A methodology for predictive topic modelling; or, any excuse to watch *Love Actually*

Vanessa Glenny

May 2018

Thesis submitted for the degree of
Master of Philosophy
in
Applied Mathematics
at The University of Adelaide
Faculty of Engineering, Computer and Mathematical Sciences
School of Mathematical Sciences

THE UNIVERSITY
of ADELAIDE

Contents

List of Tables

List of Figures

Signed Statement

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.

Signed: .... Date: ....

Acknowledgements

Firstly, I would like to thank my supervisors, Professor Nigel Bean, Dr Lewis Mitchell, and Dr Jono Tuke for the incredible amount of support and advice they have given me over the past two years, and for making my first research experience a positive one.

I would also like to acknowledge the support of the ARC Centre for Excellence for Mathematical and Statistical Frontiers, who have provided several opportunities for development over the course of my candidature.

Finally, thanks must go to my parents, for their constant understanding and encouragement, and my friends, in particular Caitlin and Angus, for their mathematical wisdom and support.

Abstract

Topic modelling is an area of natural language processing (NLP) in which a corpus of text documents is summarised by an underlying structure of 'topics', or themes. Due to the incredibly complex nature of human language, we often require ways to meaningfully summarise the information contained in a piece of text. Topic models provide a method in which we are able to keep substantial semantic information, but still work with a small number of variables. Topic modelling has mainly been applied to machine learning problems, with little emphasis on prediction from text. This thesis provides a statistical framework for prediction from, or about, text, using topic models as a data reduction method and the topics themselves as predictors.

The results of this thesis show that while using individual words as predictors in a regression model remains the most accurate method, it is far too computationally expensive to apply to large corpora. However, the topic regression models proposed here perform comparably, and at a much lower computational cost. We also show that incorporating more information, such as the structure of language, into topic model inference improves the predictive capability of the topics. This thesis therefore proposes a computationally viable, well-performing method for prediction from text.

From here, we may consider adapting additional topic models to a regression framework, depending on the problem at hand and its requirements. These methods, while tested in this thesis on relatively small corpora, would also be applicable to big data problems.

Chapter 1

Introduction

1.1 Motivation

Natural language (or human language) has the capability to convey large and complex amounts of information in relatively small communications. Most of this information is readily interpretable by human listeners or readers, but machines are more restricted in their ability to infer the nuances. The field of natural language processing (NLP) has arisen over the past 70 years in order to rectify this discrepancy, and in some areas to surpass human capability. While problems that involve the more complex, semantic and cultural information conveyed through language are less likely to be solved by machines than humans, machines have the advantage of being able to process much more information and at a much faster rate. That is, NLP can be crucial to extracting particular types of information from large amounts of speech or text data, something that is difficult or impossible for most humans.

The more large-scale problems of NLP have only been attempted fairly recently, given the computational power necessary to implement their solutions. One field of study for these problems is *topic modelling*, a type of NLP. In topic modelling, as the name would suggest, we assume that any piece of natural language (generally called a 'document'), such as a piece of text, can be summarised by an underlying structure of ‘topics’. These topics can then be used to infer things about or from the document in question. Topic models provide a way of condensing or reducing the data into a more manageable but still meaningful form, and thus can be used to analyse large amounts of text data.

For the most part [9], topic models have been applied to machine learning-type problems involving big data, such as web spam filtering [33] and database sorting [31]. Topic models have also been applied to non-language data, like genetic sequencing [40] where the genes function like words in the model. For the purpose of this thesis, however, we consider only natural language data.

While machine learning applications are how topic modelling started, the concept of topics and their ability to condense data can be applied to a much wider range of problems. Relatively little work so far has been done on predictive topic modelling [9], where topic models are used to convert unstructured text data into a usable form for the purpose of making predictions with statistical models. The purpose of this thesis is therefore to develop a methodology for predictive topic modelling, from a statistical perspective. That is, are we able to meaningfully reduce text data into topics, which can then be used as accurate predictors of some kind of response variable? For example (and as highlighted in Chapter 3), are we able to make predictions from the text of an advertisement about the content of said advertisement?

It can be argued that there are more straightforward ways of condensing text data than finding the ‘topics’ of a corpus of documents. For instance, we can look simply at whether certain words are present, a method that is compared to the predictive topic models developed later in the thesis. However, this has the disadvantage of potentially losing relevant information to the prediction, as it does not pick up on inter-word connections. Similarly, while examining pairs or triplets of words (*i.e.,* n-grams) does account for some of those connections, it still does not enable the model to identify synonyms (that is, distinct words with the same meaning). Topic modelling has been noted [39] for its approach to solving problems such as synonymy, and thus we investigate the application of topic models here over other methods. Also, these alternative methods do not have the capability to infer ‘long-range’ connections within a document; that is, they cannot group related ideas expressed in different parts of the text. One way to capture this long-range structure is through topic modelling, where topics span the document as a whole. The major advantage of topic modelling over these models, however, is the relatively small number of variables involved (as compared to, say, tens of thousands of individual words in a vocabulary).

There also exist other NLP methods that, like topic modelling, capture relationships between distinct words in a corpus. Specifically, techniques such as word embedding, where a vocabulary is mapped to a vector space, are a popular choice for many machine learning problems (see Google’s Word2vec [36] as a well-known example of this). While it would be prudent to consider such methods in comparison at a later point, we choose here to apply topic models due to the probabilistic generative models on which they are built.

Since the first topic model was developed in 1998, many models have been developed for many different purposes. Most of these models can be differentiated by the assumptions they make about the text data they are modelling. In order to save considerable computation, topic models must simplify how they consider the generation of language, compared to how a human would consider the generation of language. One example of this is the ‘bag of words’ assumption, which assumes that any piece of text has words generated independently of each other, thus rendering order unimportant. This thesis aims to not only develop a methodology for topic models in a predictive context, but also to see how some of these assumptions affect the accuracy of predictions.

1.2 Background

Topic modelling is a field of NLP in which documents in a corpus are summarised by ‘topics’, here defined as probability distributions over the vocabulary. Various topic models for different purposes exist, but this thesis focuses on three in particular.

Latent Dirichlet allocation (LDA) [13] was one of the first sophisticated topic models developed, and consequently has been used as a 'baseline' model from which many others have been created. It is noted for its simplicity, in particular regarding its assumptions of the corpus. LDA is a 'bag of words' model, meaning it disregards any document structure in model inference. Additionally, LDA, like most topic models, is an unsupervised process; that is, topics are inferred with no regards to any document labelling or response variable.

Building on LDA is supervised LDA (sLDA) [12], which allows for a response variable in the assumed generative process of the corpus. That is, when finding topics, given a corpus and some labelling of the documents in it, sLDA will take into account that labelling.

The hidden Markov topic model (HMTM) [4] is a second unsupervised model, which relaxes the 'bag of words' assumption made by the other two. While there are many different ways of incorporating document structure, or language structure, into a topic model, the HMTM does so by assuming each word's topic is dependent only on the one before it, as in a Markov chain [38]. Specifically, it is based on the structure of a hidden Markov model (HMM) [44], which assumes a sequence of observations generated by an underlying sequence of latent 'states', in this case topics, that form a Markov chain. This thesis will examine each of these models, their underlying assumptions, and their usefulness within a statistical prediction framework.

1.3 Outline of thesis

This thesis is separated into three main chapters. The first of these explains the background and key concepts necessary for the work in this thesis. The second introduces a methodology for topic modelling regression, and explores the effect of supervised learning on prediction. The third continues with the methodology, while investigating the effects of introducing document structure into the model.

In Chapter 2, we explore the background behind topic modelling regression to this point, as well as outlining the necessary mathematics for comprehension of this thesis. Specifically, we describe the history of topic models and the key features that differentiate them from each other. We also step through the three topic models that are implemented in Chapters 3 and 4 in detail (*i.e*, LDA, sLDA and the HMTM). A summary of logistic regression is also provided, as well as a discussion on model validation, in order to aid understanding of the following chapters.

Chapter 3 introduces the concept of topic modelling regression, by applying our methodology to a corpus of online advertisements for cats on the trading website, Gumtree[1]. The problem we present is whether we can use the text of the advertisements to predict the relinquished status of the cats in them, *i.e.*, whether the cat is being sold or given up by a previous owner. A preliminary analysis of the data shows that there is a human-interpretable discrepancy in the vocabulary of the advertisements pertaining to relinquished cats, versus non-relinquished cats. That is, we are able to identify key words that logically appear in either the relinquished or non-relinquished advertisements. Before applying topic modelling to the problem, we first use individual words as predictors in a logistic regression model. We use this model as a benchmark against which we measure our topic regression models. While a simpler concept than topic models, word count models can be incredibly computationally expensive and thus would not typically be used for problems with large corpora.

Latent Dirichlet allocation (LDA) is used in this thesis as the 'baseline' topic model due to the simplicity of its generative assumptions and its widespread use. Our first step in developing a methodology for topic modelling regression is to therefore use LDA as a preprocessing step on the data before feeding the topics as predictors into a logistic regression model.

[1] `www.gumtree.com.au`

Specifically, the proportions of the topics in each of the documents are used as the predictors, *i.e.*, the presence of a certain topic or topics in an advertisement may indicate a relinquished status. Using LDA, we are able to infer these topic proportions for each document in our corpus, as well as the topics themselves over the corpus, in order to build the logistic regression model for this problem.

However, when it comes to prediction, we are interested in applying this regression model to new documents (*i.e.*, documents not found in the original corpus used to build the model). Typically in topic modelling, when encountering new documents there is a tendency to simply refit the topic model (in this case, the LDA model) over the original corpus plus the new documents. This is not true prediction, and it also requires the computational expense of generating both a new LDA model and a new regression model. Instead, we develop a technique based on maximum likelihood estimation to find the topic proportions of any new documents, based on the LDA model over our existing corpus. This enables us to then use these estimated topic proportions as our predictors to feed into our existing regression model. We demonstrate that our method leads to a substantial computational improvement over the standard methods employed in the literature.

Using cross validation and, specifically, by generating receiver operating characteristic (ROC) curves [20], we are able to compare this topic regression model to our word count model found earlier. These results show the LDA regression model performs slightly worse on this corpus, although it is markedly more efficient computationally. We also perform a step-up process (outlined in Chapter 2) on the topics found by the LDA model to test if using some subset of them as predictors would be an improvement on using all topics. For this particular problem, this causes no noticeable improvement in predictive capability.

While LDA does have the computational advantage of simplicity, it makes strong assumptions about the generation of documents in our corpus. One such assumption is that the documents are generated without regard to any response variable, or labelling, such as what we are trying to predict. As such, the topics found in the LDA model are not necessarily the most discerning ones possible for the problem at hand. We can instead use supervised LDA (sLDA), a modification of LDA which assumes a response in its generative process for documents.

All topic models used in this thesis assume a fixed number of topics when performing inference. As such, we need a way of determining the 'best' number of topics for a given problem (that is, the number of topics that gives us the best performing regression model). In this case, the numbers of topics for both the LDA and sLDA regression models were found by comparing the cross validation prediction errors (CVPEs) [21] across a range of numbers of topics. That is, each model was tested to see how well it predicted a selection of documents in the corpus, given the remaining documents. The best LDA model found had 26 topics, while the sLDA model had two. Comparing these two regression models, the LDA model is found to perform slightly better than the sLDA. This is unexpected, given the unsupervised nature of LDA. However, this difference can be accounted for in the number of topics each one has, with an sLDA model of 26 topics outperforming that of its LDA counterpart. The two topic sLDA model was chosen as the 'best' model (over that of 26 topics) in order to combat overfitting, or fitting too close to the corpus and thus affecting prediction of new documents.

Chapter 4 continues the exploration of our methodology for topic modelling regression. Instead of investigating the effect of supervision, we focus here on the effect of the 'bag of words' assumption, which is adopted by many topic models including LDA. In the Gumtree problem, we examined online advertisements, where this assumption is reasonable: the advertisements tend to be short, with the sole purpose of conveying straightforward information through key words. In order to compare models incorporating document structure to those using the 'bag of words' assumption, we consider a problem where the documents are expected to contain some structure. For that purpose, we consider the dialogue of the 2003 film *Love Actually*[2] [18], in which there are ten interwoven, yet distinct, storylines. We attempt to see how each of the models fare in predicting the storyline of a scene from the movie, based on its dialogue.

As in Chapter 3, we first develop a regression model using individual words as predictors. Using a step-up process that penalises overfitting, the best model found has three words as predictors. We again use this model as the standard against which our topic regression models are judged. However, using ROC curves (and the area under the curve) as measures of prediction accuracy does not suit this problem as well as the Gumtree problem in the previous chapter, as we are now dealing with a non-binary categorical variable as our response (*i.e.*, storyline). Instead we use the Brier score [15] (a method outlined in Chapter 2 that measures predictive accuracy for categorical data) of our models as a comparison tool, with the lower score indicating better prediction accuracy. We do so by performing leave-one-out regression on each of the scenes (or documents) in our corpus.

In order to investigate the effects of document structure on the prediction capability of our topic regression models, we must measure it against a 'bag of words' model. For that reason, we use the same methodology as for the Gumtree problem to find the LDA regression model for this problem. Comparing the Brier score calculated from leave-one-out regression on the LDA model to that of our word count model, we see that the LDA model performs noticeably worse than its word count equivalent.

There are several ways of introducing document structure to topic modelling. One of the simplest ways is to assume a Markovian dependence structure across the document; that is, each topic assignment is dependent on the one before it, but conditionally independent of the rest. The nature of topic models lends itself neatly to the structure of a hidden Markov model (HMM), which incorporates this dependence. Topic assignments are analogous to the latent states of an HMM, with words being the 'observations' from

[2] `http://www.imdb.com/title/tt0314331/`

these states. We therefore choose to apply the hidden Markov topic model (HMTM) as a preprocessing step to a regression model for this problem.

In doing so, we require the estimation of topic proportions for previously unseen documents, given our existing HMTM. To solve this problem, we employ methods used in HMM inference: specifically, the Baum-Welch algorithm. While the Baum-Welch algorithm provides estimates for all HMM parameters, we assume here that the topics (analogous here to the emission probabilities of an HMM) are known, as we have already found our model. We are simply interested in finding the transition probabilities between topics. However, the Baum-Welch algorithm uses an iterative procedure to estimate these parameters, and so we are able to fix the topics and update only the transition probabilities on each iteration. Finding an estimate for these transition probabilities then gives us a way of estimating the topic proportions for each document.

We also investigate topic stability. With topic dependence introduced, it makes sense to assume that priority will be given to staying in the same topic over consecutive words. However, the current method for the HMTM does not show a particularly strong preference for this. We therefore derive a modified version of the HMTM to give preference to longer sequences of words in the same topic by adjusting the model parameter controlling transition probabilities. We call this model a *persistent HMTM*.

Overall, Chapter 4 describes how the HMTM regression model performs better than the LDA model, indicating that document structure is useful information to retain in our model. Similarly, the persistent HMTM outdoes the original HMTM in prediction accuracy, indicating that topic stability improves the topics produced by the model.

Finally, we conclude with a summary and outlook for future work in Chapter 5.

Chapter 2

Background

2.1 Introduction

2.1.1 Natural language processing

Natural language processing (NLP) refers to the field of study in which natural language, that is, language produced for human-to-human communication, is analysed and interpreted by machines. NLP covers techniques and models developed for a range of purposes, with the common aim of allowing computers to understand information contained in human language. This information could be as determinate as identifying parts of speech; or indeed it could be the more open semantic meaning of the text in question.

One of the first forays into NLP was the Georgetown-IBM experiment in 1954 [30], in which around 60 sentences were translated from Russian to English given a vocabulary of 250 words and 6 grammatical 'rules'. Thanks to the advancement of computing power, computational linguists are now able to focus on the analysis of large corpora of speech or text. It stands to reason that while humans are always likely to be better at interpreting small amounts of language, machines have the ability to interpret large amounts at a much faster rate than any human, and to hold this information. As such, problems that require summarisation of a large corpus are ideal for NLP.

2.1.2 Topic modelling

One such problem is dimension reduction of text; *i.e.*, how to express a linguistically complex passage of text with a minimal number of variables. Topic modelling is a statistical process in which latent 'topics', or themes, of a text corpus are revealed in order to ascertain the underlying structure of documents within the corpus. Topic models are being implemented in an increasingly frequent manner to various information engineering applications [9, 31, 33] and, as such, advancements in models have been made to solve a wide range of problems.

As topic models allow a corpus to be summarised by a relatively small number of variables, in particular compared to methods that consider individual words as variables, they are becoming popular as a form of dimension reduction for text data. The concept of 'topics' in this context was first introduced with latent semantic indexing (LSI) [39] in 1998, alongside a model that aimed to solve the problems of synonymy (*i.e.*, separate words with identical meanings, such as *start* and *begin*) and polysemy (*i.e.*, words that have different definitions, such as *book* which can refer to the object or the action of *booking*) that come when classifying by vocabulary only.

Building closely on that model was probabilistic latent semantic indexing (pLSI) [29], which frames LSI in a more rigorous statistical foundation. While these two models are the origins of the discipline of topic modelling, the model most commonly used and referenced is latent Dirichlet allocation (LDA) [13]. LDA is heavily based on pLSI and since its introduction many models have used it as their foundation.

The majority of topic models, including the widely used LDA, are unsupervised methods, in that they incorporate no response variable, or labelling. Therefore, they tend to be used to address machine learning questions about discovering hidden patterns in the data, rather than to explain known patterns. Relatively little work exists in applying topic modelling to statistical regression problems, in which supervised methods generally perform better. While some work has enhanced existing unsupervised models so that they may be developed with some response variable in mind, they are often done to improve the nature of the topics, moreso than for prediction. Additionally, those models relying on the method of LDA, such as supervised LDA (sLDA) [12], will often not incorporate the structure of language. That is, no regard will be paid to the order or class of words within the documents of a corpus.

2.1.3 Definitions

While many methods proposed as topic models are applied in non-literary contexts (for example, applications to gene function prediction [40]), the following literature-based definitions are used in order to aid understanding of the nature of the problem.

- **Vocabulary:** (V) a vector of length v of units, usually words, from which documents are constructed.

- **Topic:** (ϕ) a distribution over the vocabulary; *i.e.*, every word in V is assigned a probability $p_i \in [0, 1], i = 1, 2, ..., v$ with $\sum_{i=1}^{v} p_i = 1$. In general, there are a fixed number k of topics, $\boldsymbol{\phi} = \{\phi_1, ..., \phi_k\}$.

- **Document:** ($\mathbf{w}$) a collection of words $w_1, w_2, ..., w_n$ from the vocabulary V. In some models, including LDA, the order of these words is treated as insignificant, that is they are 'bag of words' models. In other more complex models, the structure of each document is relevant.

- **Corpus:** ($\mathbf{D}$) a collection of m documents over which the topic model is applied, that is $\mathbf{D} = \{\mathbf{w}_1, ..., \mathbf{w}_m\}$, each with length $n_j, j = 1, ..., m$.

- **Topic proportion:** (θ_j) a distribution of topics over the document j, *i.e.*, for every topic in the corpus, a probability between 0 and 1 is assigned of a given word in the document belonging to that topic, with probabilities summing to 1. A corpus will then have an $m \times k$ matrix $\boldsymbol{\theta}$, where each row corresponds to topic proportion θ_j, for $j = 1, 2, ..., m$.

2.2 Differences between topic models

Since the introduction of LDA [13], discussed in more detail in Section 2.3, numerous topic models have been developed, each with differences that make them suitable for various purposes. This section aims to summarise these differences. Table 2.1 shows a summary of common model features for a range of topic models. The differences in a selection of topic models are also conveyed in Figure 2.1, with their chronological development summarised in Figure 2.2.

2.2.1 Supervision

The nature of topic modelling makes it an obvious choice for text summarisation into a form that a regression model could understand, where topic proportions are used as predictor variables. That is, the presence of a topic in a document is a predictor in the regression model. Certain models however, are more immediately suited to this application than others, and they are categorised as supervised models.

Sometimes topic models are applied to problems in which there is some labelling, metadata, or response variable, of documents. In these cases, it would be appropriate to incorporate these labels when finding the topics of the corpus. As topic modeling is a method for dimension reduction, the topic found by any particular model may not necessarily be the only valid way to reduce the corpus dimension. Therefore, the incorporation of labels enable a model to find the most relevant reduction for that problem. In general, supervised models can be divided into two groups: single- and multi-labelled, where a single-labelled model may only assign one label to each document and a multi-labelled model may assign many.

The most widely known of these models is supervised LDA (sLDA) [12], a single-labelled method which builds upon LDA by assuming that in the generative process of creating a document, a response variable is generated based on topic assignments $\mathbf{z}_j$. This model is covered in more detail in

Section 2.4. Similar to this is labelled LDA (L-LDA) [45], which is analogous to sLDA but incorporates multi-labelled corpora.

Some models have been developed specifically to incorporate certain kinds of metadata. For instance, the author-topic model developed by Rosen-Zvi, Griffiths, Steyvers and Smyth [46] allows the user to generate topic distributions based on the author of a work, with topics themselves remaining consistent over the corpus. On the other hand, Topics Over Time (TOT) [55] lets the date a document was created influence the topics themselves (see Section 2.2.4 for more on this).

Other supervised topic models [32, 47], and their differences, are summarised in Figures 2.1 and 2.2, and Table 2.1.

2.2.2 'Bag of words' assumption

For the most part, thanks to the wide implementation of LDA and its simplicity, topic models tend to assume a 'bag of words' approach to linguistic structure; that is, the ordering of words within a document has no effect on the topics found. There is an obvious computational advantage to this assumption, as documents can be treated not as sequences of words with dependencies between them, but as lists of word counts. Therefore, in problems where there is little knowledge gleaned from grammatical information or overall document structure (for example, the Gumtree problem discussed in Chapter 3), it may be prudent to make this assumption.

However, there are situations in which the computational advantage of the 'bag of words' assumption may not outweigh the loss of accuracy in the model from throwing away structural information. Some models already exist that aim to summarise documents in a more complete manner, such as the hidden topic Markov model (HTMM) [25], hidden Markov topic model (HMTM) [4] and the model developed in Griffiths, Steyvers, Blei and Tenenbaum's *Integrating topics and syntax* [24] (referred to as the ITAS model in subsequent mentions), which all incorporate Markov properties and therefore some (limited) sense of structure to the documents. We discuss the HMTM in further detail and use it for prediction in Chapter 4.

Generally, this structure is introduced in one of two ways: the incorporation of linguistic information such as part-of-speech tagging, or with overall document structure and the dependency of topic assignments on those of their neighbours. For instance, the ITAS model [24] suggests a generative process that assigns topics randomly to a word, but the word is then chosen based on an assigned 'class', and the class of the word before it. On the other hand, models such as the HMTM [4] use only the topic assignment of their neighbouring words, *e.g.* by having each topic assignment dependent on the one before it when generating a document, as per the Markov property [38] (as outlined in Section 2.5).

2.2.3 Topic correlation

In general, models tend to assume topic independence for ease of calculation. However, realistically some topics will be more closely related than others, and the information gained from knowing these correlations could be used to better organise and interpret a corpus. Blei and Lafferty describe the correlated topic model (CTM) [11], in which LDA is updated to draw topic proportions from a multivariate Gaussian distribution as opposed to the traditional Dirichlet. This allows the model to incorporate relationships between topics.

The hidden Markov topic model (HMTM) [4] also introduces an aspect of correlation, with the choice of topic assignment for each word dependent on the topic assignment directly preceding it, such that there may be some correlation between adjacent topics. While this would also be expected of the very similar hidden topic Markov model (HTMM) [25], the choice of topic here is independent of the one before, save for the decision to change or remain in the preceding topic. Therefore, the model does not incorporate topic correlations.

2.2.4 Temporal change

Temporal information (*i.e.*, information regarding the change in topics or topic assignments over time) can be incorporated into topic models in two different ways: through dropping the 'bag of words' assumption as discussed in Section 2.2.2 and including document structure, or by looking at how a corpus changes over time (*i.e.*, the 'bag of words' assumption could still hold, but inter-document information changes).

The introduction of inter-document temporal information can occur in various ways. For example, Topics Over Time (TOT) [55] updates topic proportions based on the time the document was created or published. On the other hand, dynamic topic models (DTM) [10] update the topics themselves over time. Overall, it is uncommon for topic models to incorporate this information.

2.3 Latent Dirichlet allocation

First proposed in 2003 [13], LDA is the most widely used and cited topic model, with more recent developments in topic modelling tending to build upon it [10, 11, 12, 32, 34, 45, 47]. Amongst the reasons it is so commonly applied and adapted is its simplicity regarding the assumed structure of the corpus. LDA is a 'bag of words' model, as mentioned in Section 2.2.2, in that the structure of a given document is irrelevant, only the frequency of each word in the vocabulary within the document [9]. Each of the words in a document is then independent, given the model parameters, according to LDA.

LDA assumes the following generative process when creating documents:

1. Generate the k topics $\phi_l \sim \text{Dir}(\beta)$, $l = 1, 2, ..., k$, over a fixed vocabulary of length v.

2. For each document $\mathbf{w}_j \in \mathbf{D}$, $j = 1, 2, ..., m$:

    (a) Let $n_j \sim \text{Poisson}(\xi)$, the length of document j.

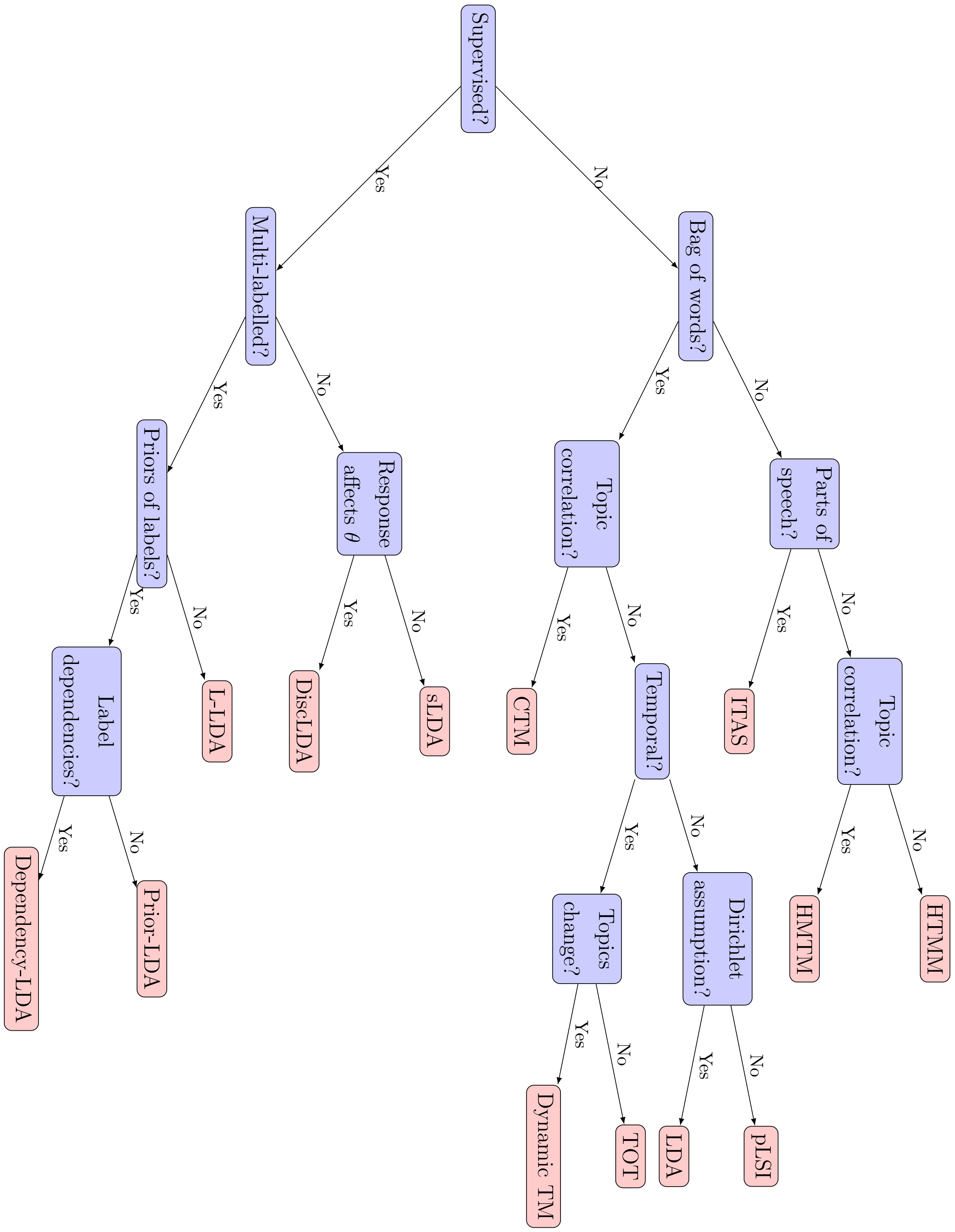


Figure 2.1: Flow chart of topic model features.

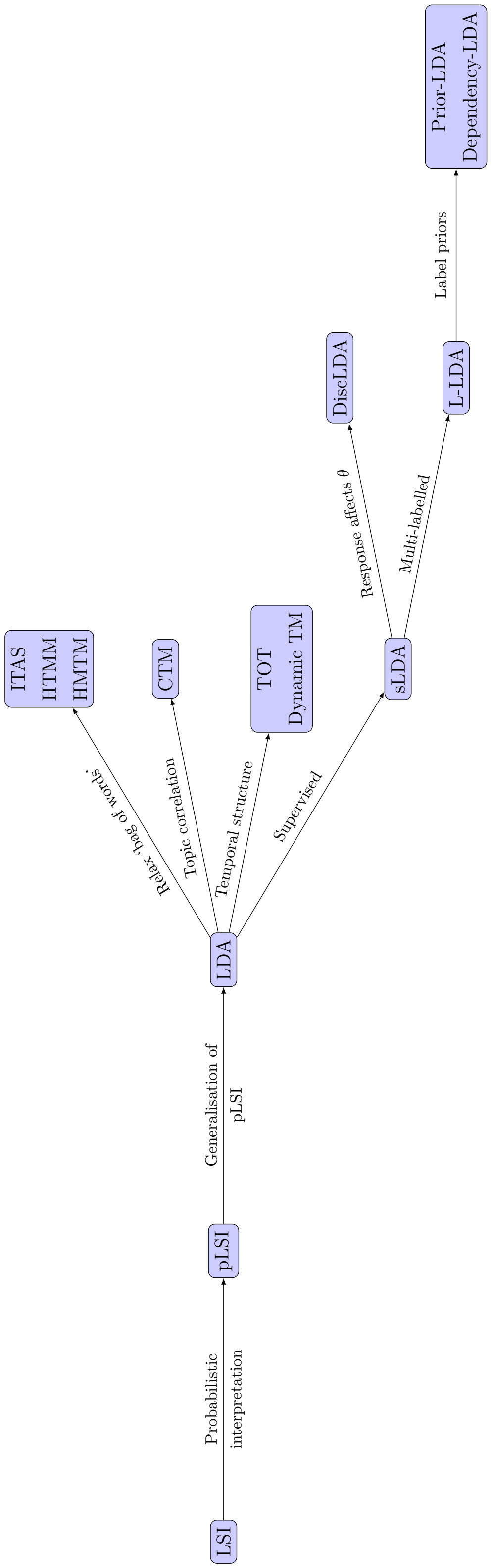


Figure 2.2: Family tree of topic models.

Model	**Supervised**	**Bag of words**	**Topic correlation**	**Temporal**	**HMM**
pLSI [29]		✓			
LDA [13]		✓			
ITAS [24]				✓	✓
CTM [11]		✓	✓		
TOT [55]		✓		✓	
Dynamic TM [10]		✓		✓	
HTMM [25]				✓	✓
HMTM [4]			✓	✓	✓
sLDA [12]	✓	✓			
L-LDA [45]	✓	✓			
DiscLDA [32]	✓	✓			
Prior-LDA [47]	✓	✓			
Dependency-LDA [47]	✓	✓			

Table 2.1: Table indicating the presence of various features in a non-exhaustive selection of common topic models.

(b) Choose the topic proportions $\theta_j \sim \text{Dir}(\alpha)$.

(c) For $i = 1, 2, ..., n_j$:

  i. Choose a topic assignment $z_{ji} \sim \text{Multi}(\theta_j)$.

  ii. Choose a word $w_{ji} \sim \text{Multi}(\phi_{z_{ji}})$.

(d) Create the document $\mathbf{w}_j = \{w_{ji}\}_{i=1,2,...,n_j}$.

Here, α and β are hyperparameters of the distributions of the θ_j, the distribution of topics over document j, and $\boldsymbol{\phi}$, a $k \times v$ matrix where each row

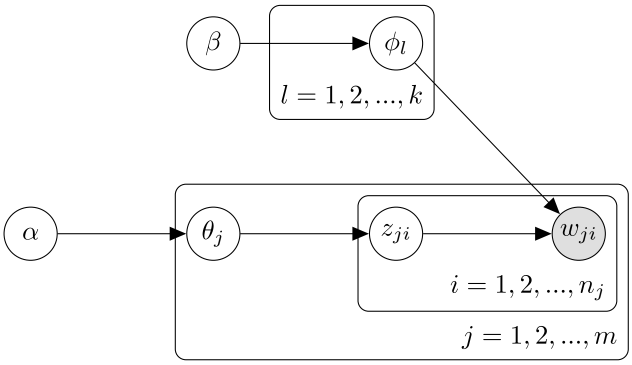


Figure 2.3: Plate diagram of the LDA generative model.

corresponds to the word distribution of a topic ϕ_l, $l = 1, 2, ..., k$, respectively. The plate diagram of this process can be seen in Figure 2.3.

2.3.1 Inference

When topic modelling, we are usually aiming to find both $\boldsymbol{\theta} = \{\theta_1, ..., \theta_m\}$, the distribution of topics over documents, and the topics $\boldsymbol{\phi} = \{\phi_1, ..., \phi_k\}$ themselves, given the corpus $\mathbf{D}$. That is, for an LDA model, we want to find $P(\boldsymbol{\theta}, \boldsymbol{\phi} | \mathbf{D}, \alpha, \beta)$, which can be written as

$$P(\boldsymbol{\theta}, \boldsymbol{\phi} | \mathbf{D}, \alpha, \beta) = \frac{P(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{D} | \alpha, \beta)}{P(\mathbf{D} | \alpha, \beta)}.$$

Where this becomes computationally prohibitive is that the denominator, $P(\mathbf{D}|\alpha, \beta)$, is generally intractable to compute. Given we are looking at the probability of a corpus occurring given the Dirichlet priors, this is unsurprising. Various approximation methods [19, 51, 5] have been proposed to overcome this, including the variational EM algorithm originally proposed by Blei, Ng and Jordan [13], and the commonly used collapsed Gibbs sampling method [42, 5]. We employ the latter when using LDA throughout this thesis.

Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm that enables us to generate samples from a distribution without having to calculate it [16]. That is, we sample from conditional distributions of the distribution in question. Collapsed Gibbs sampling occurs when certain parameters are marginalised out when calculating conditional distributions. In this instance, we are marginalising over $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, in sequence.

In the case of LDA, we wish to find estimates for $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ given our corpus $\mathbf{D}$. The collapsed Gibbs sampling method proposed by Griffiths and Steyvers [23] considers the posterior distribution of the topic assignments, rather than directly estimating the parameters in question, in order to attain these estimates. That is, we are interested in finding $P(\mathbf{z}|\mathbf{D})$; or more specifically, $P(z_{ji}|\mathbf{z}_{-ji}, \mathbf{D})$ for each word i in document j of the corpus, as each assignment is conditional on the rest. Here $\mathbf{z}_{-ji}$ is the vector of length $n_j - 1$ of all topic assignments in document j excluding that of the ith word. Rather than calculating the posterior directly, we can use the more efficient [23]:

$$P(z_{ji} = l|\mathbf{z}_{-ji}, \mathbf{D}) \quad \propto \quad P(w_{ji}|z_{ji} = l, \mathbf{z}_{-ji}, \mathbf{D}_{-ji}) P(z_{ji} = l|\mathbf{z}_{-ji}).$$

Here, $\mathbf{D}_{-ji}$ is the corpus minus the ith word of the jth document (*i.e.*, the word to which we are currently assigning a topic).

We are now simply calculating the product of the probability of a word occurring given its topic assignment and the probability of the topic assignment given the assignments to the rest of the corpus. Marginalising over ϕ_l, the lth topic, and θ_j, the topic proportion of document j, we are able to show that this is equivalent to the product of two expectations of Dirichlet distributions, and as such

$$P(z_{ji} = l|\mathbf{z}_{-ji}, \mathbf{D}) \quad \propto \quad \frac{n_{-ji}^{(w_{ji})} + \beta}{n_{-ji}^{(\cdot)} + v\beta} \times \frac{n_{-ji}^{j} + \alpha}{n_{-i,\cdot}^{j} + k\alpha}$$

where $n_{-ji}^{(w_{ji})}$ is the number of times the word w_{ji} is assigned to topic j in the corpus (excluding the ith word of the jth document), $n_{-ji}^{(\cdot)}$ is the total number of words assigned to topic l, n_{-ji}^{j} is the number of words belonging to topic l in document j and $n^{j}_{-i,\cdot}$ is the number of topics present in document j.

Generally, the Dirichlet hyperparameters α and β are fixed in order to concentrate on the effect the number of topics k has on the model [23], and to speed up computation.

2.3.2 Number of topics

The number of topics in the corpus is assumed to be fixed by the generative model and thus when performing model inference, but there are various methods for determining the best number of topics for a given corpus. Perplexity is a measure of how well a suggested set of topics and topic distributions fits a test set [13], based upon the Shannon entropy or information of the corpus. For some corpus $\mathbf{D}$ of m documents, the perplexity is defined as

$$\text{perplexity}(\mathbf{D}) = \exp\left\{ -\frac{\sum_{j=1}^{m} \log P(\mathbf{w}_j)}{\sum_{j=1}^{m} n_j} \right\}.$$

The lower the perplexity is, the better the fit of topics is to the test corpus. Perplexity is therefore used to choose the number of topics best suited for a given corpus, as well as a measure to compare different topic models [13]. Akaike's Information Criterion (AIC) is another natural choice [53] (and further discussed in Section 2.9), with the advantage that it penalises having more parameters, thus making it less prone to overfitting.

Alternatively, we can use the method outlined in Graham and Ackland [22] and Griffiths and Steyvers [23]. This method takes the harmonic mean of samples generated by Gibbs sampling and uses it as an approximation of the likelihood of the corpus given the model. We choose the number of topics that maximises this result. This model is implemented for its simplicity and speed, and as such is the chosen method throughout this thesis when considering non-predictive problems. However, it is worth noting that it does not share the same penalty for overfitting as found in the AIC measure.

2.3.3 Fragility of LDA

LDA is widely used due to its simplicity, and there has been considerable work performed to build upon it in order to solve particular problems in the topic modelling domain. However, because of its simplicity, as well as its assumptions, LDA has numerous documented limitations in performance. Notable among these is its lack of reliability when it comes to prediction [12], a large part of which is due to its unsupervised nature.

The fragility of LDA has also been noted [50], and while it provides good results when applied to traditional datasets like news articles [13], it performs less well when confronted with corpora that include documents of short length, such as tweets or other social media [50].

In order to ascertain the robustness of LDA, we conduct an experiment by applying the method to the children's novel *Anne of Green Gables* by L.M. Montgomery [37], where the corpus is the novel with the chapters representing separate documents. The structure of prose is such that 'topics' should be somewhat contained within paragraphs, that is in order to introduce another 'idea' we would require a new paragraph. By this logic, for a robust inference of topics, switching two paragraphs from two different documents, especially those contained within the same book, should not affect the overall selection of topics $\boldsymbol{\phi}$ but only the distributions $\boldsymbol{\theta}$ of topics over documents.

The R package **topicmodels** [26] provides tools with which LDA can be performed on a given corpus. Using this package, the *Anne of Green Gables* corpus was evaluated for a fixed number of topics k. Paragraphs within the corpus were then randomly switched, and LDA was again applied to the corpus for the same number of topics. Because the order of topics is random, the Euclidean distance between each of the new topics and the original topics was found using

$$d_{mn} = \sqrt{\sum_{i=1}^{v} \left(\phi_{mi}^{(1)} - \phi_{ni}^{(2)}\right)^2},$$

where $\phi_m^{(1)}$ is the mth topic of the original corpus, $m = 1, 2, ..., k$ and $\phi_n^{(2)}$ is the

nth topic of the new corpus, $n = 1, 2, ..., k$. Topics were then aligned based on minimum distance. That is, the two topics with the smallest distance between them were 'paired' (*i.e.* considered to be corresponding topics in the two corpora), and this process was repeated until all topics were matched to one in the other corpus. The overall distance between $\boldsymbol{\phi}^{(1)}$ and $\boldsymbol{\phi}^{(2)}$, the original and new topic distribution matrices, was then evaluated with

$$\text{dist}_{\boldsymbol{\phi}^{(1)}\boldsymbol{\phi}^{(2)}} = \frac{1}{k}\sum_{j=1}^{k} d_{jj}.$$

The process was then repeated using a varied number of paragraph switches, from 1 to 150. The results can be seen in Figure 2.4, for two iterations of the process. This graph shows the distance between the topics of the original corpus and those of the new corpus with switched paragraphs, for a certain number of switches. As a reference, the difference in topics between *Anne of Green Gables*, and a completely distinct corpus, *Pride and Prejudice* by Jane Austen [6], is represented by the red horizontal line on the graph.

Figure 2.4 shows the fragility of the original topics found for the corpus, in that any more than around 20 paragraph switches cause the new topics to wildly differ. This is also an argument for supervised learning, which allows the model some context in which to choose the most appropriate topics.

2.4 Supervised LDA

Supervised LDA (sLDA) [12] is an adaptation of the LDA model designed for labelled documents. It follows the same generative process as the original LDA, with an additional step to generate the response variable, $y_j|\mathbf{z}_j$ from the topic assignments for the document. That is, for some document j in the corpus, with k fixed topics $\boldsymbol{\phi}$, where $\phi_1, \phi_2, ..., \phi_k \sim \text{Dir}(\beta)$,

1. choose the topic proportion $\theta_j \sim \text{Dir}(\alpha)$.

2. For each word in the document, $i = 1, 2, ..., n_j$:

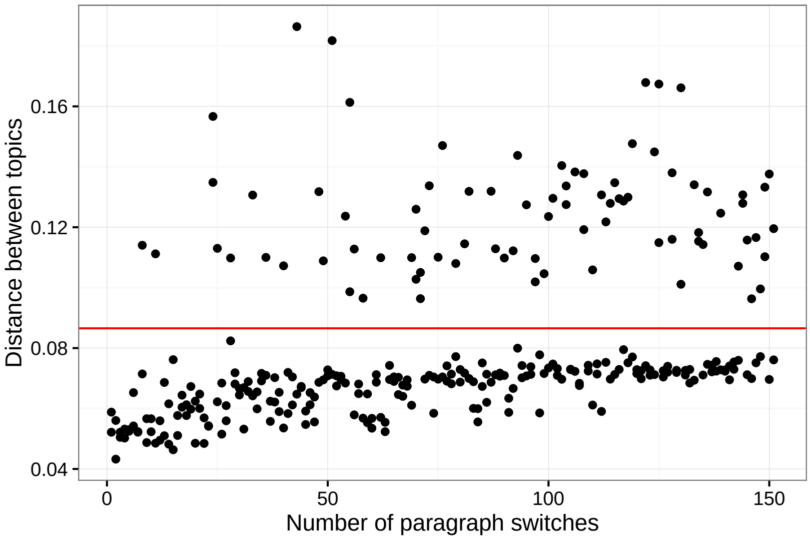


Figure 2.4: Graph showing a measure of the distance between topics between the original corpus and a corpus with switched paragraphs, for different numbers of switches between 1 and 150, for the *Anne of Green Gables* corpus with 10 topics. The horizontal line represents the difference between topics found for the *Anne of Green Gables* corpus and those found for the novel *Pride and Prejudice*.

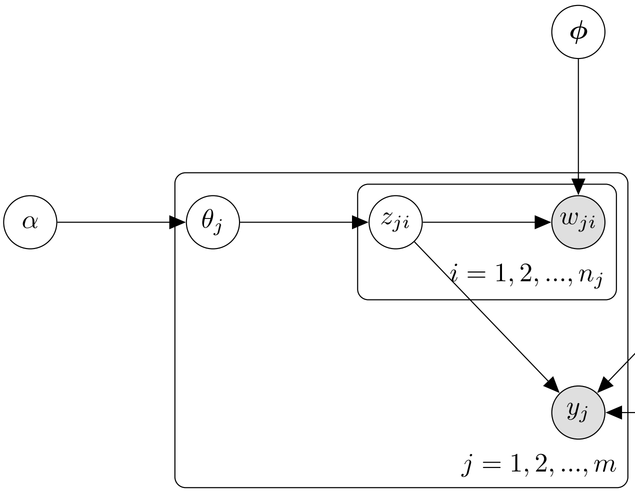


Figure 2.5: Plate diagram of the supervised LDA generative model.

(a) Choose a topic assignment $z_{ji} \sim \text{Multi}\,(\theta_j)$.

(b) Choose a word $w_{ji} \sim \text{Multi}\left(\phi_{z_{ji}}\right)$ from the vocabulary.

3. Draw the response variable $y_j|\mathbf{z}_j \sim \text{N}\left(\eta^T \bar{z}_j, \sigma^2\right)$, where $\bar{z}_j = (1/n_j)\sum_{i=1}^{n_j} z_{ij}$.

The hyperparameters of the Dirichlet distribution from which the topics and topic proportions are drawn are β and α respectively. The normal distribution from which the response variable is drawn has parameters η and σ^2. These parameters are assumed to be known. This generative process is represented in Figure 2.5.

2.4.1 Inference

When generating an sLDA model we are interested in finding the topic proportions $\boldsymbol{\theta} = \{\theta_1, ..., \theta_m\}$, and the topic assignments $\mathbf{z} = \{\mathbf{z}_j\}_{j=1,...,m}$, and we do so with the response variable $\mathbf{y} = \{y_1, y_2, ..., y_m\}$ in mind. Unlike LDA, we treat the topics $\boldsymbol{\phi}$ as unknown constants instead of random variables. That is, we are interested in maximising the following probability:

$$P(\boldsymbol{\theta}, \mathbf{z}|\mathbf{D}, \mathbf{y}, \boldsymbol{\phi}, \alpha, \eta, \sigma^2).$$

Like with LDA, we can write this as

$$P(\boldsymbol{\theta}, \mathbf{z}|\mathbf{D}, \mathbf{y}, \boldsymbol{\phi}, \alpha, \eta, \sigma^2) = \frac{P(\boldsymbol{\theta}|\alpha)\left(\prod_{j=1}^{m} P(\mathbf{z}_j|\boldsymbol{\theta})P(\mathbf{w}_j|\mathbf{z}_j, \boldsymbol{\phi})\right)P(\mathbf{y}|\mathbf{z}, \eta, \sigma^2)}{\int P(\boldsymbol{\theta}|\alpha)\left(\prod_{j=1}^{m} P(\mathbf{z}_j|\boldsymbol{\theta})P(\mathbf{w}_j|\mathbf{z}_j, \boldsymbol{\phi})\right)P(\mathbf{y}|\mathbf{z}, \eta, \sigma^2)d\boldsymbol{\theta}},$$

where m is the number of documents in the corpus. Once again, this probability is intractable to calculate due to the normalising denominator. Therefore, in order to efficiently approximate the sLDA model, we use a variational expectation-maximisation (EM) algorithm. For more detail on this method, please refer to Blei and McAuliffe [12].

When fitting sLDA models in this thesis, we use the R package **lda** [17].

2.4.2 Prediction

The method behind sLDA is specifically developed to handle prediction. As such, we are able to compute the expected response y_j from a document $\mathbf{w}_j$ and the inferred sLDA model $\{\alpha, \boldsymbol{\phi}, \eta, \sigma^2\}$, as follows:

$$E\left[Y_j|\mathbf{w}_j, \alpha, \boldsymbol{\phi}, \eta, \sigma^2\right] \approx \eta^T E_q\left[\bar{\mathbf{z}}_j\right],$$

where $\bar{\mathbf{z}}_j$ are the estimated topic assignments to document $\mathbf{w}_j$, and $E_q\left[\bar{\mathbf{z}}_j\right]$ is the variational posterior distribution of $\bar{\mathbf{z}}_j$.

However, this method assumes an unconstrained response variable y_j. When considering generalised linear models (GLMs), *e.g.* with a categorical response (as we do in this thesis), we must instead compute the following expectation:

$$E\left[Y_j|\mathbf{w}_j, \alpha, \boldsymbol{\phi}, \eta, \sigma^2\right] \approx E_q\left[\mu\left(\eta^T\bar{\mathbf{z}}_j\right)\right],$$

where $\mu\left(\eta^T\bar{\mathbf{z}}_j\right) = E\left[Y_j|\zeta = \eta^T\bar{\mathbf{z}}_j\right]$ and ζ is the natural parameter of the distribution from which the response is taken. Again, further detail on this method is found in Blei and McAuliffe [12].

The number of topics is again fixed with the sLDA model, but can be chosen in the same manner as outlined in Section 2.3.2.

2.5 Hidden Markov models

Hidden Markov models (HMMs) were first developed in the 1960s in a series of papers by Baum [7, 8] with early applications in speech processing [44], but have since been applied to a large range of problems.

An HMM assumes a sequence of latent (or hidden) states $\mathbf{S} = \{S_1, S_2, ..., S_n\}$, generated from m possible states, each of which generates some non-latent observation $\mathbf{O} = \{O_1, O_2, ..., O_n\}$, from v possible observations. The transitions between these states are governed by some $m \times m$ transition matrix $\boldsymbol{\Theta}$, where each choice of state is conditionally independent of others given the state immediately prior to it. That is, the state sequence is a Markov chain and thus obeys the Markov memoryless property,

$$P\left(S_i = s | S_1, S_2, ..., S_{i-1}\right) = P\left(S_i = s | S_{i-1}\right),$$

where S_i is the state at time i in the sequence, for $i = 2, 3, ..., n$. The observations are also conditionally independent of each other given their state, and governed by emission probabilities $\boldsymbol{\phi}$ based on the state.

The model itself can be expressed as $\Omega = (\boldsymbol{\Theta}, \boldsymbol{\phi}, \boldsymbol{\pi})$, where

- $\Theta_{jk} = P\left(S_i = k | S_{i-1} = j\right)$ for $i = 2, 3, ..., n$ and $j, k = 1, 2, ..., m$.
- $\phi_{jl} = P\left(O_i = l | S_i = j\right)$ for $i = 1, 2, ..., n$, $j = 1, 2, ..., m$ and $l = 1, 2, ..., v$.
- $\pi_j = P\left(S_1 = j\right)$, for $j = 1, 2, ..., m$, the starting probabilities of the sequence.

Figure 2.6 visually demonstrates the general structure of an HMM. From this, it is possible to see how topic modelling could be applied to this structure, with the sequence of observations being the words in a document and the states the topic assignments for each of the words.

Rabiner [44] outlines the three main problems that come with HMMs: finding the probability of the observed sequence, choosing the optimal state sequence given an observed sequence, and finding the model parameters.

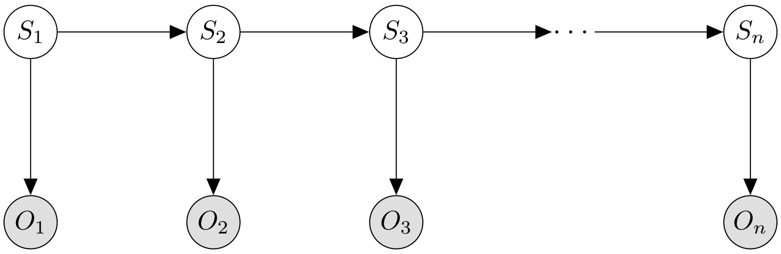


Figure 2.6: Diagram of the general structure of a hidden Markov model.

Within this thesis, we are naturally interested in the third of these: given a sequence of observations $\mathbf{O}$, what are the model parameters $\Omega = (\boldsymbol{\Theta}, \boldsymbol{\phi}, \boldsymbol{\pi})$ that maximise $P(\mathbf{O}|\Omega)$? In a topic modelling context, can we find the model that governs the transitions between topics ($\boldsymbol{\Theta}$), and the topics themselves ($\boldsymbol{\phi}$), given a document in our corpus?

2.5.1 Baum-Welch algorithm

When we are dealing with, in particular large, sequences of observations, with many states and possible observations, we often need a computational method to find the best model given our sequence. The Baum-Welch algorithm is designed to solve this problem, by iteratively calculating forward and backward probabilities given an assumed model, and then updating that model accordingly until convergence. The following steps are undertaken when implementing the Baum-Welch algorithm:

1. Calculate forward probabilities $\alpha_{li} = P(O_1 = o_1, ..., O_i = o_i, S_i = l|\Omega)$ based on the current estimate for the model Ω, for $l = 1, 2, ..., k$ and $i = 1, 2, ..., n$.

   (a) For the first word in the document, and for each topic $l = 1, 2, ..., k$:

   $$\alpha_{l1} = \pi_l \phi_{lo_1}.$$

   (b) For word $i = 2, 3, ..., n$ in the document, and for each topic $l =$

$1, 2, ..., k$:

$$\alpha_{li} = \phi_{lo_i} \sum_{m=1}^{k} \alpha_{m,i-1} \Theta_{ml}.$$

2. Calculate backward probabilities $\beta_{li} = P\left(O_{i+1} = o_{i+1}, ..., O_n = o_n | S_i = l, \Omega\right)$ based on the current estimate for the model Ω, for $l = 1, 2, ..., k$ and $i = 1, 2, ..., n$.

   (a) For the last word in the document, and for each topic $l = 1, 2, ..., k$:

   $$\beta_{ln} = 1.$$

   (b) For each word in the document $i = n-1, n-2, ..., 1$ and for each topic $l = 1, 2, ..., k$:

   $$\beta_{li} = \sum_{m=1}^{k} \beta_{m,i+1} \Theta_{lm} \phi_{mo_{i+1}}.$$

3. Update Ω based on the forward and backward probabilities calculated.

   (a) For $l = 1, 2, ..., k$, $i = 1, 2, ..., n$ and $m = 1, 2, ..., k$, calculate temporary variables $\gamma_l(i)$ and $\xi_{lm}(i)$ such that

   $$\gamma_l(i) = \frac{\alpha_{li}\beta_{li}}{\sum\limits_{m=1}^{k} \alpha_{mi}\beta_{mi}}, \quad \text{and}$$

   $$\xi_{lm}(j) = \frac{\alpha_{li}\Theta_{lm}\beta_{m,i+1}\phi_{mo_{i+1}}}{\sum\limits_{l=1}^{k}\sum\limits_{m=1}^{k} \alpha_{li}\Theta_{lm}\beta_{m,i+1}\phi_{mo_{i+1}}}.$$

   (b) Calculate an updated Ω^*:

   $$\pi_l^* = \gamma_l(1) \quad \text{for} \quad l = 1, 2, ..., k,$$

   $$\Theta_{lm}^* = \frac{\sum\limits_{i=1}^{n} \xi_{lm}(i)}{\sum\limits_{i=1}^{n} \gamma_{li}} \quad \text{for} \quad l, m = 1, 2, ..., k,$$

   $$\phi_{li}^* = \frac{\sum_{j=1}^{n} I_{o_j=i}\gamma_l(j)}{\sum_{j=1}^{n} \gamma_l(j)}.$$

4. Repeat the first three steps until Ω^* has converged satisfactorily.

We use this algorithm in Chapter 4 when fitting topic models that include document structure.

2.6 Hidden Markov topic model

The hidden Markov topic model (HMTM) [4] is one of a few topic models currently in existence that build off the structure of the HMM to infer topics. The emission probabilities for each 'state' in an HMM are analogous to the topics in the HMTM, since topics are a probability distribution over our vocabulary of a word appearing given its topic assignment (or 'latent' state). The HMTM assumes the following generative process for documents in its corpus:

1. Generate the k topics $\phi_l \sim \text{Dir}(\beta)$, for $l = 1, 2, ..., k$.
2. For each document $j = 1, 2, ..., m$:
    (a) Generate starting probabilities $\boldsymbol{\pi}_j \sim \text{Dir}(\alpha)$.
    (b) For each topic $l = 1, 2, ..., k$:
        i. Generate the lth row of the transition matrix $\boldsymbol{\Theta}_j$, $\Theta_{jl} \sim \text{Dir}(\gamma_l)$.
    (c) Choose the topic assignment for the first word $z_{j1} \sim \text{Multi}(\boldsymbol{\pi}_j)$.
    (d) Select a word from the vocabulary based on the topic assignment for the first word, $w_{j1} \sim \text{Multi}(\phi_{z_{j1}})$.
    (e) For each subsequent word in the corpus $i = 2, 3, ..., n_j$:
        i. Choose topic assignment z_{ji} based on transition matrix $\boldsymbol{\Theta}_j$.
        ii. Select a word from the vocabulary based on the topic assignment, $w_{ji} \sim \text{Multi}(\phi_{z_{ji}})$.
    (f) Create the document $\mathbf{w}_j = \{w_{ji}\}_{i=1,2,...,n_j}$.

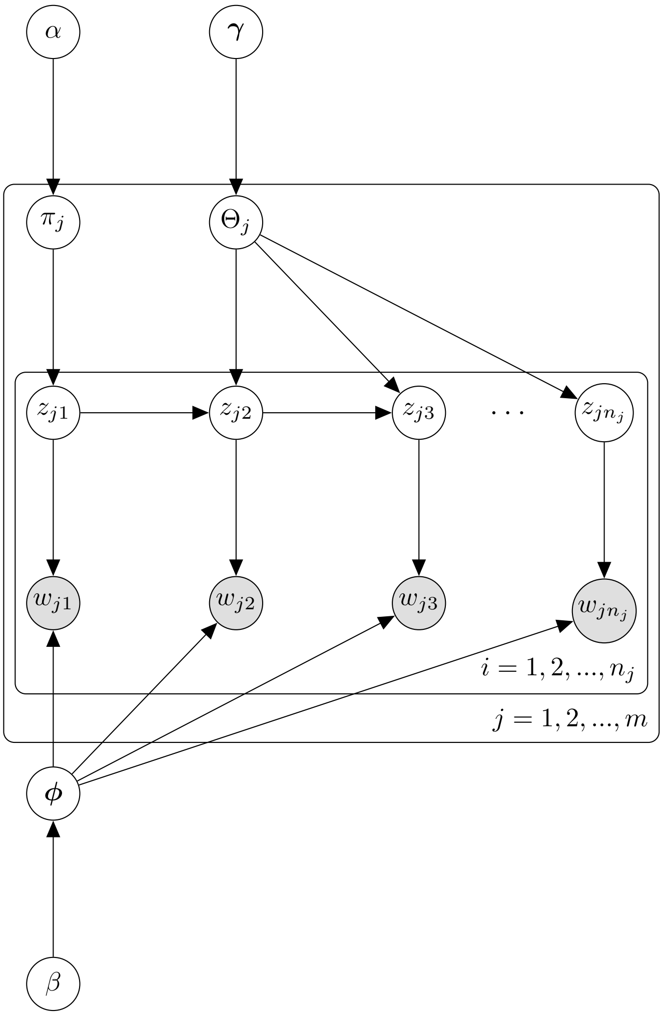


Figure 2.7: Plate diagram of the HMTM generative model.

This structure can be seen in Figure 2.7. Here α, β and $\boldsymbol{\gamma} = \{\gamma_1, ..., \gamma_k\}$ are Dirichlet priors of the starting probabilities, topics and transition matrices respectively.

This differs from the hidden topic Markov model (HTMM) [25] (another, similar topic model which is based on the structure of an HMM) in that a topic assignment is generated for each word, not each sentence, and that there is the potential for correlations between topics. For these reasons, when looking at incorporating a Markov structure into predictive topic modelling in this thesis, the HMTM is used.

2.6.1 Inference

When finding the HMTM, we are interested in maximising the posterior

$$P\left(\boldsymbol{\phi}, \alpha, \beta, \boldsymbol{\gamma} | \mathbf{D}\right) \propto P(\mathbf{D}|\boldsymbol{\phi}, \alpha, \boldsymbol{\gamma})P(\boldsymbol{\phi}|\beta)P(\alpha, \beta, \boldsymbol{\gamma}).$$

However, this problem is, like with LDA and sLDA, intractable and we must therefore use an approximation method. The method proposed by Andrews and Vigliocco [4] is a collapsed Gibbs sampler, as used for LDA and outlined in Section 2.3.1. In this case, the Gibbs sampler draws samples over Dirichlet parameters α, β and $\boldsymbol{\gamma}$, and topic assignments for the corpus $\mathbf{z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_m\}$. In order to save computation, it is able to integrate over model parameters $\boldsymbol{\phi}$, $\{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, ..., \boldsymbol{\pi}_m\}$ and $\{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, ..., \boldsymbol{\Theta}_m\}$, for the m documents of the corpus.

These model parameters can then be recovered given the found topic assignments and hyperparameters. The HMTMs in this thesis are found using the Python code provided by Mark Andrews [3].

2.7 Cleaning data

While preliminary processes are undertaken with any dataset to ensure consistency and accuracy, natural language processing generally requires a more thorough and specific approach. Most text data contains a lot of information not relevant to, or interpretable by, the model at hand. Before applying any kind of topic model to a corpus, it is advisable to first 'clean' the data, that is, strip the text down to only the necessary information. This not only makes for a smaller, and therefore more efficient, corpus to process, but is also designed to remove any non-informative data. The information removed from the corpus depends on both the model in use and the dataset at hand. Depending on the necessary information, the following measures can be considered.

- **Removing case**: in general, the case of a letter or word contains very little information, and as such making all letters in the corpus lowercase reduces the vocabulary significantly without much loss.

  - e.g. *Example* to *example*

- **Removing punctuation and numbers**: 'bag of words' models, in particular, do not rely at all on sentence structure and therefore having punctuation and non-alphabetic characters is unnecessary. Some other models may require this to be left in. Once again, this process will reduce the size of the vocabulary.

  - e.g. *example 1:* to *example*

- **Stemming**: often the model and dataset will be used in a way that relies only on lexical information, and not grammatical. As such, we are able to remove grammatical morphemes from words in order to reduce vocabulary size and ensure the model recognises lexically identical words. When stemming words in this thesis, we use the stemming algorithm developed by Porter for the Snowball stemmer project [43].

  - e.g. {*example, examples*} to *example*

- **Culling the vocabulary**: in some cases, it becomes both more efficient and more accurate to work with a subset of the original vocabulary. Mostly, this means removing stop words (which are common words such as *the* which generally serve a purely grammatical purpose in a sentence, *i.e.*, they are function words and are filtered out by a machine when performing NLP), that contribute little to the lexical meaning of the document. Another common process is tf-idf [52], explained below. When removing stop words in this thesis, we use the (English language) list compiled, once again, in the Snowball stemmer project [43].

2.7.1 tf-idf

Term frequency–inverse document frequency (tf-idf) [52] is a process used to cull the vocabulary of a text corpus. It scores words based on how frequent they are, versus the number of documents in which they appear. That is, a higher score is given to those words that appear frequently, but this score is reduced if the word appears in numerous documents. This ensures words such as stop words are filtered out by the process, as well as very infrequent words. The tf-idf score is composed of two parts: the term frequency (tf) and the inverse document frequency (idf). They are defined as follows.

Consider the frequency f_{ji} of word i in document j. Then,

$$\mathrm{tf}_{ji} = \frac{f_{ji}}{\max\limits_{k \in V} f_{jk}},$$

where V is the vocabulary of the corpus. The inverse document frequency of word i is defined as

$$\mathrm{idf}_i = \log_2 \left(\frac{N}{n_i} \right),$$

where N is the number of documents in the corpus, and the word i appears in n_i of them. Calculating the tf-idf is then simply a matter of taking the product of these two values. That is, the tf-idf of word i in document j is

$$\text{tf-idf}_{ji} = \mathrm{tf}_{ji} \times \mathrm{idf}_i.$$

We can then cull any words in the vocabulary whose tf-idf score is below some threshold, for instance the median average of the tf-idfs for the entire vocabulary.

2.8 Logistic regression

The purpose of this thesis is to apply topic models as a preprocessing step for text data in order to use that data in a predictive regression framework. The problems to which we apply these methods all consist of predicting some dependent variable that is categorical, and therefore we employ a logistic regression framework.

Binomial logistic regression models [27] return the probabilities of some categorical variable Y being either 0 or 1 given data $\mathbf{X} = \mathbf{x}$. The model has the form

$$\log\left(\frac{P\left(Y=1|\mathbf{X}=\mathbf{x}\right)}{P\left(Y=0|\mathbf{X}=\mathbf{x}\right)}\right) = \beta_0 + \boldsymbol{\beta}^T\mathbf{x},$$

where the β_j are the model coefficients. Rearranging this gives

$$\begin{aligned} P\left(Y=1|\mathbf{X}=\mathbf{x}\right) &= \frac{\exp\left(\beta_0+\boldsymbol{\beta}^T\mathbf{x}\right)}{1+\exp\left(\beta_0+\boldsymbol{\beta}^T\mathbf{x}\right)}, \quad \text{and} \\ P\left(Y=0|\mathbf{X}=\mathbf{x}\right) &= \frac{1}{1+\exp\left(\beta_0+\boldsymbol{\beta}^T\mathbf{x}\right)}, \end{aligned}$$

suiting the requirements that each probability is between 0 and 1 and that they sum to 1.

Generally, we fit a logistic regression model using the maximum likelihood method. Given N data points $(\mathbf{x}_j, y_j)$ for $j = 1, 2, ..., N$, the log likelihood can be written as

$$l(\boldsymbol{\beta}) = \sum_{j=1}^{N}\left(y_j\boldsymbol{\beta}^T\mathbf{x}_j - \log\left(1+\exp(\boldsymbol{\beta}^T\mathbf{x}_j)\right)\right).$$

This is usually maximised using some iterative algorithm, such as the Newton-Raphson algorithm [27].

We can extend the above to a situation where Y can take more than two values, $\{y_1, y_2, ..., y_k\}$ (*i.e.*, multinomial logistic regression). The model in this case is written as

$$\begin{aligned} P\left(Y=y_i|\mathbf{X}=\mathbf{x}\right) &= \frac{\exp\left(\beta_{i0}+\boldsymbol{\beta}_i^T\mathbf{x}\right)}{1+\sum_{l=1}^{k-1}\exp\left(\beta_{l0}+\boldsymbol{\beta}_l^T\mathbf{x}\right)}, \quad \text{for} \quad i=1,2,...,k-1, \quad \text{and} \\ P\left(Y=y_k|\mathbf{X}=\mathbf{x}\right) &= \frac{1}{1+\sum_{l=1}^{k-1}\exp\left(\beta_{l0}+\boldsymbol{\beta}_l^T\mathbf{x}\right)}, \end{aligned}$$

once again returning probabilities that sum to 1. We use an approximation method to find an estimation of the maximum likelihood in order to fit the model.

Multinomial regression models in this thesis are found using the R package **nnet** [54].

2.9 Model validation

Whenever we fit regression models, we aim to fit the best possible model. To that end, we require some method that will allow us to compare the accuracy of different models. Which method we use depends on the models being compared.

2.9.1 Cross validation prediction error

Cross validation is a way of using the data on which a model is built, to also test that model's validity. Specifically, K-fold cross validation [21] finds the prediction error for a model given the existing data. That is, this method partitions a dataset, for example our corpus, into K distinct folds, and for each fold i and a given potential model:

1. the documents in fold i are chosen to be the 'test' set of documents,
2. the documents in the remaining $K-1$ folds are then the 'training' set of documents, and
3. the responses for all documents in fold i are predicted from the model rebuilt on the training set.

Then, for each fold i, the mean square error (MSE) is calculated as

$$\mathrm{MSE}_i = \sum_{j \in C_i} \frac{1}{m_i} \left(y_j - \hat{y}_j\right)^2,$$

where m_i is the number of documents in the ith fold, $\hat{y}_j$ is the model estimate of the response y_j and C_i is the set of documents in the ith fold. Repeating this process for each of the folds yields the prediction error,

$$\mathrm{CVPE}_K = \sum_{i=1}^{K} \frac{m_i}{m} \mathrm{MSE}_i,$$

where m is the number of documents in the corpus.

It follows that the better a model performs, the smaller the MSE and thus CVPE. Therefore, this may be used as a measure to discriminate between models with different numbers of topics, in order to choose the best number with regards to the response.

2.9.2 Akaike information criterion (AIC)

The Akaike information criterion (AIC) [2] of a regression model is a measure of comparitive fit, that penalises overfitting. That is, it can be used to compare how various models perform on the same data. The AIC of a model is

$$\mathrm{AIC} = 2p - 2\log(\hat{L}),$$

where $\hat{L}$ is the maximised value of the likelihood function for the model, and p is the number of parameters in the model. We therefore choose the model with the minimum AIC. The penalisation term $2p$ aims to reduce overfitting by favouring less predictors.

2.9.3 Receiver operating characteristic (ROC) curves

A receiver operating characteristic (ROC) curve is a graphical way of representing and comparing the performance of a model or models [20]. The ROC graph compares true positive rate (TPR) and false positive rate (FPR) at different threshold levels, where

$$\begin{aligned} \mathrm{TPR} &= \frac{\text{number of correctly identified positive values}}{\text{number of true positive values}}, \quad \text{and} \\ \mathrm{FPR} &= \frac{\text{number of correctly identified negative values}}{\text{number of true negative values}} \end{aligned}$$

at each threshold. For that reason, they are generally used to compare models with binary responses, such as logistic regression models. These models, when used for prediction, return some value between 0 and 1 for any new document, representing the probability of that document being labelled as positive. Here, a threshold level is the minimum value at which we consider a response to be positive. *E.g.*, at the threshold level 0.1, any response above 0.1 is considered positive and any below, negative. These are then compared to the true positive and negative values of the corpus in question. It therefore follows that all ROC curves will start at the point $(0, 0)$ and end at $(1, 1)$.

The area under the curve (AUC) is a common numerical measure of the performance of the model in question. An ideal model would have an AUC of 1 and a model in which each data point is always incorrectly identified would have an AUC of 0. If we were to randomly assign a prediction value to each data point, we would expect an AUC of 0.5.

We use ROC curves and their AUC to evaluate and compare prediction models in Chapter 3.

2.9.4 Brier score

The Brier score [15, 56] is a measure that can be applied to models with a categorical response, and has applications in weather forecasting. The Brier score calculates the performance of a predictive model as follows:

$$\mathrm{BS} = \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{s} \left(\hat{y}_{ji} - o_{ji}\right)^2 ,$$

where $\hat{y}_{ji}$ is the probability according to the model of document j belonging to category i, and

$$o_{ji} = \begin{cases} 1 & \text{if} \quad \text{document } j \text{ belongs to category } i \\ 0 & \text{if} \quad \text{document } j \text{ does not belong to category } i \end{cases} \tag{2.1}$$

for document $j = 1, 2, ..., m$ and category $i = 1, 2, ...s$. Each term in the sum goes to zero the closer the model gets to perfect prediction, and as such our aim is to minimise the Brier score in choosing a model. The maximum value the score can take here is 2.

We employ the Brier score for validating document prediction in Chapter 4.

Chapter 3

Topic modelling regression and supervised learning

3.1 Introduction

In order to demonstrate how to use topic models in a regression framework, in this chapter we apply them to a problem in online advertising. This is an area where there are typically a large number of documents from which we want to extract some overarching pattern or theme. For that reason, this chapter looks at the application of topic models to predict a binary response from a dataset taken from the trading website, Gumtree[1].

This dataset consists of 4159 advertisements taken from the Gumtree website over three days in February 2016. Each advertisement pertains to the sale of a cat or cats. The problem we are interested in is whether it is possible to predict the 'relinquished status' of a cat in an advertisement, purely from the text of that advertisement.

Relinquished animals are pets that have been given up by their owner after a period of time, as opposed to cats that have been sold, either by breeders or former owners. They are increasingly becoming a problem, with more pets being relinquished than readopted. For example, in the 2015-2016

[1] `www.gumtree.com.au`

financial year the Royal Society for the Prevention of Cruelty to Animals (RSPCA) were required to euthanise over 16,000 cats in Australia [1].

We use predictive topic modelling to determine which advertisements pertain to relinquished cats, based on the words in the description. The motivation behind this is that in doing so, we may be able to understand the reasons why owners are giving up their pets, and therefore how to reduce the scale of this problem. Also, this model could be used to automatically monitor the number of relinquished animals in the future.

For the purposes of this chapter, we are interested in only the advertisements' text descriptions, and whether the cat is being relinquished. While it is possible to include other variables into our model (for instance, the age of the cat is likely to be a good indicator of relinquished status, with a kitten unlikely to be relinquished), this chapter will focus on unstructured textual features, and so only the text description will be used as a predictor.

The relinquished status of each advertisement was manually labelled by an expert, Dr Susan Hazel from the University of Adelaide, School of Veterinary Science.

3.2 Preliminary data analysis

Figure 3.1 shows that of the 4,159 advertisements, 2,187 correspond to relinquished cats and 1,964 to non-relinquished. Inspection of the data showed that eight of the advertisements have not been labelled. When applying the methods below, these eight advertisements have been removed from the corpus.

The advertisement descriptions range in length from two words, to 526. A histogram of document lengths can be seen in Figure 3.2. The median number of words in a description is 35. The distribution of document lengths for the corpus is heavily right skewed, indicating that while some documents have hundreds of words the majority are much shorter. As demonstrated in Section 2.3.3, topic models are not designed for documents of a short

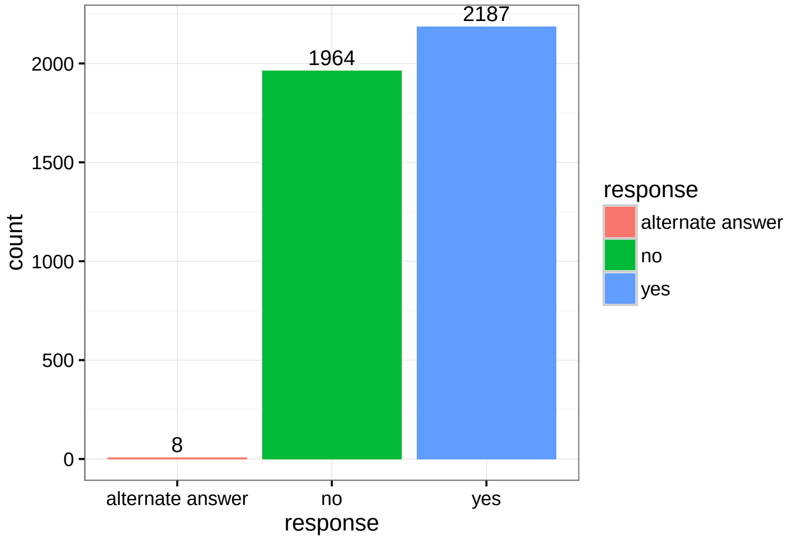


Figure 3.1: Column graph of the frequency of responses to relinquished status for each advertisement in the Gumtree dataset.

length and as such, this could potentially cause performance problems when applying them to this data set.

The vocabulary consists of 23,423 unique words. We performed the following steps on the documents (outlined in Section 2.7) to make the data more consistent and compact:

- removal of punctuation and numbers,
- setting of case to lower case,
- removal of stop words (*i.e.*, 'function words', or common words that contribute very little to the lexical meaning of a sentence), and
- removal of grammatical information from words (*i.e.*, 'stemming').

It should be noted that, as this chapter is only concerned with 'bag of words' models (*i.e.*, models in which words are treated independently of each other and there is no document or linguistic structure), we should be able to remove stop words without much loss of information. In general, stop words

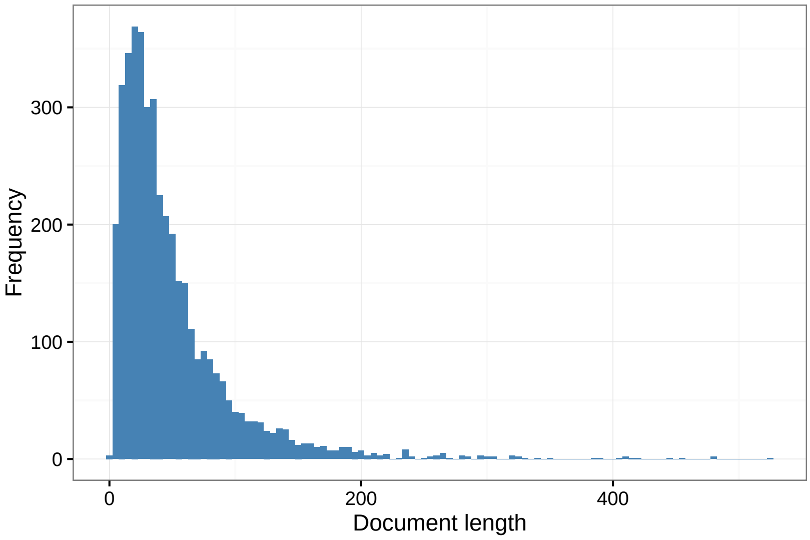


Figure 3.2: Histogram of advertisement description lengths for the Gumtree dataset.

serve a purely grammatical purpose (that is, they are function words) and as such have very little input into a 'bag of words' model, as their meaning is derived from their position in a sentence. The same argument may be applied to stemming, as this process also only removes grammatical information. For this corpus, preliminary results on the non-stemmed advertisements showed the models performed worse and thus we choose to stem words for this problem. Care should be taken when applying the same processes to more linguistically complex corpora, or those with significant grammatical information.

After these processes, the vocabulary is now exactly 13,000 words long. The 20 most frequent words in the cleaned corpus are shown in Figure 3.3. As expected, words such as *kitten* and *cat* feature highly in the corpus. Interestingly, the word *home* is mentioned more often than *cat*.

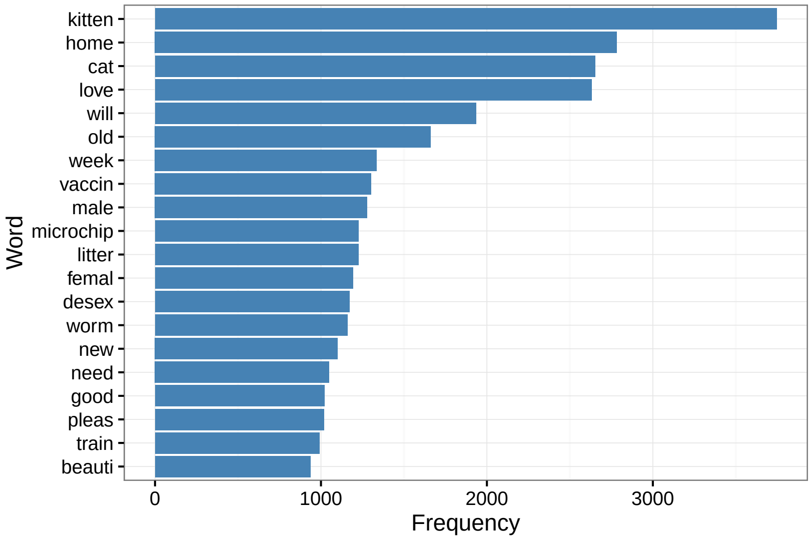


Figure 3.3: The 20 most frequent words in the cleaned Gumtree corpus, against the number of times they appear. As these words have undergone processing, some appear to be spelt incorrectly, such as *vaccin*. This is due to the stemming process outlined in Section 2.7.

3.2.1 Cross-analysis

Figure 3.4 shows the relative histograms of document length for both relinquished and non-relinquished status advertisements. It is apparent from this that the relinquished status has very little impact on how long a document could be; that is, it is unlikely we can infer anything from the length of the document.

We instead look at how the vocabulary differs between the two groups. Figure 3.5 shows the most frequent words for relinquished and non-relinquished documents. Unsurprisingly, although *kitten* features in both sets of documents, it is the most prominent in the non-relinquished group. One would expect a non-relinquished animal to be reasonably young due to their greater resale value, and as such this makes sense. The same argument can be made by the frequency of *cat* in both sets. An interesting point is the more prevalent use of emotive language in the advertisements pertaining to relinquished

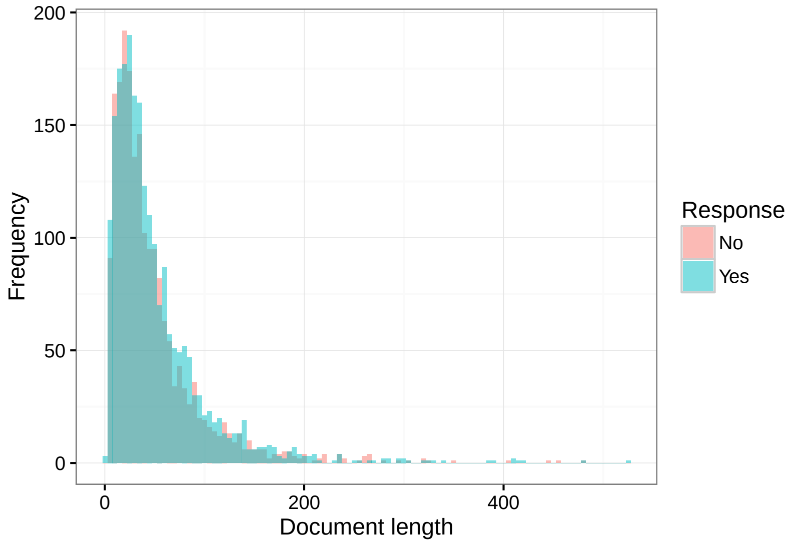


Figure 3.4: Histogram comparing the advertisement description lengths for non-relinquished and relinquished animals in the Gumtree dataset.

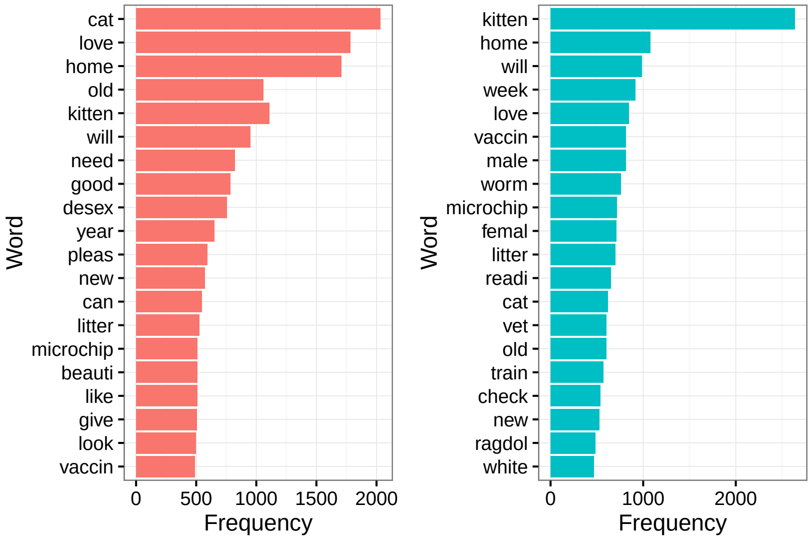
cats. Words such as *home* and *love* appear far more frequently in them, which again would be expected given that they are about animals the writers have had for a length of time and to which they would be emotionally attached.

3.3 Word count models

Rather than just comparing topic models, it would be prudent to see how they compare to a 'baseline' model; that is, if there is any value to finding topics over simply using the individual words in the documents as predictors. To that end, this section explores the application of word count models, or models where the predictors are the number of times each word appears in the document, to the Gumtree problem.

For each model generated in this chapter, the 'clean' corpus will be used; that is, the corpus with the reduced vocabulary and removed rows corresponding to missing relinquished status constructed in Section 3.2 above.

In order to use word counts as predictors, it is necessary to first convert



(a) Relinquished status (b) Non-relinquished status

Figure 3.5: Bar graphs of the most frequent words for both relinquished and non-relinquished status advertisement descriptions in the Gumtree corpus.

our documents to a suitable form. Each document is represented by a vector of the length of the vocabulary (in this case, 13,000 words), with each entry being the number of times that particular word in the vocabulary appears in the document. Unsurprisingly, this vector tends to be sparse, especially for short documents such as the Gumtree advertisements. The corpus is therefore represented as a $d \times v$ matrix, with d being the number of documents in the corpus (4,151) and v the length of the vocabulary (13,000). For this corpus, only 1% of the entries for the document term matrix are non-zero.

3.3.1 Step-up word count model

To determine the best word count model, a step-up procedure was performed where each subsequent model was chosen based on its Akaike Information Criterion (AIC) [2],

$$\mathrm{AIC} = 2p - 2\log(\hat{L}),$$

where $\hat{L}$ is the maximised value of the likelihood function for the model, and p is the number of parameters in the model. This procedure therefore balances the number of predictors and quality of fit in the model, due to the penalisation term $2p$. The algorithm for the step-up procedure is as follows:

1. Start with the null model (that is, a constant model with no predictors).
2. Find the AIC for each model with one word (predictor) added to the model found in the previous step.
3. Choose the model with the minimum AIC from the current and new models.
    (a) If this is a new model, set this as the current model and repeat Step 2.
    (b) If this is the current model, choose that as the final model and end the procedure.

We choose to perform a step-up procedure here, as opposed to a step-down procedure, due to the large number of potential predictors. With this large a vocabulary and thus set of predictors, it may be necessary to further decrease the number of unique words in our corpus so that model calculation is feasible. It is worth noting that this step is in itself an argument for topic modelling, which has the capability to encompass the entire vocabulary. There are a few ways in which we may cull our vocabulary, without much loss of information, but still providing a great increase in computation speed. The most straightforward of these in terms of computation is to take only words that appear with a certain frequency through the corpus (irrespective of the number of documents in which they appear), the idea being that these are more likely to be general predictors of relinquished status for a larger number of documents. While some uncommon words may appear in solely relinquished or non-relinquished advertisements, and therefore be great predictors for those documents, they are less likely to be applicable to a large proportion of the corpus and thus we leave them out here.

At each iteration i in the step-up algorithm, it is necessary to calculate the AIC of $N - i$ logistic regression models, where N is the number of words in the vocabulary. As this is generally in the tens of thousands, this process is highly inefficient. Performing the same process on only a few hundred of the most common words in the corpus is much more reasonable.

To this end, the Gumtree corpus vocabulary was reduced to only the 214 most common words (all words that appear in more than 2.5% of the documents) for the step-up word count model, and the model produced can be found in Appendix A.1.

The chosen model uses 97 predictors. Figure 3.6 shows the coefficients for the 20 most significant predictors in the model. The coefficients are consistent with what was found in the preliminary analysis of the data, and what would be expected. For instance, the *kitten* coefficient is negative, indicating it is more predictive of non-relinquished status, while *cat* has a positive coefficient.

As another indicator of age, *year* suggests relinquished status while *week* suggests non-relinquished. The most positive coefficient in the figure belongs to *rescu*, implying that the rescued status of an animal is, predictably, a good indication of relinquished status. As mentioned before, more emotive words also tend to be a relinquished indicator, like *need* and *good.*

3.3.2 Model cross validation

While the model given above certainly seems to make sense, it is important to verify its predictive capability through cross validation. We do so by separating the corpus into a training and test data set, with documents in the training set used to create the regression model. We then predict responses for all documents in the test set. This enables us to compare the results given by the model to the true results, and thus ascertain how well the model is performing.

For cross validation of the models in this chapter, the process is repeated 100 times, each time with a random 95% of documents forming the training

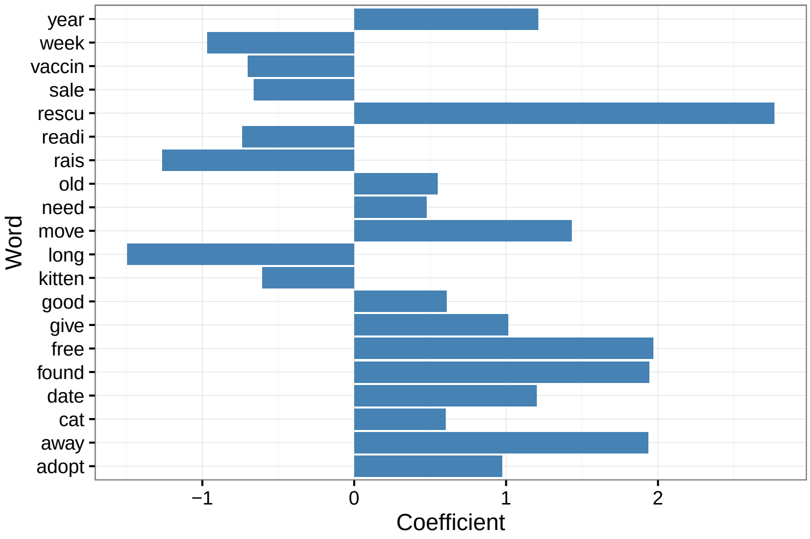


Figure 3.6: The 20 most significant terms in the common words step-up model of the Gumtree dataset with their coefficients. As these words have undergone processing, some appear to be spelt incorrectly, such as *vaccin*. This is due to the stemming process outlined in Section 2.7. The positive coefficients indicate an increased chance of an advertisement having relinquished status, given the presence of that word, and vice versa for the negative coefficients.

set to predict the remaining 5%. This differs from K-fold cross validation, outlined in Section 2.9.1, in that the training sets for each iteration are chosen independent of each other, and so test set values may be reused.

ROC curves

A receiver operating characteristic (ROC) curve (as outlined in Section 2.9.3) is a graphical way of representing and comparing the performance of a model or models [20]. The ROC graph compares the true positive rate (TPR) and false positive rate (FPR) at different threshold levels, where

$$\begin{aligned} \text{TPR} &= \frac{\text{number of correctly identified positive values}}{\text{number of true positive values}}, \quad \text{and} \\ \text{FPR} &= \frac{\text{number of correctly identified negative values}}{\text{number of true negative values}} \end{aligned}$$

at each threshold. Logistic regression models, when used for prediction, return a value between 0 and 1 for any new document, representing the probability of that document being labelled as positive (in this case, being labelled as relinquished).

As mentioned in Section 2.9.3, the area under the curve (AUC) is a measure of the performance of a model from a ROC curve. An ideal model would have an AUC of 1 and a model in which each data point is always incorrectly identified would have an AUC of 0. If we were to randomly assign a prediction value to each data point, we would expect an AUC of 0.5.

Figure 3.7 shows the threshold-averaged ROC curve for the common word step up model, based on algorithms outlined in Fawcett [20]. A threshold-averaged ROC curve, as its name suggests, is composed from averaging points from a set number (in this case, 100) of standard ROC curves. That is, the process for generating a ROC curve has been repeated 100 times with random samples of 5% of the corpus, with the remaining 95% acting as the training data for the model. For each threshold value, the corresponding point from each standard ROC curve is taken and averaged to form the threshold-averaged ROC curve. From this, we can see that the word count

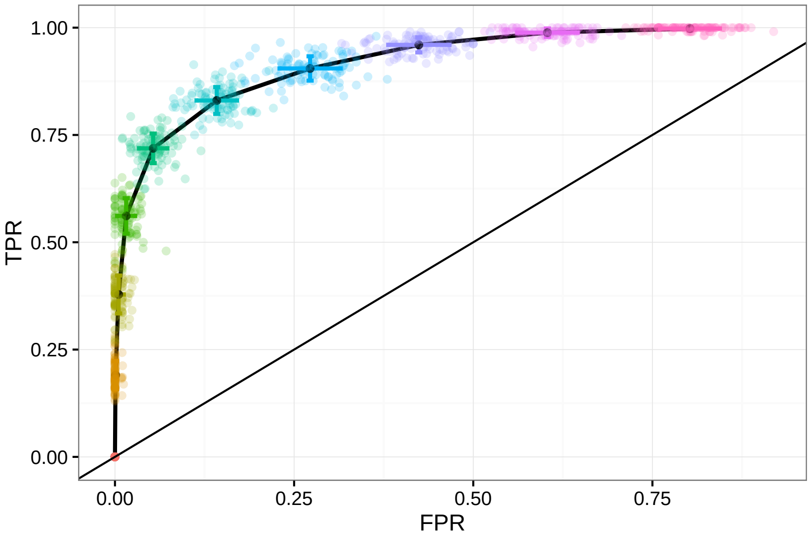


Figure 3.7: Threshold-averaged ROC curve for the step up common word count model cross validation of the Gumtree dataset. The straight black line here represents the 'baseline' case, *i.e.*, the expected value of the ROC given random assignments of probabilities to each document. The ROC has been evaluated 100 times at 10 threshold levels, represented by the different colours in the graph. Each threshold level has a confidence interval for the value of both the true positive rate (TPR) and false positive rate (FPR), shown with the coloured bars. Finally, these threshold levels are connected by the black ROC curve. The area under the curve (AUC) is 0.9264.

model performs reasonably well (that is, clearly better than a random assignation would). Specifically, the threshold-averaged ROC curve for this model has an AUC of 0.9264 (with a 95% confidence interval of $(0.9234, 0.9294)$).

For the rest of this chapter, this model serves as the 'benchmark' for any future models. That is, are we able to either improve or maintain performance with a dimension-reduced and more efficient model involving topics?

3.4 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA) [13], as explained in Section 2.3, could be considered the ‘standard’ topic model, in that most models developed to this point build upon it. For that reason, this section investigates the use of LDA as a data preprocessing step in our logistic regression model to determine the relinquished status of advertisements. As an unsupervised process however, it is not expected that this will be the ideal method.

LDA, like all models used in this chapter, makes a ‘bag of words’ assumption about documents. That is, it considers only the frequency of words that appear, and not their structure or position within the document. When we are considering online advertisements, this assumption saves a large amount of computational power, with potentially limited ramifications on the accuracy of the model. This assumption would need to be weighed more carefully in cases with longer and more structurally complex corpora.

The other assumptions LDA makes about the corpus can be seen in its generative model, under which it assumes documents are created. This generative process is outlined in Section 2.3.

In order to extract topics from a given corpus, such as our Gumtree dataset, we must use some approximation method, as to calculate the likelihood of any given set of topics is computationally infeasible (which is apparent from the sheer number of possible documents that could be created from a set of topics). In this chapter, collapsed Gibbs sampling [23] is used, a form of MCMC sampling specifically designed to handle multivariate cases. More on this method can be seen in Section 2.3.1.

3.4.1 tf-idf

In addition to the text preprocessing already taken, we also apply tf-idf, outlined in Section 2.7.1. This step is useful for removing words that are either too common or not common enough to distinguish between documents, thus improving the efficiency of the model.

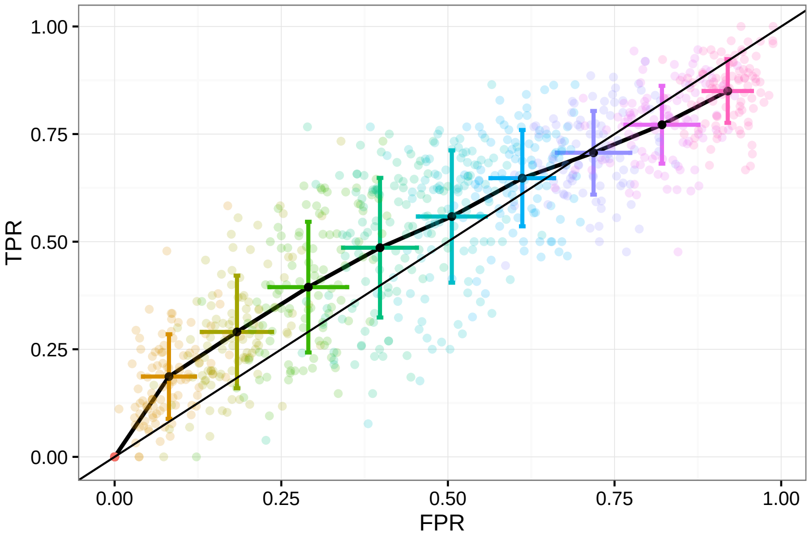


Figure 3.8: Threshold-averaged ROC curve for the cross validation of the LDA regression model with a tf-idf filtered vocabulary on the Gumtree dataset. The straight black line here represents the 'baseline' case, *i.e.*, the expected value of the ROC given random assignments of probabilities to each document. The ROC has been evaluated 100 times at 10 threshold levels, represented by the different colours in the graph. Each threshold level has a confidence interval for the value of both the true positive rate (TPR) and false positive rate (FPR), shown with the coloured bars. Finally, these threshold levels are connected by the black ROC curve. The area under the curve (AUC) is 0.5400.

However, performing this process on the Gumtree corpus removes the vast majority of the words used as predictors in the word count model, such as *kitten* and *cat*. As such, we would expect the performance of an LDA model after tf-idf is applied to be greatly reduced. The results of the cross validation of the LDA model with a vocabulary filtered by tf-idf can be seen in Figure 3.8, showing its poor performance. We therefore apply our topic models to the full 'clean' corpus, without tf-idf filtering.

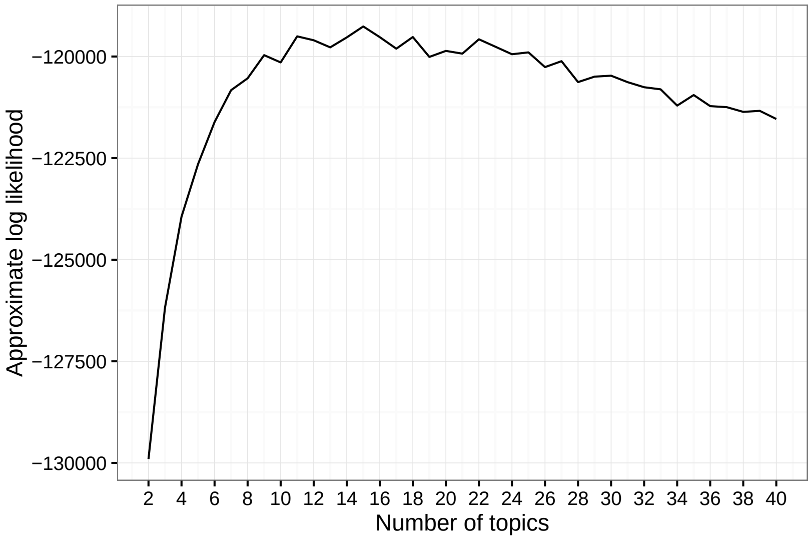


Figure 3.9: Graph showing the approximate log-likelihoods of topic models for varying numbers of topics on the Gumtree dataset.

3.4.2 LDA

Before applying this process to the prediction problem, it is worth examining the topics LDA considers to be the most indicative of this corpus. While in the predictive model, LDA still determines topics irrespective of responses, the best *number* of topics will be determined by the regression model as a whole. In contrast, we look here at the best topics whose number has been determined by the harmonic mean of samples generated by Gibbs sampling as an approximation of the log-likelihood of the model, as discussed and implemented in Graham [22] and Ponweiser [41]. First used in Griffiths and Steyvers [23], this method is preferred for its computational efficiency in finding an estimate of the likelihood of the corpus given each number of topics. Figure 3.9 shows the harmonic mean of a number of topic models with different numbers of topics. From this, we determine that 15 topics are the most appropriate for this corpus, according to LDA.

The topics can be seen in Appendix A.2, where each topic is represented as a list of the most probable words for each topic. This is clearly a truncated version of the topics, as a topic is really a probability distribution over the vocabulary.

3.4.3 Predictive LDA

We now use LDA as a preprocessing step in logistic regression to predict relinquished status. In this case, and in all topic regression models, the variables are the proportions of topics in each document, *i.e.* $\boldsymbol{\theta}$. However, a few considerations must be made in developing this model.

Number of topics

Unlike when applying LDA as a descriptive method to the corpus, the log likelihood measure is not necessarily the most sensible to determine the best model as it does not account for a response. Instead, we consider each possible number of topics in the context of the regression model; that is, compare regression models with different topic numbers k using prediction error from K-fold cross validation.

K-fold cross validation [21], as outlined in Section 2.9.1, is a process in which a dataset, in this case the Gumtree corpus, is partitioned into K distinct 'folds'. For each fold i:

1. the documents in the ith fold become a 'test' set,
2. the remaining $K - 1$ folds become a 'training' set of documents, and
3. the responses for all documents in i are predicted from the model rebuilt on the training set.

The mean square error (MSE) for each fold i is calculated as

$$\mathrm{MSE}_i = \sum_{j \in C_i} \frac{1}{m_i} \left(y_j - \hat{y}_j\right)^2,$$

where m_i is the number of documents in the ith fold, $\hat{y}_j$ is the model estimate of the response y_j and C_i is the set of documents in the ith fold. The

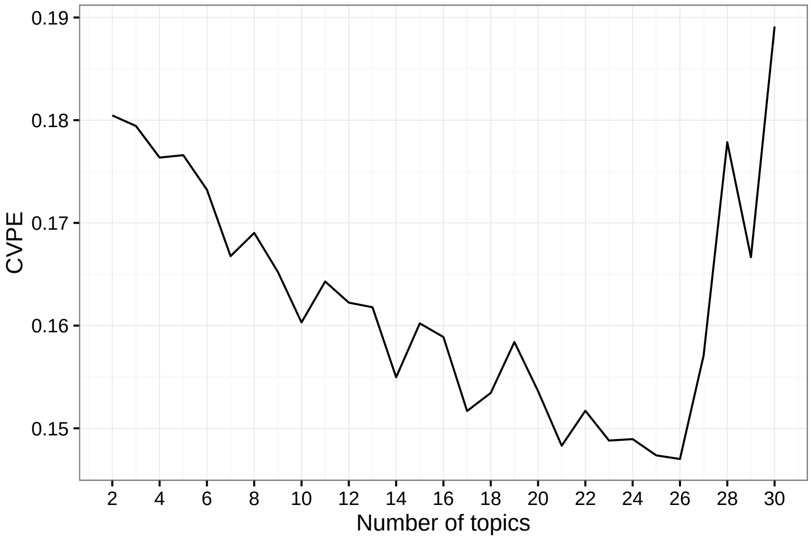


Figure 3.10: CVPE for various LDA regression models of the Gumtree dataset with differing numbers of topics. The lower the CVPE, the better the model is expected to perform.

prediction error is then calculated as

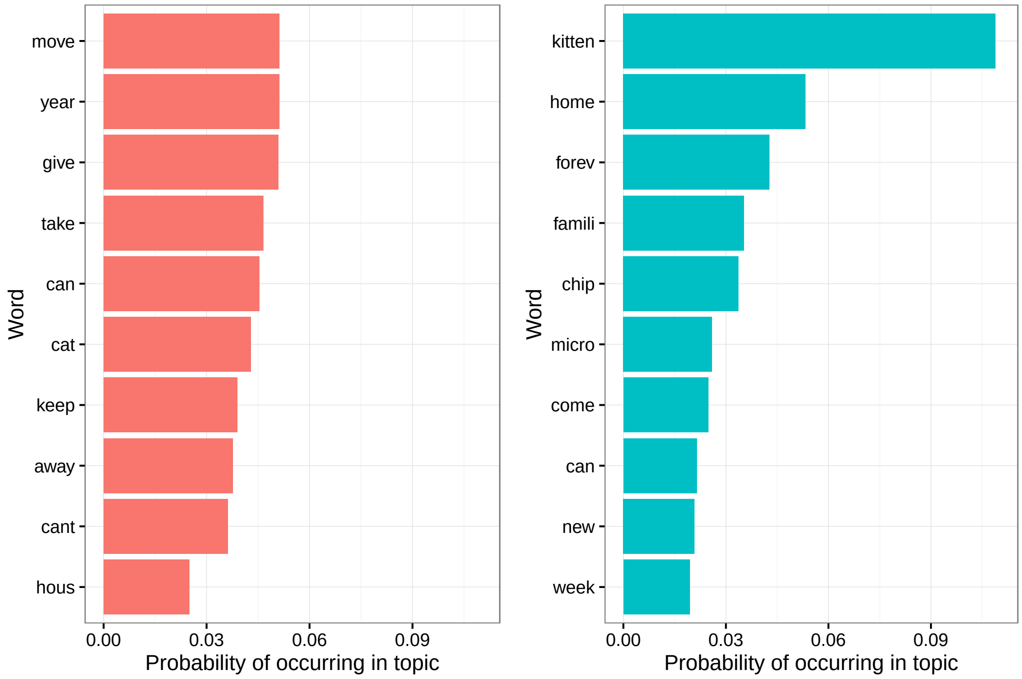
$$\mathrm{CVPE}_K = \sum_{i=1}^{K} \frac{m_i}{m} \mathrm{MSE}_i,$$

where m is the number of documents in the corpus. We therefore choose the model with the lowest CVPE.

The CVPE of regression models per number of topics can be seen in Figure 3.10. From this, we choose the model with 26 topics. The topics and coefficients for this regression model can be seen in Appendix A.3.

LDA predictive model

Figure 3.11 shows two of the topics in the LDA model: that corresponding to the most positive and the most negative coefficients (topics are displayed as in *Text mining in R* [49]). Like the word count model, words such as *kitten* and *cat* appear prominently in their respective topics, as would be expected, as well as other indications of age (in particular, *week* and *year*).



(a) Topic 7.

(b) Topic 26.

Figure 3.11: Top ten most likely words in topics corresponding to the most positive (Topic 7) and negative (Topic 26) coefficients in the LDA regression model of the Gumtree corpus.

The words which have the greatest difference between these two topics are displayed in Figure 3.12. Interestingly, the word *sad* is highly skewed to the relinquished side, indicating more emotive language in advertisements for relinquished cats. The words *micro* and *chip* are prominently correlated with non-relinquished advertisements, showing that details such as microchipping are much more likely to be discussed in those advertisements.

Introducing new documents

When it comes to prediction, we generally have some corpus over which we develop our model, and use this to predict the response of new documents that are not in the original corpus. Because our model requires us to know θ_j,

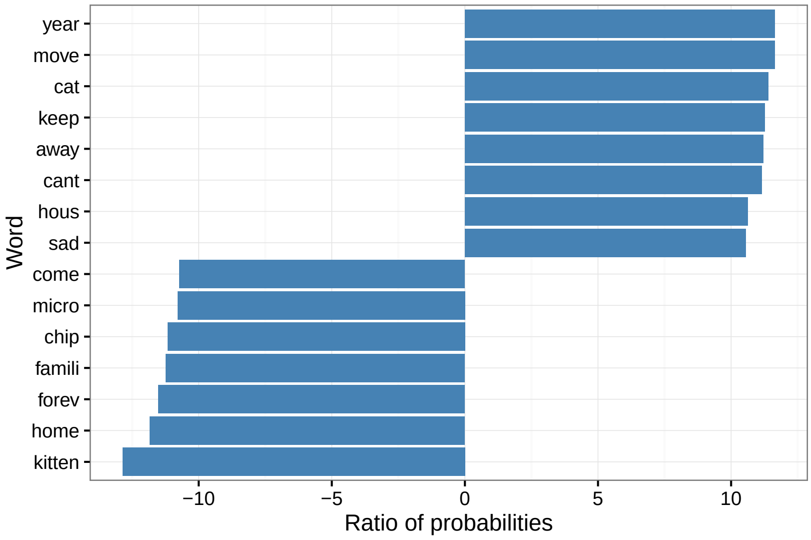


Figure 3.12: Words with the largest differences between topics 7 and 26 for the LDA regression model of the Gumtree corpus.

the proportion of the document, $\mathbf{w}_j$, made up of each topic in order to make a prediction, we have two options. Either the topic model can be retrained with the new document included, and the regression model retrained with the new topics on the old and new documents; or the θ_j of the new document can be found based on the current topic model. For efficiency's sake, the second option is preferable.

One way of approaching this problem is to treat it as a case of maximum likelihood estimation, where the estimate of θ_j, $\hat{\theta}_j$, that maximises the likelihood of that document occurring is found. That is, we have

$$\begin{aligned} L(\theta_j) &= f(\mathbf{w}_j|\theta_j) \\ &= f(w_{j1}, w_{j2}, ..., w_{jn_j}|\theta_j) \end{aligned}$$

where $w_{j1}, w_{j2}, ..., w_{jn_j}$ are the words in document j. Due to the 'bag of words' assumption that LDA makes, the words in the document can be treated as independent, as the appearance of one word has no effect on any other.

Therefore,

$$L(\theta_j) \;=\; \prod_{i=1}^{n_j} f(w_{ji}|\theta_j).$$

From the law of total probability, we can write this as

$$L(\theta_j) \;=\; \prod_{i=1}^{n_j}\sum_{l=1}^{k} f(w_{ji}|t_l,\theta_j)f(t_l|\theta_j),$$

where t_l is the lth topic of the model, $l = 1, 2, ..., k$. However, as the w_{ji} are independent of the θ_j conditional on the topic assignments t_l,

$$L(\theta_j) \;=\; \prod_{i=1}^{n_j}\sum_{l=1}^{k} f(w_{ji}|t_l)f(t_l|\theta_j).$$

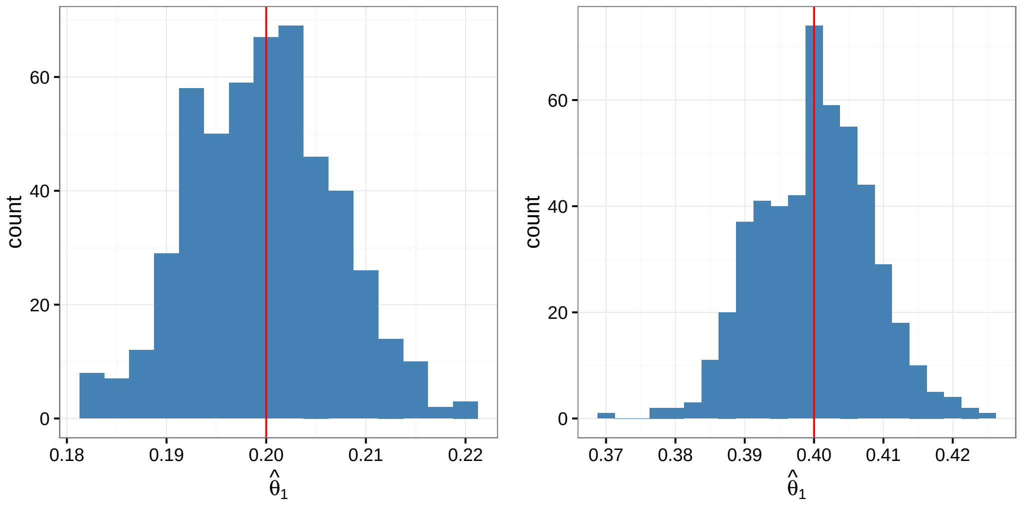
It can be seen that the likelihood is now written as products of the topic proportions and the topics themselves.

$$\begin{aligned} L(\theta_j) &= \prod_{i=1}^{n_j}\sum_{l=1}^{k} \phi_{l,w_{ji}}\theta_{jl} \\ &= \prod_{i=1}^{n_j} \left[\theta_j \boldsymbol{\phi}\right]_{w_{ji}}. \end{aligned}$$

Instead of expressing the document as each individual word, we can instead express it as a series of counts of each word in the vocabulary (due to ‘bag of words’). That is, $\mathbf{w}_j = \mathbf{n} = \{n_1, n_2, ..., n_v\}$, where n_i is the number of times the ith word of the vocabulary appears in the document for $i = 1, 2, ..., v$. The log likelihood of θ_j can therefore be expressed as

$$l(\theta_j) \;=\; \mathbf{n} \cdot \log\left(\theta_j \boldsymbol{\phi}\right).$$

One important issue to note when it comes to this estimation is the fact that the topics are distributions over the vocabulary of the original corpus; that is, if any word appears in a new document that does not appear in the corpus, then $f(w_{ji}|t_l)$ does not exist. The simplest way to handle this is to assign a probability of 0 to each new word for each topic, which is equivalent to removing them from the document.



(a) $\theta_1 = 0.2$

(b) $\theta_1 = 0.4$

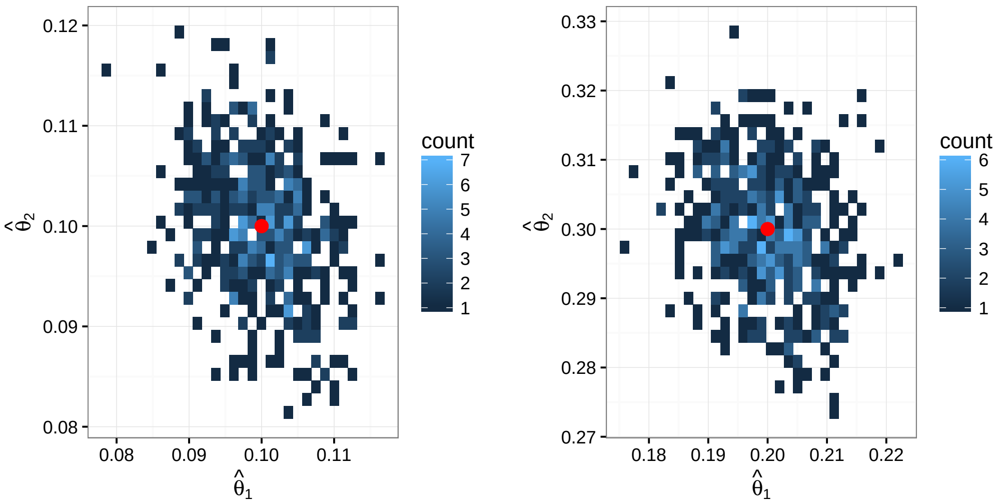
Figure 3.13: Histograms of the maximum likelihood estimates of θ_1 for corpora of two topics, given relative true values of 0.1 and 0.2.

To demonstrate the effectiveness of this, it is necessary to generate documents for which we know the topics and topic proportions.

Suppose there exists a corpus comprising of two topics, with a vocabulary of 500 words. It is possible to randomly generate these topics given these specifications, and from these topics and following the generative process outlined in LDA (see Section 2.3), documents may be generated. Due to the assumptions of LDA, these documents will be in the form of $\mathbf{n}$ above, that is a vector of word counts, as order does not matter. All documents generated have a length of between 5000 and 10,000 words.

Given our newly generated documents, and the knowledge of our $\boldsymbol{\phi}$, or topics, we are able to test the validity of the MLE process outlined above by finding the estimates $\hat{\boldsymbol{\theta}}$ of each new document and comparing them to the known topic proportions $\boldsymbol{\theta}$. In Figure 3.13, this process has been repeated 500 times for various fixed $\boldsymbol{\theta}$, and the estimates displayed in a histogram.

From these figures, there is a definite clustering of values around the true value, and thus it is reasonable to assume that the MLE process for estimating



(a) $\{\theta_1, \theta_2\} = \{0.1, 0.1\}$

(b) $\{\theta_1, \theta_2\} = \{0.2, 0.3\}$

Figure 3.14: Two-dimensional histograms of the maximum likelihood estimates of $\{\theta_1, \theta_2\}$ for corpora of three topics, given relative true values of $\{0.1, 0.1\}$ and $\{0.2, 0.3\}$.

the topic proportions of a new document given previously existing topics is sound. This process also holds for corpora with greater numbers of topics, as evidenced by Figure 3.14.

3.4.4 Model cross validation

Figure 3.15 shows the ROC curve generated for the cross validation of the LDA regression model on the Gumtree data, using the same process as outlined for the word count model cross validation. As can be seen, this model also performs reasonably well, in particular compared to random allocation. The AUC for this model is 0.8913, indicating it performs less well than the word count model with an AUC of 0.9264.

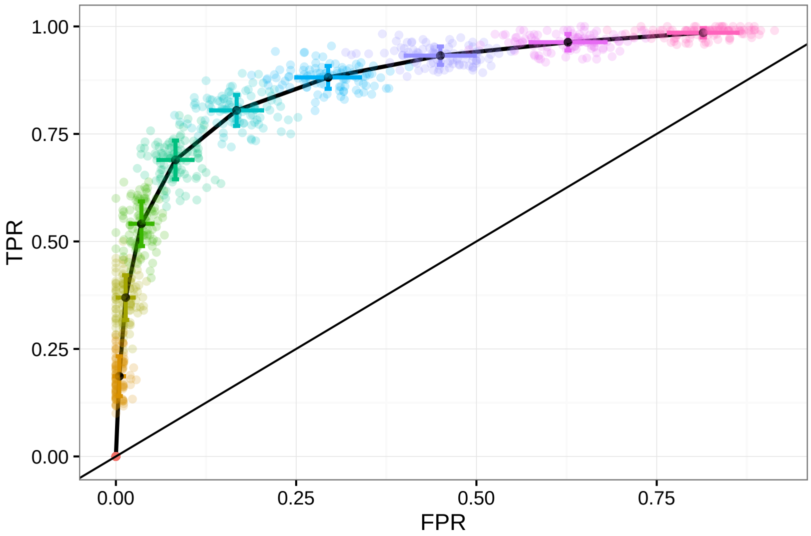


Figure 3.15: Threshold-averaged ROC curve for the cross validation of the LDA regression model on the Gumtree dataset. The straight black line here represents the 'baseline' case, *i.e.*, the expected value of the ROC given random assignments of probabilities to each document. The ROC has been evaluated 100 times at 10 threshold levels, represented by the different colours in the graph. Each threshold level has a confidence interval for the value of both the true positive rate (TPR) and false positive rate (FPR), shown with the coloured bars. Finally, these threshold levels are connected by the black ROC curve. The area under the curve (AUC) is 0.8913.

3.5 LDA step-up model

Given the unsupervised process that found the topics in the above model, it stands to reason that not all, if any, topics would be good predictors of relinquished status. It may be prudent to remove some of the predictors from the model, to avoid overfitting the data and thus negatively influencing prediction by fitting a model too closely to our current corpus. Therefore, a step-up model using the topics from the LDA model may improve results.

A step-up procedure was performed as outlined in Section 3.3.1 on the topic distributions found in the LDA model of Section 3.4.3. 20 topics were chosen from the 26 in the original LDA regression model. This model can be seen in Appendix A.4. Cross validation was then performed as with previous models, and the results can be seen in the ROC curve in Figure 3.16.

We can compare these results to the original LDA model, in Figure 3.17. From this, we can see that the two curves are practically indistinguishable, and thus removing less relevant topics from the model makes very little difference in its predictive capabilities. In fact, the 95% confidence intervals for the AUC for the full LDA and step-up models overlap each other, indicating no significant difference between the two models' predictive abilities.

However, it is worth noting, as seen in Appendices A.3 and A.4, nearly all topics in the step-up model are significant, whereas none of the topics are significant in the full LDA regression model. Therefore, it may be prudent to consider a step-up process for other problems or corpora.

3.6 Supervised LDA (sLDA)

Supervised LDA (or sLDA) [12], as outlined in Section 2.4, is a suitable method to apply to the Gumtree problem as it performs essentially the same process as LDA above, but generates topics in reference to the response. Like with LDA, inference for this model is computationally infeasible when done analytically and an approximation method must be used. In this chapter, we use a variational expectation-maximisation algorithm as mentioned in

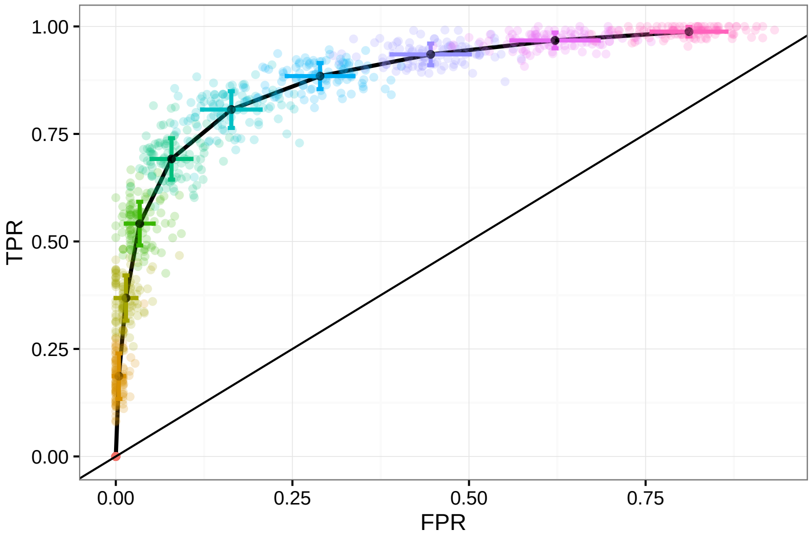


Figure 3.16: Threshold-averaged ROC curve for the cross validation of the LDA step-up regression model on the Gumtree dataset. The straight black line here represents the 'baseline' case, *i.e.*, the expected value of the ROC given random assignments of probabilities to each document. The ROC has been evaluated 100 times at 10 threshold levels, represented by the different colours in the graph. Each threshold level has a confidence interval for the value of both the true positive rate (TPR) and false positive rate (FPR), shown with the coloured bars. Finally, these threshold levels are connected by the black ROC curve. The area under the curve (AUC) is 0.8946.

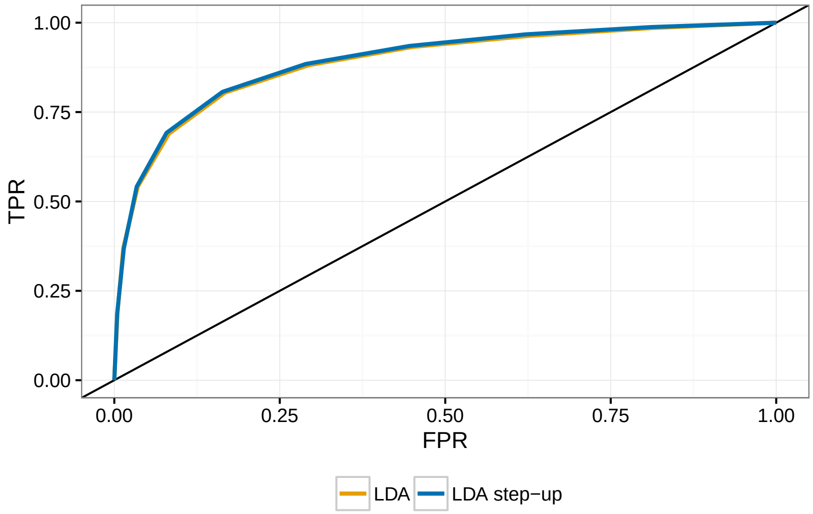


Figure 3.17: Comparison of the threshold-averaged ROC curves for the cross validation of the LDA regression model and the LDA step-up regression model on the Gumtree dataset.

Section 2.4.1 to find the topics given our corpus of Gumtree advertisements.

While sLDA has obvious applications to this problem, it is by no means the only supervised topic model. However, for the most part these models are designed for multi-labelled documents and therefore not optimal for the Gumtree problem. For instance, labelled LDA (L-LDA) [45] is a supervised topic model analogous to sLDA, but for multi-labelled documents. To see a more comprehensive list of supervised models, see Section 2.2.

3.6.1 Number of topics

As with the LDA model, to determine the best number of topics for the sLDA model we apply K-fold cross validation in order to determine the number k that minimises the CVPE of the model. Theoretically, this method should be more successful in finding the best possible number of topics than when compared to LDA, due to the supervised nature of the process. That is, while LDA found the best number of topics given a *possible* $\boldsymbol{\phi}$ for each number k, sLDA will find the best number of topics for the *best* $\boldsymbol{\phi}$ for each k given our response variable.

As evidenced in Figure 3.18, we choose two topics for our model. Given the binary response variable y_j, this seems logical. The topics and coefficients for this regression model can be seen in Appendix A.5.

3.6.2 sLDA model

The sLDA regression model with two topics has coefficients summarised in Table 3.1.

	Values	**Std. Err.**
Topic 1	3.401808	0.1062816
Topic 2	-2.538107	0.0814109

Table 3.1: Coefficients of the sLDA regression model with two topics on the Gumtree corpus.

This model implies that a large presence of Topic 1 in a document is indicative of a relinquished animal, whereas a small presence is indicative of a non-relinquished animal. Figure 3.19 shows the top words for the two topics. These topics have much in common with the topics shown by the LDA model, and the words in the word count model. They indicate once again that the age of the cat, the emotive language used and desexed status are all predictors of the relinquished status of the cats.

Figure 3.20 shows the ROC curve for the sLDA model, once again generated using the same method as outlined with the word count model. Like the previous models, this model appears to be performing quite well, with an AUC of 0.8588.

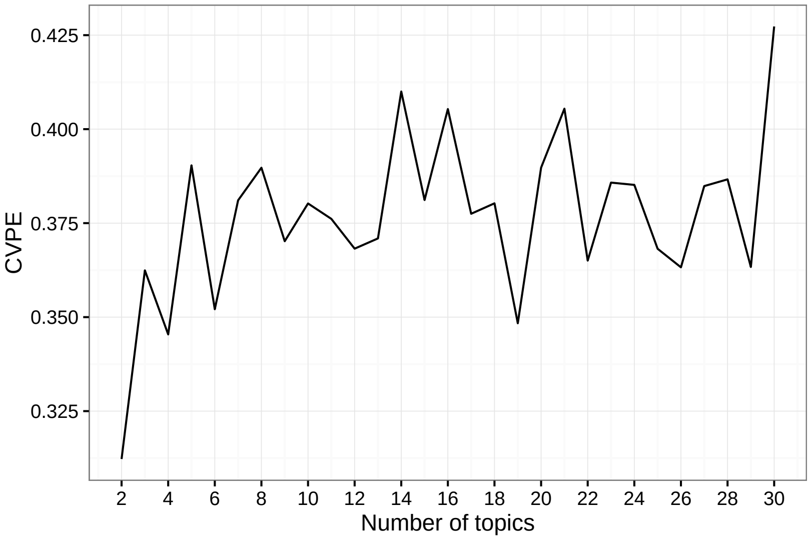
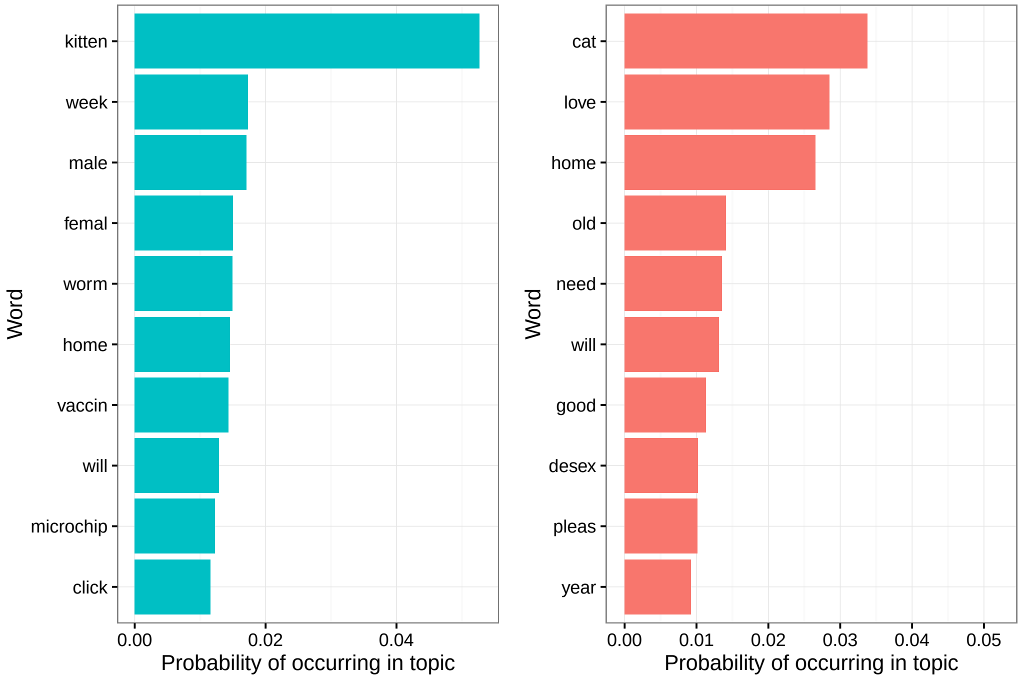


Figure 3.18: CVPE for various sLDA models of the Gumtree dataset with differing numbers of topics. The lower the CVPE, the better the model is expected to perform.

3.7 Conclusion/summary

Figure 3.21 shows the threshold-averaged ROC curves generated for the three models in this chapter. Interestingly, the sLDA model performs less well than LDA on this problem. As the sLDA model was generated with the response in mind, this is unexpected. This may be partly attributed to the simplicity of the corpus. Certain key words such as *cat*, *kitten* and *home* are almost definite indicators of relinquished status, as outlined in the earlier analysis. However, the sLDA model also only incorporates two topics, whereas the LDA model has 26. To make this a fair comparison, we see how sLDA performs with 26 topics. The sLDA model with 26 topics is summarised in Appendix A.6.

As we can see in Figure 3.22, comparing an LDA and sLDA model with the same number of topics yields a slightly better result for sLDA (specifically, AUCs of 0.8913 and 0.9030 respectively with non-overlapping 95% confidence intervals), meaning the loss in performance of the two-topic sLDA model can



(a) Topic 1.

(b) Topic 2.

Figure 3.19: Graphs of the ten most likely words in the two topics of the sLDA regression model of the Gumtree dataset.

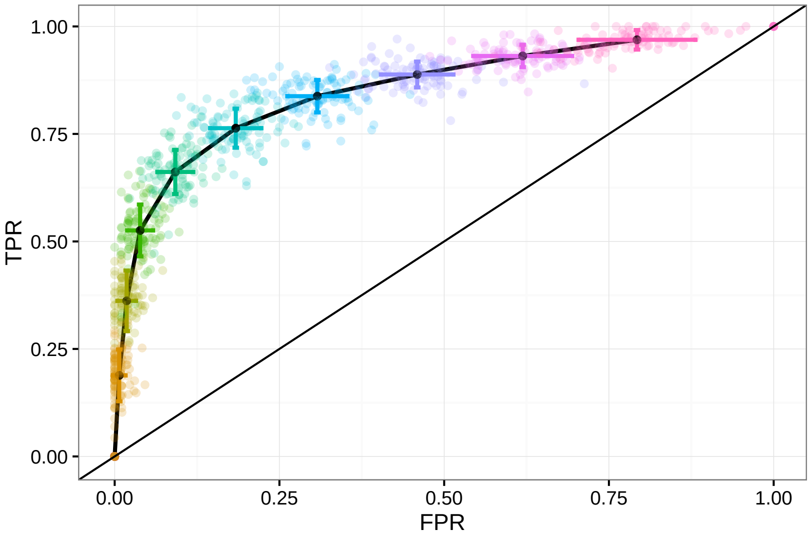


Figure 3.20: Threshold-averaged ROC curve for the cross validation of the sLDA model with two topics on the Gumtree dataset. The straight black line here represents the 'baseline' case, *i.e.*, the expected value of the ROC given random assignments of probabilities to each document. The ROC has been evaluated 100 times at 10 threshold levels, represented by the different colours in the graph. Each threshold level has a confidence interval for the value of both the true positive rate (TPR) and false positive rate (FPR), shown with the coloured bars. Finally, these threshold levels are connected by the black ROC curve. The area under the curve (AUC) is 0.8588.

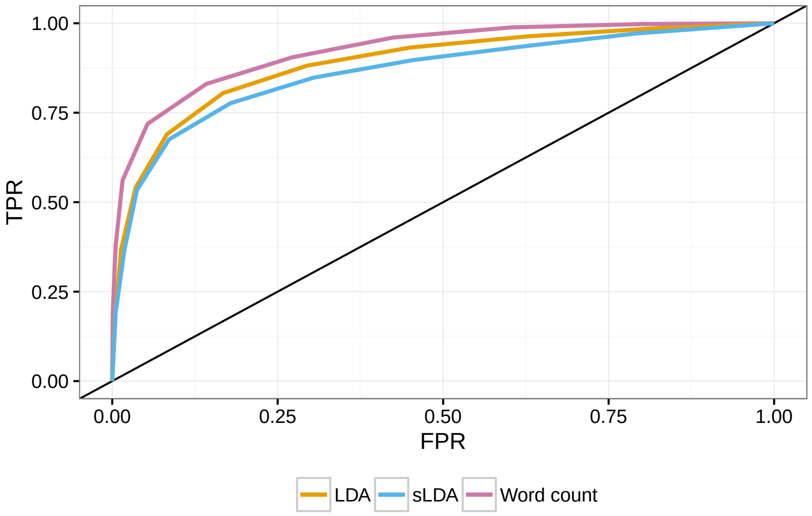


Figure 3.21: Comparison of the threshold-averaged ROC curve for the word count, LDA and sLDA models of the Gumtree dataset. The straight black line here represents the 'baseline' case, *i.e.*, the expected value of the ROC given random assignments of probabilities to each document. Each coloured curve corresponds to the average value of the true positive rate (TPR) versus the false positive rate (FPR) at different threshold levels for one of the models. These curves can be seen in more detail in their models' relative sections.

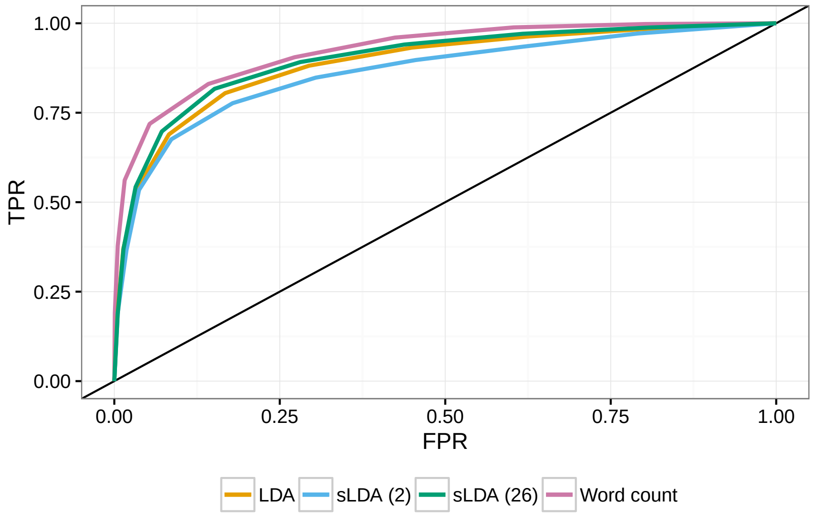


Figure 3.22: Comparison of the threshold-averaged ROC curve for the word count, LDA and two sLDA models of the Gumtree dataset. The straight black line here represents the 'baseline' case, *i.e.*, the expected value of the ROC given random assignments of probabilities to each document. Each coloured curve corresponds to the average value of the true positive rate (TPR) versus the false positive rate (FPR) at different threshold levels for one of the models. These curves can be seen in more detail in their models' relative sections.

be attributed to the number of topics. This indicates that the topics found blindly by the LDA model are nearly as good as those found by the sLDA model. This could be the result of the simplicity of language used in online advertising and the fact that the response variable is binary.

The AUC of each model in this chapter can be seen in Table 3.2.

Model	**AUC**	**95% confidence interval for AUC**
Word count	0.9264	$(0.9234, 0.9294)$
LDA	0.8913	$(0.8871, 0.8955)$
LDA step up	0.8946	$(0.8901, 0.8991)$
sLDA (2 topics)	0.8588	$(0.8534, 0.8642)$
sLDA (26 topics)	0.9030	$(0.8988, 0.9073)$

Table 3.2: Table of area under the curve (AUC) for the models used in this chapter on the Gumtree dataset, with their 95% confidence intervals.

Neither of the topic models perform as well as the step-up word count model. However, they do not perform significantly worse; future work could investigate how these results change on a more linguistically complex corpus.

Efficiency is another important factor to consider. While this is a relatively small corpus, efficiency becomes more of an issue for larger datasets. Table 3.3 shows the times each of the respective models took to compute on a 2015 iMac, with a 2.8 GHz Intel core i5 processor and 8GB RAM.

As Table 3.3 shows, the methods incorporating topic models significantly cut down on computational expense. While they lose a small amount of accuracy compared to the step-up word count model this is a great advantage, especially for larger corpora. Between the topic models, sLDA is also significantly more efficient than LDA.

Part of the success of the models in this chapter may be attributed to the simplicity of the dataset used. Firstly, the ‘bag of words’ assumption made by all the models here, as well as most topic models in existence, is

Model	**Time taken (seconds)**	**Proportion of word count model**
Word count	1885.939	1
LDA	61.546	0.0326
LDA step up	69.749	0.0370
sLDA (2 topics)	2.136	0.00113
sLDA (26 topics)	11.978	0.00635

Table 3.3: Table of computational efficiencies for the models used in this chapter on the Gumtree dataset.

sufficient for the most part when predicting relinquished status from advertisements. Very little useful information is conveyed through grammatical or document structure and therefore little is lost by discarding it. Secondly, the two statuses we are trying to separate in this problem have a lot of distinct vocabulary: the word *kitten*, for example, rarely appears in an advertisement for a relinquished animal.

For these reasons, it is well worth investigating this methodology on a more linguistically complex corpus with longer documents to see how the traditional models fare.

Chapter 4

Topic modelling regression and document structure

4.1 Introduction

In the previous chapter, we explored supervised learning for prediction using topic models. Now, we shift our focus to the 'bag of words' assumption, and whether dropping that assumption improves our models. There are several ways of incorporating language structure into regression models, as discussed in Section 2.2.2. For instance, topic models have been developed that incorporate syntactical or grammatical structure into the model [14], *e.g.* part-of-speech tagging like in the model developed by Griffiths, Steyvers, Blei and Tenenbaum [24]. A more straightforward way to drop the 'bag of words' assumption and incorporate structure is to assume there is dependence between the words in a document; *i.e.*, word order matters.

In fact, topic modelling lends itself naturally to a Markov process, and several models have incorporated this structure already [4, 25]. That is, we can consider a Hidden Markov Model (HMM) (outlined in Section 2.5) where the latent states are topic assignments, and observations are the words in our documents. A document is therefore a sequence of observations, from which we can infer the underlying model of transition probabilities between topics, and emission probabilities (which in this case are the topics themselves). We therefore replace the 'bag of words' assumption by assuming a dependence for each word on the word before it, although the two words are independent conditional on their topic assignments.

This chapter explores the predictive capability of topics generated from a hidden Markov model-like structure in a regression model, and develops a model for persistent topics in documents. In order to do so, we need an appropriate corpus on which to test our models. The idea behind incorporating Markov structure into a topic model is that oftentimes the topic assignment to a word is influenced heavily by the topic of the word before it, *i.e.*, it is less likely to switch topics in the middle of a sentence or paragraph. A 'bag of words' model has no way of enforcing what part of a document belongs to which topic and thus may not group words in the most meaningful way. For some corpora, this is an acceptable loss of information, for example in the situation presented by Blei [9] where scientific papers are classified. In such situations, topics often shift multiple times in one sentence and there is less dependence between adjacent words. However, as discussed in 2.3.3 with the *Anne of Green Gables* [37] example, we often want corpora such as books and movies, that tell a story, to have fewer topic shifts overall. This is due to the overarching structure of the document playing a large part in its nature, as opposed to scientific articles which tend to follow a similar structure; therefore more 'persistent' topics may give better information for scientific prediction. Work performed by the Stanford Literary Lab into hierarchical word clustering [35] shows words with similar 'themes' appearing in the same location within a narrative, which supports the idea that knowledge of document structure could improve topics.

For this reason, we have chosen to analyse the dialogue of the 2003 movie *Love Actually*[1] [18]. *Love Actually* is a Christmas movie known for its interwoven yet still quite distinct storylines, each exploring a different aspect of love. We therefore ask if we are able to predict to which storyline a scene in

[1] `http://www.imdb.com/title/tt0314331/`

the movie belongs, based on the dialogue in that scene. The interconnections between these storylines have been analysed in more depth from a network perspective by Hickey and Wezereck [28].

4.2 Preliminary analysis of data

The movie *Love Actually* consists of 79 scenes, each pertaining to one of 10 storylines, classified here by the characters involved:

- The airport scenes (bookends of the film, only the first and last),
- Billy Mack (Christmas single),
- Jamie and Aurelia (Portuguese romance),
- Daniel and Sam (child in love),
- Colin Frissell (America),
- Jack and Judy (film stand-ins),
- Peter, Juliet and Mark (cue cards),
- The Prime Minister and Natalie (Prime Minister),
- Sarah and Karl (office romance), and
- Harry, Karen and Mia (adultery).

The scenes in this film were hand-classified by storyline, and their dialogue forms the documents for our corpus. The classifications of each scene can be seen in Appendix B.1. The number of scenes pertaining to each storyline can be seen in Figure 4.1. From this, the most common storylines are those of the Prime Minister (13 scenes), the child in love (13 scenes) and the Portuguese romance (12 scenes).

The corpus contains 10, 140 words, with 2, 914 unique words, before any data cleaning. The size of the scenes ranges from 5 to 392 words, with a

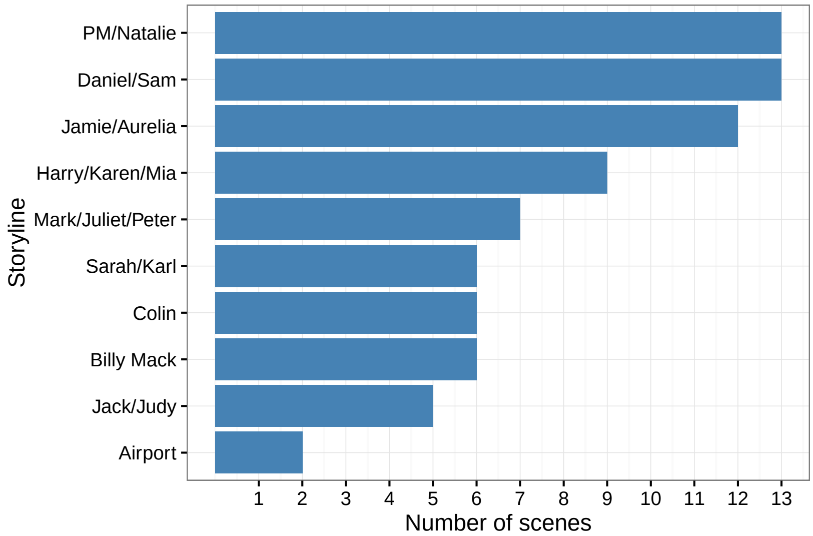


Figure 4.1: Bar graph of the number of scenes pertaining to each of the 10 storylines in the *Love Actually* corpus.

mean of 128.4 words per scene. The distribution of these document lengths is seen in Figure 4.2. As with the Gumtree data, we clean our corpus to group words more efficiently. The following steps were taken in the cleaning process:

- removal of punctuation and numbers,
- conversion to lower case, and
- removal of stop words (as outlined in Section 2.7).

As our goal is to predict storylines from dialogue, we also remove character names as they are for the most part an immediate indication of the storyline and thus allow a model to trivially predict it. This preprocessing leaves a corpus with $1,607$ unique words. While in Chapter 3 we also stemmed the words in the vocabulary, we have chosen not to perform this here; because we are investigating the effect of document structure on topics, grammatical information that would have been stripped due to the stemming process may be pertinent. We also have the advantage here of a small corpus relative to

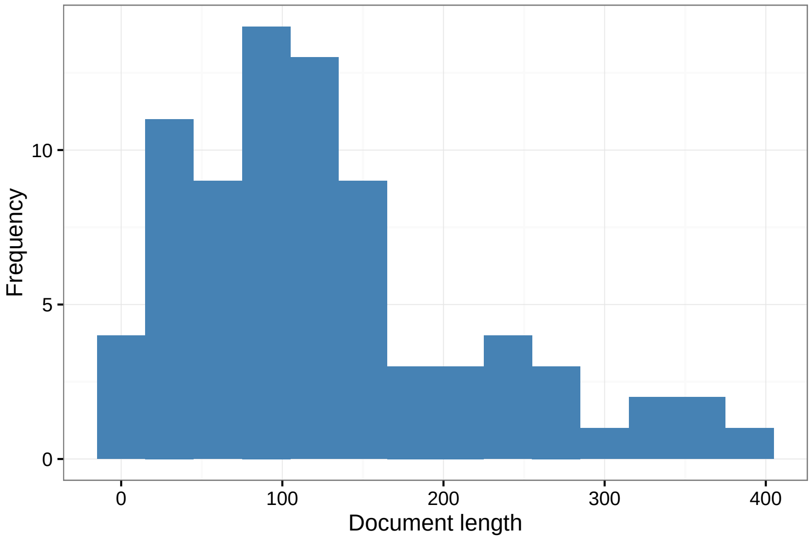


Figure 4.2: Histogram of document lengths for the *Love Actually* corpus.

most topic modelling applications, and therefore do not need to reduce our vocabulary further for computational reasons.

The most common words in the cleaned corpus are shown in Figure 4.3. Interestingly, the most frequent word in the dialogue of *Love Actually* is *just*, followed by *love*, which is expected given the subject matter.

Figures 4.4 and 4.5 show word clouds for each of the ten storylines' most frequent words. From these, we see noticeable differences in the vocabulary between stories. For instance, the word *jewellery* features heavily in the adultery storyline. Similarly, we see words such as *president* and *minister* in the Prime Minister storyline. It should be noted that the word clouds in Figures 4.4 and 4.5 purely serve as a visual aid to provide familiarity with the storylines and help with topic interpretation later in the corpus.

4.3 Word count model

As in Chapter 3, before using topic proportions as predictors it is worth developing a 'gold standard' model, or a model whose predictive capability we aim for with the other models in the chapter. Because the problem we are

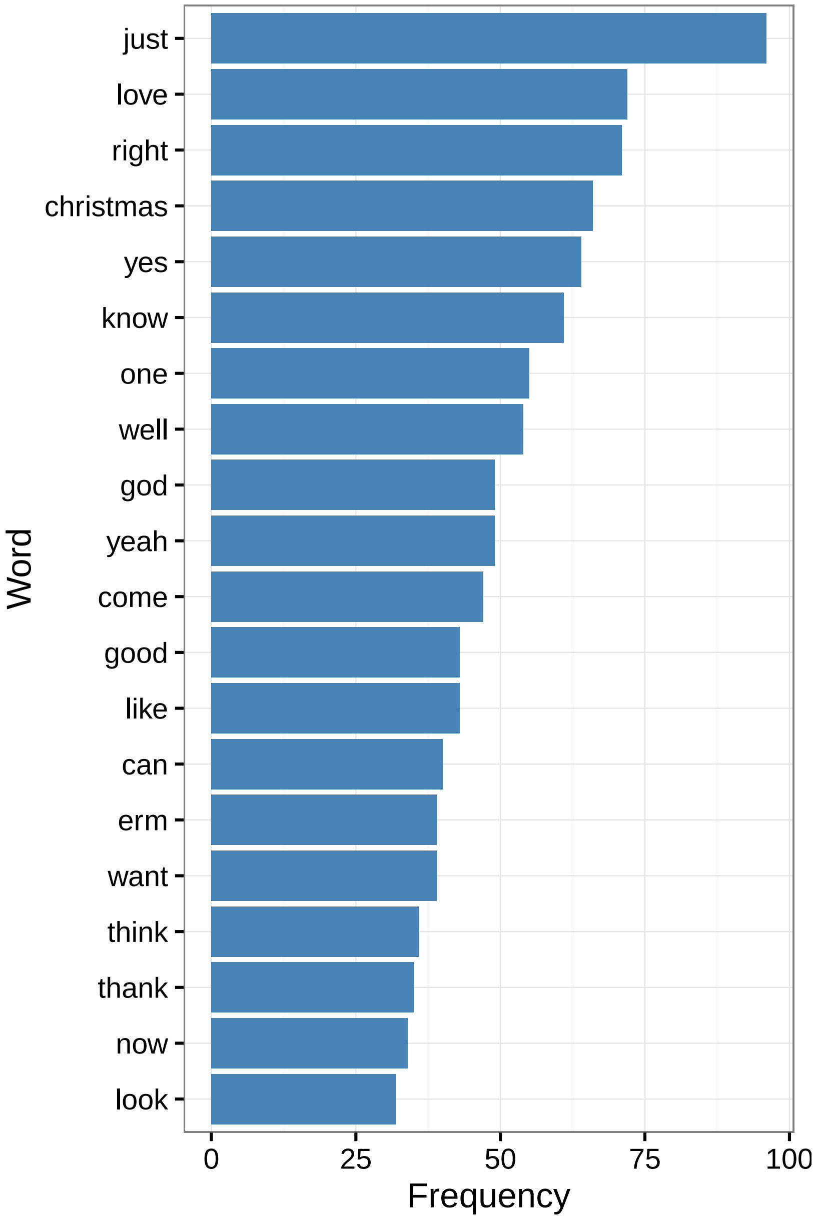
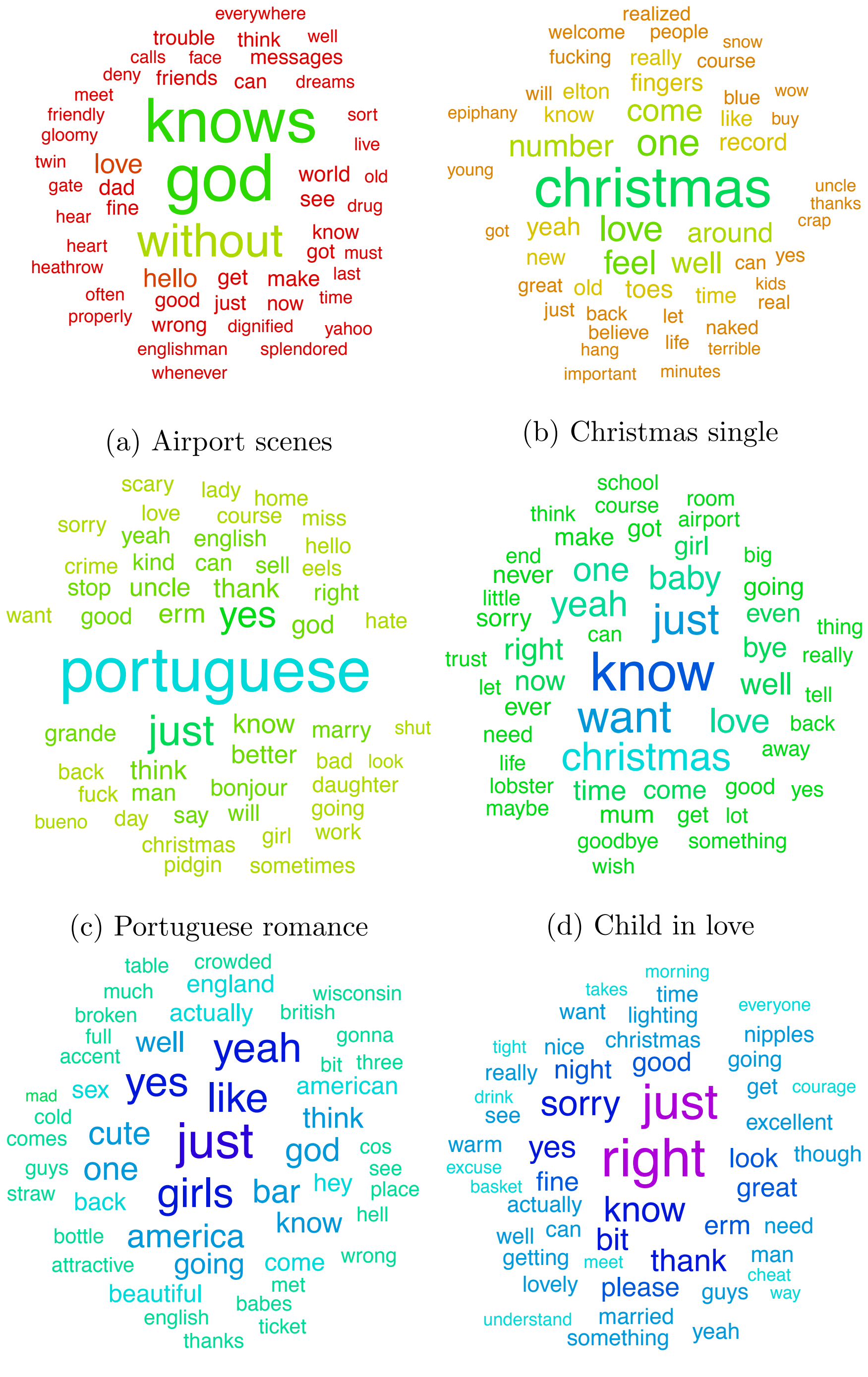


Figure 4.3: Bar graph of the 20 most common words in the cleaned *Love Actually* corpus, with the frequencies of their appearances.



(a) Airport scenes

(b) Christmas single

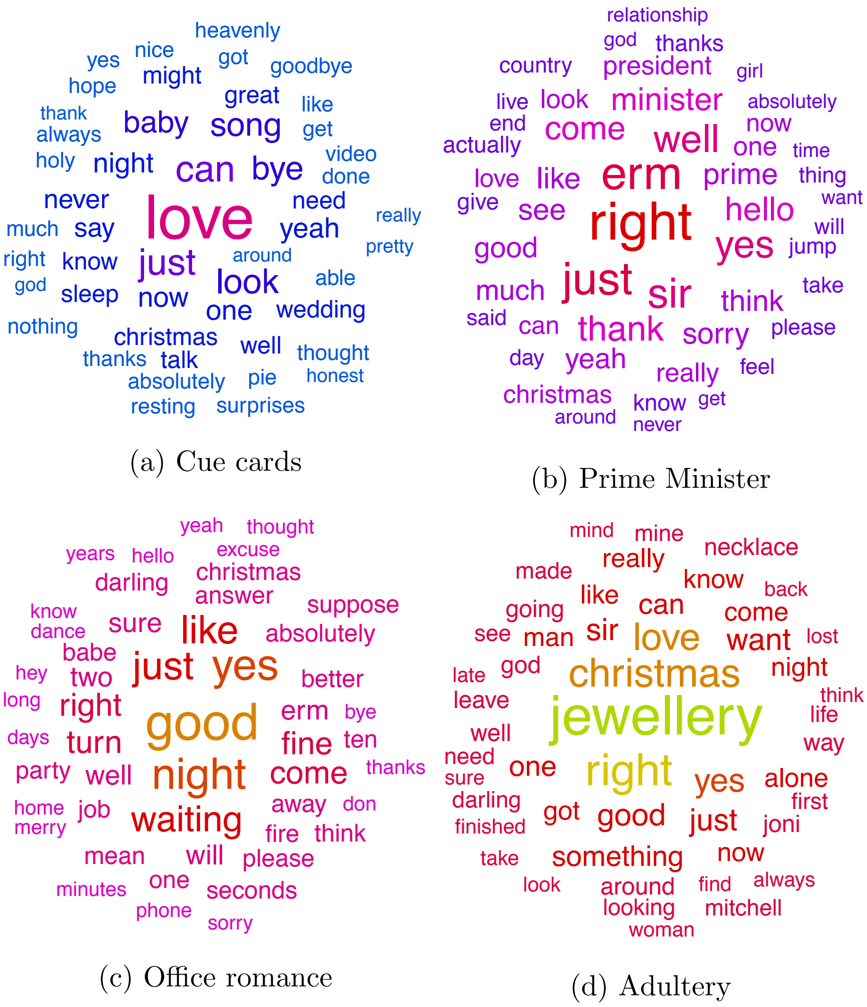
(c) Portuguese romance

(d) Child in love

(e) America

(f) Film stand-ins

Figure 4.4: Word clouds for six of the storylines in *Love Actually*.



(a) Cue cards

(b) Prime Minister

(c) Office romance

(d) Adultery

Figure 4.5: Word clouds for the remaining four storylines in *Love Actually*.

covering in this chapter involves a reasonably small corpus, we are able to compare our topic regression models to a predictive model using individual words as predictors. That is, the predictor is the number of times a word appears in a document. If we were dealing with a much larger corpus, as we often are in NLP settings [9], this kind of prediction would be cumbersome to compute (hence our reliance on topic models and other dimension reduction techniques).

All models in this chapter will use the clean corpus, with 1607 unique words. Rather than using all words in the corpus as predictors (as this would drastically overfit the model), we use a step-up algorithm based on the Akaike information criterion (AIC) [2] to choose the most significant words for the model, without overfitting. The process for this is the same as outlined in Chapter 3.

Because we are looking at predicting a categorical, non-binary variable with 10 levels, we use a multinomial logistic regression model. Further information on these models can be found in Section 2.8.

After applying the step-up process to the 1607 possible predictors, the model in Table 4.1 was found. In this case, the model produced has three predictors: *minister*, *night* and *around*. The first word, *minister*, is an indicator of the storyline involving the Prime Minister (*i.e.*, it has a positive coefficient for that storyline), which is intuitive, but also for the adultery storyline involving the Prime Minister's sister, Karen. However, the presence of the word *minister* in a scene does also make the model more likely to predict the Christmas single storyline, which is less intuitive as the characters never interact. However, this can be explained by the single mention of the Prime Minister in a Billy Mack scene, by another character.

4.3.1 Model validation

In order to cross validate, we require some measure by which we can compare the different methods used in this chapter. For that purpose, we use the Brier score [15, 56], which is outlined in Section 2.9.4. The Brier score calculates

	(Intercept)	*minister*	*night*	*around*
Christmas single	0.22	20.90	-16.66	0.88
Portuguese romance	2.31	-17.79	-14.53	-27.90
Child in love	2.39	-18.26	-15.02	-27.93
America	1.47	-13.39	13.07	-25.56
Film stand-ins	0.69	21.09	14.12	-24.04
Cue cards	0.87	-11.68	14.30	-1.01
Prime Minister	0.61	23.10	13.06	-0.37
Office romance	0.61	-8.31	14.61	-23.49
Adultery	0.73	21.36	14.13	-0.34

Table 4.1: Coefficients of the multinomial logistic regression model found using a step-up algorithm on word counts as predictors for the *Love Actually* dataset predicting storylines. Each row corresponds to a particular storyline in the movie.

the performance of a predictive model as follows:

$$\mathrm{BS} = \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{s}\left(\hat{y}_{ji} - o_{ji}\right)^2 ,$$

where $\hat{y}_{ji}$ is the probability according to the model of document j belonging to storyline i, and

$$o_{ji} = \begin{cases} 1 & \text{if} \quad \text{document } j \text{ belongs to storyline } i \\ 0 & \text{if} \quad \text{document } j \text{ does not belong to storyline } i \end{cases} \tag{4.1}$$

for document $i = 1, 2, ..., m$ and storyline $j = 1, 2, ...s$. Each term in the sum goes to zero the closer the model gets to perfect prediction, and as such our aim is to minimise the Brier score in choosing a model. The maximum value the Brier score can take is 2. We choose the Brier score because of its simplicity, interpretability and historical relationship to prediction problems.

For each document in the corpus (in this case, for each scene), we find the probabilities of each outcome (*i.e.*, of the scene belonging to each storyline) by using the remaining 78 documents (or training dataset) as the corpus in a multinomial logistic regression model with the same three predictors as found above. That is, we perform leave-one-out cross validation on each document in the corpus. We then predict the outcome based on the words found in the left-out document (or test dataset), and repeat for all 79 scenes. However, due to the short length of some scenes, and the fact that unique words must be thrown out, we restrict the testing to 57 of the 79 scenes: the remaining scenes do not generate a numerically stable approximation for θ_j for all models used in this chapter, in particular the HMTM regression model.

The Brier score calculated using this method for the step-up word count model is 0.8255. We take this to be the standard aimed for by all subsequent models in this chapter.

The probabilities of predicting the correct storyline for each scene in the model validation are shown in Figure 4.6. As there are only three words in the regression model, it is not surprising to see scenes of the same storyline with the exact same probability of being chosen correctly. If none of the three words are present in that storyline, or for most of that storyline, then this is inevitable. However, those storylines with one or more of the three words are predicted with a far wider range of accuracy (for instance, the Prime Minister storyline, with the word *minister*). For the most part, and as we would expect from the words chosen, the scenes with the highest probability of correct prediction belong to the Prime Minister storyline.

4.4 Latent Dirichlet allocation model

We once again use Latent Dirichlet allocation (LDA) as our ‘baseline’ topic model against which to compare the predictive effectiveness of others, due to its widespread use, basic structure, and because it inspired most other models. It uses the ‘bag of words’ assumption, and therefore we are able to compare its performance to those which relax this assumption, as discussed

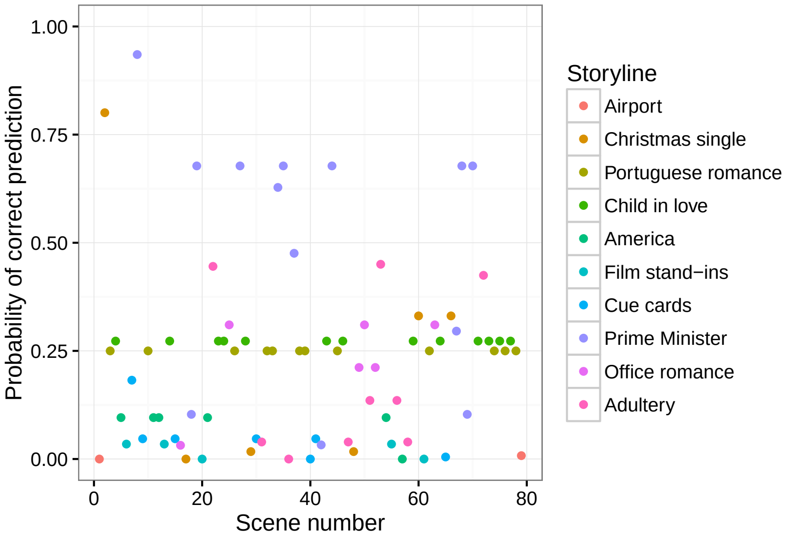


Figure 4.6: Plot of the probability for each scene in the *Love Actually* corpus being allocated to the correct storyline using leave-one-out validation of the word count model. Each scene is coloured based on its true storyline.

earlier.

4.4.1 Number of topics

Most topic models, including LDA, fix the number of topics when performing inference on a corpus. As such, it is necessary to find a way to measure the best number of topics for a particular problem. Whilst in Chapter 3, topics were chosen for the LDA logistic regression model using cross validation prediction error (CVPE), the nature of multinomial regression makes this inappropriate here, due to the categorical and nonbinary response. Therefore, we select our model using the Akaike information criterion (AIC) (discussed in Section 4.3), and choose the model with the lowest AIC to avoid overfitting. Figure 4.7 shows the AIC for each number of topics from 2 to 30. From this, we choose our LDA multinomial logistic regression model to have 16 topics.

The topics from this model are given in Appendix B.2. Given LDA is an unsupervised process and therefore was not necessarily searching for topics

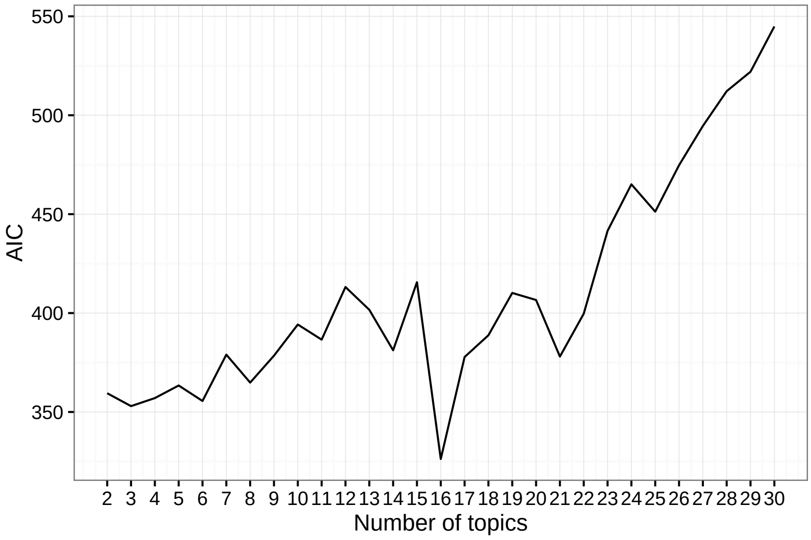


Figure 4.7: Plot of the AIC for different numbers of topics in the LDA multinomial logistic regression mode on the *Love Actually* dataset.

specifically pertaining to the different storylines, it is pleasing to see that a few of the topics in the model relate primarily to one storyline or another. For instance, Topic 6 has top (most frequent) words *christmas*, *number* and *one*, which are all prominent words found in the Christmas single storyline, as demonstrated in Figure 4.4. Looking at the model coefficients in Appendix B.2, the presence of this topic in a document is a stronger indicator of the Christmas single storyline than any other topic. Similarly, Topic 11 has frequent words *america* and *bar*, and according to the model is the strongest indicator of the America storyline, in which the character spends several scenes in a bar.

Unsurprisingly given the unsupervised nature of LDA, the topics tend to all be indicators of multiple storylines. Topic 9, for instance, heavily weights very generic vocabulary (*right*, *yes* and *hello*, for example) and as such is the strongest indicator of three different storylines.

4.4.2 Model validation

As with the word count model, we use the Brier score to evaluate the performance of this model compared to others in the chapter. We again use the leave-one-out cross validation approach to predict the probabilities of a scene belonging to each storyline. Section 3.4.3 outlines the process used to predict the response of new documents for LDA, given an existing corpus, and we employ this here to estimate the topic proportions of a new document given the LDA model found on the remaining 78 documents. Once again, we only consider the prediction of the 57 scenes that produce stable results for all models in the chapter.

This method is in opposition to finding a regression model based on topics found using the entire corpus, but only using the topic proportions of the 'training' data. While the regression model does not rely directly on new documents in this method, the topics used as predictor variables do; thus this method does not truly emulate prediction.

The Brier score found for the LDA regression model is 1.6351. While this is higher and therefore worse than the Brier score for the word count model above, this is not unexpected and we are more interested in seeing how the LDA model fares against other topic models.

Figure 4.8 shows the probability of choosing the correct storyline for each scene from the leave-one-out process on the LDA model. Compared to the word count model, very few of these probabilities are markedly above 0. Those that are, tend to be from the more common storylines (as seen in Figure 4.1), such as those of the Prime Minister and the Portuguese romance.

4.5 Hidden Markov topic model

Hidden Markov Topic Models (HMTMs) [4], like the name suggests, are topic models based on the structure of a Hidden Markov Model [44]. Here, documents function as a sequence of observations (words), with latent, or hidden, states being the topic assignments to each word. Therefore, we

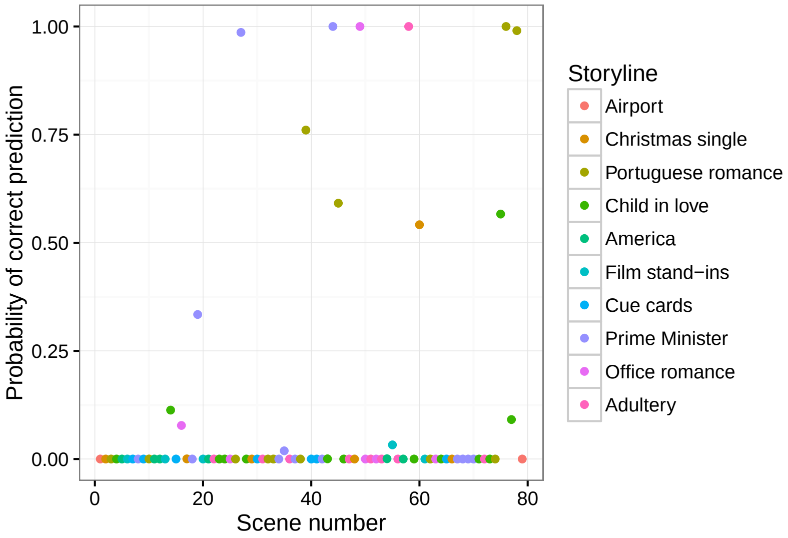


Figure 4.8: Plot of the probability for each scene in the *Love Actually* corpus being allocated to the correct storyline using leave-one-out validation of the LDA regression model. Each scene is coloured based on its true storyline.

drop the 'bag of words' assumption most topic models carry by adopting a dependency between consecutive words in a document. With the Markov property, the probability of transitioning into a new topic at word i depends on the topic assignment at word $i-1$. However, each word assignment is independent from others, conditional on their relative topic assignments.

As outlined in Section 2.6, the generative process of the HMTM is as follows:

1. Generate the k topics $\phi_l \sim \text{Dir}(\beta)$, for $l = 1, 2, ..., k$.
2. For each document $j = 1, 2, ..., m$:
   (a) Generate starting probabilities $\boldsymbol{\pi}_j \sim \text{Dir}(\alpha)$.
   (b) For each topic $l = 1, 2, ..., k$:
      i. Generate the lth row of the transition matrix $\boldsymbol{\Theta}_j$, $\Theta_{jl} \sim \text{Dir}(\gamma_l)$.
   (c) Choose the topic assignment for the first word $z_{j1} \sim \text{Multi}(\boldsymbol{\pi}_j)$.

(d) Select a word from the vocabulary based on the topic assignment for the first word, $w_{j1} \sim \text{Multi}(\phi_{z_{j1}})$.

(e) For each subsequent word in the corpus $i = 2, 3, ..., n_j$:

i. Choose topic assignment z_{ji} based on transition matrix $\mathbf{\Theta}_j$.

ii. Select a word from the vocabulary based on the topic assignment, $w_{ji} \sim \text{Multi}(\phi_{z_{ji}})$.

(f) Create the document $\mathbf{w}_j = \{w_{ji}\}_{i=1,2,...,n_j}$.

Here α, β and $\boldsymbol{\gamma} = \{\gamma_1, ..., \gamma_k\}$ are Dirichlet priors for the starting probabilities, topics and transition matrices respectively.

Like in LDA, the HMTM finds topics for the corpus, analogous to the emission probabilities of an HMM. However, while LDA deals only with topics $\boldsymbol{\phi}$ and topic proportions $\boldsymbol{\theta}_j$, the HMTM now has a $k \times k$ matrix of transition probabilities $\mathbf{\Theta}_j$ for each document j, where k is the number of topics in the corpus. That is, the (i, l)th element of $\mathbf{\Theta}_j$ is the probability of transitioning to topic l on the next word in the document given the state is currently topic i. Rather than using the elements of $\mathbf{\Theta}_j$ as predictors in our multinomial logistic regression model, and in keeping with the method developed for the LDA model, we instead take the equilibrium probabilities of any word belonging to each topic for a document to be our topic proportions $\boldsymbol{\theta}_j$. That is, we find $\boldsymbol{\theta}_j$ such that

$$\boldsymbol{\theta}_j \mathbf{\Theta}_j = \boldsymbol{\theta}_j \quad \text{and} \quad \boldsymbol{\theta}_j \boldsymbol{e} = 1, \quad \text{for} \quad j = 1, 2, ..., m.$$

This also fits with the concept of topic models as a form of dimension reduction, with only $k-1$ predictors used as opposed to $k(k-1)$ if using the entire transition matrix (we only require $k-1$ as opposed to k as the row sums are all equal to 1). As some topic models have been known to fit hundreds of topics [9, 23], this makes models faster to compute.

We again choose the most appropriate number of topics k for the regression model by minimising AIC, as with the LDA model. Figure 4.9 shows the AIC for k between 2 and 30 for the HMTM regression model on the

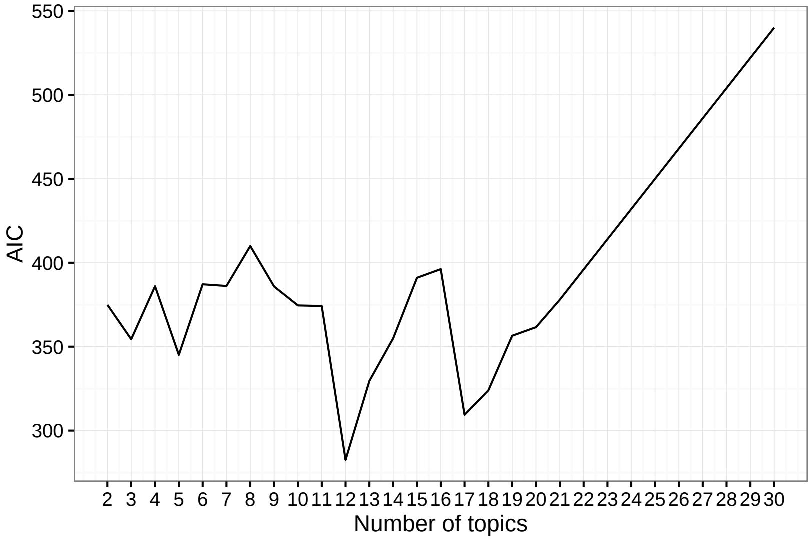


Figure 4.9: Plot of the AIC for different values of k, the number of topics in the HMTM multinomial logistic regression model on the *Love Actually* dataset.

Love Actually corpus to predict storyline. From this, we choose a model with 12 topics. These topics are again found independently of the response variable, as with LDA (that is, we use an unsupervised method to find topics). The truncated topics can be found in Appendix B.3, with a summary of the corresponding multinomial logistic regression model. Compared to the topics found by the LDA model, the topics here are not as easily humanly interpretable, especially when considering storylines. However, when taking into account the coefficients of the regression model, we see some connections between storyline and topic. For instance, the strongest indicator of the adultery storyline is Topic 9, which has frequent words *minister* and *sister* (being about the Prime Minister's sister), and the word *fool* which is prominent in one particular scene of that storyline. However, the real indicator of how well these topics map to storylines will be in how the model predicts left-out, or 'new', scenes.

4.5.1 Introducing new documents

In order to perform model validation and calculate the Brier score for the HMTM regression model, we must predict the response of a new document, or scene. That is, we must have a method for estimating the topic proportions $\boldsymbol{\theta}$ of a new document given an existing HMTM formed on the existing corpus, in a similar way to the method developed in Section 3.4.3 for LDA.

Given we are essentially working with a Hidden Markov Model (HMM), we can adopt techniques used to find model parameters for those. Specifically, we use the Baum-Welch algorithm [44], outlined in Section 2.5.1, which finds the parameters of an HMM given a sequence of observations. So for a given document (or 'sequence') $\mathbf{w} = (w_1, w_2, ..., w_n)$ and k topics, the Baum-Welch algorithm estimates the model $\Omega = (\boldsymbol{\Theta}, \boldsymbol{\phi}, \boldsymbol{\pi})$, where

- $\boldsymbol{\Theta}$ is a $k \times k$ matrix of transition probabilities between topics (or states) ($\Theta_{ij} = P\left(T_{m+1} = j | T_m = i\right)$ for $m = 1, 2, ..., n-1$, where T_m is the mth topic in the sequence),

- $\boldsymbol{\phi}$ is a $k \times v$ matrix of emission probabilities, *i.e.*, the topics themselves ($\phi_{ij} = P\left(W_m = j | T_m = i\right)$ for $m = 1, 2, ..., n$, where W_m is the mth word in the sequence), and

- $\boldsymbol{\pi}$ is a vector of length k of starting probabilities for each topic ($\pi_i = P\left(T_1 = i\right)$).

The Baum-Welch algorithm uses the following general process for estimating the model Ω:

1. Calculate forward probabilities $\alpha_{li} = P\left(W_1 = w_1, ..., W_i = w_i, T_i = l | \Omega\right)$ based on the current estimate for the model Ω, for $l = 1, 2, ..., k$ and $i = 1, 2, ..., n$.

   (a) For the first word in the document, and for each topic $l = 1, 2, ..., k$:

$$\alpha_{l1} = \pi_l \phi_{l w_1}.$$

(b) For word $i = 2, 3, ..., n$ in the document, and for each topic $l = 1, 2, ..., k$:

$$\alpha_{li} = \phi_{lw_i} \sum_{s=1}^{k} \alpha_{s,i-1} \Theta_{sl}.$$

2. Calculate backward probabilities $\beta_{li} = P\left(W_{i+1} = w_{i+1}, ..., W_n = w_n | T_i = l, \Omega\right)$ based on the current estimate for the model Ω, for $l = 1, 2, ..., k$ and $i = 1, 2, ..., n$.

(a) For the last word in the document, and for each topic $l = 1, 2, ..., k$:

$$\beta_{ln} = 1.$$

(b) For each word in the document $i = n-1, n-2, ..., 1$ and for each topic $l = 1, 2, ..., k$:

$$\beta_{li} = \sum_{s=1}^{k} \Theta_{ls} \beta_{s,i+1} \phi_{sw_{i+1}}.$$

3. Update Ω based on the forward and backward probabilities calculated.

(a) For $l = 1, 2, ..., k$, $i = 1, 2, ..., n$ and $s = 1, 2, ..., k$, calculate temporary variables $\gamma_l(i)$ and $\xi_{ls}(i)$ such that

$$\gamma_l(i) = \frac{\alpha_{li} \beta li}{\sum\limits_{s=1}^{k} \alpha_{si} \beta_{si}}, \quad \text{and}$$

$$\xi_{ls}(i) = \frac{\alpha_{li} \Theta_{ls} \beta_{s,i+1} \phi_{sw_{i+1}}}{\sum\limits_{l=1}^{k} \sum\limits_{s=1}^{k} \alpha_{li} \Theta_{ls} \beta_{s,i+1} \phi_{sw_{i+1}}}.$$

(b) Calculate an updated Ω^*:

$$\pi_l^* = \gamma_l(1) \quad \text{for} \quad l = 1, 2, ..., k,$$

$$\Theta_{ls}^* = \frac{\sum\limits_{i=1}^{n} \xi_{ls}(i)}{\sum\limits_{i=1}^{n} \gamma_{li}} \quad \text{for} \quad l, s = 1, 2, ..., k,$$

$$\phi_{li}^* = \frac{\sum_{t=1}^{n} I_{w_t = i} \gamma_l(t)}{\sum_{t=1}^{n} \gamma_l(t)}.$$

4. Repeat the first three steps until Ω^* has converged satisfactorily. Generally, this is when the sum of the squared differences between the model parameters estimated at consecutive steps is below some tolerance.

However, the key point here is that our emission probabilities $\boldsymbol{\phi}$ are common across all the documents in our corpus (since they are our topics) and thus when introducing new documents we assume that we already know them. Given that the Baum-Welch algorithm calculates forward and backward probabilities based on an assumed model, if we take the predetermined $\boldsymbol{\phi}$ to be the truth when analysing a new document, we should simply refrain from updating it in Step 3.

Another major difference between inference of an HMM and an HMTM is that we are now dealing with a collection of sequences (or documents), instead of just one sequence. As a result, while the topics $\boldsymbol{\phi}$ are common across all documents, the transition probabilities $\boldsymbol{\Theta}$ will change for each sequence. However, when finding the transition probabilities of a new document given an existing HMTM, we are only attempting to find the one $\boldsymbol{\Theta}$, and thus can treat it in the same manner as the HMM.

Usually we are dealing with very small probabilities in topic modelling; $\boldsymbol{\phi}$ generally has thousands to tens of thousands of columns (the size of the vocabulary) over which the probabilities must sum to one. While in theory this does not change how we would approach estimating the model parameters, computationally these probabilities are frequently recognised as zero, and thus calculations fall apart. To combat this, and to make the process more numerically stable, we implement the following adapted Baum-Welch algorithm (as demonstrated and justified in Shen [48]).

1. Calculate the modified forward probabilities, $\hat{\alpha}_{li}$, for $l = 1, 2, ..., k$ and $i = 1, 2, ..., n$ from the forward probabilities outlined in the original Baum-Welch algorithm.

(a) For the first word in the document, and for each topic $l = 1, 2, ..., k$:

$$\hat{\alpha}_{l1} = \frac{\ddot{\alpha}_{l1}}{\sum\limits_{s=1}^{k} \ddot{\alpha}_{s1}},$$

where

$$\ddot{\alpha}_{l1} = \alpha_{l1}.$$

(b) For word $i = 2, 3, ..., n$ in the document, and for each topic $l = 1, 2, ..., k$:

$$\hat{\alpha}_{li} = \frac{\ddot{\alpha}_{li}}{\sum\limits_{s=1}^{k} \ddot{\alpha}_{si}},$$

where

$$\ddot{\alpha}_{li} = \sum_{s=1}^{k} \hat{\alpha}_{s,i-1} \Theta_{ls} \phi_{lw_i}.$$

2. Calculate the modified backward probabilities, $\hat{\beta}_{li}$, for $l = 1, 2, ..., k$ and $i = 1, 2, ..., n$ from the backward probabilities outlined in the original Baum-Welch algorithm.

(a) For the last word in the document, and for each topic $l = 1, 2, ..., k$:

$$\hat{\beta}_{ln} = \frac{1}{\sum\limits_{s=1}^{k} \ddot{\alpha}_{sn}}.$$

(b) For each word in the document $i = n-1, n-2, ..., 1$ and for each topic $l = 1, 2, ..., k$:

$$\hat{\beta}_{li} = \frac{\ddot{\beta}_{li}}{\sum\limits_{s=1}^{k} \ddot{\alpha}_{si}},$$

where

$$\ddot{\beta}_{li} = \sum_{s=1}^{k} \Theta_{ls} \phi_{sw_{i+1}} \hat{\beta}_{s,i+1}.$$

3. Update $\boldsymbol{\Theta}$ based on the modified forward and backward probabilities.

(a) For $l, s = 1, 2, ..., k$, calculate an updated $\mathbf{\Theta}^*$:

$$\Theta^*_{ls} = \frac{\sum\limits_{i=1}^{n-1} \hat{\alpha}_{li} \Theta_{ls} \phi_{sw_{i+1}} \hat{\beta}_{s,i+1}}{\sum\limits_{i=1}^{n-1} \hat{\alpha}_{li} \ddot{\beta}_{li}}.$$

While we are ultimately interested in finding the topic proportions $\boldsymbol{\theta}$, the adapted Baum-Welch algorithm finds the transition matrix $\mathbf{\Theta}$. We are able to deal with this in the same way as when we found the original HMTM model, by taking $\boldsymbol{\theta}$ to be the equilibrium probabilities of $\mathbf{\Theta}$.

4.5.2 Model validation

Using the above process, we attempt to estimate the topic proportions $\boldsymbol{\theta}_j$ for every left-out document in the corpus in order to predict the probabilities of that scene belonging to each storyline, given the remaining 78 scenes. Given the short length of some scenes, and the fact that words unique to a scene are removed in order for the above algorithm to work, only 57 of the left-out scenes generated a numerically stable approximation for $\mathbf{\Theta}_j$. From those 57 scenes, we again calculate the Brier score as with the word count and LDA models. For the HMTM regression model, the Brier score is 1.5749. While still not up to the standard of the word count model at 0.8255, this appears to be an improvement on the LDA model at 1.6351, meaning that dropping the 'bag of words' assumption by incorporating dependencies between consecutive words in fact improves the predictive capability of the model. However, it should be kept in mind that while the Brier score for the HMTM regression model is technically better, it is still less capable than LDA of handling small documents (as the scenes not predicted by the HMTM regression model tend to be those with fewer words). It would be worth applying these methods to larger corpora, with longer documents, in future to see how they compare.

Figure 4.10 shows the probability of each scene being correctly labelled based on the rest of the corpus. Interestingly, although with the LDA model there appeared to be a correlation between how well a scene was predicted

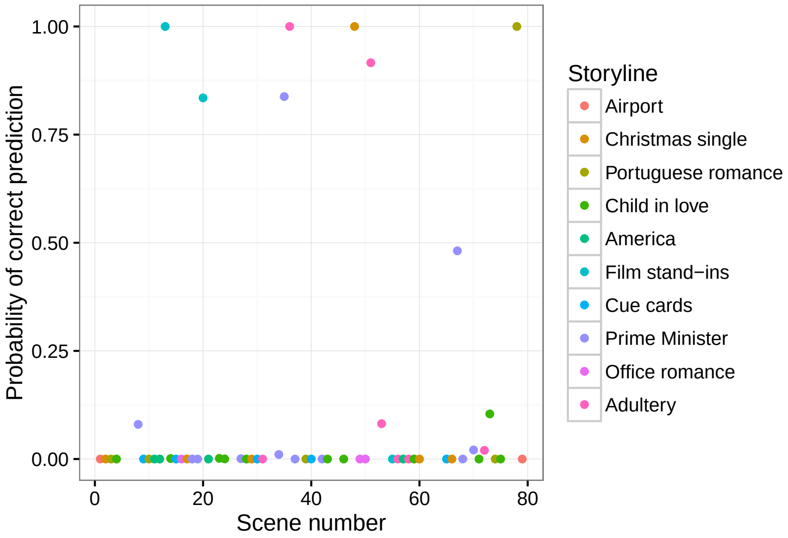


Figure 4.10: Plot of the probability for 57 of the 79 scenes in the *Love Actually* corpus being allocated to the correct storyline using leave-one-out validation of the HMTM regression model. Each scene is coloured based on its true storyline.

and how frequent the storyline is, this is less obvious here. In particular, one of the best predicted storylines appears to be that of the film stand-ins, a reasonably minor part of the film.

4.6 Persistent hidden Markov topic model

One of the motivating ideas behind having topic dependencies between consecutive words, as in the HMTM model, is that some documents will have a predisposition to stay in the same ‘topic’ for a long sequence, such as a sentence or a paragraph. As discussed earlier, this argument could apply to story-driven dialogue such as in the *Love Actually* corpus. Figure 4.11b shows a heatmap of all the topic transitions for the HMTM model on the entire corpus. If this model were to have a predisposition for staying in one topic over long periods of time, we would see a strong diagonal element to the plot. While there appears to be some difference to the transitions of the

LDA model on the corpus in Figure 4.11a, it is not marked. Therefore, we develop a new modification to HMTM to enforce topic persistence.

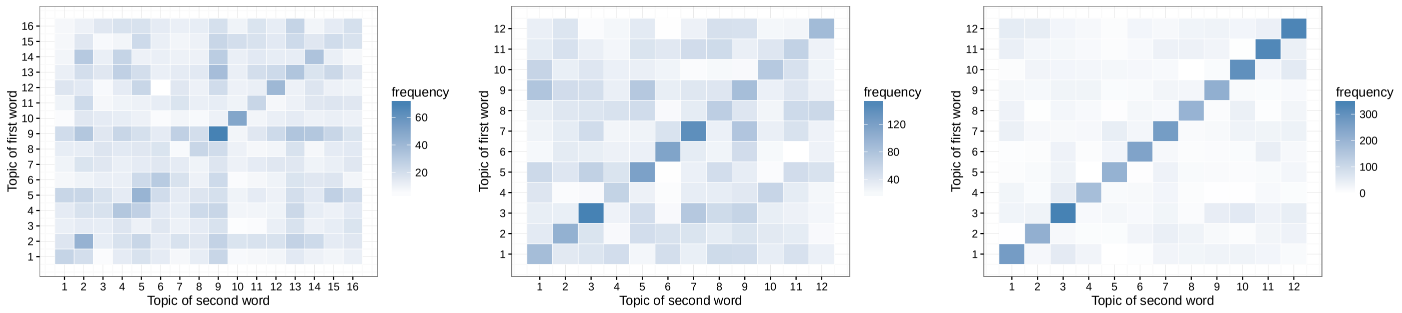
In the generative process for the HMTM (outlined in Section 4.5), the rows of the transition matrix $\mathbf{\Theta}_j$ are influenced by Dirichlet distributions with hyperparameters $\gamma_1, \gamma_2, ..., \gamma_k$ (vectors of length k) for the k rows, for document $j = 1, 2, ..., m$. These γ_l are updated along with the other model hyperparameters when performing Gibbs sampling for model inference. In the standard HMTM as proposed in Section 4.5, the initial values for these parameters are

$$\gamma_l = \left(\frac{1}{k}, \frac{1}{k}, ..., \frac{1}{k}\right) \quad \text{for} \quad l = 1, 2, ..., k.$$

That is, we assume that, given the state T is in a certain topic for word i, we have an equal probability of word $i + 1$ being in any topic. While the γ_l update at each iteration of the Gibbs sampling procedure, there is still a tendency to converge towards topics that switch rapidly. This is because the topics found by any topic model are not necessarily the only topics that can adequately summarise the corpus, which may not be appropriate for the corpus. To promote more persistent topics throughout a document, we instead choose the initial values of the γ_l to favour going into the same topic at word $j + 1$. That is, we choose

$$\gamma_{ls} = \begin{cases} \delta + \frac{(1-\delta)}{k} & \text{if} \quad l = s \\ \frac{(1-\delta)}{k} & \text{elsewhere,} \end{cases}$$

for $l, s = 1, 2, ..., k$ and $\delta \in [0, 1]$. The element of the hyperparameter corresponding to staying in the same topic between words is above δ. For the *Love Actually* analysis, we choose a persistent HMTM with $\delta = 0.99$. Figure 4.11c shows a heatmap of the transitions between topics for all documents in the *Love Actually* corpus given the modified persistent HMTM with the same number of topics as the HMTM found in Section 4.5. From this, we can see that the topics chosen are clearly more predisposed to longer sequences in the same topic than the original HMTM.



(a) LDA

(b) HMTM

(c) Persistent HMTM

Figure 4.11: Heatmap of the frequencies of transitioning from each topic to each topic, for the LDA model with 16 topics, the HMTM with 12 topics and the persistent HMTM with $\alpha = 0.99$ and 12 topics, over the entire *Love Actually* corpus.

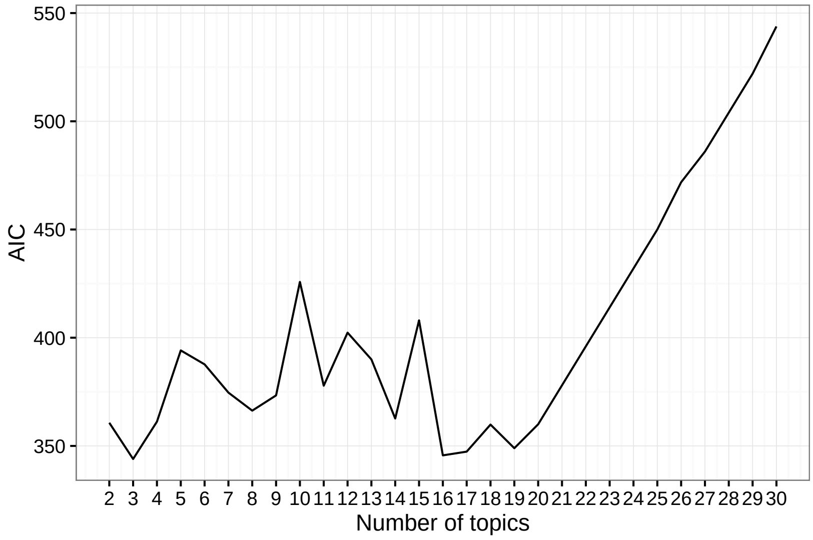


Figure 4.12: Plot of the AIC for different numbers of topics in the persistent HMTM multinomial logistic regression model on the *Love Actually* dataset.

Using the same method as with the previous models, we find the best number of topics for the persistent HMTM regression model by comparing AIC values. From Figure 4.12, we choose the model with three topics. These topics, as well as the regression model parameters, are in Appendix B.4. With three topics, the most frequent words in each topic now tend to be very general, mostly function words such as *just*, *yes* and *well*. However, there are still some reasonably indicative words that appear frequently, such as *jewellery* in Topic 2, which corresponds (as seen clearly in Figure 4.5) to the adultery storyline. This is then reflected in the regression model, where Topic 2 is an indicator of this storyline.

4.6.1 Model validation

As with the standard HMTM regression model, we estimate the topic proportions $\boldsymbol{\theta}_j$, $j = 1, 2, ..., m$, using the method outlined in Section 4.5. For this model, the process was more stable, allowing 77 of the 79 scenes to be estimated. For the interest of comparison however, we again calculate the

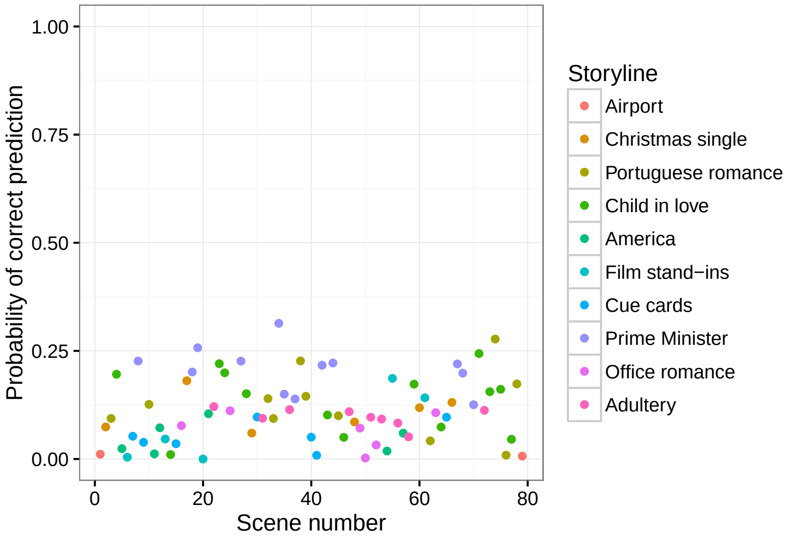


Figure 4.13: Plot of the probability for 77 of the 79 scenes in the *Love Actually* corpus being allocated to the correct storyline using leave-one-out validation of the persistent HMTM regression model. Each scene is coloured based on its true storyline.

left-out probabilities of each of the 57 'stable' scenes in the corpus belonging to each storyline, given the remaining scenes. The Brier score for the persistent HMTM regression model with three topics is therefore 0.9124. This is a definite improvement on the standard HMTM regression model at 1.5749, and is much more competitive with the word count model at 0.8255 (also, with all but two of the scenes predicted in the persistent HMTM regression model, we are able to draw a much better comparison here than between the standard HMTM and LDA or word count models).

The probabilities for each scene's storyline being correctly predicted are in Figure 4.13. Compared to the previous models, where there were much more distinct values between scenes, the persistent HMTM regression model is a lot more 'cautious' in its approach to prediction. To get a better idea of how these models compare, we instead look at the final 'hard' classifications for their predictions of left-out scenes. That is, given the model, in which storyline is each scene *most* likely to appear?

The percentage of correct hard classifications for each model is seen in Table 4.2. These match up reasonably well with the Brier scores found earlier. However, the word count model performs markedly better than the persistent HMTM regression model according to hard classifications, but less so when comparing Brier scores. This difference could be attributed to the cautiousness of the persistent HMTM.

Model	**Percentage correct (%)**	**Brier score**
Word count	26.58	0.8255
LDA	12.66	1.6351
HMTM	14.04	1.5749
Persistent HMTM	15.58	0.9124

Table 4.2: Table of the percentage of hard classifications of storylines for each left-out scene in the corpus that are correct, alongside the Brier score, for each model.

The value of $\delta = 0.99$ was chosen in order to obtain noticeably persistent topics. However, this is not to say it is necessarily the best value for the problem at hand. Figure 4.14 shows the Brier scores for the persistent HMTM regression model with 12 topics for values of δ between 0.1 and 0.9. These scores range between 1.3958 and 1.6839, around the Brier score of 1.5749 of the original HMTM regression model. Interestingly, the scores increase until $\delta = 0.6$, when they markedly decrease below the value of the original model. These results suggest that a strong preference towards staying in the same topic for consecutive words would benefit the performance of the model, whereas a weaker preference may disadvantage the model.

While increasing the persistence parameter δ may not necessarily improve the predictive capability of the model, Figure 4.15 shows the effect it has on the stability of the model. That is, as δ increases so do the number of documents for which the model is capable of estimating the topic proportions $\boldsymbol{\theta}_j$. For models with small documents, this is a worthwhile improvement.

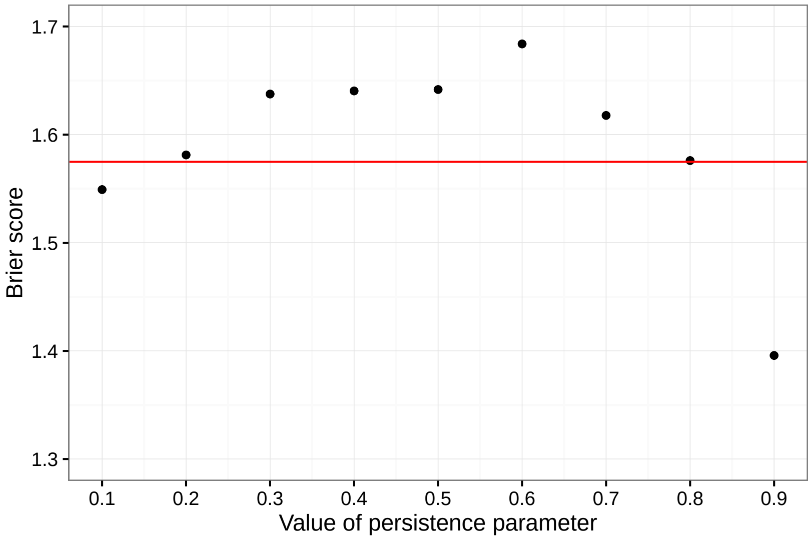


Figure 4.14: Plot of the Brier score for the persistent HMTM regression model for various values of δ on the *Love Actually* corpus. The horizontal line represents the Brier score of the original HMTM regression model found in Section 4.5.

4.7 Discussion

From the results in this chapter, when it comes to prediction on the *Love Actually* corpus, it is better to not make the 'bag of words' assumption when applying topic models as a dimension reduction step. Moreover, preferencing a reliance on staying in the same topic over a sequence of words in our corpus has the ability to improve the model's performance in predicting storyline. The best model in this chapter is still that which uses individual words as predictors, rather than topics. However, while we consider this to be a standard to attempt to reach with topic model-based regression models, it is impractical to use that method for large corpora.

From here, we may consider investigating a supervised method which incorporates the same Markovian structure as the HMTM regression model proposed in this chapter. As seen in Chapter 3, this would mean the model is more likely to find relevant and indicative topics for the response and

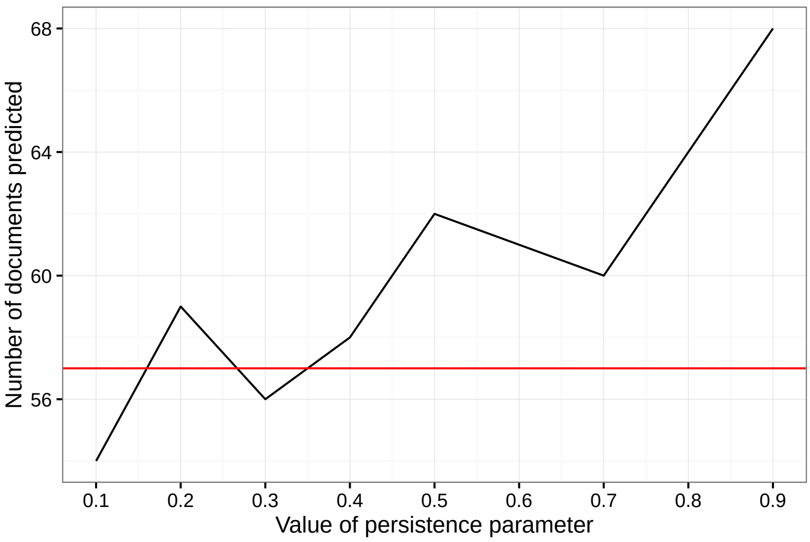


Figure 4.15: Plot of the number of documents on which the persistent HMTM regression model is able to perform leave-one-out prediction in the *Love Actually* corpus, for various values of δ. The horizontal line represents the number of documents of the original HMTM regression model found in Section 4.5.

therefore is more likely to perform better predictions.

Chapter 5

Conclusions and further research

5.1 Conclusions

The purpose of this thesis has been to outline and implement a methodology for prediction using topic models as a data processing step in a regression model. In doing so, we investigate how various topic model features affect how well the topic regression model makes predictions.

Chapter 3 focuses on, firstly, the implementation of a logistic regression model using latent Dirichlet allocation (LDA) [13] as a preprocessing step, and then compares it to a supervised equivalent (supervised LDA, or sLDA [12]). These models are tested on a corpus of cat advertisements from the trading website, Gumtree[1]. While neither of these models were able to perform quite as well as the regression model using individual word counts as predictors, they both performed markedly better than random allocation. In developing the LDA regression model, it was necessary to create a method using maximum likelihood estimation for estimating the true topic proportions of new, or left out, documents. While also saving on the computation of a new topic model whenever new documents are introduced, this method

[1] www.gumtree.com.au

allows us to perform true prediction on the response for these documents. Through simulations in which documents were created based on the LDA generative process, we can see that this estimation is effective in approximating the true topic proportions for a much smaller computational cost.

We also showed that, given the same number of topics, the supervised equivalent of LDA, sLDA, performs slightly better than its unsupervised counterpart. This is expected, as the only change is that the topics found by sLDA are chosen with the response variable in mind, whereas LDA simply fits topics that suit the corpus in an arbitrary way. However, in this particular case these results are reasonably close; this may be attributed to the simplicity of the corpus we are dealing with, and its vocabulary. This also corroborates the success of the word count model on the Gumtree problem. Likewise, the model chosen using cross validation prediction error (CVPE) [21] for sLDA, containing only two topics, performs worse than the 26-topic LDA regression model.

Interestingly, this chapter also shows that while a step-up procedure is necessary for the word count model, when used on the LDA regression model, it makes no significant difference to the predictive performance. However, this behaviour may change, or indeed be necessary, when considering models with larger numbers of topics, applied to larger datasets than covered here.

Chapter 4 continues the investigation into the effect of various topic model features on predictive capability by relaxing the ‘bag of words’ assumption made by all models in the previous chapter. This time, the dialogue from the 2003 movie, *Love Actually*[2] [18], is our corpus. As with the Gumtree problem, we use a regression model using individual words as predictor variables as our basis for comparison. In this case, the step-up procedure used to find the word count model gives us three words as predictors: *minister*, *night* and *around*.

LDA is again used as our ‘baseline’ topic model for this chapter. As a ‘bag of words’ model, we are able to use it as a comparison to later models which

[2] http://www.imdb.com/title/tt0314331/

introduce document structure. Using the method developed in Chapter 3, LDA does not perform as well as the word count model, according to the Brier scores [15] calculated on leave-one-out regression for each model. This is not totally surprising, given the relative success of the word count model in the Gumtree problem. However, recall that topic models are used as a form of dimension reduction in order to be able to process large amounts of data. We are simply able to use a word count model here due to the relatively small size of our corpus. What we are really interested in is how various topic models fare against each other in a regression setting.

In order to investigate the incorporation of document structure into our topic regression models, we consider the hidden Markov topic model (HMTM) [4]. As with the LDA regression model, we require some way of estimating new documents' topic proportions, given an existing model over our corpus. To do so, we employ techniques used to estimate the parameters of hidden Markov models (HMMs), specifically, the Baum-Welch algorithm [44]. While the Baum-Welch algorithm uses an iterative method to update all parameters in the model, we are interested in only updating the transition probabilities, which will give us an estimate for our topic proportions in the new document. To do so, we fix the emission probabilities (topics, in this case) in the algorithm, as they are 'known' from our existing model, and apply the Baum-Welch algorithm with this constraint to the new document.

The HMTM performs slightly better than the LDA regression model (in that it has a lower Brier score for leave-one-out regression), although still not as well as the word count model. This indicates that incorporating document structure into the topic model improves the predictive capability of the regression model, but only slightly using this approach.

One of the reasons behind employing document structure in our model is that we assume that words in the same sentence or paragraph would have a higher chance of belonging to the same 'topic' (in particular, those in documents with a narrative structure, such as the *Love Actually* corpus). However, looking at topic transition frequencies for the HMTM shows that, while still preferring sequences in the same topic more than the LDA model, it is not a remarkable trait of the model. We therefore develop a method for a persistent HMTM, where longer sequences of words in one topic are preferenced.

The Brier score for the persistent HMTM regression model with a persistence parameter value of $\delta = 0.99$ indicates that it outperforms both the original HMTM and LDA models, and is more on par with the word count model for which we are aiming. These results imply that, for this particular kind of problem, more persistent topics are more useful for prediction. This indicates further the importance of document structure in our model. We investigate the effect of changing the persistence parameter δ, and show that for δ close to 1, the model improves the predictive accuracy over the original HMTM regression model. However, for weaker preferences, the persistent HMTM does not outperform the original HMTM. It is worth noting that, while the accuracy of the model may not necessarily improve, the stability of the model also tends to increase with the strength of the persistence.

Overall, this thesis has provided a statistical framework by which we are able to implement topic modelling as a reduction process on our text data, in order to make predictions about or from the text. We have done so through the adaptation of existing topic models into a novel regression framework, with particular regard to the prediction of new documents. The importance of supervised learning and the incorporation of document structure into the model for predictive accuracy have also been highlighted here.

5.2 Outlook for further research

This thesis has examined the effect of two features on the predictive accuracy of topic regression models: supervised learning and the ‘bag of words’ assumption. Given that our results find that the presence of supervision and document structure in a model both improve prediction, a logical next step would be to investigate a model that contains both of these things.

Specifically, we may look into incorporating a response variable into the generative process of a hidden Markov topic model, as used in Chapter 4, where transition probabilities are affected by said response. Something to bear in mind when doing so, however, is the computational expense created by both supervised learning and assumed document structure.

Similarly, we have only examined one way in which the 'bag of words' assumption can be dropped. Further work could look into other document structures (for instance, assuming a topic per sentence as in the hidden topic Markov model [25]), and even the inclusion of language structure and part-of-speech tagging [24]. It may also be worth investigating a dynamic topic regression model, where either topics or topic proportions change over time [10, 55]. For example, one would assume the order of documents in the *Love Actually* corpus to be important, and investigating whether the placement of a scene in the movie has an effect on its prediction would potentially show this. This idea can be extended to larger corpora, for example Google Books[3]: topic modelling regression could be applied to these datasets in order to predict major cultural events, from the change in topics over time.

In general, each topic model has been developed with specific features for specific purposes. These features should be considered when applying topic regression models. While this thesis provides a framework for three existing topic models in a regression context, this principle can be easily extended to a much more varied range of topic models, as discussed in Chapter 2.

A useful tie-in to the methods developed in this thesis could be sentiment analysis. In both Chapters 3 and 4, sentiment analysis could feasibly be employed as an analytical tool to understand more about the topics and the corpus. For the Gumtree problem, and as mentioned in Chapter 3, words belonging to advertisements with a relinquished status appear to be more emotive, and thus sentiment analysis may find interesting correlations between the status of a cat advertisement and the emotion in its language. In the case of the *Love Actually* problem, much work has been performed in

[3] `https://books.google.com/`

modelling narrative structure through sentiment [35]. These techniques have the ability to aid in our understanding of the corpus we are using, as well as the potential to be integrated into our topic regression models through analysis of the sentimental value of the topics themselves.

Another major trend in the field of natural language processing at the moment is word embedding, *i.e.* mapping individual words to a vector space. The most well known of these is Google's Word2vec [36], which aims to represent relationships between words through their position in the vector space. As another form of dimension reduction for text, it would be prudent to compare the performance of word embedding to topic modelling in a predictive framework using a suitable corpus.

Appendix A

Gumtree corpus

A.1 Word count regression model coefficients

Tables A.1 to A.4 show the model coefficients of the step-up word count regression model found for the Gumtree corpus in Section 3.3.

	Estimate	Std. Error	z value	$P(>\|z\|)$
(Intercept)	0.3136	0.0988	3.17	0.0015
kitten	-0.6054	0.0670	-9.04	0.0000
cat	0.6026	0.0840	7.17	0.0000
readi	-0.7366	0.1411	-5.22	0.0000
away	1.9342	0.3591	5.39	0.0000
good	0.6062	0.1141	5.32	0.0000
need	0.4776	0.1126	4.24	0.0000
move	1.4327	0.3316	4.32	0.0000
free	1.9699	0.1956	10.07	0.0000
ragdol	-0.4311	0.1507	-2.86	0.0042
year	1.2092	0.2078	5.82	0.0000
breeder	-0.8378	0.2128	-3.94	0.0001
month	0.4046	0.1808	2.24	0.0252
check	-0.4480	0.2211	-2.03	0.0427
week	-0.9678	0.1254	-7.72	0.0000
rescu	2.7668	0.5163	5.36	0.0000
found	1.9431	0.3845	5.05	0.0000
blue	-0.6243	0.1670	-3.74	0.0002
vaccin	-0.6985	0.1279	-5.46	0.0000
rais	-1.2639	0.2760	-4.58	0.0000
give	1.0121	0.2234	4.53	0.0000
desex	0.5368	0.1357	3.96	0.0001
old	0.5494	0.1334	4.12	0.0000
sale	-0.6590	0.1551	-4.25	0.0000
longer	1.6599	0.4479	3.71	0.0002
train	-0.4677	0.1236	-3.79	0.0002
outsid	1.0201	0.3400	3.00	0.0027
cant	1.0795	0.2798	3.86	0.0001
parent	-1.2613	0.3727	-3.38	0.0007
breed	-0.8080	0.2845	-2.84	0.0045

Table A.1: Table of the first 30 model coefficients for the step-up word count regression model on the Gumtree corpus.

	Estimate	Std. Error	z value	$P(>\|z\|)$
rehom	1.1293	0.3496	3.23	0.0012
long	-1.4947	0.2875	-5.20	0.0000
adopt	0.9749	0.2380	4.10	0.0000
date	1.1996	0.3019	3.97	0.0001
mother	-0.6260	0.2190	-2.86	0.0043
inform	-1.0306	0.3297	-3.13	0.0018
suit	1.2529	0.3519	3.56	0.0004
first	-0.6578	0.2492	-2.64	0.0083
dad	-0.8698	0.3706	-2.35	0.0189
box	0.9049	0.3351	2.70	0.0069
play	-0.4304	0.1164	-3.70	0.0002
friend	0.2732	0.1525	1.79	0.0732
feel	-0.9136	0.3410	-2.68	0.0074
unfortun	0.7700	0.3621	2.13	0.0335
chocol	-0.6098	0.3437	-1.77	0.0760
chip	-0.6497	0.2540	-2.56	0.0105
sad	1.0984	0.4705	2.33	0.0196
seal	-0.3583	0.2285	-1.57	0.1169
view	-0.6982	0.2961	-2.36	0.0184
leav	-1.0697	0.3285	-3.26	0.0011
look	0.3266	0.1318	2.48	0.0132
indoor	0.5001	0.2046	2.44	0.0145
find	0.4288	0.2074	2.07	0.0387
happi	-0.5729	0.2571	-2.23	0.0258
price	-0.6944	0.2465	-2.82	0.0049
socialis	-1.0758	0.4349	-2.47	0.0134
fluffi	-0.7307	0.2762	-2.65	0.0081
attent	0.8891	0.3337	2.66	0.0077
due	0.3979	0.2093	1.90	0.0574
meet	0.5740	0.2719	2.11	0.0348

Table A.2: Table of the second 30 model coefficients for the step-up word count regression model on the Gumtree corpus.

	Estimate	Std. Error	z value	$P(>\|z\|)$
also	-0.3474	0.1667	-2.08	0.0372
famili	-0.4282	0.1714	-2.50	0.0125
around	0.3112	0.1688	1.84	0.0653
just	0.2711	0.1750	1.55	0.1213
black	0.2523	0.1012	2.49	0.0126
male	-0.1644	0.1013	-1.62	0.1046
got	0.6575	0.3441	1.91	0.0560
boy	-0.2357	0.1157	-2.04	0.0417
ginger	0.3341	0.1575	2.12	0.0339
ador	-0.4915	0.2351	-2.09	0.0366
persian	-0.4797	0.2734	-1.75	0.0793
hous	0.4024	0.2077	1.94	0.0527
sure	0.6166	0.3792	1.63	0.1039
cudd	0.6014	0.2649	2.27	0.0232
lot	-0.5248	0.2713	-1.93	0.0531
one	-0.1686	0.0955	-1.77	0.0775
life	0.6942	0.4075	1.70	0.0885
vet	-0.3517	0.2004	-1.75	0.0793
get	0.2953	0.1632	1.81	0.0705
send	-0.5764	0.3185	-1.81	0.0703
pleas	0.2500	0.1268	1.97	0.0488
babi	-0.4184	0.2087	-2.00	0.0450
best	0.5742	0.2889	1.99	0.0468
question	-0.7287	0.3808	-1.91	0.0557
reveal	-0.6056	0.2450	-2.47	0.0135
click	0.5391	0.2605	2.07	0.0385
sweet	-0.5972	0.3168	-1.88	0.0594
kitti	-0.3567	0.1997	-1.79	0.0740
well	0.3304	0.1916	1.72	0.0846
person	-0.4188	0.2217	-1.89	0.0589

Table A.3: Table of the third 30 model coefficients for the step-up word count regression model on the Gumtree corpus.

	Estimate	Std. Error	z value	$P(>\|z\|)$
great	0.2824	0.1673	1.69	0.0914
care	0.4097	0.2233	1.83	0.0665
come	-0.2342	0.1388	-1.69	0.0915
femal	-0.1586	0.1081	-1.47	0.1423
left	-0.3152	0.2016	-1.56	0.1180
day	0.4841	0.2997	1.62	0.1063
number	-0.4162	0.2918	-1.43	0.1538
extrem	0.4916	0.3389	1.45	0.1469

Table A.4: Table of the final eight model coefficients for the step-up word count regression model on the Gumtree corpus.

A.2 LDA non-predictive topics

Tables A.5 to A.7 show the 30 most frequent words in each of the topics found by the LDA non-predictive model on the Gumtree corpus, as discussed in Section 3.4.2.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	grey	cute	mit	asap	cute
2	half	tiger	immunis	gone	cream
3	sex	urgent	flee	deflead	fold
4	decemb	today	allerg	landlord	scottish
5	homepleas	rear	sock	giveaway	three
6	mix	vacc	himalayan	anymor	playful
7	seper	five	pair	rough	needl
8	properti	pair	pickup	vacc	allerg
9	partner	vac	msg	bella	longhair
10	havent	asap	funni	blackwhit	law
11	most	entir	organ	laid	missi
12	collect	havent	puss	cutest	tablet
13	cur	mia	town	reg	duke
14	document	multi	british	rid	malex
15	lulu	mths	lil	sms	andi
16	mchip	playful	decemb	arabella	freight
17	sterilis	ten	upload	bye	ocicat
18	catterypleas	boo	awesom	champagn	stripi
19	christma	deaf	dsh	oldpleas	tomorrow
20	russian	inbox	marbl	boypic	wild
21	summer	sealpoint	princ	girlpic	downsiz
22	charg	tame	rubi	kit	farm
23	cute	bundl	amber	park	fluff
24	beig	daddi	collect	toliet	malescottish
25	dollar	leo	half	unfortunat	scoop
26	forest	partner	mine	affectionatesh	unknown
27	kittenpleas	pickup	pearl	estim	weekday
28	mth	potti	deflead	firm	agre
29	unknown	sox	maximus	fix	chair
30	upload	airport	eight	fleed	foot

Table A.5: Table of the 30 most probable words in Topics 1 to 5 for the LDA non-predictive model on the Gumtree corpus.

	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
1	four	stripe	russian	cute	approx
2	cross	approx	rag	orang	cross
3	anymor	alot	doll	wks	scratcher
4	energet	rex	half	birman	british
5	sms	molli	vac	drink	park
6	cinnamon	tick	steril	persian	sterilis
7	novemb	msg	grey	buddi	giveaway
8	sabrina	inject	unit	plz	hill
9	temper	mitt	feb	there	seven
10	marbl	wash	tammi	mix	scoop
11	willow	himalayan	catch	cheer	shot
12	deflea	proper	eve	twin	today
13	vic	read	sign	mous	tortoiseshel
14	pug	unknown	bundl	energet	reg
15	snow	devon	collect	fold	cutest
16	nine	sterilis	vacat	hey	inject
17	persian	charg	bathroom	onlypleas	oldveri
18	quarter	farm	british	sock	selkirk
19	candi	fluff	five	coco	treatedlitt
20	hill	oscar	kittensal	desexedpleas	vac
21	carolyn	pregnant	reluct	half	catswa
22	clay	present	revealno	msg	most
23	dublin	shell	entir	neg	nervous
24	firm	bombay	potti	pudpud	read
25	oldfre	destruct	runt	rear	chose
26	oldsh	hii	salesh	reg	figur
27	oliv	liverpool	vicki	unforeseen	hey
28	onlysh	machin	astro	vaccinatedworm	hug
29	properti	meetoo	azzi	batman	moggi
30	rid	michael	kim	bob	oldthey

Table A.6: Table of the 30 most probable words in Topics 6 to 10 for the LDA non-predictive model on the Gumtree corpus.

	Topic 11	Topic 12	Topic 13	Topic 14	Topic 15
1	detail	ginger	shell	persian	manx
2	three	grey	tortois	drink	asap
3	loveabl	rough	needl	exot	tortoiseshel
4	feb	cross	flame	chinchilla	drink
5	pregnant	farm	nswcfa	sex	christma
6	countri	kiss	dark	dark	color
7	sorri	sms	lynx	choc	patch
8	toffe	unsur	rough	christma	cute
9	flame	ono	cute	rental	friday
10	shadow	cinnamon	cross	shade	urgent
11	sunday	tower	farm	freight	wild
12	afraid	burmilla	british	ect	flat
13	gray	christma	chill	munchkin	most
14	nswcfa	correct	kit	minut	tortishel
15	reg	fold	central	present	siberian
16	allerg	penni	await	boot	geelong
17	catch	teddi	besid	femalex	snowsho
18	desexedsh	applic	lilli	onclick	approx
19	manx	blackbrown	oldsh	smokey	maleal
20	shini	onlythey	adelaid	fleed	nala
21	van	spare	anymor	thankyou	sweetheart
22	vacc	tigger	birthday	harri	aloud
23	golden	anymor	bombay	jorgiaryan	begin
24	produc	birthday	immun	middl	devon
25	stuff	daisi	moo	stripe	hey
26	charcoal	homew	proven	toowoomba	lighter
27	cross	island	seaford	american	dark
28	exact	puppi	sphynx	cheap	defin
29	freight	ridg	thursday	cooki	etci
30	hill	shell	unknown	dudley	milli

Table A.7: Table of the 30 most probable words in Topics 11 to 15 for the LDA non-predictive model on the Gumtree corpus.

A.3 LDA: topics and regression model coefficients

Table A.8 shows the coefficients of the LDA regression model on the Gumtree corpus, as found in Section 3.4.3. Tables A.9 to A.14 show the 30 most frequent words in each of the topics of this model.

	Estimate	Std. Error	z value	$P(>\|z\|)$
(Intercept)	3.9967	4.8178	0.83	0.4068
Topic 1	-8.5316	7.6336	-1.12	0.2637
Topic 2	-7.5924	7.4823	-1.01	0.3102
Topic 3	-7.0083	7.2508	-0.97	0.3338
Topic 4	-1.8518	7.4819	-0.25	0.8045
Topic 5	-2.8442	7.4180	-0.38	0.7014
Topic 6	-8.4528	6.9368	-1.22	0.2230
Topic 7	-7.0907	7.3979	-0.96	0.3378
Topic 8	-8.5869	7.4502	-1.15	0.2491
Topic 9	-6.3963	7.3634	-0.87	0.3850
Topic 10	-10.1214	7.7404	-1.31	0.1910
Topic 11	-11.9580	7.2116	-1.66	0.0973
Topic 12	-1.8043	7.1017	-0.25	0.7994
Topic 13	-11.4328	7.6519	-1.49	0.1351
Topic 14	-3.9951	7.7478	-0.52	0.6061
Topic 15	-5.9370	7.6918	-0.77	0.4402
Topic 16	-1.9307	7.6891	-0.25	0.8017
Topic 17	-8.9829	7.3485	-1.22	0.2216
Topic 18	-4.5563	7.6950	-0.59	0.5538
Topic 19	-7.0897	7.8347	-0.90	0.3655
Topic 20	2.8637	7.4700	0.38	0.7015
Topic 21	-6.1064	7.7221	-0.79	0.4291
Topic 22	-10.8742	7.0615	-1.54	0.1236
Topic 23	-11.5412	7.9514	-1.45	0.1466
Topic 24	3.6529	7.3477	0.50	0.6191
Topic 25	0.3343	7.2812	0.05	0.9634

Table A.8: Table of the model coefficients for the LDA regression model on the Gumtree corpus.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	half	tiger	sex	cross	cute
2	homepleas	allerg	havent	grey	three
3	rear	partner	decemb	msg	approx
4	persian	vac	himalayan	landlord	gone
5	most	properti	pickup	flee	laid
6	collect	vacc	toffe	half	cur
7	sterilis	awesom	lil	bella	plz
8	christma	asap	rubi	puss	arabella
9	document	deaf	british	temper	oldpleas
10	catterypleas	deflead	mine	immunis	sterilis
11	lulu	buddi	pair	cute	anymor
12	mia	bundl	playful	champagn	bob
13	russian	four	boo	calico	devon
14	seper	oldh	dsh	park	femalethey
15	cute	rosi	cutest	reaction	puss
16	mchip	ten	leo	rear	strip
17	charg	thankyou	colourpoint	reluct	toliet
18	fasa	today	daddi	wks	unit
19	forest	unknown	lovley	airport	beig
20	kittenpleas	bunni	marbl	girlblack	bribi
21	panda	councilkitten	mchip	hill	buki
22	summer	dollar	oldsh	min	desexedmicrochip
23	urgent	doubl	oliv	mixtur	exact
24	willow	effection	scope	simba	furrev
25	accommod	fat	upload	asappleas	himalayan
26	availablewil	freight	amus	british	malethey
27	cur	homeif	beach	ginger	mit
28	dogsreadi	homew	bell	iam	ppl
29	four	inject	bulli	mia	puppi
30	gooleydol	lake	calltext	milo	stephani

Table A.9: Table of the 30 most probable words in Topics 1 to 5 for the LDA predictive model on the Gumtree corpus.

	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
1	cream	asap	mit	grey	flame
2	fold	gone	grey	gone	cute
3	scottish	rough	molli	vacc	four
4	urgent	today	shot	cute	deflead
5	british	alot	freight	giveaway	immunis
6	anymor	anymor	playful	mix	christma
7	sms	five	allerg	feb	deflea
8	sock	town	blackwhit	tick	asap
9	amber	reg	missi	princ	mitt
10	ocicat	seper	malex	cheer	alot
11	pair	cross	pug	kit	sms
12	seper	funni	flee	boypic	bye
13	strip	lynx	fluff	carolyn	marbl
14	canberra	multi	shell	deflead	snow
15	estat	unfortunat	sox	girlpic	wash
16	flaco	collect	steril	half	feb
17	foot	fluff	tame	marbl	present
18	forth	housem	banana	toowoomba	sealpoint
19	homethes	sex	destruct	upload	upload
20	inbox	sms	heidi	jack	meetoo
21	kittens	sunday	initi	mesh	mths
22	kittenw	cutest	liverpool	multi	pend
23	leftkitten	decemb	mous	nala	addor
24	rehous	giveaway	needl	pearl	bluegrey
25	tom	macgyv	oldthey	pepper	direct
26	tortis	stripe	scoop	poo	herth
27	abyssinian	talker	animalsh	proper	maximus
28	cameo	tickl	archi	runt	read
29	casey	airport	arni	sign	rough
30	catworm	apricot	availab	trainedwel	russian

Table A.10: Table of the 30 most probable words in Topics 6 to 10 for the LDA predictive model on the Gumtree corpus.

	Topic 11	Topic 12	Topic 13	Topic 14	Topic 15
1	rag	grey	detail	dark	wks
2	doll	ginger	orang	mix	giveaway
3	grey	stripe	three	rex	unit
4	steril	approx	drink	fix	novemb
5	entir	sock	pair	countri	park
6	fleed	tammi	russian	grey	energet
7	immunis	bundl	organ	energet	inject
8	law	present	tomorrow	tower	onlypleas
9	nervous	duke	vic	unforeseen	immunis
10	orang	tablet	cross	azzi	today
11	devon	energet	needl	homeveri	loveabl
12	half	strip	firm	hug	read
13	nine	unknown	hey	norwegian	sharon
14	oldveri	asap	chose	septemb	twin
15	rid	dublin	femaleblack	trainedkitten	british
16	vicki	patch	forc	willow	hill
17	allerg	tlc	neg	homew	kittensal
18	kim	weekday	sunni	jorgiaryan	manx
19	persian	affectionatesh	bendigo	numbersregist	oldh
20	benni	agre	bombay	oliv	sterilis
21	bub	anymor	buddi	onclick	wormedpleas
22	catpleas	bathroom	energet	patch	anymor
23	clay	burmilla	flat	playful	aqua
24	decept	dust	freight	reg	astro
25	energet	malescottish	golden	sept	bread
26	firm	packthey	meclick	there	cleo
27	homewil	sam	ono	web	cream
28	leas	seven	poppi	zeus	daisi
29	northern	shot	recept	affectionatelov	eve
30	onlysh	woman	stripi	ancatsal	finish

Table A.11: Table of the 30 most probable words in Topics 11 to 15 for the LDA predictive model on the Gumtree corpus.

	Topic 16	Topic 17	Topic 18	Topic 19	Topic 20
1	birman	persian	approx	russian	three
2	ginger	exot	cute	four	detail
3	loveabl	chinchilla	four	reg	nswcfa
4	energet	ect	scratcher	plz	cinnamon
5	most	shade	lynx	cross	wild
6	revealno	fold	unsur	collect	drink
7	michael	anymor	nswcfa	five	sterilis
8	mths	ono	pearl	applic	afraid
9	oldthey	ten	birthday	catswa	gray
10	pregnant	catch	desexedsh	desexedpleas	boot
11	rex	correct	buddi	penni	vac
12	stuff	treatedlitt	landlord	read	entir
13	vacat	colourpoint	onlythey	seaford	coco
14	detail	purri	sms	decemb	rear
15	gone	tawni	batman	detail	sunday
16	grey	whitefemal	butt	foodpleas	burmilla
17	ita	applic	gir	oldveri	giveaway
18	nicknam	awesom	half	reluct	kiara
19	oldsh	beati	msg	scoop	lola
20	organ	flame	potti	goofi	puppi
21	similar	lilli	simba	himalayan	qicc
22	sms	salesh	fluffythey	inject	sms
23	tortoiseshel	steril	garfield	kati	third
24	urgent	thursday	kittensh	oldsh	bombay
25	advantag	vac	littler	preffer	exact
26	bluepoint	aprox	mom	soul	fluff
27	charm	bathurst	nearest	await	inlov
28	cream	befor	neg	barri	kiss
29	drysdal	betti	nicki	candi	saleh
30	dsh	boyging	oldth	cheer	tigger

Table A.12: Table of the 30 most probable words in Topics 16 to 20 for the LDA predictive model on the Gumtree corpus.

	Topic 21	Topic 22	Topic 23	Topic 24
1	cute	ginger	asap	cross
2	rough	shell	needl	ginger
3	sorri	tortois	farm	sex
4	catch	christma	dark	grey
5	teddi	feb	vac	freight
6	stuff	farm	shadow	patch
7	there	smokey	flee	cinnamon
8	american	british	hill	kit
9	grey	van	kiss	central
10	produc	chill	seven	birman
11	begin	countri	thankyou	urgent
12	blackbrown	detail	vacc	harri
13	cutest	pudpud	allerg	bombay
14	hey	dudley	pregnant	collect
15	star	luci	proven	indi
16	unknown	selkirk	adelaid	oldthey
17	exact	charcoal	ginger	pointskitten
18	fleed	choc	sphynx	realis
19	park	entir	anymor	shade
20	shini	needl	besid	shell
21	chinchilla	pregnant	chill	tiger
22	foodpick	present	firm	exot
23	geelong	tiffani	funni	happier
24	gender	tilli	lilli	michael
25	hill	underfoot	most	misha
26	homemal	afternoon	oldh	onlypleas
27	island	app	read	organ
28	jack	asap	tortoiseshel	selkirk
29	salethey	catson	ween	temper
30	soldno	chinchilla	alot	toowoomba

Table A.13: Table of the 30 most probable words in Topics 21 to 24 for the LDA predictive model on the Gumtree corpus.

	Topic 25	Topic 26
1	drink	manx
2	tortoiseshel	christma
3	rough	color
4	choc	longhair
5	rental	dark
6	sabrina	friday
7	munchkin	drink
8	anymor	cute
9	urgent	siberian
10	minut	approx
11	rex	maleal
12	femalex	panther
13	flat	aloud
14	snowsho	hey
15	nala	milli
16	sweetheart	newcastl
17	cheap	playful
18	cooki	ribbon
19	lighter	sempr
20	lynx	beautiful
21	mouser	defin
22	tortishel	fella
23	leo	immunis
24	middl	med
25	rene	msg
26	stripe	oldon
27	allerg	pickup
28	bun	shed
29	decemb	sheldon
30	devon	tower

Table A.14: Table of the 30 most probable words in Topics 25 to 26 for the LDA predictive model on the Gumtree corpus.

A.4 LDA step-up regression model coefficients

Table A.15 shows the coefficients for the LDA step-up regression model on the Gumtree corpus, as found in Section 3.5.

	Estimate	Std. Error	z value	$P(>\|z\|)$
(Intercept)	-0.6063	0.7178	-0.84	0.3983
Topic 7	41.5124	3.2532	12.76	0.0000
Topic 11	32.6879	3.2182	10.16	0.0000
Topic 20	-15.9501	2.2315	-7.15	0.0000
Topic 17	-15.3193	2.2417	-6.83	0.0000
Topic 12	-21.4743	2.8699	-7.48	0.0000
Topic 5	-17.1792	2.3546	-7.30	0.0000
Topic 24	21.0427	2.8110	7.49	0.0000
Topic 25	18.3314	2.5340	7.23	0.0000
Topic 4	10.9564	2.5643	4.27	0.0000
Topic 15	12.3711	2.8166	4.39	0.0000
Topic 13	11.4562	2.6915	4.26	0.0000
Topic 16	-15.1081	2.5054	-6.03	0.0000
Topic 3	8.5677	2.6090	3.28	0.0010
Topic 1	-11.9828	2.4187	-4.95	0.0000
Topic 26	-13.5158	2.9439	-4.59	0.0000
Topic 10	-8.5033	2.3658	-3.59	0.0003
Topic 6	-6.7162	2.1502	-3.12	0.0018
Topic 18	-7.5236	2.7026	-2.78	0.0054
Topic 23	-5.5031	2.8258	-1.95	0.0515
Topic 19	4.8047	2.6129	1.84	0.0659

Table A.15: Table of the model coefficients for the LDA step-up regression model on the Gumtree corpus.

A.5 sLDA (2 topics): topics and regression model coefficients

Table A.16 shows the coefficients of the sLDA regression model on the Gumtree corpus, as found in Section 3.6.2. Table A.17 shows the 30 most frequent words in both of the topics of this model.

	Estimate	Std. Error	z value	$P(>\lvert z\rvert)$
Topic 1	-2.5581	0.0825	-30.9961	0.0000
Topic 2	3.3608	0.1045	32.1635	0.0000

Table A.16: Table of the model coefficients for the sLDA regression model with two topics on the Gumtree corpus.

	Topic 1	Topic 2
1	kitten	cat
2	week	love
3	male	home
4	femal	old
5	worm	need
6	home	will
7	vaccin	good
8	will	desex
9	microchip	pleas
10	click	year
11	white	like
12	readi	go
13	vet	look
14	reveal	can
15	check	give
16	litter	get
17	black	play
18	train	new
19	old	litter
20	x	month
21	avail	beauti
22	ragdol	dog
23		friend
24	regist	come
25	one	microchip
26	blue	time
27	pleas	famili
28	boy	littl
29	new	just
30	girl	food

Table A.17: Table of the 30 most probable words in the topics for the sLDA model with two topics on the Gumtree corpus.

A.6 sLDA (26 topics): topics and regression model coefficients

Table A.18 shows the coefficients of the sLDA regression model with 26 topics on the Gumtree corpus, as found in Section 3.7. Tables A.19 to A.24 show the 30 most frequent words in each of the topics of this model.

	Estimate	Std. Error	z value	$P(>\|z\|)$
Topic 1	-4.9713	0.6914	-7.1906	0.0000
Topic 2	-3.4367	0.9766	-3.5189	0.0004
Topic 3	0.5843	0.9047	0.6458	0.5184
Topic 4	9.5811	0.9354	10.2422	0.0000
Topic 5	0.8552	0.9012	0.9489	0.3427
Topic 6	-10.4523	1.0143	-10.3047	0.0000
Topic 7	-6.3810	0.6982	-9.1386	0.0000
Topic 8	-6.8311	0.6476	-10.5488	0.0000
Topic 9	-1.3885	0.7151	-1.9417	0.0522
Topic 10	1.4404	0.6130	2.3499	0.0188
Topic 11	9.1393	0.9061	10.0866	0.0000
Topic 12	-11.3784	0.9584	-11.8725	0.0000
Topic 13	0.3245	0.4482	0.7240	0.4691
Topic 14	-0.1769	0.5864	-0.3017	0.7629
Topic 15	16.6795	1.2071	13.8173	0.0000
Topic 16	10.1753	1.0128	10.0468	0.0000
Topic 17	9.3029	1.0331	9.0049	0.0000
Topic 18	7.9644	0.5681	14.0206	0.0000
Topic 19	1.1429	0.8496	1.3453	0.1785
Topic 20	0.5519	0.6779	0.8142	0.4155
Topic 21	-7.3745	0.7649	-9.6414	0.0000
Topic 22	-6.1083	0.6738	-9.0660	0.0000
Topic 23	-6.4877	0.8804	-7.3694	0.0000
Topic 24	-7.6773	0.9509	-8.0741	0.0000
Topic 25	1.3250	0.8556	1.5487	0.1215
Topic 26	14.7852	1.2855	11.5018	0.0000

Table A.18: Table of the model coefficients for the sLDA regression model with 26 topics on the Gumtree corpus.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	ragdol	kitten	home	cat	will
2	blue	will	famili	year	get
3	point	home	love	home	like
4	seal	affection	new	love	well
5	male	happi	forev	old	just
6	check	colour	can	outsid	back
7	readi	burmes	go	insid	love
8	femal	now	us	kid	can
9	worm	care	de	desex	want
10	vet	email	sex	move	know
11	microchip	vet	great	hous	take
12	pure	chip	make	go	dog
13	chocol	siames	pet	month	cat
14	purebr	arrang	take	get	heart
15	now	healthi	friend	keep	realli
16	vaccin	natur	come	need	also
17	lilac	certif	meet	good	look
18	litter	mani	lot	around	give
19	mit	pet	look	prefer	go
20	russian	litter	question	attent	alway
21	torti	famili	member	children	care
22	sale	march	pleas	older	much
23	bi	includ	household	anim	use
24	will	feed	peopl	suit	peopl
25	avail	need	use	sinc	someon
26	father	deliveri	happi	sad	person
27	parent	photo	kitten	coupl	find
28	bicolour	veri	feel	cant	around
29	dad	sold	fit	realli	human
30	kitten	bred	healthi	dog	sit

Table A.19: Table of the 30 most probable words in Topics 1 to 5 for the sLDA model with 26 topics on the Gumtree corpus.

	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
1	kitten	regist	train	one	litter
2	mum	breeder	toilet	two	food
3	will	kitten	worm	i	come
4	week	avail	week	go	tray
5	mother	pedigre	flea	left	kitti
6	readi	vaccin	litter	kitten	toy
7	train	rais	treat	last	scratch
8	first	paper	play	togeth	bed
9	litter	pack	eat	three	bowl
10	worm	check	go	good	use
11	microchip	microchip	old	pick	post
12	dad	bengal	well	week	old
13	avail	inform	readi	u	box
14	can	pet	food	play	great
15	view	health	kitten	old	play
16	info	com	rag	must	sell
17	tonkines	brown	kitti	four	also
18	ask	desex	flead	near	water
19	go	includ	doll	person	beauti
20	new	www	wet	gone	microchip
21	done	vet	children	look	includ
22	owner	breed	home	want	train
23	girl	australia	solid	need	treatment
24	current	show	dri	can	dri
25	price	royal	forev	brother	flea
26	etc	facebook	day	friend	bag
27	eye	council	socialis	children	sale
28	pic	micro	born	free	love
29	vaccin	qualiti	end	alreadi	friend
30	complet	us	i	sell	collar

Table A.20: Table of the 30 most probable words in Topics 6 to 10 for the sLDA model with 26 topics on the Gumtree corpus.

	Topic 11	Topic 12	Topic 13	Topic 14	Topic 15
1	cat	kitten	white	reveal	give
2	love	will	black	click	home
3	need	week	grey	pleas	away
4	year	babi	ginger	call	adopt
5	old	photo	tabbi	text	pleas
6	home	home	hair	email	month
7	beauti	vaccin	short	interest	free
8	desex	readi	one	phone	desex
9	around	leav	domest	contact	beauti
10	great	first	fluffi	can	hous
11	dog	contact	colour	number	work
12	kid	litter	femal	messag	love
13	go	worm	week	com	rescu
14	due	age	kitten	send	forev
15	peopl	avail	long	kitten	interest
16	move	free	mark	sms	feel
17	friend	main	mother	area	old
18	affection	gorgeous	eat	inform	anim
19	hi	microchip	light	answer	foster
20	live	boy	dark	i	found
21	daughter	go	stripe	pm	still
22	shell	inform	brown	detail	m
23	night	beauti	orang	txt	fee
24	yr	rais	friend	british	along
25	rehom	also	drink	leav	help
26	companion	alreadi	paw	anytim	full
27	offer	vet	beauti	see	given
28	take	pleas	tortoiseshel	femal	famili
29	im	now	boy	you	look
30	urgent	via	old	shorthair	around

Table A.21: Table of the 30 most probable words in Topics 11 to 15 for the sLDA model with 26 topics on the Gumtree corpus.

	Topic 16	Topic 17	Topic 18	Topic 19	Topic 20
1	need	cat	old	love	girl
2	home	indoor	good	look	boy
3	just	new	home	home	microchip
4	want	love	free	time	vaccin
5	cat	desex	week	pleas	desex
6	due	owner	male	contact	worm
7	old	live	femal	new	beauti
8	unfortun	keep	month	call	natur
9	find	outdoor	kitten	also	come
10	us	can	pick	healthi	vet
11	young	name	need	care	i
12	hi	move	text	extrem	affection
13	year	take	messag	littl	check
14	rehom	also	cute	thank	now
15	love	dog	go	get	date
16	sell	get	train	cat	n
17	pleas	much	locat	soon	we
18	sure	longer	friend	best	look
19	new	look	onli	just	kid
20	name	small	toilet	waster	view
21	around	abl	giveaway	attent	old
22	circumst	month	thank	pet	pic
23	age	children	look	famili	moment
24	sister	chang	love	friend	adult
25	along	home	year	environ	christma
26	health	due	natur	come	x
27	asap	good	readi	sweet	park
28	desex	fulli	rough	togeth	quick
29	good	pet	soon	asap	jan
30	move	microchip	us	messag	have

Table A.22: Table of the 30 most probable words in Topics 16 to 20 for the sLDA model with 26 topics on the Gumtree corpus.

	Topic 21	Topic 22	Topic 23	Topic 24
1	kitten	male	vaccin	kitten
2	beauti	femal	worm	vet
3	play	x	chip	new
4	littl	kitten	flea	price
5	gorgeous	sale	vet	check
6	ador	th	micro	go
7	cute	persian	fulli	home
8	friend	readi	will	come
9	eye	born	check	view
10	sale	microchip	treat	coat
11	cudd	tabbi	train	will
12	thank	sold	breed	well
13	cross	pic	person	colour
14	well	avail	dog	enquiri
15	natur	silver	tail	parent
16	male	rais	affection	cost
17	text	pictur	manx	also
18	readi	week	companion	ring
19	call	fold	kitten	we
20	left	handl	time	detail
21	last	worm	new	stun
22	mum	scottish	great	genuin
23	new	march	natur	txt
24	fur	locat	rex	children
25	half	nd	excel	see
26	fluffi	ear	social	show
27	sweet	second	fun	owner
28	interest	exot	brought	care
29	absolut	request	d	high
30	may	vaccin	gentl	welcom

Table A.23: Table of the 30 most probable words in Topics 21 to 24 for the sLDA model with 26 topics on the Gumtree corpus.

	Topic 25	Topic 26
1	love	cat
2	like	will
3	s	like
4	littl	life
5	cuddl	pet
6	can	hous
7	cat	sad
8	play	adopt
9	will	pleas
10	lot	friend
11	find	day
12	meet	part
13	give	sleep
14	long	quit
15	lap	keep
16	make	pat
17	attent	work
18	affection	prefer
19	happi	give
20	perfect	see
21	big	time
22	quiet	shi
23	someon	desex
24	see	know
25	enjoy	rescu
26	hope	away
27	this	need
28	time	settl
29	place	indoor
30	hard	safe

Table A.24: Table of the 30 most probable words in Topics 25 to 26 for the sLDA model with 26 topics on the Gumtree corpus.

Appendix B

Love Actually corpus

B.1 Scene classifications

Tables B.1 to B.3 show the classification of each of the 79 scenes of *Love Actually* into the ten storylines, as discussed in Section 4.2.

Scene	Description	Classification
1	Introductory airport scene	Airport
2	Recording studio	Christmas single
3	Sick wife	Portuguese romance
4	Phone call (Karen and Daniel)	Child in love
5	Colin delivers sandwiches	America
6	Jack and Judy meet	Film stand-ins
7	Discussion with best man	Cue cards
8	Prime Minister meets staff	Prime Minister
9	Marriage	Cue cards
10	Brother sleeping with wife	Portuguese romance
11	Colin and Nancy the caterer	America
12	Colin's plan	America
13	Traffic discussion	Film stand-ins
14	Funeral	Child in love
15	Mark and Sarah talk at wedding	Cue cards
16	Sarah and Harry discuss Sarah's feelings	Office romance
17	Radio interview	Christmas single
18	First cabinet meeting	Prime Minister
19	Natalie brings biscuits	Prime Minister
20	Jack and Judy discuss Prime Minister	Film stand-ins
21	Colin has bought plane tickets	America
22	Organising office Christmas party	Adultery
23	Karen and Daniel discuss Sam	Child in love
24	Sam confesses he is in love	Child in love
25	Sarah and Karl say good night	Office romance
26	Jamie arrives in France	Portuguese romance
27	Natalie discusses ex-boyfriend	Prime Minister
28	Sam describes Joanna	Child in love
29	Ant and Dec television appearance	Christmas single
30	Mark and Juliet phone call	Cue cards

Table B.1: Scene classifications of the *Love Actually* corpus, with a short description of each scene.

Scene	Description	Classification
31	Mia proposes art gallery location	Adultery
32	Jamie and Aurelia meet	Portuguese romance
33	Jamie drives Aurelia home (first day)	Portuguese romance
34	Prime Minister meets President	Prime Minister
35	Press conference	Prime Minister
36	Harry and Karen discuss Joni Mitchell	Adultery
37	Prime Minister dances	Prime Minister
38	Jamie and Aurelia search for phone	Portuguese romance
39	Book falls in pond	Portuguese romance
40	Juliet sees wedding video	Cue cards
41	Mark leaves house	Cue cards
42	Prime Minister redistributes Natalie	Prime Minister
43	Watching Titanic	Child in love
44	Prime Minister receives paperwork	Prime Minister
45	Saying goodbye	Portuguese romance
46	Sam comes up with plan	Child in love
47	Harry and Mia dance	Adultery
48	Parkinson appearance	Christmas single
49	Sarah and Karl dance	Office romance
50	Sarah and Karl at Sarah's place	Office romance
51	Harry and Karen go to bed	Adultery
52	Sarah visits brother	Office romance
53	Harry buys a necklace	Adultery
54	Colin rents his place out	America
55	Jack asks Judy out	Film stand-ins
56	Christmas play rehearsal	Adultery
57	Colin goes to a bar	America
58	Karen gets Joni Mitchell CD	Adultery
59	Daniel and Sam discuss Claudia Schiffer	Child in love
60	Billy Mack is number one	Christmas single

Table B.2: Scene classifications of the *Love Actually* corpus, with a short description of each scene.

Scene	Description	Classification
61	Jack and Judy kiss	Film stand-ins
62	Jamie comes home and leaves	Portuguese romance
63	Sarah calls her brother	Office romance
64	Sam is not hungry	Child in love
65	Cue cards	Cue cards
66	Billy Mack talks to manager	Christmas single
67	Christmas card and door knocking	Prime Minister
68	Prime Minister meets Natalie's family	Prime Minister
69	Prime Minister goes backstage	Prime Minister
70	Karen and Prime Minister run into each other	Prime Minister
71	*All I want for Christmas is you* performance	Child in love
72	Karen confronts Harry	Adultery
73	Daniel meets Carol	Child in love
74	Jamie meets Aurelia's family	Portuguese romance
75	Sam runs through security	Child in love
76	Walking to restaurant	Portuguese romance
77	Sam catches Joanna	Child in love
78	Jamie proposes	Portuguese romance
79	Final airport scene	Airport

Table B.3: Scene classifications of the *Love Actually* corpus, with a short description of each scene.

B.2 LDA: topics and regression model coefficients

Tables B.1 to B.2 show the coefficients of the LDA regression model with 16 topics on the *Love Actually* corpus, as found in Section 4.4.1. Tables B.4 to B.7 show the 30 most frequent words in each of the topics of this model.

	(Intercept)	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
Christmas single	-21.82	-183.48	-270.47	353.27	208.78	-57.39	814.00	-89.39	-295.91	-108.03
Portuguese romance	5.78	-122.29	-168.91	303.40	-223.48	113.30	-32.71	-3.60	-21.09	64.85
Child in love	5.68	174.55	-30.47	4.54	-165.83	304.53	-126.52	320.79	-396.32	-386.77
America	1.25	-139.60	151.54	158.86	-398.22	193.33	-209.75	134.67	-59.36	53.47
Film stand-ins	-13.01	-303.37	290.67	-346.31	-253.46	-20.85	31.27	125.55	322.40	454.52
Cue cards	-10.85	243.60	267.40	-427.58	160.92	288.29	-34.27	-33.04	202.45	-499.30
Prime Minister	-1.19	73.07	281.43	-269.11	-31.62	-184.42	164.45	-208.32	-50.67	440.04
Office romance	-2.49	256.78	111.16	309.59	-148.72	-471.69	43.39	-116.09	3.63	371.34
Adultery	-0.80	190.12	-297.63	52.96	129.77	20.95	-512.95	-199.98	209.59	15.32

Figure B.1: Table of coefficients for Topics 1 to 9 of the LDA regression model with 16 topics on the *Love Actually* corpus. Significance levels: 0 ‘***’, 0.001 ‘**’, 0.01 ‘*’, 0.05 ‘.’, 0.1 ‘ ’.

	Topic 10	Topic 11	Topic 12	Topic 13	Topic 14	Topic 15	Topic 16
Christmas single	-516.99	10.17	-38.76	148.35	93.34	13.63	-102.93
Portuguese romance	-138.72	-218.92	57.90	-63.11	541.77	-86.64	4.03
Child in love	-223.69	660.99	117.60	72.36	-73.03	-256.86	9.81
America	-23.76	671.34	-120.77	-181.71	-188.69	-90.78	50.67
Film stand-ins	-95.32	-248.74	-79.35	-187.30	216.10	-31.93	113.11
Cue cards	-35.93	-201.60	144.78	311.74	-146.12	55.93	-308.13
Prime Minister	-162.03	-334.99	-30.58	136.07	-83.97	227.10	32.35
Office romance	-1.58	-256.89	24.27	-66.00	62.42	309.87	-433.97
Adultery	-305.99	284.40	113.11	182.47	106.14	69.14	-58.22

Figure B.2: Table of coefficients for Topics 10 to 16 of the LDA regression model with 16 topics on the *Love Actually* corpus. Significance levels: 0 ‘***’, 0.001 ‘**’, 0.01 ‘*’, 0.05 ‘.’, 0.1 ‘ ’.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	baby	yeah	darling	love	want
2	bye	know	one	can	just
3	absolutely	just	good	feel	one
4	right	yes	first	song	make
5	girl	even	answer	nothing	now
6	goodbye	give	gone	done	know
7	end	get	marry	need	ever
8	suppose	mum	maybe	say	lot
9	world	think	really	feeling	let
10	party	well	bit	take	great
11	know	now	life	toes	year
12	presents	thing	man	back	never
13	wanted	able	right	old	away
14	around	anything	see	around	day
15	boyfriend	best	though	fingers	help
16	called	never	cos	marriage	wish
17	car	always	door	sleep	bring
18	everything	comes	either	easy	girl
19	will	end	elton	every	listen
20	american	hold	every	everywhere	airport
21	apart	morning	friends	golden	enough
22	asked	phone	jesus	merry	fade
23	book	pie	little	news	get
24	call	problem	merry	relationship	hope
25	fat	thanks	mine	remember	joni
26	kill	almost	realized	school	pocket
27	late	eve	thing	son	real
28	might	free	tonight	wife	song
29	play	guys	yeah	cock	true
30	pretty	huh	asking	erm	unless

Table B.4: Table of the 30 most probable words in Topics 1 to 5 of the LDA model with 16 topics on the *Love Actually* corpus.

	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
1	christmas	always	night	right	god
2	number	course	man	sir	knows
3	one	going	thank	yes	without
4	uncle	gonna	alone	sorry	dad
5	new	ten	can	erm	night
6	everyone	think	find	hello	two
7	real	cold	lovely	well	fine
8	crap	wait	around	thank	wrong
9	great	bit	got	come	england
10	important	check	necklace	much	good
11	made	give	come	please	now
12	manager	thing	later	prime	trouble
13	grande	till	nobody	see	coming
14	just	absolutely	particularly	fine	gorgeous
15	live	although	brilliant	sure	round
16	moment	hell	half	said	sex
17	now	someone	let	one	along
18	pay	way	place	better	blind
19	record	work	please	actually	boa
20	bueno	boring	sorry	hope	bright
21	bugger	classic	understand	way	business
22	came	country	yes	dodgy	come
23	dead	dear	baby	last	englishman
24	enjoy	enough	call	live	fool
25	fool	favour	definitely	second	holy
26	fucking	goodness	english	sister	imagine
27	late	happy	far	street	inside
28	mike	lighting	game	busy	last
29	mitchell	looks	heart	look	meet
30	saw	married	heaven	must	must

Table B.5: Table of the 30 most probable words in Topics 6 to 10 of the LDA model with 16 topics on the *Love Actually* corpus.

	Topic 11	Topic 12	Topic 13	Topic 14
1	well	jewellery	look	portuguese
2	back	something	come	time
3	little	erm	really	just
4	yeah	thought	just	better
5	got	need	know	fuck
6	hey	can	yes	day
7	big	will	life	kind
8	actually	going	president	stop
9	america	christmas	cute	excellent
10	bar	never	line	shit
11	good	quite	say	years
12	trust	want	sure	bad
13	one	just	will	hate
14	total	tell	actually	naked
15	woman	get	bad	sometimes
16	chance	looking	friend	work
17	girls	see	going	babe
18	hold	leave	great	bloody
19	see	pop	now	concert
20	song	put	dark	crime
21	wow	really	english	eels
22	beautiful	arse	show	father
23	competition	bag	way	hell
24	course	box	ask	hello
25	deal	finished	course	miss
26	deep	resting	don	nice
27	happen	three	every	pretty
28	keep	truth	fact	scary
29	knowing	yes	may	today
30	leo	anything	minutes	become

Table B.6: Table of the 30 most probable words in Topics 11 to 14 of the LDA model with 16 topics on the *Love Actually* corpus.

	Topic 15	Topic 16
1	like	think
2	good	mean
3	thanks	got
4	minister	meet
5	just	girls
6	believe	talk
7	waiting	get
8	home	good
9	turn	around
10	ask	bonjour
11	away	nice
12	blue	america
13	guys	american
14	jump	house
15	lobster	perfect
16	much	question
17	people	actually
18	true	art
19	dance	big
20	gonna	british
21	left	brother
22	long	bullied
23	boss	buy
24	catering	calls
25	cool	careful
26	fine	chocolate
27	gay	course
28	hear	epiphany
29	laughs	felt
30	looks	going

Table B.7: Table of the 30 most probable words in Topics 15 to 16 of the LDA model with 16 topics on the *Love Actually* corpus.

B.3 HMTM: topics and regression model coefficients

Tables B.3 to B.4 show the coefficients of the HMTM regression model with 16 topics on the *Love Actually* corpus, as found in Section 4.5. Tables B.8 to B.10 show the 30 most frequent words in each of the topics of this model.

The presence of words such as *s*, *t* and *ve* in the topics (that are not present in previous models) is due to the fact we have not performed stemming on the data for this model. Therefore, these grammatical morphemes have been retained in the corpus.

	(Intercept)	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Christmas single	-24.59	-26.10	95.02	149.00	-127.52	-170.60
Portuguese romance	-1.15	-95.03	89.84	59.29	-5.39	68.63
Child in love	8.81	-68.34	-21.85	53.59	-10.82	16.55
America	-22.26	47.56	85.34	-64.00	47.67	77.33
Film stand-ins	-10.14	-18.84	-87.09	34.39	102.09	65.61
Cue cards	-19.93	171.08	-19.62	-53.77	37.15	120.34
Prime Minister	-1.56	-75.60	-105.01	132.61	-46.62	37.41
Office romance	-2.32	-76.84	-5.83	86.00	11.74	-11.56
Adultery	7.95	-54.64	-19.45	27.57	6.73	27.12

Figure B.3: Table of coefficients for Topics 1 to 5 of the HMTM regression model with 12 topics on the *Love Actually* corpus. Significance levels: 0 '***', 0.001 '**', 0.01 '*', 0.05 '.', 0.1 ' '.

	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12
Christmas single	-187.40	77.32	270.70	43.37	-75.10	-178.05	104.76
Portuguese romance	-131.73	125.39	-104.90	-150.84	6.76	58.77	78.06
Child in love	34.58	-11.82	-48.77	14.18	-40.87	57.94	34.44
America	-48.12	157.19	28.40	-199.92	91.75	8.04	-253.50
Film stand-ins	-3.96	15.80	-81.38	110.37	-149.82	129.03	-126.34
Cue cards	193.05	-58.89	-161.29	-62.04	-85.05	-234.77	133.88
Prime Minister	-103.97	12.22	4.02	96.85	-45.09	39.12	52.49
Office romance	51.46	-40.33	15.81	154.95	-102.27	-8.76	-76.68
Adultery	28.47	-26.40	-24.88	46.46	-63.59	31.02	29.54

Figure B.4: Table of coefficients for Topics 6 to 12 of the HMTM regression model with 12 topics on the *Love Actually* corpus. Significance levels: 0 '***', 0.001 '**', 0.01 '*', 0.05 '.', 0.1 ' '.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	just	portuguese	right	god	just
2	look	back	well	can	yes
3	sorry	better	yes	actually	thank
4	without	day	sir	nice	er
5	t	let	s	girl	erm
6	thing	number	ok	late	great
7	mean	way	er	will	jewellery
8	never	y	much	made	need
9	always	even	ah	wife	oh
10	little	first	absolutely	marry	something
11	really	real	thanks	beautiful	please
12	bit	away	thought	boyfriend	no
13	now	big	will	cos	girls
14	find	gonna	bad	goodness	might
15	lovely	never	sorry	play	leave
16	america	fuck	must	three	morning
17	m	last	president	best	put
18	pretty	stuff	two	check	guys
19	home	feeling	ha	hoping	stop
20	years	go	news	marriage	american
21	line	kind	say	sell	get
22	looking	job	tonight	wedding	give
23	name	school	blue	worse	trust
24	people	second	excellent	asking	almost
25	ve	try	huh	classic	heart
26	able	wow	left	coming	keep
27	lot	english	next	daughter	meet
28	ooh	important	tell	either	nobody
29	total	naked	wait	foot	pop
30	busy	definitely	wanted	given	scary

Table B.8: Table of the 30 most probable words in Topics 1 to 5 of the HMTM model with 12 topics on the *Love Actually* corpus.

	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
1	yeah	hello	christmas	good	know
2	night	one	like	s	knows
3	baby	prime	yeah	one	get
4	bye	hey	really	oh	feel
5	got	hi	around	well	go
6	man	hope	know	think	think
7	make	cute	year	us	ever
8	goodbye	hell	ask	fine	end
9	though	meet	lot	minister	maybe
10	turn	sex	old	sure	talk
11	long	welcome	anything	life	live
12	sleep	called	work	ll	round
13	call	england	world	say	shit
14	time	terrible	friend	time	wrong
15	even	bonjour	mind	tell	ten
16	today	lady	show	new	toes
17	case	luck	listen	dad	airport
18	christ	o	minutes	mum	fingers
19	full	worst	necklace	waiting	mine
20	gate	exactly	record	country	ok
21	jesus	happy	time	d	arse
22	mum	house	trouble	merry	everywhere
23	problem	je	bar	party	looked
24	run	press	brilliant	thank	street
25	shag	shut	dear	tv	yes
26	suppose	surprises	door	room	buy
27	woman	apart	fact	sister	dance
28	anyway	beckham	friends	calls	english
29	broken	boss	fucking	children	fat
30	bugger	change	pay	fool	ready

Table B.9: Table of the 30 most probable words in Topics 6 to 10 of the HMTM model with 12 topics on the *Love Actually* corpus.

	Topic 11	Topic 12
1	right	love
2	oh	come
3	want	song
4	going	take
5	now	give
6	see	can
7	course	every
8	erm	nothing
9	darling	uncle
10	said	alone
11	hi	done
12	girl	true
13	moment	believe
14	answer	hold
15	everyone	look
16	gone	l
17	help	wish
18	lobster	joni
19	dodgy	jump
20	hate	wants
21	bring	car
22	getting	dark
23	relationship	dj
24	seems	mitchell
25	understand	things
26	cool	boring
27	final	else
28	high	hang
29	instance	move
30	okay	still

Table B.10: Table of the 30 most probable words in Topics 11 and 12 of the HMTM model with 12 topics on the *Love Actually* corpus.

B.4 Persistent HMTM: topics and regression model coefficients

Table B.11 shows the coefficients of the persistent HMTM regression model with 3 topics on the *Love Actually* corpus, as found in Section 4.6. Tables B.12 shows the 30 most frequent words in each of the topics of this model.

Storyline	(Intercept)	Topic 1	Topic 2	Topic 3
Christmas single	1.32	-3.29	2.07	2.54
Portuguese romance	0.93	-11.94**	2.65	10.22*
Child in love	1.89*	-5.37	0.40	6.86
America	0.11	-13.80**	3.89	10.02*
Film stand-ins	0.55	0.81	-0.39	0.13
Cue cards	1.38	-2.56	2.21	1.73
Prime Minister	1.97*	-4.73	1.96	4.75
Office romance	0.81	$-6.81^{\cdot}$	-1.91	9.54*
Adultery	1.57·	-2.37	1.22	2.73

Table B.11: Table of coefficients for the persistent HMTM regression model with 3 topics on the *Love Actually* corpus. Significance levels: 0 ‘***’, 0.001 ‘**’, 0.01 ‘*’, 0.05 ‘.’, 0.1 ‘ ’.

	Topic 1	Topic 2	Topic 3
1	love	just	s
2	just	yes	oh
3	oh	erm	er
4	god	want	right
5	christmas	one	well
6	right	christmas	know
7	knows	sir	hello
8	now	ok	like
9	yeah	come	yeah
10	thank	never	baby
11	come	good	sorry
12	good	like	bye
13	know	around	think
14	can	jewellery	us
15	one	will	man
16	well	back	ah
17	without	say	hi
18	great	much	yes
19	need	better	actually
20	course	go	day
21	got	really	look
22	look	sure	darling
23	night	think	god
24	t	going	going
25	something	portuguese	life
26	feel	s	little
27	get	let	song
28	give	lot	way
29	time	make	y
30	minister	can	end

Table B.12: Table of the the 30 most probable words in the topics of the persistent HMTM with 3 topics on the *Love Actually* corpus.

Bibliography

[1] RSPCA report on animal outcomes from our shelters, care and adoption centres: 2015 - 2016. Technical report, RSPCA, 2016. Available at *https://www.rspca.org.au/facts/annual-statistics-2015-16*.

[2] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[3] Mark Andrews. Mark Andrews: code. 2016. Available at *http://www.mjandrews.net/code/*. Last checked 6 September 2017.

[4] Mark Andrews and Gabriella Vigliocco. The hidden Markov topic model: A probabilistic model of semantic representation. *Topics in Cognitive Science*, 2(1):101–113, 2010.

[5] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 27–34. AUAI Press, 2009.

[6] Jane Austen. *Pride and Prejudice.* T. Egerton, Whitehall, 1813.

[7] Leonard E Baum and J A Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363, 1967.

[8] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[9] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[10] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120. ACM, 2006.

[11] David M Blei and John D Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, pages 17–35, 2007.

[12] David M Blei and Jon D McAuliffe. Supervised topic models. In *Advances in Neural Information Processing Systems*, pages 121–128, 2008.

[13] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[14] Jordan L Boyd-Graber and David M Blei. Syntactic topic models. In *Advances in Neural Information Processing Systems*, pages 185–192, 2009.

[15] Glenn W Brier and Roger A Allen. Verification of weather forecasts. In *Compendium of Meteorology*, pages 841–848. Springer, 1951.

[16] George Casella and Edward I George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[17] Jonathan Chang. *lda: Collapsed Gibbs Sampling Methods for Topic Models*, 2015. R package version 1.4.2. Available at *https://CRAN.R-project.org/package=lda.*

[18] Richard Curtis. *Love Actually.* Universal Studios and StudioCanal, London, 2003.

[19] Ali Daud, Juanzi Li, Lizhu Zhou, and Faqir Muhammad. Knowledge discovery through directed probabilistic topic models: a survey. *Frontiers of Computer Science in China*, 4(2):280–301, 2010.

[20] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[21] Seymour Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328, 1975.

[22] Timothy Graham and Robert Ackland. Topic modeling of tweets in R: A tutorial and methodology. 2015.

[23] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[24] Thomas L Griffiths, Mark Steyvers, David M Blei, and Joshua B Tenenbaum. Integrating topics and syntax. In *Advances in Neural Information Processing Systems*, pages 537–544, 2005.

[25] Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. Hidden topic Markov models. In *Artificial Intelligence and Statistics*, pages 163–170, 2007.

[26] Bettina Grün and Kurt Hornik. topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, 40(13):1–30, 2011.

[27] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, second edition, 2009.

[28] Walt Hickey and Gus Wezerek. The definitive analysis of 'Love Actually', the greatest Christmas movie of our time. *FiveThirtyEight*, December 2016. Available at *https://fivethirtyeight.com/features/the-definitive-analysis-of-love-actually-the-greatest-christmas-movie-of-our-time/*. Last checked 15 December 2017.

[29] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM, 1999.

[30] John Hutchins. From first conception to first demonstration: the nascent years of machine translation, 1947–1954. a chronology. *Machine Translation*, 12(3):195–252, 1997.

[31] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent Dirichlet allocation for tag recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*, pages 61–68. ACM, 2009.

[32] Simon Lacoste-Julien, Fei Sha, and Michael I Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *Advances in Neural Information Processing Systems*, pages 897–904, 2009.

[33] Jiwei Li, Claire Cardie, and Sujian Li. TopicSpam: A topic-model based approach for spam detection. In *51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, volume 2, pages 217–221, 2013.

[34] Ximing Li, Jihong Ouyang, and Xiaotang Zhou. Supervised topic models for multi-label classification. *Neurocomputing*, 149:811–819, 2015.

[35] David McClure. A hierarchical cluster of words across narrative time. July 2017. Available at *http://litlab.stanford.edu/hierarchical-cluster-across-narrative-time/*. Last checked 19 December 2017.

[36] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *International Conference on Learning Representations Workshop*, 2013.

[37] Lucy Maud Montgomery. *Anne of Green Gables*. L.C. Page & Co., 1908.

[38] James R Norris. *Markov chains.* Cambridge University Press, 1998.

[39] Christos H Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 159–168. ACM, 1998.

[40] Pietro Pinoli, Davide Chicco, and Marco Masseroli. Latent Dirichlet allocation based on Gibbs sampling for gene function prediction. In *Computational Intelligence in Bioinformatics and Computational Biology, 2014 IEEE Conference on*, pages 1–8. IEEE, 2014.

[41] Martin Ponweiser. *Latent Dirichlet allocation in R.* PhD thesis, WU Vienna University of Economics and Business, 2012.

[42] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed Gibbs sampling for latent Dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 569–577. ACM, 2008.

[43] Martin Porter. Snowball: A language for stemming algorithms. 2001. Available at *http://snowballstem.org*. Last checked 13 December 2017.

[44] Lawrence Rabiner and B Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[45] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 1, pages 248–256. Association for Computational Linguistics, 2009.

[46] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Pro-*

ceedings of the 20th Conference on Uncertainty in Artificial Intelligence, pages 487–494. AUAI Press, 2004.

[47] Timothy N Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. Statistical topic models for multi-label document classification. *Machine Learning*, 88(1):157–208, 2012.

[48] Dawei Shen. Some mathematics for HMM. *Massachusetts Institute of Technology*, 2008.

[49] Julia Silge and David Robinson. *Text Mining with R.* O'Reilly, 2017.

[50] Jian Tang, Zhaoshi Meng, Xuanlong Nguyen, Qiaozhu Mei, and Ming Zhang. Understanding the limiting factors of topic modeling via posterior contraction analysis. *Proceedings of The 31st International Conference on Machine Learning*, pages 190–198, 2014.

[51] Yee Whye Teh, David Newman, and Max Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *NIPS*, volume 6, pages 1378–1385, 2006.

[52] Jeffrey David Ullman. *Mining of massive datasets.* Cambridge University Press, 2011.

[53] Denis Valle, Benjamin Baiser, Christopher W Woodall, and Robin Chazdon. Decomposing biodiversity data using the latent Dirichlet allocation model, a probabilistic multivariate statistical method. *Ecology Letters*, 17(12):1591–1601, 2014.

[54] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S.* Springer, fourth edition, 2002.

[55] Xuerui Wang and Andrew McCallum. Topics over time: A non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 424–433. ACM, 2006.

[56] Daniel S. Wilks. *Statistical methods in the atmospheric sciences.* Academic Press, second edition, 2006.