A Gamma-Poisson topic model for short text

by

Jocelyn Rangarirai Mazarura

Submitted in partial fulfilment of the requirements for the degree

Philosophiae Doctor in Mathematical Statistics

In the Department of Statistics

In the Faculty of Natural and Agricultural Sciences

University of Pretoria

Pretoria

30 November 2020

I, *Jocelyn Rangarirai Mazarura*, declare that this thesis, which I hereby submit for the degree Philosophiae Doctor in Mathematical Statistics at the University of Pretoria, is my own work and has not previously been submitted by me for a degree at this or any other tertiary institution.

Signature: 

Date: 11/02/2021

ABSTRACT

Most topic models are constructed under the assumption that documents follow a multinomial distribution. The Poisson distribution is an alternative distribution to describe the probability of count data. For topic modelling, the Poisson distribution describes the number of occurrences of a word in documents of fixed length. The Poisson distribution has been successfully applied in text classification, but its application to topic modelling is not well documented, specifically in the context of a generative probabilistic model. Furthermore, the few Poisson topic models in literature are admixture models, making the assumption that a document is generated from a mixture of topics.

In this study, we focus on short text. Many studies have shown that the simpler assumption of a mixture model fits short text better. With mixture models, as opposed to admixture models, the generative assumption is that a document is generated from a single topic. One topic model, which makes this one-topic-per-document assumption, is the Dirichlet-multinomial mixture model. The main contributions of this work are a new Gamma-Poisson mixture model, as well as a collapsed Gibbs sampler for the model. The benefit of the collapsed Gibbs sampler derivation is that the model is able to automatically select the number of topics contained in the corpus. The results show that the Gamma-Poisson mixture model performs better than the Dirichlet-multinomial mixture model at selecting the number of topics in labelled corpora. Furthermore, the Gamma-Poisson mixture produces better topic coherence scores than the Dirichlet-multinomial mixture model, thus making it a viable option for the challenging task of topic modelling of short text.

The application of GPM was then extended to a further real-world task: that of distinguishing between semantically similar and dissimilar texts. The objective was to determine whether GPM could produce semantic representations that allow the user to determine the relevance of new, unseen documents to a corpus of interest. The challenge of addressing this problem in short text

from small corpora was of key interest. Corpora of small size are not uncommon. For example, at the start of the Coronavirus pandemic limited research was available on the topic. Handling short text is not only challenging due to the sparsity of such text, but some corpora, such as chats between people, also tend to be noisy. The performance of GPM was compared to that of word2vec under these challenging conditions on labelled corpora. It was found that the GPM was able to produce better results based on accuracy, precision and recall in most cases. In addition, unlike word2vec, GPM was shown to be applicable on datasets that were unlabelled and a methodology for this was also presented. Finally, a relevance index metric was introduced. This relevance index translates the similarity distance between a corpus of interest and a test document to the probability of the test document to be semantically similar to the corpus of interest.

ACKNOWLEDGEMENTS

I would like to extend my deep and sincere gratitude to my supervisor, Dr A de Waal. I could not have asked for a better supervisor. She was a constant source of encouragement, enthusiasm and support. It was such a pleasure and honour to have been able to walk this journey with her.

I would also like to thank my co-supervisor, Prof P de Villiers, for his wise counsel, patience and guidance.

I would like to thank Prof A Bekker and the Department of Statistics at the University of Pretoria for affording me this research opportunity as well as the colleagues who supported and encouraged me. A special thanks to Dr H Strydom for the hard work and sacrifices she made for me, and to Dr I Fabris-Rotelli for the wisdom she imparted during the teas we shared.

I would also like to acknowledge the Centre of Artificial Intelligent Research (CAIR) and the SARChI Research Chair in Nonparametric, Robust Statistical Inference and Statistical Process Control for their financial support.

I would like to express my deepest appreciation to my family and friends for always believing in me and being my pillars of strength. I am sincerely grateful to my mom and dad for all their prayers, love and encouragement, as well as my loving husband who was always there to uplift me and lend an ear. Lastly, I would like to extend gratitude to my siblings, Grace, Esther and Michael, and my friends, Iketle, Brenda, Seite and Priyanka. I could not have made it through without any of them.

Above all, I would like to thank the Lord for carrying me throughout this entire journey and for extending His grace, love and mercy towards me. To Him be all the glory, honour and praise.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations

AJSD	Adjusted Jensen-Shannon distances
BERT	Bidirectional Encoder Representations from Transformers
BTM	Biterm topic model
CGP	Conditional Gamma-Poisson model
DMM	Dirichlet multinomial mixture
DSM	Distributional semantic model
DP-BMM	Dirichlet process biterm-based mixture model
DSTM	Dual sparse topic model
EM	Expectation-Maximisation (algorithm)
GaP	Gamma-Poisson (model)
GPM	Gamma-Poisson mixture (model)
GPU-DMM	Generalised Pólya urn Dirichlet multinomial mixture
GPU-PDMM	Generalised Pólya urn Poisson-based Dirichlet multinomial mixture
GSDMM	Gibbs sampler Dirichlet multinomial mixture
JSD	Jensen-Shannon distance
KL	Kullback-Leibler
LDA	Latent Dirichlet allocation
LF-DMM	Latent feature Dirichlet multinomial mixture
LTM	Latent topic mode
NMF	Non-negative matrix factorisation
PCA	Principal component analysis
PDM	Poisson decomposition model
PDMM	Poisson-based Dirichlet multinomial mixture

PSTR	Pattern set-based text representation
PTM	Pseudo-document-based topic model
SADTM	Self-aggregating dynamic topic model
SATM	Self-aggregation based topic model
STC	Sparse topical encoding
SPTM	Sparsity-enhanced pseudo-document-based topic model
VSM	Vector space model
WNTM	Word network topic model

CHAPTER ONE

INTRODUCTION

Topic modelling is a text mining technique used to uncover latent topics in large collections of documents. Unlike supervised methods, such as regression and classification, most topic models are able to draw topical information from documents that are unlabelled. Thus, such topic models fall into the class of unsupervised learning techniques along with methods such as clustering and dimensionality reduction. This means that a collection of documents can be analysed without having any prior knowledge regarding what they may be about.

Traditional topic models have a proven history of success on long documents, such as news articles and e-books. However, due to the increasing popularity of micro-blogging websites, social media platforms and online shopping (which involves product reviews), text that is significantly shorter has become increasingly relevant. Such sources of text potentially hold valuable information that can be useful in many applications, such as event tracking (Lin et al., 2010), interest profiling (Weng et al., 2010) and product recommendation (Zhang and Piramuthu, 2018).

1.1 MOTIVATION

Traditional topic models infer topics based on word co-occurrence relationships between words (Yan et al., 2013). In order to extract meaningful topics, a topic model must successfully infer

these relationships from a corpus. Per definition, short text contains few words and consequently tends to contain less co-occurrence information than long text. This in turn has a negative impact on the performance of traditional topic models and has created a need for topic models that are able to overcome the challenges associated with topic modelling of short text.

Most topic models are constructed under the assumption that documents follow a multinomial distribution. The Poisson distribution is an alternative distribution to describe the probability of count data. It has been successfully applied in text classification and has also been shown to outperform its multinomial equivalent (Ogura et al., 2013). Despite this, its application to topic modelling is not well documented, specifically in the context of a generative probabilistic model. Furthermore, the few Poisson topic models in literature are admixture models, making the assumption that a document is generated from a mixture of topics. Many studies have shown that the simpler assumption of a mixture model fits short text better. With mixture models, as opposed to admixture models, the generative assumption is that a document is generated from a single topic. One topic model, which makes this one-topic-per-document assumption, is the Dirichlet-multinomial mixture model (DMM) (Yin and Wang, 2014). However, this topic model is based on a multinomial distribution. In light of the mixture model's success on short text and the positive results obtained in other Poisson-based text mining tasks, the objective of this research is to study and derive new Poisson-based topic models for short text.

1.2 CONTRIBUTIONS OF THIS WORK

This thesis makes the following contributions.

1. It presents a unifying framework that describes the connection between topic models and other well-known statistical techniques. It highlights how topic models possess characteristics of both dimensionality reduction techniques and cluster analysis. A clear connection is made between clustering and principal components analysis, thus making it apparent how topic models are related.

2. This thesis introduces a new topic model for short text that has not been proposed before in the literature, the Gamma Poisson mixture model (GPM). The GPM is a modification of the Gibbs Sampling Dirichlet multinomial mixture model (GSDMM) of Yin and Wang (2014). Instead of modelling text according to a multinomial distribution, as does GSDMM,

GPM changes this assumption and assumes a Poisson distribution instead. On the datasets considered, GPM was able to produce topics with better coherence scores than GSDMM.

3. It details the derivation of a collapsed Gibbs sampler for the estimation of the parameters of the new topic model. Alternative estimation procedures such as the EM algorithm could have been used. However, it is this estimation procedure that gives GPM the favourable characteristic of being able to estimate the number of topics automatically.

4. Various experiments were conducted to investigate the characteristics of the new model and the results are documented in this thesis. GPM was found to produce stable results with small variances across different runs. Furthermore, unlike typical Gibbs samplers which may need long burn-in periods, collapsed Gibbs sampler for the GPM converged quickly. Just 15 iterations were suitable for the datasets that were used. Lastly, GPM not only produced better topic coherence scores than GSDMM, but the estimated number of clusters it found for the labelled corpora was closer to the true value than the estimate of the GSDMM in most cases.

5. In addition, an open source software package was produced based on this research. GPM is available as a Python package at `https://github.com/jrmazarura/GPM`. Moreover, the GSDMM has also been included in this package, thus making the application and comparison of these models easy and convenient.

6. This thesis also presents several experiments demonstrating the utility of the GPM in a real-world application: determining the relevance of new, unseen documents to a collection of documents of interest. The focus was on the challenging problem of handling short text from small corpora which may also be noisy. Small corpora are not uncommon as they can easily arise from emerging topics. For example, at the start of the Coronavirus pandemic limited research was available on the topic. Furthermore, short text is not only challenging due to the sparsity of such text, but some corpora, such as chats between people or comments, also tend to be noisy. The performance of GPM was compared to that of word2vec under these challenging conditions on labelled corpora. It was found that the GPM was able to produce better results based on accuracy, precision and recall in most cases. In order to compare results from the different models, a relevance index was also defined. Finally, a framework

was then presented showing how GPM can be successfully applied in the unsupervised context where documents were unlabelled, as is often the case in reality.

1.3 RESEARCH OUTPUT

Since starting with my doctoral studies, the following research output has been produced.
Conference presentations:

- 2016 South African Statistical Association (SASA) and awarded second prize in the 2016 SASA postgraduate paper competition, Capetown, South Africa
- 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference, Stellenbosch, South Africa
- 2019 International Symposium in Statistics and Biostatistics, Pretoria, South Africa

Conference proceedings:

- Mazarura, J., and De Waal, A. (2016). A comparison of the performance of latent Dirichlet allocation and the Dirichlet multinomial mixture model on short text. PRASA RobMech) (pp. 1-6). IEEE.
- Mazarura , J., de Waal, A., and de Villiers, P. (2019). Semantic representations for under resourced languages. Proceedings of the South African Institute of Computer Scientists and Information Technologists (SAICSIT) 2019 (pp. 1-10)
- Another paper titled, Probabilistic Distributional Semantic Methods for Small Unlabelled Text, has been accepted in the conference proceedings and will be presented early in 2021 at the Southern African Conference for Artificial Intelligence Research (SACAIR) 2020.

Journal publication:

- Mazarura, J., de Waal, A., and de Villiers, P. (2020). A Gamma Poisson Mixture Topic Model for Short Text. Mathematical Problems in Engineering, 2020.

1.4 CODES

The Python codes for the various experiments conducted in this research have also been made available.

- The Python package used in the experiments in Chapter 5 at `https://github.com/jrmazarura/GPM`.
- The codes to conduct the semantic similarity experiments from Chapter 6 are available at `https://github.com/jrmazarura/Similarity_Experiments`.

1.5 THESIS STRUCTURE

The thesis is structured as follows.

- Chapter 2 begins with an introduction to topic modelling and highlights some of the connections between topic models and other well-known statistical and mathematical methods. This is then followed by a study of the literature on topic modelling of short text.
- Chapter 3 presents a study of the use of the Poisson and multinomial distributions in text modelling. It also discusses some of the existing topic models based on these distributions. The objective of this section is to provide motivation for the new topic model proposed in this thesis.
- Chapter 4 presents the main contribution, the new Gamma-Poisson topic model for short text topic modelling. The derivation of the associated collapsed Gibbs sampler is also presented.
- Many experiments were conducted to evaluate the utility of the model and the results are presented and discussed in Chapter 5.
- Chapter 6 presents a further application of the new Gamma-Poisson topic model relating to semantic similarity in text.
- Finally, the conclusions and discussion of future work are presented in Chapter 7.

CHAPTER TWO

BACKGROUND

2.1 INTRODUCTION

Variables that are unobserved are often described as being *hidden* or *latent*. Models that try to leverage associations between observed variables and latent variables are called latent variable models, and these include models such as mixture models and techniques such as factor analysis (Murphy, 2012). In the context of unsupervised topic modelling, the topics covered in a corpus are not assumed to be known in advance. Topic models are considered latent variable models as they try to use the words in documents (observed variables) to infer the hidden topics (latent variables) contained in them.

The connection between topic modelling and other well-known statistical techniques, such as clustering and principal component analysis, is not always clear from the literature. Therefore, within the upcoming sections, a unifying framework that describes some of these relationships is discussed. The focus of this research is on the application of topic models on short text. In light of this, a study of the existing literature on short text is also presented. Before going into this discussion, some background on the typical topic modelling procedure is first described.

2.2 THE BASIC TOPIC MODEL

Topic models are statistical techniques for discovering topics within large collections of documents. By virtue of their being mathematical models, the first challenge is converting this unstructured and typically noisy textual data to a mathematically computable form. Most multivariate datasets can be represented as fixed-size vectors. In the case of a corpus with varying-length documents, one way of representing a document as a fixed-size vector is by simply counting the number of occurrences of each unique word (the vocabulary). Following this procedure transforms the corpus into a document-by-word matrix of word frequencies. This representation is often referred to as the *bag-of-words* representation and it implies that syntax and word order are completely ignored (Inouye et al., 2017). These document vectors typically lie in high-dimensional spaces. The vocabulary size (number of variables) is usually much larger than 1000, therefore motivating the need for multivariate count-valued distributions, which are able to capture the rich dependencies between variables.

In practice, a corpus can contain thousands of words, yet not all the words will occur in all the documents. Consequently, document-by-word matrices tend to be very large and sparse. In order to mitigate the dimensionality problem and improve the performance of such models there are some common pre-processing strategies that are often applied prior to forming the document-by-word matrix. One of the most important pre-processing procedures is the removal of stop words. Stop words are words such as "and", "is" and "but", which do not provide any information about the thematic content of a document. Another procedure which may also be useful in reducing the vocabulary size and consequently the sparsity, is referred to as lemmatisation. In this process, different forms (inflections) of a word are reduced to their root form. For example, inflections such as "runs", "ran" and "running" are all replaced by their base word, "run". In some cases, the removal of words which occur highly infrequently in the entire corpus may also be a useful practice. These techniques help the models to perform better by combating the sparsity problem and reducing the dimensionality of the corpus.

Consider the following example. Table 2.1 shows a manually created corpus of 10 (short) documents. Each was created by selecting a few sentences from various webpages[1] that fall under

[1] https://en.wikipedia.org/wiki/Cancer
https://www.cancerresearchuk.org/about-cancer/what-is-cancer
https://en.wikipedia.org/wiki/Donald_Trump

5 predetermined topics: cancer, Donald Trump, cats, broccoli and oil.

Table 2.1: Each cell in the table represents one document.

1	Cancer is a group of diseases involving abnormal cell growth with the potential to invade or spread to other parts of the body.
2	Cancer is when abnormal cells divide in an uncontrolled way. Some cancers may eventually spread into other tissues.
3	Donald John Trump (born June 14, 1946) is the 45th and current President of the United States, in office since January 20, 2017.
4	Trump has his own star on the Hollywood Walk of Fame which he received for the reality TV show The Apprentice.
5	The domestic cat is a small, typically furry, carnivorous mammal.
6	While not well known, the collective nouns used for cats and kittens are a clowder of cats and a kindle of kittens
7	Broccoli is known to be a hearty and tasty vegetable which is rich in dozens of nutrients.
8	On their own, broccoli has a stronger, greener flavor, while cauliflower is more delicate.
9	What you know as oil is actually called petroleum or crude oil and may exist as a combination of liquid, gas, and sticky, tar-like substances.
10	Oil and natural gas are cleaner fuels than coal, but they still have many environmental disadvantages.

After data cleaning (stop word and special character removal) the first and second documents, for example, become "cancer group disease involve abnormal cell growth potential invade spread part body" and "cancer abnormal cell uncontrolled cancer eventually spread tissue". The first 5 rows and columns of the associated document-by-word matrix for the corpus in Table 2.1 thus become

https://www.unbelievable-facts.com/2016/10/facts-about-donald-trump.html
http://justfunfacts.com/interesting-fact-about-cats/
http://umi123456.blogspot.com/
https://timesofindia.indiatimes.com/life-style/health-fitness/diet/11-health-benefits-of-broccoli/articleshow/30843390.cms
https://www.sheknows.com/home-and-gardening/articles/998289/broccoli-vs-cauliflower
https://www.dummies.com/education/science/environmental-science/what-is-the-environmental-impact-of-petroleum-and-natural-gas/

as follows:

$$
\begin{array}{c}
 \\
\text{document 1} \\
\text{document 2} \\
\text{document 3} \\
\text{document 4} \\
\text{document 5} \\

\end{array}
\begin{array}{c}
\begin{array}{cccccc}
\text{cancer} & \text{group} & \text{disease} & \text{involve} & \text{abnormal} & \\
\end{array} \\
\left[\begin{array}{cccccc}
1 & 1 & 1 & 1 & 1 & \dots \\
2 & 0 & 0 & 0 & 1 & \dots \\
0 & 0 & 0 & 0 & 0 & \dots \\
0 & 0 & 0 & 0 & 0 & \dots \\
0 & 0 & 0 & 0 & 0 & \dots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}\right]
\end{array}.
$$

The group of all the unique words that occur in a corpus is referred to as the *vocabulary*. In this case, the number of words in the vocabulary is 75. Thus, the full matrix contains 75 columns (one for each word) and 10 rows (one for each document). The first column corresponds to the word "cancer". From rows 1 and 2, we see that the word cancer occurs in the first document once and it occurs twice in the second document. Rows 3 to 5 indicated that the word "cancer" does not occur in documents 3 to 5. The other cells are interpreted in a similar fashion.

In a typical topic model, the topics are then represented in the model by a latent variable which can be estimated from the observed data, the document-by-word matrix. The output of a topic model is then groups of words that describe a topic. A topic is defined as a probability distribution over all the words in the vocabulary. During inference, each word in the vocabulary is assigned a probability of belonging to each topic, and usually only the ten words with the highest probabilities in a topic are then taken as the group of words which describe the topic. The name of the topic is typically determined by human judgment based on the most probable words in that topic. For example, if a topic model produces the following 3 most probable words in a topic $\{$*lions, tigers, leopards*$\}$, one could label the topic *animals* or even *wild cats*. This part of the process is clearly subjective.

In our example, a simple topic model was applied and Table 2.2 illustrates 3 of the 5 topics that were discovered.

Table 2.2: Examples of 3 topic models uncovered by a topic model. The group of words in topics 1, 2 and 3 discovered by the topic model correspond to the cancer, cat and broccoli topics, respectively.

Topic 1	Probability of word in topic 1	Topic 2	Probability of word in topic 2	Topic 3	Probability of word in topic 3
cancer:	0.112	cat:	0.137	broccoli:	0.102
abnormal:	0.076	kitten:	0.093	flavor:	0.053
cell:	0.070	typically:	0.048	delicate:	0.049
spread:	0.056	small:	0.039	stronger:	0.041
tissue:	0.040	domestic:	0.021	nutrient:	0.037

As previously mentioned, most topic models are unsupervised. Analogous to the manner in which the number of clusters must be specified prior to applying K-means clustering, it is usually also necessary to specify the number of topics prior to applying the topic model. In this example, the number of topics was assumed to be five. From Table 2.2, the group of words in topics 1, 2 and 3 clearly correspond to the cancer, cat and broccoli topics, respectively. In practice these labels are typically not known in advance so the onus is on the user to determine the topic which corresponds to the top words. Naturally, this creates challenges when it comes to evaluating the performance of the topic model as the true topics will not be known in advance to make a comparison.

Lastly, most topic models also incorporate a parameter which determines the proportion of each topic in a document. This can typically be represented by a document-by-topic matrix where, each row represents a document and each column represents the proportion of the document that belongs to each topic. Under the (strong) assumption that each document can only be about one

topic, the document-by-topic word matrix for the toy dataset in Table 2.1 would be as follows.

$$\begin{array}{c} \\ \text{document} \end{array} \begin{array}{c} \begin{array}{ccccc} \multicolumn{5}{c}{\text{topic}} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

As expected, the first and second documents were both assigned to the same topic, topic 1 (cancer) in Table 2.2, and the fifth and sixth belong to topic 2 (cats).

Usually, the one-topic-per document assumption is too rigid for long texts such ebooks or news articles, so topic models which relax this assumption and allow each document to contain multiple topics are usually more preferable. In such cases, each element in the document-by-topic matrix is a value between 1 and 0 and all the columns of each row sum to unity.

2.3 NOTATION

In the subsequent sections, topic modelling, as well as various related aspects, will be discussed in more detail. Unless otherwise specified, the notation that will be used throughout this thesis is listed and explained below.

- $\mathcal{C}$ denotes the corpus.
- $M \in \mathbb{N}$ is the number of documents in the corpus.
- $V \in \mathbb{N}$ is the size of the vocabulary. The vocabulary is defined as the list of all the unique words in the corpus. Each word in the vocabulary is assigned a unique label number, $v \in \{1, 2, \ldots, V\}$.

- $K \in \mathbb{N}$ is the number of topics. In most topic modelling procedures the user needs to assume a value of K beforehand.
- $n_m \in \mathbb{N}$ is the length of m-th document where $m = 1, 2, \cdots, M$.
- $\mathbf{x}_m = [x_{m1}, x_{m2}, \ldots, x_{mV}]$ denotes the word occurrence row vector of the m-th document and x_{mv} is the number of times word v occurs in the document.
- $\mathbf{X}$ denotes the $M \times V$ document-by-word matrix whose rows are $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M$. $\mathbf{X}'$ denotes the transpose of $\mathbf{X}$.
- Sometimes a document is represented as a *sequence* of words. The m-th document is then denoted $\mathbf{w}_m = [w_{m1}, w_{m2}, \ldots, w_{mn_m}]$ where w_{mn} (for $n = 1, 2, \ldots, n_m$) is the label number for the n-th word in document m after pre-processing.
- $\mathbf{z} = [z_1, z_2, \ldots, z_M]$ denotes the vector of topic assignments for the documents in the corpus where $z_m \in \{1, 2, \ldots, K\}$ denotes the topic assignment indicator variable of the m-th document. In the context of topic modelling, this vector is a latent variable.
- Θ is used to denote the document-by-topic matrix. Each row is a vector, $\theta_m = [\theta_{m1}, \theta_{m2}, \cdots, \theta_{mk}]$, where θ_{mk} denotes the proportion of the m-th document which is about topic $k \in \{1, 2, \ldots, K\}$. Furthermore, it is assumed that $0 \leq \theta_{mk} \leq 1$ and $\sum_{k=1}^{K} \theta_{mk} = 1$ for all m.
- Φ will be used to denote the topic-by-word matrix. This is the matrix that is of most interest in topic modelling. Each row, $\phi_k = [\phi_{k1}, \phi_{k2}, \ldots, \phi_{kV}]$ is such that ϕ_{kv} is the probability of word v belonging to topic k. It is also assumed that $0 \leq \phi_{kv} \leq 1$ and $\sum_{v=1}^{V} \phi_{kv} = 1$ for all k.

2.4 STATISTICAL APPROACHES TO TEXT ANALYSIS

In general, topic models bear some resemblance to some of the well-known techniques that are commonly found in the statistical literature. In this section, various approaches towards gaining deeper insight into the contents of a corpus are described. This discussion ultimately leads to an exposition of where topic models fit in with other well-known statistical methods.

2.4.1 CLUSTERING

Given a similarity (or dissimilarity) measure, clustering groups similar documents together, which consequently reveals information regarding the relationships between documents Aggarwal and Zhai (2012). The simplest form of clustering associates each document to exactly one cluster. However, given the complex nature of documents, *soft* (or probabilistic) clustering techniques may be more suitable as they allow a document to be associated with more than one cluster. In the context of topic modelling, this clustering is achieved through the inference of the document-by-topic matrix, θ. As previously discussed, each row represents the fraction of each topic contained in a document. One weakness of clustering analysis is that labels for the clusters are not automatically inferred.

2.4.2 DIMENSIONALITY REDUCTION

Owing to the vector representation of documents and the magnitude of the corpora of interest, some dimensionality reduction techniques from applied mathematics and machine learning may also be applicable.

2.4.2.1 CLASSICAL PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) (Jolliffe, 1986) is a well-known dimensionality reduction technique. PCA tries to find a linear mapping of each data point, $\mathbf{x}_m \in \mathbb{R}^V$, onto an orthonormal subspace $\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_K\}$, that maximises the variability of the projections, $\hat{\mathbf{x}}_m$, of the data points.[2] This subspace is called the *principal subspace*. It can be shown that the subspace used to achieve this is made up of the eigenvectors of the sample covariance matrix of the data points. Dimensionality reduction is achieved by selecting the subspace containing the $K < V$ eigenvectors which correspond to the highest eigenvalues.

The principal subspace can also be interpreted as the orthonormal subspace that minimizes the sum of the squared distances between the data points and their projections (or *reconstruction error*), $\sum_{m=1}^{M} ||\mathbf{x}_m - \hat{\mathbf{x}}_m||^2$ (Bishop, 2006). If $\mathbf{H}$ denotes an orthogonal $V \times K$ dimensional weight (or *loading*) matrix and $\mathbf{z}_m \in \mathbb{R}^K$ denotes a set of real-valued latent variables corresponding to the m-th data point, then $\hat{\mathbf{x}}_m = \mathbf{H}\mathbf{z}_m$ (Murphy, 2012). According to Theorem 12.2.1 in

[2]For notational simplicity it is assumed that each data point, $\mathbf{x}_m$, is first centred around 0 before applying PCA.

Murphy (2012), the optimal solution is found by setting the columns of $\mathbf{H}$ to be the eigenvectors, $\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_K\}$, corresponding to the K highest eigenvalues of the sample covariance matrix. Consequently, the optimal linear reconstruction of $\mathbf{x}_m$, is $\hat{\mathbf{z}}_m = \mathbf{H}'\mathbf{x}_m$ (Tipping and Bishop, 1999).

Subramanian (2015) refers to PCA as a one-mode factor analysis because the input is a matrix of associations relating only a single entity (for example, in our context (Borko and Bernick, 1963) the single entity would be a document). Although, PCA enables the user to find a reduced representation of the corpus, the new dimension onto which the documents are projected are not always easily interpretable thus limiting the usefulness of this technique.

In the field of text mining, PCA is closely related to a well-known technique called latent semantic indexing (also called latent semantic analysis) (Deerwester et al., 1990), which was initially developed for information retrieval. Instead of applying PCA to the covariance matrix of the data, latent semantic indexing (LSI) is performed by applying PCA directly to the data matrix, $\mathbf{X}$, via a singular value decomposition (SVD). This approach is regarded as a two-mode factor analysis as it allows for the analysis of two variables, namely documents and words, instead of just one as in PCA (Deerwester et al., 1990). The data matrix is then factorised into a product of 3 matrices. When the SVD is applied directly to the word by document matrix, the resulting matrices allows for word to word comparisons, document to document comparisons and document to word comparisons. The reader is referred to (Deerwester et al., 1990) for further details.

2.4.2.2 EXPONENTIAL FAMILY PRINCIPAL COMPONENT ANALYSIS

Exponential family principal component analysis (ePCA) (Collins et al., 2001) is a generalisation of PCA which is better suited to dimensionality reduction in discrete data. The generalisation is analogous to the manner in which generalised linear models expand regression to other members of the exponential family of distributions. Suppose each observed data point, $\mathbf{x}_m$, is assumed to belong to an unknown distribution in the exponential family, $\mathbf{x}_m$ can be regarded as a noise-corrupted version of the true points $\hat{\mathbf{x}}_m$, where $\hat{\mathbf{x}}_m$ denotes the natural parameter of the exponential family. Consequently, the objective of PCA becomes finding the parameters, $\hat{\mathbf{x}}_m$, which lie in a lower dimensional space that maximises the likelihood of the data.

This "probabilistic" interpretation is equivalent to the minimisation of the sum of the squared distances interpretation of PCA if the data is assumed to follow a unit Gaussian distribution (stan-

dard normal distribution). This is due to the fact that the negative log-likelihood under a unit Gaussian model (ignoring constants) is equal to the loss function, $\sum_{m=1}^{M} ||\mathbf{x}_m - \hat{\mathbf{x}}_m||^2$, which is minimised under classical PCA. When the data is assumed to follow a different distribution, which is also a member of the exponential family, the form of this loss function is different and the $\hat{\mathbf{x}}_m$ typically no longer lie in a linear subspace. Furthermore, where it is inappropriate to assume normally distributed data, such as for binary or integer-valued data, it can be advantageous to assume other distributions such as the Poisson or Bernoulli distributions, and apply ePCA instead of classical PCA (Collins et al., 2001).

Exponential family PCA falls into the class of deterministic latent variable models, along with classical PCA (Welling et al., 2008). As ePCA is prone to overfitting, a probabilistic equivalent, Bayesian ePCA, was proposed (Mohamed et al., 2009). The Bayesian ePCA formulation leads to an elegant generative model.[3] Let $\mathbf{W}$ denote a $V \times K$ dimensional matrix with columns $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K\}$ and $\mathbf{Z}$ denote an $M \times K$ dimensional matrix with columns $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K\}$. Bayesian PCA assumes that the natural parameter of the distribution of the data, $\mathbf{X}'$, can be represented by the product $\mathbf{WZ}'$. The data-independent matrix, $\mathbf{W}$, is called the factor loading matrix. It is similar to the $\mathbf{H}$ matrix introduced in classical PCA, but the columns are not necessarily eigenvectors. The matrix $\mathbf{Z}$ denotes the factor score matrix which represents the reduced vectors. $\mathbf{Z}$ and $\mathbf{W}$ are both latent variables which can be estimated via Monte Carlo sampling methods.

In the generative process, it is first assumed that parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are first randomly selected from a normal and inverse gamma distribution, respectively. Then for each data point, $\mathbf{x}_m$, a lower dimensional score, $\mathbf{z}_m$, is randomly drawn from

$$\mathbf{z}_m \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

The data is then sampled from the conditional distribution

$$\mathbf{x}_m | \mathbf{z}_m, \mathbf{W} \sim Expon(\mathbf{W}\mathbf{z}_m),$$

where the notation $Expon(\theta)$ denotes any member of the exponential family with corresponding natural parameter θ. It is also assumed that the columns of $\mathbf{W}$ are selected from an appropriate

[3] A generative model is a statistical model that represents the process by which data is assumed to have been formed.

conjugate distribution. If $\mathbf{z}_m \sim N(\mathbf{0}, \mathbf{I})$ and $\mathbf{x}_m|\mathbf{z}_m, \mathbf{W} \sim N(\mathbf{x}_m|\mathbf{W}\mathbf{z}_m, \sigma^2\mathbf{I})$, this is known as probabilistic PCA (Tipping and Bishop, 1999).

Under this formulation of ePCA, the appropriate member of the exponential distribution for a document would be a multinomial distribution. Thus

$$\mathbf{x}_m|\mathbf{z}_m, \mathbf{W}, n_m \sim Mult(n_m, S(\mathbf{W}\mathbf{z}_n)),$$

where n_m denotes the length of the m-th document and $S(\cdot)$ denotes the softmax function which is used to convert the natural parameter of the multinomial distribution to the dual parameter[4].

Under this formulation, it can be seen that the latent variable becomes embedded in the natural parameter of the distribution. However, reformulating these results with a focus on the dual parameters produces useful results, which form the basis of some of the most widely used and effective topic models.

2.5 TOPIC MODELS: WHERE CLUSTERING AND DIMENSIONALITY REDUCTION MEET

Topic models are very powerful tools as they possess characteristics from both clustering and dimensionality reduction techniques:

1. A corpus is represented in a lower dimensional form by a set of topics. Assuming $K < M$, then the $M \times V$ dimensional corpus is summarised by the lower dimensional $K \times V$ topic-by-word matrix.

2. Similar to clustering, each document is associated with a single topic or multiple topics, depending on the model. This aspect is captured by the document-by-topic matrix. Unlike clustering, "labels" for each cluster are also produced in the form of topics.

In order for topic models to be useful, they must not only provide data compression, but also produce topics which are interpretable. In the following sections, a few basic topic models will be discussed so as to demonstrate how clustering and dimensionality are performed in a single model.

[4]The dual parameter of a distribution which is a member of the exponential family tends to be the parameter which is more commonly known. For instance, the natural parameter for the multinomial distribution is the vector of log odds whereas the dual parameter is the probability vector, $\mathbf{p}$. Similarly the natural parameter of the Poisson distribution is the log of the mean whereas the dual parameter is the mean, λ.

2.5.1 MULTINOMIAL PRINCIPAL COMPONENT ANALYSIS

Multinomial principal component analysis (mPCA) (Buntine, 2002) is a topic model which draws heavily from probabilistic PCA (PPCA) (Tipping and Bishop, 1999), a variant of the Gaussian version of ePCA. One of the key differences between ePCA and mPCA, is that ePCA models $\mathbf{W}\mathbf{z}_m$ as the natural parameter whereas mPCA models it as the dual parameter. Owing to this, constraints are added to $\mathbf{W}\mathbf{z}_m$ so as to ensure it falls in the correct domain. Multinomial PCA assumes a likelihood similar to that of ePCA, however, the softmax function is omitted, as $\mathbf{W}\mathbf{z}_m$ represents the dual parameter of the multinomial distribution. If it is assumed that there are K classes, the dual parameter must lie in the K-dimensional simplex.

Instead of assuming a Gaussian prior, the latent variable, $\mathbf{z}_m$, is assumed to follow a K-dimensional Dirichlet distribution.[5] Such a prior has the advantage of adding computational ease due to its conjugacy to the multinomial distribution.[6] More importantly, when the latent variable is defined in this manner, it captures the proportion of each document belonging to each topic. This aspect of the model brings in the "clustering" aspect of the topic model.

Lastly, the matrix which was previously referred to as $\mathbf{W}$ is constrained so that the columns all sum to 1 and each entry lies between 0 and 1. The matrix $\mathbf{W}$ is a $V \times K$ dimensional matrix which denotes the word-by-topic matrix described in Section 2.1. It is from this matrix that topics similar to those given in Table 2.2 are derived. As these topics can be regarded as lower dimensional representations of the documents, this aspect of the model brings in the "dimensionality" reduction aspect of the topic model.

So as to clearly differentiate between the unconstrained version of $\mathbf{W}$ and $\mathbf{z}_m$, the constrained versions will be denoted as $\mathbf{B}$ and $\boldsymbol{\pi}_m$, respectively. Letting α denote the hyper-parameter of the

[5] The K-dimensional Dirichlet distribution has the following probability density function,

$$p(\theta|\vec{\alpha}) = \int \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1},$$

where $\Gamma(x)$ denotes the gamma function, $\theta = [\theta_1, \theta_2, ..., \theta_K]$ and $\alpha = [\alpha_1, \alpha_2, ..., \alpha_K]$ are both K-dimensional vectors whose elements have the following properties: $\theta \geq 0$, $\sum_{k=1}^{k}\theta_i = 1$ and $\alpha_i > 0$. The Dirichlet distribution is a member of the exponential family of distributions and is conjugate to the multinomial distribution.

[6] Suppose $Q \sim F$ and $X|q \sim G$. Q is said to be conjugate to X if $Q|X$ follows a distribution in the same family as Q.

Dirichlet distribution, mPCA assumes a prior of the form

$$\boldsymbol{\pi}_m|\alpha \sim Dir(\alpha), \tag{2.1}$$

and a likelihood of the form

$$\mathbf{x}_m, n_m|\boldsymbol{\pi}_m \sim Mult(n_m, \mathbf{B}\boldsymbol{\pi}_m). \tag{2.2}$$

2.5.2 LATENT DIRICHLET ALLOCATION

Latent Dirichlet allocation (LDA) (Blei et al., 2003) is the most popular and widely studied of all topic models due to its long record of success. Since its inception, much of the development in the field has been in producing modifications and extensions of the model. It is a three-level hierarchical Bayesian model (Blei et al., 2003) which is very similar to mPCA. The main difference is the assumed representation of the data, and hence the form of the likelihood. Instead of representing each document as a vector of word counts, the m-th document, is represented as a variable sequence of words $\mathbf{w}_m = [w_{m1}, w_{m2}, \ldots, w_{mn_m}]$ where n_m denotes the length of the document. If all the words in the vocabulary are uniquely labelled from 1 to V, then each element in $\mathbf{w}_m$ is such that $w_{mn} \in \{1, 2, \ldots, V\}$ where $n = 1, 2, \ldots, n_m$.

The natural choice of likelihood for such a distribution is a categorical (or multinoulli) distribution. Under LDA, the prior is the same as that of mPCA given in Equation (2.1), but the likelihood is

$$p(\mathbf{w}_m|\pi_n) = \prod_{n=1}^{n_m} Cat(w_{mn}|\mathbf{B}\boldsymbol{\pi}_m). \tag{2.3}$$

When applying LDA, an assumption regarding the number of topics contained in the corpus, K, must be made prior to fitting the model. The latent parameter, $\boldsymbol{\pi}_m = [\pi_{m1}, \pi_{m2}, ..., \pi_{mK}]$, is constrained such that $0 \leq \pi_{mk} \leq 1$ for all $k = 1, 2, ..., K$ and $\sum_{k=1}^{K} \pi_{mk} = 1$ and, it represents the topic distribution of the m-th document. In other words, the proportion of the m-th document belonging to topic k is π_{mk}. Similarly, the columns of the matrix $\mathbf{B}$, $\{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_K\}$, denote the topic distribution of topic k where the sum of the elements of $\mathbf{b}_k$ all lie between 0 and 1, and sum to unity.

LDA is related to the multinomial mixture model, which is merely a special case of the well-

known mixture model theory applied in statistical theory. Section 2.5.2.1 will highlight this relationship.

2.5.2.1 MIXTURE MODELS

Mixture models are amongst the simplest latent variable models. They too have been considered for the problem of text analysis – especially the multinomial mixture model which can be regarded as a more restrictive version of LDA.

In the context of topic modelling this approach is typically referred to as the mixture of unigrams model (Nigam et al., 2000) or Dirichlet multinomial mixture model (DMM) (Yin and Wang, 2014). In order to easily see the relationship between LDA and the mixture models, we look at the form of the complete data distribution. Let $\mathbf{z}_m$ denote a discrete latent indicator variable so that the complete data distribution is

$$p(\mathbf{w}_m) = \sum_{\mathbf{z}} p(\mathbf{w}_m|\mathbf{z}_m)p(\mathbf{z}_m). \tag{2.4}$$

(Note, the summation is replaced with an integral if $\mathbf{z}_m$ is continuous.) In both LDA and mixture models, the likelihood $p(\mathbf{w}_m|\mathbf{z}_m)$ is assumed to follow a product of categorical distributions as in Equation (2.3). The key difference is in the form of prior. Latent Dirichlet allocation assumes a Dirichlet prior (thus replacing the summation in Equation (2.4) with an integral) whereas mixture models always assume a categorical prior (Murphy, 2012).

The implication of this difference is that the mixture model uniquely assigns each document to a single topic, whereas the LDA possesses the flexibility of allowing each document to contain multiple different topics in different proportions. It is for this reason that LDA is sometimes referred to as an admixture model (Erosheva et al., 2004). In practice, there are many instances where the LDA assumption is more sensible, which is part of the reason for its popularity.

The focus of this thesis is on topic modelling with a special focus on short text. Section 2.6 will now draw attention to some of the research that has been conducted over the years in this field.

2.6 RECENT RESEARCH ON TOPIC MODELLING OF SHORT TEXT

Traditional topic models have a proven history of success on long documents, such as news articles and e-books. However, texts that are significantly shorter have become of increasing relevance over recent years. Short text is prominent on the Internet and arises in many forms, such as tweets or status updates on social media platforms, web page snippets, news headlines and product reviews. Such sources of text potentially hold valuable information that can be useful in many applications, such as event tracking (Lin et al., 2010), interest profiling (Weng et al., 2010) and product recommendation (Zhang and Piramuthu, 2018).

Traditional topic models typically infer topics based on word co-occurrence relationships between words (Yan et al., 2013). In order to extract meaningful topics, a topic model must successfully infer these relationships from a corpus. Per definition, short text contains few words and consequently tends to contain less co-occurrence information than long text. As a result, this sparsity makes it difficult for traditional topic models to uncover the relationships between words and hence, extract meaningful and coherent topics (Cheng et al., 2014). This has created a need for tools and techniques that can effectively overcome the challenges posed by short texts. The remainder of this chapter is an analysis of the existing literature on the subject of short text topic modelling.

Much research has been conducted to address different aspects of the complications that typically arise in the application of topic models to short text. In order to systematically analyse the literature, the topic models are divided according to the approach that the researchers proposed to overcome these challenges. Where a model incorporates more than one approach, it is discussed under the section that best describes the prominent idea. Only probabilistic topic models for short text are considered.

2.6.1 POOLING OR AGGREGATION OF SHORT TEXTS INTO LONGER DOCUMENTS

One common approach that has been explored by researchers is aggregating short texts into longer pseudo-documents. Pooling approaches try to address the sparsity problem of short text by creating word co-occurrence information. One of the earliest and most popular works following this approach involved pooling tweets according to hashtag (Mehrotra et al., 2013), then applying

traditional LDA. Other pooling schemes include pooling according to author (Weng et al., 2010), common terms (Hong and Davison, 2010), time or trending topics (Mehrotra et al., 2013). Despite the favourable results observed, such methods have the flaw of being data dependent as texts must have appropriate meta-data to allow for sensible aggregation (Mehrotra et al., 2013). However, such auxiliary information is not always available or easy to access.

In order to address the aforementioned shortcoming, some researchers proposed another class of aggregation-type topic models which assume that each short text is linked to some longer latent document. The self-aggregation based topic model (SATM) (Quan et al., 2015) is one such model. As the name suggests, it tries to automatically aggregate texts independently of the availability of suitable meta-data, thus making it a form of "generalisation". SATM is a generative model which assumes that each short text originally belonged to a single longer latent document. It was shown to outperform several baselines on artificially generated text (Quan et al., 2015). However, it is plagued with several problems, including overfitting and severe high time complexity (Zuo et al., 2016a), (Li et al., 2016a).

The pseudo-document-based topic model (PTM) (Zuo et al., 2016a), which is another aggregation-based approach to topic modelling of short text, was proposed to overcome these challenges. It also does not depend on the availability of auxiliary information. PTM overcomes the overfitting problem of SATM by assuming that each short text is generated by sampling a word from the topic distribution of a longer document, instead of the word distribution of the longer document, as is the case with SATM. The sparsity-enhanced PTM (SPTM) was also proposed as a sparsity-enhanced version of PTM. It was specially designed to be applied when the size of the latent pseudo-document corpus is assumed to be small. This variation of PTM was achieved by incorporating a Spike and Slab prior (Ishwaran and Rao, 2005) in order to induce sparsity. The latent topic model (LTM) (Li et al., 2017b) is another topic model which is almost identical to PTM, only differing in that it does not define a prior for the distribution of the document assignments of the short texts. Both PTM and LTM were shown to outperform SATM as well as other baselines, thus making them competitive modelling procedures for short texts (Zuo et al., 2016a; Li et al., 2017b).

The self-aggregating dynamic topic model (SADTM) is another model based on self-aggregation, but it goes a step further by also accounting for the time-varying nature of dynamic short texts, such as those commonly found in social media. As many topic models are designed for

static corpora, SADTM (Shi et al., 2019) was designed to be a more suitable model for short texts whose topics vary over time. Despite the success of these models, one common problem which is inherent to models that assume short texts are formed from longer pseudo-documents is determining the optimal number of pseudo-documents as this has an impact on model performance (Zuo et al., 2016a). This choice is difficult due to the fact that these pseudo-documents are regarded as latent (unobserved).

2.6.2 ASSUMING EACH DOCUMENT BELONGS TO A LIMITED NUMBER OF TOPICS

Traditional topic models often assume that each document contains multiple topics. It has been shown that this assumption is not always suitable as some texts are so short it is more likely that they focus on only a single topic (Lin et al., 2014; Mazarura and de Waal, 2016). This assumption can easily be modelled via the use of a mixture model. The mixture of unigrams model (Nigam et al., 2000) assumes that the corpus can be modelled as a mixture of documents (multinomial observations) where each cluster component represents the distribution over words of a topic. This model is sometimes referred to as the Dirichlet multinomial mixture (DMM) model due its use of Dirichlet priors. In the original model by Nigam et al. (2000), parameter estimation was performed using an EM algorithm, but it was later adapted to use a Bayesian approximation method by Yin and Wang (2014) who proposed a collapsed Gibbs sampler version, GSDMM (Gibbs Sampler DMM). It was a significant improvement upon DMM as it not only converged quickly, but also had the favourable property of being able to automatically infer the number of topics/clusters in the corpus. Twitter-LDA (Zhao et al., 2011) is another topic model that only allows each document (tweet) to belong to one topic. Unlike the other one-topic-per-document topic models, it is regarded as an extension of LDA (Zhao et al., 2011).

In general, most researchers in the short text topic modeling field tend to avoid making the one-topic-per-document assumption as they feel that it is too restrictive and may not hold for all short texts. In light of this, the Poisson-based Dirichlet multinomial mixture model (PDMM) (Li et al., 2017a) was proposed as an extension to DMM which relaxed the one-topic assumption. It models the number of topics in each short text according to a Poisson distribution and only allows each one to belong to either 1, 2 or 3 topics. By allowing for this flexibilty, PDMM was able to outperform standard DMM with respect to topic coherence.

2.6.3 MODELLING INNOVATIVE DOCUMENT REPRESENTATIONS

When researchers deviate from traditional ways of modelling short text, it creates room for innovations in the field. Typically, topic models such as LDA and DMM model the document generation process and, only implicitly, model document-level word co-occurrences. The biterm topic model (BTM) (Yan et al., 2013) deviates from this by modelling the global word co-occurrence patterns in the corpus instead of document co-occurrence. BTM directly models the generation of biterms[7] in the global corpus (Yan et al., 2013). Given its widespread success, BTM is regarded as one of the state-of-the-art models for short text. However, it has the unfavourable weakness of having a high time complexity (Liang et al., 2018). Another competing method which is able to outperform BTM is the word network topic model (WNTM) (Zuo et al., 2016b). Unlike LDA which models the topic distribution of each document, WNTM models the distribution over topics for each word (Zuo et al., 2016b). Instead of representing a documents as a bag-of-words as in LDA and DMM, under WNTM each document is represented as a network depicting the number of other words in the vicinity of each word based on a sliding window across each document. Such a representation deals with the sparsity issue as the word-word space is denser than the document-word space (Zuo et al., 2016b).

Zhou and Yang (2018) proposed using LDA with a pattern set-based text representation (PSTR) to create a new topic model, LDA-PSTR. LDA-PSTR is a probabilistic topic model that uses frequent pattern mining to represent documents as pattern frequency vectors as opposed to a bags-of-words. The representation addresses the sparsity problem in that it explicitly captures co-occurrence patterns and semantic relations of words on a corpus-level (Zhou and Yang, 2018).

More recently, Chen et al. (2020) proposed a new biterm-based topic model, the Dirichlet process biterm-based mixture model (DP-BMM). Their model differs from previous biterm-based topic models in that it models biterms at a document level, it can directly obtain document-topic distributions and assumes each document belongs to a single topic (Chen et al., 2020).

2.6.4 ENRICHING SHORT TEXTS WITH AUXILIARY INFORMATION

In an attempt to alleviate the data sparsity problem in short text, another approach that has been widely investigated is that of expanding short texts with suitable external knowledge to create more

[7] A biterm is defined as an unordered pair of words that co-occur in a window.

co-occurrence information and context in short text. Phan et al. (2011) proposed a framework that uses topics from a larger, more diverse "universal corpus", such as Wikipedia articles, to enrich short texts. Other researchers noticed that some short texts are sometimes followed by a thread of related posts. LeadLDA (Li et al., 2016b) tries to leverage this auxiliary information by incorporating it into its prior.

There are also topic models which incorporate word embeddings. Word embeddings are a group of language modelling techniques that map words to vectors in a manner that retains the contextual information of the words as well as semantic and syntactical relationships (Li et al., 2016a). Latent feature DMM (LF-DMM) (Nguyen et al., 2015) incorporates word embeddings by the inclusion of a Bernoulli indicator variable which determines whether a word is generated from a Dirichlet multinomial distribution or a probability distribution dependent on pre-trained word embeddings. Li et al. (2016a) also identified the utility of word embeddings in short text topic modelling and proposed extensions of DMM and PDMM, namely the generalised Pólya urn DMM (GPU-DMM) and generalised Pólya urn PDMM (GPU-PDMM). They incorporated information about the relatedness of words based on word embeddings through the generalised Pólya urn model (Mahmoud, 2008). The Pólya urn captures the relatedness of words by promoting semantically related words in a topic. These models were proposed as improvements to LF-DMM and were shown to perform better with respect to topic coherence (Li et al., 2016a). Furthermore, it was observed that GPU-DMM was generally outperformed by GPU-PDMM, but this superior performance came at the expense of significantly higher computational costs. More recently, prompted by the success of these DMM and PDMM variants, Guo et al. (2020) proposed the generalised Pólya urn biterm topic model (GPU-BTM).

Since word embeddings are trained on large sets of external corpora there is a risk of introducing irrelevant information. In order to lower this risk, Zhang et al. (2018) proposed an improvement to the GPU-based models, which uses Point Mutual Information between words to filter semantic relatedness in the word embeddings. Furthermore, unlike DMM which assumes all the words in a document are associated with one topic, their model is more similar to Twitter-LDA which assumes each document belongs to one topic, yet each word is either related to this topic or some other global background topic (Zhang et al., 2018). Liang et al. (2018) later proposed a global and local word embedding-based topic model, GLTM, that also tries to ensure that the incorporated word embeddings are suitable. Unlike the model of Zhang et al. (2018), GLTM

uses local word embeddings from the training corpus in addition to the global word embeddings from the external corpus. Overall, incorporating word embeddings is a very popular approach amongst researchers. Other topic models that take advantage of word embeddings for short text topic modelling include Li et al. (2019); Gao et al. (2019) and Huang et al. (2020).

The ULW-DMM model (Yu and Qiu, 2019) is another more recent contribution to short text topic modelling. This model was built based on the premise that the strengths of three different approaches can be leveraged by their combination, whilst their individual weaknesses are simultaneously overcome. ULW-DMM is based on DMM, yet it also incorporates both external information via the use of word-embeddings and internal information by modelling texts according to user. The model was shown to outperform baselines such as DMM, LDA and LF-DMM (Yu and Qiu, 2019).

Although taking advantage of external sources of information has been shown to improve topic model performance, this approach may not always be viable for short texts that lack suitable external data sources to exploit. In the cases where external data can be identified, there is no standard way of determining the extent to which the external data is appropriate and there is also a risk that its incorporation can introduce noise into the short text corpus.

2.6.5 INDUCING SPARSITY INTO TOPIC MODELS

Most topic models assume that the corpus contains K topics and they assign probability across these K topics for each document. Similarly, for each topic, probabilities are assigned to each of the V words in the vocabulary. LDA, for instance, assigns non-zero probabilities to all topics and all words. However, in practice individual topics tend to contain a few main topics as opposed to all K topics and only a smaller subset of the vocabulary is relevant to a topic (Lin et al., 2014). This is especially true for short text. One intuitive, but unfortunately ineffective, approach to inducing this sparsity is selecting smaller values for the hyperparameters of the Dirichlet priors in LDA (Zhu and Xing, 2011). Sparse topical encoding (STC) (Zhu and Xing, 2011) on the other hand, is able to effectively induce this sparsity by introducing a Laplace prior, which is possible due the non-probabilistic nature of the model. Although it is the state-of-the-art non-probabilistic model that induces the this sparsity, it is generally outperformed by its probabilistic counterpart, the dual sparse topic model (DSTM) (Lin et al., 2014). DSTM is of special interest as it has also proven to be an effective model for topic extraction in short text (Lin et al., 2014). DSTM

is a variation of LDA which modifies traditional LDA by incorporating a spike and slab prior to induce sparsity into the topic distribution of the document and word distribution of the topics. Despite the positive results reported in the original paper, the model does not always perform well on all short text (Lin et al., 2014). This may be because the assumption that each document only contains a few prominent topics may not always be valid for a given corpus of short text. In the cases where the each document was indeed about a single topic, DSTM was outperformed by the DMM model (Lin et al., 2014). Lastly, the conditional Gamma-Poisson model (CGP) Buntine and Jakulin (2006) is an extension of the Gamma-Poisson model proposed by Canny (2004). This model incorporates a zero inflated gamma distribution in its design so as to account for excess zeros in the score matrix.

2.7 CONCLUSION

This chapter began with an introduction to topic models. The basic application and features of typical topic models were demonstrated with a small toy corpus. This was then followed by a unifying framework that described the connection between topic models and other well-known statistical techniques. This lead to the exposition that topic models possessed both dimensionality reduction and clustering characteristics.

It was then shown, through a literature study, that there is still no universal solution to addressing the short text problem in topic modelling, as every model presents its own challenges. Some models depend on the availability of external information in order to be applicable; yet, such information is not always readily accessible for all texts. Others introduce new parameters for which the selection of optimal values is difficult. Upon evaluation of the experimental results presented in the literature, it was often seen that the performance of models was not always consistent.

It is clear that topic modelling of short text is still an active and open area for further research as there is a need for topic models that are both flexible and robust[8]. In other words, they should still be applicable and perform well across different types of documents. In addition, such methods need to be simple in order to promote their widespread use and they also need to be fast to effectively handle the large datasets that are prevalent in practice.

[8]In statistics, a statistical method is said to be robust if it can operate well under violations of its underlying assumptions.

Now that the basics have been covered, Chapter 3 will now present the foundations of the new topic model presented in this thesis.

CHAPTER THREE

MOTIVATION FOR POISSON-BASED TOPIC MODELS

3.1 INTRODUCTION

Topic models are closely related to classification (supervised learning) and cluster analysis (unsupervised learning). In the literature, their performance is mostly assessed based on topic quality, classification performance and, occasionally, clustering ability. Most topic models are constructed under the assumption that documents follow a multinomial distribution. The Poisson distribution is an alternative distribution to describe the probability of count data. It has been successfully applied in other text mining fields, such as text classification, but its application to topic modelling is not well documented, specifically in the context of generative probabilistic models.

The new topic model for short text proposed in this thesis combines the strengths of both the Poisson distribution and mixture modelling. The aim of this chapter lays the foundation of this innovation. More specifically, the multinomial distribution and Poisson distribution are discussed in detail with a focus on their application in text mining. This then leads to the motivation for the choice of using the Poisson distribution with mixture models. This chapter is concluded by a study of the empirical characteristics of short text in relation to the Poisson distribution.

3.2 DISTRIBUTIONS FOR COUNT DATA IN TOPIC MODELLING

The bag-of-words representation is a common and simple way of representing text. Under this representation, each document is represented as a V-dimensional vector of word frequencies. A direct consequence of this is that the corpus becomes a collection of discrete multivariate data. Natural choices for modelling count data include the multinomial and Poisson distributions which shall be discussed in the following sections.

3.2.1 THE MULTINOMIAL DISTRIBUTION

Under the bag-of-words representation, each of the M documents is represented as a vector, $\mathbf{x}_m = [x_{m1}, x_{m2}, \ldots, x_{mV}]$, where x_{mv} is the frequency of word v in the m-th document for $v \in \{1, 2, ..., V\}$ and $m \in \{1, 2, ..., M\}$. Assuming that each document belongs to exactly one of the K topics, the topic assignment of the m-th document is denoted by $z_m \in \{1, 2, \ldots, K\}$. The topic assignments for the entire corpus is given by the latent vector $\mathbf{z} = [z_1, z_2, \ldots, z_M]$.

Under the multinomial distribution, a topic is defined as a multinomial distribution over words and each document is assumed to be a draw from a multinomial distribution conditioned on the topic. In the $K \times V$ topic-by-word matrix, Φ, each row, denoted $\phi_k = [\phi_{k1}, \phi_{k2}, \ldots, \phi_{kV}]$, corresponds to a topic $k \in \{1, 2, \ldots, K\}$ and ϕ_{kv} is the probability of word v belonging to topic k. Thus, the probability of the document $\mathbf{x}_m$ given the topic is given by

$$p(\mathbf{x}_m|\mathbf{z}, \Phi) = (\sum_{v=1}^{V} x_{mv})! \prod_{v=1}^{V} \frac{\phi_{kv}^{x_{mv}}}{x_{mv}!}. \tag{3.1}$$

The multinomial distribution models the assumption that words are generated independently given the topic. This assumption is generally not valid for real-world text, however the assumption is commonly made as it greatly simplifies computation, and text mining tasks built on this assumption have often proven to still produce satisfactory results despite the violation of the assumption (eg. multinomial naïve Bayes text classification (Eyheramendy et al., 2003)). The multinomial is also a convenient choice due to its simplicity even in the presence of high dimensional data.

When the number of trials is fixed and known, the multinomial distribution is a member of the exponential family of distributions. The multinomial distribution has the favourable property of having a conjugate prior, the Dirichlet distribution, which is computationally convenient

and simplifies the derivation of the posterior distribution. In some applications, researchers have discovered other benefits of imposing a Dirichlet prior. For instance, the Dirichlet-multinomial is able to model burstiness in text whereas the traditional multinomial distribution fails to do so as effectively (Madsen et al., 2005). The term burstiness is used to describe the phenomenon in which rare words appear many times in a single document (Church and Gale, 1995). In light of the sparsity of a typical document-by-word matrix, it is clear that most words will not occur in a document. However, if a word occurs once, there is a considerable chance of the word occurring multiple times in the same document, i.e. words appear in bursts (Madsen et al., 2005).

3.2.2 THE POISSON DISTRIBUTION

The Poisson distribution is a member of the exponential family whose conjugate prior is the gamma distribution. It models the distribution of the number of events in a fixed interval given the average number of occurrences for the interval. In the context of text modelling, the Poisson distribution can be used to model the number of occurrences of a word in documents of fixed length. Similar to the multinomial, the Poisson models the assumption that words occur independently.

The parameter of the multinomial distribution is a vector containing the probabilities of each word belonging to a topic. In contrast, the Poisson parameter is the expected number of occurrences of a word in a topic. Let λ_{kv} denote the expected frequency of word v in topic k. Assuming that each document belongs to exactly one topic and that the frequencies of the words in a document are independent given a topic, the distribution of a document $\boldsymbol{x}_m$ is given by

$$p(\boldsymbol{x}_m|\mathbf{z}, \boldsymbol{\lambda}) = \prod_{v=1}^{V} \frac{\lambda_{kv}^{x_{mv}} e^{-\lambda_{kv}}}{x_{mv}!}, \tag{3.2}$$

where $\boldsymbol{\lambda}$ is a $K \times V$ matrix whose rows are $\lambda_k = [\lambda_{k1}, \lambda_{k2}, \ldots, \lambda_{kV}]$.

One of the important differences between the multinomial and Poisson distributions, is that the multinomial distribution models counts in documents of arbitrary length whereas the Poisson models counts in documents of *fixed* length. Failure to normalise document length prior to the application of the Poisson distribution can result in poor performance in some text mining applications (Ogura et al., 2014).

Despite it being an option for modelling count data, the Poisson distribution is often disregarded. One reason for the limited research in generative models based on the Poisson distribution

could be the work of Church and Gale (1995) who showed empirically that the Poisson distribution was inappropriate for modelling natural text. The Poisson assumes that the mean and variance of the count data are equal, yet in real data, the variance for most words tends to be greater – a phenomenon commonly referred as overdispersion (Ogura et al., 2013). The negative binomial distribution is a suitable alternative in this situation as it has an additional parameter which allows it to account for overdispersion. The frequency of a word in a corpus depends on latent factors such as genre, time, author and topic. The systematic error in the variance estimate when using a Poisson to model word frequency may be due to the assumption that there are no such latent dependencies between words (Church and Gale, 1995). Church and Gale (1995) proposed the K-component mixture of Poissons as a more suitable alternative model for individual word frequencies.

3.3 THE POISSON DISTRIBUTION IN CLASSIFICATION

The conclusion of Church and Gale (1995) regarding the unsuitability of the Poisson distribution for modelling text was further supported in other research such as that of Eyheramendy et al. (2003) and Bouguila (2010). They clearly showed that the multinomial-based supervised classifiers were better for classification than the Poisson-based equivalents. However, in spite of their poor findings, Ogura et al. (2013) and Ogura et al. (2014) were able to create Poisson and Gamma-Poisson naíve Bayes classifiers that outperformed the multinomial-based classifiers upon identifying the need for appropriate document length normalisation. The Poisson classifier was also shown to be able to outperform the negative binomial and K-component mixture of Poissons classifier equivalents (Ogura et al., 2014). Furthermore, the Gamma-Poisson classifier was able to achieve classification performance similar to that of the state-of-the-art classifier, support vector machines (SVM) (Ogura et al., 2013).

As previously mentioned, topic models can be thought of as a combination of dimensionality reduction and clustering. The clustering aspect is captured by the document-by-topic parameter that is learned during the modelling process. The aim of clustering is to group data points into clusters in such a way as to ensure that data points in the same cluster are similar to each other yet as different as possible to data points in other clusters. As the true groupings are not known in advance, clustering can be described as an unsupervised method. In fact, clustering is also called

unsupervised classification (Gámez et al., 2006). On the other hand, when labels for the training dataset are available, supervised classifiers, such as the naïve Bayes classifier and SVMs, become applicable. It has been shown that Poisson-based supervised classifiers for text can outperformed their multinomial counterparts in spite of the poor reputation of the Poisson distribution in text modelling. It is the belief that the success of the Poisson-based supervised classifiers may be transferable to topic models that forms part of the motivation for the consideration of Poisson-based topic models.

3.4 EMPIRICAL ANALYSIS OF WORD OCCURRENCES IN SHORT TEXT

3.4.1 DISTRIBUTION OF WORD OCCURRENCES ACROSS SHORT TEXT DOCUMENTS

One of the ways in which Church and Gale (1995) demonstrated the inappropriateness of the Poisson distributions for modelling word frequencies involved analysing the distribution of word frequency in a corpus that they referred to as the Brown Corpus (Frances and Kucera, 1982). They began by selecting a word from their corpus, specifically the word "said", and then recording the number of documents in which the word was used 0, 1, 2, ..., or 32 times. Figure 3.1 shows a graph of their results. The curve shows the predicted number of documents from a Poisson distribution calculated using the maximum likelihood estimate of the parameter. It is clear from Figure 3.1 that the Poisson does not provide a good fit, thus they proposed the mixture of Poisson distributions or negative binomial distribution as more suitable alternatives.

It is important to note that the documents that were under consideration were long and different results are observed when the same graph is plotted for a word in a short text corpus. To demonstrate this, I selected one of the short text corpora that was selected for this research. The word "jet" was selected and analysed in the documents belonging to class 0 in the Pascal Flickr corpus.[1][2] A similar graph to Figure 3.1 was plotted for the word "jet" and the results obtained are shown in Figure 3.2. The length of the documents belonging to Brown Corpus considered in Figure 3.1 was approximately 2 000 words per document whereas the average length of a document

[1]The Pascal Flickr corpus is discussed in detail in Chapter 5 and summary statistics are given in Table 5.1.

[2]Only documents belonging to a single class are considered because the topic modelling process allocates each document to a class and the parameters are estimated for each topic based on the documents in the class. It is assumed that each document belongs to a single class and that each class corresponds to a "topic".

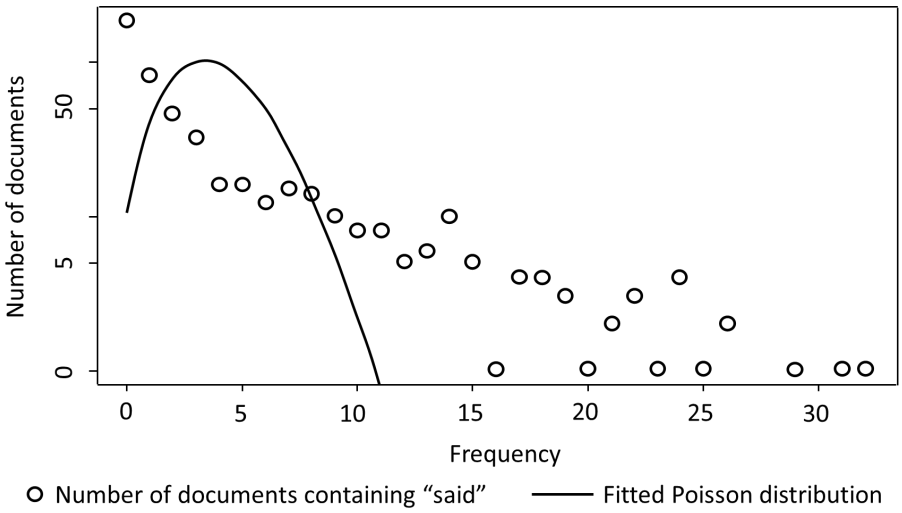


Figure 3.1: The circles show the number of documents that contain the word "said" for different frequencies. The curve denotes predicted frequencies from a Poisson distribution fit to the data. Adapted from (Church and Gale, 1995). Copyright by Cambridge University Press 1995.

in the Pascal Flickr corpus was merely 5 words with a minimum and maximum length of 1 and 19 words, respectively. Taking this into consideration, it is highly unlikely that large frequencies would be observed. From Figure 3.2, the maximum frequency of the word "jet" is 1 and, as we no longer have heavy tails, the predicted values from the Poisson distribution (solid line) are close to the observed values.

The observed frequencies of the all the words in the vocabulary of the Pascal Flickr corpus are shown in Figure 3.3. The Pascal Flickr (class 0) corpus represents 236 of the documents in the full corpus and the full vocabulary contains is 3 132 unique words. The distribution of the different words shown in Figure 3.3 is similar to that of the word "jet" shown in Figure 3.2. The long tails observed in the word "said" from the Brown corpus are not present in this corpus and that the Poisson distributions fit the data well in the majority of cases. Similar results were observed when similar plots were made for other short text datasets that were considered and the results are shown in Section I of the Appendix. It is clear from this analysis that document length has an impact on whether the Poisson distribution will be appropriate for a dataset or not.

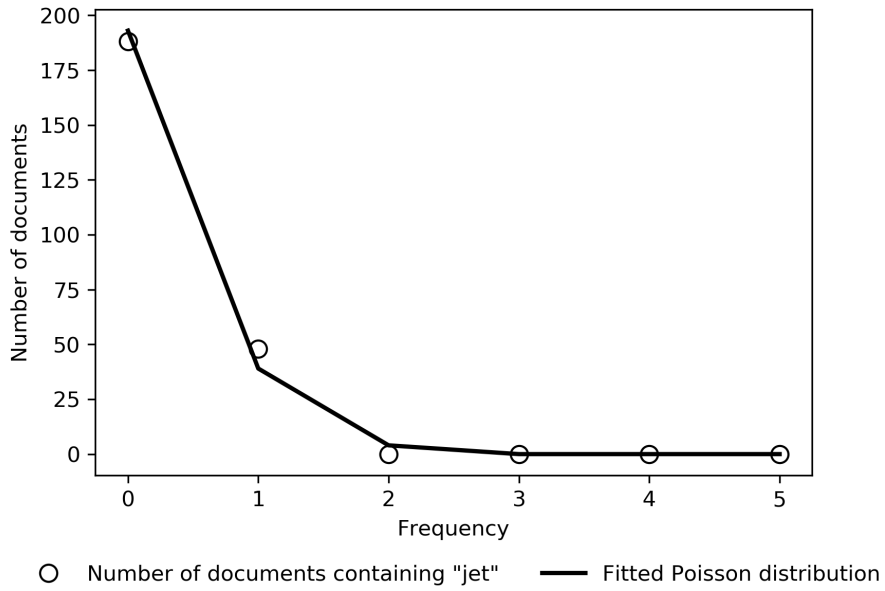


Figure 3.2: The circles show the number of documents that contain the word “jet” for different frequencies in the Pascal Flickr dataset. The curve denotes predicted frequencies from a Poisson distribution fit to the data.

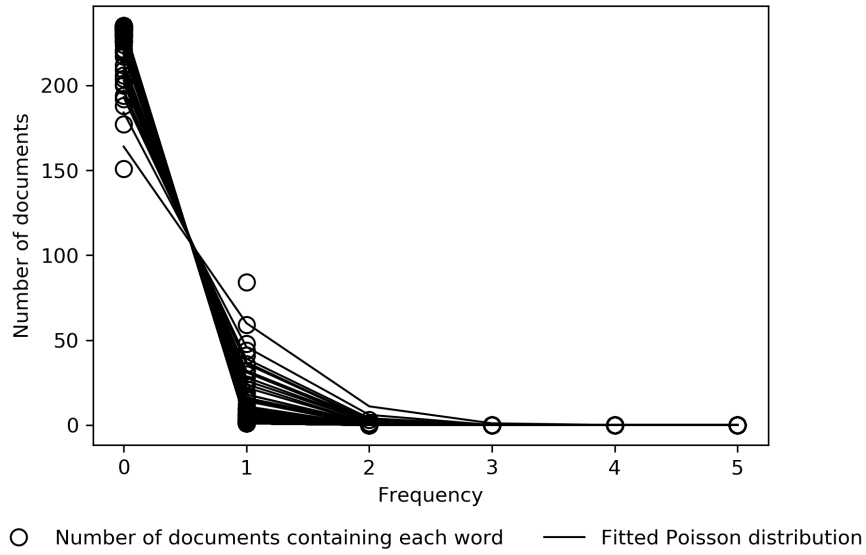


Figure 3.3: The circles represent the number of documents in which each of the words appears 0, 1, 2, ..., 5 times in class 0 of the Pascal Flickr corpus with straight lines indicating the predicted Poisson distribution for each word.

3.4.2 OVERDISPERSION

Church and Gale (1995) stated that, in their experience "the observed variance of the frequency of a word (or ngram) across documents is almost always larger than the mean, and therefore, larger than what would be expected under either the binomial or the Poisson." In the presence of such overdispersion, a natural alternative choice is the negative binomial as it relaxes the assumption made by the Poisson distribution that the mean and variance are equal. Upon fitting a negative binomial distribution to the Brown corpus, Church and Gale (1995) observed the results shown in Figure 3.4 which clearly substantiate the use of a negative binomial distribution.

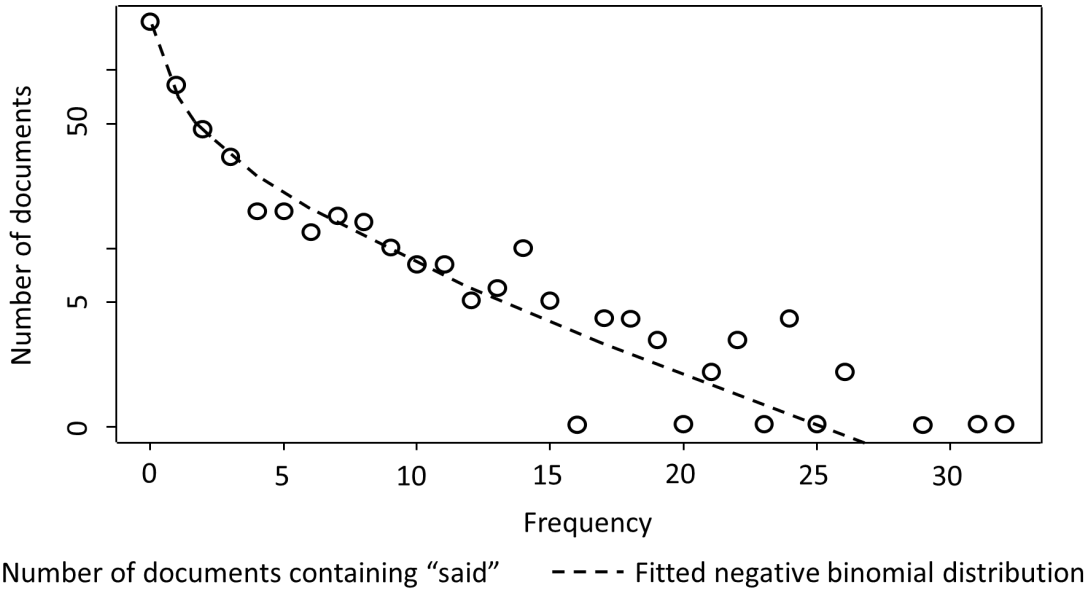


Figure 3.4: This graph shows the number of documents that contain the word "said" for different frequencies (circles) and predicted frequencies from a negative binomial distribution fit to the data (line). Adapted from (Church and Gale, 1995). Copyright by Cambridge University Press 1995.

To assess the extent to which overdispersion was a factor in short text, a comparison of the means and variances for various words from the Pascal Flickr (class 0) corpus that was considered in the previous section was performed. The variance and mean of the occurrences of each word across the corpus are shown in Figure 3.5. Most points fall on or below the 45 degree line, indicating that the variances are mostly less than the mean and, hence, overdispersion may not be a concern in this corpus. In fact, only 3 of the 3 132 words in the vocabulary have occurrences whose variance is larger than the mean. As an example, consider the word "jet" in the Pascal Flickr (class 0) corpus. It occurs 48 times in the 236 documents and the mean and variance of its occurrences

are 0.203 and 0.162, respectively. The absolute difference between the mean and variance is only 0.041. Of the 3 132-word vocabulary, only 300 of the words occur in the 236 documents that make up the Pascal Flickr (class 0) corpus and the average of the absolute differences between the means and variances is only 0.0014. Similar results were observed for other corpora and the results are presented in Section II of the Appendix. In conclusion, overdispersion may not be as much of a concern in short text corpora as it may be in long text corpora. The use of the negative binomial distribution does not appear necessary as the simpler Poisson distribution seems appropriate. In fact, it is ill-advised to use the negative binomial distribution on data whose mean is larger than its variance (ie. in the presence of underdisperseion) (Johnson et al., 2005).

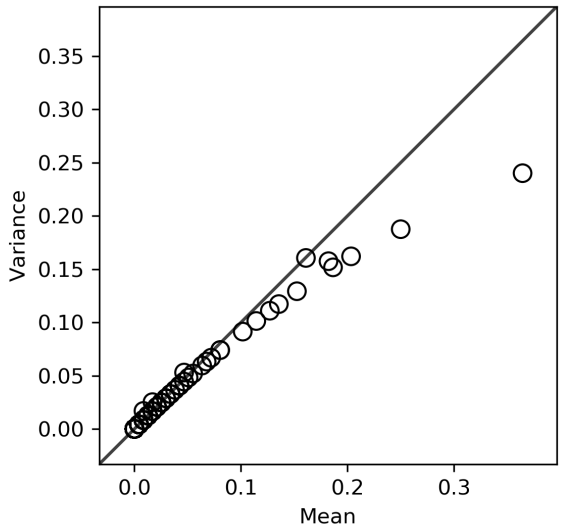


Figure 3.5: Each point represents a word in the vocabulary and the corresponding mean and variance of its occurrences across the documents in class 0 of the Pascal Flickr dataset.

3.4.3 BURSTINESS

The final characteristic that was evident in the Brown corpus was that of burstiness. Bursty words can also be described as being contagious. Such words only appear in a few documents/genres, yet they tend to occur in abundance within the documents that they do appear (Church and Gale, 1995). In the Brown corpus, bursty words were identified to be those which tended to be concentrated in a few documents. Table 3.1 is an excerpt of the results. Each of the words considered occurred

140 or 141 times in the entire corpus, yet only appeared in 38 to 97 documents. Upon fitting a Poisson distribution, it overestimated the number of documents that would be expected to contain these words. The bottom row of Table 3.1 shows the predicted number of documents to be 122.

Table 3.1: Table showing the number of documents which contain words occurring 140 or 141 times in the Brown corpus. Adapted from (Church and Gale, 1995). Copyright by Cambridge University Press 1995.

Word	Frequency	Document frequency (df)
Kennedy	140	38
East	141	62
letter	140	68
production	140	71
son	140	75
Well	140	82
statement	141	83
increased	141	90
results	141	90
thinking	140	97
Predicted df from Poisson distribution		122

A similar analysis on the Pascal Flickr (topic 0) corpus yielded the results in Table 3.2. The word “plane” occurred the most in the corpus, followed by “airplane”, “jet” and so on. In contrast to the Brown corpus, the number of documents containing these highly frequent words is close to the number of occurrences of each word. Thus, there does not appear to be the same burstiness effect as was observed in the Browns corpus. Further more the predicted number of documents predicted by the Poisson is close to the observed document frequency. The average difference between observed and predicted document frequency is 4 for the 10 words shown in Table 3.2, but the average difference for the 300 words that occur in the corpus is 0. Similar results were observed with other corpora and are shown in Section III of the Appendix.

Table 3.2: Table showing the number of documents which contain the top 10 occurring words in the Pascal Flickr (topic 0) corpus.

Word	Frequency	Document frequency (df)	Predicted df from Poisson distribution	$\lvert \text{df} - \text{predicted df} \rvert$
plane	86	85	72	13
airplane	59	59	52	7
jet	48	48	43	5
white	44	44	40	4
flying	43	42	39	3
blue	38	35	35	0
parked	36	36	33	3
sky	32	32	30	2
small	30	30	28	2
runway	27	27	26	1

3.5 DISCUSSION OF EMPIRICAL STUDY

Perhaps it is understandable why Church and Gale (1995) observed the results they did. Word frequency in text is influenced by different factors, such as topic, genre and time. Considering the influence of topic, it makes sense that the frequency of the word "dog", for example, is more likely to have a higher frequency amongst documents about domestic animals than documents that are about finance. When a word is modelled with a *single* Poisson across the entire corpus, dependency on topic is not accounted for. Most topic models such as, LDA and DMM, introduce a latent variable which accounts for topic dependency. The result is that, even though a single distribution is used to model a word, only documents that belong to the same topic are used to estimate the associated parameters. Church and Gale (1995) used the entire corpus of 500 documents which came from 9 very different topics/genres: Press, Religion, Hobbies, Popular Lore, Belle-Lettres, Government and House Organs, Learned, Fiction, and Humor.

It is also not surprising that "Kennedy", who was president at the time the Brown corpus was collected, was more concentrated in the Press documents than other topics/genres, such as Religion or Fiction. The estimate for the Poisson parameter used in Table 3.1 was calculated by dividing the frequency by the number of documents. They used all 500 documents contained in the Brown

corpus to get their estimate and ultimately their prediction of 122. Whereas, to get the predictions in Table 3.2 only the 236 (of 4 834) documents that belonged to the Pascal Flickr (topic 0) corpus were used to estimate the Poisson parameter. Had Church and Gale (1995) only considered the 89 documents from the Press genre, the word "Kennedy" would have had a frequency and document frequency of approximately 118 and 30 respectively. This would have resulted in a predicted document frequency of 65 which is significantly closer to 30 than 122 is to 38. It is likely that this estimate would improve if time was also accounted for in the modelling process.

In conclusion, from the findings of Church and Gale (1995) and other researchers, it is comprehensible that the multinomial distribution would be a preferred choice for modelling text over the Poisson distribution. However, from this previous analysis it is shown that the unfavourable behavior of the Poisson distribution that was identified in long texts does not necessarily occur in short text. In the long text corpus of Church and Gale (1995), the Poisson distribution did not fit individual word occurrences very well and it did not fair well in the presence of overdispersion and burstiness. Yet, the Poisson distribution fitted short text word occurrences better and there was no significant evidence indicating that overdispersion or burstiness were factors.

In remainder of this chapter, the application of the multinomial and Poisson distributions in the field of topic modelling is considered.

3.6 THE POISSON AND MULTINOMIAL DISTRIBUTIONS IN TOPIC MODELLING

Owing to the discrete nature of textual data while using the bag-of-words representation, exponential family PCA (ePCA) is preferred over traditional PCA. When a document is represented as a vector of counts (or sequence of words) it is sensible to assume documents follow a multinomial distribution or product of categorical distributions[3]. Topic models such as LDA and DMM adopt this representation (Buntine and Jakulin, 2006). However, the Poisson distribution is another natural approach to modelling word counts. Models such as the Gamma-Poisson (GaP) model (Canny, 2004) and the Poisson decomposition model (Jiang et al., 2017) follow this approach.

In the sections that follow, a few different topic models are discussed as well as classified according to how they model the observed data. Only models that are probabilistic and assume documents follow either a multinomial (or categorical) distribution or a Poisson distribution are considered.

3.7 MULTINOMIAL-BASED TOPIC MODELS

Two important multinomial-based topic models will be presented in more detail: LDA, as one of the most important state-of-the-art topic models, and GSDMM, which forms the basis of the new short text topic model proposed in this thesis.

3.7.1 LATENT DIRICHLET ALLOCATION

LDA (Blei et al., 2003) is a three level hierarchical Bayesian model which models the assumption that each document is formed through the following generative process (Blei et al., 2003):

1. For all topics, randomly choose $\phi_k \sim Dirichlet(\beta)$.
2. For each document, randomly choose a topic distribution, $\theta_m \sim Dirichlet(\alpha)$.
3. For each word, w_{mn}, in document m:

[3]The product of categorical distributions is equivalent to the multinomial representation as they only differ by a constant. The only difference is that in the product of categorical distributions formulation the combinatoric term, $\frac{n_m!}{x_{m1}!\cdots w_{mV}!}$, from the multinomial distribution is dropped (Buntine and Jakulin, 2006).

a) Randomly choose a topic assignment, $z_{mn} \sim Multinomial(\theta_m)$.

b) Randomly choose a word, $w_{mn} \sim Multinomial(\phi_{z_{mn}})$.

This generative process can easily be summarised visually in a graphical model as in Figure 3.6. Observed variables are represented by shaded circles (nodes) whereas unobserved variables are

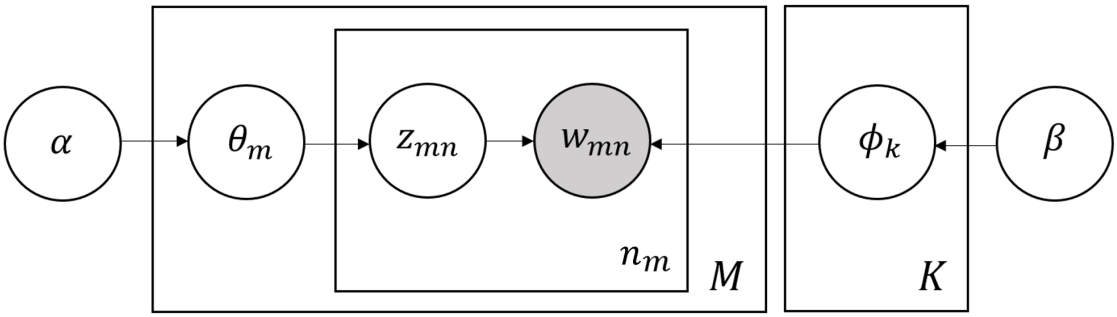


Figure 3.6: Graphical model for LDA. Observed variables are represented by shaded circles (nodes) whereas unobserved variables are represented by unshaded ones. The arrows (edges) represent possible conditioning between variables. The rectangles (plates) indicate a replicated structure.

represented by unshaded ones. The arrows (edges) represent possible conditioning between variables. For example, in Figure 3.6, the arrow between z_{mn} and $\mathbf{w}_{mn}$ indicates that the word, $\mathbf{w}_{mn}$, is conditioned on the topic z_{mn}. The rectangles (plates) indicate a replicated structure, such as the repeated generation of words for the each document, which is indicated by the n_m plate. In practice, symmetric Dirichlet priors are typically used with fixed concentration parameters, α and β, but it is possible to estimate them from the data.

GSDMM assumes all the words in a document belong to a single topic which is dependent on the topic proportions for the entire corpus. LDA, on the other hand, assumes that each word in a document is selected from its own distribution which is dependent on the document-specific topic proportions (Murphy, 2012). Thus, LDA introduces more flexibility than GSDMM as it allows a document to be about *multiple* topics as opposed to only a single topic. It is for this reason that LDA is referred to as an admixture model or mixed membership model (Murphy, 2012). In many cases, especially with longer documents such as books and journal articles, it is more sensible to assume that multiple topics are covered in the body of text as opposed to only a single topic.

The document-by-topic and word-by-topic matrices are the main parameters of interest when fitting the LDA model. This can easily be achieved via Gibbs sampling or variational inference,

but a convenient option that is applicable to LDA is collapsed Gibbs sampling. Typically, a "full" Gibbs sampler would involve sampling all parameters/variables, however collapsed Gibbs sampling provides a simpler estimation procedure as it only involves sampling z_m. The estimates for the parameters θ and Φ can then be inferred after these assignments as they only depend on the word counts that will arise from the sampled topic allocations. The collapsed Gibbs sampler approach involves sampling topic assignments for each word according to the following distribution (Griffiths, 2002; Heinrich, 2005):

$$p(z_{mn} = k|\mathbf{z}_m^{(n)}, \mathbf{w}_m) \propto \frac{n_{kv}^{(n)} + \beta}{n_k^{(n)} + V\beta} \times \frac{n_{mk}^{(n)} + \alpha}{n_m^{(n)} + K\alpha}, \tag{3.3}$$

where

1. $\mathbf{z}_m^{(n)}$ denotes the vector containing the topic assignments of each of the words in the m-th document except that of the current word under consideration, w_{mn},
2. $n_{kv}^{(n)}$ denotes the number of times word v is observed with topic k excluding w_{mn},
3. $n_k^{(n)}$ denotes the total number of words in topic k not including word w_{mn},
4. $n_{mk}^{(n)}$ denotes the number of words in document m that are allocated to topic k excluding the word w_{mn} and
5. $n_m^{(n)}$ denotes the number of words in document m excluding word w_{mn}.

The topic assignments are sampled repeatedly and after a sufficient burn-in period is observed, the elements of the parameters Θ and Φ are then inferred based on the final assignments as follows:

$$\phi_{kv} = \frac{n_{kv} + \beta}{\sum_{v=1}^{V} n_{kv} + V\beta}, \tag{3.4}$$

and

$$\theta_{mk} = \frac{n_{mk} + \alpha}{\sum_{k=1}^{K} n_{mk} + K\alpha}, \tag{3.5}$$

where n_{kv} denotes the number of times word v is observed with topic k and n_{mk} denotes the number of times words in the m-th document are allocated to topic k.

3.7.2 GIBBS SAMPLER DIRICHLET MULTINOMIAL MIXTURE MODEL

GSDMM (Yin and Wang, 2014) is one of the simplest modelling techniques that can be used to uncover the latent topics in a collection of documents. Yin and Wang (2014) termed it GSDMM as they proposed a variant of DMM which made use of a collapsed Gibbs sampler as opposed to the EM algorithm originally used by (Nigam et al., 2000). Owing to GSDMM's assumption that a document can only belong to a single topic, it is often a better option for short text than LDA (Lin et al., 2014; Mazarura and de Waal, 2016). Under this model, the probability of a document, $\mathbf{w}_m$, is expressed as

$$p(\mathbf{w}_m) = \sum_{k=1}^{K} p(\mathbf{w}_m|z_m = k)p(z_m = k), \tag{3.6}$$

where z_m denotes the topic label for the m-th document in the corpus. GSDMM makes the naïve Bayes assumption that words in a document are conditionally independent given the topic. It also assumed that words are exchangeable.[4] Consequently, the probability of a document generated from topic k is given by

$$p(\mathbf{w}_m|z_m = k) = \prod_{w_{mn} \in \mathbf{w}_m} p(w_{mn}|z_m = k). \tag{3.7}$$

It is assumed that

$$\theta \sim Dirichlet(\alpha),$$
$$\Phi \sim Dirichlet(\beta),$$
$$z_m|\theta \sim Multinomial(\theta)$$

and

$$w_{mn}|z_m, \Phi \sim Categorical(\phi_{z_m}).$$

The generative process associated with GSDMM is summarised as follows:

1. For all topics, randomly choose a distribution over words, ϕ_k.

[4]The random variables $X_1, X_2, \ldots, X_n$ are said to exchangeable if $P(x_1, x_2, \ldots, x_n) = P(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$ for all permutations $\pi \in S(n)$ where P denotes the joint distribution of $X_1, X_2, \ldots, X_n$ and $S(n)$ is the group of all permutations acting on $\{1, 2, \ldots, n\}$ (Niepert and Domingos, 2014).

2. For the entire corpus, randomly choose a topic distribution, θ.[5]

3. For each of the m documents in the corpus:

   a) Randomly choose a topic assignment, z_m.

   b) Randomly choose a word, w_{mn}.

Figure 3.7 gives a graphical representation of this generative process. The GSDMM graphical

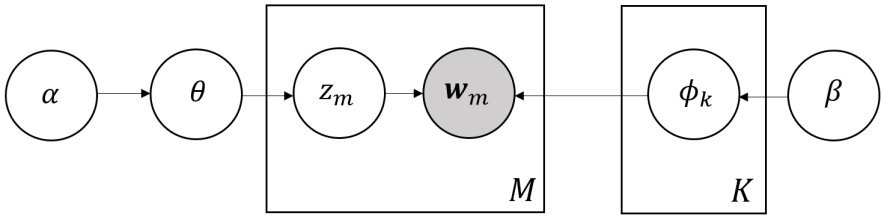


Figure 3.7: Graphical model for GSDMM. Observed variables are represented by shaded circles (nodes) whereas unobserved variables are represented by unshaded ones. The arrows (edges) represent possible conditioning between variables. The rectangles (plates) indicate a replicated structure.

model is similar to that of the LDA graphical model in Figure 3.6. The GSDMM graphical model differs in that it is missing the n_m plate which accounts for the assignment of individual words in the same document to different topics. In addition, the M plate in the LDA graphical includes θ_m. This corresponds to the assignment of different topic proportion for each document. On the other hand, the GSDMM graphical model does not include θ in a plate as it is assumed to be sampled only once for the entire corpus. Unlike LDA, GSDMM assigns each document to a single topic and θ contains the proportion of each topic across the entire corpus.

The collapsed Gibbs sampler for GSDMM involves sampling topic assignments for each document repeatedly until convergence. These topic assignments are sampled from

$$p(z_m|\mathbf{z}^{(m)}, \mathbf{X}) \propto \frac{m_k^{(m)} + \alpha}{M - 1 + K\alpha} \frac{\prod_{w \in \mathbf{w}_m} \prod_{j=1}^{n_{mv}} (n_{kv}^{(m)} + \beta + j - 1)}{\prod_{i=1}^{n_m} (n_k^{(m)} + V\beta + i - 1)}, \tag{3.8}$$

where

1. z_m denotes the topic assignment of the m-th document, $\mathbf{w}_m$,

[5]This definition of θ differs slightly from the definition given in the notation list. Instead of denoting per document topic proportions (as in LDA) it represents the global topic proportions for the entire corpus.

2. $\mathbf{z}^{(m)}$ is the vector of topic assignments for each document excluding $\mathbf{w}_m$,
3. $m_k^{(m)}$ denotes the number of documents in topic k excluding $\mathbf{w}_m$.
4. n_m denotes the number of words in document $\mathbf{w}_m$,
5. n_{mv} denotes the number of times word v occurs in document $\mathbf{w}_m$,
6. $n_{kv}^{(m)}$ denotes the number of occurrences of word v in topic k excluding $\mathbf{w}_m$ and
7. $n_k^{(m)}$ denotes the number of words in topic k excluding $\mathbf{w}_m$.

Based on the final document assignments, the parameter estimates for the elements of Φ and θ can then be calculated as

$$\phi_{kv} = \frac{n_{kv} + \beta}{\sum_{v=1}^{V} n_{kv} + V\beta}$$

and

$$\theta_k = \frac{m_k + \alpha}{\sum_{k=1}^{K} m_k + K\alpha},$$

where n_{kv} denotes the number of occurrences of word v in topic k and m_k denotes the number of documents assigned to topic k.

3.8 POISSON-BASED TOPIC MODELS

The focus of this section is to investigate the usage of the Poisson distribution in the context of topic models.

3.8.1 GAMMA-POISSON MODEL

Non-negative matrix factorisation (NMF) (Lee and Seung, 2001) is regarded as a precursor of the gamma-Poisson (GaP) model (Canny, 2004). The objective under NMF is to display a matrix, the word-by-document count matrix in this case, as a product of two matrices whose values are restricted to be non-negative. In the case of text, these two matrices would be the word-by-topic and topic-by-document matrices. However, unlike the Gamma-Poisson model, NMF is not a probabilistic generative model (Murphy, 2012). As the focus is on probabilistic topic models this model will not be discussed further and attention will be drawn to the GaP model instead.

Under the GaP model, word counts are assumed to come from Poisson distributions and the expected value of the word-by-document frequency matrix is factorised into a product of two matrices, Λ and $\mathbf{L}$. The GaP model, as well as several other models (such as LDA and NMF) all have the following in common: (1) they assume observations are described by a number of discrete variables (2) they try to uncover useful relationships in the data by inferring latent variables for each observation, and relating latent variables to observed variables (Buntine and Jakulin, 2006). Collectively, these groups of methods have been termed discrete components analysis (Buntine and Jakulin, 2006). In general, under discrete components analysis, it is assumed that, for the word-by-document matrix $\mathbf{X}'$,

$$E_{\mathbf{X}'|\Lambda,\mathbf{L}}[\mathbf{X}'] = \Lambda\mathbf{L},$$

where Λ is a $V \times K$ word-by-topic matrix and $\mathbf{L}$ is a $K \times M$ topic-by-document matrix (Buntine and Jakulin, 2006).[6] Under this formulation, the matrix Λ makes it possible for topics to be inferred as is the objective of topic modelling.

An important difference between the GaP model and NMF is that the GaP model assumes a gamma prior on $\mathbf{L}$ whereas NMF assumes no prior (Buntine and Jakulin, 2006). In other words, GaP can be regarded as a Bayesian version of NMF under Kullback-Leibler scoring (Buntine, 2015). It is important to also note the relationship between LDA and GaP. Despite differences in representation, these models are equivalent except that under LDA document length is assumed to be known (Murphy, 2012).

3.8.2 POISSON DECOMPOSITION MODEL

The Poisson decomposition model (PDM) is another Poisson-based alternative for topic modelling (Jiang et al., 2017). PDM models the number of times a topic has been chosen in a document and it models them as variables instead of parameters (Jiang et al., 2017). The GaP model, on the other hand, models each topic's contribution to a document as a non-negative gamma random variable (Canny, 2004). Under the PDM model, a document is represented as a word count matrix as

[6]Discrete components analysis can be regarded as a form of independent components analysis (ICA) which is customised for document data and is more relaxed as it assume $E[\mathbf{X}'] = \Lambda\mathbf{L}$ rather than $\mathbf{X}' = \Lambda\mathbf{L}$ (Buntine and Jakulin, 2006; Canny, 2004).

follows

$$\mathbf{x}_m = \begin{bmatrix} x_{m11} & \cdots & x_{m1V} \\ \cdots & x_{mkv} & \cdots \\ x_{mK1} & \cdots & x_{mKV} \end{bmatrix},$$

where y_{mkv} denotes the number of times that word v is assigned to topic k in document m. Furthermore, it also assumed that

$$x_{mkv} \sim Poisson(\lambda_{mkv}), \tag{3.9}$$

where $\lambda_{mkv} = \lambda_m \times \theta_{mk} \times \phi_{kv}$. λ_m is a document specific constant and it can be shown that the maximum likelihood estimate of λ_m is the length of document m. Each λ_{mkv} is thus directly proportionate to the product of θ_{mk} and ϕ_{kv}, which denote the proportion of topic k in document m and the probability of word v in topic k respectively. Under the assumption that θ and ϕ have symmetric Dirichlet priors with parameters α and β respectively, the following maximum a posteriori estimators are

$$\theta_{mk} = \frac{\sum_{v=1}^{V} x_{mkv} + \alpha}{\sum_{k=1}^{K}\sum_{v=1}^{V} x_{mkv} + K\alpha} \tag{3.10}$$

and

$$\phi_{kv} = \frac{\sum_{m=1}^{M} x_{mkv} + \beta}{\sum_{m=1}^{M}\sum_{v=1}^{V} x_{mkv} + V\beta}. \tag{3.11}$$

3.9 CONCLUSION

The purpose of this chapter was to reintroduce the Poisson distribution as a possible distribution for building new topic models, as well as provide motivation for its use. This chapter addressed the concerns of Church and Gale (1995) by showing the validity of the Poisson distribution for short text. It showed that, unlike the long text considered by Church and Gale (1995), short texts do not necessarily possess issues of overdispersion and burstiness.

This chapter also provided an overview of some of the existing topic modelling strategies. The Poisson-based topic models that have been considered can be classified as long text topic models. The ultimate goal is to produce new models that are suitable for short text. Consequently, this chapter forms the foundation for the new topic model that is presented in Chapter 4.

CHAPTER FOUR

THE GAMMA-POISSON TOPIC MODEL FOR SHORT TEXT

4.1 INTRODUCTION

Under the multinomial distribution, zero word counts are not modelled directly (Inouye et al., 2014); instead they are implicitly modelled in the term probabilities (McCallum and Nigam, 1998). In contrast, the Poisson distribution models them directly as it treats them as observations (Gopalan et al., 2014). In the context of topic modelling, especially with regards to short text, zero counts are a relevant aspect due to the sparsity of such data. The advantage of the Poisson approach to modelling documents is that it is able to capture the numerical characteristics of the documents (Jiang et al., 2017). Another difference between the two models is that the multinomial distribution assumes document length is known whereas the Poisson distribution relaxes this assumption and does not model document length explicitly (Murphy, 2012). This could potentially introduce more flexibility for topic models that are based on the Poisson model. In light of this and the empirical evidence observed in Chapter 3, the new Poisson-based topic model for short text will now be presented.

4.2 THE GAMMA-POISSON MIXTURE MODEL

Table 4.1 shows a summary of the notation that will be used throughout this chapter.

Table 4.1: Notation.

Symbol	Description
M	number of documents in the corpus
V	size of the vocabulary
K	number of topics
n_m	length of m-th document where $m = 1, 2, \ldots, M$
$\mathbf{x}$	collection of documents in the corpus
$\mathbf{x}_m$	m-th document in the corpus
x_{mv}	number of times word v occurs in the m-th document where $v = 1, 2, \ldots, V$
$\mathbf{z}$	vector of topic assignments of each document
z_m	topic assignment of m-th document
m_k	number of documents in topic k where $k = 1, 2, \ldots, K$
n_{kv}	number of times word v is observed in topic k
n_k	number of words in topic k
$a^{(m)}$	if a is a quantity that describes a characteristic of the corpus, $a^{(m)}$ denotes the same characteristic of the corpus excluding the m-th document

The Gamma-Poisson mixture (GPM) topic model is a hierarchical Bayesian model for topic modelling of short text. It assumes that the frequencies of words in a document are independent of each other and that the corpus is a mixture of documents, which belong to different topics. Mixture models are amongst the simplest of latent variable models. Considering the success of GSDMM on short text (Erosheva et al., 2004; Mazarura and de Waal, 2016; Zhao et al., 2011), the GPM topic model makes similar assumptions: (1) Documents are formed from a mixture model and (2) each document belongs to exactly one topic (cluster). This embodies the following probabilistic generative process for a document, $\mathbf{x}_m$:

1. A topic, k, is randomly selected depending on mixing weights $p(z = k)$.

2. A document is then randomly selected from $p(x_m|z = k)$.

Consequently, the likelihood of a document is given by

$$p(\mathbf{x}_m) = \sum_{k=1}^{K} p(\mathbf{x}_m|z=k)p(z=k),$$

where K denotes the total number of topics in the corpus. Similar to GSDMM, GPM makes the Naïve Bayes assumption: given the topic, the frequency of the words in the document are independent of each other. Thus, under GPM the conditional probability of a document given a topic is given by

$$p\left(\boldsymbol{x}_m|z=k\right) = \prod_{v=1}^{v} p\left(x_{mv}|\lambda_{kv}\right),$$

where x_{mv} denotes the frequency of word v in document $\boldsymbol{x}_m$, and λ_{kv} denotes the expected frequency of word v in topic k. The key difference between GPM and GSDMM arises at this point. The GPM assumes the frequencies, x_{mv}, are modelled according to independent Poisson distributions as opposed to modelling the joint distribution of the counts with a multinomial distribution as in the GSDMM. In addition, due to its conjugacy, a gamma prior with shape parameter α_k and scale parameter β_k is imposed on λ_{kv}.

Under GPM, the mixing weights represent the proportion of each of the K topics in the corpus. The topic assignment z of each document is modelled by a multinomial distribution. Thus, $p(z = k) = \pi_k$ where $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$. Furthermore, a Dirichlet prior with parameter γ is imposed on $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_K]$. As GPM is inherently a mixture model, this part of the model is the same as GSDMM. The generative process of GPM can be summarised in a graphical model as in Figure 4.1. Shaded squares are used to indicate fixed parameters. Shaded

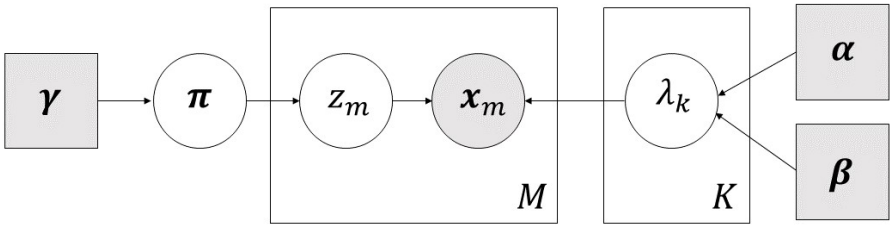


Figure 4.1: Graphical model of GPM. Shaded squares are used to indicate fixed parameters. Shaded circles denote observed variables and unshaded circles represent latent variables. Rectangles represent repeated structures, whereas arrows indicate conditioning.

circles denote observed variables, such as a document, $\mathbf{x}_m$, and unshaded circles represent latent

variables, such as the topic distribution, λ_k. Rectangles represent repeated structures, whereas arrows indicate conditioning, such as the conditioning of documents on both topic distribution and topic assignment. The only random variable that is observed is the corpus, whereas all others are latent variables. In the following section, we will discuss the estimation procedure for the GPM.

4.2.1 THE COLLAPSED GIBBS SAMPLER

A typical Gibbs sampler (Geman and Geman, 1984) requires that each parameter be sampled in turn conditioned on all the other parameters. As the topics are only dependent on the topic assignment of each document, it is only necessary to sample the topic assignments. The conjugacy of the chosen priors introduces analytic tractability that makes it possible to easily integrate out the other parameters that would otherwise need to be sampled. This reduced sampling scheme is called a collapsed Gibbs sampler. One of its advantages is that it tends to be more efficient than its uncollapsed counterpart as the sampling is conducted on a lower dimensional space (Murphy, 2012).

Other estimation techniques, such as the Expectation-Maximisation (EM) algorithm, could have also been considered. However, another benefit of the use of the collapsed Gibbs sampler is that it gives the model the favourable property of being able to automatically select the number of topics. The explanation for how the model is able to do this is discussed in Section 4.3.2.2. In practice, one popular way of selecting the number of topics is achieved via the use of non-parametric topic models (Teh et al., 2006). Thus, although parametric in nature, the GPM model somewhat displays this "non-parametric" behaviour.

In order to estimate the model parameters, the collapsed Gibbs sampler assigns each document to a single topic. This is achieved by sampling from the conditional probability of document $\mathbf{x}_m$ belonging to a class, $p\left(\mathbf{z}_m = z|\mathbf{z}^{(m)}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\right)$, which is given by

$$p\left(\boldsymbol{z}_m = z|\boldsymbol{z}^{(m)}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\right) = \frac{p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{p\left(\boldsymbol{x}, \boldsymbol{z}^{(m)}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\right)} \propto \frac{p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{p\left(\boldsymbol{x}^{(m)}, \boldsymbol{z}^{(m)}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\right)}, \tag{4.1}$$

where the superscript (m) is used to denote that document $\mathbf{x}_m$ is excluded. $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_V]$ and $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_V]$ are the hyperparameters of the gamma prior, whereas $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_K]$ denotes the hyperparameter of the Dirichlet prior.

In order to sample a topic assignment for each document according to Equation 4.1, only the

joint distribution, $p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$, is required:

$$p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \frac{\Delta(\boldsymbol{m} + \boldsymbol{\gamma})}{\Delta(\boldsymbol{\gamma})} \prod_{k=1}^{K} \prod_{v=1}^{V} \frac{\Gamma(n_{kv} + \alpha_v)}{\boldsymbol{x}!\Gamma(\alpha_v)} \times \frac{\beta_v^{n_{kv}}}{(m_k\beta_v + 1)^{n_{kv}+\alpha_v}}. \tag{4.2}$$

By substituting Equation 4.2 into Equation 4.1, under the assumption that $\alpha_v = \alpha$, $\beta_v = \beta$ and $\gamma_k = \gamma$ for all v and k, it follows that Equation 4.1 can be expressed as

$$\begin{aligned} &p\left(\mathbf{z}_m = z|\mathbf{z}^{(m)}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\right) \\ &\propto \tfrac{m_z^{(m)}+\gamma}{M-1+K\gamma} \times \tfrac{\beta^{n_m}}{\boldsymbol{x}_m!} \times \frac{\left(m_z^{(m)}\beta+1\right)^{n_z^{(m)}+V\alpha}}{\left(m_z^{(m)}\beta+\beta+1\right)^{n_z^{(m)}+n_m+V\alpha}} \times \textstyle\prod_{v=1}^{V} \prod_{j=1}^{x_{mv}} \left(n_{zv}^{(m)} + \alpha + j - 1\right). \end{aligned} \tag{4.3}$$

Thus, for each document, a topic is sampled repeatedly until convergence is achieved. Full details of the derivations of Equations 4.1 and 4.3 are shown in Section 4.2.1.1.

Topic k is given by the parameter $\lambda_k = [\lambda_{k1}, \lambda_{k2}, \ldots, \lambda_{kV}]$ where λ_{kv} denotes the expected frequency of word v in topic k and $\lambda_{kv} \geq 0$ for all v and k. These parameters are estimated by

$$\hat{\lambda}_{kv} = \frac{n_{kv} + \alpha_v}{\left(m_k + \frac{1}{\beta_v}\right)}. \tag{4.4}$$

The top words that describe topic k are the words with the highest estimated expected frequencies, $\hat{\lambda}_{kv}$. The derivation of Equation 4.4 is shown in Section 4.2.1.2.

The collapsed Gibbs sampler for the GPM can be summarised as in Algorithm 1. Firstly, the counts m_z, n_z and n_{zv} are initialised to zero. Each document is then randomly assigned to a topic and the aforementioned counts are updated accordingly. From here the algorithm cycles through each document one at a time. Suppose document $\boldsymbol{x}_m$ is the current document under consideration, its current topic label is noted and its contribution to the counts m_z, n_z and n_{zv} are removed accordingly. A new topic assignment is then sampled from Equation 4.3 and the counts are updated. The process must be repeated across the entire corpus until convergence is reached. At this point, the topics can then be found from Equation 4.4.

It was stated earlier that topic models possess characteristics from both clustering and dimensionality reduction techniques: (1) A corpus is represented in a lower dimensional form by a set of topics and, (2) similar to clustering, each document is associated with a single topic or multiple topics depending on the model. The GPM topic model possesses both of these qualities. The first

Algorithm 1: Collapsed Gibbs sampler for GPM.

Data: Corpus, $\mathbf{x}$
Result: Topic labels, $\mathbf{z}$
begin
  Initialise m_z, n_z and n_{zv} to zero for each topic z;
  for *each document* $\mathbf{x}_m$, $m = 1, 2, \ldots, M$ **do**
    randomly sample a topic for $\mathbf{x}_m$;
    $\mathbf{z}_m \leftarrow z \sim$ Categorical$(1/K)$;
    $m_z \leftarrow m_z + 1$ and $n_z \leftarrow n_z + n_m$;
    for *each word frequency* x_{mv} *in* $\mathbf{x}_m$ **do**
      $n_{zv} \leftarrow n_{zv} + x_{mv}$
  for $i = 1, 2, \ldots, I$ *iterations* **do**
    for *each document* $\mathbf{x}_m$, $m = 1, 2, \ldots, M$ **do**
      record the current topic of document
      $\mathbf{x}_m : z = z_m$;
      $m_z \leftarrow m_z - 1$ and $n_z \leftarrow n_z - n_m$;
      for *each word frequency* x_{mv} *in* $\mathbf{x}_m$ **do**
        $n_{zv} \leftarrow n_{zv} - x_{mv}$
      sample a new topic for $\mathbf{x}_m$;
      $z_m \leftarrow z \sim p\left(\mathbf{z}_m = z|\mathbf{z}^{(m)}, \boldsymbol{x}\right)$ (Equation 4.3);
      $m_z \leftarrow m_z + 1$ and $n_z \leftarrow n_z + n_m$;
      for *each word frequency* x_{mv} *in* $\mathbf{x}_m$ **do**
        $n_{zv} \leftarrow n_{zv} + x_{mv}$
end

property is captured by the λ_k parameters. The second is satisfied in Equation 4.3. The upcoming sections give details of the derivation of the collapsed Gibbs sampler.

4.2.1.1 DERIVATION OF THE COLLAPSED GIBBS SAMPLER

Since the topic estimates are only dependent on the topic assignments, it is only necessary to sample the topic assignment for each document. This is achieved by sampling from the conditional probability of a document belonging to a class,

$$p\left(\mathbf{z}_m = z|\mathbf{z}^{(m)}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\right) = \frac{p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{p\left(\boldsymbol{x}, \mathbf{z}^{(m)}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma\right)} \propto \frac{p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{p\left(\boldsymbol{x}^{(m)}, \mathbf{z}^{(m)}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\right)}. \tag{4.5}$$

In order to define Equation 4.5, we need to find $p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$. Owing to conditional independence between $\boldsymbol{x}$ and $\boldsymbol{z}$, it follows that

$$p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}) p(\mathbf{z}|\boldsymbol{\gamma}). \tag{4.6}$$

Since it is assumed that $p(\mathbf{z}|\boldsymbol{\pi})$ is a multinomial and $p(\boldsymbol{\pi}|\boldsymbol{\gamma})$ is a Dirichlet distribution, the second term on the right-hand side of Equation 4.6 can be expressed as follows:

$$\begin{aligned} p(\mathbf{z}|\boldsymbol{\gamma}) &= \int p(\mathbf{z}|\boldsymbol{\pi}) p(\boldsymbol{\pi}|\boldsymbol{\gamma}) d\boldsymbol{\pi} \\ &= \int \frac{1}{\Delta(\boldsymbol{\gamma})} \prod_{k=1}^{K} \pi_k^{m_k + \gamma_k - 1} d\pi_k \\ &= \frac{\Delta(\boldsymbol{m} + \boldsymbol{\gamma})}{\Delta(\boldsymbol{\gamma})}, \end{aligned} \tag{4.7}$$

where $\boldsymbol{m} = [m_1, m_2, \ldots, m_K]$ and m_k denotes the number of documents assigned to the k-th topic, $\Delta(\boldsymbol{\gamma}) = \frac{\prod_{k=1}^{K} \Gamma(\gamma_k)}{\Gamma\left(\sum_{k=1}^{K} \gamma_k\right)}$ and $\Delta(\boldsymbol{m} + \boldsymbol{\gamma}) = \frac{\prod_{k=1}^{K} \Gamma(m_k + \gamma_k)}{\Gamma\left(\sum_{k=1}^{K} (m_k + \gamma_k)\right)}$. The integral is solved by multiplying the equation by $\frac{\Delta(\boldsymbol{m}+\boldsymbol{\gamma})}{\Delta(\boldsymbol{m}+\boldsymbol{\gamma})}$, which results in an integral over a Dirichlet distribution with parameter $\boldsymbol{m} + \boldsymbol{\gamma}$.

The first term on the right-hand side of Equation 4.6, can be expressed as

$$p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \int p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\lambda}) p(\boldsymbol{\lambda}|\boldsymbol{\alpha}, \boldsymbol{\beta}) d\boldsymbol{\lambda}. \tag{4.8}$$

Under GPM, documents and words are assumed to be independent. In addition, the word counts are assumed to follow a Poisson distribution. Thus, given the topics, the corpus can be modelled as

$$p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\lambda}) = \prod_{m=1}^{M} \prod_{v=1}^{V} p\left(x_{mv}|\lambda_{kv}\right) = \prod_{m=1}^{M} \prod_{v=1}^{V} \frac{\lambda_{kv}^{x_{mv}} e^{-\lambda_{kv}}}{x_{mv}!}. \tag{4.9}$$

From Equation 4.9, notice that the term $p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\lambda})$ in Equation 4.8 is a product over M and V. However, the term $p(\boldsymbol{\lambda}|\boldsymbol{\alpha}, \boldsymbol{\beta})$ in Equation 4.8, which is assumed to be a product of independent gamma distributions, is a product over K and V. In order to simplify Equation 4.8, it is necessary to re-express $p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\lambda})$ as a product over K and V. By expanding these product terms and carefully analysing the relationship between the terms, it was found to be possible to re-express Equation 4.9 in this manner by introducing m_k, the number of documents assigned to the k-th

topic, and n_{kv}, the number of times word v is observed in topic k. Thus, Equation 4.9 becomes

$$p(\boldsymbol{x}|\mathbf{z},\boldsymbol{\lambda}) = \prod_{k=1}^{K}\prod_{v=1}^{V}\frac{\lambda_{kv}^{n_{kv}}e^{-m_k\lambda_{kv}}}{\boldsymbol{x!}}, \tag{4.10}$$

where $\boldsymbol{x}! = \prod_{m=1}^{M}\prod_{v=1}^{V}x_{mv}$. By assuming a gamma distribution for $\boldsymbol{\lambda}$ and substituting Equation 4.10 into Equation 4.8, we obtain

$$\begin{aligned}
p(\boldsymbol{x}|\mathbf{z},\boldsymbol{\alpha},\boldsymbol{\beta}) &= \int_{-\infty}^{\infty} p(\boldsymbol{x}|\mathbf{z},\boldsymbol{\lambda})p(\boldsymbol{\lambda}|\boldsymbol{\alpha},\boldsymbol{\beta})d\boldsymbol{\lambda} \\
&= \int_{-\infty}^{\infty}\prod_{k=1}^{K}\prod_{v=1}^{V}\frac{\lambda_{kv}^{n_{kv}}e^{-m_k\lambda_{kv}}}{\boldsymbol{x}!}\times\frac{\lambda_{kv}^{\alpha_v-1}e^{-\frac{\lambda_{kv}}{\beta_v}}}{\Gamma(\alpha_v)\,\beta_v^{\alpha_v}}d\lambda_{kv} \\
&= \prod_{k=1}^{K}\prod_{v=1}^{V}\frac{1}{\boldsymbol{x}!\Gamma(\alpha_v)\,\beta_v^{\alpha_v}}\int_0^{\infty}\lambda_{kv}^{n_{kv}+\alpha_v-1}e^{-\lambda_{kv}\left(m_k+\frac{1}{\beta_v}\right)}d\lambda_{kv} \\
&= \prod_{k=1}^{K}\prod_{v=1}^{V}\frac{1}{\boldsymbol{x}!\Gamma(\alpha_v)\,\beta_v^{\alpha_v}}\times\Gamma(n_{kv}+\alpha_v)\left(\frac{\beta_v}{m_k\beta_v+1}\right)^{n_{kv}+\alpha_v} \\
&= \prod_{k=1}^{K}\prod_{v=1}^{V}\frac{\Gamma(n_{kv}+\alpha_v)}{\boldsymbol{x}!\Gamma(\alpha_v)}\times\frac{\beta_v^{n_{kv}}}{(m_k\beta_v+1)^{n_{kv}+\alpha_v}}.
\end{aligned} \tag{4.11}$$

The integral is solved by multiplying the equation by a constant equal to 1, which is $\Gamma(n_{kv}+\alpha_v)\left(\frac{\beta_v}{m_k\beta_v+1}\right)^{n_{kv}+\alpha_v}$ divided by itself in this case. The result is an integral over a gamma distribution with parameters $n_{kv}+\alpha_v$ and $\frac{\beta_v}{m_k\beta_v+1}$. By substituting Equation 4.7 and 4.11, Equation 4.6 can now be written as

$$\begin{aligned}
p(\boldsymbol{x},\mathbf{z}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) &= p(\boldsymbol{x}|\mathbf{z},\boldsymbol{\alpha},\boldsymbol{\beta})p(\mathbf{z}|\boldsymbol{\gamma}) \\
&= \frac{\Delta(\boldsymbol{m}+\boldsymbol{\gamma})}{\Delta(\boldsymbol{\gamma})}\prod_{k=1}^{K}\prod_{v=1}^{V}\frac{\Gamma(n_{kv}+\alpha_v)}{x!\Gamma(\alpha_v)}\times\frac{\beta_v^{n_{kv}}}{(m_k\beta_v+1)^{n_{kv}+\alpha_v}}.
\end{aligned} \tag{4.12}$$

The derivation of the conditional distribution in Equation 4.5 can now be concluded by substituting

Equation 4.12 and applying the property of the Γ function, $\frac{\Gamma(x+m)}{\Gamma(x)} = \prod_{j=1}^{m}(x+j-1)$, as follows

$$
\begin{aligned}
& p\left(\mathbf{z}_m = z|\mathbf{z}^{(m)}, \boldsymbol{x}\right) \\
& \propto \frac{p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{p\left(\boldsymbol{x}^{(m)}, \mathbf{z}^{(m)}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\right)} \\
& = \frac{\Delta(\boldsymbol{m}+\boldsymbol{\gamma})}{\Delta\left(\boldsymbol{m}^{(m)}+\boldsymbol{\gamma}\right)} \times \prod_{v=1}^{V}\left(\frac{\Gamma\left(n_{zv}+\alpha_v\right)}{\Gamma\left(n_{zv}^{(m)}+\alpha_v\right)}\right)\left(\frac{\beta_v^{n_{zv}}}{\beta_v^{n_{zv}^{(m)}}}\right)\left(\frac{x^{(m)}!\Gamma\left(\alpha_v\right)}{x!\Gamma\left(\alpha_v\right)}\right) \\
& \qquad \times \left(\frac{\left(m_z^{(m)}\beta_v+1\right)^{n_{zv}^{(m)}+\alpha_v}}{\left(m_z\beta_v+1\right)^{n_{zv}+\alpha_v}}\right) \\
& = \frac{m_z^{(m)}+\gamma_z}{M-1+\sum_{k=1}^{K}\gamma_k} \times \prod_{v=1}^{V}\prod_{j=1}^{x_{mv}}\left(n_{zv}^{(m)}+\alpha_v+j-1\right) \times \beta_v^{x_{mv}} \times \frac{1}{\boldsymbol{x}_m!} \\
& \qquad \times \frac{\left(m_z^{(m)}\beta_v+1\right)^{n_{zv}^{(m)}+\alpha_v}}{\left(m_z^{(m)}\beta_v+\beta_v+1\right)^{n_{zv}^{(m)}+x_{mv}+\alpha_v}}, \qquad (4.13)
\end{aligned}
$$

where $n_{zv} = n_{zv}^{(m)} + x_{mv}$ and $m_z = m_z^{(m)} + 1$. If it is assumed that $\alpha_v = \alpha$, $\beta_v = \beta$ and $\gamma_k = \gamma$ for all v and k, then Equation 4.13 simplifies to

$$
\begin{aligned}
& p\left(\mathbf{z}_m = z|\mathbf{z}^{(m)}, \boldsymbol{x}\right) \\
& \propto \tfrac{m_z^{(m)}+\gamma_z}{M-1+K\gamma} \times \tfrac{\beta^{n_m}}{\mathbf{x}_m!} \times \frac{\left(m_z^{(m)}\beta+1\right)^{n_z^{(m)}+V\alpha}}{\left(m_z^{(m)}\beta+\beta+1\right)^{n_z^{(m)}+n_m+V\alpha}} \\
& \qquad \times \textstyle\prod_{v=1}^{V}\prod_{j=1}^{x_{mv}}\left(n_{zv}^{(m)}+\alpha+j-1\right),
\end{aligned}
$$

where n_m is the length of the m-th document and $\sum_{v=1}^{V} n_{zv} = n_z$ denotes the total number of documents in topic k. This concludes the derivation of Equation 4.3.

4.2.1.2 DERIVATION OF TOPIC REPRESENTATION

After sampling from Equation 4.3 until convergence, the λ parameters, which produce the topic distributions, are estimated by the posterior means. The posterior is given by

$$
p\left(\lambda_{kv}|\boldsymbol{x}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) \propto \lambda_{kv}^{n_{kv}+\alpha_v-1} e^{-\lambda_{kv}\left(m_k+\frac{1}{\beta_v}\right)}.
$$

It follows that $\lambda_{kv} \sim GAM\left(n_{kv} + \alpha_v, \frac{\beta_v}{m_k \beta_{v+1}}\right)$ and the topic distribution estimates are given by

$$\hat{\lambda}_{kv} = \frac{n_{kv} + \alpha_v}{\left(m_k + \frac{1}{\beta_v}\right)}.$$

The top words that describe topic k are the words with the highest expected frequencies, $\hat{\lambda}_{kv}$.

4.3 DISCUSSION

This section presents a discussion of the important aspects related to the application of the GPM. It also highlights the impact of the different parameters that are used in the model.

4.3.1 DOCUMENT LENGTH NORMALISATION

Since the Poisson distribution gives the probability of observing a given number of events in a fixed interval, it is necessary to normalise the lengths of the documents. There are different strategies that can be used to achieve this. Two strategies are discussed and compared below.

4.3.1.1 METHOD 1: DIRECT DOCUMENT LENGTH NORMALISATION

This strategy involves replacing the word frequencies, x_{mv}, with

$$x_{mv}^{\text{new}} = \frac{N x_{mv}}{\sum_{v=1}^{V} x_{mv}},$$

where N denotes a predefined length (Ogura et al., 2013). The impact of this normalisation is that each document will have a length of N words. In order to investigate the performance of the model under different choices of N, specifically $N = 10, 20, 30$, the GPM was applied to the Tweet, Pascal Flicker and Search Snippets datasets (discussed in Section 5.2).

The performance of the model was assessed based on two measures. The first involves comparing the number of topics found by the model against the number of topics found by the human annotators who were used in the compilation of the datasets (this value will be referred to as the true K). The second performance measure is average topic coherence (discussed in Section 5.4), which measures topic quality. The higher the score, the better the performance.

Figure 4.2 shows the number of topics found, coherence and runtime (in minutes) of the GPM

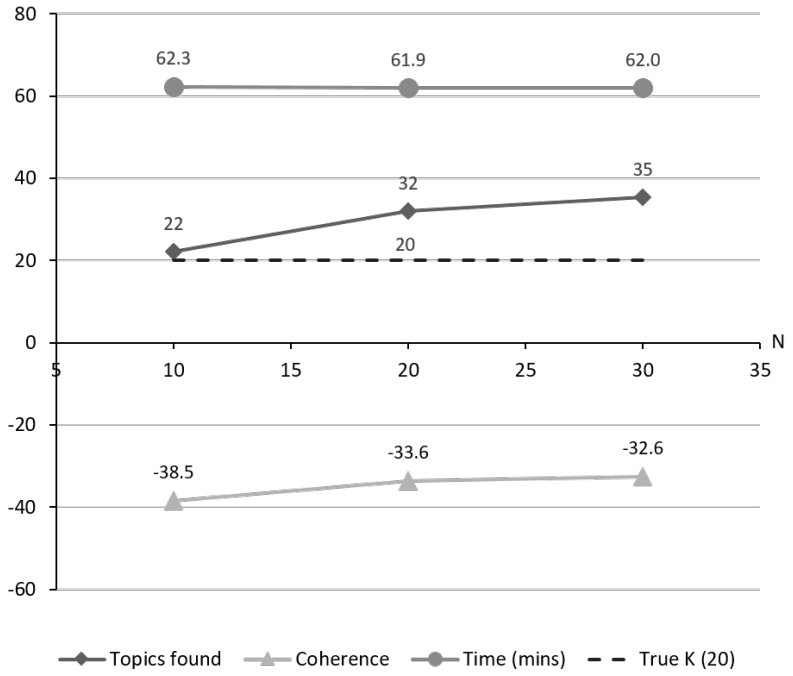


Figure 4.2: Number of topics found, average coherence and runtime of the GPM on the Pascal Flickr corpus for $N = 10, 20, 30$. The runtime of the model is not significantly affected by N. However, as N increases, the average coherence scores improve whilst the number of topics found moves further away from the true K.

for the different values of N for the Pascal Flickr corpus. In general, the runtime of the model is not significantly affected by N. However, as N increases, the average coherence scores improve whilst the number of topics found moves further away from the true K. The results for the other datasets are shown in Section IV of the Appendix.

Based on these results and those observed on the other datasets, $N = 20$ appears to result in a good trade-off between the different performance measures across the different datasets. Thus, going forward all experiments performed with the GPM with this normalisation are performed with $N = 20$ (with each x_{mv}^{new} rounded to the nearest integer).

4.3.1.2 METHOD 2: MODELLING DOCUMENT LENGTH IN THE TOPIC MODEL

The second normalisation method can be found in the paper of Church and Gale (1995). This approach involves modelling the frequency of each word, x_{mv}, as

$$x_{mv}|z = k \sim Poi(N_m \lambda_{kv}),$$

where N_m denotes the number of words in the m-th document, as opposed to

$$x_{mv}|z = k \sim Poi(\lambda_{kv})$$

as in the previously proposed derivation. This change affects the collapsed Gibbs sampler derivation. In the new derivation which can be found in Section V of the Appendix, Equation 4.3 would then be replaced by

$$\begin{aligned} & p\left(\mathbf{z}_m = z|\mathbf{z}^{(m)}, \boldsymbol{x}\right) \\ & \propto \frac{m_z^{(m)}+\gamma_z}{M-1+K\gamma} \times \frac{\beta^{n_m}}{\mathbf{x}_m!} \times \frac{\left(n_z^{(m)}+1\right)^{n_z^{(m)}+V\alpha}}{\left(n_z^{(m)}\beta+N_m\beta+1\right)^{n_z^{(m)}+n_m+V\alpha}} \\ & \qquad \times \prod_{v=1}^{V}\prod_{j=1}^{x_{mv}}\left(n_{zv}^{(m)}+\alpha+j-1\right), \end{aligned} \tag{4.14}$$

where $n_z^{(m)}$ denotes the number of words assigned to topic z excluding the m-th document. Modelling document length in this manner alleviates the need to select a value for N as in the previous method. This simplifies its application.

4.3.1.3 COMPARISON OF NORMALISATION METHODS

Figure 4.3 shows a comparison of the performance of the different normalisation methods on the Pascal Flickr dataset. The results on the other corpora are similar and are shown in Section VI of the Appendix. On the datasets that were considered, normalisation method 1 ran faster than method 2. Furthermore, normalisation method 1's coherence was better and the number of topics found was closer to the true K. In light of these results, normalisation method 1 was used in all experiments going forward.

4.3.2 MEANING OF HYPERPARAMETERS

The GPM has 3 hyperparameters that must be selected prior to the application of the model: $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_V]$ and $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_V]$ are the hyperparameters of the gamma prior, whereas $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_K]$ denotes the hyperparameter of the Dirichlet prior. Although it is possible to estimate parameter values from the data, this can be computationally expensive. Consequently, in practice researchers will often use symmetric fixed priors. In this section, the meaning of each of the hyperparameters is discussed. In addition, recommendations for choosing

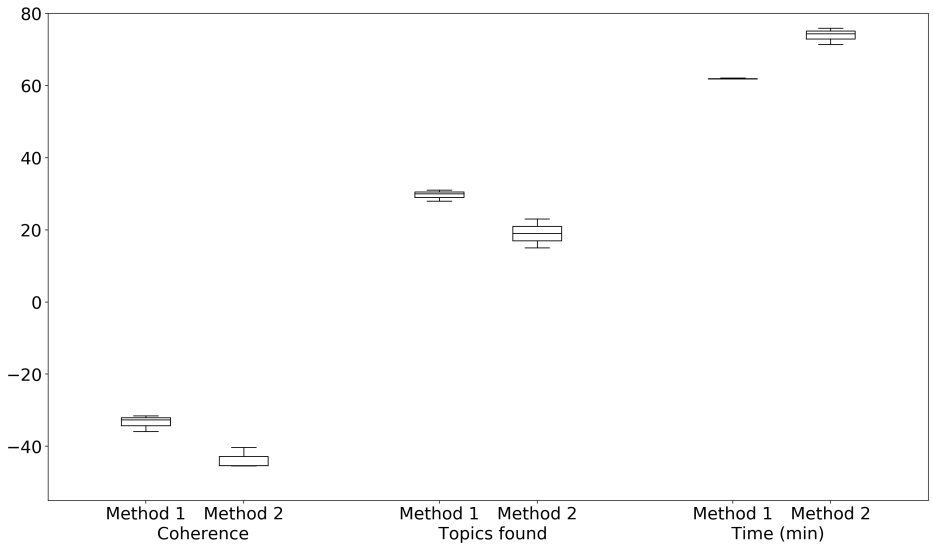


Figure 4.3: Comparison of average coherence, number of topics found and runtime of GPM under normalisation methods 1 (direct document length normalisation) and 2 (modelling document length in the topic model) on the Pascal Flickr corpus (True K = 20).

their values are also made.

4.3.2.1 MEANING OF γ

The GPM topic model assumes

$$\mathbf{z} \sim Cat(\boldsymbol{\pi}),$$

with a prior of the form

$$\boldsymbol{\pi} \sim Dir(\boldsymbol{\gamma}).^{1}$$

The topic assignment of a document is given by $\mathbf{z}$. The selection of $\mathbf{z}$ is governed by $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_K]$ where π_k gives the proportion of the corpus that belongs to topic k. π_k can also be thought of as the probability of a document belonging to topic k. The implementation of a full Gibbs sampler would have required that $\boldsymbol{\pi}$ also be sampled. However, $\boldsymbol{\pi}$ can be determined from the sampled topic labels and it is simple to integrate out due to the conjugacy of the Dirichlet distribution to the multinomial/categorical distribution. This makes it convenient to opt for a collapsed

[1] In (Yin and Wang, 2014), this $\boldsymbol{\gamma}$ is equivalent to the $\boldsymbol{\alpha}$ used in GSDMM.

Gibbs sampler and it means that one only needs to determine values for the hyperparameters.

As previously stated, $\boldsymbol{\gamma} = [\gamma_1, \gamma_2 \ldots, \gamma_K]$ is the hyperparameter of the Dirichlet distribution. Values of γ_k less than 1 lead to a more sparse distribution of probability; that is, many of the π_k will be near zero whilst most of the probability is allocated to only a few of the π_k. Conversely, when the γ_k are larger, the result is a more uniform distribution of probability. This γ_k parameter influences the prior probability of a document belonging to a topic. It appears in the first part of Equation 4.3 in the term $\frac{m_z^{(m)}+\gamma}{M-1+K\gamma}$. If all the γ_k are set to zero, a topic without a document assigned to it will never be chosen. This is because an empty topic will have $m_z^{(m)}$ (the number of documents contained in topic z) equal to zero. This will force the numerator to be zero for such topics and make Equation 4.3 equal zero. On the other hand, as the γ_k increase the probability of choosing an empty topic increases. Note, in all experiments, a symmetric Dirichlet prior was used, i.e. all the γ_k $(k = 1, 2, .., 3)$ were set to the same value, γ.

4.3.2.2 MEANING OF α AND β

The parameters α_v and β_v represent the shape and scale parameters of the gamma distribution, respectively. For simplicity, it is assumed that $\alpha_v = \alpha$ and $\beta_v = \beta$ for all v. As previously mentioned, each word count, x_{mv}, is assumed to follow a Poisson distribution conditioned on the document's topic assignment,

$$x_{mv}|z = k \sim Poi(\lambda_{kv}),$$

with prior

$$\lambda_{kv} \sim Gam(\alpha, \beta).$$

The gamma distribution is a family of skewed distributions. Given the sparsity of the text data under consideration, it makes sense that the expected frequencies of words be modelled by a distribution that assigns most of its probability near zero.

Consider Equation 4.3:

$$\begin{aligned} &p\left(\mathbf{z}_m = z|\mathbf{z}^{(m)}, \boldsymbol{x}\right) \\ &\propto \frac{m_z^{(m)}+\gamma}{M-1+K\gamma} \times \frac{\beta^{n_m}}{\boldsymbol{x}_m!} \times \frac{\left(m_z^{(m)}\beta+1\right)^{n_z^{(m)}+V\alpha}}{\left(m_z^{(m)}\beta+\beta+1\right)^{n_z^{(m)}+n_m+V\alpha}} \times \prod_{v=1}^{V}\prod_{j=1}^{x_{mv}}\left(n_{zv}^{(m)}+\alpha+j-1\right). \end{aligned}$$

This is the equation from which the topics are sampled.

The scale parameter of a gamma distribution does not change the shape of the data, but simply scales the graph. Consequently, the gamma parameter does not have significant impact on Equation 4.3 and hence, does not greatly impact the performance of the model. Empirical evidence of this is shown in Section 5.5.4.

As the name suggests, the shape parameter, α, influences the shape of the gamma distribution. The gamma distribution for different selections of α can be seen in Figure 5.10. If α is set to equal zero, the last term in Equation 4.3 will equal zero if $n_{zv}^{(m)} = 0$. In other words, if a topic is missing even a single word that is contained in a document, then that document can never be assigned to the topic even if the document is similar to other documents contained in the topic and should actually be assigned to the topic. Fortunately, since α can take on any positive value, this can be avoided by selecting a nonzero value. For small values of α, the probability of a document belonging to a topic is more sensitive to n_{kv}, the number of times word v is observed in topic k. This means that, when a topic has more words in common with a document it is more likely to be assigned to that topic. On the other hand, when α is large, the probability of being assigned to a topic is less sensitive to n_{kv}. Instead, the probability is influenced more by the first term in Equation 4.3 which is dependent on m_k, the number of documents in topic k. As a result, a topic with more documents is likely to get larger since Equation 4.3 will assign more probability to topics that contain more documents. Empirical evidence of this can be seen in Figure 5.9 where the number of topics found is recorded for different values of α; for small values of α, GPM found more topics whilst, for large values, it tended to assign all the documents to a single topic.

4.3.3 SELECTION OF PRIOR VALUES

Selecting optimal values for the hyperparamters can be challenging due to the unsupervised nature of the topic model. However, for simplicity and also following the common practice in the literature, specific fixed values were used in all experiments (Rigouste et al., 2007). Firstly, the γ parameter was set to 0.1 as was proposed for the GSDMM, since this parameter is equivalent to the α parameter in GSDMM. As for α and β, from the previous discussion and empirical results, it is clear that the selection of α is of vital importance, whereas the choice of β is not as crucial.

In the Bayesian literature, the gamma distribution with shape and *rate* parameters both equal to 0.001 is a commonly used non-informative prior (Lee and Wagenmakers, 2014). In the GPM derivation, the gamma distribution is parameterised by shape and *scale* parameters. Despite using

the scale-parameter instead of the rate-parameter formulation, it is shown empirically in Section 5.5.4 that setting both the shape and scale parameters to 0.001 yielded better results than the other values that were considered. In fact, it is shown in Section 5.5.5 that for this choice of values, the GPM outperformed GSDMM on the datasets that were considered. In light of this, $\alpha = 0.001$ and $\beta = 0.001$ are used in all experiments (unless specified otherwise).

4.4 RELATIONSHIP BETWEEN GPM AND DIFFERENT TOPIC MODELS

Many topic models can be considered as versions of discrete components analysis (Buntine and Jakulin, 2006). Such models express the expected values of the document vector as a product of matrices. This product typically consists of one matrix which captures the distribution of words in a topic, and another which accounts for the contribution of each topic to the document. In this section, this representation is explored in detail so as to demonstrate the relationships between GPM and different models.

In general, many of the models assume each document, $\mathbf{x}_m$, can be regarded as having an expected value of the form

$$E(\mathbf{x}_m) = \theta \mathbf{l} = \begin{bmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \theta_{vk} & \vdots \\ \theta_{V1} & \cdots & \theta_{VK} \end{bmatrix} \begin{bmatrix} l_1 \\ \vdots \\ l_K \end{bmatrix}. \tag{4.15}$$

The matrix θ accounts for the word distribution of a topic whilst the vector $\mathbf{l}$ captures the contribution of each topic to document $\mathbf{x}_m$. The different models make different assumptions regarding $\mathbf{x}_m$, θ and $\mathbf{l}$.

4.4.1 MULTINOMIAL-TYPE MODELS

4.4.1.1 MULTINOMIAL PCA

Multinomial PCA (mPCA) (Buntine, 2002) it is assumed that

$$\mathbf{x}_m \sim Multinomial(\theta \mathbf{l}),$$

where the columns of θ are assumed to be normalised such that $0 \leq \theta_{vk} \leq 1$ and $\sum_{v=1}^{V} \theta_{vk} = 1$ for all k and v. It is also assumed that the elements of $\mathbf{l}$ are such that $0 \leq l_k \leq 1$ and $\sum_{k=1}^{K} l_k = 1$ for all k, and that

$$\theta_k = [\theta_{1k}, \theta_{2k}, \cdots, \theta_{Vk}] \sim Dirichlet(\omega)$$

and

$$\mathbf{l} \sim Dirichlet(\psi)$$

where ω and ψ are the parameters for the Dirichlet distributions.

4.4.1.2 LATENT DIRICHLET ALLOCATION

LDA is the same as multinomial PCA except that each document is modelled as a sequence of words instead of a vector of word counts. In other words, a document is represented as a product of categorical distributions. Consequently, the only difference between the LDA and mPCA models is that under the LDA model the combinatoric term, $\frac{L!}{w_1! \cdots w_V!}$, from the multinomial distribution of mPCA is dropped (Buntine and Jakulin, 2006).

4.4.1.3 MULTINOMIAL MIXTURE MODEL

The multinomial mixture model can also be expressed the form equation 4.15. θ and $\mathbf{x}_m$ are both modelled in the same way as mPCA. However, $\mathbf{l}$ is a vector in which exactly one element is equal to one and the rest are all zeros. This is often referred to as a 1-of-K or one-hot encoding. Thus, $\mathbf{l}$ is modelled as

$$\mathbf{l} \sim Categorical(\pi),$$

where π denotes the parameter for the categorical distribution. It follows that

$$E(\mathbf{x}_m) = \theta_k,$$

where θ_k denotes the k-th column of θ (Buntine and Jakulin, 2006).

4.4.2 POISSON-TYPE MODELS

4.4.2.1 GAMMA-POISSON MODEL

The gamma-Poisson (GaP) model introduced by Canny (2004) should not be confused with the GPM introduced in this thesis. The key difference between the models is that the GPM assumes each document belongs to a single topic whereas the GaP model assumes multiple topics can contribute to a document.

The gamma-Poisson model assumes that all of the elements of $\mathbf{x}_m$ are independently Poisson distributed. That is,

$$x_{mv} \sim Poisson((\theta \mathbf{l})_{mv}),$$

where $(\theta \mathbf{l})_{mv}$ denotes the element in position (m, v) of $\theta \mathbf{l}$. The matrix θ follows the same distributional assumptions as mPCA. However, the elements of $\mathbf{l}$, denoted l_k, are such that

$$l_k \sim Gamma(\alpha_k, \beta_k).$$

It can be shown that, if all the β_k are equal, then the GaP model is equivalent to the mPCA model (Buntine and Jakulin, 2006).

4.4.2.2 POISSON DECOMPOSITION MODEL

The PDM model assumes a similar representation as in 4.15 as it only differs by a constant.

$$E(\mathbf{x}_m) = \lambda_m \theta \mathbf{l}. \tag{4.16}$$

As was mentioned in section 3.8.2, PDM assumes

$$x_{mv} = \sum_k x_{mkv} \sim Poisson(\sum_k \lambda_{mkv})$$

where $\lambda_{mkv} = \lambda_m \times \theta_{mk} \times \phi_{kv}$. The distributional assumptions on θ and $\mathbf{l}$ are exactly the same as in mPCA.

4.4.2.3 GAMMA-POISSON MIXTURE TOPIC MODEL

GPM is similar to the multinomial mixture in that $\mathbf{l}$ will also be distributed according to a categorical distribution. However, instead of assuming the data comes from a multinomial distribution, the data is follows a Poisson such that

$$E(\mathbf{x}_m) = \boldsymbol{\theta}_k.$$

All of the models, including GPM can be summarised as in Table 4.2.

Table 4.2: Summary of the distributions of the data, θ and $\mathbf{l}$ assumed by each model.

Multinomial-type models			
Model	$\mathbf{x}_m$	θ_k	$\mathbf{l}$
mPCA	$Multinomial(\theta\mathbf{l})$	$Dirichlet(\omega)$	$Dirichlet(\psi)$
LDA	$\prod Categorical(\theta\mathbf{l})$	$Dirichlet(\omega)$	$Dirichlet(\omega)$
Multinomial mixture	$Multinomial(\theta\mathbf{l})$	$Dirichlet(\omega)$	$Categorical(\pi)$
Poisson-type models			
Model	x_{mv}	θ_k	$\mathbf{l}_k$
GaP	$Poisson((\theta\mathbf{l})_{mv})$	$Dirichlet(\omega)$	$Gamma(\alpha_k, \beta_k)$
PDM	$Poisson((\lambda_m\theta\mathbf{l})_v)$	$Dirichlet(\omega)$	$Dirichlet(\omega)$
GPM	$Poisson((\boldsymbol{\theta}\mathbf{l})_v)$	$Dirichlet(\omega)$	$Categorical(\pi)$

4.5 CONCLUSION

In this chapter, the GPM was introduced as a new topic model for short text. It presented details of the derivation of a collapsed Gibbs sampler, which was possible due to the conjugacy between the chosen distributions. In addition, it gave the GPM the favourable characteristic of being able to automatically estimate the number of topics contained in a corpus.

As previously mentioned, topic models display clustering and dimensionality reduction properties. In line with this, it was shown that the GPM displayed these characteristics through Equation 4.3 and Equation 4.4. By iteratively sampling topic lables for each document from Equation 4.3, the clustering aspect was satisfied. The dimensionality reduction aspect was covered via the estimation of Equation 4.4, which results in a lower dimensional topical representation of the high dimensional corpus.

This section also investigated the impact of different document normalisation methods and recommended normalising the document lengths prior to applying the GPM model as opposed to incorporating the normalisation into the model. Moreover, the influence of the hyperparameters on the performance of the GPM model was also explored and some recommendations were also made.

Finally, the new GPM model was put in context with other existing topic models using a discrete component analysis framework. The GPM model is unique in that it not only assumes each document belongs to a single topic, but also models the counts by a Poisson distribution.

In order to demonstrate the GPM model's utility, extensive experimentation on the model was performed. The model was applied to different real-life data sets was done and various aspects of the model were studied. Details of these experiments and the results obtained will be presented in Chapter 5.

CHAPTER FIVE

EXPERIMENTS

5.1 INTRODUCTION

In order for a topic model to be useful, it must be able to uncover interpretable topics. As most topic models are unsupervised, their evaluation poses a significant challenge. The true topics are not known in advance, thus making it difficult to determine how good a job the model did at uncovering the topics. Despite this, there are some measures that are typically considered to evaluate a topic model's performance. In this section, we perform experiments to compare the performance of GPM with that of GSDMM (Yin and Wang, 2014) on different datasets.

5.2 DATASETS

The datasets on which the models were applied have been summarised in Table 5.1. All statistics were collected from the datasets after basic pre-processing (removal of stop words, punctuation, special symbols and numbers).

- The Tweet dataset (Yin and Wang, 2014) is a collection of tweets from the 2011 and 2012 Text REtrieval Conference. The most relevant tweets in 89 different categories were selected to create this collection. Each tweet is regarded as an individual document.

- The Pascal Flickr dataset contains captions of images from Flickr and the Pattern Analysis, Statistical Modelling, and Computational Learning (PASCAL) Visual Object Classes Challenge (Everingham et al., 2010). The captions are divided into 20 different classes and, altogether, the corpus contains 4 834 captions which are each treated as individual documents.

- The Search Snippet dataset (Phan et al., 2008) was created by first selecting 8 different domains: Business, Computers, Culture-Arts-Entertainment, Education-Science, Engineering, Health, Politics-Society and Sports. For each domain, 11 to 118 related phrases were typed into the Google search engine, and then the snippets from the top 20 to 30 results were collected to create a corpus of 12 295 snippets.

Note, the original number of classes/categories for each dataset will sometimes be referred to as the true number of topics/clusters or true K.

Table 5.1: Document statistics.

Corpus	M	V	K	Mean (Standard dev.) of document length	Min. (Max.) document length
Tweet	2472	5098	89	8.5 (3.2)	2 (20)
Pascal Flickr	4834	3132	20	4.9 (1.8)	1 (19)
Search Snippets	12295	4705	8	14.4 (4.4)	1 (37)

M = number of documents, V = size of vocabulary, K = number of topics

All datasets can be obtained from `https://github.com/qiang2100/STTM`.

5.3 EXPERIMENTAL DESIGN

All experiments were executed in Python 3.6 in Windows 10 on a computer with a 3.50 GHz quad core processor and 16 GB RAM. The GPM topic model is publicly available as a Python package at `https://github.com/jrmazarura/GPM`. It can also be installed using the Python 'pip install' functionality. The package contains both the GPM and GSDMM models.

For the GSDMM, the parameter values were set to $\alpha = \beta = 0.1$ and the algorithm was run for 15 iterations, as in the original paper. For the GPM, the γ parameter plays the same role as the

α parameter in GSDMM, thus it was also set to 0.1. Unless otherwise stated, the shape and scale parameters, α and β, were both set to 0.001.

5.4 TOPIC COHERENCE

To investigate the performance of the models, the average of the topic coherence score (Mimno et al., 2011) for each topic was calculated. The coherence score for each topic, T, is given by

$$coherence(T) = \sum_{(v_l, v_j) \in T} \log \frac{D\left(v_i, v_j\right) + \epsilon}{D\left(v_j\right)},$$

where v_i denotes the i^{th} word in topic T, $D\left(v_i, v_j\right)$ denotes the number of documents in which words v_i and v_j co-occur and $D\left(v_j\right)$ denotes the number of documents in which word v_j occurs. ϵ is a smoothing parameter to prevent taking the logarithm of zero and it is set to equal 1 as proposed in the original paper. As with most topic models, the GPM is an unsupervised technique. Model evaluation is generally not a trivial task in the context of unsupervised learning as datasets lack labels upon which evaluations can be based. The coherence score is a well-known measure of the degree of interpretability of a topic and it has been shown to align well with human evaluations of coherence (Mimno et al., 2011). Naturally, topics that are coherent are most desirable; therefore, a higher average coherence score is preferable. Similar to GSDMM, our model has the special characteristic of being able to automatically select the number of topics, thus, the coherence score is only calculated on the topics found by the model.

5.5 RESULTS AND DISCUSSION

This section presents and discusses the results of various experiments investigating the influence of several parameter settings on the topic model performance. These parameters are the starting number of topics, number of sampling iterations and the hyperparameters, alpha and beta

5.5.1 INFLUENCE OF THE STARTING NUMBER OF TOPICS

Topic modelling is typically an unsupervised technique. Similar to K-means clustering, the number of topics (clusters), K, is a challenge to select as the value is not usually known in advance. The GPM is able to infer the number of topics automatically provided that the starting value of

K is large enough. This is due to the dependence of the topic assignment probability, Equation 4.3, on m_k, which is the number of documents in topic k. This implies that a document is more likely to be assigned to a topic which has documents assigned to it, than a topic that does not have documents assigned to it.

As will be shown in the next section, the collapsed Gibbs sampler is quick to converge, thus the Gibbs sampler was run for 15 iterations. As the GPM also provides stable and relatively consistent results (as will be shown in the next section), experiments were repeated 3 times assuming $K = 5, 10, 20, 30, 40, 50, 100, 200, 300, \ldots, 800$. The parameters were set to $\alpha_v = \beta_v = 0.001$ for all v and $\gamma_k = 0.1$ for all k. Table 5.2 shows the average number of topics found by the model for some of the different starting values of K, whereas Table 5.3 shows the corresponding average coherence scores.

Table 5.2: Average (and standard deviation) of the final number of topics found by GPM.

		Starting value of K				
Dataset	True K	50	100	200	400	800
Tweet	89	42(2)	61 (3)	67 (7)	76 (5)	77 (5)
Pascal Flickr	20	14 (2)	26 (2)	33 (6)	33 (5)	39 (3)
Search Snippets	8	19(6)	16 (4)	25 (5)	26 (3)	32 (1)

Table 5.3: Average topic coherence score (and standard deviation).

	Starting value of K				
Dataset	50	100	200	400	800
Tweet	-25.02 (0.91)	-23.76 (1.65)	-19.92 (1.47)	-18.89 (0.06)	-18.13 (0.58)
Pascal Flickr	-37.01 (4.16)	-34.50 (2.06)	-33.85 (2.92)	-30.39 (2.59)	-31.53 (1.78)
Search Snippets	-50.56 (1.85)	-50.17 (2.06)	-50.71 (2.26)	-49.47 (4.15)	-51.19 (1.81)

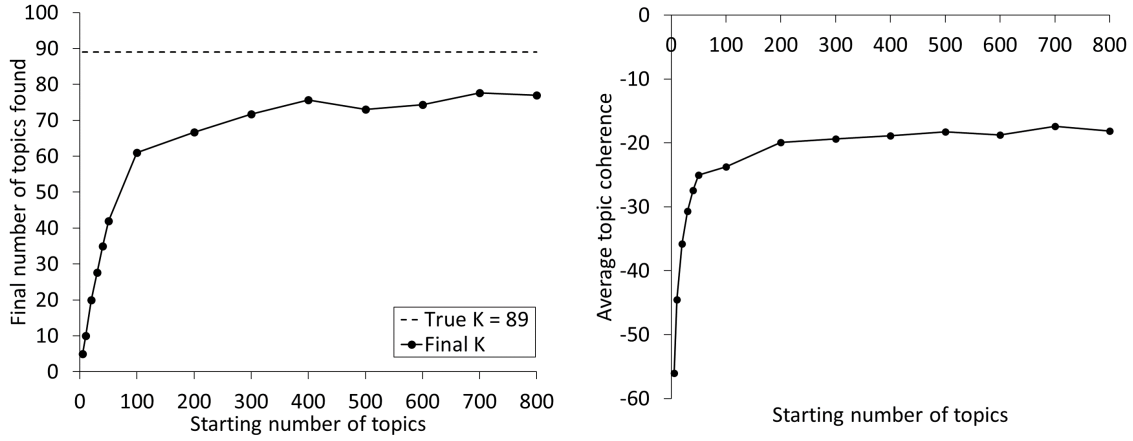
Figures 5.1 to 5.3 provide a visual summary of these results. According to Figures 5.1(a), 5.2(a) and 5.3(a), in all cases, the model approaches the true number of topics as the starting number of topics increases. In most cases, the most accurate number of topics was found by setting K to 400. For the Tweet dataset, the model converged to between 70 and 80 topics, which is close to the true value of 89. For the other datasets, the model slightly over-estimated the number of topics. On the Pascal Flickr dataset, at $K = 400$, the final number of clusters is over-

estimated by about 10 topics (true $K = 20$) whereas on the Search Snippets dataset, the final number of clusters was over-estimated by about 20 topics (true $K = 8$). One possible reason for this difference could be that the human labelling may have been too rigid and documents were classified into too few topics yet there may have been subtopics present. Consequently, it is possible that such a discrepancy could also arise if different human reviewers were tasked with labelling each document independently. In the context of topic modelling, this difference is not usually a problem especially if the topics are interpretable, as the model may have simply identified subtopics present in the corpus. Since the model does not differentiate between "main" topics or subtopics, they would all be included together in the final topic count. Nonetheless, it is still striking that in both cases, the model was able to automatically discard the extra 80-90% of topics that were unnecessary. This greatly alleviates the challenge of selecting the appropriate value of K and it avoids the additional computational complexity that is usually associated with non-parametric methods which are use to determine K.

In topic modelling, one of the most important aspects is the interpretability of the uncovered topics. Even if the final number of clusters found is not necessarily the same as what human annotators would find, it is important that the words in the topics are coherent. Figures 5.1(b), 5.2(b) and 5.3(b) show that the coherence improved as the initial K increased. In fact, a point was reached where there was almost no more improvement in average coherence when the initial number of topics was increased. In most cases, there appears to be an insignificant improvement to the coherence score when K is set to be greater than 200.

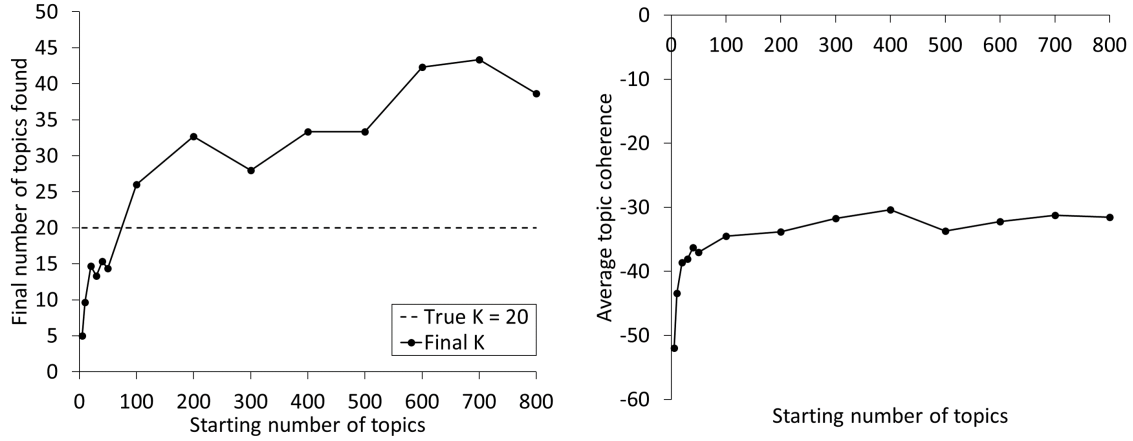
5.5.2 INFLUENCE OF THE NUMBER OF ITERATIONS

One of the challenges faced when using sampling methods to estimate parameters is determining the appropriate number of sampling iterations to perform. In order to investigate the performance of the models with respect to the number of iterations, the average coherence and number of topics found at each of 30 iterations was recorded. This was repeated three times for each dataset. From the previous results, it was found that the number of clusters was close to the human annotated number and the coherence scores reached their maximum when the model started with 400 topics, thus this value was used in all the experiments. The results are shown in Figures 5.4 to 5.6. The (a) graphs all show the number of clusters that the model found at each iteration, whereas the (b) graphs show the topic coherence at each iteration. In general, similar patterns are observed. It



(a) (b)

Figure 5.1: Tweet dataset: (a) Average final number of topics found by the model (b) Average topic coherence scores.



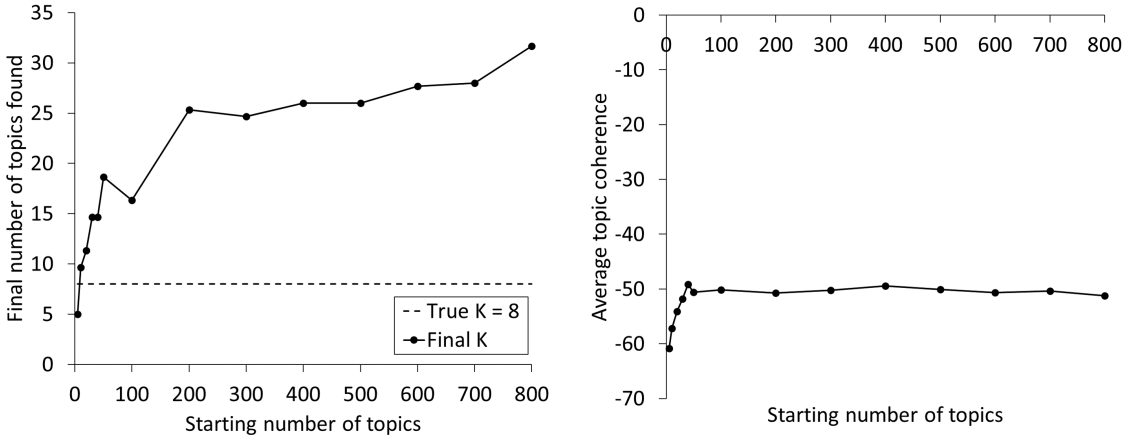
(a) (b)

Figure 5.2: Pascal Flickr dataset: (a) Average final number of topics found by the model (b) Average topic coherence scores.

is evident that convergence is reached quickly. In all cases, convergence is reached by the 15th iteration and the variation in the results is typically relatively small.

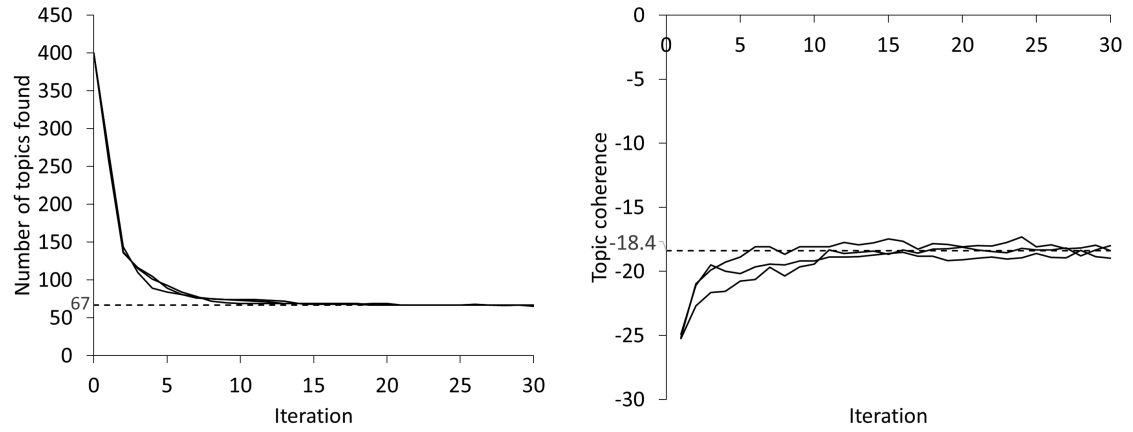
5.5.3 INFLUENCE OF GAMMA

To investigate the influence of γ on the performance of the GPM, the model was applied to the different datasets. α and β were both set to 0.001 and the starting number of topics was set to 400. The results shown here are for the Pascal Flickr corpus, whilst the results for the other corpora can



(a) (b)

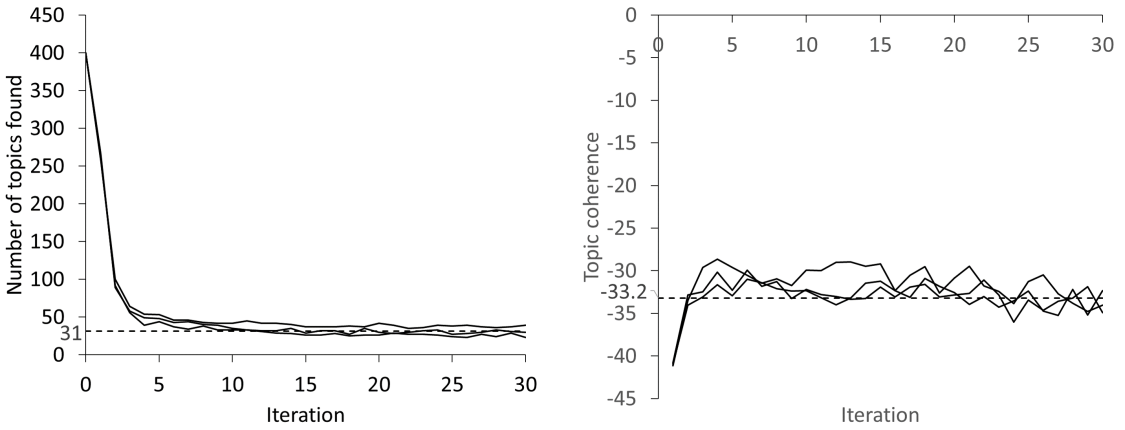
Figure 5.3: Search Snippets dataset: (a) Average final number of topics found by the model (b) Average topic coherence scores.



(a) (b)

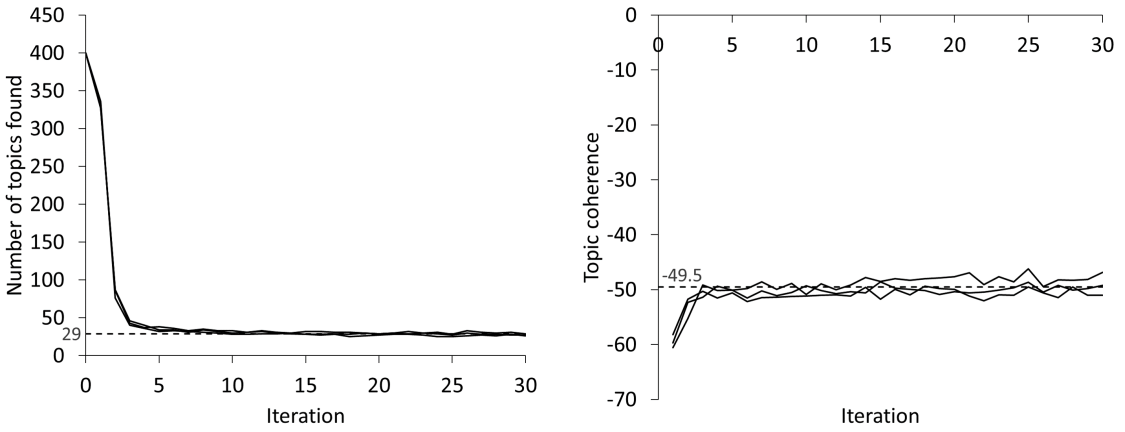
Figure 5.4: Tweet dataset: (a) Number of topics found by the model per iteration (b) Average topic coherence score per iteration.

be found in Section VII of the Appendix. The number of topics found and coherence scores for $\gamma = 0, 0.2, 0.4, 0.6, 0.8, 1$ are shown in Figures 5.7 and 5.8, respectively. Based on these graphs, the performance of the model is better for lower values of γ as the number of topics found is closer to the true K of 20 and the coherence scores were higher. For the Tweet dataset, the value of gamma does not have a significant impact on the performance of the model with respect to both performance measures. On the Search Snippets dataset, the GPM was able to get closer to the true K for values γ equal to 0 or 1. However, the variation in the number of topics found was higher



(a) (b)

Figure 5.5: Pascal Flickr dataset: (a) Number of topics found by the model per iteration (b) Average topic coherence score per iteration.



(a) (b)

Figure 5.6: Search Snippets dataset: (a) Number of topics found by the model per iteration (b) Average topic coherence score per iteration.

for $\gamma = 0$. As with the Pascal Flickr corpus, the topic coherence scores are better for smaller values of γ. This may be because Dirichlet distributions with smaller parameter values encourage a more sparse distribution of probability that aligns with the sparse distribution of topics within most of the considered corpora. In conclusion, these results provide empirical evidence in support of setting $\gamma = 0.1$ for GPM.

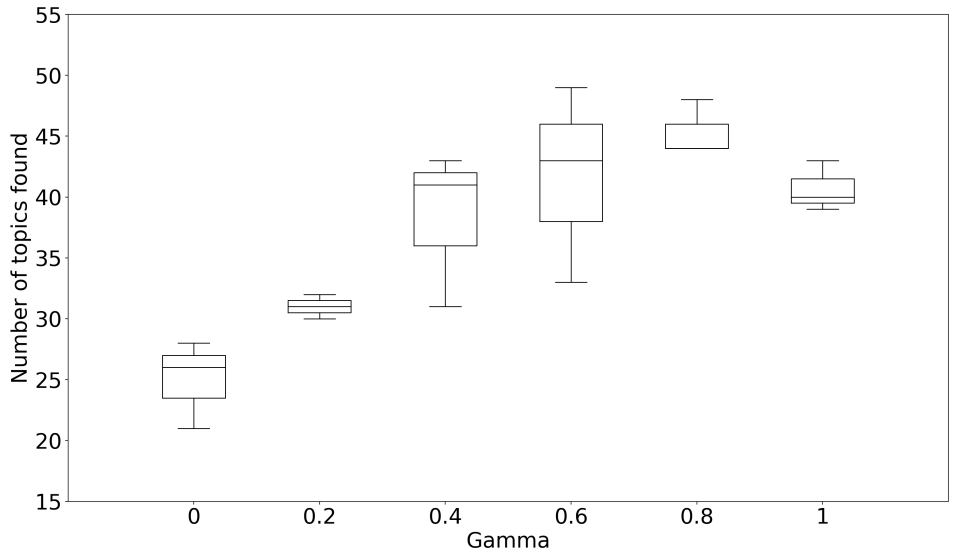


Figure 5.7: Influence of gamma on number of topics found.

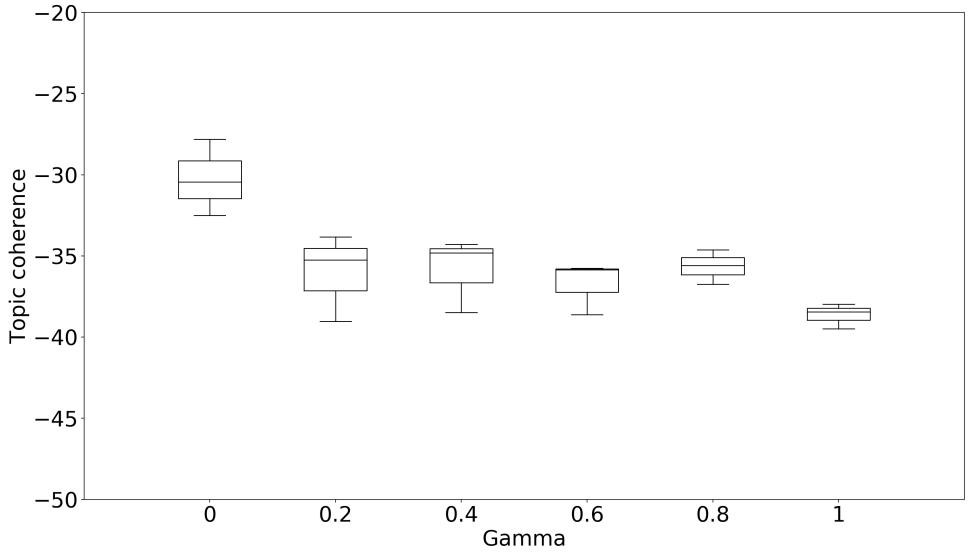


Figure 5.8: Influence of gamma on average coherence.

5.5.4 INFLUENCE OF ALPHA AND BETA

The hyperparameters α_v and β_v represent the shape and scale parameters of the gamma distribution respectively, and γ_k represents the hyperparameter of the Dirichlet prior. We assume that $\alpha_v = \alpha, \beta_v = \beta$ and $\gamma_k = \gamma$ for all v and k. The GPM was run on the Pascal Flickr dataset for $K = 40$, $\gamma = 0.1$, $\alpha = 0.01, 0.05, 0.25, 0.5, 0.75, 1, 2$ and $\beta = 5, 2, 1, 0.5, 0.2,$. Then the final number of clusters found was recorded. The results on the Pascal Flickr dataset are shown in Figure 5.9.

Owing to the computationally heavy nature of performing a grid search, each experiment was run only once per pair of α and β values, with the starting number of topics set to be at least 20 more than the true value. Figure 5.9 shows a clear downward trend for all values of β, the scale parameter. However, the final number of topics found is clearly influenced by the shape parameter, α. On the Pascal Flickr dataset, the model was only able to get close to the true number of topics (20) when α was chosen to be near 0.5. Similar downward trends were also observed on the other two datasets and β was also found to be of minimal impact on the number of topics found. However, for the Tweet dataset, α was required to be near 0.05 for the model to find close to 89 topics, whereas the Search Snippets dataset required an α value close to 1.5 to find close to 8 topics.

Figure 5.10 shows the probability density functions of gamma distributions with these different values of α and a fixed value of $\beta = 0.5$. These choices of alpha clearly produce skewed distributions which place most of their probability near zero. Based on the chosen values of α and β, the expected value of the gamma priors for the Tweet, Pascal Flickr and Search Snippets datasets are 0.025, 0.25 and 0.75, respectively. Considering the short length of the documents and the massive sizes of the vocabularies, it is not surprising that most words will have very low observed frequencies. In fact, since many zeros are observed for each word, the estimates of the Poisson parameters are also very small which results in most of the probability being loaded on zero. For example, $p(x) = 0.975$ for $x = 0$ where $X \sim Poi(0.025)$.

A similar comparison to that of Figure 5.9 was also conducted to investigate the impact of α and β on the coherence scores and the results from the Pascal Flickr dataset are shown in Figure 5.11. It can be seen that increasing the value of α decreases the number of topics the model tends to find. Whereas decreasing α increases the number of topics found. (Interestingly, this behaviour

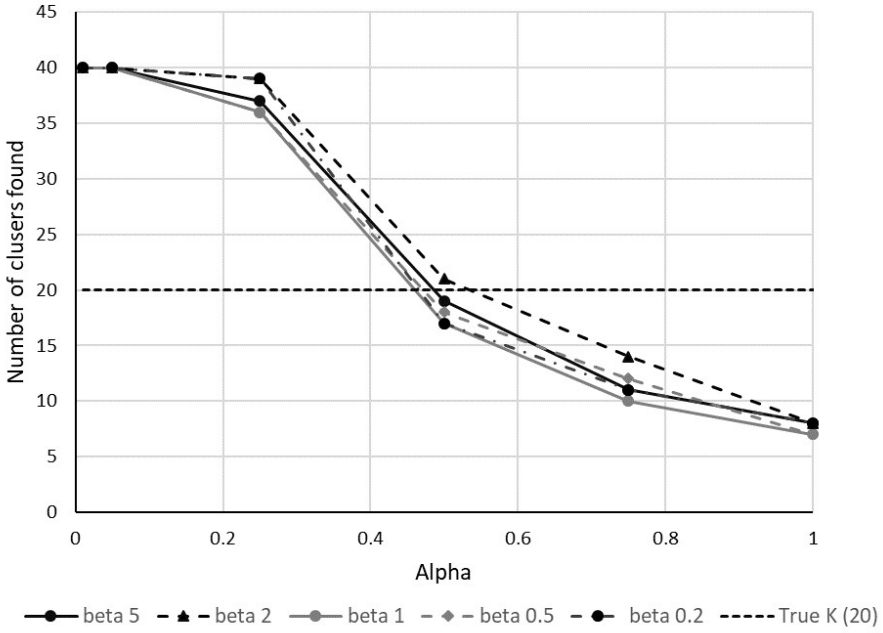


Figure 5.9: Final number of topics found for different values of alpha and beta on the Pascal Flickr dataset.

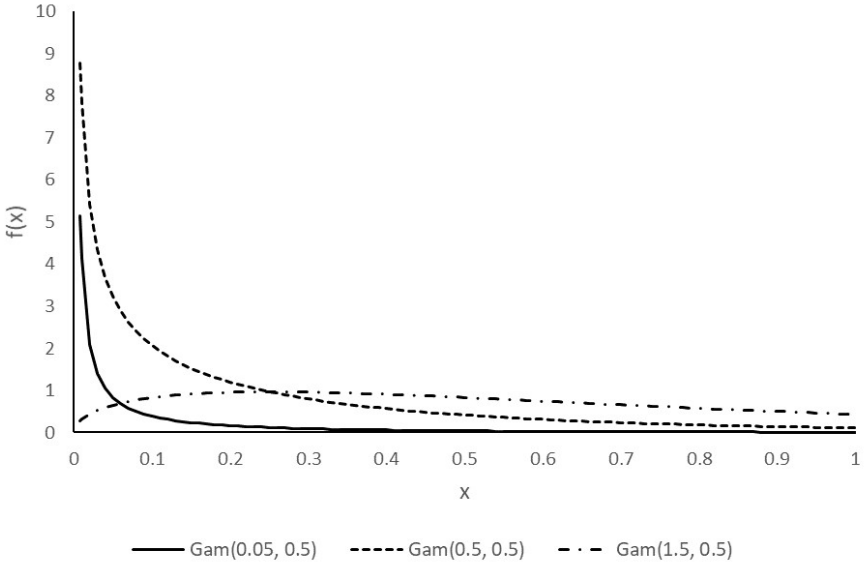


Figure 5.10: Probability density functions of the gamma distribution (denoted Gam(α,β)) for $\alpha = 0.05, 0.5, 1.5$ and a fixed value of $\beta = 0.5$.

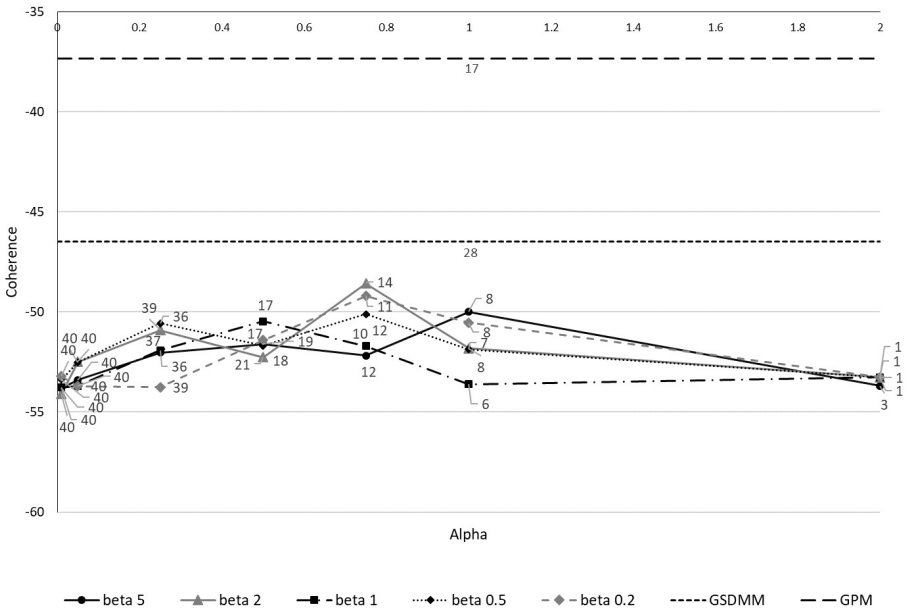


Figure 5.11: Average topic coherence of topics found for different values of alpha and beta on the Pascal Flickr dataset. The labels at each point indicate the number of topics found by the model.

of α in GPM is similar to the behaviour that was observed with the β parameter of GSDMM (Yin and Wang, 2014).) Figure 5.11 displays a general pattern; the coherence scores appear to increase, then drop as α increases and, once again, β does not appear to have a significant impact. The Tweet and Search Snippets datasets also displayed general patterns, but the pattern was not necessarily the same across the datasets. This simply serves as an indication that the selection of α is not a trivial task.

In practice, the number of clusters is not usually known in advance so it is not possible to use the true K to choose a suitable value for α. Furthermore, the coherence is also not always highest at the true number of clusters. In order to overcome this challenge, α and β are both fixed to 0.001 for all datasets. The motivation for this choice was discussed in Section 4.3.3. The top horizontal line in Figure 5.11 shows the coherence score found by GPM under this prior. Note, although this result is for fixed values of α and β, it is shown as a horizontal line across all α values to emphasise that the GPM with this choice of parameter outperforms the GPM with other choices of α and β. For ease of comparison, the results of the GSDMM are also indicated by a horizontal line although its hyperparameters are also fixed. Figure 5.11 also shows that GPM outperforms the GSDMM model (indicated by the lower horizontal line). In addition, the average number of

clusters found by GPM was also closer to the true value (20).

In conclusion, it is clear that setting of $\alpha = \beta = 0.001$ greatly simplifies the topic modelling process for GPM. In addition, we have also seen that the model possesses the flexibility of allowing the user to easily adjust the number of topics found by simply changing the value of α.

5.5.5 COMPARISON WITH DIRICHLET-MULTINOMIAL MIXTURE MODEL AND THE BITERM TOPIC MODEL

The GSDMM model was originally presented as a clustering algorithm, as opposed to a topic model, and was consequently assessed on its ability to cluster documents (Yin and Wang, 2014). As the GPM is designed for topic modelling, it was assessed on its ability to extract meaningful topics by investigating the topic coherence. The GPM is related to the GSDMM in that it also makes the one-topic-per document assumption and is able to automatically select the number of topics. Hence, the performances of GPM and GSDMM can be compared by looking at both the topic coherence scores and the number of topics automatically found by the models. For completeness, the performance of GPM will also be compared against the Biterm topic model (BTM) (Yan et al., 2013) as it is one of the state-of-the-art topic models for short text.[1] The BTM does not have the ability to automatically infer the correct number of topics so it was trained assuming the number of clusters found by the human annotators (true K). For this reason, BTM is only assessed on topic coherence. The results are summarised in the figures and tables that follow. On all the datasets, the GPM was run for 15 iterations starting with 400 initial topics with $\alpha_v = \beta_v = 0.001$ for all v and $\gamma_k = 0.1$ for all k. The GSDMM and BTM were run for 15 and 1000 iterations, respectively, which is in line with the values used in the original papers for these models. Their respective parameters were also set to the default values proposed by the original authors. All experiments were repeated 10 times. Figure 5.12 shows boxplots of the topic coherence scores. It is evident that the GPM generally outperforms the GSDMM and BTM in all three datasets, as the topic coherence of the topics obtained by the GPM is mostly larger those of the GSDMM and BTM.

For completeness, the number of clusters found by each model are considered and shown in Table 5.4. For the Tweet corpus, the true number of topics, as determined by human annotators,

[1] The code to run BTM is available in a Java based open-source library at `https://github.com/qiang2100/STTM`.

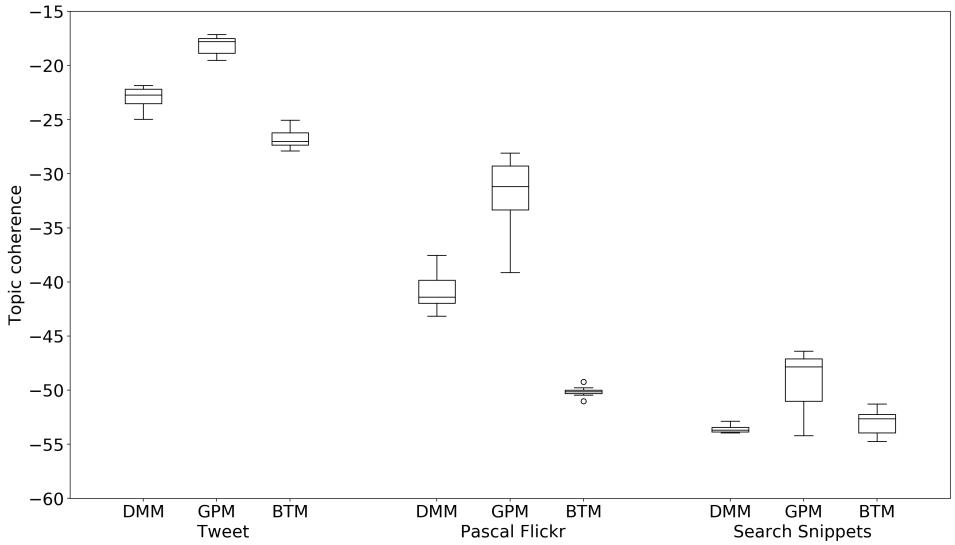


Figure 5.12: Coherence scores of the different models.

is 89. On average, the GSDMM was more inclined to find more clusters than the GPM. It is also worthwhile to note that the results obtained for the GSDMM on the Tweet dataset are close to those obtained in the original paper (Yin and Wang, 2014). On the Pascal Flickr and Search Snippets datasets, both models tended to find more clusters than those determined by the human annotators. However, the GPM was able to get closer to the true K value than the GSDMM. Interestingly, on the Search Snippets corpus, the GSDMM found significantly more topics than were found by the GPM. It is likely the case that the GSDMM found finer-grained topics, thus increasing the number of topics found, whereas the GPM model discovered fewer, but more general, topics.

Table 5.4: Summary of number of topics found by each model.

Dataset	True K	GSDMM		GPM	
		Average	Standard deviation	Average	Standard deviation
Tweet	89	98	3.56	75	4.42
Pascal Flickr	20	48	4.58	35	5.68
Search Snippets	8	303	7.19	26	2.16

Let us now consider the actual topics found by the models in one of the data sets – specifically

the Search Snippets – in order to observe what other topics were found by the GSDMM model that were not found by the GPM. Table 5.5 lists some of the top words for each of the topics found by the GPM (column 2), as well as possible labels for each topic (column 1). The labels were assigned based on the original 8 topics of the dataset and then a possible subtopic label was added in parentheses. This labelling and selection of subtopics was performed subjectively, so another annotator's assessment may produce different results.

Table 5.5: Topics found by GPM.

Topic (subtopic)	Top words
Business (software)	trillian instant pro studios creators messenger accounting
Business (trade)	import trade export leads business international global
Business (consumer)	consumption consumer motives goals ratneshwar glen mick
CAE (Chris Pirillio)	pirillo chris live internet broadcast podcast itunes streaming
CAE (music)	lyrics song com archive searchable songs database search
CAE (painting)	surreal leonardo del vinci picasso surrealism artlex artchive
CAE (videos)	videos metacafe ping pong movies internet tags amazing clips
CAE (movies)	imdb movies celebs title name diesel movie mtv aesthetic weapon
CAE (posters)	posters allposters com prints custom professional framing
CAE (transformers)	transformers movie world bay war alien directed races
Computers (networking)	approach computer networking featuring ross kurose
Computers (root)	root roottalk expression formula cern draw retrieve rene value
Computers (programming)	computer programming software web memory wikipedia intel
Computers (code)	formula expression kspread value user symbol log api input
Computers (connections)	speed test com accurate flash cable speedtest dsl connections
ES (news)	information com news wikipedia research edu home science
ES (history)	eawc edu classic ancient exploration greece evansville anthony
ES (dictionary)	dictionary online definition word christ merriam webster
Health (diet)	calorie calories energy drink enviga counter nutrition picnics
Health (disease)	treatment arthritis cause symptoms diagnosis lupus disease
PS (society)	bombs smoke homepage police press blogspot accounting bank
PS (politics)	party bob led revolutionary worker communist revolution
Sports (cars)	wheels rims car custom chrome tires truck inch tire
Sports (tennis)	match hits russia anna chakvetadze sania financial india
Sports (quad biking)	quad china atv automatic reverse quads gear product showroom
Sports/Business	goalkeepers cricket nasdaq information stock market security
CAE/Computer	span painting election contractors staining servicemagic

* Key: CAE = Culture-Arts-Entertainment, ES = Education-Science, PS = Politics-Society

In assigning the topics to the predefined labels, one challenge faced was that some topics had potential overlaps. For instance, a topic in the Engineering category could also have fallen in the Education-Science category. By analysing the first column, we also observe that 7 out of the 8 original predefined topics appear to be represented in these results. According to our labelling, the

missing topic is the Engineering topic. This is most likely due to the fact that only 369 of the 12 295 documents belonged to this topic, which is merely 3% of the entire corpus. The proportions of each topic in the Search Snippets corpus are shown in Figure 5.13.

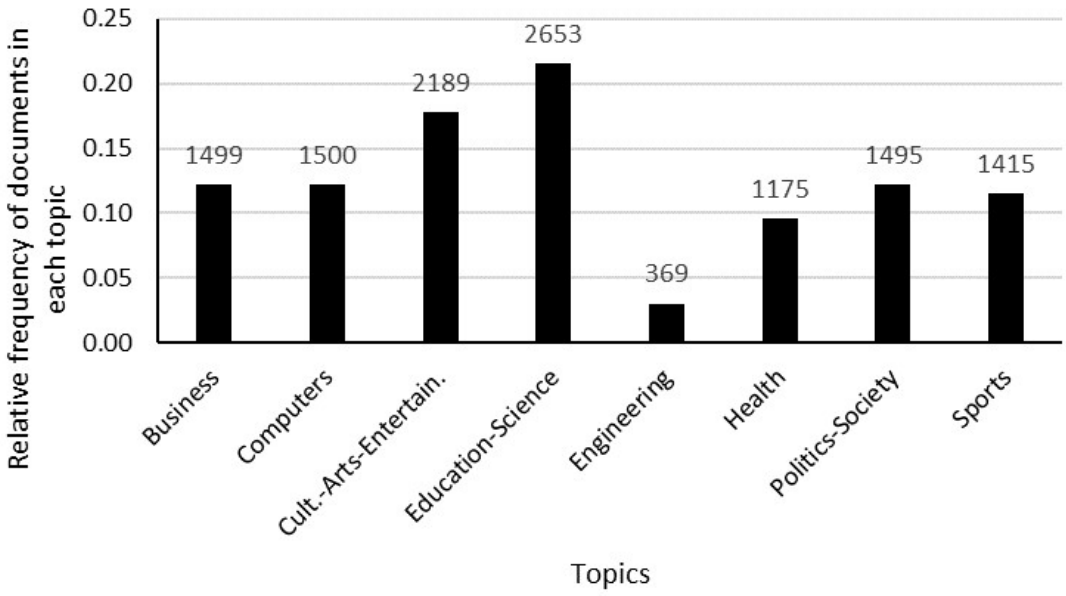


Figure 5.13: Relative frequency of documents belonging to each topic in the Search Snippets corpus. The number above each bar is the frequency of documents belonging to each topic. The corpus contains a total of 12 295 documents.

As was observed in Table 5.4, the GSDMM found more than 250 extra topics. Table 5.6 shows two additional topics for each of the 8 predefined categories that were found by the GSDMM, but not the GPM.

Since GSDMM found significantly more topics, it was able to uncover finer-grained topics. Thus, in such cases where a brief overview is desired, the model producing the smaller number of topics might be preferable. Where more detail is desired, one can opt for a model that produces more topics.

5.6 CONCLUSION

Despite the lack of attention on the Poisson distribution in topic modelling, its utility in modelling short text has been shown in the new topic model for short text, the Gamma-Poisson mixture

Table 5.6: Selected topics found by GSDMM.

Topic	Top words
Business (economics)	gdp economy product domestic gross economic value market
Business (jobs)	jobs job com search careerbuilder accounting marketing sales sites
CAE (fashion)	fashion designers design designer clothing accessories milan clothes
CAE (famous places)	ballet hollywood california angeles los universal florida studios
Computers (systems)	systems theory analysis design information programming amazon
Computers (security)	security computer network spam virus spyware viruses networking
ES (genetics)	research national gov laboratory genetic home institute genome
ES (earth)	earth structure interior edu crust tectonics model kids gov core
Engineering (physics)	physics quantum theory theoretical solid edu research technology
Engineering (Einstein)	einstein albert physics nobel eric literature weisstein world time
Health (aids)	hiv aids prevention epidemic cdc information gov health infection
Health (medical care)	hospital patient doctor medical care news information health
PS (elections)	party democratic political communist socialist republican labor news
PS (army)	force navy naval air mil commander news fleet web reserve
Sports (swimming)	swimming swim swimmers help information coaching technique
Sports (football)	football fans game nba playoff story players assault adidas university

* Key: CAE = Culture-Arts-Entertainment, ES = Education-Science, PS = Politics-Society

(GPM) topic model. This chapter presented the results of extensive experimentation on the GPM model. It explored the influence of various parameter choices, identified key influential parameters and proposed default values that can be use in practice.

As is well-known in the field of topic modelling, the selection of the appropriate number of topics is a challenge. The GPM was shown to address this problem, which is one of the most important contributions of the model. When the initial number of topics, K, is set to a high enough value, the GPM is able to automatically select the number of topics. This is achieved via the use of the collapsed Gibbs sampler. It was able to find estimates that were close to the true number of topics on labelled corpora. A further benefit of the collapsed Gibbs sampler, is that it also converges very quickly, thus evading the need for long burn-in periods as is typical in the application of traditional Gibbs samplers.

It was also shown that the number of topics found by the GPM can be adjusting the changing the value of β. This is a favourable characteristic as it gives the user flexibility and control over the model output. A further benefit of the GPM is that is also tends to produce consistent results with little variation. In addition, when compared with the GSDMM and BTM, it was shown thet the GPM outperformed these models on the datasets that were considered: Firstly, using the

recommended settings of 0.001 for α and β, the number of topics found by GPM was closer to the true value than what was found by GSDMM. Secondly, the GPM was able to find topics with higher average coherence scores, thus making it a good option for topic modelling on short text.

To further demonstrate the utility of the new GPM topic model, Chapter 6 presents an application of GPM, where it is used to assess semantic similarity between texts in order to distinguish between relevant and irrelevant documents in a corpus.

CHAPTER SIX

PROBABILISTIC DSMS FOR SMALL UNLABELLED TEXT

6.1 INTRODUCTION

Consider the scenario where a company, e.g. Discovery Ltd., wants to gather information on the public perception of their services. This is a typical task for the resident data scientist who is then tasked with scouring digital media for documents including the keyword 'Discovery'. Digital media includes social media posts, digital newspapers and blogs. Keyword search methods are quite straightforward and output all documents in the search space containing the keyword. The next step is to determine if all these documents actually relate to the intended keyword. Consider the Discovery example. Whilst some documents may refer to the company, as is desired, others may refer to the TV channel of the same name and others may simply contain the word discovery, referring to a revelation or a finding. The question driving this research is:

Can one determine if a new text is semantically related to a corpus of interest based on unstructured information only?

Let us refer to the corpus of interest as the reference corpus. For our Discovery example, this is a set of documents that are *known* to have the correct context of interest. The question can be

restated as follows: can one determine whether new texts (query corpus) are relevant or irrelevant to the reference corpus? By assuming the validity of the manifold hypothesis, this problem can be solved by transforming the corpus into a vector lying in a lower dimensional space and analysing these vectors. According to the manifold hypothesis, real-world high-dimensional data tends to lie within low-dimensional manifolds that lie within the high dimensional space (Fefferman et al., 2016).

This is where the field of distributional semantics becomes relevant. Distributional semantics is a sub-field of Natural Language Processing. Distributional semantic models (DSMs) explore the meaning in language and aim to create semantic representations through learning by association. DSMs are distributional in the sense that their parameters are learned through context from other observed co-occurring words (Ó Séaghdha and Korhonen, 2014). They are based on the assumption that the meaning of a word can be inferred from its usage in combination with other words; an idea that was famously summarised by Firth (1957) in the saying, 'You shall know a word by the company it keeps."

A vector-space approach is the most common DSM methodology and a very recent and successful example is word embeddings, such as word2vec. Once words are represented in Euclidean space, the applications are almost endless, ranging from collaborative filtering (Hofmann, 2004; Wang and Blei, 2011), aspect-based sentiment analysis (Brody and Elhadad, 2010) and text classification (McCallum, 1999). Under the vector space approach, the acquired transformation $\mathbf{f}_w$ is defined as a vector for $w \in \mathcal{C}$, where w is each word in the corpus vocabulary $\mathcal{C}$. Models which produce such representation of documents are called vector space models (VSMs). The tf-idf (term frequency-inverse document frequency) model is an example of a VSM. Tf-idf creates a vector-space representation of a document that tries to capture the importance of each word to the document. Unfortunately, tf-idf produces vectors that are not only sparse, but also remain in the same high-dimensional space as the original documents. Word2vec, on the other hand, is able to produce short, dense vector-space representations that capture semantic properties between words.

A popular focus in the field of distributional semantics is determining similarity between linguistic entities, such as words or documents. This application is based on the distributional hypothesis which states that words with similar $\mathbf{f}_w$ vectors have similar meaning (Levy et al., 2015). The comparison between two vectors is made with similarity functions such as cosine similarity (Murphy, 2012). With recent advances in deep learning, word embeddings such as word2vec

(Mikolov et al., 2013) became popular distributional representations of words to such an extent that pre-trained word embeddings have been developed and are available for open-source use.[1] These pre-trained embeddings are trained on millions, if not billions, of words from web-based corpora (Pennington et al., 2014) .

If the vector $\mathbf{f}_w$ is normalised to unit sum, then it parameterises a discrete distribution which can be defined as the conditional probability of observing a particular context given that word w is observed (Ó Séaghdha and Korhonen, 2014). One does not need to search far for such a probabilistic representation, as topic models such as the well-known Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and the recent Gamma-Poisson mixture (GPM) (Mazarura et al., 2020) for short text, produce such representations. These topic models are examples of probabilistic DSMs. They decompose high dimensional count vectors into two lower dimensional probability distributions: one which acts as a clustering mechanism, and the other which acts as distributional semantic representation for a document.

Returning back to the 'Discovery' example, suppose that the data scientist only had access to a 2 000-document corpus made up of social media posts, short reviews and comments from various websites. This problem is plagued with various challenges.

1. Firstly, the corpus itself is small, so contains limited information. A vector space model like word2vec may struggle in this scenario as it is well-known that it requires training on large collections of data.

2. The fact that the collection of documents is made up of short text also creates further challenges as short text generally tends to provide very limited contextual information.

3. Lastly, such collections of data tend to be very noisy as they are often fraught with colloquialism, spelling errors and social media acronyms.

The Discovery example can be thought of as a supervised case due to the assumption that there exists a pre-labelled corpus of training documents that are known to be about the Discovery Ltd. This chapter takes this application a step further by making a contribution in an unsupervised case. In this scenario, it is assumed that the researcher is presented with a collection of short documents which are unlabelled as is often the case in real-life data. Consequent the contents of the corpus

[1] `https://nlp.stanford.edu/projects/glove/`

is unknown in advance. The researcher would then want to determine the topics contained in this corpus then, upon deciding on a topic of interest, she would then continue to determine whether new documents are related to the topic of interest.

Ultimately, the objective of this chapter is to demonstrate the utility of the new GPM topic model in distinguishing between semantically similar and dissimilar documents under such circumstances. In order to demonstrate this, the GPM's performance will be compared to that of word2vec on 3 datasets. The first two datasets represent applications that fall under the supervised scenario, similar to the Discovery example, whereas the third represents the unsupervised scenario.

1. The first dataset contains a collection of news article titles that are labelled according to the topic that they belong to. This collection is used to demonstrate how GPM is able to perform well even with small corpora.

2. The second dataset is a labelled corpus of online conversations. Conversations are labelled according to whether they are predatory or not and the objective is to identify whether a new unseen conversation (query corpus) involves a sexual predator or not. This corpus has the challenges of not only being short, but also being very noisy.

3. The last dataset is a collection of abstracts from COVID-19 papers. As the pandemic is a fairly recent development, naturally there is a limited amount of data on the subject. As is often the case in practice, this dataset is also unlabelled, which poses a further challenge. In this application, the unsupervised nature of the topic model is exploited to provide labels for the data. Finally, GPM and word2vec are used to address the problem of determining the relevance of a query corpus to a reference corpus.

The following section provides details of the architecture of this application.

6.2 SEMANTIC SIMILARITY ARCHITECTURE

The objective of this section is to provide a breakdown of the tasks involved in establishing a semantic similarity score. Figure 6.1 summarises the steps that will be followed to achieve this. The white blocks represent input or output artefacts, such as corpora, matrices or scores. The embedded grey blocks represent an algorithm or calculation. The dashed grey blocks contain section

numbers which indicate the respective subsection explaining different aspects of the architecture in more detail. In this architecture, the datasets are assumed to be pre-processed as discussed in Section 6.3.2. In order to test the DSM's generative abilities, the corpus is first split into a training and test set. The experimental workflow shown in Figure 6.1 can be summarised briefly as follows:

1. Train the DSM on the training set. This will yield a semantic representation of the training set. (Section 6.2.1)

2. If the documents are unlabelled, label them by taking the dominant topic[2] of each document from the semantic representation derived by the topic model. If the documents are pre-labelled, this step is omitted.

3. Select a class of interest. Documents belonging to this class make up the reference corpus.

4. Index the test set and use the trained model to infer the topic distributions of documents in the test set. This will yield a semantic representation of the test set. (Section 6.2.2)

5. Calculate the semantic similarities within the reference corpus and between between the reference and query corpora. (Section 6.2.3)

6. For each document in the query set, calculate the probability of the semantic similarity measure being obtained from the reference corpus semantic similarities. This probability is then defines a *relevance index score*, which can then be used to determine whether a document is in the query corpus is relevant to the reference corpus or not. (Section 6.2.4)

It is important to observe that the labels are not required for training, thus making this architecture an unsupervised technique in practice. Their main purpose is to provide a baseline for the evaluation of the results.

The different stages of the workflow will now be discussed in more detail in the subsequent sections.

[2]This is simply the topic with the highest proportion in the document.

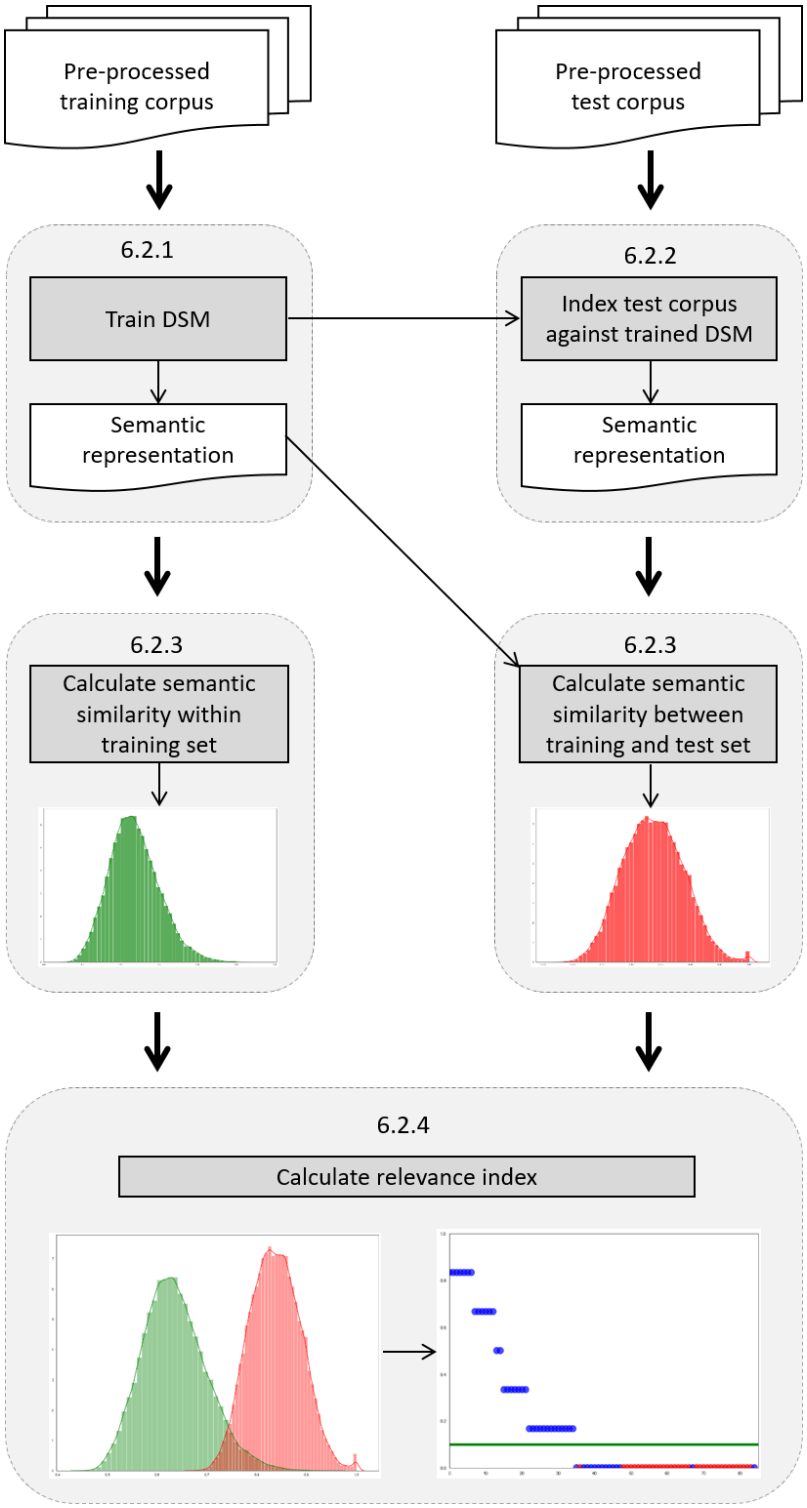


Figure 6.1: Semantic similarity architecture.

6.2.1 TRAIN DISTRIBUTIONAL SEMANTIC MODELS

The first task in the architecture is to train the DSM on the training corpus. Two DSMs will be considered: GPM and word2vec. GPM has been discussed extensively and the theory behind word2vec can be found in (Mikolov et al., 2013). After training the models, the output is a lower-dimensional semantic representation for each document in the training corpus.

6.2.2 INDEX AND TRAIN DSMS ON TEST CORPORA

The next step is to index the test set. The pre-processing of the training set results in a vocabulary of unique words and, in the case of topic models, with a bag-of-words transformation of the unstructured documents. The test set may contain words which are not present in the training set vocabulary. Therefore, we need to index the test set to the same bag-of-words representation as the training set. After this is done, the DSMs can be trained on the test corpus. The result is a semantic representation of the test set.

6.2.3 CALCULATE SEMANTIC SIMILARITY

To determine the similarity between documents, appropriate distance measures between the semantic representations of the documents must be calculated. For the word2vec output, the soft-cosine similarity metric will be used to calculate similarity scores. Since the output of the GPM model yields doc-topic distributions of the corpus, it will assessed using Jensen-Shannon distances, which is a probabilistic measure. An overview of both distance measures is given below.

6.2.3.1 SOFT-COSINE SIMILARITY

Given two texts represented as vectors A and B, the cosine similarity between them can be calculated according to the formula

$$\cos(A, B) = \frac{A' \cdot B}{\sqrt{A' \cdot A}\sqrt{B' \cdot B}}, \tag{6.1}$$

where $0 \leq \cos(A, B) \leq 1$. A value near 1 indicates high similarity whilst a value near 0 indicates low similarity between documents (Charlet and Damnati, 2017; Jurafsky and Martin, 2019). One weakness of this measure is that $cos(A, B) = 0$ when texts do not share any common words.

This characteristic is undesirable as texts can be semantically related even when there are no common words. This problem can be overcome by using the soft-cosine similarity. The soft-cosine similarity measure achieves this by introducing a similarity matrix, S, into Equation 6.1 which yields,

$$\text{softcos}(A, B) = \frac{A' \cdot S \cdot B}{\sqrt{A' \cdot M \cdot A}\sqrt{B' \cdot S \cdot B}}. \tag{6.2}$$

Cosine similarity regards VSM features as being independent. Whereas, soft-cosine similarity generalizes the concept of cosine similarity by also considering semantic similarity between features (Sitikhu et al., 2019). The soft-cosine similarity measure can be easily calculated using the SoftCosineSimilarity function from the gensim Python package. A tutorial detailing the application of word2vec for document similarity in the Python package gensim is available at `https://praveenbezawada.com/2019/03/22/document-similarity-using-gensim-word2vec/`.

6.2.3.2 JENSEN-SHANNON DISTANCES

A well-known analogue for vector-space similarity measures is the Kullback-Leibler (KL) divergence Kullback and Leibler (1951). KL divergence measures the distance between two probability distributions. KL divergence is not a distance metric as it is asymmetric and does not satisfy the triangle inequality (Murphy, 2012). It can be symmetrised to produce the Jensen-Shannon divergence, which is defined as

$$JS(P, Q) = 0.5 \cdot KL(P||R) + 0.5 \cdot KL(Q||R),$$

where $R = 0.5(P + Q)$ and $KL(P||M)$ denotes the KL divergence between probability vectors P and Q.

Taking the square root of the Jensen-Shannon divergence produces a distance metric called the Jensen-Shannon distance (JSD). In the context of this chapter, P and Q denote topic-distributions (or semantic representations) of documents. Smaller JSD values indicate more similarity between documents whilst larger values indicate less similarity. This is in contrast to the soft-cosine similarity measure where small values indicate less similarity and larger values indicate more similarity. In order to give the JSD-based measure the same interpretation, comparisons between the

semantic representations were conducted using the *adjusted* Jensen-Shannon distances (AJSD), where AJSD = 1-JSD. Consequently, very similar documents will have higher AJSDs whilst less similar documents will have lower values.

6.2.4 CALCULATE RELEVANCE INDEX

JSDs and soft-cosine similarities are not directly comparable, thus there is a need for a suitable conversion that will allow for the comparison of word2vec and GPM. To this end, a relevance index will now be defined.

Consider the previously-introduced Discovery example. Suppose that a corpus of documents that is known to be related to the company Discovery Ltd is isolated (reference corpus). If new documents (query corpus) were to become available, one would want to determine whether or not they are semantically related to the reference corpus. One way of doing this is by calculating the mean similarity of the documents *within* the reference corpus and comparing them with the mean similarity between the reference corpus and the query corpus. These distances between the documents could be summarised visually in a histogram similar to those shown in Figure 6.1. Naturally, it would be expected that distances between documents within a reference corpus would be short. A relevance index can now be developed based on the answer to the following question: "How likely is it to obtain a mean semantic similarity from the reference set that is less than or equal to the mean similarity between the reference and query sets?"

Let μ_r denote the mean similarity between documents in the reference set. If $\mu_{q,i}$ denotes the average distance between documents in the reference set and the i-th document in the query set, then the relevance index is found by calculating the probability represented by shaded area in Figure 6.2. This probability is estimated based on the observed average distances between the reference set documents. If a hypothesis test was constructed where the null hypothesis was that the new document was relevant (high similarity) and the alternative hypothesis was that is was irrelevant (low similarity), the relevance index would be equivalent to the p-value for this test. Consequently, high probabilities indicate high relevance and low probabilities indicate low relevance.

The relevance indices can be displayed graphically to get a visual idea of the performance of the DSMs. Since the documents are labelled (either using the topic model or pre-existing labels), data points can then be labelled according to whether the corresponding document was related to

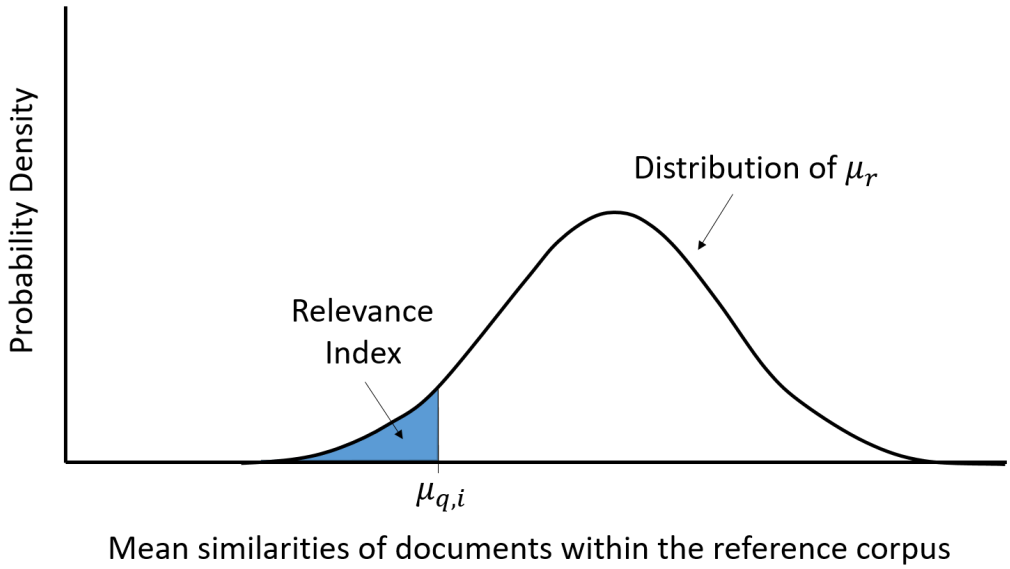


Figure 6.2: Illustration of calculation of the relevance index. $\mu_{q,i}$ denotes the average distance between the i-th query document and documents in the reference set.

the query set (blue) or whether it was irrelevant (red). Good performance indicators for the DSMs are high relevance indices for 'blue' documents and low relevance indices for 'red' documents.

One can also go a step further and set a threshold value between 0 and 1, on the query set relevance indices in order to calculate hard classification metrics such as accuracy, precision and recall. Documents with relevance indices higher than the threshold are classified as relevant to the query set and those below the threshold are classified as being irrelevant. If the results are displayed graphically, then 'blue' documents would be expected to be above the threshold line, whilst 'red' documents would be expected to be below the line if the DSMs were successful. This means that 'red' documents above the threshold are regarded as false positives and 'blue' documents below the threshold are regarded as false negatives.

The following section will now provide details regarding the experimental setup.

6.3 EXPERIMENTAL DESIGN

The objective of the experiments is to compare vector-space (word2vec) and probabilistic (GPM) DSMs in their ability to distinguish between semantically similar and dissimilar text. This section covers various aspects of the experiments, such as parameter settings, pre-processing procedures and data descriptions. Note, all experiments were executed in Python 3.6 in Windows 10

on a computer with a 3.50 GHz quad core processor and 16 GB RAM. The codes to conduct the experiments are available at `https://github.com/jrmazarura/Similarity_Experiments`. Note, all experiments were repeated 10 times in order to provide averages and standard deviations for the classification metrics.

6.3.1 PARAMETER SETTINGS

In this application, setting the parameter values for GPM to $\alpha = 0.001$, $\beta = 0.25$ and $K = 10$ yielded good results. For word2vec, word vectors of size 200 were generated using the skip-gram model from the gensim Python package.

6.3.2 PRE-PROCESSING

Prior to the application of the models, the corpora underwent standard pre-processing. This included reducing uppercase words to lower case, the removal of stop words, numbers and special characters, as well as stemming each word.

6.3.3 DATASETS

As previously mentioned, three datasets are considered. The first is a collection of news article, which shall be referred to as the NEWS-2020 corpus. The second corpus is a collection of conversations labelled as either non-predatory or predatory. This corpus shall be referred to as the PAN-2012 dataset. The third is a COVID-19 dataset, which shall be referred to as the CORD-19 corpus.

6.3.3.1 NEWS-2020

This dataset contains 108 774 news articles that come from one of 8 topics: Business, Entertainment, Health, Nation, Science, Sports, Technology and World. This dataset is readily available as an open-source dataset on Kaggle[3]. As the focus of this study is on short text, only the titles of the news articles were considered. After pre-processing, the average length of the titles was 7 words and the vocabulary size was 4 300 words. In order to test the DSMs on small corpora, two smaller corpora were created by randomly selecting 10 000 and 5 000 documents, where each category

[3] `https://www.kaggle.com/kotartemiy/topic-labeled-news-dataset`

was equally represented. The Health class was chosen as the class of interest. The test (query) corpora was made by randomly selecting 20% of the Health titles and 20% of the Science titles from the smaller corpus. The remaining documents made up the training corpus.

6.3.3.2 PAN-2012

The PAN-2012 dataset (Inches and Crestani, 2012) is a collection of 66 927 conversations labelled according to whether the conversation involved a predator or not. This dataset was created for researchers to have a common reference point to compare different approaches for identifying potential sexual predators. After pre-processing, the corpus contained 61 243 conversations as some documents ended up being empty. These conversations were made up of 59 456 non-predatory conversations with an average length of 40 words per conversation (median 6), and 1 787 predatory conversations with an average length of 86 words (median 23). The non-predatory and predatory conversations were then split according to an 80-20 ratio. The training set was made up of the two 80% portions from the non-predatory and predatory conversations, whilst the remaining 20% portions made up the test set.

6.3.3.3 CORD-19

The COVID-19 Open Research Dataset[4] (CORD-19) (Wang et al., 2020) is a growing collection of medical papers that was launched in March 2020. It was created with the objective of allowing the global research community the opportunity to study the corpus and develop data mining tools to generate new insights that may help fight the ongoing pandemic. The experiments on this dataset were conducted on the abstracts associated with the 'comm_use_subset' and 'noncomm_use_subset' papers. We will refer to these short text corpora as the 'comm_abstracts' corpus (training set) and 'noncomm_abstracts' corpus (test set). Since, some papers did not have abstracts, the 'comm_abstracts' and 'noncomm_abstracts' corpora only contained 8 750 and 1 830 documents, respectively. After pre-processing the average length of the documents in the 'comm_abstracts' and 'noncomm_abstracts' corpora was 104 and 92 respectively.

In previous experiments, GPM was applied to cleaned text that averaged 8 to 15 words per document. In light of this, prior to the application of the GPM each document was truncated

[4]Data is available at `https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge`

and only the first 10 words of each document were kept. As opposed to randomly selecting 10 words from each document or using more sophisticated feature selection methods, such as tf-idf, this procedure ensures that the selected words originate from the first few sentences. Thus, if the pre-processing was to be reversed, the selected words would be found in coherent sentences. In addition, we also deemed it sensible to assume that most authors are likely to give careful consideration to the first few sentences in their abstracts, thus we selected the first 10 words rather than 10 words from elsewhere in the abstracts. This resulted in a vocabulary containing 9 777 unique words.

6.4 RESULTS

6.4.1 NEWS-2020 DATASET

The first step was to choose a class of interest. In this case, the 'Health' topic was chosen, thus the reference set was made up of 'Health' documents from the training set. In order to compare the performance of the models, the semantic representations of the documents that belonged to the reference set were then compared with those of documents belong to two classes from the test set (query sets). The first query set was made up of documents in the test set belonging to the same class as the reference set, ie. 'Health'. The second query set contained documents from a class that was different to the class of interest. In this case, documents from the 'Science' topic were chosen as the second query set. The distances between the reference and query sets were then calculated and summarised in histograms. High values of these metrics indicate high similarity whereas low values indicate low similarity.

The top graph in Figure 6.3 shows results from GPM. The GPM tends to produce semantic representations whose distances tend to be either extremely large or extremely small. This is a common characteristic of the GPM histograms, which is not only apparent in Figure 6.3, but will also be observed in later graphs. The reason for this is that the GPM assigns documents to topics in a manner akin to hard clustering due to its one-topic-per-document assumption. The yellow histogram in Figure 6.3 shows the distribution of semantic similarities (adjusted Jensen-Shannon distances) between 'Health' the reference set and the 'Health' query set. The expectation was that, since these documents came from the same topic, their distances will generally be large, indicating high similarity. In contrast, the blue histogram is expected to be centred near lower

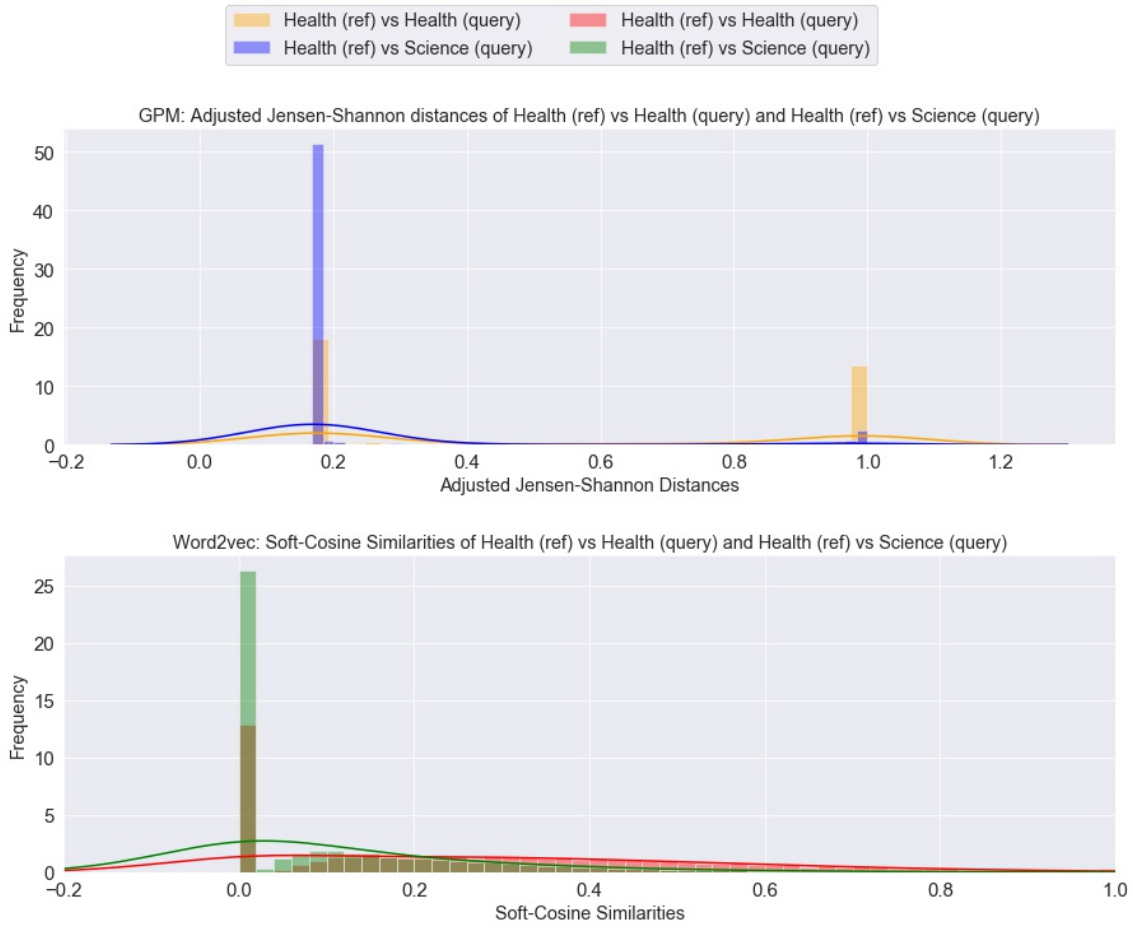


Figure 6.3: Distributions of distances between semantic representations of NEWS-2020 documents from GPM (top) and wordvec (bottom).

values as it shows the distribution of semantically dissimilarities sets, the 'Health' reference set and the 'Science' query set. The results for word2vec are shown in lower graph of Figure 6.3. The distribution of distances between the similar sets are shown in the red histogram, whilst that of the dissimilar sets is shown in green. From the blue and green histograms, it is evident that both models were able to produce semantic representations that captured the dissimilarity between the 'Health' and 'Science' sets. When it came to the similar sets, some of the distances were large as expected, but there is some overlap with the dissimilar set distances. Overall, both models appear to have performed well and will likely be useful for determining relevance for these topics.

The next step was to calculate relevance indices for the documents in the query sets. The results are summarised graphically in Figure 6.4. Blue points correspond to documents belonging to the relevant set, i.e. the 'Health' query set whilst red point indicate documents belonging to

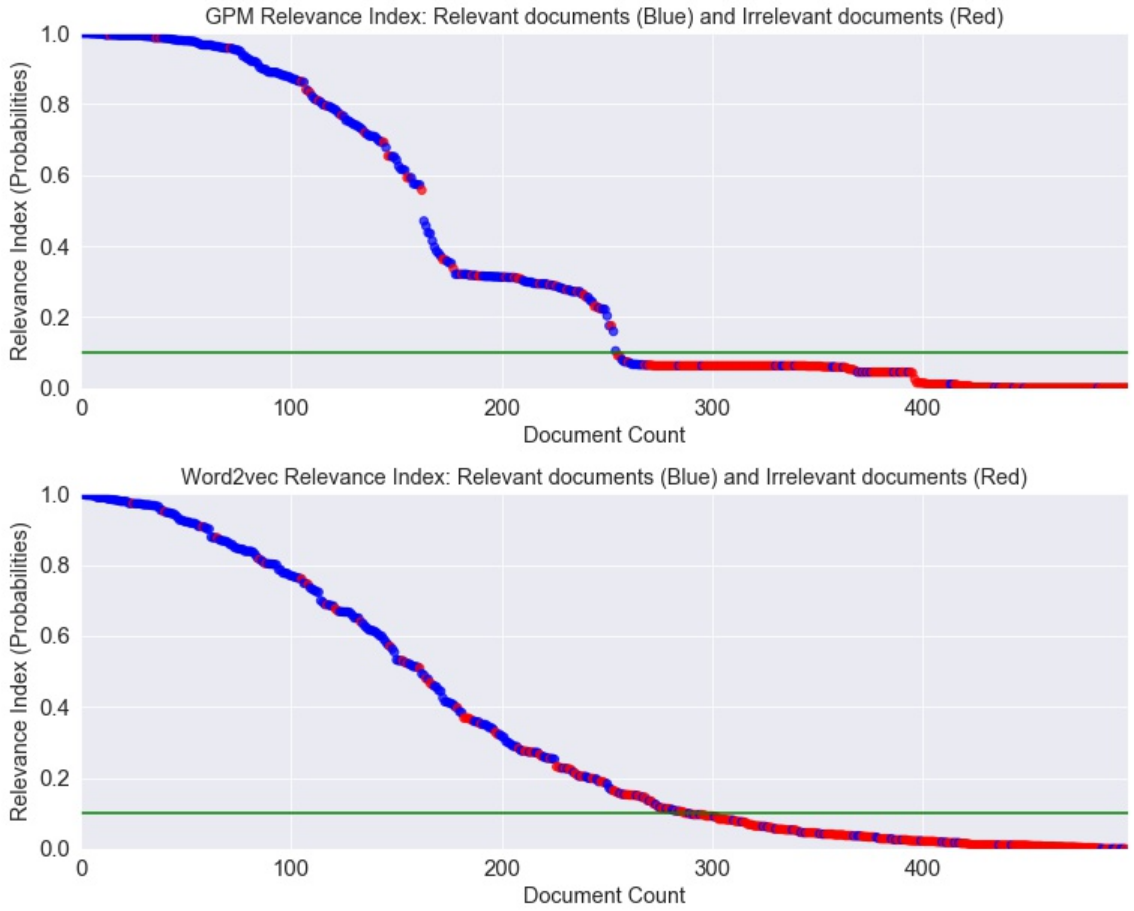


Figure 6.4: Relevance index results for GPM and word2vec on 10 000-document corpus from the NEWS-2020 corpus.

the irrelevant set, i.e. the 'Science' query set. If a document is highly relevant, it should have a high relative index value, whereas an irrelevant document should have a low relevance index. Evidently, both models perform well as blue points predominantly have higher relevance values whilst red points tend to have lower values.

As described in Section 6.2.4, the accuracy, precision and recall of each model can be calculated by comparing these relevance indices with a threshold value. For this application, the value was chosen to be 0.1 and is indicated by the green horizontal line in Figure 6.4. Documents above this line are then classified as relevant and documents below are classified as irrelevant. The experiment was repeated 10 times, and the averages and standard deviations of the classification metrics are shown in Table 6.1 under the 10 000 documents heading. Based on this threshold, the GPM was able to classify documents according to relevance more accurately than word2vec. Precision

Table 6.1: Averages (and standard deviations) of classification metrics for different models on the NEWS-2020 corpus.

Metric	10 000 documents		5 000 documents	
	GPM	word2vec	GPM	word2vec
Accuracy	0.820 (0.035)	0.775 (0.006)	0.824 (0.030)	0.611 (0)
Precision	0.738 (0.083)	0.689 (0.010)	0.756 (0.065)	0.624 (0)
Recall	0.883 (0.021)	0.834 (0.004)	0.877 (0.028)	0.614 (0)
F1 Score	0.801 (0.050)	0.755 (0.007)	0.810 (0.039)	0.619 (0)

tells us the proportion of the documents that were classified as relevant that were actually relevant (ie. proportion of all points above the threshold line that are blue). In this context, precision is the proportion of documents that were classified as being about 'Health' that were actually about 'Health'. If there are no false positives, then the value of the precision will be 1. Recall gives us the proportion of actually relevant documents that were correctly identified (ie. proportion of blue dots that are correctly positioned above the threshold line). In other words, recall is the proportion of actual 'Health' documents that were correctly identified. The F1 score is a trade off between precision and recall and a higher value is generally preferable. The performance of GPM is better with respect to all these measures at a threshold of 0.1.

The classification performance can also be summarised visually by ROC (Receiver Operating Characteristics) curves as in Figure 6.6. Figure 6.6 shows results from a single run of the codes. AUC (Area Under The Curve) values for each model are also indicated. The closer the AUC value is to 1 the better the performance. An AUC value is 0.5 indicates poor performance as this means that the model was unable to classify any point correctly. Figure 6.6(a) shows that both models generally perform well, but the GPM is slightly better than word2vec.

Lastly, as mentioned previously, one of the problems that arises in practice is that of only having small-sized datasets. To investigate the performance of the models on a smaller corpus, the experiments were repeated again, but this time only 5 000 documents were randomly selected instead of 10 000. These documents were selected in the same manner as described in Section 6.3.3.1. The relevance indices are plotted in Figure 6.5. It is evident that the performance of word2vec severely deteriorated. Unlike GPM which generally produced high relevance values for blue point and low values for red point, GPM struggled to make this distinction. The relevance

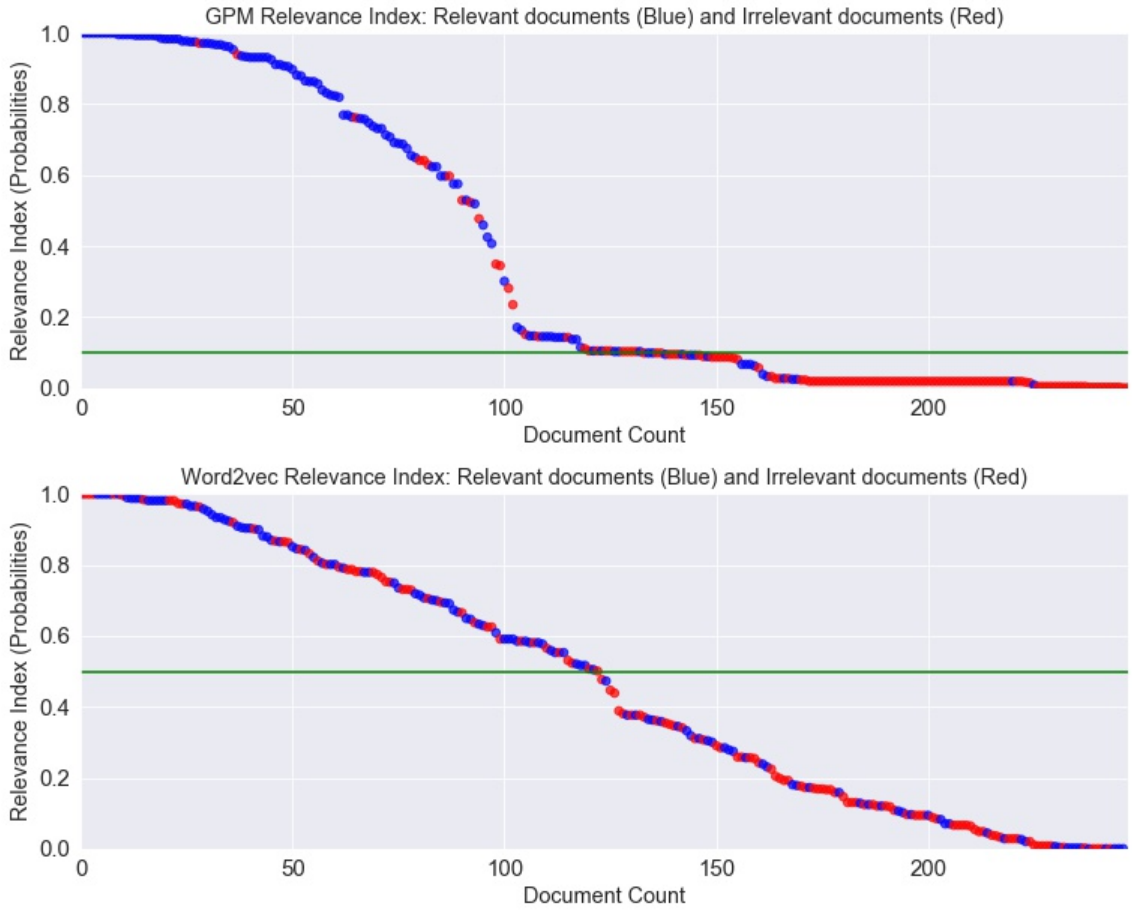
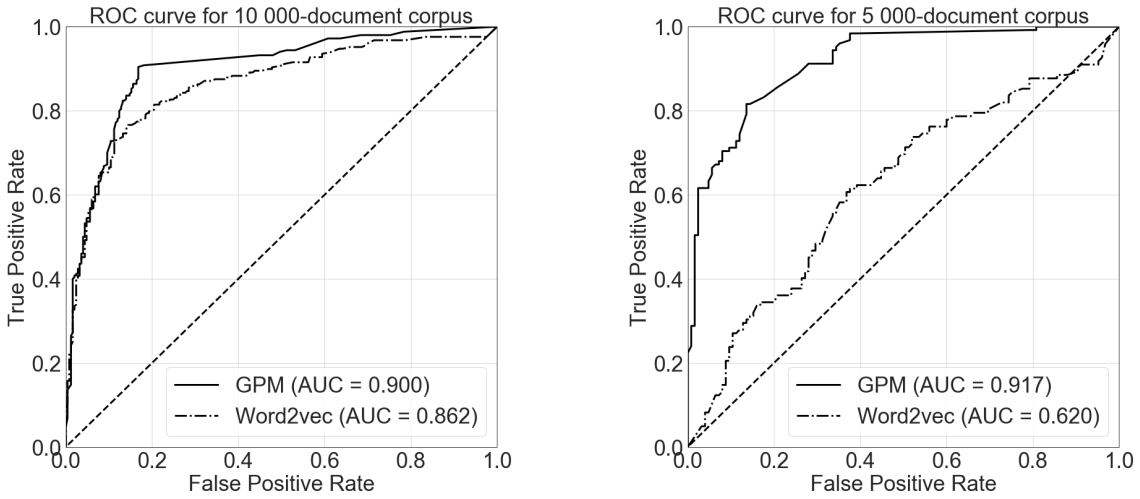


Figure 6.5: Relevance index results for GPM and word2vec on 5 000-document corpus from the NEWS-2020 corpus.

index values for red and blue points appear to be randomly scattered between 0 an 1 thus indicating that the model struggled to distinguish between relevant and irrelevant documents. This is also evident in the classification metrics shown in Table 6.1 under the 5 000 documents heading as well as in Figure 6.6(b). For word2vec, the threshold was changed to 0.5 to find a balance between precision and recall for the model. Table 6.1 shows a large decrease in performance for word2vec, whereas the GPM performance only drops slightly. Furthermore, it is clear that the GPM outperformed word2vec with respect to accuracy, recall, precision and F1 score by a large margin. The ROC curve and AUC values shown in Figure 6.6(b) also support this. In conclusion, for the smaller dataset, GPM is the better option.



(a) (b)

Figure 6.6: ROC curves for NEWS-2020 corpora for (a) 10 000-document corpus and (b) 5 000-document corpus.

6.4.2 PAN-2012 CORPUS

Like the NEWS-2020 dataset, the PAN-2012 corpus is also pre-labelled. However, this dataset poses somewhat of a greater challenge than the NEWS-2020 dataset as it is a collection of conversations. Conversations tend to be informal and contain colloquialism, spelling errors and social media acronyms, which make this a very noisy dataset. This collection is labelled according to whether the conversation threads were predatory or not. The reference corpus is thus chosen to be predatory conversations as it is important to be able to identify such conversations in order to detect potential predators.

Similar to the NEWS-2020 corpus experiments, relevance indices were calculated between the reference set (predatory documents from the training set) and two query sets, a similar and a dissimilar query set. The similar query set contained the predatory conversations from the test set and the dissimilar query set was made of the non-predatory conversations from the test set. The relevance index results are shown in Figure 6.7. Both models perform well as blue point tend to have higher relevance indices whilst red points tend to have lower indices. The averages and standard deviations of classification metrics from 10 runs of the codes using thresholds of 0.25 and 0.1 for GPM and word2vec, respectively, are shown in Table 6.2. In this context, the precision gives the proportion of conversations that were classified as predatory that were actually predatory. The

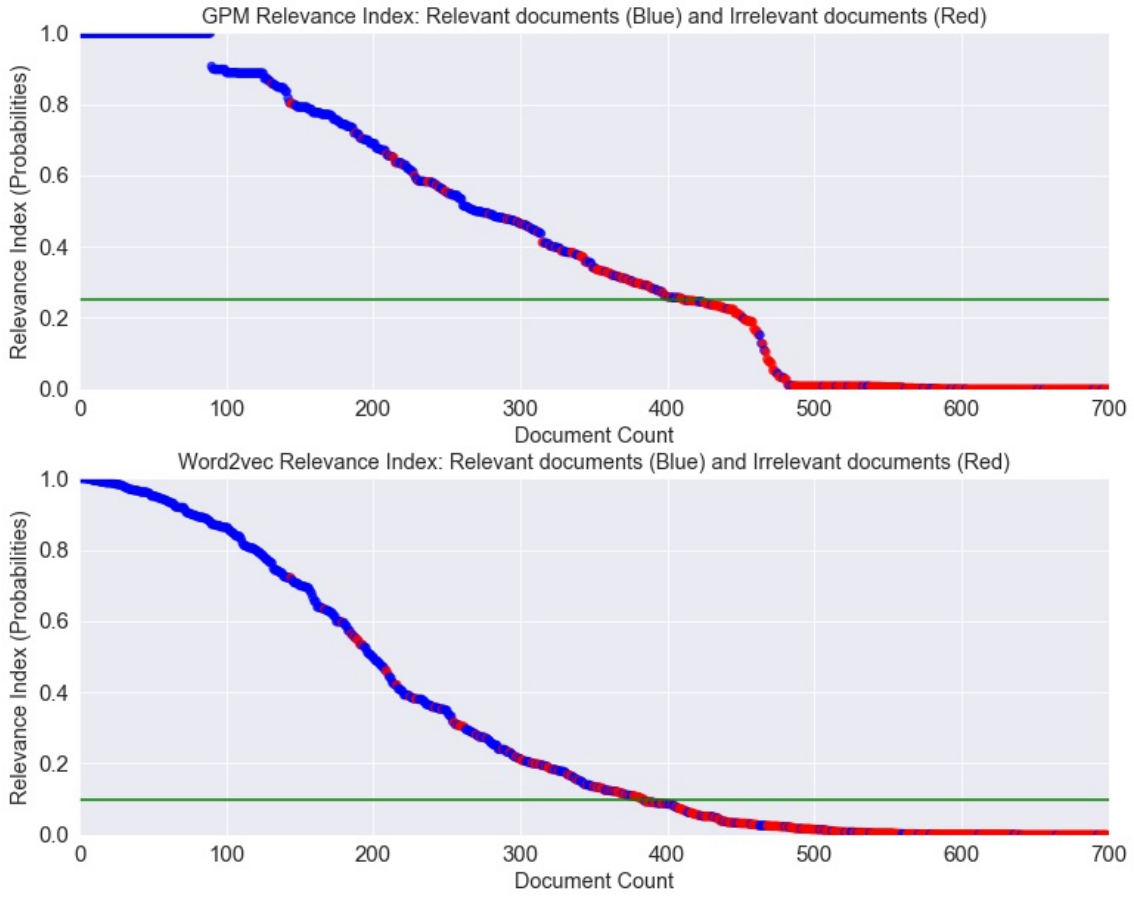


Figure 6.7: Relevance index results for GPM and word2vec on PAN-2012 corpus.

average accuracy, precision and F1 score for both GPM and word2vec was approximately 82%, 80% and 83%, respectively. In this application, recall is actually more important. Recall gives the proportion of true predatory conversations that were correctly identified. A higher recall is therefore desirable. For the selected threshold values, the GPM's recall is 89.5% whilst word2vec's recall is 87%. The results for different choices of threshold are summarised in Figure 6.8.[5] The GPM generally outperforms word2vec.

6.4.3 CORD-19 CORPUS

Unlike the previous corpora, the CORD-19 corpus is unlabelled, which is a common occurrence in real-life. This dataset will be used to demonstrate how one can pick a class of interest from an

[5]Figure 6.8 shows results from a single run of the codes.

Table 6.2: Evaluation of GPM and word2vec PAN-2012 dataset.

Metric	**PAN-2012**	
	GPM	word2vec
Accuracy	0.822 (0.029)	0.827 (0.003)
Precision	0.785 (0.048)	0.807 (0.005)
Recall	0.895 (0.014)	0.870 (0.002)
F1 Score	0.835 (0.021)	0.837 (0.002)

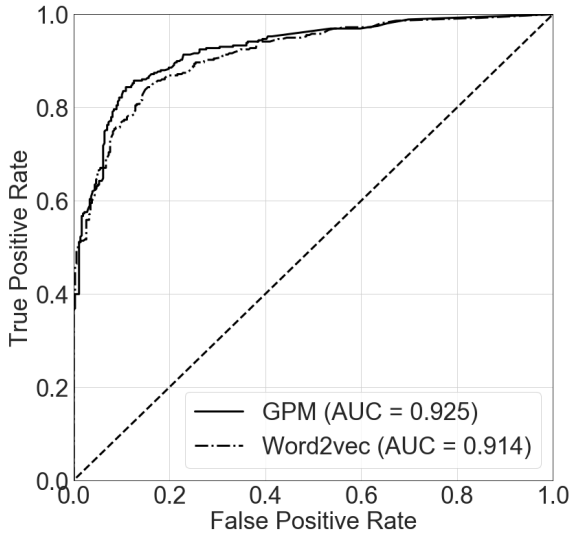


Figure 6.8: ROC curves for GPM and word2vec on PAN-2012 corpus.

unlabelled corpus and then determine the relevance of new documents. Word2vec will no longer be considered here, due to that absence of labels.

As discussed in the Section 6.2, the first step is to train the GPM model on the training corpus. By virtue of GPM being a topic model, the output will not only be a semantic representation of the documents, but also a set of topics. From here, the user can then pick a topic of interest. Once a topic of interest is chosen, the reference corpus is then chosen to be documents whose semantic representations assign the highest probability to the topic of interest. From here, the user can then follow the same procedure as before to determine relevance indices for new documents. The remainder of this section investigates this process on the CORD-19 corpus.

GPM was first trained on the 'comm_abstracts' subset of the CORD-19 dataset. Below are the top 10 words for 3 of the 10 topics that were found by the GPM model and the topic labels are

indicated in bold.[6]

- **Virology**: [‘use’, ‘sequence’, ‘study’, ‘method’, ‘gene’, ‘detect’, ‘protein’, ‘develop’, ‘genome’, ‘virus’]
- **Pulmonology**: [respiratory’, ‘coronavirus’, ‘syndrome’, ‘middle’, ‘east’, ‘severe’, ‘mers-cov’, ‘cause’, ‘acute’, ‘human’]
- **Immunology**: [‘study’, ‘use’, ‘vaccine’, ‘effect’, ‘human’, ‘antibody’, ‘active’, ‘treatment’, ‘infect’, ‘virus’]

The topic labels, ‘Virology’, ‘Pulmonology’ and ‘Immunology’, were manually assigned which is typically necessary in the application of topic models. Consequently, this makes the process subjective as different users may arrive at different labels. The topic labels provided for the 3 topics above were selected in consultation with a medical doctor. Based on the semantic representation of each document, the training set was then labelled according to its dominant topic[7]. The test set was then indexed against the trained GPM model, thus producing semantic representations of the test set. It is at this stage that the user could select a topic of interest an then calculate relevance indices.

In order to study the results in greater depth, let us go a step further and also label the test corpus according to the dominant topic of each document. These labels will allow for the graphical comparison of semantic representations for different topics as well as calculation of classification metrics.

Suppose that ‘Virology’ documents from the test set are selected as the reference corpus. The semantic representations can be compared with ‘Virology’ documents from the test corpus (similar query set) and ‘Pulmonology’ documents from the test corpus (dissimilar query set). These results are summarised in the top graph of Figure 6.9. The yellow graph shows the distribution of semantic similarities (adjusted Jensen-Shannon distances) between the ‘Virology’ reference set and the ‘Virology’ query set. The blue histogram, showing the distribution of semantic similarities between the‘Virology’ reference set and the ‘Pulmonology’ query set. Similarly, in the bottom

[6]These three topics were specifically selected because they were the 3 topics into which the documents in the test corpus, the ‘noncomm_abstracts’ subset, were assigned by the GPM topic model.

[7]The dominant topic of a document is the topic with the highest proportion in the document based on its semantic representation.

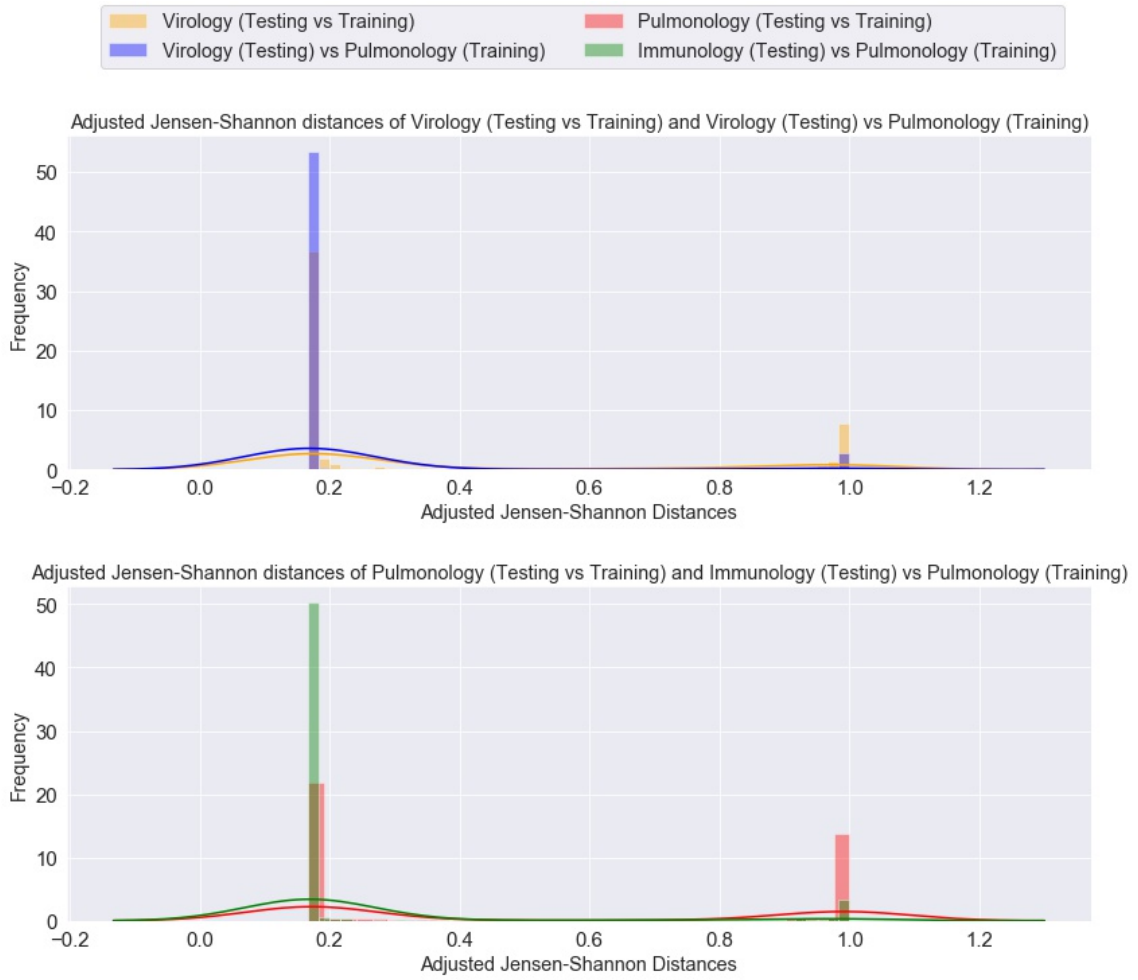


Figure 6.9: Distributions of semantic similarities between semantic representations of different documents from GPM.

graph of Figure 6.9, the red and green histogram shows semantic similarities for 'Pulmonology' (testing vs training) and 'Immunology. (testing) vs 'Pulmonology. (training), respectively. Ideal results should show high AJSD values between corpora that are similar and lower values for those that are dissimilar. In both graphs, most of the lower AJSD values arise from the comparison of the dissimilar sets, which is a desirable result. There is some overlap visible in both graphs around the lower AJSD values, which indicates that some of the semantic representations within the similar sets had a low similarity. Despite this, there were some semantic representations from the similar sets that produced high AJSD values. It is also interesting to note that there is very little overlap around the higher AJSD values. In other words, AJSD values between the dissimilar sets generally did not have high semantic similarity.

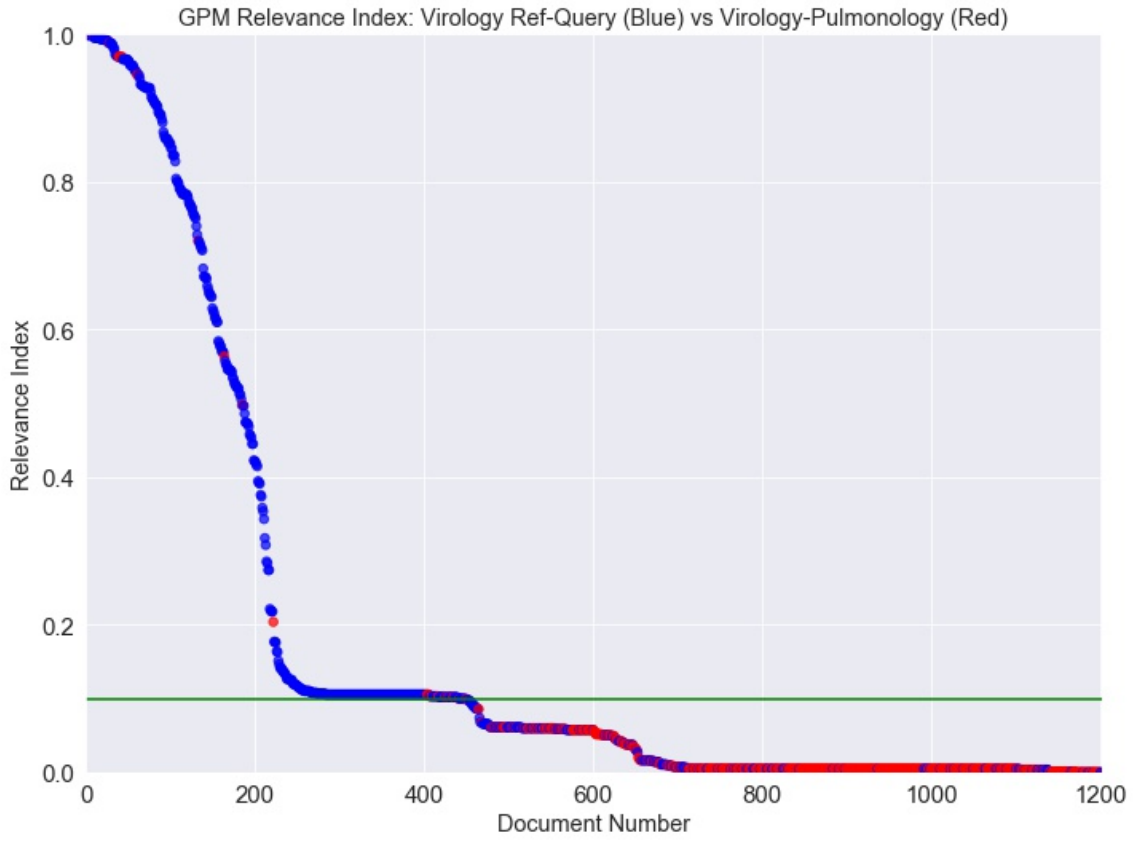


Figure 6.10: Relevance index results for GPM on CORD-19 dataset.

The next stage is to calculate the relevance indices. Due to the presence of labels, the accuracy, precision and recall of each model can be calculated by comparing these relevance indices with a threshold value. For this application, the value was chosen to be 0.1. The results are summarised graphically in Figure 6.10. The reference set in this example was 'Virology'. Relevance indices were then calculated between the reference corpus and the 'Virology' query set as well as the 'Pulmonology' query set. As previously mentioned, if the models are performing well, then relevant documents (blue) should appear above the green threshold line, whilst irrelevant documents (red) should appear below it. It is clear from Figure 6.10 that the GPM performed well. This can be quantified by looking at the classification metrics. The GPM model was run on the labelled documents ten times and the average and standard deviations of the classification metrics are shown in Table 6.3. At a threshold of 0.1, the average accuracy is 70.5% and the average precision is 93.1%, average recall is 63.6%.

Table 6.3: Averages and standard deviations of classification metrics for GPM on the CORD-19 corpus.

Metric	CORD-19	
	Mean	Standard Deviation
Accuracy	0.771	0.058
Precision	0.950	0.033
Recall	0.694	0.055
F1 Score	0.801	0.041

6.5 CONCLUSION

The objective of this chapter was to demonstrate the GPM model's ability to successfully create semantic representations that would produce high relevance indices for a test documents which were semantically similar to a reference corpus. It was shown that the GPM, a probabilistic DSM, was able to do this successfully. GPM and word2vec were applied to 3 different datasets and the key findings were as follows:

1. It was shown that NEWS-2020 dataset, GPM was better than Word2vec at distinguishing between new documents belonging to the 'Health' topic (relevant) and those belonging to the 'Science'. It was also shown on this dataset, that GPM was the better option when the size of the corpus was small.

2. In the PAN-2012 dataset, GPM was able to perform well on this small corpus of noisey short text. Furthermore, for similar precision, GPM had the better recall, thus making it the better option for this application.

3. The most important contribution was in the CORD-19 application, which is a fully unsupervised technique. It covers a scenario where a user may have a collection of unlabelled documents whose topics are unknown. Based on what is discovered by the topic model, the technique makes it possible for a user to then zoom in on a particular class of interest and find other relevant documents when new, unseen documents become available.

Apart from the experimental results, an experimental workflow for evaluating DSMs was also introduced. An important contribution s the definition of a relevance index, which is a normalized

performance metric to compare different similarity measures.

CHAPTER SEVEN

CONCLUSION

The study of topic models for short text remains an open area of research due the great abundance of short text corpora and the vast applications that follow. In this thesis, a new topic model for short text, the Gamma-Poisson mixture model was developed and a Python package for its implementation was created. The GPM was shown to outperform state-of-the-art topic models for short text based on different evaluation metrics. Various experiments were performed and it was found that GPM produced topics with better average topic coherence scores than GSDM and BTM on three datasets. A collapsed Gibbs sampler for the GPM was derived and it proved to be of great benefit to the model as it allowed for quick convergence and there was very little variation between different runs of the model. More importantly, the collapsed Gibbs sampler gave the new topic model the ability to automatically infer the number of topics contained in a corpus, which is typically a challenge in practice. When compared to GSDMM, the number of topics found by GPM on various datasets was found to be closer to the number of topics found by human annotators.

The usefulness of the new topic model was then shown in a real-world application: determining the relevance of a new corpus to a collection of documents of interest. In doing so, a framework for this was developed and a relevance index for the comparison of different similarity measures was defined. It was shown that the GPM was able to successfully produce semantic representations that allowed for the discrimination of relevant and irrelevant documents in short texts

that were also small in size. Upon identifying the prevalence of unlabelled documents in reality, a methodology for determining relevance in this unsupervised setting was proposed and shown to be successful.

Other contributions of this work include, establishing the validity of the Poisson distribution as a valid model for modelling short text. It was shown empirically that, contrary to popular belief, the Poisson distribution was appropriate for modelling text from short documents, as such texts do not necessarily display burstiness and over-dispersion. This formed part of the basis for the new model. In addition, this thesis presented a unifying framework that situated topic modelling in the wider context of other well-known statistical models. It was shown that topic models possess both dimensionality reduction and clustering capabilities.

There are several areas for future work.

- One area of interest is extending the model to allow each document to contain 1, 2 or 3 topics. Li et al. (2017a) extended the Dirichlet multinomial mixture model to allow for this more relaxed assumption and found that the performance of the model improved. As is, the GPM only assumes one topic per document and introducing this variant will add some flexibility that may be useful for documents that do not satisfy this assumption.

- GPM is not able to take advantage of external information about the relationships between words that can be derived from word-embeddings. Given the success of other researchers who incorporated such information via models such as the Generalised Pólya urn, it seems promising that GPM would benefit from this modification.

- GPM was mainly compared with GSDMM due to their shared ability to detect the number of topics automatically. Further experiments could still be performed comparing GPM to other topic models such as LF-DMM and GPU-DMM.

- GPM's performance was only compared with that of word2vec in the document similarity experiments. These experiments can be expanded to compare GPM with other models such as BTM and NMF.

- The focus of this research was on the topic modelling aspect of the models, but it is not uncommon to use short text topic models for clustering (Qiang et al., 2020). The performance of GPM was not assessed on its clustering capability, yet in the original paper of Yin and

Wang (2014), the GSDMM's clustering performance was compared with other models such as K-means (Jain, 2010) and DMAFP (Huang et al., 2012).

- It has been mentioned that selecting the number of topics can be achieved using non-parametric topic models. The hierarchical Dirichlet process (Teh et al., 2006) is one such model and it has be used for long text. Another area for further research would be the study of non-parametric topic models for short text, leading into a comparison of such models with GPM.

- Lastly, there is also room to expand the experiments around semantic similarity. This would include using other corpus genres as well as other word embeddings, such as tf-idf and GloVe, and Bidirectional Encoder Representations from Transformers (BERT).

REFERENCES

AGGARWAL, C. C. AND ZHAI, C. (2012). *Mining text data*. Springer Science & Business Media.

BISHOP, C. M. (2006). *Pattern recognition and machine learning*, volume 1. Springer New York.

BLEI, D. M., NG, A. Y., AND JORDAN, M. I. (2003). Latent Dirichlet allocation. *The Journal of Machine Learning Research*, **3**, 993–1022.

BORKO, H. AND BERNICK, M. (1963). Automatic document classification. *Journal of the ACM (JACM)*, **10** (2), 151–162.

BOUGUILA, N. (2010). Count data modeling and classification using finite mixtures of distributions. *IEEE Transactions on Neural Networks*, **22** (2), 186–198.

BRODY, S. AND ELHADAD, N. (2010). An unsupervised aspect-sentiment model for online reviews. *In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 804–812.

BUNTINE, W. (2002). Variational extensions to EM and multinomial PCA. *In European Conference on Machine Learning*. Springer, pp. 23–34.

BUNTINE, W. (2015). What is the difference between NMF and LDA? Why are the priors of LDA sparse-induced? [Online]. Available: https://www.quora.com/What-is-the-difference-between-NMF-and-LDA-Why-are-the-priors-of-LDA-sparse-induced. [Accessed: 09-Nov-2017].

BUNTINE, W. AND JAKULIN, A. (2006). Discrete component analysis. *In Subspace, Latent Structure and Feature Selection*. Springer, pp. 1–33.

CANNY, J. (2004). GaP: a factor model for discrete data. *In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 122–129.

CHARLET, D. AND DAMNATI, G. (2017). SimBow at SemEval-2017 task 3: Soft-cosine semantic similarity between questions for community question answering. *In Proceedings of the 11th Workshop on Semantic Evaluations (SemEval-2017)*. Association for Compuational Linguistics, Vancouver, Canada, pp. 315–319.

CHEN, J., GONG, Z., AND LIU, W. (2020). A dirichlet process biterm-based mixture model for short text stream clustering. *Applied Intelligence*, 1–11.

CHENG, X., YAN, X., LAN, Y., AND GUO, J. (2014). BTM: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, **26** (12), 2928–2941.

CHURCH, K. W. AND GALE, W. A. (1995). Poisson mixtures. *Natural Language Engineering*, **1** (2), 163–190.

COLLINS, M., DASGUPTA, S., AND SCHAPIRE, R. E. (2001). A generalization of principal components analysis to the exponential family. *In Advances in neural information processing systems*. pp. 617–624.

DEERWESTER, S. C., DUMAIS, S. T., LANDAUER, T. K., FURNAS, G. W., AND HARSHMAN, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, **41** (6), 391–407.

EROSHEVA, E., FIENBERG, S., AND LAFFERTY, J. (2004). Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, **101** (suppl 1), 5220–5227.

EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C. K., WINN, J., AND ZISSERMAN, A. (2010). The PASCAL visual object classes (voc) challenge. *International journal of computer vision*, **88** (2), 303–338.

EYHERAMENDY, S., LEWIS, D. D., AND MADIGAN, D. (2003). On the naive Bayes model for text categorization. *In Proceedings of the Ninth International Workshop on Artificial*

Intelligence and Statistics, AISTATS 2003, Key West, Florida, USA, January 3-6, 2003.
URL: *http://research.microsoft.com/en-us/um/cambridge/events/aistats2003/proceedings/165.pdf*

FEFFERMAN, C., MITTER, S., AND NARAYANAN, H. (2016). Testing the manifold hypothesis. *Joural of the American Mathematical Society*, **29** (4), 983–1049. doi:10.1090/jams/852.
URL: *http://dx.doi.org/10.1090/jams/852*

FIRTH, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

FRANCES, N. AND KUCERA, H. (1982). Frequency analysis of English usage.

GÁMEZ, J. A., RUMÍ, R., AND SALMERÓN, A. (2006). Unsupervised naive bayes for data clustering with mixtures of truncated exponentials. *In Probabilistic Graphical Models*. pp. 123–130.

GAO, W., PENG, M., WANG, H., ZHANG, Y., XIE, Q., AND TIAN, G. (2019). Incorporating word embeddings into topic modeling of short text. *Knowledge and Information Systems*, **61** (2), 1123–1145.

GEMAN, S. AND GEMAN, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6), 721–741.

GOPALAN, P., RUIZ, F. J., RANGANATH, R., AND BLEI, D. (2014). Bayesian nonparametric Poisson factorization for recommendation systems. *In Artificial Intelligence and Statistics*. pp. 275–283.

GRIFFITHS, T. (2002). Gibbs sampling in the generative model of latent Dirichlet allocation. Technical report, Stanford University.

GUO, Y., HUANG, Y., DING, Y., QI, S., WANG, X., AND LIAO, Q. (2020). GPU-BTM: A topic model for short text using auxiliary information. *In 2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*. IEEE, pp. 198–205.

HEINRICH, G. (2005). Parameter estimation for text analysis. Technical report, Technical report.

HOFMANN, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, **22** (1), 89–115.

HONG, L. AND DAVISON, B. D. (2010). Empirical study of topic modeling in Twitter. *In Proceedings of the First Workshop on Social Media Analytics*. ACM, pp. 80–88.

HUANG, J., PENG, M., LI, P., HU, Z., AND XU, C. (2020). Improving biterm topic model with word embeddings. *World Wide Web*, 1–26.

HUANG, R., YU, G., WANG, Z., ZHANG, J., AND SHI, L. (2012). Dirichlet process mixture model for document clustering with feature partition. *IEEE Transactions on knowledge and data engineering*, **25** (8), 1748–1759.

INCHES, G. AND CRESTANI, F. (2012). Overview of the international sexual predator identification competition at PAN-2012. *In CLEF (Online working notes/labs/workshop)*, volume 30.

INOUYE, D., RAVIKUMAR, P., AND DHILLON, I. (2014). Admixture of poisson MRFs: A topic model with word dependencies. *In International Conference on Machine Learning*. pp. 683–691.

INOUYE, D. I., YANG, E., ALLEN, G. I., AND RAVIKUMAR, P. (2017). A review of multivariate distributions for count data derived from the poisson distribution. *Wiley Interdisciplinary Reviews: Computational Statistics*, **9** (3), e1398.

ISHWARAN, H. AND RAO, J. S. (2005). Spike and slab variable selection: frequentist and Bayesian strategies. *Annals of statistics*, 730–773.

JAIN, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, **31** (8), 651–666.

JIANG, H., ZHOU, R., ZHANG, L., WANG, H., AND ZHANG, Y. (2017). A topic model based on Poisson decomposition. *In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, pp. 1489–1498.

JOHNSON, N. L., KEMP, A. W., AND KOTZ, S. (2005). *Univariate discrete distributions*, volume 444. John Wiley & Sons.

JOLLIFFE, I. T. (1986). Principal component analysis and factor analysis. *In Principal component analysis*. Springer, pp. 115–128.

JURAFSKY, D. AND MARTIN, J. H. (2019). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Third edition. doi:10.1515/zfsw.2002.21.1.134.

KULLBACK, S. AND LEIBLER, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, **22** (1), 79–86.

LEE, D. D. AND SEUNG, H. S. (2001). Algorithms for non-negative matrix factorization. *In Advances in neural information processing systems*. pp. 556–562.

LEE, M. D. AND WAGENMAKERS, E.-J. (2014). *Bayesian cognitive modeling: A practical course*. Cambridge university press.

LEVY, O., GOLDBERG, Y., AND DAGAN, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, **3**, 211–225.

LI, C., DUAN, Y., WANG, H., ZHANG, Z., SUN, A., AND MA, Z. (2017a). Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Transactions on Information Systems (TOIS)*, **36** (2), 11.

LI, C., WANG, H., ZHANG, Z., SUN, A., AND MA, Z. (2016a). Topic modeling for short texts with auxiliary word embeddings. *In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, pp. 165–174.

LI, J., LIAO, M., GAO, W., HE, Y., AND WONG, K.-F. (2016b). Topic extraction from microblog posts using conversation structures. *In ACL (1)*. World Scientific.

LI, X., LI, C., CHI, J., AND OUYANG, J. (2017b). Short text topic modeling by exploring original documents. *Knowledge and Information Systems*, 1–20.

LI, X., ZHANG, A., LI, C., GUO, L., WANG, W., AND OUYANG, J. (2019). Relational biterm topic model: Short-text topic modeling using word embeddings. *The Computer Journal*, **62** (3), 359–372.

LIANG, W., FENG, R., LIU, X., LI, Y., AND ZHANG, X. (2018). GLTM: A global and local word embedding-based topic model for short texts. *IEEE Access*, **6**, 43612–43621.

LIN, C. X., ZHAO, B., MEI, Q., AND HAN, J. (2010). PET: a statistical model for popular events tracking in social communities. *In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 929–938.

LIN, T., TIAN, W., MEI, Q., AND CHENG, H. (2014). The dual-sparse topic model: mining focused topics and focused terms in short text. *In Proceedings of the 23rd International Conference on World Wide Web*. ACM, pp. 539–550.

MADSEN, R. E., KAUCHAK, D., AND ELKAN, C. (2005). Modeling word burstiness using the Dirichlet distribution. *In Proceedings of the 22nd international conference on Machine learning*. ACM, pp. 545–552.

MAHMOUD, H. (2008). *Pólya urn models*. CRC press.

MAZARURA, J. AND DE WAAL, A. (2016). A comparison of the performance of latent Dirichlet allocation and the Dirichlet multinomial mixture model on short text. *In Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), 2016*. IEEE, pp. 1–6.

MAZARURA, J., DE WAAL, A., AND DE VILLIERS, P. (2020). A Gamma-Poisson mixture topic model for short text. *Mathematical Problems in Engineering*, **2020**.

MCCALLUM, A. (1999). Multi-label text classification with a mixture model trained by EM. *In AAAI workshop on Text Learning*. pp. 1–7.

MCCALLUM, A. AND NIGAM, K. (1998). A comparison of event models for naive Bayes text classification. *In AAAI-98 workshop on learning for text categorization*, volume 752. Citeseer, pp. 41–48.

MEHROTRA, R., SANNER, S., BUNTINE, W., AND XIE, L. (2013). Improving LDA topic models for microblogs via tweet pooling and automatic labeling. *In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 889–892.

MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

MIMNO, D., WALLACH, H. M., TALLEY, E., LEENDERS, M., AND MCCALLUM, A. (2011). Optimizing semantic coherence in topic models. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 262–272.

MOHAMED, S., GHAHRAMANI, Z., AND HELLER, K. A. (2009). Bayesian exponential family PCA. *In Advances in neural information processing systems*. pp. 1089–1096.

MURPHY, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. First edition. MIT Press, Cambridge, Massachusetts. doi:10.1007/978-94-011-3532-0{_}2.

NGUYEN, D. Q., BILLINGSLEY, R., DU, L., AND JOHNSON, M. (2015). Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, **3**, 299–313.

NIEPERT, M. AND DOMINGOS, P. (2014). Exchangeable variable models. *In International Conference on Machine Learning*. pp. 271–279.

NIGAM, K., MCCALLUM, A. K., THRUN, S., AND MITCHELL, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine learning*, **39** (2-3), 103–134.

Ó SÉAGHDHA, D. AND KORHONEN, A. (2014). Probabilistic distributional semantics with latent variable models. *Computational Linguistics*, **40** (3), 587–631.

OGURA, H., AMANO, H., AND KONDO, M. (2013). Gamma-Poisson distribution model for text categorization. *ISRN Artificial Intelligence*, **2013**.

OGURA, H., AMANO, H., AND KONDO, M. (2014). Classifying documents with poisson mixtures. *Transactions on Machine Learning and Artificial Intelligence*, **2** (4), 48–76.

PENNINGTON, J., SOCHER, R., AND MANNING, C. D. (2014). Glove: Global vectors for word representation. *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pp. 1532–1543.

PHAN, X.-H., NGUYEN, C.-T., LE, D.-T., NGUYEN, L.-M., HORIGUCHI, S., AND HA, Q.-T. (2011). A hidden topic-based framework toward building applications with short web documents. *IEEE Transactions on Knowledge and Data Engineering*, **23** (7), 961–976.

PHAN, X.-H., NGUYEN, L.-M., AND HORIGUCHI, S. (2008). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. *In Proceedings of the 17th international conference on World Wide Web*. ACM, pp. 91–100.

QIANG, J., QIAN, Z., LI, Y., YUAN, Y., AND WU, X. (2020). Short text topic modeling techniques, applications, and performance: a survey. *IEEE Transactions on Knowledge and Data Engineering*.

QUAN, X., KIT, C., GE, Y., AND PAN, S. J. (2015). Short and sparse text topic modeling via self-aggregation. *In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*. pp. 2270–2276.

RIGOUSTE, L., CAPPÉ, O., AND YVON, F. (2007). Inference and evaluation of the multinomial mixture model for text clustering. *Information processing & management*, **43** (5), 1260–1280.

SHI, L., DU, J., LIANG, M., AND KOU, F. (2019). Dynamic topic modeling via self-aggregation for short text streams. *Peer-to-Peer Networking and Applications*, **12** (5), 1403–1417.

SITIKHU, P., PAHI, K., THAPA, P., AND SHAKYA, S. (2019). A comparison of semantic similarity methods for maximum human interpretability. *In 2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1. IEEE, pp. 1–4.

SUBRAMANIAN, G. (2015). *Python Data Science Cookbook*. Packt Publishing Ltd.

TEH, Y., JORDAN, M., BEAL, M., AND BLEI, D. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, **101** (476), 1566–1581.

TIPPING, M. E. AND BISHOP, C. M. (1999). Mixtures of probabilistic principal component analyzers. *Neural computation*, **11** (2), 443–482.

WANG, C. AND BLEI, D. M. (2011). Collaborative topic modeling for recommending scientific articles. *In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 448–456.

WANG, L. L., LO, K., CHANDRASEKHAR, Y., REAS, R., YANG, J., BURDICK, D., EIDE, D., FUNK, K., KATSIS, Y., KINNEY, R., LI, Y., LIU, Z., MERRILL, W., MOONEY, P., MURDICK, D., RISHI, D., SHEEHAN, J., SHEN, Z., STILSON, B., WADE, A. D., WANG, K., XIN,

N., WANG, R., WILHELM, C., XIE, B., RAYMOND, D., WELD, D. S., ETZIONI, O., AND KOHLMEIER, S. (2020). CORD-19: The COVID-19 Open Research Dataset. *In Proceedings of the Workshop on NLP for COVID-19 at ACL 2020*. Association for Computational Linguistics. **URL:** *https://arxiv.org/abs/2004.10706*

WELLING, M., CHEMUDUGUNTA, C., AND SUTTER, N. (2008). Deterministic latent variable models and their pitfalls. *In Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM, pp. 196–207.

WENG, J., LIM, E.-P., JIANG, J., AND HE, Q. (2010). Twitterrank: finding topic-sensitive influential twitterers. *In Proceedings of the Third ACM International Conference on Web Search and Data Mining*. ACM, pp. 261–270.

YAN, X., GUO, J., LAN, Y., AND CHENG, X. (2013). A biterm topic model for short texts. *In Proceedings of the 22nd International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 1445–1456.

YIN, J. AND WANG, J. (2014). A Dirichlet multinomial mixture model-based approach for short text clustering. *In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 233–242.

YU, J. AND QIU, L. (2019). ULW-DMM: An effective topic modeling method for microblog short text. *IEEE Access*, **7**, 884–893.

ZHANG, J. AND PIRAMUTHU, S. (2018). Product recommendation with latent review topics. *Information Systems Frontiers*, **20** (3), 617–625.

ZHANG, X., FENG, R., AND LIANG, W. (2018). Short text topic model with word embeddings and context information. *In International Conference on Computing and Information Technology*. Springer, pp. 55–64.

ZHAO, W. X., JIANG, J., WENG, J., HE, J., LIM, E.-P., YAN, H., AND LI, X. (2011). Comparing Twitter and traditional media using topic models. *In European Conference on Information Retrieval*. Springer, pp. 338–349.

ZHOU, K. AND YANG, Q. (2018). LDA-PSTR: A topic modeling method for short text. *In International Conference on Advanced Data Mining and Applications*. Springer, pp. 339–352.

ZHU, J. AND XING, E. P. (2011). Sparse topical coding. **831**, 838.

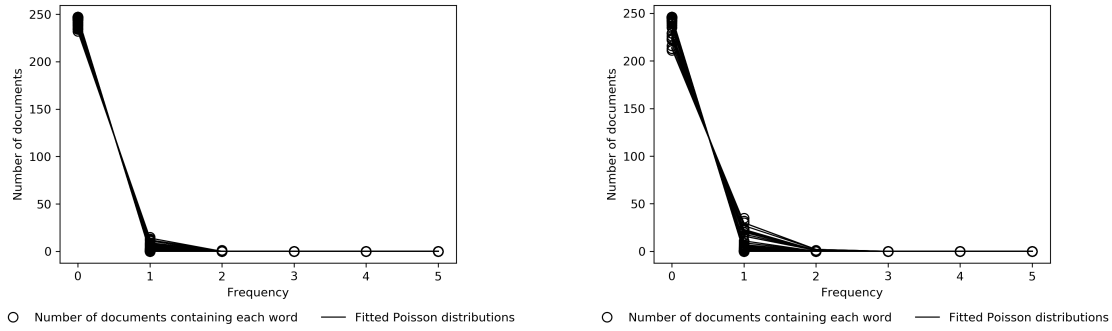
ZUO, Y., WU, J., ZHANG, H., LIN, H., WANG, F., XU, K., AND XIONG, H. (2016a). Topic modeling of short texts: A pseudo-document view. *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 2105–2114.

ZUO, Y., ZHAO, J., AND XU, K. (2016b). Word network topic model: a simple but general solution for short and imbalanced texts. *Knowledge & Information Systems*, **48** (2).

APPENDIX

I WORD FREQUENCY GRAPHS

Figures 7.1 to 7.3 show the distribution of different words across different corpora for different classes. The corpora used here are described in Section 5.2. In each graph, the circles represent the number of documents in which each of the words appears 0, 1, 2, ..., 5 times in the respective corpus and the straight lines indicate the predicted Poisson distribution associated with each word. Apart from a few words, the Poisson distribution appears to be a good fit. None of these graphs displays the heavy tails that were observed by Church and Gale (1995) in the Brown corpus.

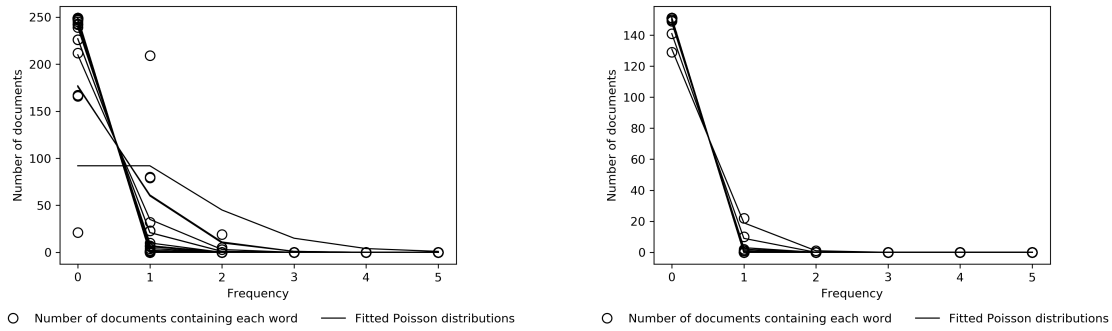


(a) Class 1 (b) Class 2

Figure 7.1: Pascal Flickr corpus.

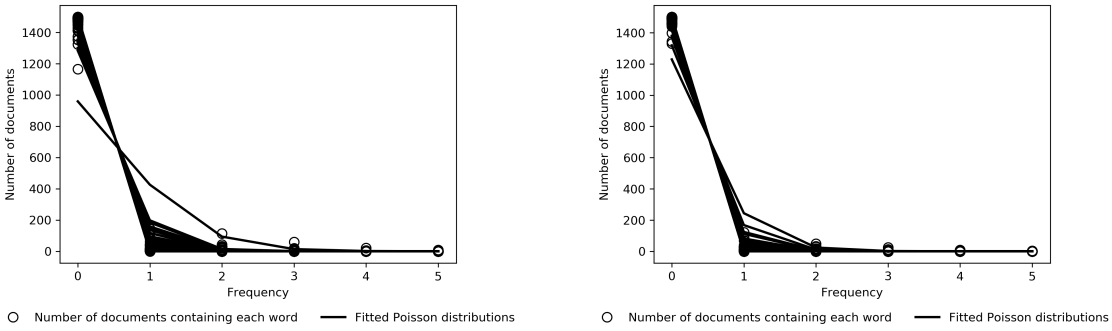
II COMPARISON OF MEANS AND VARIANCES

Scatterplots of the means of word occurrences against their variances are shown in Figures 7.4 to 7.6. In most cases, means are less than variances. Only, Search Snippets corpus appears to display overdispersion.



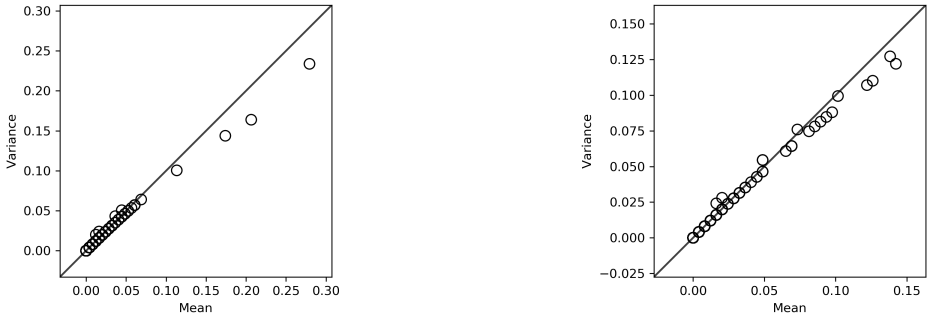
(a) Class 99 (b) Class 88

Figure 7.2: Tweet corpus.



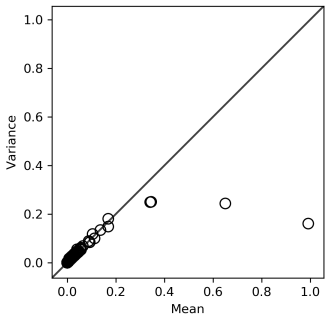
(a) Class 1 (b) Class 2

Figure 7.3: Search Snippets corpus.

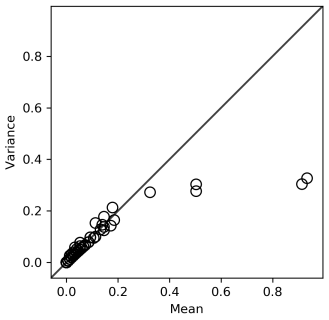


(a) Class 1 (b) Class 2

Figure 7.4: Pascal Flickr corpus.

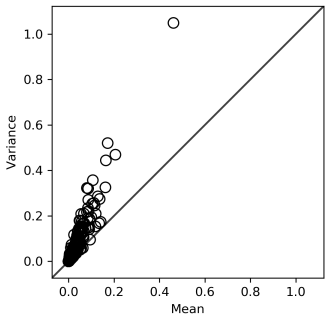


(a) Class 99

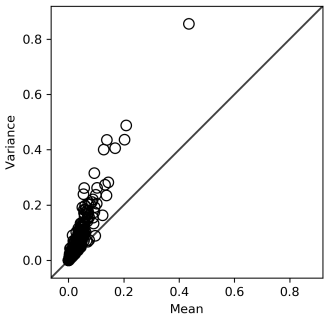


(b) Class 88

Figure 7.5: Tweet corpus.



(a) Class 1



(b) Class 2

Figure 7.6: Search Snippets corpus.

III ASSESSMENT OF BURSTINESS IN OTHER CORPORA

Tables 7.1 to 7.3 give an indication of whether there is burstiness in the top 10 occurring words in a few of the classes.

In the Pascal Flickr and Tweet corpora, there does not appear to be significant burtsiness. The words in the Search Snippets corpus appear to display some burstiness. A possible reason for the corpus displaying characteristics as that of the long text Brown corpus could be that the topics are too broad. The 12 295 document corpus is divided into only 8 categories and these categories can be subdivided further into subtopics. For instance, subtopics such as Swimming and Football would fall under the Sports topic. Words such as “swim” or “pool” may not be common throught

the entire corpus, but are likely to occur in high concentration in documents about swimming. Thus, it is not surprising to observe burstiness in this dataset.

Table 7.1: Pascal Flickr corpus.

Word	Frequency	df	Predicted df
bike	69	65	60
man	51	51	46
riding	43	43	39
person	28	28	26
wearing	17	17	16
front	15	15	15
girl	15	15	15
young	15	15	15
helmet	15	15	15
street	14	14	14

Class 1

Word	Frequency	df	Predicted df
water	35	35	33
white	34	33	32
perched	31	31	29
small	30	30	28
sitting	25	24	24
standing	24	24	23
black	23	23	22
green	22	22	21
branch	21	21	20
blue	20	20	19

Class 2

Table 7.2: Tweet corpus.

Word	Frequency	df	Predicted df
commercial	247	228	157
superbowl	162	160	119
super	86	83	73
bowl	85	82	72
ad	42	37	39
doritos	42	41	39
best	34	32	32
volkswagen	28	28	26
pepsi	26	23	25
youtube	23	23	22

Class 99

Word	Frequency	df	Predicted df
king	141	122	92
speech	138	121	90
award	76	72	60
oscar	76	74	60
nomination	49	45	42
academy	28	27	26
best	27	22	25
sag	26	26	24
lead	22	22	20
win	22	18	20

Class 88

Table 7.3: Search Snippets corpus.

Word	Frequency	df	Predicted df
business	692	334	554
market	309	160	279
trade	258	109	237
stock	246	112	227
news	243	146	224
com	212	176	198
economic	205	120	192
information	202	168	189
finance	193	105	181
financial	179	124	169

Class 1

Word	Frequency	df	Predicted df
computer	653	357	529
web	313	175	283
software	304	169	275
programming	253	120	233
wikipedia	216	105	201
memory	208	82	194
com	206	157	192
internet	198	120	185
intel	191	73	179
information	185	152	174

Class 2

IV PERFORMANCE MEASURES FOR NORMALISATION METHOD 1

Figures 7.7 and 7.8 show a comparison of the topics found, coherence and runtime for $N = 10, 20, 30$ on the Tweet and Search Snippets corpora, respectively, for normalisation method 1.

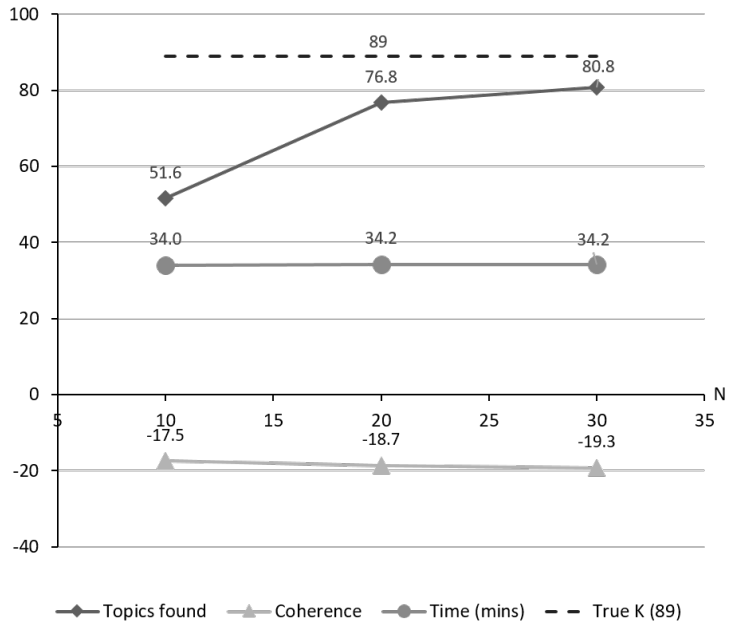


Figure 7.7: Number of topics found, average coherence and runtime of the GPM for $N = 10, 20, 30$ on the Tweet corpus.

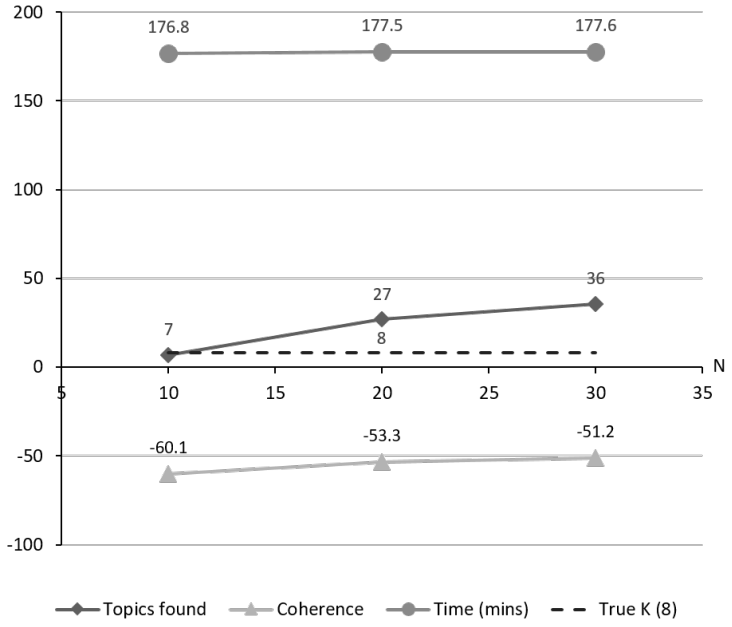


Figure 7.8: Number of topics found, average coherence and runtime of the GPM for $N = 10, 20, 30$ on the Search Snippets corpus.

V DERIVATION OF NORMALISATION METHOD 2

The topic estimates are only dependent on the topic assignments, thus it is only necessary to sample the topic assignment for each document. This is achieved by sampling from the conditional probability of a document belonging to a class,

$$p\left(\mathbf{z}_m = z|\mathbf{z}^{(m)}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\right) = \frac{p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{p\left(\boldsymbol{x}, \mathbf{z}^{(m)}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma\right)} \propto \frac{p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{p\left(\boldsymbol{x}^{(m)}, \mathbf{z}^{(m)}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\right)}. \tag{A1}$$

Owing to conditional independence between $\boldsymbol{x}$ and $\boldsymbol{z}$, it follows that

$$p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta})p(\mathbf{z}|\boldsymbol{\gamma}). \tag{A2}$$

It was shown in Equation 4.7 that the second term on the right hand side of Equation A2 can be expressed as

$$p(\mathbf{z}|\boldsymbol{\gamma}) = \frac{\Delta(\boldsymbol{m} + \boldsymbol{\gamma})}{\Delta(\boldsymbol{\gamma})}. \tag{A3}$$

where $\boldsymbol{m} = [m_1, m_2, \ldots, m_K]$ and m_k denotes the number of documents assigned to the k-th topic, $\Delta(\boldsymbol{\gamma}) = \frac{\prod_{k=1}^{K}\Gamma(\gamma_k)}{\Gamma(\sum_{k=1}^{K}\gamma_k)}$ and $\Delta(\boldsymbol{m}+\boldsymbol{\gamma}) = \frac{\prod_{k=1}^{K}\Gamma(m_k+\gamma_k)}{\Gamma(\sum_{k=1}^{K}(m_k+\gamma_k))}$.

The first term on the right-hand side of Equation A2, can be expressed as

$$p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \int p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\lambda}) p(\boldsymbol{\lambda}|\boldsymbol{\alpha}, \boldsymbol{\beta}) d\boldsymbol{\lambda}. \tag{A4}$$

Unlike normalisation method 1 (direct document length normalisation) which assumed the frequency of each word, x_{mv}, was modelled as

$$x_{mv}|z = k \sim Poi(\lambda_{kv}),$$

normalisation method 2 (modelling document length in the topic model) assumes

$$x_{mv}|z = k \sim Poi(N_m\lambda_{kv}),$$

where N_m denotes the number of words in the m-th document. Consequently,

$$p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\lambda}) = \prod_{m=1}^{M}\prod_{v=1}^{V} p\left(x_{mv}|N_m\lambda_{kv}\right) = \prod_{m=1}^{M}\prod_{v=1}^{V} \frac{(N_m\lambda_{kv})^{x_{mv}} e^{-N_m\lambda_{kv}}}{x_{mv}!}. \tag{A5}$$

Equation A5 can be re-expressed by the introduction of n_k, then number of words assigned to the k-th topic, and n_{kv}, the number of times word v is observed in topic k, as

$$p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\lambda}) = \prod_{k=1}^{K}\prod_{v=1}^{V} \frac{N_M^{N_M}\lambda_{kv}^{n_{kv}} e^{-n_k\lambda_{kv}}}{\boldsymbol{x!}}, \tag{A6}$$

where $\boldsymbol{x}! = \prod_{m=1}^{M}\prod_{v=1}^{V} x_{mv}$. By assuming a Gamma distribution for $\boldsymbol{\lambda}$ and substituting Equation

A6 into Equation A4, we obtain

$$
\begin{aligned}
p(\boldsymbol{x}|\mathbf{z},\boldsymbol{\alpha},\boldsymbol{\beta}) &= \int p(\boldsymbol{x}|\mathbf{z},\boldsymbol{\lambda})p(\boldsymbol{\lambda}|\boldsymbol{\alpha},\boldsymbol{\beta})d\boldsymbol{\lambda} \\
&= \int \prod_{k=1}^{K}\prod_{v=1}^{V} \frac{N_M^{N_M}\lambda_{kv}^{n_{kv}}e^{-n_k\lambda_{kv}}}{\boldsymbol{x!}} \times \frac{\lambda_{kv}^{\alpha_v-1}e^{-\frac{\lambda_{kv}}{\beta_v}}}{\Gamma(\alpha_v)\,\beta_v^{\alpha_v}}d\lambda_{kv} \\
&= \prod_{k=1}^{K}\prod_{v=1}^{V} \frac{N_M^{N_M}}{\boldsymbol{x}!\Gamma(\alpha_v)\,\beta_v^{\alpha_v}} \int \lambda_{kv}^{n_{kv}+\alpha_v-1}e^{-\lambda_{kv}\left(n_k+\frac{1}{\beta_v}\right)}d\lambda_{kv} \\
&= \prod_{k=1}^{K}\prod_{v=1}^{V} \frac{N_M^{N_M}}{\boldsymbol{x}!\Gamma(\alpha_v)\,\beta_v^{\alpha_v}} \times \Gamma(n_{kv}+\alpha_v)\left(\frac{\beta_v}{n_k\beta_v+1}\right)^{n_{kv}+\alpha_v} \\
&= \prod_{k=1}^{K}\prod_{v=1}^{V} \frac{\Gamma(n_{kv}+\alpha_v)}{\boldsymbol{x}!\Gamma(\alpha_v)} \times \frac{N_M^{N_M}\beta_v^{n_{kv}}}{(n_k\beta_v+1)^{n_{kv}+\alpha_v}}.
\end{aligned} \tag{A7}
$$

The integral is solved by multiplying the equation by $\Gamma(n_{kv}+\alpha_v)\left(\frac{\beta_v}{n_k\beta_v+1}\right)^{n_{kv}+\alpha_v}$ divided by itself. The result is an integral over a gamma distribution with parameters $n_{kv}+\alpha_v$ and $\frac{\beta_v}{n_k\beta_v+1}$. By substituting Equation A3 and A7, Equation A2 can now be written as

$$
\begin{aligned}
p(\boldsymbol{x},\mathbf{z}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) &= p(\boldsymbol{x}|\mathbf{z},\boldsymbol{\alpha},\boldsymbol{\beta})p(\mathbf{z}|\boldsymbol{\gamma}) \\
&= \frac{\Delta(\boldsymbol{m}+\boldsymbol{\gamma})}{\Delta(\boldsymbol{\gamma})} \prod_{k=1}^{K}\prod_{v=1}^{V} \frac{\Gamma(n_{kv}+\alpha_v)}{\boldsymbol{x}!\Gamma(\alpha_v)} \times \frac{N_M^{N_M}\beta_v^{n_{kv}}}{(n_k\beta_v+1)^{n_{kv}+\alpha_v}}.
\end{aligned} \tag{A8}
$$

The derivation of the conditional distribution in Equation A1 can now be concluded by substituting Equation A8 and applying the property of the Γ function, $\frac{\Gamma(x+m)}{\Gamma(x)} = \prod_{j=1}^{m}(x+j-1)$, as

follows

$$
\begin{aligned}
& p\left(\mathbf{z}_m = z|\mathbf{z}^{(m)}, \boldsymbol{x}\right) \\
& \propto \frac{p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{p\left(\boldsymbol{x}^{(m)}, \mathbf{z}^{(m)}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\right)} \\
& = \frac{\Delta(\boldsymbol{m}+\boldsymbol{\gamma})}{\Delta\left(\boldsymbol{m}^{(m)}+\boldsymbol{\gamma}\right)} \times \prod_{v=1}^{V}\left(\frac{\Gamma\left(n_{zv}+\alpha_v\right)}{\Gamma\left(n_{zv}^{(m)}+\alpha_v\right)}\right)\left(\frac{N_m^{N_m}}{N_m^{N_m}}\right)\left(\frac{\beta_v^{n_{zv}}}{\beta_v^{n_{zv}^{(m)}}}\right)\left(\frac{x^{(m)}!\Gamma\left(\alpha_v\right)}{x!\Gamma\left(\alpha_v\right)}\right) \\
& \qquad \times\left(\frac{\left(n_z^{(m)}\beta_v+1\right)^{n_{zv}^{(m)}+\alpha_v}}{\left(n_z\beta_v+1\right)^{n_{zv}+\alpha_v}}\right) \\
& = \frac{m_z^{(m)}+\gamma_z}{M-1+\sum_{k=1}^{K}\gamma_k} \times \prod_{v=1}^{V}\prod_{j=1}^{x_{mv}}\left(n_{zv}^{(m)}+\alpha_v+j-1\right) \times \beta_v^{x_{mv}} \times \frac{1}{\boldsymbol{x}_m!} \\
& \qquad \times \frac{\left(n_z^{(m)}\beta_v+1\right)^{n_{zv}^{(m)}+\alpha_v}}{\left(n_z^{(m)}\beta_v+N_m\beta_v+1\right)^{n_{zv}^{(m)}+x_{mv}+\alpha_v}}, \qquad \text{(A9)}
\end{aligned}
$$

where $n_{zv} = n_{zv}^{(m)} + x_{mv}$ and $n_z = n_z^{(m)} + N_m$. If it is assumed that $\alpha_v = \alpha$, $\beta_v = \beta$ and $\gamma_k = \gamma$ for all v and k, then Equation A9 simplifies to

$$
\begin{aligned}
& p\left(\mathbf{z}_m = z|\mathbf{z}^{(m)}, \boldsymbol{x}\right) \\
& \propto \frac{m_z^{(m)}+\gamma_z}{M-1+K\gamma} \times \frac{\beta^{n_m}}{\mathbf{x}_m!} \times \frac{\left(n_z^{(m)}+1\right)^{n_z^{(m)}+V\alpha}}{\left(n_z^{(m)}\beta+N_m\beta+1\right)^{n_z^{(m)}+n_m+V\alpha}} \qquad \text{(A10)} \\
& \qquad \times \prod_{v=1}^{V}\prod_{j=1}^{x_{mv}}\left(n_{zv}^{(m)}+\alpha+j-1\right),
\end{aligned}
$$

thus concluding the derivation of Equation 4.14.

VI COMPARISON OF NORMALISATION METHODS 1 AND 2

Figures 7.9 and 7.10 compare the topics found, coherence scores and runtimes of normalisation methods 1 (direct document length normalisation) and 2 (modelling document length in the topic model).

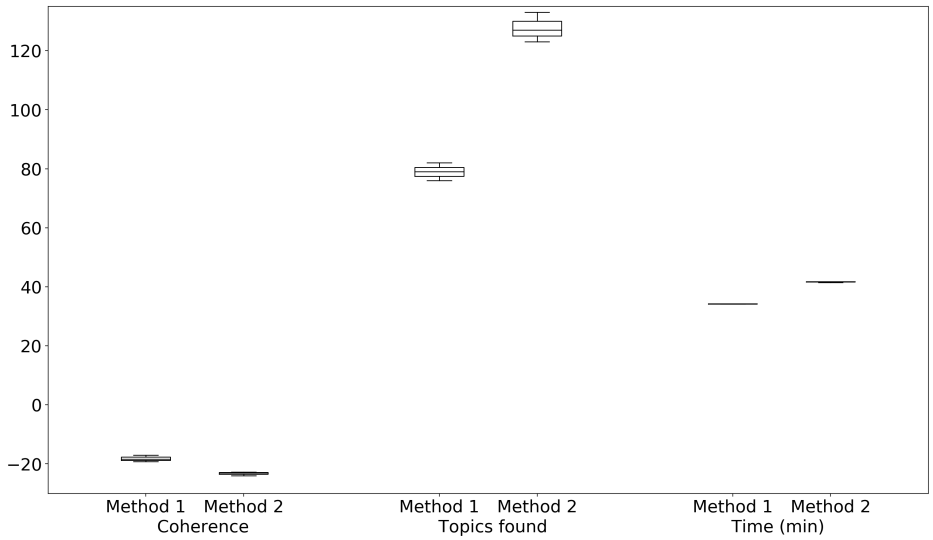


Figure 7.9: Comparison of average coherence, number of topics found and runtime of GPM under normalisation methods 1 and 2 on the Tweet corpus (True K = 89).

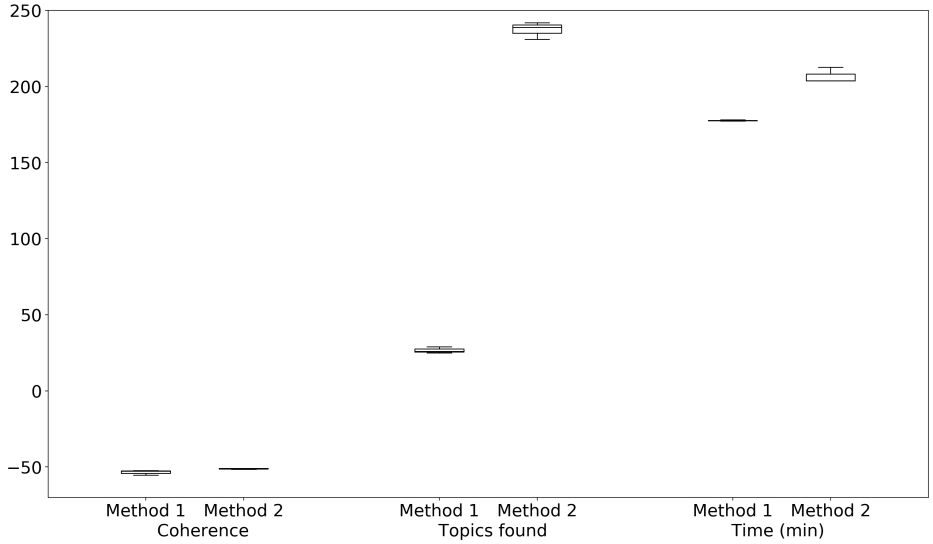


Figure 7.10: Comparison of average coherence, number of topics found and runtime of GPM under normalisation methods 1 and 2 on the Search Snippet corpus (True K = 8).

VII INFLUENCE OF GAMMA

Figures 7.11 and 7.12 show the influence of the choice of gamma on the number of topics found for the Tweet and Search Snippets datasets, respectively.

Figures 7.13 and 7.14 show the influence of the choice of gamma on the topic coherence scores for the Tweet and Search Snippets datasets, respectively.

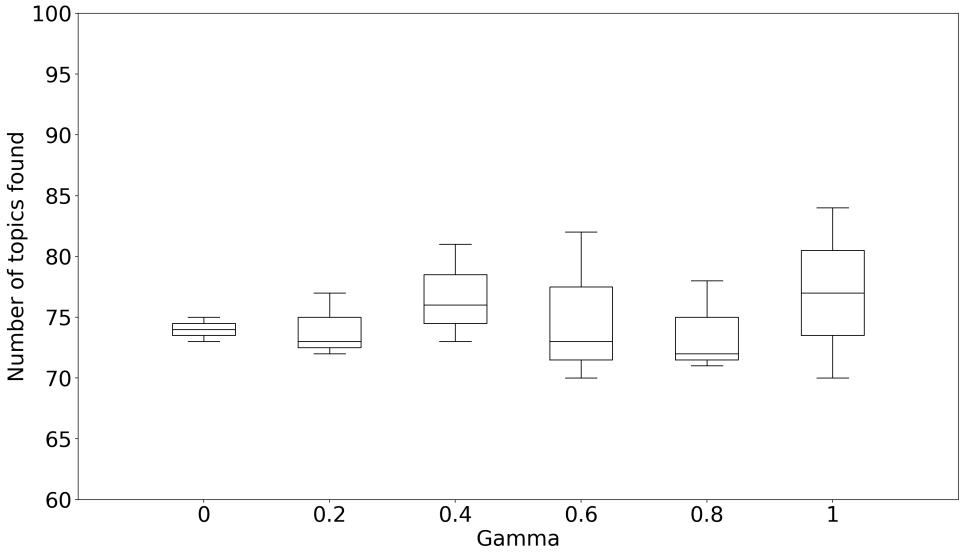


Figure 7.11: Influence of gamma on number of topics found.

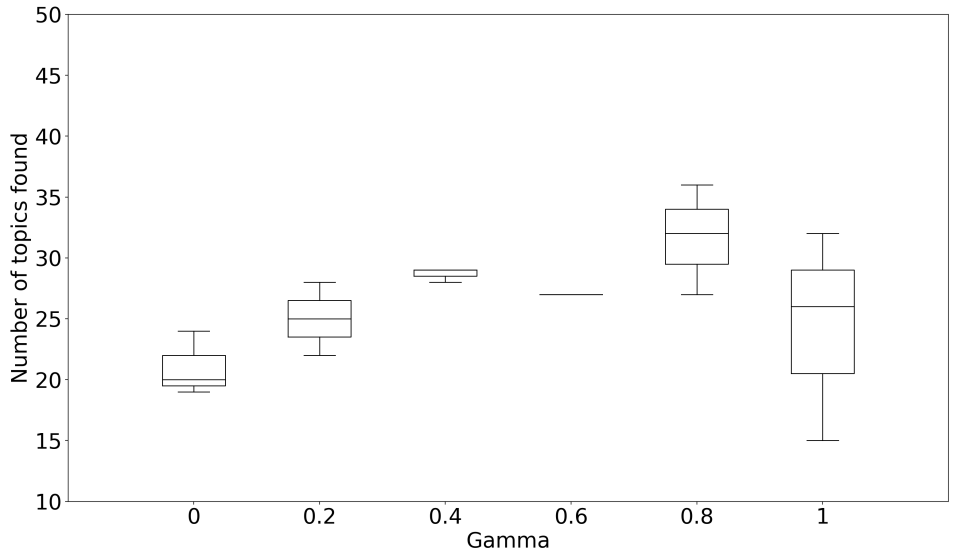


Figure 7.12: Influence of gamma on number of topics found.

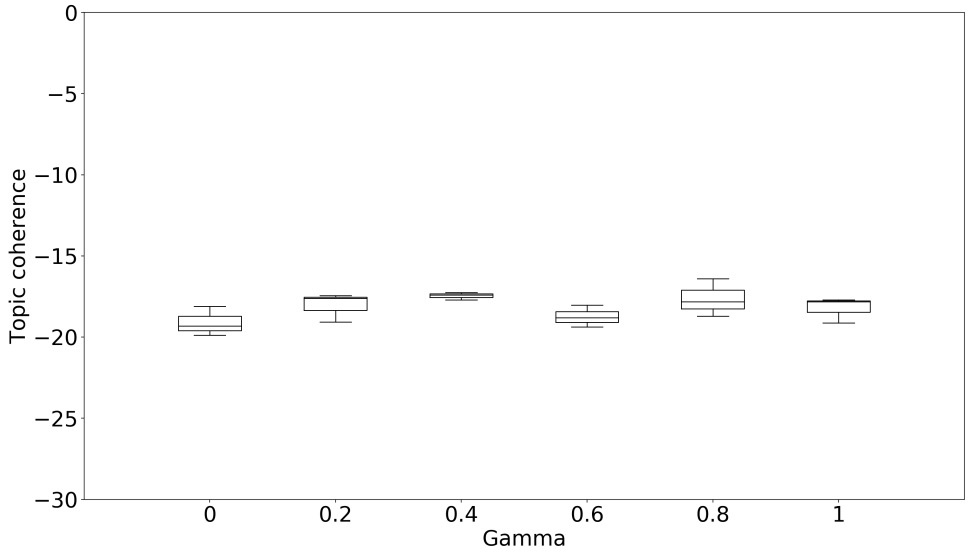


Figure 7.13: Influence of gamma on average coherence.

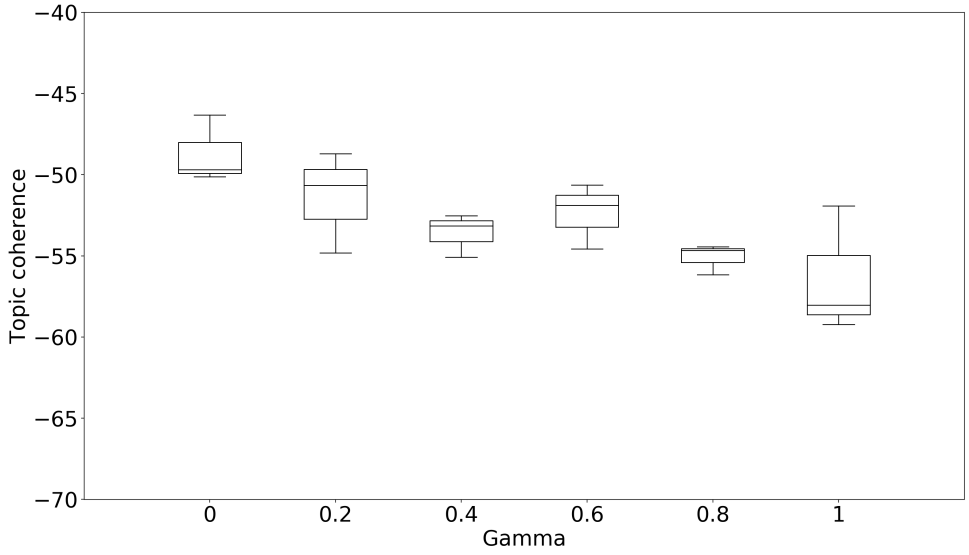


Figure 7.14: Influence of gamma on average coherence.