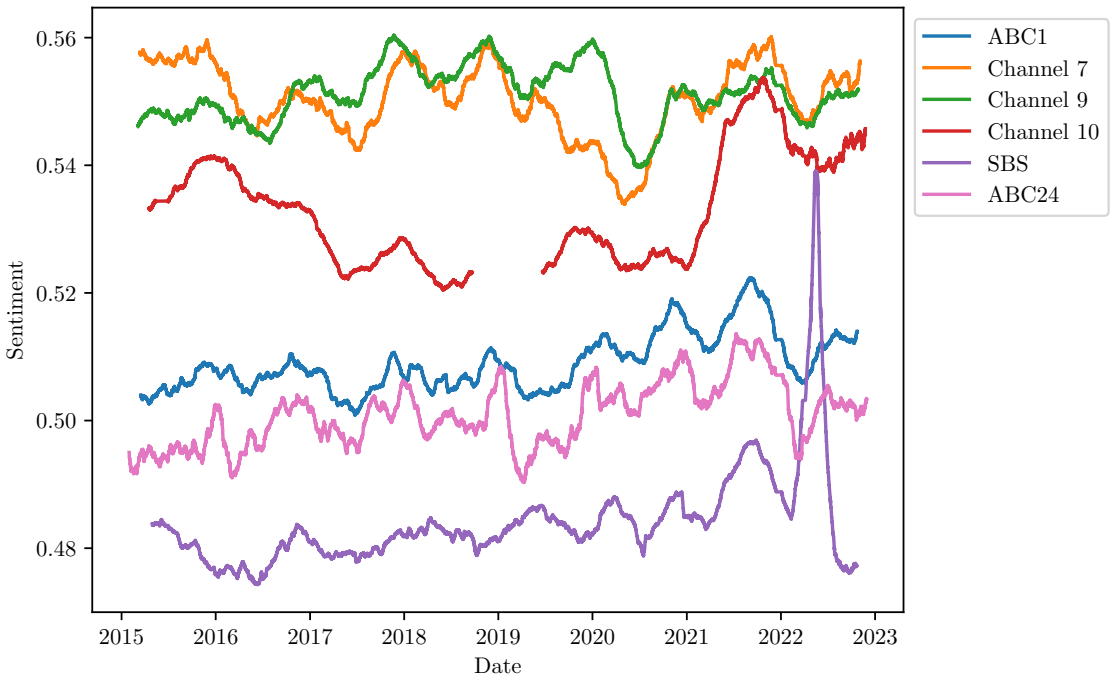

Figure 4.7: The sentiment of news text calculated using the Mittens lexicon. There is generally a spike around the end of each year corresponding with the festive season. The cause of the spike in the sentiment of SBS at the beginning of 2022 is unclear.

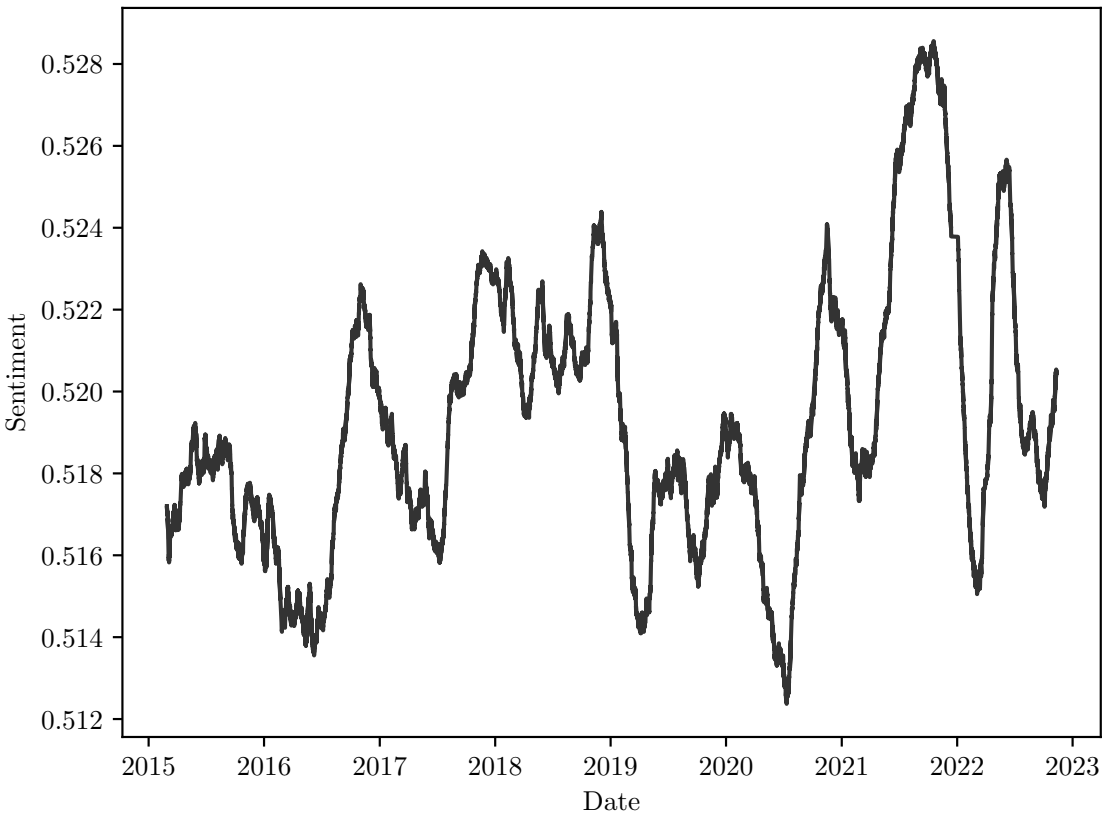

Figure 4.8: The sentiment of news text calculated using the Mittens lexicon aggregated over all channels. Specific peaks and troughs are explored in the text.

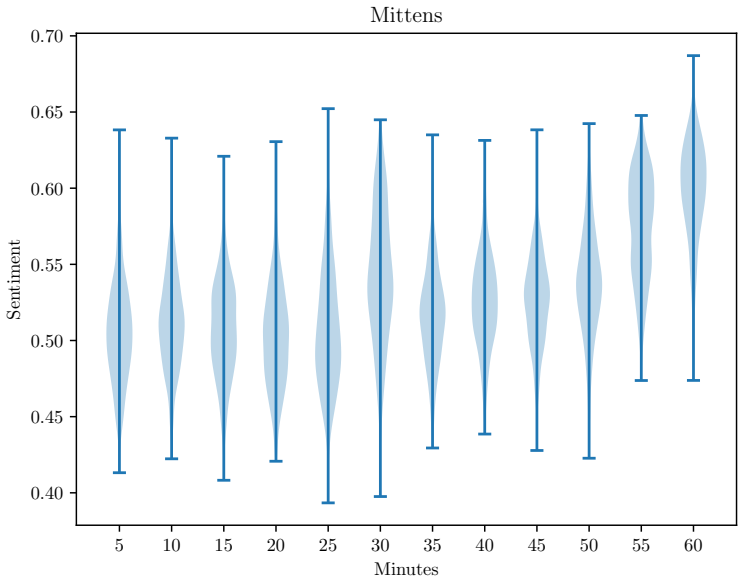

Figure 4.9: The sentiment of each 5-minute interval of hour-long news programs calculated with the Mittens lexicon. This indicates that the final two intervals have a higher sentiment than others.

4.6 If it bleeds, it leads

This section investigates the phrase 'if it bleeds, it leads'. This phrase describes the idea that news programs often begin with stories with negative sentiment [41]. These stories have been shown to invoke a higher level of emotion from audiences and thus capture their attention [40, 45]. We look at this phrase to obtain further evidence that the Mittens lexicon performs better at analysing news text than the original NRC lexicon.

We compare each the 5-minute intervals of 1-hour news programs with on ABC1 in Figures 4.9 and 4.10. This demonstrates that the median sentiment of the first five minutes is slightly lower than others, while the final five minutes has a much higher sentiment. Mean values of the first and final five minutes of every news program in 2022 are given in Tables 4.6 and 4.7. To test the significance of these results, we calculate an empirical p-value. This is done by randomly selecting 10,000 5-minute intervals of text for each channel. We calculate the sentiment of each interval and use the method described in Section 1.8 to compare with the means that we have found. We find single-sided p-values, the first of which represents the probability of seeing the value found for the first 5-minutes or *lower* if the samples were randomly drawn. Similarly, the second p-value is the probability of seeing the value found for the final 5-minutes or *higher* if the samples were randomly drawn.

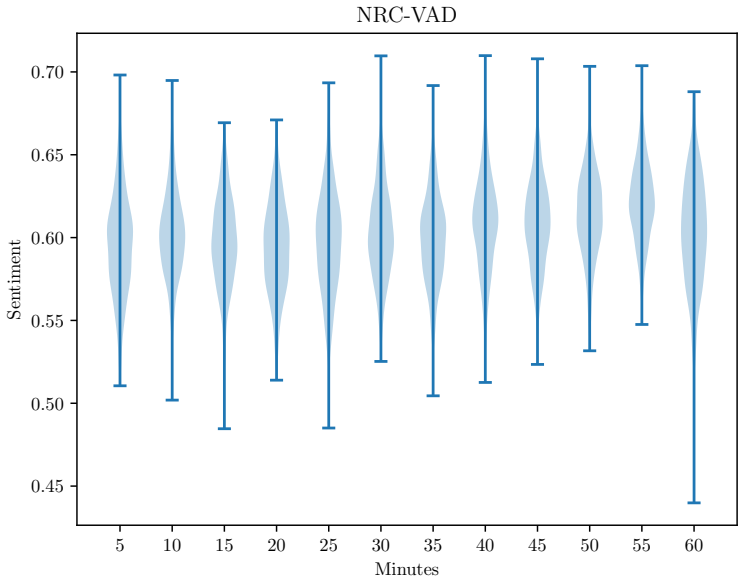

Figure 4.10: The sentiment of each 5-minute interval of hour-long news programs calculated with using NRC-VAD valence. There is no clear increase in sentiment throughout the programs.

All values found using the Mittens lexicon are significant, while some values found using NRC are not. Some values found using NRC instead show the opposite result. As Australian television media has not been rigorously analysed in this way, the 'if it bleeds, it leads' hypothesis may not be true for all channels. It has, however, been shown to be the case on a wide variety of other broadcasts internationally and one could safely assume that it is also the case for these programs. The Mittens lexicon has been able to establish results known internationally on this smaller data set. As such, we believe it to be well-suited to analyse text from the news genre.

4.7 Political sentiment analysis

In Chapter 3, we took a preliminary look at potential political biases within the text of each channel. Our previous analysis was limited to the coverage bias of each political party. We now begin to work the content of the attention into our bias analysis with a sentiment analysis of political text. Using a sentiment analysis, we can determine the polarity of the text and calculate statement bias.

Mittens	First 5-minutes	p-value	Final 5-minutes	p-value
ABC1	0.5158	0.0001	0.5955	0.0001
Channel 7	0.5364	0.0001	0.5739	0.0001
Channel 9	0.5533	0.0001	0.6068	0.0001
Channel 10	0.5642	0.0001	0.6015	0.0001
SBS	0.5370	0.0001	0.5761	0.0001
ABC24	0.5147	0.0001	0.5810	0.0001

Table 4.6: The mean sentiment of the first and last 5-minute intervals of news programs calculated using the Mittens lexicon. These values indicate that the first 5-minute interval has a lower sentiment than other intervals on average, while the final 5-minute interval has a higher sentiment than other intervals on average. All values are statistically significant at the 5% level of significance.

NRC	First 5-minutes	p-value	Final 5-minutes	p-value
ABC1	0.6035	0.0001	0.6183	0.0001
Channel 7	0.5154	0.0001	0.5315	0.9999
Channel 9	0.6104	0.0001	0.6264	0.0001
Channel 10	0.6131	0.9999	0.6262	0.0001
SBS	0.6045	0.8271	0.6227	0.0001
ABC24	0.5926	0.0001	0.6201	0.0001

Table 4.7: The mean sentiment of the first and last 5-minute intervals of news programs calculated using the NRC lexicon. Most values indicate that the first 5-minute interval has a lower sentiment than other intervals on average, while the final 5-minute interval has a higher sentiment than other intervals on average. Most values are statistically significant at the 5% level of significance.

4.7.1 Selecting text

We must create a corpus of text corresponding with each political party in order to analyse its sentiment. We define documents containing a topic as documents which have higher than 0.5 probability of the topic given that document. Although a high $p(y|\bar{x})$ term indicates text that contains a political topic, each document also contains large quantities of text that is irrelevant to the topics due to different news items in a single news bulletin. This means that the majority of the text used in the sentiment analysis would not be related to the political topics that we are investigating, skewing the calculated sentiment. We therefore choose to reduce the size of the documents out of necessity for this analysis. Although this increases computation time, it is crucial for us to accurately select political text without large quantities of noise. We choose to use 1-minute documents since these contain enough words to reasonably perform a sentiment analysis without containing too much irrelevant text.

We can quantitatively determine that these documents contain less irrelevant text by calculating the number of topics contained in each document. Again, a document 'containing' a topic is defined as a document with a greater than 0.5 probability of the topic given that document. We count the number of topics in each document and take the mean. On average, 5-minute documents contain 13.0068 topics, while 1-minute documents contain just 10.5711. The decrease in document size has led to a decrease in overlapping topics, therefore reducing the amount of irrelevant text in the sentiment calculation.

Similarly, we compare the 1-minute and 5-minute documents that are assigned to the Liberal and Labor topics trained on the two data sets with a word shift analysis. The shifts in entropy are shown in Table 4.8. The terms that are used proportionately more in the 1-minute documents are words relating to both parties. Words such as 'liberal', 'labor', and 'party' feature high on this list. Terms used proprtionately more in 5-minute documents include 'translation' and 'showers'. It is clear that the 1-minute documents contain a greater proportion of political text with fewer miscellaneous terms.

In order to select text corresponding to particular political parties, we then train a topic model on the entire corpus now split into 60-second intervals. We anchor the political terms listed in Section 3.8.1 to this topic model to ensure that we obtain both a Liberal and Labor topic. Training returns the probabilities of each topic given each document, $p(y|\bar{x})$. We repeat this 30 times to find an average probability for each topic in each document. Documents that correspond to a particular party are documents which have higher than 0.5 probability of containing the political party topic. The sentiment of these documents is then used to generate the time series plots in Figures 4.11 and 4.12.

Also note that rather than selecting text with a binary classification, we could instead find a weighted sentiment by multiplying the sentiment by the probabilities of topic y_j given the document. This is

Liberal 1-minute documents	scott, morrison, liberal, turnbull, malcolm, party, prime, minister, turnbulls, morrisons, abbott, tony, election, pm, mr, labor, shorten, leadership, liberals
Liberal 5-minute documents	translation
Labor 1-minute documents	labor, bill, anthony, shorten, albanese, party, election, leader, coalition, opposition, labors, government, greens, shortens, seats, liberal, vote, birtles, tax
Labor 5-minute documents	showers

Table 4.8: The words with the greatest difference in Tsallis entropy between 1-minute and 5-minute documents. Words used more in 1-minute documents clearly relate to their respective political party. This indicates that reducing the document size to 1-minute in length has reduced the number of irrelevant words and increased the proportion of words related to each party. This in turn has increased the quality of the political documents.

$$S_i = s_i p(y_j|\bar{x}_i),$$

where s_i is the sentiment of document i. This ensures that weight is given to documents with any non-zero probability of belonging to each topic. If a document has a low, but non-zero, probability of belonging to the topic, it will still receive some weighting. Plotting the weighted sentiment values alone will, however, skew the results by also taking into account the media attention. This can be useful when we would like to look at both sentiment and attention simultaneously, however in a general sentiment analysis we would like to focus on the sentiment alone, which may be obscured by the topic modelling. We instead use a binary classification here to ensure that we are analysing the sentiment alone, however we return to this idea later when comparing the two parties.

4.7.2 Analysis

We obtain political text from the entire corpus without restricting to text from the news genre. This ensures that we don't miss any which may belong to another genre such as 'social/political issues/economics (general)'. Since the majority of political text is from the news genre, it is appropriate to use the Mittens lexicon here. We have shown that political terms such as 'party' are given a more neutral sentiment in this lexicon and thus it should be more appropriate to use on this data than the NRC-VAD lexicon.

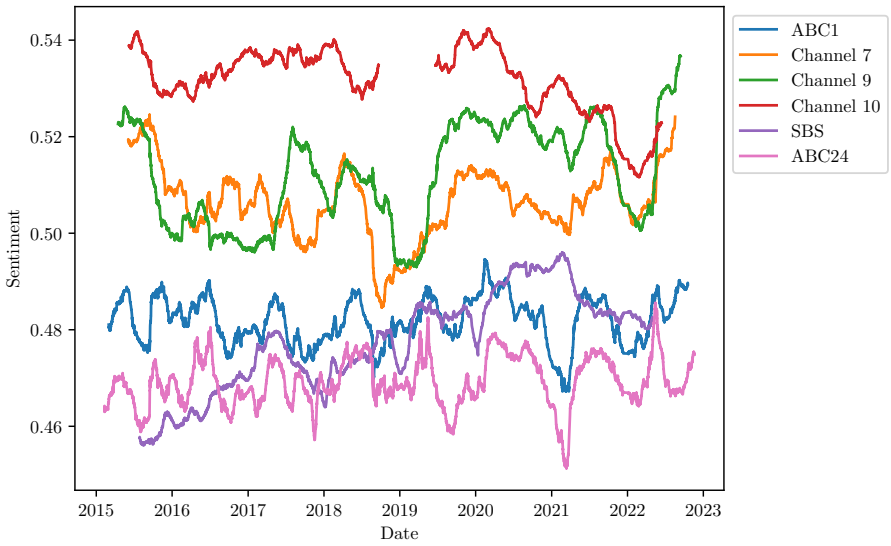

Figure 4.11: The sentiment of the Liberal text from each channel calculated using the Mittens lexicon.

The sentiment of Liberal and Labor text is shown in Figure 4.11 and Figure 4.12 respectively. We see a similar positive sentiment for Channels 7, 9, and 10 and more negative sentiment for the other channels.

We are also able to discern events surrounding each political party using a word shift analysis. Words contributing the most to extreme sentiment in selected months are given in Table 4.9. The low sentiment of the Liberal text, especially on ABC1 and ABC24, in February 2021 corresponds with the Brittany Higgins sexual misconduct scandal within the Liberal Party [9]. Low sentiment of Labor text in 2019 corresponds with protests in Hong Kong [65], as well as Brexit [22]. Finally, low sentiment of Labor text in March 2022 corresponds with the death of Kimberley Kitching and allegations of bullying [52].

There are some peaks that we cannot account for, however those peaks that we have identified indicate that the sentiment analysis is able to detect several major events. It is restricted by the simplicity that comes with every dictionary-based sentiment analysis. Words with multiple meanings, or multiple people with the same name, are given the same sentiment. We have also not factored in negation or other more complex grammatical structures which can alter the sentiment of a document, although these would only provide very minor improvements to the analysis. Despite its limiations, the Mittens lexicon does pick up the sentiment surrounding several major political events and some use as a simple but reasonably effective sentiment analysis tool for this data set.

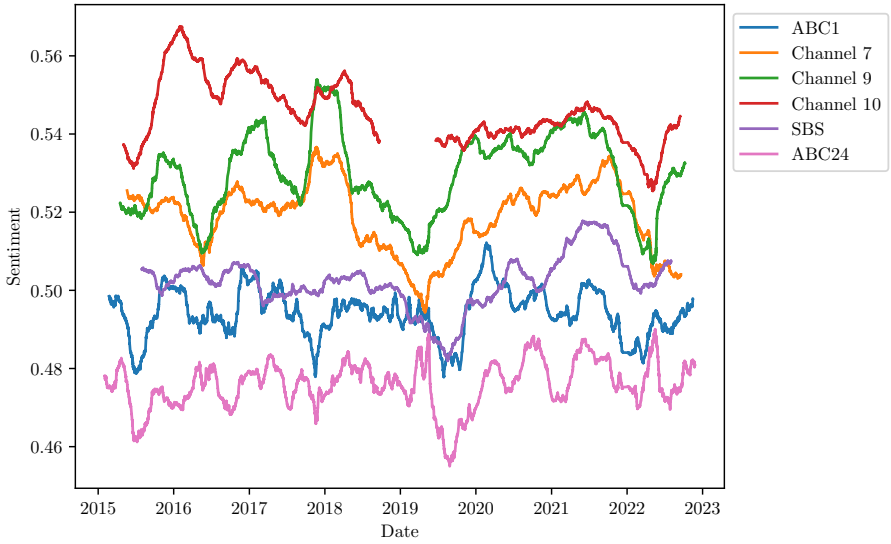

Figure 4.12: The sentiment of the Labor text from each channel calculated using the Mittens lexicon.

Liberal February 2021	raped vaccine, kelly, craig, reynolds, alleged, staffer, rape, brittany, higgins
Used more	minister, higgins, prime, morrison, staffer, parliament, rape, brittany, alleged, allegations
Used less	
Labor September 2019	lius, brexit, lam, kong, carrie, johnson, boris, hong, gladys, liu
Used more	hong, kong, parliament, gladys, mps, brexit, boris, protesters, inquiry
Used less	we
Labor March 2022	barry, warnes, tinker, kitchings, ukraine, malinauskas, flood, bullying, kimberley, kitching
Used more	kitching, bullying, albanese, senator, kimberley, labor, inquiry, claims, ukraine
Used less	you

Table 4.9: The words that contribute the most to a difference in political sentiment during specific months. Positive terms are shown in orange, while negative terms are shown in blue.

4.7.3 Comparison of party sentiments

Again, we are interested in the difference between the values for each party, rather than the values themselves. To plot the difference between the two parties, we must ensure that the dates align. We therefore calculate a sentiment score for each day and compare these. This is done by taking the average of the document scores corresponding to each topic for each day. The difference in daily means is plotted in Figure 4.13.

The majority of channels have a negative value, or a Labor sentiment bias, for the majority of the time. This is because Labor-related terms generally appear closer to the positive seed words and are therefore given a more positive sentiment in the lexicon. This may be due to a variety of reasons such as use outside of a political context, and this most likely doesn't indicate an intentional bias. This highlights the importance of forming comparisons between the channels rather than looking at each one individually. There may be an underlying reason for an apparent bias which becomes evident when all channels are consolidated.

Recall that we previously suggested an alternative method for displaying sentiment, by plotting the sentiment values weighted by the probabilities of topic y_j. That is,

$$S_i = s_i p(y_j|\bar{x}_i),$$

where s_i is the raw sentiment of document i. By simply taking the difference between this value for two different topics, this allows us to easily compare their sentiment without having to take a daily average. We call S_i the weighted sentiment of document i. Recall that this will skew the sentiment slightly by also taking into account the probability of each topic and so we should be careful when referring to this as 'sentiment'.

The difference between the weighted sentiment values is plotted in Figure 4.14. It is clear that ABC24 and ABC1 have the most extreme differences as they had the highest weighted sentiment values. In this case, the absolute difference has a greater potential for being large, as the individual sentiments themselves are large. In this metric, we are more concerned about the direction of the difference, especially in relation to other channels. The difference in weighted sentiment for Channel 10, for example, lies almost entirely below 0, even when the values for all other channels are positive.

Note that this plot looks similar to the topic probability plot in Figure 3.23. This indicates that the topic probabilities have had a much larger effect on these values than the sentiment. Normalising the sentiment values does not have a large effect on this outcome. The difference in sentiment alone may be modelled much better from the difference in daily sentiment which does not take into account the topic probabilities at all.

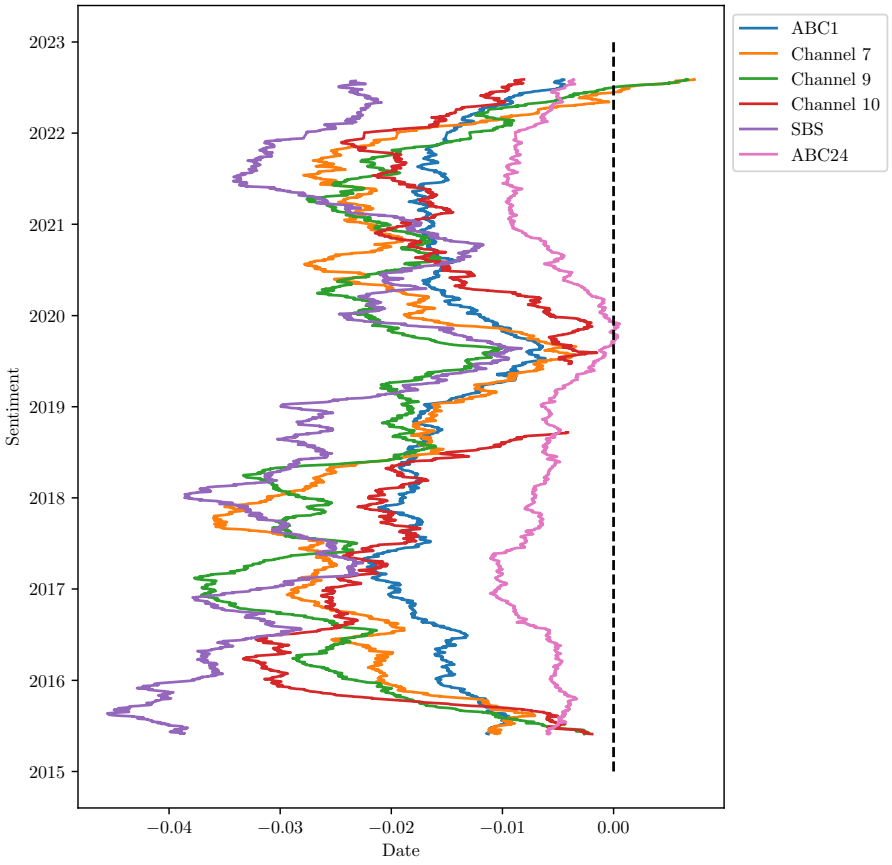

Figure 4.13: The difference between the daily average sentiment scores for Liberal and Labor text. A positive value indicates a Liberal sentiment bias, while a negative value indicates a Labor sentiment bias. The plot has been rotated for clarity, and to align with the left-wing and right-wing stances of the Labor and Liberal parties respectively. The majority of the daily average scores are negative, indicating a Labor sentiment bias.

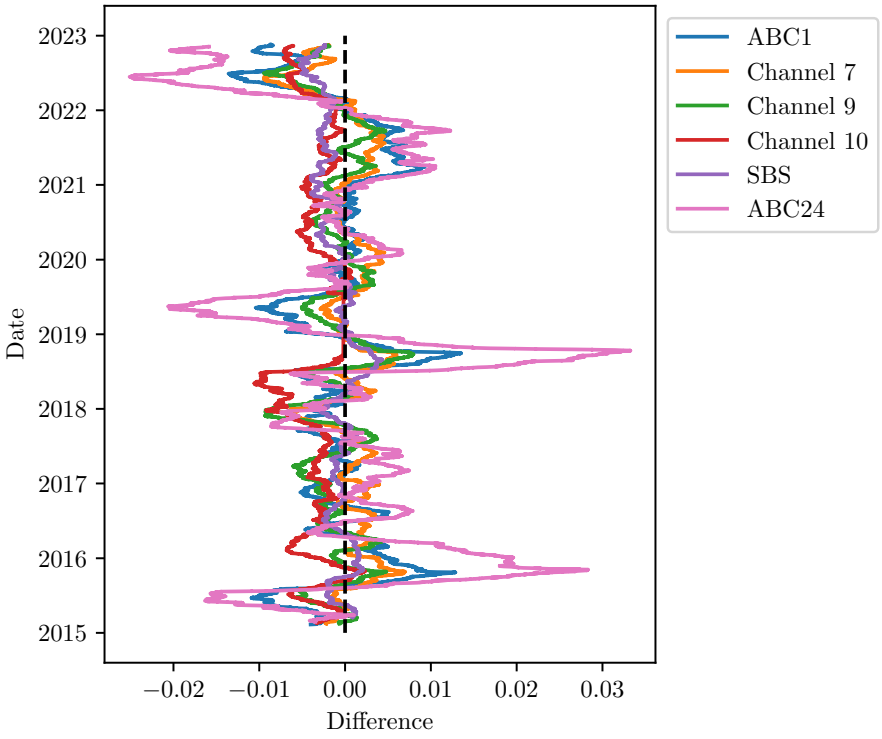

Figure 4.14: The difference in weighted sentiment of the Liberal and Labor topics. A positive value indicates a Liberal sentiment bias while a negative value indicates a Labor sentiment bias. ABC24 has the most extreme values using this metric due to the high topic probabilities. The peaks towards the Liberal party correspond with their change in leadership. Note the clear shift in 2022 towards a Labor bias after the federal election.

4.8 Statement bias

We previously investigated coverage bias and how this could be measured using topic modelling. We now focus on statement bias, a different form of bias that is concerned with the positivity of text. This section develops two simple methods to measure statement bias.

4.8.1 Mean sentiment score

The first method to find statement bias is to simply take the mean sentiment of documents surrounding each political party. The statement bias is then the difference between these, given by

$$\beta_S = \frac{1}{n_1}\sum_{i=1}^{n} s_{1i} - \frac{1}{n_2}\sum_{i=1}^{n} s_{2i},$$

where s_{1i} and s_{2i} are the sentiment values of the n_1 and n_2 Liberal and Labor documents respectively.

The mean sentiment of the Liberal and Labor documents, as well as the difference between them for each channel are given in Table 4.10. Empirical p-values were found using the method described in Section 1.8, by randomly selecting one sixth of the documents and computing the statement bias for these. This was repeated 10,000 times and p-value is the proportion of values that were more extreme than the values found. The p-values obtained indicate significance for all channels at the 5% significance level. Note that all of the difference values are negative which indicates that the Labor documents have a higher sentiment. This is likely the result of Labor-related terms generally having a slightly higher sentiment in our lexicon. We should therefore compare these values to the mean of the results from the empirical significance test, -0.0126. A value higher than this indicates a Liberal statement bias, while a lower value indicates a Labor statement bias.

4.8.2 Word embeddings

After fine-tuning the GloVe word embeddings, we are also able to make use of the semantic properties of word vectors to generate a second statement bias measure. We fine-tune the GloVe embeddings separately on each channel's text instead of on the entire corpus. Recall that this enables us to create a sentiment score for each term in the corpus using (1.16). By comparing the sentiment scores for political terms between channels, we then obtain a relative measure of sentiment bias for each channel.

Channel	Liberal	Labor	Difference	Difference from mean	p-value
ABC1	0.4821	0.4938	−0.0117	+0.0009	0.0032
Channel 7	0.5069	0.5171	−0.0102	+0.0024	0.0001
Channel 9	0.5119	0.5275	−0.0156	−0.0030	0.0001
Channel 10	0.5318	0.5454	−0.0136	−0.0010	0.0006
SBS	0.4764	0.5027	−0.0262	−0.0136	0.0001
ABC24	0.4697	0.4763	−0.0066	+0.0060	0.0001

Table 4.10: Comparison of the mean sentiment score for documents from Liberal and Labor topics. P-values are given from an empirical significance test with 10,000 values. All of the difference values are negative indicating a higher Labor sentiment. This could be the result of Labor-related terms generally having a slightly higher sentiment in our lexicon. We should therefore compare these values to the mean of the results from the empirical significance test, −0.0126.

liberal	labor
scott	anthony
morrison	albanese
malcolm	bill
turnbull	shorten

We use the same terms from the topic modelling chapter and calculate the mean sentiment score for each of these terms,

$$\beta_S = \frac{1}{n}\sum_{i=1}^{n} s_i. \tag{4.1}$$

This forms a measure for how positively each channel generally discusses the two political parties. The 'bias' values for each channel and each political party are given in Table 4.11. This defines a measure for the statement bias of each channel without considering the attention that each party receives.

Again, these results should not be taken at face value but rather compared between channels. The mean sentiment of each channel's lexicon should also be considered as this dictates a different 'neutral' value for each channel. The statement bias measures produce fairly different results. It is difficult to determine which is correct since we do not have any labelled testing data. Future work may focus on quantitatively analysing each method to compare performance on testing data.

Note that we may also use this same measure to find the sentiment of other topics outside of politics as in Section 3.8. For example, taking the mean sentiment of a list

Channel	Lexicon mean	Liberal	Labor	Difference
ABC1	+0.0117	−0.4302	−0.3522	−0.0780
Channel 7	−0.0007	−0.3023	−0.2952	−0.0071
Channel 9	+0.0053	−0.3137	−0.3357	+0.0220
Channel 10	+0.1069	−0.2239	−0.1673	−0.0566
SBS	−0.0507	−0.3624	−0.2625	−0.0999
ABC24	+0.0343	−0.2189	−0.1997	−0.0192

Table 4.11: Comparison of the sentiment of the terms 'liberal' and 'labor' from topic models trained on data from each channel. Most values are negative. We notice that Channel 10 and ABC24 have the highest sentiment scores, while ABC1 has the lowest.

of words related to a certain celebrity can indicate how positively they are discussed on each channel.

4.9 Including sentiment in the bias measure

It is evident that the sentiment values produced from the Mittens embeddings provide a reasonable estimate for the positivity of a channel. We now include sentiment in our bias measure introduced in Section 3.8.1. Recall that previously we could only analyse the amount of attention that a topic received. By including sentiment, we can now determine whether this attention, and therefore the general bias, is positive or negative.

We include the sentiment of the text by using it to weight the attention values in (3.3) and (3.4), giving us two measures for the sentiment bias. The mean of differences is now given by

$$MOD = \frac{1}{N}\sum_{i=1}^{N} s_i \left(p_{1i} - p_{2i}\right), \tag{4.2}$$

where s_i represents the sentiment of each document. Note that the values $s_i(p_{1i} - p_{2i})$ are the 'weighted sentiment' values plotted in Figure 4.14. Similarly, the ratio of sums becomes

$$ROS = \frac{\sum_{i=1}^{N} s_i p_{1i}}{\sum_{i=1}^{N} s_i p_{2i}}. \tag{4.3}$$

For these measures of bias we do not need to select political documents using the method detailed in Section 4.7.1. The multiplication by the probabilities p_{1i} and p_{2i}

Channel	MOD	p-value
ABC1	−0.0008	0.0559
Channel 7	+0.0005	0.0010
Channel 9	−0.0011	0.0020
Channel 10	−0.0035	0.0010
SBS	−0.0010	0.0609
ABC24	+0.0001	0.0010

Table 4.12: Comparison of the Liberal and Labor 'bias'. A MOD greater than −0.0009 indicates a Liberal 'bias', while a negative score indicates a Labor 'bias'. Most empirical p-values indicate significance.

mean that the contribution of sentiment when there is a low probability is minimised. Since most of these values are close to 0, their contribution is negligible.

For the MOD, a positive score indicates a Liberal 'bias', and a negative score indicates a Labor 'bias'. The ROS becomes difficult to work with in this case. We have shown previously that the sentiment lexicon contains many negative terms, and most political terms are given a negative sentiment. This means that the sums become negative in all cases we have tested. This reverses the score so that a value less than 1 indicates a Liberal 'bias', while a value greater than 1 now indicates a Labor 'bias'. The inconsistency of the scores given different signs makes it difficult to compare values should we come across a positive sentiment. We therefore disregard the ROS value as a measure for 'bias'. Note, however, that it can still provide meaningful results for coverage bias, since the probabilities are guaranteed to be positive.

The results for the MOD are shown in Table 4.12. We have chosen the Liberal topic as topic 1, and the Labor topic as topic 2. Empirical p-values calculated from 10,000 samples indicate significance at a significance level of 5% for all values except those for ABC1 and SBS.

This automatic media bias measure concludes that Channel 7, and ABC24 have Liberal biases, while Channels 9 and 10 have Labor statement biases. The MODs for ABC1 and SBS are not significant and we therefore cannot conclude that they are 'biased' either way.

Recall that the media bias that we are looking at is not necessarily intentional. These biases are likely the result of a wide range of factors such as the aim and content of each channel. For example, we may expect SBS to have a higher bias towards the party currently in power because it is broadcasting to a multicultural audience and is less concerned with domestic politics.

The MOD is the mean of the weighted sentiment difference that we have already plotted in Figure 4.14. Since it was so dependent on the topic probabilities, the overall bias is a much more fitting name for these values. This allows us to see not only the bias

of a channel but how this bias may change over time. The Liberal bias peaks twice, after their two changes in party leadership in 2015 and 2018. This is as we expect; a sense of optimism could surround media coverage of the Liberal Party after a change in leadership, with journalists predicting that a new leader will improve the party. The bias of all channels also shifts towards the Labor party after their win in the 2022 federal election, as we might expect.

Note that Figure 4.14 is incredibly similar to Figure 3.23 in Section 3.7. Because the difference in sentiment values is so small, the topic probabilities overpower them in this metric. This makes it difficult to discern statement bias within the plot and it also does not have a high weighting in the final bias measure. Further research may focus on giving the sentiment a more appropriate weighting in this measure.

Finally, we remind the reader to compare bias between channels, rather than as a single figure alone. Reporting on political parties' positive or negative actions is not biased in itself; it is when reporting is excessively different to other channels that a bias may be evident. Each channel should therefore be given an *allowable bias*, the amount by which they are allowed to deviate away from a value of zero according to current events. We define the overall allowable bias as -0.0009, the mean of the 10,000 empirical tests. The allowable bias at a given time should also be the mean of all channels' bias at that point. This is plotted in Figure 4.15. For example, near the end of 2018, it was acceptable to have a bias towards the Liberal Party due to the change in party leadership; all channels were biased towards the Liberal Party after this event. Significant deviation from this, especially towards the other party, may indicate bias at a given time.

4.10 Comparison with polling and electoral data

Public opinion analysis has been performed extensively on data from social media with reasonable success [5, 25]. This, however, has not been extended to television captions. We compare the allowable bias from news captions with public opinion polls to establish whether there is a correlation between the allowable bias of news media captions, and public opinion, at a given time. Although polling data is not always an accurate reflection of public opinion [27], these polls are performed at least once per fortnight and perform reasonably well at gauging public sentiment between elections.

We obtain two-party preferred voting data from several opinion polls including YouGov, Roy Morgan, and Essential [3, 7, 6, 39, 49, 50, 51, 66]. This data was collated on Wikipedia [60, 61, 62]. The difference between the two-party preferred polling proportion is plotted in Figure 4.16, along with the allowable bias shown in blue. The Pearson correlation of the two sets of values is 0.2891, indicating a weak correlation. We do see that the allowable bias provides a good estimate for some swings and spikes that are important in the polling data. A peak in late 2015, and a Labor swing in 2022 align

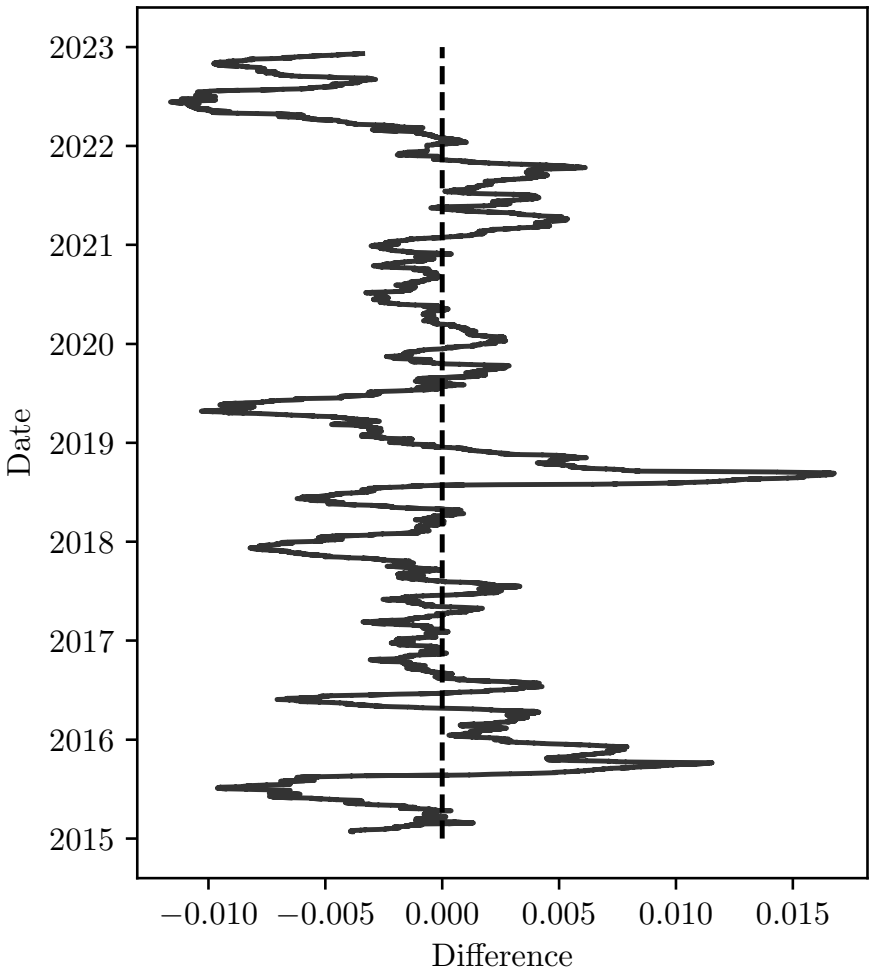

Figure 4.15: The allowable bias for each channel at a given time. This changes with major events such as changes in party leadership and election wins.

	2016	2019	2022
Allowable bias	0.0259	−0.0877	−0.1050
Actual result	Liberal	Liberal	Labor

Table 4.13: Comparison of federal election results with allowable bias on a given day. The allowable bias measure correctly predicts the 2016 and 2022 results. 2019 was a notoriously difficult election to predict and most polling sites also predicted this incorrectly.

with the allowable bias. This indicates that the bias measure roughly follows voting preference towards each party; we are somewhat able to understand the sentiment of the general public through an analysis of television captions. Some discrepancies, such as the Liberal spike in late 2018 of the allowable bias, are not reflected in the polling results. This is the result of change in party leadership and does not align well with the polling data at all.

We now compare the results with the most recent three federal elections in 2016, 2019, and 2022 to determine whether the allowable bias may be able to predict these results. Since these are verified results taken from the entire population, they form a more reliable indicator of public opinion at a given time. The mean allowable bias on each election day is given in Table 4.13. This has correctly predicted the election results in 2016 and 2022. The federal election in 2019 was notoriously difficult to predict, and most polling sites were incorrect [27]. Our method has also predicted this result incorrectly.

We conclude that the allowable bias models polling results to an extent, especially the general trends that polling data follows. We observe that the allowable bias was able to successfully predict the outcome of a federal election two-thirds of the time, however we would need significantly more data to determine whether this can reliably make predictions.

4.11 Conclusion

This chapter has provided an in-depth overview of the sentiment of each channel. We focussed on news text to understand the sentiment of real-world events without noise from irrelevant programs. A preliminary sentiment analysis revealed several problems with using a standard lexicon on this text. These problems were rectified by training our own lexicon on the corpus. This allowed us to capture the sentiment of a wider range of terms including recently coined terms like ‘covid’, and proper nouns such as ‘albanese’. In addition, the training neutralised common words such as ‘party’ and ‘know’ that skewed the sentiment. Applying the new sentiment lexicon to news data allowed us to spot periods where major events occurred and understand the sentiment

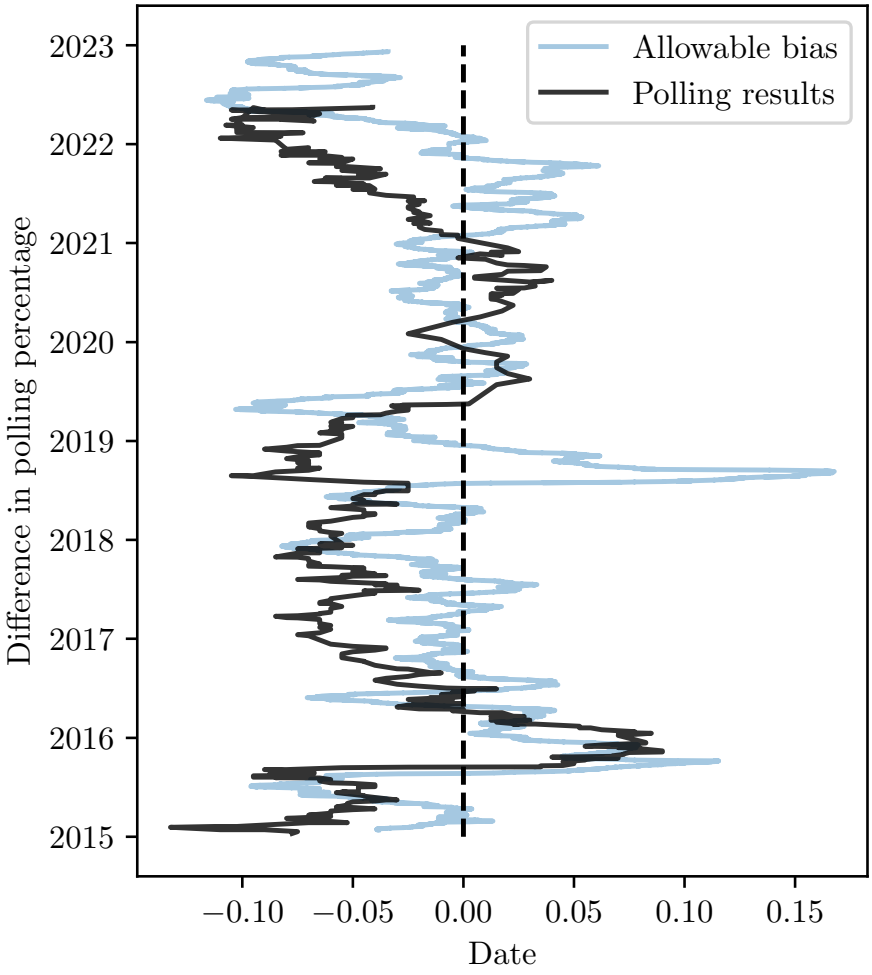

Figure 4.16: Public opinion polling data plotted over time in black. A positive value indicates a higher proportion of respondents stating that they will vote Liberal. The allowable bias divided by ten is shown in blue. The Liberal swing corresponds with the change in party leadership and aligns with the bias measure. The swing towards Labor in 2022 also aligns with our findings.

of the nation at a given time.

We aimed to use the sentiment analysis to develop a comprehensive bias measure. We therefore performed a sentiment analysis of political text comparing Liberal and Labor sentiment to obtain a basic statement bias score. Finally, we combined both the coverage and the sentiment of text relating to political topics and used this measure to calculate biases with significant empirical p-values. Results were compared with polling data which showed some correlation between the two, although it may not be appropriate to use as a predictor for federal elections.

Conclusion and future work

This thesis has built a new and unique comprehensive framework to quantify media bias by implementing both topic modelling and sentiment analysis on a data set of Australian television captions. Analysing this data set has increased our understanding of the relationships between television captions, real-life events, and the sentiment of the general population. A broad overview of the data set presented in Chapter 2, including an exploratory analysis of the characteristics of all television channels, is the first detailed analysis of the data set to our knowledge. This paves the way for further research into television media with the Tveeder data set.

Chapter 1 also provides a thorough background for the mathematics presented throughout this thesis. Within this chapter, we have presented a derivation of the CorEx topic modelling framework [46] in detail, improving accessibility of the technique. Further mathematical methods are discussed here. Of note is a novel method for comparing two topic models using Pearson correlation. This technique can be extended to topic models trained on different text and can therefore be useful in text comparison. We also extend Gallagher's [15] concept of word shifts to a comparison of sentiment lexicons.

Chapter 3 focusses on topic modelling, developing techniques to improve topics and broaden the range of applications. We begin by discussing topic model parameters. We introduce various methods for choosing the number of topics, and then introduce the document length as a parameter. If correctly adjusted, we find that an appropriate document length can vastly improve the quality of topics. Further research may focus on developing a method for comparing the quality of topic models which is not affected by the document length. This would allow us to quantitatively compare the size of documents to determine an optimal document size using an optimisation algorithm. Another avenue for research could look at adjusting the document size for various corpora; Does adjusting the document size improve topic modelling in text such as books or tweets where documents generally have a predefined size and so the 'best' choice for a document seems evident? In a complete reversal to this, recall from Chapter 2 that part of the Tveeder data set is a speech tag to track the speaker. A comparison between the quality of topic models trained on 1-minute documents and text split by speaker would form an interesting area for research.

The chapter continues by exploring strategies for making inference on unseen data with

a pre-trained topic model. Our proposed techniques are extremely successful in either filling in gaps in data or predicting the probabilities for unseen documents that did not exist at the time of training.

We then collate our research thus far and implement CorEx topic models on real-life data. This enables us to develop a detailed understanding of the fundamental characteristics of each channel, and ensures a solid foundation before we begin to look at bias. As part of our analysis on the data, we implement a hierarchical topic model to showcase how this extension of CorEx may perform on real-life data. This provides a good way to easily break larger topics into smaller topics.

By implementing CorEx on real-life data, we can model the media attention received by each topic. We discuss parallels between the media attention and real-world events over time. We use the media attention as a measure for coverage bias, taking the average over time to determine a level of 'bias' for each channel. The topic model performs well at determining the media attention over time. Although we draw parallels between media attention and coverage bias that make us confident in the measue, further research may work on developing testing data to quantitatively assess the quality of the coverage bias measure. This approach could be further refined by also considering shorter documents, or weighting the measure by the proportion of each document containing political text.

Chapter 4 focusses on sentiment anlaysis, specifically how we can use this to measure statement bias. We spend much of this chapter developing an appropriate sentiment tool for this analysis after finding several issues with generic sentiment lexicons when used on text from news programs. A new sentiment lexicon trained specifically on the corpus that we are looking to analyse performs better but still has several drawbacks. A more complicated sentiment analysis tool could be used, such as one including AI, however this would significantly increase computation time and, despite recent advances [26], would reduce explainability of the model while likely not greatly increasing accuracy.

We finally combine sentiment analysis and topic modelling to form an overarching bias measure which incorporates both statement and coverage bias. This new measure theoretically allows us to see both the prevalence of a political party in the media and whether their coverage is positive or negative. This is a novel approach for detecting bias in one television channel relative to other stations. Unfortunately, topic modelling dominates this metric and it is difficult to see statement bias within. We also have no testing data to verify the results of this measure. This opens the door for future work investigating an optimal weighting for topic probabilities and sentiment scores to create a reliable bias measure that can be appropriately verified. Testing data may include 5-minute intervals that are manually annotated with a Liberal or Labor bias depending on their content.

Throughout the thesis we also reiterate that this method requires a benchmark for

neutrality and cannot be used as an explicit bias measure without context. A method to detect bias without the need for comparison would be much more desirable.

We finally compare the allowable bias, or collated bias, with polling results and election outcomes. The allowable bias measure performs reasonably well at modelling the polling data. It was also able to successfully predict the outcome of two-thirds of federal elections in the data set. Significantly more data would be required to determine whether these results indicate a good outcome, and whether television captions could be used as a predictor of elections and public opinion.

Bibliography

[1] Ghadah Alomani and Mohamed Kayid. Further Properties of Tsallis Entropy and Its Application. *Entropy*, 25(2):199, January 2023.

[2] Australian Communications and Media Authority. TV captioning. https://www.acma.gov.au/tv-captioning, September 2023.

[3] Australian Financial Review. Federal election. https://www.afr.com/politics/federal/election, December 2023.

[4] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010. European Language Resources Association (ELRA).

[5] Nicholas Beauchamp. Predicting and Interpolating State-Level Polls Using Twitter Textual Data. *American Journal of Political Science*, 61(2):490–503, April 2017.

[6] Nicholas Biddle. ANU Poll 50 (April 2022): Volunteering, aged care, policy priorities and experiences with COVID-19, 2022.

[7] Nicholas Biddle, Ben Edwards, Diane Herz, Toni Makkai, and Ian McAllister. ANU Poll 2020: Bushfires, The Environment, and Optimism For The Future, 2020.

[8] Margaret M. Bradley and Peter J. Lang. Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings. *NIMH Center for the Study of Emotion and Attention*, 1999.

[9] Kathleen Calderwood. Brittany Higgins to pursue complaint of rape in Parliament office with Australian Federal Police. https://www.abc.net.au/news/2021-02-15/brittany-higgins-parliament-house-rape-allegations/13157168, February 2021.

[10] Christopher Cochrane, Ludovic Rheault, Jean-François Godbout, Tanya Whyte, Michael W.-C. Wong, and Sophie Borwein. The Automatic Analysis of Emotion in Political Speech Based on Transcripts. *Political Communication*, 39(1):98–121, 2022.

[11] Dave D'Alessio and Mike Allen. Media Bias in Presidential Elections: A Meta-Analysis. *Journal of Communication*, 50(4):133–156, December 2000.

[12] Nicholas Dingwall and Christopher Potts. Mittens: An Extension of GloVe for Learning Domain-Specialized Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 212–217, New Orleans, Louisiana, 2018. Association for Computational Linguistics.

[13] Peter Sheridan Dodds, Eric M. Clark, Suma Desu, Morgan R. Frank, Andrew J. Reagan, Jake Ryland Williams, Lewis Mitchell, Kameron Decker Harris, Isabel M. Kloumann, James P. Bagrow, Karine Megerdoomian, Matthew T. McMahon, Brian F. Tivnan, and Christopher M. Danforth. Human language reveals a universal positivity bias. *Proceedings of the National Academy of Sciences*, 112(8):2389–2394, February 2015.

[14] Jakob-Moritz Eberl, Hajo G. Boomgaarden, and Markus Wagner. One Bias Fits All? Three Types of Media Bias and Their Effects on Party Preferences. *Communication Research*, 44(8):1125–1148, December 2017.

[15] Ryan J. Gallagher, Morgan R. Frank, Lewis Mitchell, Aaron J. Schwartz, Andrew J. Reagan, Christopher M. Danforth, and Peter Sheridan Dodds. Generalized Word Shift Graphs: A Method for Visualizing and Explaining Pairwise Comparisons Between Texts. *EPJ Data Science*, 10(1):4, December 2021.

[16] Ryan J. Gallagher, Kyle Reing, David Kale, and Greg Ver Steeg. Anchored Correlation Explanation: Topic Modeling with Minimal Domain Knowledge. *Transactions of the Association for Computational Linguistics*, 5:529–542, December 2017.

[17] Lorenzo Gatti, Marco Guerini, and Marco Turchi. SentiWords: Deriving a High Precision and High Coverage Lexicon for Sentiment Analysis. *IEEE Transactions on Affective Computing*, 7(4):409–421, October 2016.

[18] Bert F. Green, Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. The General Inquirer: A Computer Approach to Content Analysis. *American Educational Research Journal*, 4(4):397, November 1967.

[19] Felix Hamborg, Karsten Donnay, and Bela Gipp. Automated identification of media bias in news articles: An interdisciplinary literature review. *International Journal on Digital Libraries*, 20(4):391–415, December 2019.

[20] Te Sun Han. Nonnegative entropy measures of multivariate symmetric correlations. *Information and Control*, 36(2):133–156, February 1978.

[21] Jan Havrda and František Charvát. Quantification method of classification processes. Concept of structural a-entropy. *Kybernetika*, 3(1):30–35, 1967.

[22] Samantha Hawley. Boris Johnson's Brexit plans are a mess as MP crosses the floor. https://www.abc.net.au/news/2019-09-04/boris-johnsons-brexit-plans-are-a-mess/11476066, September 2019.

[23] C. Hutto and Eric Gilbert. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225, May 2014.

[24] Isabel M. Kloumann, Christopher M. Danforth, Kameron Decker Harris, Catherine A. Bliss, and Peter Sheridan Dodds. Positivity of the English Language. *PLoS ONE*, 7(1):e29484, January 2012.

[25] Jin-ah Kwak and Sung Kyum Cho. Analyzing Public Opinion with Social Media Data during Election Periods: A Selective Literature Review. *Asian Journal for Public Opinion Research*, 5(4):285–301, August 2018.

[26] Bernadetta Maleszka. A Survey of Explainable Artificial Intelligence Approaches for Sentiment Analysis. In Ngoc Thanh Nguyen, Siridech Boonsang, Hamido Fujita, Bogumiła Hnatkowska, Tzung-Pei Hong, Kitsuchart Pasupa, and Ali Selamat, editors, *Intelligent Information and Database Systems*, volume 13996, pages 52–62. Springer Nature Singapore, Singapore, 2023.

[27] Luke Mansillo and Simon Jackman. National polling and other disasters. In Anika Gauja, Marian Sawer, and Marian Simms, editors, *Morrison's Miracle: The 2019 Australian Federal Election*. ANU Press, 1st edition, July 2020.

[28] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113, December 2014.

[29] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, September 2013.

[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality, October 2013.

[31] Saif Mohammad. #Emotional Tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, Montreal, Canada, June 2012. Association for Computational Linguistics.

[32] Saif Mohammad. Obtaining Reliable Human Ratings of Valence, Arousal, and Dominance for 20,000 English Words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–184, Melbourne, Australia, 2018. Association for Computational Linguistics.

[33] Saif M. Mohammad and Peter D. Turney. Crowdsourcing a Word-Emotion Association Lexicon, August 2013.

[34] B.V. North, D. Curtis, and P.C. Sham. A Note on the Calculation of Empirical P Values from Monte Carlo Procedures. *The American Journal of Human Genetics*, 71(2):439–441, August 2002.

[35] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics.

[36] Robert Plutchik. A psychoevolutionary theory of emotions. *Social Science Information*, 21(4-5):529–553, July 1982.

[37] Kyle Reing, David C. Kale, Greg Ver Steeg, and Aram Galstyan. Toward Interpretable Topic Discovery via Anchored Correlation Explanation. 2016.

[38] Peter Roget. The Project Gutenberg eBook of Roget's Thesaurus of English Words and Phrases, April 2004.

[39] Roy Morgan. Federal Voting – Two Party Preferred Voting Intention (%) (2016-2023). https://www.roymorgan.com/morgan-poll/two-party-preferred-voting-intention, November 2023.

[40] Paul Rozin and Edward B. Royzman. Negativity Bias, Negativity Dominance, and Contagion. *Personality and Social Psychology Review*, 5(4):296–320, November 2001.

[41] Susan Ruel. If it Bleeds, it Leads: Coverage of Violent Crime by the U.S. Television News Media. In Jean-Michel Lacroix, editor, *Violence et télévision: Autour de l'exemple canadien*. Presses Sorbonne Nouvelle, 1997.

[42] Diego Saez-Trumper, Carlos Castillo, and Mounia Lalmas. Social media news communities: Gatekeeping, coverage, and statement bias. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management - CIKM '13*, pages 1679–1684, San Francisco, California, USA, 2013. ACM Press.

[43] Antony Samuels and John Mcgonical. News Sentiment Analysis, July 2020.

[44] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, July 1948.

[45] Stuart Soroka, Patrick Fournier, and Lilach Nir. Cross-national evidence of a negativity bias in psychophysiological reactions to news. *Proceedings of the National Academy of Sciences*, 116(38):18888–18892, September 2019.

[46] Greg Ver Steeg and Aram Galstyan. Discovering Structure in High-Dimensional Data Through Correlation Explanation, October 2014.

[47] Greg Ver Steeg and Aram Galstyan. Maximally Informative Hierarchical Representations of High-Dimensional Data, January 2015.

[48] Soonh Taj, Baby Bakhtawer Shaikh, and Areej Fatemah Meghji. Sentiment Analysis of News Articles: A Lexicon based Approach. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–5, Sukkur, Pakistan, January 2019. IEEE.

[49] The Age. Resolve Political Monitor. https://www.theage.com.au/national/resolve-political-monitor-20210322-p57cvx.html, November 2023.

[50] The Australian. Newspoll. https://www.theaustralian.com.au/nation/newspoll, December 2023.

[51] The Guardian. Essential poll. https://www.theguardian.com/australia-news/essential-poll, December 2023.

[52] Laura Tingle. Kimberley Kitching's death has exposed allegations of bad behaviour in Labor ranks, stopping Albanese's momentum in its tracks, March 2022.

[53] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, April 2000.

[54] Franco Trimboli. Tveeder. www.tveeder.com, December 2023.

[55] Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346, October 2003.

[56] Peter D. Turney and Patrick Pantel. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188, February 2010.

[57] Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7):5731–5780, October 2022.

[58] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior Research Methods*, 45(4):1191–1207, December 2013.

[59] Satosi Watanabe. Information Theoretical Analysis of Multivariate Correlation. *IBM Journal of Research and Development*, 4(1):66–82, January 1960.

[60] Wikipedia. Opinion polling for the 2016 Australian federal election. https://en.wikipedia.org/wiki/Opinion_polling_for_the_2016_Australian_federal_election, September 2023.

[61] Wikipedia. Opinion polling for the 2019 Australian federal election. https://en.wikipedia.org/wiki/Opinion_polling_for_the_2019_Australian_federal_election, September 2023.

[62] Wikipedia. Opinion polling for the 2022 Australian federal election. https://en.wikipedia.org/wiki/Opinion_polling_for_the_2022_Australian_federal_election, September 2023.

[63] Wikipedia. Wikipedia:Neutral point of view. https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view, December 2023.

[64] Alden Williams. Unbiased Study of Television News Bias. *Journal of Communication*, 25(4):190–199, December 1975.

[65] Karson Yiu. Hong Kong's embattled leader withdraws bill that sparked months of unrest; protesters say 'too little, too late', September 2019.

[66] YouGov. Politics and current affairs. https://au.yougov.com/topics/politics/articles-reports, November 2023.

[67] Lori Young and Stuart Soroka. Affective News: The Automated Coding of Sentiment in Political Texts. *Political Communication*, 29(2):205–231, April 2012.

[68] Xinran Yu, Chao Zhong, Dandan Li, and Wei Xu. Sentiment analysis for news and social media in COVID-19. In *Proceedings of the 6th ACM SIGSPATIAL International Workshop on Emergency Management Using GIS*, pages 1–4, Seattle Washington, November 2020. ACM.