Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Computer Engineering
2020 | LIU-IDA/LITH-EX-A--20/014--SE

Evaluating Hierarchical LDA Topic Models for Article Categorization

Jennifer Lindgren

Supervisor : Amanda Olmin
Examiner : Fredrik Lindsten

External supervisor : Magnus Gasslander

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Upphovsrätt

Copyright

Abstract

With the vast amount of information available on the Internet today, helping users find relevant content has become a prioritized task in many software products that recommend news articles. One such product is Opera for Android, which has a news feed containing articles the user may be interested in. In order to easily determine what articles to recommend, they can be categorized by the topics they contain.

One approach of categorizing articles is using Machine Learning and Natural Language Processing (NLP). A commonly used model is Latent Dirichlet Allocation (LDA), which finds latent topics within large datasets of for example text articles.

An extension of LDA is hierarchical Latent Dirichlet Allocation (hLDA) which is an hierarchical variant of LDA. In hLDA, the latent topics found among a set of articles are structured hierarchically in a tree. Each node represents a topic, and the levels represent different levels of abstraction in the topics.

A further extension of hLDA is constrained hLDA, where a set of predefined, constrained topics are added to the tree. The constrained topics are extracted from the dataset by grouping highly correlated words. The idea of constrained hLDA is to improve the topic structure derived by a hLDA model by making the process semi-supervised.

The aim of this thesis is to create a hLDA and a constrained hLDA model from a dataset of articles provided by Opera. The models should then be evaluated using the novel metric word frequency similarity, which is a measure of the similarity between the words representing the parent and child topics in a hierarchical topic model.

The results show that word frequency similarity can be used to evaluate whether the topics in a parent-child topic pair are too similar, so that the child does not specify a subtopic of the parent. It can also be used to evaluate if the topics are too dissimilar, so that the topics seem unrelated and perhaps should not be connected in the hierarchy.

The results also show that the two topic models created had comparable word frequency similarity scores. None of the models seemed to significantly outperform the other with regard to the metric.

Acknowledgments

I would like to thank everyone at Opera who have helped me by answering questions and providing invaluable feedback on my work. Thank you to the stats team who all made me feel like a member of the team from day one. A special thank you to my supervisor at Opera, Magnus Gasslander, for always being curious and challenging me to always learn more, but also for telling me when it is ok to hold back in order to reach the goal on time.

I would also like to thank my supervisor at Linköping University, Amanda Olmin, for helping me understand complicated topic models and cheering me on via email, even when the circumstances prevented us from meeting in real life. A big thank you also goes to my excellent examiner Fredrik Lindsten who helped steer the work in the right direction and provided many helpful insights.

Contents

List of Figures

List of Tables

List of Abbreviations

CRP Chinese Restaurant Process.

DAG Directed Acyclic Graph.

DP Dirichlet Process.

DPM Dirichlet Process Mixture Model.

FP Frequent Pattern.

hLDA hierarchical Latent Dirichlet Allocation.

KL Kullback-Leibler.

LDA Latent Dirichlet Allocation.

LSA Latent Semantic Analysis.

MCMC Markov Chain Monte Carlo.

MH Metropolis-Hastings.

nCRP nested Chinese Restaurant Process.

NLP Natural Language Processing.

PAM Pachinko Allocation Model.

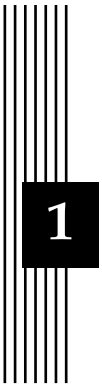
PLSA Probabilistic Latent Semantic Analysis.

POS Part of Speech.

RPC Rate of Perplexity Change.

RSS RDF Site Summary.

VB Variational Bayes.



1 Introduction

This chapter provides a motivation for the thesis by explaining why a hierarchical topic model for categorizing articles is useful. It also describes the background and aim of the thesis as well as what results can be expected. Finally, the delimitations of the thesis are stated.

1.1 Motivation

The internet contains an almost incomprehensible amount of information from a wide range of sources. Because of this, many services exist that help users navigate the internet and find relevant content. When recommending relevant content to users, several factors need to be considered. First of all, the content that is potentially interesting to the users needs to be categorized, so that it can be compared to the type of content that the users interact with. Second of all, it may be desirable to recommend content from a diverse range of categories that interest the users. Even if the users are very interested in technology for example, they may also be quite interested in cooking. Their recommendations should then probably mostly contain technology-related articles, but also some cooking articles. Finally, the categories of articles on the internet are constantly changing and new topics appear every day. Efficient and flexible modeling solutions are required to account for this.

When categorizing content on the internet, a simple approach is to analyze the source domain. For example, you would mostly expect articles related to cars when browsing the domain Automotive News. The problem with this approach is that the content on websites can be quite diverse and hard to predict. This results in overly general topics having to be assigned to such sites. With general categories, it becomes hard to pinpoint relevant articles, both because the articles the users are browsing cannot be categorized very well, and because the articles that could be recommended to them are not categorized very well. Categorizing content by source domain also typically requires continuous manual work.

A more effective solution is to categorize the articles based on the words they contain. This is frequently done within the domain of Machine Learning and Natural Language Processing (NLP), through topic modeling. A commonly used topic model for discrete data such as text is Latent Dirichlet Allocation (LDA), which is a generative probabilistic model proposed by David M Blei et al. [1]. When applying LDA in the context of topic modeling, articles are modeled as a mixture of topics, where the topics in turn are modeled as a mixture of words.

R. Yasotha and E.Y.A. Charles [2] used an LDA-based approach to automatically classify documents from 9,100 computer science related articles. Their results show that using LDA is an effective way of classifying documents covering a large number of categories.

A drawback of LDA is that the number of topics needs to be determined before creating the model. R. Yasotha and E.Y.A. Charles [2] used Kullback-Leibler (KL) divergence for this task. KL divergence is a measure used to compare probability distributions. Weizhong Zhao et al. have evaluated another approach which is using the Rate of Perplexity Change (RPC) as a function of the number of topics [3]. Perplexity is a measure of how well a model predicts a sample. No matter which approach is taken, determining the number of topics to model is often a time-consuming task.

Another drawback of LDA is that it does not model any hierarchical relationships within the topics. Knowing how topics relate to each other in a hierarchy can be useful when recommending articles to users. It gives more control over how specific the recommendations should be, and allows a wider range of interests to be covered. If users like technology and cooking, a mixture of those two more general topics can be recommended, while also pinpointing some specific technology-related articles that could be of interest.

A model that overcomes both of the above mentioned drawbacks is hierarchical Latent Dirichlet Allocation (hLDA), proposed by David M. Blei et al. [4]. It uses a method based on the nested Chinese Restaurant Process (nCRP) to create a nonparametric prior that is combined with a likelihood based on a hierarchical variant of LDA. This results in a tree structured model of topics, and instead of defining the number of topics to model, the depth of this tree is specified.

There are other hierarchical topic modeling solutions besides hLDA. Lin Liu et al. [5] compared hLDA and the alternative modeling solution Pachinko Allocation Model (PAM). One difference between the models is that in PAM, the hierarchy is represented as a Directed Acyclic Graph (DAG) instead of a tree. Another difference is that each node in hLDA is associated with a topic in the form of a distribution of words, while in PAM, the leaves of the DAG represent single words and each node represents the topic made up from the distribution of its child nodes. Therefore, topics in PAM are distributions of other topics.

Lin Liu et al. [5] conclude that hLDA is unable to capture the relations between parent and child node using probability parameters. Although, a distribution of words for each topic is obtained instead. PAM, having its structure represented as a DAG, is more flexible with regard to extensions but can not obtain a distribution of words for each topic.

In the context of suggesting articles to users, both PAM and hLDA are certainly viable options, as they provide topics with varying degrees of specificity. However, this thesis focuses on evaluating hLDA because of how topics are represented as distributions of words.

The hierarchical topic representation in hLDA has enabled an extension called constrained hLDA to be introduced. It was proposed by Wei Wang et al. [6] and improves hLDA by allowing predefined topics. The authors argue that their method improves the interpretability and the predictive abilities of hLDA.

This work proposes a novel evaluation metric for hierarchical topic models like hLDA and constrained hLDA, where the topics are represented as distributions of words. The metric is called word frequency similarity, and is a measure of how similar the word frequencies in the parent and child topics in the hierarchy are. Word frequency similarity is used in this thesis to evaluate and compare a hLDA and a constrained hLDA topic model.

1.2 Background

This thesis work was performed at Opera in Linköping. Opera's products reach more than 360 million users every month. While the company has expanded to other markets, their main products are desktop and mobile web browsers. The main focus of Opera in Linköping is one of the mobile browsers, Opera for Android.

A central part of Opera for Android is a news feed containing RDF Site Summary (RSS) articles from a diverse range of sources on the internet. The thesis assignment is to develop a categorization system that can be used to separate these articles into different topics.

The system should be able to create a hierarchy of topics so that both general and more specific topics can be identified. It is to replace the current categorization system based on content origin, and is to be part of a larger system. However, the details of that larger system are outside the scope of the assignment.

1.3 Aim

The aim of this thesis is to create a hLDA model and a constrained hLDA model from a data collection of text from a wide range of RSS articles. Both models are then to be evaluated using the novel evaluation metric word frequency similarity.

Finally, conclusions are drawn from the results of this study, arguing whether constrained hLDA outperforms hLDA with regard to this metric.

In addition to this, the aim is also that the result can be used by Opera to categorize articles.

1.4 Research questions

The following research questions will be addressed and answered in this thesis work:

1. How can a topic hierarchy, where the topics are represented as distributions of words, be evaluated based on the word frequency similarity between parent and child topics?

2. How does a topic hierarchy created using hLDA compare to one created using constrained hLDA with regard to the word frequency similarity between parent and child topics?

1.5 Delimitations

The purpose of this thesis is mainly to evaluate two topic models, that could potentially be utilized by Opera to categorize articles. Any aspects of the systems in which the models are later used by Opera are outside the scope of the assignment.

2 Theory

This chapter covers the theoretical background and related work for the thesis. An introduction to topic modeling is given in the first section, followed by description of the multinomial and Dirichlet distributions that are central in the models used in the thesis. Then, the LDA modeling process is described along with the Dirichlet process. The details of the two hierarchical LDA models that are evaluated in this work, hLDA and constrained hLDA, are also included in this chapter.

2.1 Topic Modeling

A common problem when working with documents and text data is determining which topics they concern. For this purpose, there are two commonly used groups of unsupervised learning methods, where unsupervised means that they require no labeled training data [7, 8].

The first one is *clustering*, which partitions the text documents into clusters that represent different topics. In this setting, it is assumed that a document belongs to only one cluster and thus only contains one topic. This is sometimes called hard clustering.

The second group of methods is called *topic modeling*, and although closely related to clustering, there are a few key differences. In topic modeling, a generative probabilistic model[1] is used to describe the probabilities of the documents belonging to each of the clusters, instead of assigning each document only to a single cluster. Because of this, a document is considered to contain a mixture of different topics instead of just one. Each topic in the model is in turn represented by a distribution of words, where the words that best describe the topic have higher probabilities. This is sometimes called soft clustering.

A benefit of soft clustering is that ambiguous words can have high probabilities in several topics. For example, the word jaguar can have a high probability in an animal topic where it refers to jaguar the cat, as well as in a car topic where it refers to Jaguar the car brand.

In terms of dimensionality, topic modeling has an advantage over clustering as each document can be described as a linear combination of different topics, and not just as one

[1] A generative probabilistic model is a model created through what can be thought of as reverse engineering. It produces a model that has the highest possible chance of producing the data it has been given. Since it is probabilistic, it produces different results each run.

topic within the dimension. This makes topic modeling a good choice for both clustering and dimension reduction problems.

One topic modeling method is Probabilistic Latent Semantic Analysis (PLSA), proposed by Hofmann [9]. It is based on Latent Semantic Analysis (LSA), proposed by Deerwester [10], which takes the vector space representation of the document corpus with regard to word frequencies, and performs dimension reduction using linear projection. The idea behind this is that documents that share frequent words will have a similar representation in the reduced space. Thus, it is able to detect synonyms and words belonging to the same topic.

PLSA was suggested to provide a more solid statistical foundation which LSA lacks. PLSA is based on the likelihood principle, meaning that standard statistical techniques can be used for model fitting and complexity control. The topics in PLSA are multinomial random variables from which words can be sampled.

One weakness of PLSA is that the generative probability model does not include the topic distributions. Instead, the topic distribution of each document in the corpus is used as a parameter for the model. This means that the number of parameters grows linearly as more documents are added to the corpus, with overfitting issues as a result. It is also unclear how probabilities should be assigned to previously unseen documents.

A method introduced to tackle the drawbacks of PLSA is Latent Dirichlet Allocation (LDA), which was proposed by Blei et al. [1]. LDA extends PLSA to include a generative probabilistic model over the topic distributions as well. In a k-topic LDA model, instead of letting each topic distribution be a parameter, one k-dimensional hidden random variable is used to sample the topic distributions from a Dirichlet distribution. This prevents the complexity from escalating as the size of the document corpus grows. LDA is described in further detail in Section 2.4.

2.2 Multinomial Distribution

To model the probabilities of words in documents, a commonly used representation is the multinomial distribution, which captures the relative frequency of words in a document [7]. A common scenario used to describe the multinomial distribution is that of throwing a dice n times. Every time the dice is thrown, there is a $1/6$ chance of obtaining each dice value. The probabilities can be represented in a 1 x 6 vector, where the sum of the probability values is 1. The outcome of throwing the dice n times can also be represented in a 1 x 6 vector, where each element represents how many times a given value appeared. So in summary, a k-dimensional multinomial distribution is parameterized by a k-dimensional vector, whose elements sum to one, and an integer n. It models the probability of a k-dimensional occurrence vector, whose elements (occurrences) sum to n. The notation $X \sim \text{Mult}(\theta,\ n)$ denotes that x_i is the count for outcome i when n trials (e.g. dice rolls) are performed.

A probability distribution over k variables $\theta = \{\theta_1,\ \theta_2,\ \ldots\ ,\ \theta_k\}$ where $\sum_{i=0}^{k} \theta_i = 1$ and $\theta_i \geqslant 0$, is called a (k - 1)-simplex. Figure 2.1 shows the plane containing all possible values of the 2-simplex, i.e. all possible probability distributions over three variables. Each point in the plane can be used as a parameter of a three-dimensional multinomial distribution, since there are three variables, and the sum of the probabilities is always one.

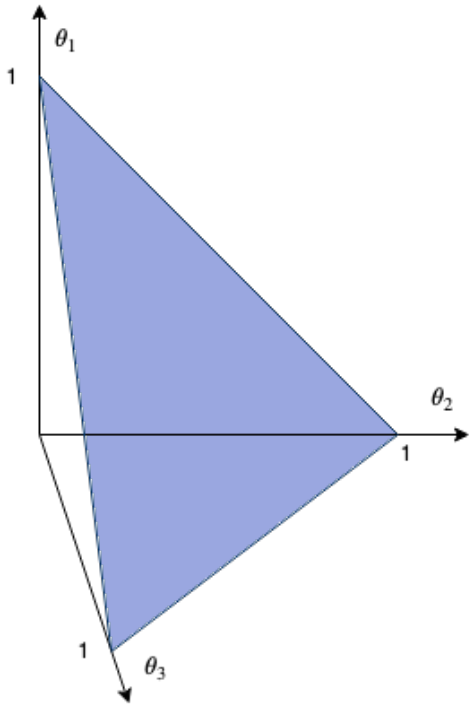


Figure 2.1: The plane containing all values of the 2-simplex.

2.3 Dirichlet Distribution

In Bayesian statistics, the Dirichlet distribution is a *conjugate prior* of the multinomial distribution. This means that modeling multinomially distributed data with a Dirichlet prior probability distribution[2] results in a posterior probability[3] that is also a Dirichlet distribution. This posterior can in turn be used as a prior when repeating the process. Because of this convenient fact, the Dirichlet distribution is often used in combination with multinomially distributed data.

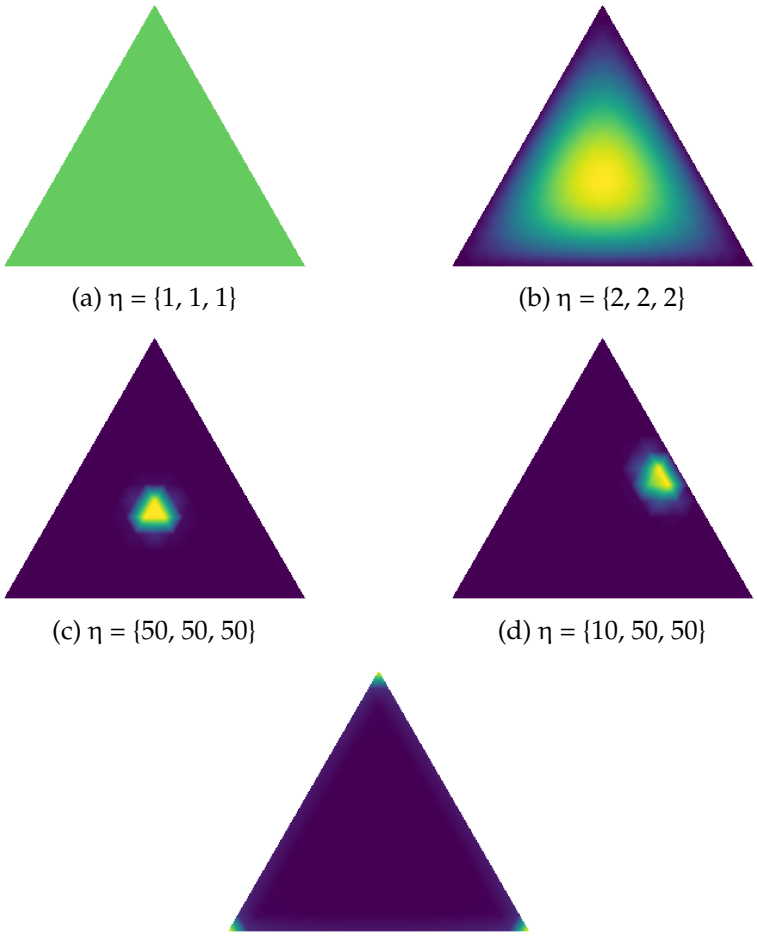
The notation $\theta \sim \text{Dir}(\eta)$, where $\theta = \{\theta_1, \theta_2, ..., \theta_k\}$, denotes that θ is a probability distribution over k variables. The Dirichlet distribution can thus be used to sample multinomial distributions from a k-dimensional space. It can be thought of as a distribution of distributions. The parameter that determines the probabilities of the distributions is $\eta = \{\eta_1, \eta_2, ..., \eta_k\}$, where $0 < \eta_i < \infty$ for each i.

Figure 2.3 shows five Dirichlet probability distributions resulting from different values of η. (a) shows the distribution for η = {1, 1, 1}, where all distributions have equal probability. (b) and (c) show how increasing the values in η above one affects the distribution. Higher values cause higher peaks as well as more concentrated distributions. Note that in (a), (b) and (c), all values in η are equal, which results in symmetrical distributions. When this is the case, η can be given as a scalar, and is called a *concentration parameter*.

Figure 2.3 (d) shows an asymmetrical distribution where the value of one component is slightly lower than the other two, causing a distribution that is skewed toward the two stronger components. (e) shows what happens when the η values are decreased below one. Peaks in the distribution form at each component, favouring distributions where one of the components dominate. The smaller the values, the higher and more concentrated the peaks.

[2] A prior probability distribution, or prior for short, is used to account for prior knowledge or beliefs about a distribution.

[3] The posterior probability distribution is the distribution resulting from taking relevant knowledge about the data, including any priors, into account.



(a) $\eta = \{1, 1, 1\}$ (b) $\eta = \{2, 2, 2\}$

(c) $\eta = \{50, 50, 50\}$ (d) $\eta = \{10, 50, 50\}$

(e) $\eta = \{0.5, 0.5, 0.5\}$

Figure 2.2: Dirichlet probability distributions for different values of η.

2.4 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a generative probabilistic topic model used to model discrete data collections, described by Blei et al. [1]. The goal is to find short descriptions of the members of a collection that enable efficient processing of large collections, while preserving the essential statistical relationships. In their work, Blei et al. [1] used LDA to reduce the feature space of a dataset of 8000 documents and 15,818 words by 99.6 percent, with little reduction in classification performance. This shows that LDA is an effective method for representing text data in a reduced dimension. LDA is a popular model when the discrete data is in text form and the topics occurring in the text should be identified.

Model Description

As previously mentioned, LDA is a generative model. This means that it generates documents based on statistical relationships present in the training data. To make the model description intuitive, the following three central terms are used:

- *Word:* The smallest unit of discrete data. The set of all unique words form a *vocabulary*.
- *Document:* A collection of words.
- *Corpus:* A collection of documents.

LDA is a model of a corpus, where the documents in the corpus are represented as random mixtures of latent *topics*. The topics are in turn distributions of words.

The steps of LDA can be described as:

1. Determine the prior parameters α and η based on previous knowledge of the topic and word distributions.
2. For each topic $k \in \{1, \ldots, K\}$:
   - Draw a word distribution vector $\beta_k \sim \text{Dir}(\eta)$.
3. For each document $m \in \{1, \ldots, M\}$:
   - Draw a topic distribution vector $\theta_m \sim \text{Dir}(\alpha)$.
   - For each word $n \in \{1, \ldots, N\}$:
     - Draw a topic $z_{m,n} \sim \text{Multinomial}(\theta_m)$, $z_{m,n} \in \{1, \ldots, K\}$
     - Draw a word $w_{m,n} \sim \text{Multinomial}(\beta_{z_{m,n}})$, $w_{m,n} \in \{1, \ldots, N\}$

LDA generates a set of topics, i.e. distributions over words from the corpus vocabulary. The words in the distributions are called keywords. The top keywords in a topic could for example be *{pet, cat, dog, feed, cute}*, where one might conclude that the topic is about pets.

Figure 2.3 shows a graphical representation of the generative process. The letters at the bottom right of each plate represent the number of times the variables are drawn.

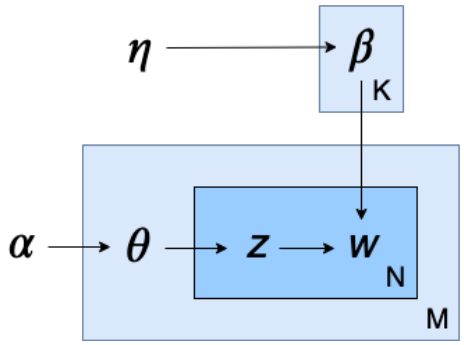


Figure 2.3: Plate representation of LDA.

Posterior Inference in LDA

Some of the variables in the LDA model are latent, meaning that they are not explicitly observed. The latent variables are the topic distributions in the documents (θ), the topic assignments (z) and the word distributions in the topics (β). It is sometimes desirable to know these variables, as they can be used for prediction and data exploration [11]. Conveniently, they can be inferred from the documents generated by the model, by computing the posterior distribution using Bayes' theorem:

$$p(\theta, z, \beta | w, \alpha, \eta) = \frac{p(\theta, z, \beta, w | \alpha, \eta)}{p(w | \alpha, \eta)} \tag{2.1}$$

This is called posterior inference. Unfortunately, the posterior distribution in LDA is intractable to compute, as normalizing the distribution requires marginalization over the hidden variables in the denominator $p(w \mid \alpha, \eta)$. This computation results in a summation over latent topics where two unknown variables are coupled, making it hard to determine their individual values [7, 12].

Despite the difficulties of computing the exact value of the posterior, several approximation methods exist. There are two main categories of approaches; optimization approaches and sampling approaches [11].

A popular optimization approach is variational inference, or Variational Bayes (VB) as it is called for hierarchical Bayesian models such as LDA. Blei et al. [1] describe a convexity-based variational algorithm for inference in LDA. The idea is to find a family of lower bounds for the posterior distribution, indexed by variational parameters.

Traditional variational inference algorithms, so called batch algorithms, alternate between updating the variational parameters and analyzing observations. In these algorithms, the corpus is re-iterated each time a new batch is processed. Hoffman et al. [11] proposed an alternative algorithm, online variational inference for LDA, where online means that each observation is only processed once, making the algorithm faster and more suitable for large corpora.

A popular sampling approach is using Markov Chain Monte Carlo (MCMC) to generate samples from the posterior. With this method, the posterior can be approximated arbitrarily well if enough iterations are made [13]. A drawback is that it is hard to know how many iterations it takes to get a good enough approximation [14]. An advantage of MCMC over variational inference is that it is nonparametric.

Gibbs sampling is the simplest form of the MCMC algorithm [7]. It is used by both Blei et al. [4] and Wang et al. [6] when approximating the posterior distributions of the hLDA and constrained hLDA models. It is also the algorithm used in this thesis, and the details are explained further in Section 2.8.

Selecting the Number of Topics in LDA

The number of topics to model in LDA needs to be defined before creating the model. Unfortunately, there is no single method known to be optimal for determining the most suitable number. Instead, authors often combine several metrics that indicate that a certain number of topics is representative of the corpus. Geva et al. [15] used LDA to model the topics appearing in Twitter users' self-produced tweets and retweets. In their work, the authors ran LDA using different values for the number of topics and utilized four different methods to identify the best choice:

- **Method 1:** Based on work by Griffiths and Steyvers [16], the log-likelihood of the model is approximated. The number of topics yielding the highest scores is considered the best.
- **Method 2:** In this method, the average cosine distance among the topics is minimized to find the most suitable number of topics. Based on work by Cao et al. [17].
- **Method 3:** The third method is based on symmetric KL divergence and work by Arun et al. [18]. The divergence should be minimized to find the optimal number of topics.
- **Method 4:** Finally, the fourth method focuses on maximizing the information divergence using Jensen-Shannon divergence. Thus, the goal is to find topics that differ from each other as much as possible. This method is based on work by Deveaud et al. [19].

The scores from the four methods were normalized and the two scores that were to be maximized were converted to (1 - score) [15]. This allowed for all four scores to be added together and the number of topics that yielded the lowest total score could be selected as the number of topics for the model.

2.5 Nonparametric Bayesian Models

LDA is a parametric model, meaning that the number of parameters is fixed. For example, the vector representing the Dirichlet prior probability for the topic distribution in a document with k topics is k-dimensional.

Another category of models is nonparametric Bayesian models, where nonparametric means that the number of parameters is not determined without considering the data [8]. The priors used in this case do not have a fixed number of parameters, but are instead stochastic processes. A stochastic process is a collection of random variables, where the number of variables is in theory allowed to be infinite.

In topic modeling, a powerful stochastic process is the Dirichlet process. If used as a prior, an infinite number of topics can be modeled before the most suitable finite number is determined. This is used in both the hLDA model and the constrained hLDA model to allow for a non-fixed number of topics.

The Dirichlet process is similar to another stochastic process called the Chinese Restaurant Process (CRP), which is described in this section. With the CRP as reference, a simple extension can be added to it in order to explain the Dirichlet process.

In addition to these two stochastic processes, this section also explains how stick-breaking constructions can be used to represent the probability distribution of an infinite-dimensional mixture model.

The Chinese Restaurant Process

The Chinese Restaurant Process (CRP) is a stochastic process described by a scenario taking place in a Chinese restaurant. The process was described by Aldous [20], but the analogy was attributed to Jim Pitman and Lester Dubins. They found that every time they went to a Chinese restaurant, no matter how many customers there were, there was always room for more.

The process distributes data, being the restaurant customers in the analogy, over partitions, being the tables. It can be thought of as a clustering method where each table is a cluster and the guests are the data assigned to them.

Imagine that M customers enter a restaurant with an infinite number of tables. The first customer sits down at the first table with probability one [4]. The mth subsequent customer sits down at a table, either previously occupied or unoccupied, with the following probabilities:

$$\begin{aligned} p(\textit{occupied table } i \mid \textit{previous customers}) &= \frac{m_i}{\gamma + m - 1} \\ p(\textit{unoccupied table} \mid \textit{previous customers}) &= \frac{\gamma}{\gamma + m - 1} \end{aligned} \tag{2.2}$$

where m_i is the number of customers already seated at table i, and γ is a parameter that controls how often a new table is chosen over a previously occupied one.

The Chinese restaurant process models a nonparametric scenario where the number of clusters is determined by the data. Note that the maximum number of clusters is equal to the total number of customers, M.

The Dirichlet Process

The CRP in its simplest form describes only a distribution of customers over tables. However, just like the Dirichlet distribution, the Dirichlet Process (DP) is a distribution of distributions. This means that the result of drawing from a DP is a distribution of values, and not just a single value. To account for this, we can imagine that each table is assigned a dish in the form of a parameter vector θ_i. Each customer that sits at table i is associated with this parameter

vector. Thereby the Chinese restaurant analogy has been extended to represent a distribution of distributions, and can be used to describe a Dirichlet process.

A more formal definition of the Dirichlet process will now be presented. The parameters are a base distribution H and a concentration parameter γ [7]. A stochastic process G is a Dirichlet process, written as $G \sim \mathrm{DP}(\gamma,\ H)$, if for an arbitrary finite measurable partition $A_1,\ A_2,\ \ldots,\ A_r$ of the probability space of H, the following holds:

$$(G(A_1),\ G(A_2),\ \ldots,\ G(A_r)) \sim \mathrm{Dir}(\gamma H(A_1),\ \gamma H(A_2),\ \ldots,\ \gamma H(A_r))$$

where $G(A_i)$ and $H(A_i)$ are the marginal probabilities of G and H over partition A_i. The marginal distribution of G is thereby Dirichlet distributed, which is how the process got its name.

The base distribution H is the mean distribution of the the DP, and the concentration parameter γ works as an inverse variance parameter. A large γ results in a small variance, and draws are concentrated around the mean distribution H. The value of γ also controls how many samples are in the resulting distribution, if γ is small, few values with high probabilities are included. As $\gamma \to \infty$, the distribution becomes close to continuous. In other words, the base distribution H can be continuous, but draws from G will still be discrete. This means that a Dirichlet process can be used to make a discrete copy of a continuous distribution. Also note that if many distributions are drawn from the DP, duplicates can appear.

Generating distributions θ_m from G is a random process that can be described as:

$$\theta_m \mid \theta_{m-1},\ \ldots,\ \theta_1 = \begin{cases} \theta_i^*, & \text{with probability } \frac{m_i}{\gamma+m-1} \\ \text{New draw from H}, & \text{with probability } \frac{\gamma}{\gamma+m-1} \end{cases} \tag{2.3}$$

where θ_i^* represents the ith *unique* distribution drawn from H. In other words, each time a distribution θ_n is drawn from G, it is either a duplicate of a previously drawn distribution, or it is a new one. Each unique distribution θ_i^* drawn can be used to indicate the distribution of cluster i. m_i is the number of times θ_i^* has been drawn previously. It is clear that the Dirichlet process can be described nicely by the CRP.

A mixture model that uses a Dirichlet process as a prior is called a Dirichlet Process Mixture Model (DPM). Since the Dirichlet process is non-parametric, so is the DPM. A popular DPM is hLDA [7], which is described in detail in Section 2.8.

Stick-breaking Constructions

The proportions of the clusters generated in a Dirichlet process can be defined using something called a stick-breaking construction [7, 8]. It can be described by imagining a stick of length 1, and breaking off a piece from it. The size of the piece broken off is proportional to a variable $V_k \in (0,\ 1)$. The remainder of the stick is then broken off in the same way. This can be done an infinite number of times, generating infinitely small pieces of the stick. The stick-breaking construction is illustrated in Figure 2.4.

Each time a piece is broken off from the stick, the length of it is assigned to a variable π_k. The length of the kth piece is given by:

$$\pi_k \ = \ V_k \prod_{p=1}^{k-1} (1 - V_p) \tag{2.4}$$

When V_k is drawn from a Beta distribution so that $V_k \sim \mathrm{Beta}(1,\ \alpha)$, the one-parameter stochastic process known as the GEM distribution is obtained. However, this is only a special case of a more general two-parameter definition where $V_k \sim \mathrm{Beta}(m\rho,\ (1-m)\rho)$ for $\rho > 0$ and $m \in (0,\ 1)$, with the special case being $m\rho = 1$ and $\alpha = (1-m)\rho$. Using the two-parameter definition allows control over both mean and variance of the distribution, and is the variant that will be used in this thesis.

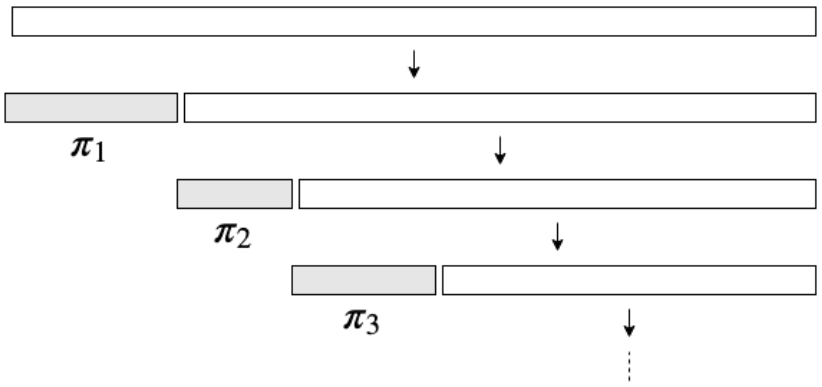


Figure 2.4: Pieces of a stick of length 1 being broken off in a stick-breaking construction.

The resulting vector $\pi \sim \text{GEM}(m, \rho)$ can be used to define the proportions of components in a mixture model. For example, it can be used to define a Dirichlet Process as a mixture model over the distributions θ_i^* drawn from H using Equation 2.3:

$$G \sim \sum_i \pi_i \delta_{\theta_i^*}$$

where $\delta_{\theta_i^*}$ denotes a point mass at θ_i^*. The parameter vector π thus describes the distribution of distributions θ^*.

2.6 Hierarchical Topic Models and Hierarchical Clustering

Topic models that capture abstraction levels of topics are called *hierarchical topic models* [8]. They are structured as trees with one root node, where each node represents a topic.

Each topic in the tree is not a summary of the topics below it, as in the similar *hierarchical clustering models*, but rather contain words that are shared between all topics below. Thus, words that occur frequently across all child topics are represented in the parent topics, while words that are more specific for each topic occurs further down in the tree.

Both models used in this work, hLDA and constrained hLDA, are hierarchical topic models that model abstraction levels.

2.7 The Nested Chinese Restaurant Process

An extended variant of the Chinese Restaurant Process (CRP) called the nested Chinese Restaurant Process (nCRP) is described by Blei et al. [4, 8]. The idea is to create a distribution that can be used as a prior in a hierarchical topic model, where the topics are structured in a tree.

The nCRP tree has one root node and, in theory, infinite depth and width in the levels below. Each node in the tree has the same structure as a CRP, containing an infinite number of topics. Every document in the corpus is assigned to a path in the tree, from the root node to one of the leaves. The paths that have documents assigned to them form a random subtree of the infinite tree. The branching factor of each node is at most equal to the number of documents in the corpus.

An example nCRP tree with three levels is illustrated in Figure 2.5. β_k represents the topic distribution of the underlying CRP in the kth node.

In the Chinese restaurant analogy, the following scenario can be imagined. A city has an infinite amount of restaurants, where each restaurant has an infinite amount of tables as in the original CRP. A tourist is visiting the city, and on the first night they eat at the first

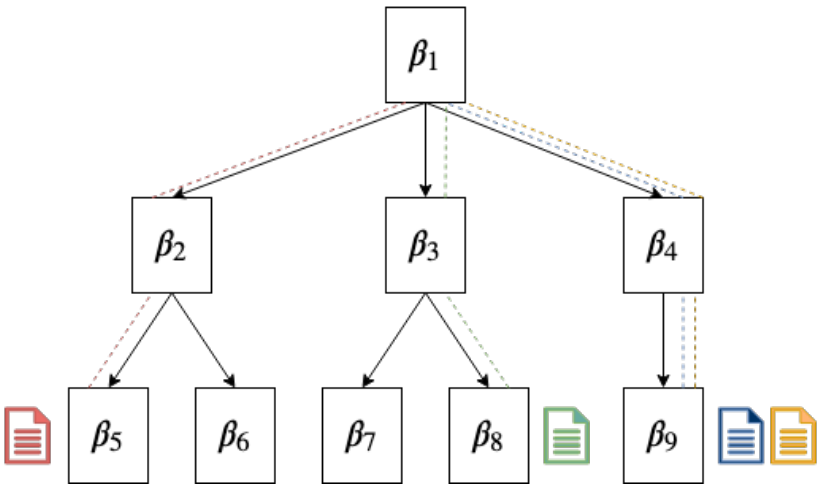


Figure 2.5: Example nCRP tree with three levels. The paths of four documents are indicated. β_k represents the topic distribution of the underlying CRP in the kth node.

restaurant. This is the root node in the tree and all tourists eat here on their first night. They select a table according to the CRP, and the table is associated with a parameter vector like in the original definition. What is new in the nested version is that each table points to a table in another restaurant in the level below, and the tourist will eat there in the next evening.

In theory, the tourist could eat at an infinite amount of restaurants, but in practice it is convenient to decide on a maximum depth of the tree. In this thesis, the depth of the tree is considered fixed and denoted L.

The nCRP defines a distribution over the paths of the documents in the tree. Each path, denoted c_d for $d \in \{1, 2, ..., M\}$, is represented by a L-dimensional vector, containing the indices of the nodes in the path. The path of the red document (leftmost) in Figure 2.5 is for example represented by the vector $\{1, 2, 5\}$. A specific node in a path is denoted c_{dl}, where d refers to the path/document number and l refers to the level in the tree. A node that is part of a path is said to be visited, and a node that no document has passed through is unvisited. Only visited nodes will be part of the final tree.

A new path c_d for a document is generated by the following steps [21]:

1. Let the first node in the path, c_{d1}, be the root node.
2. For each of the levels $\ell \in \{2, 3, ..., L\}$ below, either let $c_{d\ell}$ be a previously visited child node of the parent $c_{d(\ell-1)}$, or an unvisited node. The probability of choosing a visited or unvisited node depends on the number of times the nodes have been visited previously as well as on the hyperparameter γ_ℓ:

$$\begin{cases} p(\text{visited node } c_{child} \mid \text{visits at other nodes}, \gamma_\ell) = \dfrac{m_{c_{child}}}{\gamma_\ell + m_{c_{d(\ell-1)}}} \\ p(\text{unvisited node } c_{new} \mid \text{visits at other nodes}, \gamma_\ell) = \dfrac{\gamma_\ell}{\gamma_\ell + m_{c_{d(\ell-1)}}} \end{cases}$$

   where $m_{c_{child}}$ and $m_{c_{d(\ell-1)}}$ are the number of visits at the child and parent nodes, respectively. The parameter $\gamma = \{\gamma_1, \gamma_2, ..., \gamma_L\}$ dictates how likely it is to pick unvisited nodes and thus extend the tree.

Note that if a previously unvisited node is chosen in step 2, it will have no visited child nodes below it. Therefore the path will visit new nodes all the way down to level L.

2.8 Hierarchical Latent Dirichlet Allocation (hLDA)

Using the CRP as a prior provides a means of modeling an infinite number of topics in a corpus. However, it does not indicate how the topics relate to each other hierarchically. Neither does it capture any levels of abstraction in the topics.

Hierarchical Latent Dirichlet Allocation (hLDA) was described by Blei et al. [4, 8] and is a hierarchical topic model that uses the nested Chinese Restaurant Process (nCRP) as a prior. The hLDA model allows representation of both abstraction levels and relationships between topics.

Model Description

Blei et al. [8] augment the nCRP in two ways in their hLDA definition. First, each node in the tree is assigned a topic, so that each path of length L in the tree consists of L topics. Second, a probability distribution over the topics in the path is defined using the stick-breaking (GEM) distribution.

The model is generative, and thus describes how a corpus of documents was generated. Each document is assumed to have been generated by drawing words from the topics in the tree. The process of generating a document starts by choosing a path in the tree, as well as a distribution over the levels [8]. Then for each word to generate, a level is selected using the level distribution, and a word is drawn from that topic.

The generative process is more formally described as:

1. For each node in the infinite tree, draw a topic $\beta_k \sim \text{Dir}(\eta)$.
2. For each document $d \in \{1,\ 2,\ ...,\ M\}$:
   (a) Draw a path $c_d \sim \text{nCRP}(\gamma)$.
   (b) Draw a distribution over levels $\theta_d \sim \text{GEM}(m,\ \rho)$.
   (c) For each word $n \in \{1,\ 2,\ ...,\ N_d\}$ in d:
      (i) Choose a level $z_{d,n} \sim \text{Discrete}(\theta_d)$.
      (ii) Choose a word $w_{d,n} \sim \text{Discrete}(\beta_{c_d,z_{d,n}})$ where $\beta_{c_d,z_{d,n}}$ is the topic distribution for the node at the chosen level $z_{d,n}$.

The levels and words are drawn from one-parameter discrete distributions, denoted "$Z \sim \text{Discrete}(\alpha)$", meaning $Z = i$ with probability α_i. A graphical representation of the model is shown in Figure 2.6 [8].

It should be noted that the stick-breaking (GEM) distribution, from which the distributions over levels in each path is drawn, is likely to assign relatively high probabilities to nodes closer to the root node compared to nodes further down. This is because the pieces of the stick that are broken off early in the stick-breaking construction tend to be larger than the pieces that are broken off from the smaller remainders of the stick later on. This means that words from the more general topics will generally be drawn more often than words from more topic-specific words.

Further, the first step in the generative process involves drawing an infinite amount of topics β_k, one for each node in an infinite tree. However, only the topics that eventually get visited will be used. Therefore, topics can be generated in a "lazy" manner instead, where a topic is only drawn the first time a node is visited.

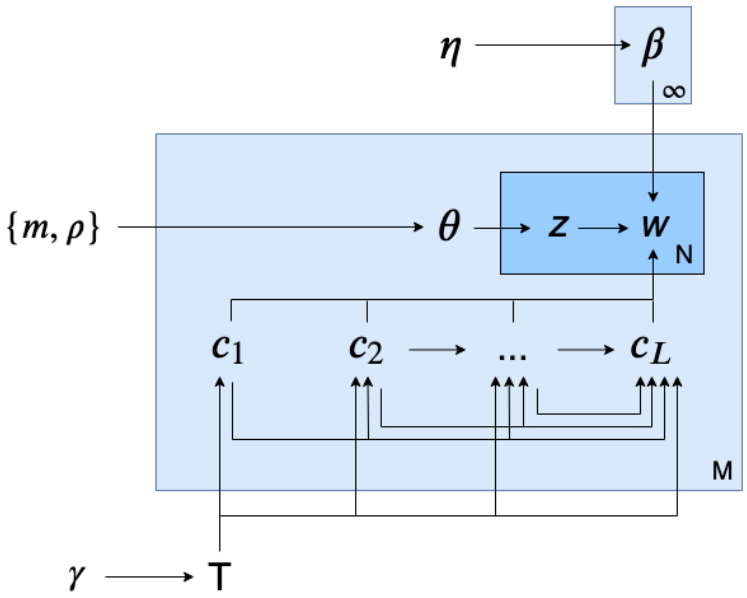


Figure 2.6: Graphical representation of the hLDA model. T represents the infinite number of possible paths in the infinite tree from which paths are chosen.

Posterior Inference in hLDA

Just like in the LDA model, there are latent variables in the hLDA model that need to be inferred from an approximation of the posterior distribution. This process can be thought of as inverting the hLDA generative process previously described [8]. Instead of generating documents using a model, we want to approximate a model that may have generated the observed documents.

As previously mentioned, there are two main approaches for posterior inference, optimization and sampling. This thesis uses a sampling approach called collapsed Gibbs sampling.

Collapsed Gibbs Sampling

Collapsed Gibbs sampling is the sampling method used for posterior inference by both Blei et al. [8] for hLDA and Wei et al. [6] for constrained hLDA. It was first described by Liu [22] and later applied to topic models by Griffiths and Steyvers [23].

The Gibbs sampler is a Markov Chain Monte Carlo (MCMC) algorithm, where there is an underlying Markov Chain of states iteratively generated according to an algorithm [13]. Each iteration samples the latent variables conditioned on all other latent variables as well as the observed data.

The target distribution of the chain is the conditional distribution of the latent variables given the observed data [8]. Therefore, an approximation of the posterior distribution can be made by collecting samples from the distribution of the chain when it has approached its target distribution.

In a *collapsed* Gibbs sampler, some of the latent variables are marginalized out to speed up the convergence of the Markov chain. In the case of hLDA, the topic parameters β and word distributions θ are marginalized out. The posterior we want to approximate is thus $p(c_{1:D},\ z_{1:D} \mid \gamma,\ \eta,\ m,\ \pi,\ w_{1:D})$, i.e. the latent variables are the distribution of the paths allocated for each document, and the level allocations for each word in each document.

Given the current state of the Gibbs sampler $\{c_{1:D},\ z_{1:D}\}$, each latent variable is iteratively sampled conditioned on the other variables. The sampling algorithm can be described with the following steps [8]:

1. For each document $d \in \{1,\ 2,\ ...,\ D\}$:
   (a) Sample a new path c_d.
   (b) For each word $n \in \{1,\ 2,\ ...,\ N\}$ in the document:
      - Sample a new level allocation $z_{d,n}$.

It should be noted that because the sampling of each variable is conditioned on all the other variables, the distributions from which paths and level assignments are chosen from will vary in each iteration. This means that the run time of the algorithm will also vary.

The following sections describe the steps of the algorithm in further detail. They will also discuss how priors can be introduced to the hyperparameters to include them in the inference. Finally, a method of assessing the convergence of the sampler is described.

Sampling the Document Paths

Recall that the depth of the infinite tree is fixed and denoted L. The first step in the collapsed Gibbs sampling algorithm (1a) is drawing a path for a document, conditioned on all other paths, the observed words and their level allocations. All possible paths of length L are considered. This includes all paths with documents currently assigned to them (paths already in the tree), as well as paths that include unvisited nodes, i.e. new paths. The probability of choosing a path c_d for a document d is given by [8]:

$$p(\mathbf{c}_d \mid \mathbf{w},\ \mathbf{c}_{-d},\ \mathbf{z},\ \eta,\ \gamma) \propto p(\mathbf{c}_d \mid \mathbf{c}_{-d},\ \gamma)p(\mathbf{w}_d \mid \mathbf{c},\ \mathbf{w}_{-d},\ \mathbf{z},\ \eta) \tag{2.5}$$

which is an instance of Bayes's theorem where $p(\mathbf{c}_d \mid \mathbf{c}_{-d},\ \gamma)$ is the prior probability on paths given by the nCRP (see Section 2.7), and $p(\mathbf{w}_d \mid \mathbf{c},\ \mathbf{w}_{-d},\ \mathbf{z},\ \eta)$ is the probability of the data given that choice of path. The notation $\mathbf{c}_{-d}$ denotes the vector $\mathbf{c}$ but leaving out the element d.

By integrating over the multinomial parameters, a ratio of normalizing constants for the Dirichlet distribution is obtained:

$$\begin{aligned} p(\mathbf{w}_d \mid \mathbf{c},\ \mathbf{w}_{-d},\ \mathbf{z},\ \eta) \ = \prod_{\ell=1}^{L} & \frac{\Gamma(\sum_w \#[\mathbf{z}_{-d} = \ell,\ \mathbf{c}_{-d,\ell} = c_{d,\ell},\ \mathbf{w}_{-d} = w] + V\eta)}{\prod_w \Gamma(\#[\mathbf{z}_{-d} = \ell,\ \mathbf{c}_{-d,\ell} = c_{d,\ell},\ \mathbf{w}_{-d} = w] + \eta)} \\ & \frac{\prod_w \Gamma(\#[\mathbf{z} = \ell,\ \mathbf{c}_\ell = c_{d,\ell},\ \mathbf{w} = w] + \eta)}{\Gamma(\sum_w \#[\mathbf{z} = \ell,\ \mathbf{c}_\ell = c_{d,\ell},\ \mathbf{w} = w] + V\eta)} \end{aligned} \tag{2.6}$$

where the notation #[condition 1, condition 2, ...] refers to the number of times the given conditions are fulfilled. Γ denotes the Gamma function, a generalization of the factorial function for non-integer values [24].

Sampling the Level Distributions

In step 1b of the sampling algorithm, each word in each document is assigned a level in the tree. The nth word in document d is assigned a level $z_{d,n}$ according to:

$$p(z_{d,n} \mid \mathbf{z}_{\text{-(d,n)}},\ \mathbf{c},\ \mathbf{w},\ m,\ \pi,\ \eta) \propto p(z_{d,n} \mid \mathbf{z}_{-d,n},\ m,\ \pi)p(w_{d,n} \mid \mathbf{z},\ \mathbf{c},\ \mathbf{w}_{-(d,n)},\ \eta) \tag{2.7}$$

where the first term, $p(z_{d,n} \mid \mathbf{z}_{-d,n},\ m,\ \pi)$, is a stick-breaking distribution (see Section 2.5) over the levels in the tree. This level distribution is, in theory, infinite-dimensional. Therefore, the components are defined one by one.

First, only the levels currently represented by the other words in the document are considered. Let the deepest level that currently has a word from document d assigned to it be denoted $max(\mathbf{z}_{d,-n})$. The probabilities of the k components down to that level ($1 \leqslant k \leqslant max(\mathbf{z}_{d,-n})$) are then given by [8]:

$$\begin{aligned} p(z_{d,n} = k \mid \mathbf{z}_{d,-n},\, m,\, \pi) &= E\left[V_k \prod_{j=1}^{k-1} (1 - V_j) \mid \mathbf{z}_{d,-n},\, m,\, \pi\right] = \\ &= E\left[V_k \mid \mathbf{z}_{d,-n},\, m\, \pi\right] \prod_{j=1}^{k-1} E\left[1 - V_j \mid \mathbf{z}_{d,-n},\, m,\, \pi\right] = \\ &= \frac{m\pi + \#[\mathbf{z}_{d,-n} = k]}{\pi + \#[\mathbf{z}_{d,-n} \geqslant k]} \prod_{j=1}^{k-1} \frac{(1-m)\pi + \#[\mathbf{z}_{d,-n} > j]}{\pi + \#[\mathbf{z}_{d,-n} \geqslant j]} \end{aligned} \tag{2.8}$$

Second, one more component is defined for the level distribution; the one for $k > max(\mathbf{z}_{d,-n})$, i.e. assigning to one of the levels below where words from document d have currently been assigned. The probability for this component is given by [8]:

$$\begin{aligned} &p(z_{d,n} > max(\mathbf{z}_{d,-n}) \mid \mathbf{z}_{d,-n},\, \mathbf{w},\, m,\, \pi,\, \eta) = \\ &= 1 - \sum_{j=1}^{max(\mathbf{z}_{d,-n})} p(z_{d,n} = j \mid \mathbf{z}_{d,-n},\, \mathbf{w},\, m,\, \pi,\, \eta) \end{aligned} \tag{2.9}$$

If this last component is sampled during level allocation, a specific level $k > max(\mathbf{z}_{d,-n})$ needs to be chosen. This is done by repeatedly sampling from a Bernoulli distribution until $z_{d,n}$ is determined. For increasing values of i, starting with $i = 1$, the ith sample represents choosing $k = max(\mathbf{z}_{d,-n}) + i$:

$$\begin{aligned} &p(z_{d,n} = k \mid z_{d,-n},\, z_{d,n} > k-1,\, \mathbf{w},\, m,\, \pi,\, \eta) = \\ &= (1-m) p(w_{d,n} \mid \mathbf{z},\, \mathbf{c},\, \mathbf{w}_{-(d,n)},\, \eta) p(z_{d,n} > k \mid z_{d,-n},,\, z_{d,n} > k-1) \end{aligned} \tag{2.10}$$

Sampling the last component, and allocating to a previously unallocated level, changes the value of $max(\mathbf{z}_{d,-n})$. This is important to note when performing subsequent allocations.

The second term in Equation 2.7, $p(w_{d,n} \mid \mathbf{z},\, \mathbf{c},\, \mathbf{w}_{-(d,n)},\, \eta)$, is the probability of the word $w_{d,n}$ being assigned to the topic on level $z_{d,n}$ in path c_d, given how the other words have been assigned [8]:

$$p(w_{d,n} \mid \mathbf{z},\, \mathbf{c},\, \mathbf{w}_{-(d,n)},\, \eta) \propto \#[\mathbf{z}_{-(d,n)} = z_{d,n},\, \mathbf{c}_{z_{d,n}} = c_{d,z_{d,n}},\, \mathbf{w}_{-(d,n)} = w_{d,n}] + \eta \tag{2.11}$$

where η is the Dirichlet distribution concentration parameter used to sample the topic distributions β_i.

Setting and Sampling the Hyperparameters

It is important to understand what assumptions are made when setting the hyperparameters of the hLDA model. Their values affect the probability distributions over words within the topics, as well as the size of the tree.

When drawing the topic distributions from a Dirichlet distribution, the parameter vector η is used. Thus, using small values in η corresponds to placing high probability on only a few words, while larger values favour mixtures of many words. (See Section 2.3).

Often times a symmetrical Dirichlet distribution is used, meaning that all values in the L-dimensional Dirichlet parameter vector are equal. Recall that η is then a scalar called a concentration parameter.

The parameter of the nCRP, γ, is used for sampling paths from the infinite tree. It governs the likelihood of visiting previously unvisited nodes, and therefore the size of the

tree. However, it should be noted that η will also affect the size of the tree, as the probabilities of words within the topics also governs how many topics are required to model the corpus [8].

The stick-breaking distribution, from which level distributions within paths are drawn, has two parameters; m and ρ. They determine the proportions of assigning words to general topics near the root, relative to assigning words to more specific topics further down in the path.

The hyperparameters can be included in the posterior inference process by assigning prior distributions to them, and adding Metropolis-Hastings (MH) iterations between the iterations in the Gibbs sampler [8]:

$$\begin{aligned} m &\sim Beta(\alpha_1,\ \alpha_2) \\ \pi &\sim Exponential(\alpha_3) \\ \gamma &\sim Gamma(\alpha_4,\ \alpha_5) \\ \eta &\sim Exponential(\alpha_6) \end{aligned} \tag{2.12}$$

The parameters of the hyperparameters, α_i, are called *hyper-hyperparameters*.

The MH algorithm is a MCMC method like Gibbs sampling, but the steps in the algorithm are different. Each new candidate state (in this case hyperparameter setting) is sampled from a proposal distribution, conditioned on the current state [13]. The new state is accepted or rejected with a certain probability called acceptance probability. The acceptance probability for a new state x' given current state x is defined as:

$$A(x',\ x) = min\left(\frac{f(x)}{q(x|x')}\frac{q(x'|x)}{f(x')},\ 1\right) \tag{2.13}$$

where f is the target distribution that the MH chain converges toward, and q is the conditional probability of proposing x' given x.

Sampling the hyperparameters given fixed hyper-hyperparameters influences the inference less than using fixed hyperparameters. However, including the MH steps in the sampling process may slow down the convergence of the Markov chain [13].

Assessing Convergence of the Markov Chain

The goal of the sampling process is to approximate the mode of the posterior distribution. In order to make it a good approximation, this should be done when the Markov chain is close to its stationary distribution. There are thus two challenges; approximating the mode of the posterior distribution and assessing the convergence of the chain.

Approximating the mode can be done by calculating the log probability of each state, $\mathcal{L}^{(t)}$, i.e. of a certain configuration of document paths, level allocations and observations, given the hyperparameters [8]:

$$\mathcal{L}^{(t)} = \log p(\mathbf{c}_{1:D}^{(t)},\ \mathbf{z}_{1:D}^{(t)},\ \mathbf{w}_{1:D} \mid \gamma,\ \eta,\ m,\ \pi) \tag{2.14}$$

The state with the highest log probability can then be chosen to approximate the mode.

To assess the convergence of the chain, and determine how many iterations to perform, the autocorrelation of $\mathcal{L}^{(t)}$ can be evaluated. Autocorrelation is a measure of how strongly samples correlate to each other, depending on how many iterations are performed between them. The number of iterations between the samples is called lag. For example, a lag k autocorrelation is the correlation between samples that are k iterations apart.

If the autocorrelation does not decrease quickly when increasing the lag, it is likely that the chain has not yet converged to its stationary distribution. This is because without a perfect starting state, the chain will start exploring areas with low probability and move toward areas of higher probability. The movement of the chain is not completely random as it does this, and therefore there will be high correlation between the samples.

However, when the chain has approached its stationary distribution, it will only make smaller, random movements around the mode. The result is that the autocorrelation approaches zero with increasing lag.

The initial set of iterations it takes for the chain to approach the mode, and thus for the autocorrelation to quickly move toward zero for increasing lag, can be discarded to achieve a more accurate mode approximation. The approximation will then not be biased by the early, far from converged, states of the chain.

There may be several local modes that the chain can converge toward depending on its initial state and the random seed used. Therefore, several chains are usually run and the local mode with the highest log probability is chosen.

2.9 Constrained hLDA

This section describes how hLDA can be extended to incorporate prior knowledge about the topics that exist in the corpus into the model. The goal is to derive a model with improved interpretability and predictive abilities compared to a hLDA model. The extension was introduced by Wang et al. [6] and is called constrained hLDA.

The inspiration for constrained hLDA comes from work by Andrzejewski and Zhu [25], where partial supervision was added to the LDA model. An indicator function was defined, that given some condition either evaluates to zero or one. The indicator function can be used to create topics that contain predefined words, by only allowing assignment of certain words to certain topics.

The constrained hLDA model is similar to the original hLDA model, but a slightly modified nCRP scenario is considered. The resulting topic tree has the same hierarchical structure as hLDA, but some of the topics are predefined by a set of words that are commonly observed together in the corpus.

The nodes of the predefined topics are called *constrained nodes*, and are considered in the path allocation step of the model. If a document contains words that match a set of words associated with a constrained node, it has a higher probability of being assigned there.

Extracting the Constrained Nodes

The constrained nodes are defined by utilizing the Frequent Pattern (FP) tree algorithm. A FP-tree is a data structure used for storing frequency pattern information, described by Han et al. [26].

A FP-tree can be constructed from a corpus of documents by collecting word frequencies across all documents as well as for each document. Only words that occur in more than a certain number of documents, and in less than a certain number of documents, will be included in the FP-tree. The upper and lower limits are called *maximum* and *minimum support*, respectively.

One result from the corpus iteration is a set of words that fall within the minimum and maximum support constraints. This set is denoted $frequent\ words$ and sorted by frequency in descending order.

Another result from the iteration is a set of words for each document. However, all words that are not in $frequent\ words$ are now discarded from those sets.

The nodes in the FP-tree represent words, and each document is assigned a path in the tree according to its word set. The order of the nodes in a path reflects the frequencies of the words across the corpus. The most frequent word in a document, according to the order in $frequent\ words$, should appear on the level below the root node, followed by the second most frequent word on the level below and so on. Multiple documents can share the same path, or parts of a path. A count is usually associated with each node to keep track of how many documents pass through it. An example FP-tree is shown in Figure 2.7.

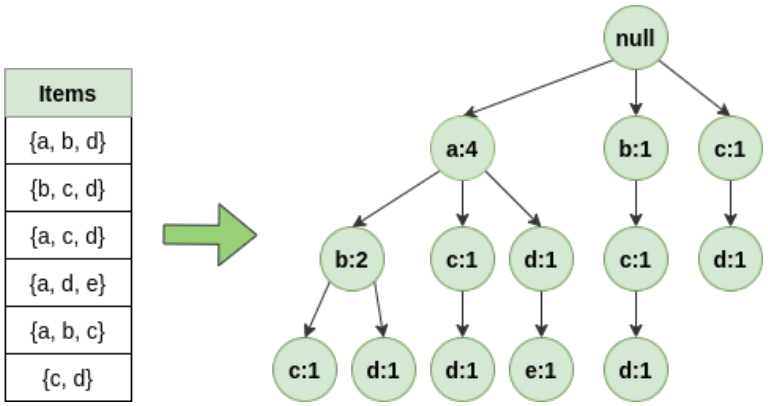


Figure 2.7: Example FP-tree created from six item sets ordered by frequency (descending). The frequency order is {a, b, c, d, e}. The number shown in each node represents how many paths passed through it. Think of {a, b, c, d, e} as the *frequent words* set and the items sets as documents containing words from it.

The FP-tree is used to define the constrained nodes. If the correlation between words in $frequent\ words$ is above a certain threshold, they are grouped to form a constrained node. In the work by Wang et al [6] and in this thesis, the correlation between two words is given by:

$$\text{correlation}(A,\ B) = \frac{P_{A,B}}{\min(P_A,\ P_B)} \tag{2.15}$$

where $P_{A,B}$ is the number of times word A occurs in the same document as word B, and P_A is the number of times word A occurs in a document. The correlation is a factor between zero and one.

The algorithm for extracting the constrained nodes from the corpus is shown in Algorithm 1 [6].

Algorithm 1 EXTRACTING THE CONSTRAINED NODES

```
constrained_nodes ← [ ]
for freq_word_i in freq_words do
    constrained_node_i ← [freq_word_i]
    for freq_word_{j | j != i} in freq_words do
        if correlation(freq_word_i, freq_word_j) > threshold then
            Add freq_word_j to constrained_node_i
        end if
    end for
    Add constrained_node_i to constrained_nodes
end for
for constrained_node_i in constrained_nodes do
    for constrained_node_{j | j != i} in constrained_nodes do
        if constrained_node_i == constrained_node_j then
            Remove constrained_node_j from constrained_nodes
        end if
    end for
end for
```

Sampling Document Paths in Constrained hLDA

As previously mentioned, a slightly modified variant of the nCRP is used in constrained hLDA. The original nCRP is described in Section 2.7. The difference in this variant is that some of the customers now have a list of special dishes they would like to taste, which is taken into consideration when selecting the next restaurant to visit. Some of the restaurants offer these special dishes, and if there is one that matches the customer's list, the customer is more likely to eat there.

In this case, the restaurants with special dishes are the constrained nodes, and the special dishes are the words from *frequent words* associated with the nodes. The customers are the documents, and are said to have a list of special dishes if they contain words from *frequent words*.

In this thesis, only the restaurants on the first level below the root restaurant will contain constrained nodes.

With this modified variant of the nCRP in mind, the generative process of constrained hLDA can be described as:

1. For each node in the infinite tree, draw a topic $\beta_k \sim$ Dirichlet(η).
2. For each document $d \in \{1, 2, \ldots, M\}$:
   (a) Choose a path c_d:
      (i) Let $c_{d,1}$ be the root node.
      (ii) If d contains the set of words that represents a constrained node c_i:
      Let $c_{d,2}$ be c_i with probability $\frac{n_{c_i}+\gamma}{\sum(n_{c_i}+\gamma)}$, where n_{c_i} is the number of customers at c_i.
      (iii) Otherwise:
      Draw $c_{d,2} \sim nCRP(\gamma)$ from the restaurants that do not have special dishes.
      (iv) For each of the levels $\ell \in [3, 4, \ldots, L]$, draw $c_{d,l} \sim nCRP(\gamma)$.
   (b) Draw a distribution over levels $\theta_d \sim$ GEM(m, ρ).
   (c) For each word $n \in \{1, 2, \ldots, N_d\}$ in d:
      (i) Choose a level $z_{d,n} \sim$ Discrete(θ_d)
      (ii) Choose a word $w_{d,n} \sim$ Discrete($\beta_{c_d,z_{d,n}}$) where $\beta_{c_d,z_{d,n}}$ is the topic distribution for the node at the chosen level $z_{d,n}$.

Figure 2.8 shows an example tree structure of a constrained hLDA model with $L = 3$.

To incorporate the constrained nodes into the path sampling, an indicator function like the one defined by Andrzejewski and Zhu [25] is added to the probability of selecting a path [6]:

$$p(\mathbf{c}_d \mid \mathbf{w},\ \mathbf{c}_{-d},\ \mathbf{z},\ \eta,\ \gamma) \propto (\eta'\delta(\mathbf{w}_d,\ \mathbf{c}_d) + 1 - \eta')p(\mathbf{c}_d \mid \mathbf{c}_{-d},\ \gamma)p(\mathbf{w}_d \mid \mathbf{c},\ \mathbf{w}_{-d},\ \mathbf{z},\ \eta) \quad (2.16)$$

Here the indicator function is denoted $\delta(\mathbf{w}_d,\ \mathbf{c}_d)$, and evaluates to one if the path $\mathbf{c}_d$ contains a *matching constrained node*. A matching constrained node is a constrained node represented by a set of words from *frequent words* that also occur in $\mathbf{w}_d$.

The parameter $\eta' \in [0,\ 1]$ is used to control the relaxation of the constraint:

$$\eta'\delta(\mathbf{w}_d,\ \mathbf{c}_d) + 1 - \eta' = \begin{cases} 1, & \text{If } \mathbf{c}_d \text{ contains matching constrained node} \\ 1 - \eta', & \text{Otherwise} \end{cases} \quad (2.17)$$

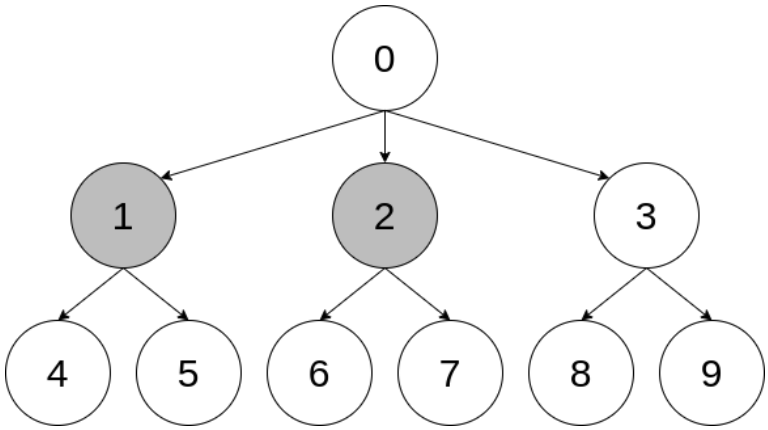


Figure 2.8: Example tree structure of a constrained hLDA model with $L = 3$. The grey nodes are predefined constrained nodes and the white nodes have been generated through the nested Chinese Restaurant Process.

Sampling Level Allocations in Constrained hLDA

Another modification of hLDA included in constrained hLDA concerns the level allocation of words. Wang et al. [6] used the frequency of words in combination with whether they are part of a predefined list of Part of Speech (POS) words, to determine whether to allocate the word directly to the root node, to the levels below or to sample a level from all levels.

This modification is not included in the constrained hLDA model used in this thesis. The motivation is that the extra computational power the modification requires would likely not yield a significant improvement in level allocation performance.

3 Method

This chapter details the method used to answer the research questions of the thesis. It describes the format of the data that was used and how it was preprocessed. It also describes how a hLDA and a constrained hLDA model were generated based on the data as well as how the resulting models were visualized.

All data processing and model visualization was done using Python2.7 and Python3.6. The hLDA model implementations are written in C.

This work presents a novel evaluation metric called word frequency similarity. Its definition and how it was used to evaluate the two topic models is described in this section.

3.1 Data

The dataset used in this work was provided by Opera and consists of 333,552 text articles. The articles were collected from a diverse range of domains on the Internet.

Each article has a title and a description. The lengths of the descriptions vary from article to article. Some article descriptions are only a few sentences long while others contain over a thousand words.

Before the data was used in the models, four preprocessing steps were performed. First, the title and description of each document were concatenated. Second, any HTML code was removed along with any non-alphabetic characters. Third, a stemmer algorithm[1] was applied to each word. Finally, words from a certain list of unwanted words were removed from the text. The stemmer and the list of unwanted words used are from a Python library called *nltk*[2].

The list of unwanted words contained words that generally do not contribute to identifying the topics in a text, such as *what*, *was* and *have*. Words like these occurred frequently across the articles, and were expected to be assigned to the root node of the hLDA models. They would therefore not affect any of the topics below. However, removing them reduced the size of the corpus, and thereby the amount of data to be processed, significantly.

The preprocessed articles containing less than 100 or more than 350 words were discarded. This resulted in a set of 12,587 articles. From these, 10,000 were randomly selected to be used in the hLDA models. 10,000 was chosen as the number of articles to include as it resulted in a

[1] A stemmer algorithm reduces words to their base form, e.g. *walking* is stemmed to *walk*.
[2] https://www.nltk.org

reasonable runtime for the sampling process. This was also the reasoning behind discarding articles containing too many words. The articles containing too few words were discarded because they are harder to categorize, as there are fewer words to motivate the categorization.

3.2 Topic Modeling

This section describes how the hLDA and constrained hLDA models were derived. For the hLDA model, a C implementation written by David Blei[3] was used. David Blei is one of the authors who first presented the hLDA topic model [4].

No publicly available implementation of constrained hLDA was found. Therefore, the hLDA implementation was extended to implement constrained hLDA as part of this thesis work.

Hierarcical Latent Dirichlet Allocation (hLDA)

The hLDA implementation initializes a Gibbs sampler and performs sampling iterations according to the collapsed Gibbs sampling algorithm described in Section 2.8.

Before running the sampler, the iteration- and hyperparameter settings are defined. The iteration settings determine how many times the program should initialize the sampler to find a good start state, how many iterations to perform and how many times the entire sampling process should be run. The iteration settings used are shown in Table 3.1.

Initialize start state	100 times
Iterations	3000
Restarts	1

Table 3.1: Iteration settings used for the hLDA model.

It is important to perform a sufficient number of iterations of the sampler in order to ensure that the Markov chain has converged. However, as described in Section 2.8, it is challenging to determine exactly how many iterations are required. The number of iterations used in this work was determined based on test runs. In these test runs, only minor changes in the model structure were observed after about 2,000 iterations. Therefore, 3,000 was selected as the number of iterations to perform.

The hyperparameter settings define the depth of the tree, the starting values of the hLDA hyperparameters, and whether the hyperparameters should be sampled between each iteration. The depth of the tree was set to $L = 3$, as it gave the level of abstraction in the topics desired by Opera.

The hLDA hyperparameter settings used for η and γ are shown in Table 3.2. The GEM parameters m and ρ were set to 0.3 and 1,000, respectively, and the hyperparameters were not sampled between each iteration.

Level	0 (root)	1	2
η	10.0	2.0	0.4
γ	1.0	1.0	-

Table 3.2: Hyperparameter settings used for the hLDA model.

The hyperparameter settings were selected through performing test runs to find a suitable tree structure for the corpus data. Section 2.8 discusses the assumptions made when setting the hyperparameters.

[3] http://www.cs.columbia.edu/~blei/topicmodeling_software.html

The implementation requires the corpus data to have a specific format. Each article in the corpus is represented as a line in a corpus file. The line begins with the number of unique words in the article, followed by the word counts for each word. The words are represented as line indices of a vocabulary file, which contains one word per line. This means that the actual words do not have to be handled during the model generation.

Only words that occur in 100 or more articles are included in the vocabulary and thus considered in the modeling. All other words are deemed too infrequent to be included in any topic.

As previously mentioned, the mode of the distribution of topic trees is approximated with the state with the highest log probability (see Section 2.8). The model outputs this state through three files; one containing the topic tree structure, one with the path assignments for each article, and one with the level allocations for each for in each article.

Constrained hLDA

The hLDA implementation was extended in two ways to implement constrained hLDA; to include the predefined topics in the hierarchy, and to incorporate the modification to the path sampling probability shown in Equation 2.16.

The predefined topics were extracted from the corpus by creating a FP-tree and forming groups of words based on their correlation. This process is described in Section 2.9. The minimum and maximum support used was 0.025 and 0.06, respectively, and a correlation threshold of 0.35 was used. These values were chosen through experimentation and evaluating the resulting constrained topics.

A subset of the generated predefined topics were discarded due to one or several of the following reasons; they contained only one or more than six words, other predefined topics were too similar or they contained only HTML code.

The same iteration and hyperparameter settings were used for constrained hLDA as for the hLDA model. The parameter η' in the path sampling (Equation 2.16) was set to 10^{-12}, which meant that matching nodes would almost certainly be chosen whenever there were any. This was done to exaggerate the effect of the constrained topics in the model.

The format of the output from the model was not modified.

3.3 Visualization

This section describes how the output from the two hLDA models was used to visualize the topic hierarchies, the path allocations of the articles and the level allocations of the words.

Visualizing the Hierarchy

The topic hierarchy was visualized by parsing the output file describing the tree structure. Each node in the tree is denoted by an ID and all nodes except the root node has a parent. Using this information, each node is associated with a list of child nodes, and the entire tree can be visualized by starting from the root node and iteratively showing its child nodes. The tool used for the graphic representation was *GraphViz*[4].

The seven most frequently occurring words for each topic are shown inside the nodes. The nodes also show how many words and documents were assigned to them. These two values can be seen at the top of each node inside brackets.

As hLDA is based on the nCRP, each iteration may cause new topics to be generated. Initially, these topics may not be well defined, as they are only represented by a handful of articles. Therefore, topics with less than 20 articles assigned to them were disregarded to achieve a better overview of the more established topics.

[4]https://www.graphviz.org

In the case of constrained hLDA, the predefined topics were colored blue while the regular topics were colored green. The predefined topics show the words used when determining if the node is a matching node, i.e. the words from the predefined topics, in bold. Shown below them are the seven most frequent words in the topic, just as in the regular nodes.

Figure 4.1 and Figure 4.3 show the hierarchy trees from the hLDA and constrained hLDA models generated in this work.

Visualizing Path- and Level Allocations

The article paths and level allocations were visualized for each document in the corpus using output from the model generation.

To visualize the article paths, each topic in the path is shown, starting from the root topic and ending with the leaf topic. For each topic in the path the ten most frequent words are shown along with how many words and articles have been assigned to that topic. Next to each word is also a weight, indicating how often that word occurs within the topic.

To visualize the article specific word allocations, color coding is used. The root topic is green and the following topics are purple and blue, respectively. The input document, i.e. the preprocessed article text, is shown below the hierarchy. Each word in the text has the same color as the topic it was assigned to. For example, a word will be green if it was assigned to the root topic, or purple if it was assigned to the topic below the root. If a word occurred in less than 100 articles across the corpus, and thus deemed too infrequent to be included in the model, it is colored black.

The path- and level allocation visualized for an article in the hLDA model is shown in Figure 4.2, and for the same article in the constrained hLDA model in Figure 4.4.

3.4 Word Frequency Similarity

When the words from the articles in the corpus are allocated to topics in the hierarchy, the result is a word count vector for each topic. The word counts were normalized to derive a word frequency between zero and one for each word.

Word frequency similarity is a novel measure introduced in this work. It indicates how similar the word frequencies in two topics are. If the relative frequency of a word is similar in two topics, the similarity score is close to one. On the other hand, if it is higher or lower in one compared to the other, the similarity score is closer to zero.

The similarity between two word frequencies wf_1 and wf_2 is given by:

$$\text{similarity}(wf_1,\ wf_2) \ = \ \frac{\min(wf_1,\ wf_2)}{\max(wf_1,\ wf_2)} \tag{3.1}$$

This word frequency similarity was calculated for a specific subset of words from the vocabulary. Only words that occurred frequently in one or both of the topics were considered, and words with low frequencies in both topics were disregarded. The reasoning behind this is that these words have little importance in the topics but would yield high similarity scores. If the vocabulary is large, they would have a large impact on the mean similarity while the impact from the more meaningful words would be less significant.

The set of words was selected by merging the 15 most frequent words from each of the two topics, resulting in a set of 15 to 30 words. The size of the sets varied as some topic pairs may share one or several words among their most frequent words. Calculating the word frequency similarity for each word in the set resulted in a vector of word frequency similarities for each topic pair.

By adding all word frequency similarities in the vector and dividing that sum by the number of words, the mean frequency similarity was derived. This value was calculated for all pairs of parent and child topics in the hierarchy.

4 Results

This chapter presents the results from the work. It includes visualizations of the hLDA and constrained hLDA topic models created from the article corpus. It also includes the word frequency similarities between the parent and child nodes in the hierarchy.

4.1 Hierarchical Latent Dirichlet Allocation (hLDA) Model

A hLDA model with depth three was generated from a corpus of 10,000 articles. The collapsed Gibbs sampler was run for 3,000 iterations. A portion of the tree structure is shown in Figure 4.1. The information shown in brackets in the topics are the number of words and articles assigned there. The total number of topics in the full tree is 244, where one is the root node, 28 are on the second level and 215 are on the lowest level in the tree.

Each article in the corpus has been assigned a path in the tree. For example, an article with the title *Spicy Vegan Beyond Breakfast Sausages Launch in Stores Nationwide*[1] was processed and used in the model. A snippet from the article is:

> "By the end of March, retailers nationwide will stock Beyond Meat's newest product, vegan Beyond Breakfast Sausage in two flavors: Classic and Spicy. The new vegan pork sausage—which is both Kosher- and Halal-certified—is made with a base of pea and brown rice protein combined with herbs and spices, such as sage and black pepper, that impart a flavor similar to traditional animal-based breakfast sausages."

The processed text used in the model, the path assigned to the article and the level allocations of each word are shown in Figure 4.2. The topics on the different levels in the tree have different colors. The root topic is green, and words assigned to the root level are shown in green in the input text. The topics and words on the second and third level in the tree are colored purple and blue, respectively. The words in black are not included in the model as they do not occur frequently enough across the corpus.

The information shown in brackets in the topics are the number of words and documents assigned there. The factors shown next to each word in the topics are the weights of each word, where a higher weight means the word occurs more often within the topic.

[1] https://vegnews.com/2020/3/spicy-vegan-beyond-breakfast-sausages-launch-in-stores-nationwide

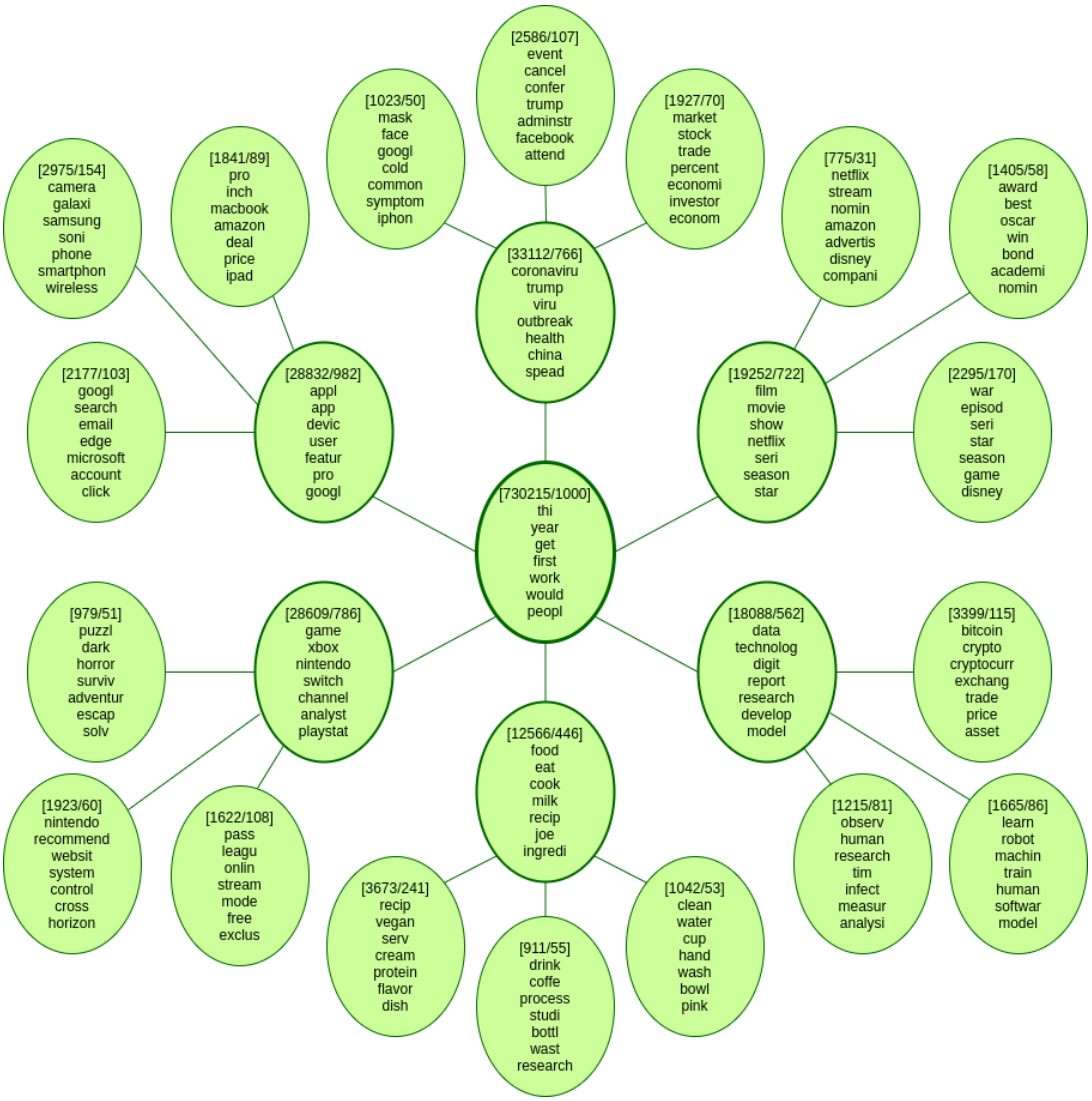


Figure 4.1: Portion of the tree structure of the hLDA model. The corpus consisted of 10,000 articles and the collapsed Gibbs sampler was run for 3,000 iterations. For each topic, the number of words and documents assigned to it is shown along with the seven most frequent words.

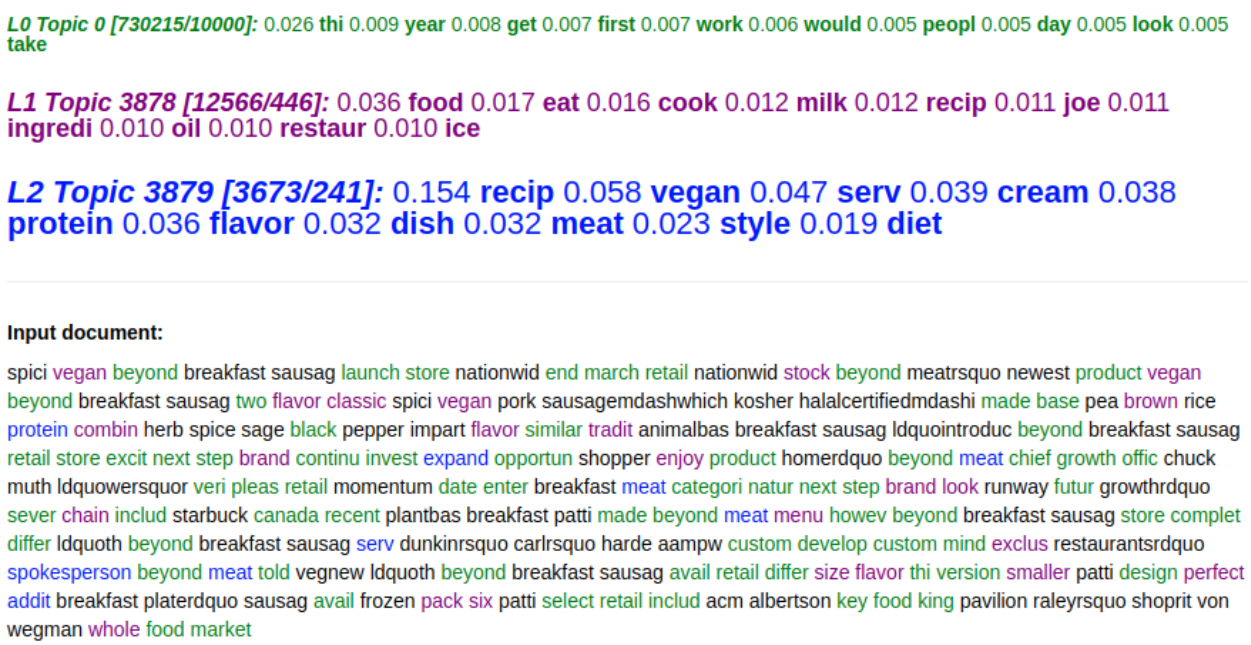


Figure 4.2: Input text, path and level allocations in the hLDA model for an article in the corpus.

4.2 Constrained hLDA Model

A constrained hLDA model with depth three was generated from the same corpus of 10,000 articles that was used for the hLDA model. The collapsed Gibbs sampler was run for the same number of iterations, i.e. 3,000. A portion of the tree structure is shown in Figure 4.3. The total number of topics in the full tree is 220, where one is the root node, 25 are on the second level and 194 are on the lowest level in the tree.

The path and the level assignments for the same article as for the hLDA model is shown in Figure 4.4.

4.3 Word Frequency Similarity

This section presents the word frequency similarity results from the two hLDA topic models. For both models, the mean word frequency similarity was calculated for all parent-child pairs. The parents are the topics on the level below the root node in the hierarchy, and the children are the topics below the parents. Only topics with 20 or more articles assigned to them were considered.

The results include the topic pairs with the lowest and highest mean word frequency similarity for each of the models, as well as a few example topic pairs.

Hierarchical Dirichlet Allocation (hLDA) Results

The average mean word frequency similarity for the parent-child topic pairs in the hLDA model was a 7.2 %. The lowest mean word frequency similarity was 0.3 % and the highest was 22.8 %. Figures 4.5 and 4.6 show the parent-child topic pairs with the lowest and highest word frequency similarity, respectively.

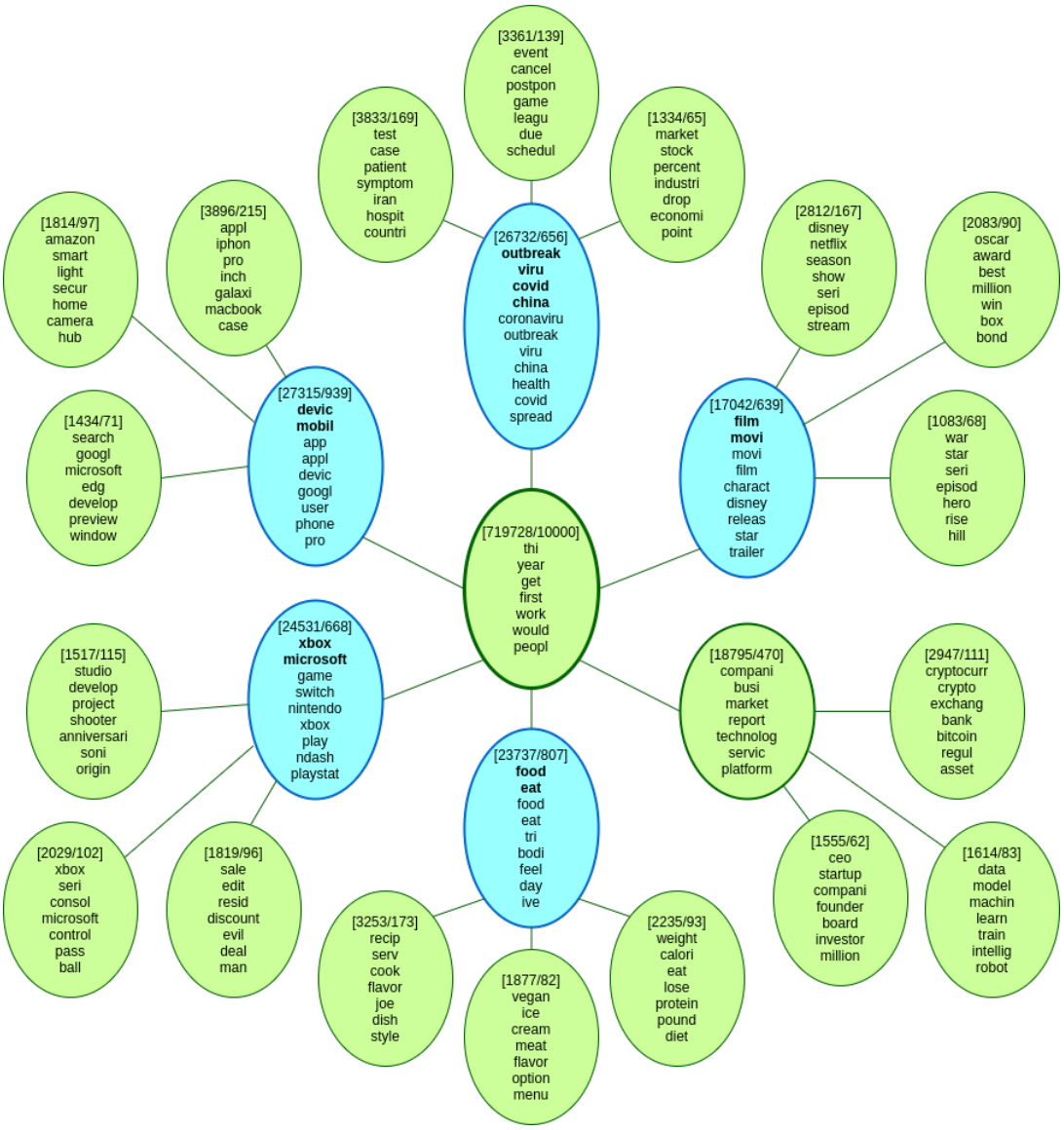


Figure 4.3: Portion of the tree structure of the constrained hLDA model. The corpus consisted of 10,000 articles and the collapsed Gibbs sampler was run for 3,000 iterations. For each topic, the number of words and documents assigned to it is shown along with the seven most frequent words. The constrained topics are shown in blue and also contain the words used to match documents to it in bold.

Figures 4.7 and 4.8 show two topic pairs with a parent related to design[2]. The topics in the first pair, with a mean word frequency similarity of 12.8 %, look well-defined and the child specifies a clear subtopic related to design. The child topic in the second pair, with a mean word frequency similarity of 2.6 %, however, seems to define a subtopic about football, which most people may not deem a subtopic of design.

[2]It should be noted that stating that the parent is related to design is a subjective interpretation. Additional interpretations like this one are made as the results are presented in this work.

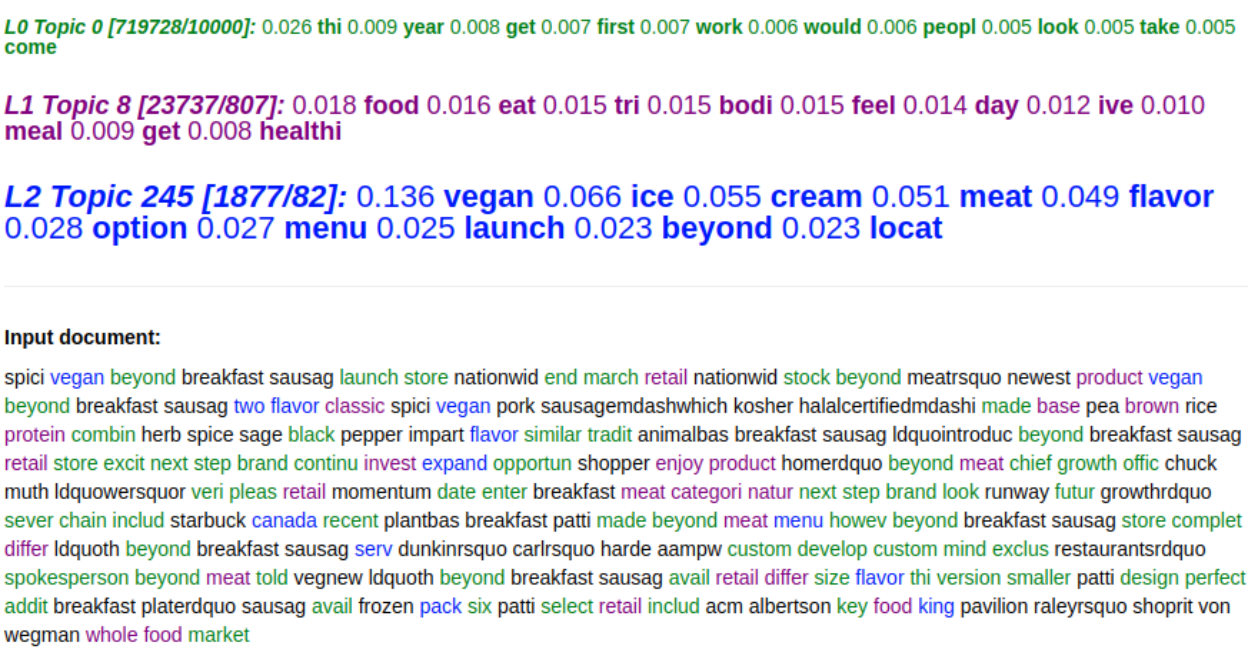


Figure 4.4: Input text, path and level allocations in the constrained hLDA model for an article in the corpus.

Constrained hLDA Results

The average mean word frequency similarity for the parent-child topic pairs in the constrained hLDA model was a 6.2 %. The lowest mean word frequency similarity was 0.4 % and the highest was 18.6 %. Figures 4.9 and 4.10 show the parent-child topic pairs with the lowest and highest word frequency similarity, respectively.

Figures 4.11 and 4.12 show two topic pairs from the constrained model, with a parent related to movies. The first pair shows a clear movie subtopic and a mean word frequency similarity of 5.8 %. The child in the second pair is about hair and makeup, which most may not consider an obvious subtopic of movies, and a mean word frequency similarity of only 0.8 %.

Figures 4.13 and 4.14 show another two topic pairs from the constrained model, with a parent related to American politics. The first figure shows a pair with word frequency similarity 12.2 % and a child that defines a subtopic related to presidential candidates. The second figure shows a pair with a word frequency similarity of 0.8 %, where the child topic is about tech companies.

L1 Topic 39 [12306/351]: bodi workout muscl fit exercis train gym weight leg tri strength feel ive get goal

L2 Topic 350 [1032/22]: trump impeach senat presid trial wit republican vote democrat block defens donald argument hous white

Similarity Mean: 0.0028590745701857637

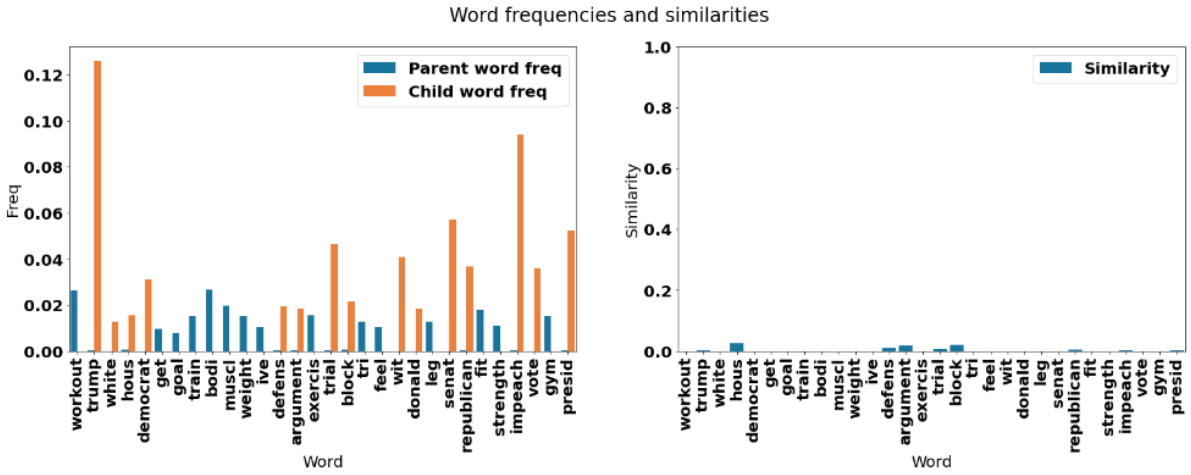


Figure 4.5: Parent-child topic pair with minimum word frequency similarity (0.3 %) in the hLDA model.

L1 Topic 193 [28609/786]: game xbox nintendo switch channel analyst playstat youtub play twitter content launch twitch support consol

L2 Topic 194 [5862/241]: unit sale analyst vgchartz million chart sold lead channel author gap junior ndash articl expand

Similarity Mean: 0.22837043329497048

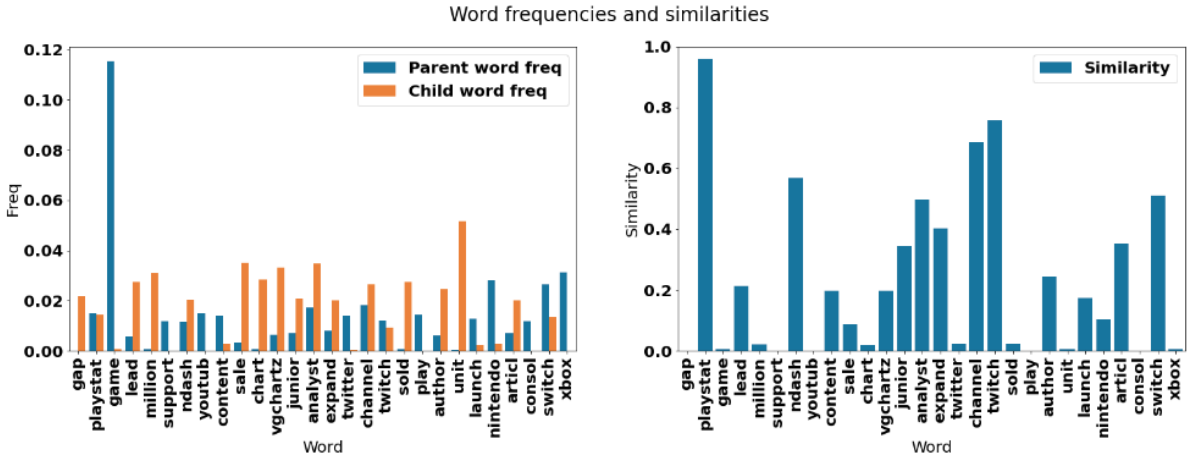


Figure 4.6: Parent-child topic pair with maximum word frequency similarity (22.8 %) in the hLDA model.

L1 Topic 985 [14451/497]: design view post angel share featur art star pst collect open instagram photo feb jan

L2 Topic 1843 [2538/127]: exhibit galleri art artist hypebeast paint usd work click piec collabor view instal collect explor

Similarity Mean: 0.12770957075414124

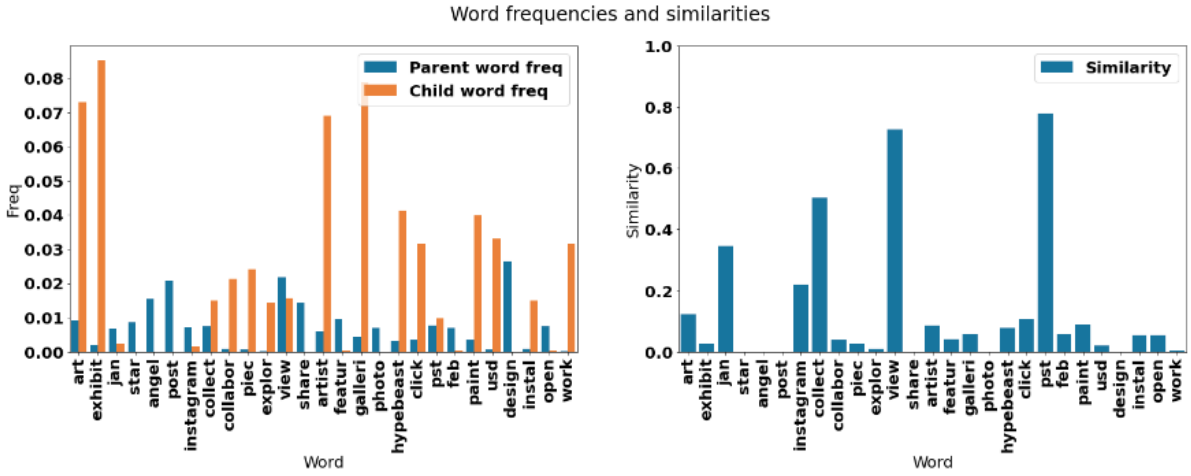


Figure 4.7: Parent-child topic pair from the hLDA topic hierarchy with mean word frequency similarity 12.8 %. Both topics seem related to design, and the child topic seems to specify a subtopic related to art/galleries.

L1 Topic 985 [14451/497]: design view post angel share featur art star pst collect open instagram photo feb jan

L2 Topic 2960 [663/35]: bowl super commerci squar footbal san francisco mountain chief sunday advertis flavor championship ball score

Similarity Mean: 0.0262420634433746

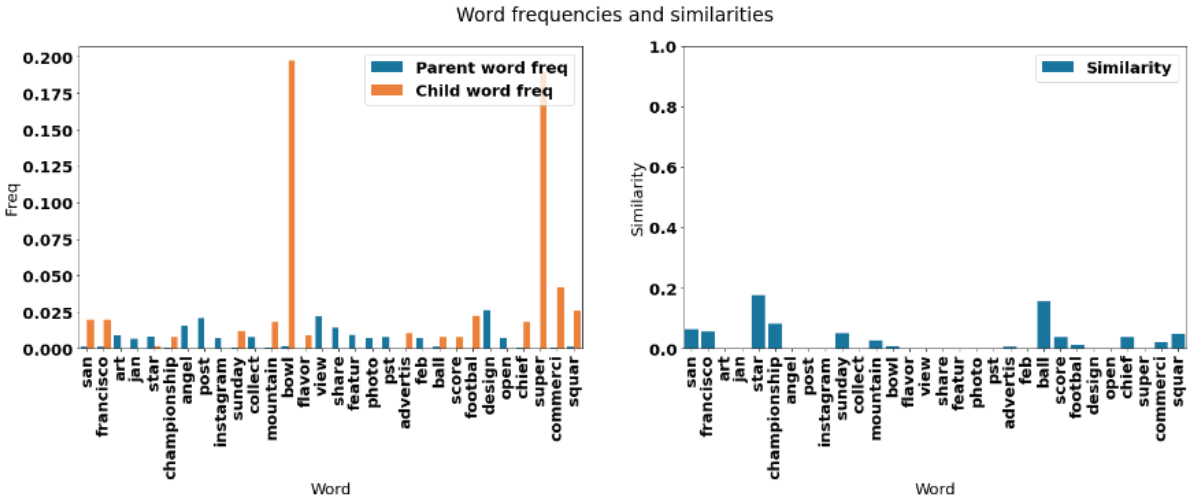


Figure 4.8: Parent-child topic pair from the hLDA topic hierarchy with mean word frequency similarity 2.6 %. The parent topic seems related to design, but the child topic seems related to football.

L1 Topic 3 [11272/409]: sale million month home increas sold compar market total ndash design data hous adjust rate

L2 Topic 634 [904/26]: articl dog copi copyright appreci law awar paid rest piec consid public entir love appear

Similarity Mean: 0.004379999755124488

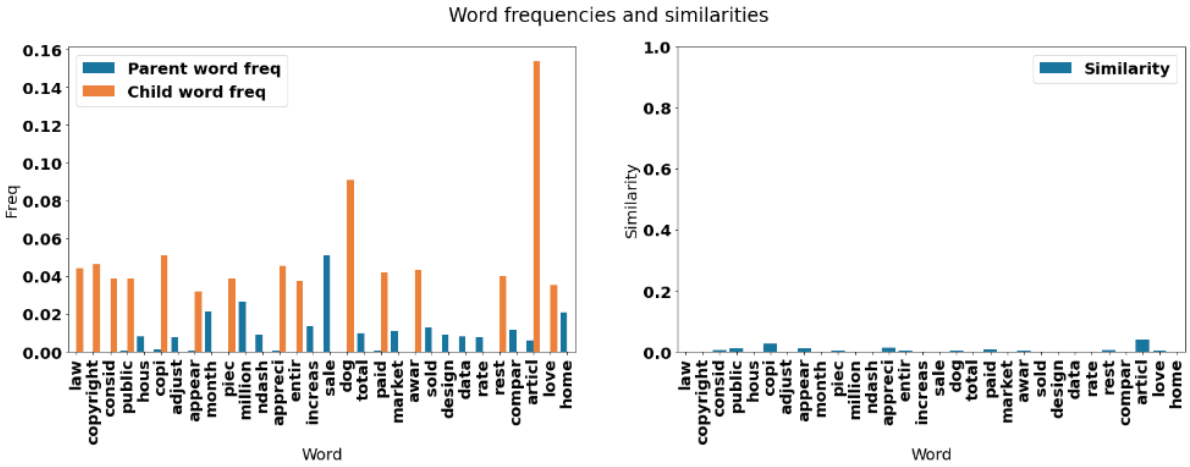


Figure 4.9: Parent-child topic pair with minimum word frequency similarity (0.4 %) in the constrained hLDA model.

L1 Topic 67 [7017/190]: area weather water temperatur south wind across mountain degre winter citi region air cold north

L2 Topic 68 [2264/77]: storm rain inch weather across wind heavi sever temperatur southern forecast thursday air averag central

Similarity Mean: 0.18628716045948054

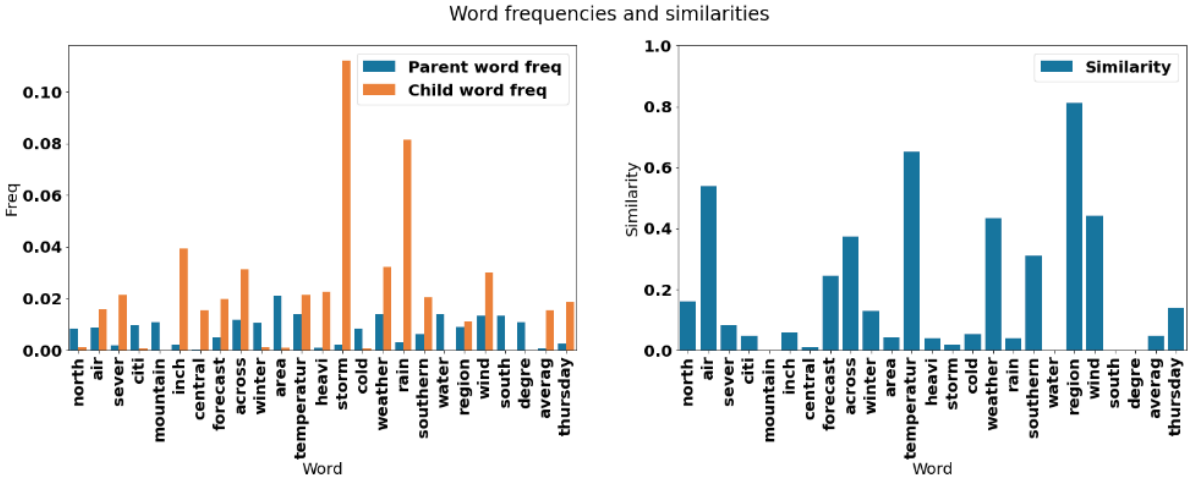


Figure 4.10: Parent-child topic pair with maximum word frequency similarity (18.6 %) in the constrained hLDA model.

L1 Topic 4 [17042/639]: movi film charact disney releas star trailer netflix actor origin role cast director seri stori

L2 Topic 278 [2812/167]: disney netflix season show seri episod stream plu servic nomin amazon park prime execut subscrib

Similarity Mean: 0.05802559265657631

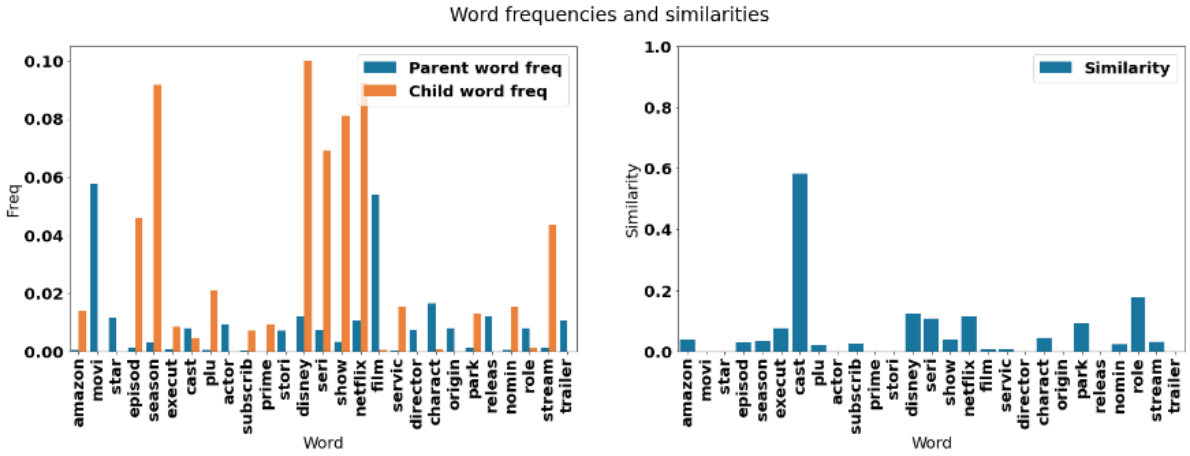


Figure 4.11: Parent-child topic pair from the constrained hLDA topic hierarchy with mean word frequency similarity 5.8 %. Both topics seem related to movies, and the child topic seems to specify a subtopic related to streaming.

L1 Topic 4 [17042/639]: movi film charact disney releas star trailer netflix actor origin role cast director seri stori

L2 Topic 1919 [1289/52]: hair color trend style spring highlight cut ahead makeup shade brown stress remov martin skin

Similarity Mean: 0.007813434207559464

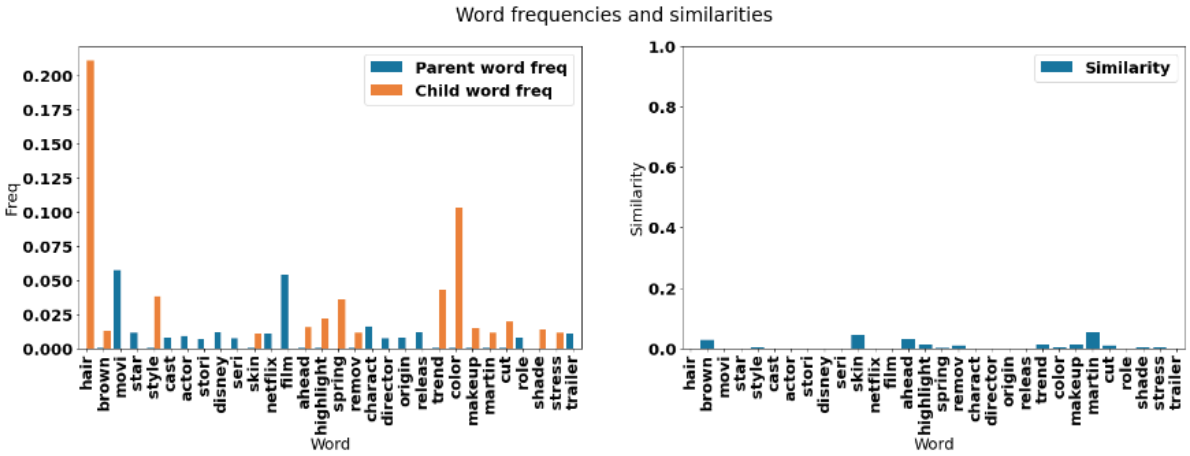


Figure 4.12: Parent-child topic pair from the constrained hLDA topic hierarchy with mean word frequency similarity 0.8 %. The parent topic seems related to movies, but the child topics seems related to hair and makeup.

L1 Topic 2 [23522/572]: trump presid democrat former campaign state donald vote elect senat polit stori bloomberg sen parti

L2 Topic 133 [4447/164]: sander biden primari voter poll berni joe carolina south vote win candid percent tuesday super

Similarity Mean: 0.12167283030381108

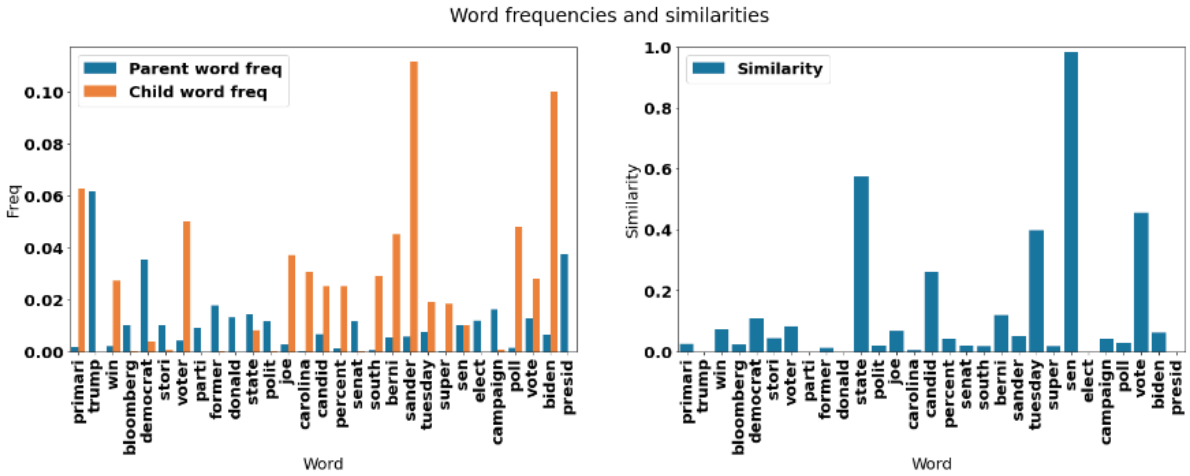


Figure 4.13: Parent-child topic pair from the constrained hLDA topic hierarchy with mean word frequency similarity 12.2 %. Both topics seem related to American politics, and the child topics seems to specify a subtopic related to presidential candidates.

L1 Topic 2 [23522/572]: trump presid democrat former campaign state donald vote elect senat polit stori bloomberg sen parti

L2 Topic 2082 [645/27]: facebook tech compani regul ceo employe climat jack twitter amazon protect microsoft smart googl firm

Similarity Mean: 0.008472687088224459

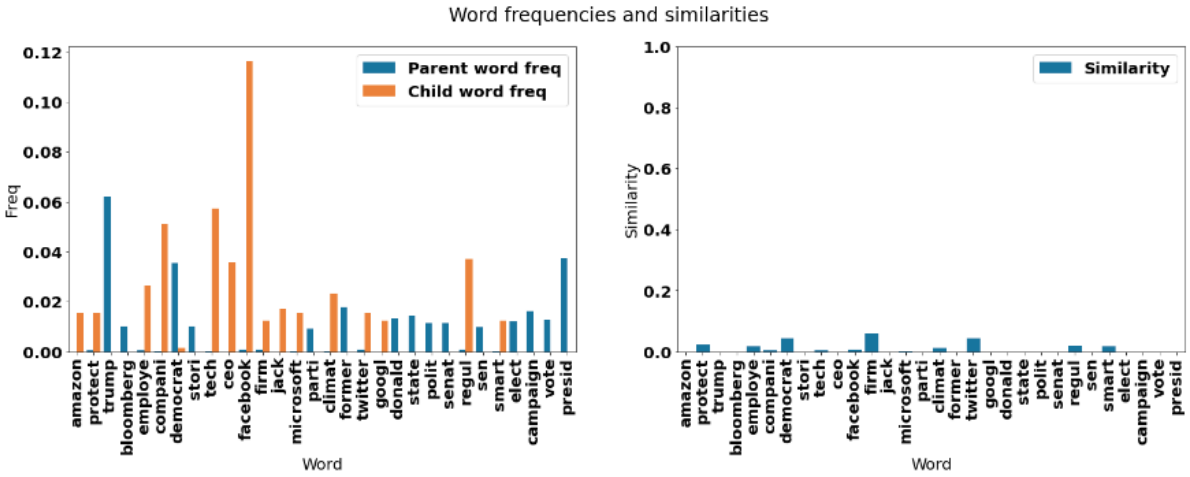


Figure 4.14: Parent-child topic pair from the constrained hLDA topic hierarchy with mean word frequency similarity 0.8 %. The parent topic seems related to American politics, but the child topic seems related to tech companies

5 Discussion

This chapter discusses the results from the study as well as the method used to derive them. Sections 5.1 and 5.2 contain reflections of the results and what they indicate. Section 5.3 discusses the method and addresses some limitations of it. The ethical and societal aspects related to the work are brought up in Section 5.4.

5.1 Topic Hierarchy Results

Two topic models were created in this work, one hLDA model and one constrained hLDA model. From observing the hierarchy structure and topics of the two, one can conclude that they are similar. The total number of topics in the models were 244 and 220, respectively. The number of topics on each of the levels was also similar, the hLDA model had 28 topics on the second level and 215 on the third while the constrained hLDA model has 25 topics on the second level and 194 on the third.

The two models are not only structurally similar, the words in the topics are also comparable. The predefined topics generated from the FP-tree resulted in topics that closely resemble the topics from the hLDA hierarchy. This can be seen in Figures 4.1 and 4.3, when comparing the predefined topics in blue from the constrained hLDA model with the regular nodes on the same level in the hLDA model.

The two models were generated from the same corpus and run for the same number of iterations using the same hyperparameter settings. Structurally, one would therefore expect similar models as a result. One thing that might cause the structure to differ is the fact that hLDA is a non-deterministic model containing some random variation. Another reason is of course the predefined topics and the difference in the path sampling probability introduced in constrained hLDA. However, since the predefined topics were extracted from the corpus by grouping words with high correlation to each other, one might expect that these words would be seen in a topic together in the hLDA model too. This could explain the similarity between the two models.

The resemblance between the models can also be seen when comparing Figures 4.2 and 4.4 that show the path and level assignments of the same article in the two models. In the constrained hLDA model, the food topic is constrained, yet the child topic is very similar in both models.

Although similar topics exist in both models, it is possible that the constrained hLDA model is able to derive them faster than a hLDA model, as the probability of assigning the highly correlated words to the same topics is greater. Proving this hypothesis would however require a deeper analysis of the sampling process.

How the predefined topics are chosen is of course also a large factor in the resulting topic hierarchy. For example, when several almost identical topics were included in the model, only one of the topics would be assigned articles related to those words, while the other topics would become related to something else.

How many words make up the predefined topics could also affect the hierarchy, as all words must occur in an article for it to be matched to a predefined topic. A large number of words thus requires more specific articles.

There are several numeric parameters in constrained hLDA to consider. These are described in detail in Section 2.9. In summary, one has to determine which words to use when matching documents to the predefined topics. This involves selecting minimum and maximum support thresholds, i.e. at least and at most how frequently the words may appear across all articles to be considered. It also involves setting a correlation threshold, determining at least how correlated words must be to be grouped together. In this work, different values were tested until reasonable groups of words were derived.

The parameter η' in Equation 2.16 was set to a very small value, which means documents that match a predefined topic will almost certainly be assigned there. This can be seen in Equation 2.16 and 2.17. Using a larger η' would relax the constraint and allow articles to be assigned to other topics although they match a constrained one. This could have an impact on the topics generated in the model. For example, a topic with similar words to those in a predefined topic may form outside of the predefined one. This may or may not be desirable depending on if two generated topics in fact cover different subjects, only with ambiguous words for example.

One should also keep in mind that evaluating topic hierarchies is highly subjective. What one person considers a reasonable topic or hierarchy might not make sense to someone else. The topic models derived in this work were generated as a result of parameter experimentation by one person; the author of this work. It is thus biased by my subjective opinion of what the topics should look like. I do think that both models created in this work are good in that they have captured a wide range of topics on the level below the root, as well as number of subtopics on the level below that seem related to the parent topic while also specifying subtopics.

5.2 Word Frequency Similarity Results

Both models had a lowest mean word frequency similarity of around 0.5 %, and a highest of around 20 %. The average mean word frequency similarity for the models was 7.2 % and 6.2 % for the hLDA and constrained hLDA models, respectively. In other words, the two models have similar word frequency similarities between the parent and child topics in the hierarchies.

The topic pairs with the lowest mean word frequency similarity seem either too unrelated or just hard to interpret. For the hLDA model, the parent relates to fitness, but the child is more related to politics (Figure 4.5). In the constrained model, it is hard to determine what the any of the topics are (Figure 4.9). Neither of the topic pairs with the lowest word frequency similarity seem good.

Theoretically, it makes sense that a low word frequency similarity means that the topics do not have much in common and therefore may be unrelated to each other.

The topic pairs with the highest mean word frequency similarities, shown in Figure 4.6 and Figure 4.10, have a clearer relation between parent and child topics. However, the child topics do not seem to define any specific subtopics of their parents. This is also theoretically

logical as a high mean word frequency similarity means the parent and child have a lot in common. Thus, the child may not further specify a clear subtopic.

It seems like mean word frequency similarity can be used as a measure of how a child topic relates to its parent. If the similarity score is high, the child is very similar to its parent and may not add any further specificity to the parent topic. If the similarity score is low, chances are that the parent and child topic should not be related in the topic hierarchy because they are too dissimilar. In a good topic pair, i.e. where the child is related to the parent but also specifies a subtopic, the mean word frequency similarity is somewhere in between. It is not too high, and not too low.

However, determining precise upper and lower limits for a "good" word frequency similarity score may be challenging. No such limits were determined in this work and therefore remains as future work.

5.3 Method

This section considers the method used in this work with regard to why it was chosen and what possible limitations of it are. It also includes a discussion about the replicability, reliability and validity of the study. Further, the sources used in the work are discussed.

Method Choice

The goal of the thesis was to investigate whether the novel topic hierarchy evaluation metric word frequency similarity could be used to draw conclusions about topic hierarchies. The metric requires the model to represent topics hierarchically as distributions of words. A popular model that fulfills those requirements is hLDA, which is why it was chosen. It was also found interesting to evaluate its extension, constrained hLDA, to see if any significant differences could be observed with regard to the word frequency similarity metric.

The Models

Hierarchical topic models such as hLDA and constrained hLDA, where the levels are based on abstraction, are good at grouping related articles on a high level while also capturing more specific subtopics in the levels below. This allows for a better understanding of how topics relate to each other compared to if the structure had been flat.

The non-parametric nature of the hLDA model allows it to adapt to the data, without making assumptions about the latent topics. Constrained hLDA is a semi-supervised approach, where assumptions are made about the topics with the predefined topics. However, the model is still able to create new topics if the data requires it. This flexibility is one of the biggest strengths of the hLDA models.

Another advantage of the hLDA models is that they are able to assign general words, that do not belong in any specific topics, to the root node. This means that less weight can be put on the data preprocessing step and removing words that do not provide context. However, it is still a good idea to remove frequently occurring words known not to provide context, as having less words to process will speed up the modeling process.

One weakness of the hLDA models, as they are implemented in this work, is that each article is assigned only one path in the tree. Each article is thus a mixture of the topics in one path, but those topics are generally very similar and only become increasingly specific closer to the leaf. It is thus possible that the models fail to capture mixtures of diverse topics in articles. This is something that the regular LDA model potentially handles better, as any mixture of topics is possible.

For example, if an article discusses a movie about fitness, one could argue that it is a mixture of topics related to movies and fitness. The hLDA model would perhaps assign the article to a parent topic related to movies and a child topic relating to fitness. However, this

means that the parent topic needs child topics that do not really relate to them, just to cover all possible combinations of topic mixtures. It may have been better if the article could be modeled as a mixture of the topics in a movie path and a fitness path in the tree. Then the tree structure could be kept more simple and minimal, and thus more interpretable.

Possible Method Improvements

There are a few improvements that could be made to the method used in this work. One is that instead of performing test runs to determine how many sampling iterations are required, a more sophisticated method could be used for determining convergence, such as autocorrelation described in Section 2.8. This would require less time and computing resources, but in the end it was not prioritized in this work, because the amount of time it would take to implement was deemed greater than the amount of time it would save. It also does not affect the results as long as a sufficient number of iterations are performed, allowing the Markov chain to converge. However, if the work was to be used to create more models with different data, it may be worth taking the time to implement such a convergence evaluation method.

Using a convergence evaluation method such as autocorrelation is not only less time-consuming, but of course also a more precise way to determine the number of iterations to perform. It decreases the risk of performing too few iterations and therefore deriving results from a non-converged Markov chain.

As previously described, the constrained topics in the constrained hLDA model were extracted from the corpus. Thus, they are based on statistical relationships in the data. However, it is possible use other methods for defining the constrained topics. For example, one option is to define them manually. Regardless of what method is used, it is important to keep in mind that the words used together in the constrained topics, also have to be present together in the corpus. If not, no articles will be matched the constrained topics.

Another possible improvement is sampling the hyperparameters between the Gibbs sampling iterations as described in Section 2.8. This could result in finding hyperparameters that yield better models. It was not done in this work as a simplification and because it could potentially have caused the hyperparameters of the two models to differ, making it harder to compare them.

Replicability, Reliability and Validity

The study described in this work could be repeated by following the steps described in this report and the referenced work. However, the data used was provided by Opera and is not publicly available. If another dataset was used, one could expect different topic hierarchies and thus different word frequency similarity scores.

Even if the study was repeated using the same dataset, one would expect slightly different results due to how the Gibbs sampling process is defined. As described in Section 2.8, the Markov chain may approach different modes depending on its start state and the random seed.

In order to approach the same mode and achieve the same results every time, one would have to use the same random seed and the same starting state. However, in the implementation used in this work, a different random seed and a different start state would be used each run. The motivation for this is that one wants to find an as good start state as possible to increase the chances of approximating a good mode. Recall that in this work, 100 different start states were evaluated before running the sampler using the best one.

The measure of word frequency similarity could of course be used in the same way to evaluate any hLDA topic hierarchies.

Source Critisism

The method used in this thesis is heavily based on the methods used by Blei et. al [4, 8] and Wang et al. [6]. In turn, Wang et al. [6] based their work on that of Blei et. al. [4, 8]. All models used in this and their work is based on the same implementation of hLDA. While this can be a good thing in terms of comparability, it could also be a weakness as a large part of the work stems from only a few authors. However, the authors of the original hLDA model are well-known in the machine learning and Bayesian statistics fields. According to Google Scholar, work by David Blei has been cited 81,609 times, and his co-author in both the original LDA and hLDA paper, Michael I. Jordan, has been cited 177,180 times.

A majority of the other sources used in this work were found through the library resources available at Linköping University. They are peer-reviewed and published in respectable Journals or Conferences such as *Journal of the ACM* and *IEEE 7th International Conference on Intelligent Computing and Information Systems (ICICIS)*. A book on Monte Carlo statistical methods published by the well-known international publisher Springer was also used as reference.

5.4 The Work in a Wider Context

Content mining is a powerful means of gathering large amounts of data from different sources on the Internet. The data can then be processed using models such as hLDA to extract otherwise latent information.

Wang et. al [6] used constrained hLDA to process data gathered from microblogs through content mining. Kaveri and Maheswari [27] used hLDA to process data gathered from the social media web page Twitter in order to discover health-related topics. In this thesis work, the articles used to create topic hierarchies were gathered from a wide range of sources on the Internet.

Most of the information on the Internet is of course available to anyone, but it is possible that the people who posted it did not intend or account for it to be used in this way. Some content sources are more sensitive than others when it comes to this. One example is social media web pages where people sometimes post very personal information about themselves.

It is important to remember that with the power of large-scale content mining comes the responsibility of handling the data in an ethical way. One needs to make sure no individuals are intentionally or unintentionally exposed or harmed during the data handling process, or when conclusions drawn from the data are presented.

6 Conclusion

The purpose of this work was to create and evaluate a hLDA and a constrained hLDA model with regards to the novel evaluation metric word frequency similarity. Both models have been successfully created and evaluated.

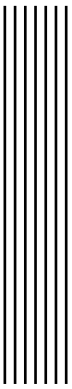
To answer to the first research question; a topic hierarchy, where the topics are represented as distributions of words, can be evaluated based on the word frequency similarity between parent and child topics using the novel metric introduced in this work called *word frequency similarity*. It is a measure of how similar the word frequencies in the parent and child topics are, and therefore indicates how much the two topics have in common. Typically, one would expect a parent-child topic pair to have some things in common, otherwise they should not be related. However, they should not be too similar, because then the child does not contribute with any further specification of a subtopic and the hierarchical relationship becomes redundant. Word frequency similarity is thus a good way to evaluate the parent-child topic relationships in a topic hierarchy.

The answer to the second research question, of how a topic hierarchy created using hLDA compares to one created using constrained hLDA with regard to the word frequency similarity between parent and child topics, is that the results were very similar. Both models had comparable lowest, highest as well as average mean word frequency similarities. The conclusion is therefore that none of the models outperformed the other with regard to mean word frequency similarity.

To further elaborate on this work, one could evaluate hLDA models created from different datasets with regard to word frequency similarity. Perhaps more precise upper and lower limits for what a "good" word frequency similarity score is could be defined.

One could also include hyperparameter sampling, which could lead to more structural differences in the models, and evaluate whether either of the models seems to outperform the other under those circumstances.

The theory of whether predefined topics cause the Markov chain to converge faster could also be explored. It would also be interesting to investigate whether it is possible to use word frequency similarity as a convergence measure, or even to incorporate it into the sampling process itself.



Bibliography

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* (2003), pp. 993–1022. ISSN: 1532-4435.

[2] R. Yasotha and E.Y.A. Charles. "Automated text document categorization." In: *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS), Intelligent Computing and Information Systems (ICICIS), 2015 IEEE Seventh International Conference on* (2015), pp. 522–528. ISSN: 978-1-5090-1949-6. DOI: 10.1109/IntelCIS.2015.7397271.

[3] Zhao Weizhong, James J. Chen, Roger Perkins, Liu Zhichao, Ge Weigong, Ding Yijun, and Zou Wen. "A heuristic approach to determine an appropriate number of topics in topic modeling." In: *BMC Bioinformatics* 16 (2015), pp. 1–10. ISSN: 14712105.

[4] David M. Blei, Thomas L. Griffiths, Jordan Michael I., and Tenenbaum Joshua B. "Hierarchical topic models and the nested Chinese restaurant process." In: (2004). DOI: 10.1.1.131.3884.

[5] Lin Liu, Lin Tang, Libo He, Wei Zhou, and Shaowen Yao. "An overview of hierarchical topic modeling". In: *Proceedings - 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2016*. Vol. 1. Institute of Electrical and Electronics Engineers Inc., Dec. 2016, pp. 391–394. ISBN: 9781509007684. DOI: 10.1109/IHMSC.2016.101.

[6] Wei Wang, Hua Xu, Weiwei Yang, and Xiaoqiu Huang. "Constrained-hLDA for Topic Discovery in Chinese Microblogs". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Vincent S. Tseng, Tu Bao Ho, Zhi-Hua Zhou, Arbee L. P. Chen, and Hung-Yu Kao. Springer International Publishing, 2014, pp. 608–619. ISBN: 978-3-319-06605-9.

[7] Yizhou Sun, Hongbo Deng, and Jiawei Han. "Probabilistic Models for Text Mining". In: *Mining Text Data*. Ed. by Charu C. Aggarwal and ChengXiang Zhai. Springer US, 2012, pp. 259–295. ISBN: 978-1-4614-3223-4. DOI: 10.1007/978-1-4614-3223-4_8.

[8] David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. "The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies". In: *Journal of the ACM* 57.2 (Jan. 2010). DOI: 10.1145/1667053.1667056.

[9] Thomas Hofmann. "Probabilistic Latent Semantic Indexing". In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '99. Association for Computing Machinery, 1999, pp. 50–57. ISBN: 1581130961. DOI: 10.1145/312624.312649.

[10] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. "Indexing by latent semantic analysis". In: *Journal of the American Society for Information Science* 41.6 (1990), pp. 391–407. DOI: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[11] Matthew D. Hoffman, David M. Blei, and Francis Bach. "Online Learning for Latent Dirichlet Allocation". In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*. NIPS'10. Curran Associates Inc., 2010, pp. 856–864.

[12] James M. Dickey. "Multiple Hypergeometric Functions: Probabilistic Interpretations and Statistical Uses". In: *Source: Journal of the American Statistical Association* 78.383 (1983), pp. 628–637. DOI: 10.2307/2288131.

[13] Christian P. Robert and George Casella. *Monte Carlo statistical methods.* Springer texts in statistics. Springer, 1999. ISBN: 038798707X.

[14] Tim Salimans, Diederik P. Kingma, and Max Welling. "Markov Chain Monte Carlo and Variational Inference: Bridging the Gap". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning* 37 (2015), pp. 1218–1226.

[15] Hilah Geva, Gal Oestreicher-Singer, and Maytal Saar-Tsechansky. "Using Retweets When Shaping our Online Persona: Topic Modeling Approach". In: *MIS Quarterly: Management Information Systems* 43.2 (June 2019), pp. 501–524. DOI: 10.25300/MISQ/2019/14346.

[16] Thomas R. L. Griffiths and Mark Steyvers. "Finding scientific topics." In: *Proceedings of the National Academy of Sciences of the United States of America* 101 Suppl 1 (2004), pp. 5228–35.

[17] Juan Cao, Tian Xia, Jintao Li, Yongdong Zhang, and Sheng Tang. "A density-based method for adaptive LDA model selection." In: *Neurocomputing* 72.7 (2009), pp. 1775–1781. ISSN: 0925-2312.

[18] R. Arun, V. Suresh, C.E.V. Madhavan, and M.N.N. Murthy. "On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations". In: *Lecture Notes in Computer Science*. 6118. 2010, pp. 391–402. ISBN: 978-3-642-13656-6.

[19] Romain Deveaud, Eric Sanjuan, and Patrice Bellot. "Accurate and Effective Latent Concept Modeling for Ad Hoc Information Retrieval." In: (2014). DOI: 10.3166/DN.17.1.61-84.

[20] David J. Aldous. "Exchangeability and related topics". In: *École d'Été de Probabilités de Saint-Flour XIII — 1983*. Springer Berlin Heidelberg, 1985, pp. 1–198. ISBN: 978-3-540-39316-0.

[21] Jianfei Chen, Jun Zhu, Jie Lu, and Shixia Liu. "Scalable Inference for Nested Chinese Restaurant Process Topic Models". In: (2017). DOI: 10.475/123.

[22] Liu Jun S. "The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem." In: *Journal of the American Statistical Association* 89.427 (1994), p. 958. ISSN: 01621459.

[23] T. L. Griffiths and M. Steyvers. "Finding scientific topics." In: *Proceedings- National Academy of Sciences USA* SUPP/1 (2004), p. 5228. ISSN: 0027-8424.

[24] Pascal Sebah and Xavier Gourdon. "Introduction to the gamma function". In: *American Journal of Scientific Research* (2002), pp. 2–18.

[25] David Andrzejewski and Xiaojin Zhu. “Latent Dirichlet Allocation with Topic-in-Set Knowledge”. In: *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing* (Jan. 2009). DOI: 10.3115/1621829.1621835.

[26] J. Han, J. Pei, Y. Yin, and R. Mao. “Mining Frequent Patterns without Candidate Generation A Frequent-Pattern Tree Approach.” In: *Data Mining and Knowledge Discovery* 1 (2004), p. 53. ISSN: 1384-5810.

[27] V. Vijeya Kaveri and V. Maheswari. “A framework for recommending health-related topics based on topic modeling in conversational data (Twitter)”. In: *Cluster Computing* 22 (2019), pp. 10963–10968. DOI: 10.1007/s10586-017-1263-z.