LUISS

Degree program in Management and Computer Science

Course of Data Analysis for Business

Latent Dirichlet Allocation for Topic Modeling

Supervisor
Prof. Alessia Caponera

Candidate
Ginevra Augello - 288401

Academic Year 2024/2025

To my mother

Contents

Introduction and Overview

In this thesis, we have addressed one of the key problems in natural language processing: discovering topics from a set of unordered documents, also known as *topic modeling*. The method implemented to tackle this problem was Latent Dirichlet Allocation (LDA), a generative probabilistic model designed to uncover the latent thematic structure of text corpora.

To evaluate the effectiveness of LDA, we applied it to a dataset comprising four novels by Jane Austen: *Pride and Prejudice*, *Emma*, *Sense and Sensibility*, and *Mansfield Park*. The purpose of this application was to identify and model the core topics of the novels. Particular attention was placed on the two different inference methods used to approximate the intractable posterior distribution of LDA: Gibbs sampling and Variational Expectation Maximization (VEM). The former is a sampling-based MCMC approach, while the latter treats inference as an optimization problem by leveraging variational bounds.

The two methods produced different, yet equally insightful results. Gibbs sampling yielded more interpretable, distinct, and semantically coherent topics that captured the core themes explored by Jane Austen in her works. In contrast, VEM produced less coherent and more redundant topics. However, it proved capable of capturing subtler patterns and thematic nuances, revealing more complex connections within the text.

In Chapter 1, we review the fundamental concepts of statistical inference, highlighting the differences between the Frequentist and Bayesian paradigms, and introducing the key probability distributions used in LDA. In Chapter 2, we provide a theoretical analysis of Latent Dirichlet Allocation, focusing on its generative process, hierarchical Bayesian structure, and core assumptions such as exchangeability and conditional independence. We also examine Gibbs sampling and VEM in detail. Lastly, in Chapter 3, we describe the implementation of these methodologies on our dataset and evaluate the performance and interpretability of the results obtained with both inference techniques.

Chapter 1

Background

Statistical inference can be defined as a logical framework that can be used to test opinions against and draw conclusions from data. The result of this process is the formalization of beliefs into models of probability. In other words, it is an inductive process, through which we learn about the general characteristics of a population by analyzing a subset of its members.

Such a process of inference can be carried out by following two different and diverging approaches: Frequentist and Bayesian, which employ different interpretations of both probability and inference.

The objective of this chapter is to provide an overview of the main characteristics and differences between Frequentist and Bayesian inference. In addition, further preliminary information will be provided to enhance the understanding of the following chapters.

1.1 The Frequentist approach

In the frequentist paradigm, probability is considered synonymous with long-run frequency: if an outcome has occurred g times in m similar events where m is a very large number then its probability is approximately $\frac{g}{m}$. The reason why m is a large number is because the underlying assumption is that the event can be repeated for a theoretically infinite series of experimental repetitions. For example, if we flip fair a coin, the probability of getting tails in the long run is equal to 0.5, meaning that if we were to make an infinite number of trials we would expect to get tails from a coin flip 50% of the time. However, if we flip a coin only a few times, we could observe an entirely different distribution: in a series of 10 trials, we could get tails 8 times. From a Frequentist perspective, this oddity results from sample variation: due to randomness of data, the sample we pick may not be representative of the population of repeated trials. In simple terms, any sample that is collected and analyzed is just one of the many hypothetical samples and randomness is what determines uncertainty.

Using this interpretation of probability as a basis, the Frequentist approach to statistical inference is centered on the idea that probabilities are assigned to data, whereas parameters are a fixed, unvarying quantity, albeit an unknown one. This implies that probability statements about the parameters of a statistical process are not allowed, something which becomes very apparent when considering confidence intervals: the fact that a 95% confidence interval for the normal mean μ is [-0.5, 1.0] does not imply that there is a 95% probability that μ is in that interval. In fact, it means that if the experiment were to be repeated for an infinite number of times, for different samples, constructing the confidence interval each time, then in 95% of the cases the true mean μ would lie in the 95% confidence interval. As a result, Frequentist inference is focused on determining the value of an unknown parameter for which the observed data is most likely.

Ultimately, Frequentist statistics relies on hypothesis testing, which is based on calculating the probability of obtaining statistic results as extreme as, or more extreme than the one actually obtained, assuming that a certain hypothesis - the so-called *null hypothesis* - is true. In other words, hypothesis testing is based on an evaluation of how far the observed results are from those that we would expect under the null hypothesis. The more extreme the results, the less likely it is that they could have occurred under the null hypothesis by random chance.

In this context, a hypothesis is a statement about the value of a specific parameter, whereas the probability on which hypothesis testing is based is referred to as P value, which - as the definition suggests - is partly determined by data that has never been observed.

The P value is used in order to test the validity of a hypothesis: by convention, a P value smaller than 0.05 is considered significant, meaning that there is a small probability of observing a greater or equal outcome under the null hypothesis, which is therefore rejected. In short, the P value is an indicator of the strength of evidence against the null hypothesis.

1.2 The Bayesian approach

According to the Bayesian school of thought, probability is an expression of degrees of belief regarding the truthfulness and correctness of a proposition, which - albeit based on the objective data available - is inherently subjective in nature. In other terms, Bayesian probability is a quantification of an individual's beliefs, which can be updated in light of new data. This updating process is carried out through the Bayes' theorem.

Let us bring back our coin flip example, but with a twist: instead of having a single fair coin, we have two coins, one which is fair and another which is loaded. The loaded coin has tails on both sides. Clearly, if we pick a coin at random and by flipping it we get heads, we know for sure that the coin was fair. But what if we get tails instead? Intuitively, it is more plausible that the coin that we have flipped is the loaded one, but we cannot say this for certain. In order to quantify this uncertainty, we employ the Bayes' theorem, also known as the inverse probability theorem. This is a mathematical formula that expresses the probability of a cause given its effect, by reversing conditional probabilities

Consider two events: A and B. The conditional probability of A given B (i.e. the probability of A happening given that B has happened) is formally defined as follows:

$$P(A|B) \ = \frac{P(A \cap B)}{P(B)} \, , with \ P(B) \neq 0.$$

The question is how to get from $P(A|B)$ to $P(B|A)$. In other words, how is it possible to invert this conditional probability? Firstly, let us recall the multiplication rule:

$$P(A \cap B) \ = \ P(A|B) \ P(B).$$

Because $A \cap B$ is symmetric:

$$P(A \cap B) \ = P(B|A) \ P(A).$$

Which means that:

$$P(A|B) \ P(B) \ = \ P(B|A) \ P(A).$$

Hence, we can express $P(B|A)$ and therefore write the formula of the Bayes' theorem as follows:

$$P(B|A) \ = \ \frac{P(A|B) \ P(B)}{P(A)}.$$

Let us get back to our previous example, providing a formal definition of the problem that we have described:

$$P(L|T) \;=\; \frac{P(T|L)\;P(L)}{P(T)},$$

where

- $P(L|T)$ is the probability of the coin being loaded (L), given that we got tails (T).
- $P(T|L)\;P(L)$ is the probability of getting tails, given that the coin is loaded, multiplied by the probability the coin being loaded.
- $P(T)$ is the probability of getting tails.

The term on the left side of the equation is a measure of how strongly the proposition *I got tails from flipping a coin* implies the proposition *The coin that I have flipped is loaded*, by combining probabilities related to our knowledge of the loaded coin to the probability of getting tails irrespective of such knowledge. This is the starting point of Bayesian inference. As Jerome Cornfield noted in his article on the Bayes theorem *it is clear that it is not possible to think about learning from experience and acting on it without coming to terms with Bayes theorem.*

The Bayesian approach to inference is based on an intuitive reasoning: whenever we examine a set of observations, we try to draw up some conclusions, which are logically consistent with our observations, albeit with different degrees of reasonableness. Our conclusions will be uncertain and influenced by our beliefs. Then, as we get more information, we are able to update our prior beliefs in light of new evidence, obtaining posterior beliefs that we can use to test our hypotheses.

Let us revisit our coin flip problem: we have a single coin which we assume is fair and we flip it once. We get tails. This, by itself, is not a significant result and it does not challenge our belief of fairness either, but what if we get tails over and over again? If we keep getting tails consistently, after a number of flips, we might start believing that the coin is not fair at all: by accumulating evidence we are able to update our prior belief. The question is: how do we express this process in a formal way?

Generally, numerical values associated with the characteristics of a population are expressed in terms of a parameter (θ), while numerical descriptions of a subset of that population make up a dataset y. In the absence of a dataset, both numerical values and descriptions are uncertain, since - contrary to the Frequentist approach - both the observed data y and the unknown parameter θ are considered to be random variables, which possess a probability distribution.

As soon as a dataset y is obtained, it is possible to decrease the level of uncertainty associated with the population characteristics. To quantify this change in uncertainty, Bayesian inference uses the Bayes theorem to estimate the probability distribution of unknown parameters in light of the observed data.

Let Y be the sample space (i.e. the set of all possible datasets) and let Θ be the parameter space (i.e. the set of all possible parameter values):

- For each numerical value $\theta \in \Theta$, the prior distribution $p(\theta)$ represents our prior beliefs - independent from the observed data - that θ represents the true population characteristics.
- For each $\theta \in \Theta$ and $y \in Y$, the likelihood $p(y|\theta)$ describes the probability of generating a specific sample of data y if θ were to be true.
- For each numerical value of $\theta \in \Theta$, the posterior distribution $p(\theta|y)$ is the expression of our uncertainty over the value of θ, describing the strength of our belief that θ is the true value, once we have observed dataset y. In short, the posterior distribution reflects the process of updating our original beliefs in light of new data.

$$p(\theta|y) \;=\; \frac{p(y|\theta)\;p(\theta)}{p(y)}. \tag{1.1}$$

The denominator $p(y)$ represents the probability of obtaining a specific dataset y if we assume a particular model and prior. In other words, it is the probability distribution of y before the dataset is actually observed. This component is much more complex than it might seem first-hand: it is a normalizing factor that ensures that $p(\theta|y)$ is a valid probability distribution.

The necessity of $p(y)$ arises from the fact that the likelihood, which Bayesian inference tries to invert to obtain the posterior distribution $p(\theta|y)$, is not by itself a valid probability distribution. In fact, the likelihood does not meet both criteria for validity, these being that all values of the probability distribution must be real and non-negative, and that the sum (for discrete random variables) or integral (for continuous random variables) must be equal to 1. Specifically, the likelihood fails to meet the second criterion. On the other hand, $p(y)$ is a valid probability distribution, which makes it so that the posterior distribution sums or integrates to 1, meeting both criteria of validity.

Depending on whether we are dealing with discrete or continuous variables, $p(y)$ can assume two possible forms. Respectively:

$$p(y) \;=\; \sum_{\theta \in \Theta} p(y|\theta)\; p(\theta),$$

$$p(y) \;=\; \int_{\Theta} p(y|\theta)\; p(\theta)\, d\theta.$$

In principle, such computations are relatively uncomplicated, but only if we assume that our model involves a single parameter, something which is quite unlikely in real life applications. The more the parameters in the model are, the more computations become difficult, to the point that they are practically impossible to calculate, unless the model is very simple.

When dealing with more complex models, there are two possible approaches that can be employed: either use a conjugate prior to the likelihood, or sample from the posterior.

Conjugate priors are probability distributions that, when combined with a likelihood function, result in a posterior probability distribution, which belongs to the same probability distribution family as the prior. They are used to derive a mathematically convenient, short-form expression of the posterior distribution.

Although conjugate priors are very useful, they can also be too restrictive. However, by abandoning the goal of exact calculation of the posterior, it is possible employ another strategy to infer the nature of the posterior: use computational methods based on sampling from the posterior itself. Specifically, by using enough samples, we are able to make some accurate, albeit approximate estimates about the probability distribution of the posterior. Such an approach is quite complex and computationally expensive, but it was rendered possible thanks to the combination of modern technologies with advanced numerical techniques, the most important of which is the Markov chain Monte Carlo (MCMC).

Some of the most important MCMC algorithms for posterior sampling include Random Walk Metropolis, Gibbs sampling and Hamiltonian Monte Carlo.

1.3 The false dilemma

The Frequentist approach and the Bayesian one display significantly different interpretations of the concept of probability. However, from a practical standpoint, they are not always inherently different: when large samples are involved, the influence of the Bayesian prior decreases, leading to conclusions that can easily be reconciled with those resulting from a Frequentist analysis.

Even from a purely theoretical perspective, where one of the main sources of criticism of the Bayesian approach is represented by its subjectivity, the actual differences are liable: both approaches, as a matter of fact, display a certain degree of subjectivity. Specifically, in both cases it is necessary to make a subjective judgement to obtain knowledge from data. This is very apparent in the case of the Bayesian approach, which explicitly employs a subjective prior. In the case of the Frequentist approach, it can be argued that the thresholds used to decide whether to reject the null hypothesis or not are quite arbitrary and subjective.

Ultimately, each approach presents its own set of advantages and disadvantages depending on the specific case-by-case application. Hence, depending on the circumstances, one method may be preferable to the other. For example, the Bayesian approach is of pivotal importance in the context of machine learning algorithms. Therefore, in the analysis of data, it becomes an extremely useful and powerful tool.

1.4 De Finetti's theorem

It is worth noting that the subjective interpretation of probability embraced by Bayesian statistics has, in fact, a theoretical validation: *De Finetti's representation theorem*. Notably, this theorem motivates the use of prior distributions, which constitutes the core of Bayesian inference. However, in order to discuss De Finetti's theorem, it is necessary to introduce the concept of *exchangeability* first.

A finite set of random variables $\{X_1, \dots, X_n\}$ is said to be exchangeable if the joint distribution is invariant to permutation. Thus, for all $x_1, \dots, x_n$ and all permutations π of the integers from 1 to n, the following equation holds true:

$$p(x_1, \dots, x_n) = p(x_{\pi(1)}, \dots, x_{\pi(n)}). \tag{1.2}$$

This property is of utmost importance in the context of predictive inference, since it states that the order in which data points are observed does not affect their joint probability. In other terms, it implies that there exists a symmetry between the future and the past, meaning that past information can be used in order to predict future outcomes.

It is also possible to define *infinite exchangeability*. Specifically, an infinite sequence of random variables $\{X_1, X_2, \dots\}$ is said to be infinitely exchangeable if every finite subsequence is exchangeable. As a result, equation 1.2 holds true for any n and for every permutation π. This is the starting point for De Finetti's representation theorem.

Let $\{X_1, X_2, \dots, X_n, ...\}$ be an infinitely exchangeable sequence of variables. Then, the joint distribution of the sequence can be expressed as a mixture of conditionally independent and identically distributed random variables. Formally, there exists a distribution function P on Θ such that

$$p(x_1, x_2, \dots, x_n) = \int_\Theta \prod_{i=1}^{n} p(x_i \mid \theta)\ P(d\theta).$$

In this context, θ is a latent random parameter and P is a distribution for θ. The joint distribution $p(x_1, x_2, \dots, x_n)$ behaves as if the random parameter θ were randomly drawn from P, and then all x_i were generated independent and identically distributed, conditioned on θ.

The importance of this result lies in its interpretation. Specifically, De Finetti's theorem states that, if the data is exchangeable, a parameter θ must exist. In addition, the likelihood $p(x_i \mid \theta)$ and the distribution P must exist as well. Lastly, all the above quantities must exist so that $X_1, X_2, \dots, X_n, ...$ are conditionally independent. In short, the theorem justifies why parameters should be used and why priors should be put on parameters.

1.5 Some useful distributions

The following sections are meant to provide an overview of the main distributions that will be used in the next chapters, in order to enhance their understanding.

1.5.1 Poisson distribution

The Poisson distribution is a distribution used to model the probability of a certain number of events occurring within a fixed time. It is an approximation of the binomial distribution, provided that the number of trials n is large, the probability of success p is small, and the product np is constant.

Let us consider a random variable X. Then, the probability distribution such that $X \sim Poisson(\lambda)$ is defined by the following probability formula:

$$P(X = x) = e^{-\lambda}\frac{\lambda^x}{x!},$$

where $\lambda > 0$ is a constant expressing both the mean and the variance of the underlying random variable X.

To better understand the concept, consider the following example: let the random variable X denote the number of phone calls received per hour in a call center, which - on average - receives 5 calls per hour (thus $\lambda = 5$). Then, the probability of receiving exactly 3 phone calls in an hour can be expressed as follows:

$$P(X = 3) = e^{-5}\frac{5^3}{3!} \approx 0.14\ .$$

1.5.2 Dirichlet distribution

The Dirichlet distribution is a multivariate continuous probability distribution, used to model the randomness of probability mass functions (PMFs).

Let us consider the following example: we are analyzing multiple authors, who write about three topics, these being history, science and technology. Clearly, the focus put on each topic may vary, depending on the author. For instance, one author might write 70% about science, 20% about technology, and 10% about history, while another author might write 40% about science, 40% about technology, and 20% about history. These differences in focus can be represented as PMFs:

$$p_{Author1} = \{Science : 0.7,\ Technology : 0.2,\ History : 0.1\},$$

$$p_{Author2} = \{Science : 0.4,\ Technology : 0.4,\ History : 0.2\}.$$

Now, let us assume that we have a set of 100 authors, each with a PMF that reflects their respective topic focus. Of course, due to different preferences and tendencies, such PMFs may vary significantly. In order to simulate this randomness, we use a Dirichlet distribution with a concentration parameter $\alpha = [\alpha_{Science},\ \alpha_{Technology},\ \alpha_{History}]$.

The values of α will determine the uniformity or variability of the generated PMFs. Hence, if $\alpha_{Science} = \alpha_{Technology} = \alpha_{History} = 1$ we will observe a high variety of topics, while if $\alpha_{Science} = 10$ and $\alpha_{Technology} = \alpha_{History} = 1$, then we expect greater uniformity, with science being the dominant topic.

In order to provide a formal expression of the Dirichlet distribution, it is necessary to define the k-simplex and the (k-1)-simplex, this being the probability simplex in which a k-dimensional Dirichlet random variable (i.e. a PMF with k components) lies.

Let $c \in \mathbb{R}$, with $c > 0$. Then, the k-dimensional closed simplex in $\mathbb{R}^k$ is defined as follows:

$$\mathbb{T}_k(c) = \left\{ (x_1, \ldots, x_k)^T : x_i > 0,\ 1 \le i \le k,\ \sum_{i=1}^{k} x_i < c \right\}.$$

On the other hand, the (k-1)-dimensional open simplex in $\mathbb{R}^{k-1}$ is defined as:

$$\mathbb{V}_{k-1}(c) = \left\{ (x_1, \ldots, x_{k-1})^T : x_i > 0,\ 1 \le i \le k-1,\ \sum_{i=1}^{k-1} x_i < c \right\}.$$

Thus, a random vector $\mathbf{X} = (X_1, \ldots, X_k)^T \in \mathbb{T}_k$ is said to have a Dirichlet distribution if the density function of $\mathbf{X}_{-k} = (X_1, \ldots, X_{k-1})^T$ can be expressed as follows:

$$p(x_1, \ldots, x_{k-1} \mid \alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha)} \left(\prod_{i=1}^{k-1} x_i^{\alpha_i - 1} \right) \left(1 - \sum_{i=1}^{k-1} x_i \right)^{\alpha_k - 1},$$

where $\mathbf{x_{-k}} = (x_1, \ldots, x_{k-1}) \in \mathbb{V}_{k-1}$, and $\alpha = (\alpha_1, \ldots, \alpha_k)$ is a k-dimensional vector such that $\alpha_i > 0$ for each i, with $i = 1, \ldots, k$. The Gamma function defined by

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt, \qquad z > 0,$$

is the generalization of the factorial to reals.

Given this definition, it is possible to either write $\mathbf{X} = (X_1, \ldots, X_k)^T \sim Dirichlet_k(\alpha)$ on $\mathbb{T}_k$, or $\mathbf{X}_{-k} = (X_1, \ldots, X_{k-1})^T \sim Dirichlet(\alpha_1, \ldots, \alpha_{k-1} \mid \alpha)$ on $\mathbb{V}_{k-1}$.

The probability density of $\mathbf{X} = (X_1, \ldots, X_k)^T$ can also be expressed in the following way:

$$p(x_1, \ldots, x_k \mid \alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma\alpha} \left(\prod_{i=1}^{k} x_i^{\alpha_i - 1} \right),$$

with $\mathbf{x} = (x_1, \ldots, x_k)^T \in \mathbb{T}_k$.

Lastly, given $\alpha_0 = \sum_{i=1}^{n} a_i$, the mean and variance of the distribution are given by

$$E[X_i] = \frac{\alpha_i}{\alpha_0},$$

$$Var[X_i] = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}.$$

1.5.3 Multinomial distribution

The multinomial distribution deals with events that may have a number of k discrete outcomes per trial. It is a generalization of the binomial distribution, where trials may only have two possible outcomes ($k = 2$).

Consider a vector $\mathbf{X} = (X_1, X_2, \ldots, X_k)$ with k integer elements, a total number of independent trials n, and a probability vector $\mathbf{p} = (p_1, p_2, \ldots, p_k)$, such that $p_i \geq 0$ for $i = 1, \ldots, k$ and $\sum_{i=1}^{k} p_i = 1$. Then, $\mathbf{X}$ is said to have a multinomial distribution of parameters n and $\mathbf{p}$, or, in other words, $\mathbf{X} \sim Multinomial(n, \mathbf{p})$. Its PMF is given by

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k) = \frac{n!}{x_1!\, x_2!\, \ldots\, x_k!}\, p_1^{x_1}\, p_2^{x_2}\, \ldots\, p_k^{x_k},$$

where $\sum_{i=1}^{k} x_i = n$. The mean and variance of the distribution are given by

$$E[X_i] = np_i,$$

$$Var[X_i] = np_i(1 - p_i).$$

To better understand what this entails, consider a simple example, involving the throwing of six-sided die. Specifically, let us assume that the die is fair (thus $p_i = \frac{1}{6}$ for each $i = 1, 2, \ldots, 6$), and that it is rolled $n = 10$ times. To model the probability of observing a certain number of specific outcomes during those n trials, we use the multinomial distribution. For instance, the probability of obtaining 3 rolls of "1", 2 rolls of "6", 4 rolls of "3" and 1 roll of "5" can be computed as follows:

$$P(X_1 = 3, X_2 = 0, X_3 = 4, X_4 = 0, X_5 = 1, X_6 = 2) = \frac{10!}{3!\, 0!\, 4!\, 0!\, 1!\, 2!} \left(\frac{1}{6} \right)^{10} \approx 2.08 \times 10^{-4}.$$

Chapter 2

Latent Dirichlet Allocation

The goal of this chapter is to provide a comprehensive theoretical analysis of the Latent Dirichlet Allocation model in the context of Bayesian topic modeling.

The term *topic modeling* refers to important methods widely adopted in the context of natural language processing for the purpose of discovering topics from unordered documents. Latent Dirichlet Allocation (LDA) is one of the most commonly used topic modeling methods. Specifically, it is a generative probabilistic model for text corpora, which assumes that the words of each document emerge from a mixture of topics. Such topics do not need - and usually are not - known in advance. As a result, we refer to these topics as *latent topics* that are shared by all documents in a collection.

The general idea behind LDA is that documents can be represented as probabilistic distributions over such latent topics. Notably, each document can be associated with a number of topic proportions, which are used to model a corpus as a Dirichlet distribution. This implies that LDA has the capability of handling heterogeneity, allowing data to exhibit multiple latent patterns.

However, before delving deeper into this model, it is necessary to introduce the notation and terminology that will be adopted later on in the chapter.

2.1 Language of text collections: notation and terminology

Given the objective of using Latent Dirichlet Allocation for the purpose of defining a probabilistic model for a corpus, it is necessary to introduce three key terms, which will be employed in the following sections:

- **Word** - Defined as an item belonging to a vocabulary indexed by $\{1, ..., V\}$. It is the basic unit of discrete data, represented as a one-hot encoded vector, whose components are all equal to zero except one, which is equal to one. Formally, this means that the v-th word in the vocabulary can be represented by a V-dimensional vector $w \in \mathbb{R}^V$, where:

$$w^v = 1 \quad \text{and} \quad w^u = 0 \quad \forall u \neq v, \quad \text{where } u, v \in \{1, 2, \ldots, V\}.$$

- **Document** - Defined as a sequence of N words, formally described by the expression $\mathbf{w} = (w_1, w_2, ..., w_N)$.

- **Corpus** - Defined as a collection of M documents. It is possible to express it in a formal way as $D = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_M\}$.

2.2 LDA's generative process

As previously stated and in contrast to standard clustering models, the basic idea behind Latent Dirichlet Allocation is that documents can exhibit multiple topics, which are represented as distributions over words. Given this intuition, LDA assumes that a two-step generative process is employed for the creation of documents:

1. A distribution over the topics is chosen, meaning that a probability is selected for each topic in the document.

2. For each word in the document, a topic is chosen from the distribution, and a word is chosen from the corresponding topic over the vocabulary. In short, the word is selected conditionally to the topic.

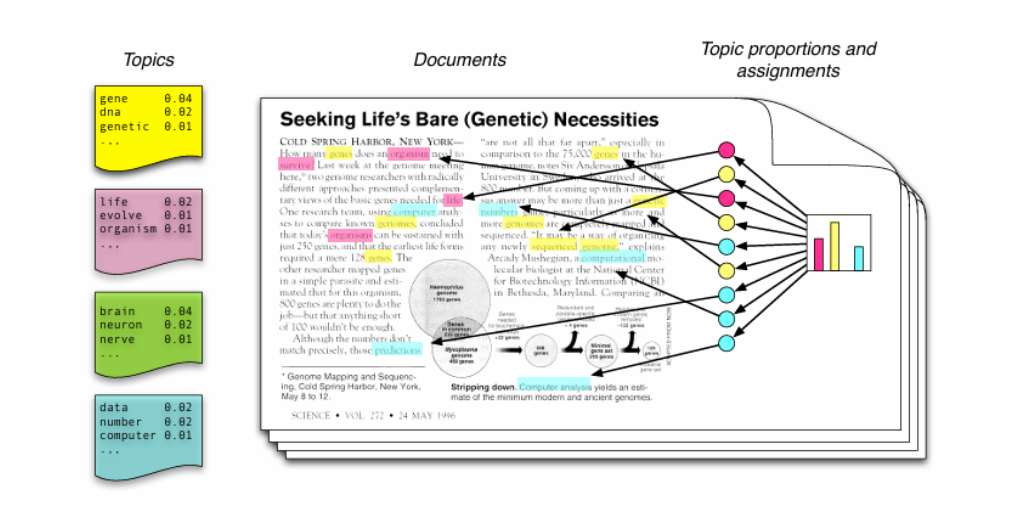

Figure 2.1: Graphical representation of LDA's generative process. The topics are represented by the four boxes on the left side of the picture, while their distribution is exemplified by the histogram on the far right. The circles on the left side of the graph depict the topic assignment for each word, which is then used to select a word over the vocabulary.

It is worth noting that the key concept at the core of the procedure described above is that LDA assumes that the topics have been generated prior to the document itself.

Formally, it is possible to describe LDA's generative process for each document $\mathbf{w}$ in a corpus D as follows:

1. **A distribution for the number of words N is chosen**.

   Normally, the model assumes that $N \sim Poisson(\lambda)$. However, more appropriate distributions may be selected depending on the application context.

   In general, this can be considered a setup step that is not crucial for the understanding of LDA, hence why it is said that the model involves a two-step generative process. In fact, N is an ancillary variable, meaning that it is independent from all data-generating variables.

2. **A topic distribution $\theta \sim Dir(\alpha)$ is chosen, which has a known and fixed dimensionality k**.

   It should be noted that the choice of a Dirichlet distribution for θ is not random. As a matter of fact, it is a convenient distribution over the simplex, being a conjugate prior for the multinomial distribution.

   In this context, θ is a k-dimensional random variable, representing the topic proportions, where k is the number of topics. For example, a document may be 60% about biology and 40% about technology.

   On the other hand, the parameter α is a k-dimensional vector that encodes the document-topic distribution, controlling the sparsity of θ. It represents the prior beliefs on the topic distribution: assuming a symmetric distribution, the higher it is, the greater the expectation that the document will be distributed over multiple topics.

In terms of θ, if each component α_i of α - representing the weight of each topic - is large, θ becomes more uniform. On the contrary, if all α_i values are small, θ will be sparse, meaning that documents will display few dominant topics.

3. For each of the N words w_n in the vocabulary:

   - **A topic z_n is sampled from a $Multinomial(1, \theta)$.**

     This is a step of utmost importance, since it determines which topic will be used to generate the word.

   - **A word w_n is sampled from $p(w_n|z_n, \beta)$, a multinomial distribution conditioned on the topic z_n.**

     In this context, β is a $k \times V$ matrix where the component $\beta_{i,j} = p(w_j|z_i)$ represents the probability that the j-th word in the vocabulary is generated, given the i-th topic. It reflects the word distribution for each topic, meaning that - similarly to α - low values for all $\beta_{i,j}$ imply that a topic is made up by few dominant words, which are strong indicators of the topic itself. The opposite is true for a high values of all $\beta_{i,j}$, meaning that topics will be spread out over multiple words.

This process can be expressed as follows:

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta),$$

where

- $p(\theta|\alpha)$ expresses the prior over the topic proportions θ. It is a probability density such that:

  $$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma\alpha_i} \theta_1^{\alpha_1 - 1} \dots \theta_k^{\alpha_k - 1},$$

  with Γ representing the Gamma function.

- $p(z_n|\theta)$ encapsulates the probability of assigning a topic z_n, given the topic distribution θ.

- $p(w_n|z_n, \beta)$ captures the likelihood that a specific word w_n is extracted given the selected topic z_n and the word-topic probability matrix β.

Considering the generative process defined above, in order to obtain the marginal distribution of a document, it is necessary to integrate over θ and to sum over z, thus obtaining:

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta)p(w_n|z_n, \beta) \right) d\theta \tag{2.1}$$

However - as previously stated - the main goal that is pursued when employing Latent Dirichlet Allocation is to define a probabilistic model for a corpus, not a document. Therefore, it is necessary to take the product of the marginal probabilities of single documents. In this way, it is possible to obtain the probability of the corpus. The latter is thus expressed as follows:

$$p(D|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d)p(w_{dn}|z_{dn}, \beta) \right) d\theta_d.$$

2.3 Bag-of-Words assumption and exchangeability

Standard Latent Dirichlet Allocation relies on the bag-of-words assumption, according to which a document can be represented as an unordered set of words. This simplifying assumption leads to both grammar and syntax being neglected, to focus instead on the frequency of the words that appear in the document. Specifically, a document can be represented as a vector of word occurrences, whose elements represent unique words in the vocabulary. The value associated to each element reflects the frequency of individual words in the document.

It is worth noting that the bag-of-words assumption is extended to the documents themselves, meaning that the order of documents in a corpus is negligible as well.

In the language of probability, the bag-of-words assumption is an assumption of exchangeability. This implies that the random variables in the model will be conditionally independent[1] and identically distributed.

In the context of LDA, the exchangeability of words has important implications related to the topics of a given document. Specifically, since words are generated from topics, it is assumed that topics are exchangeable as well, meaning that the order in which they appear in the document is irrelevant. Notably, they are said to be *infinitely exchangeable*. This aspect is crucial, since it allows the model to infer latent topics without the need to impose a fixed number of topics in advance.

In accordance with De Finetti's theorem, it is possible to express the probability of a sequence of words and topics as follows:

$$p(\mathbf{w}, \mathbf{z}) = \int p(\theta) \left(\prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n) \right) d\theta.$$

2.4 LDA as a hierarchical model

In the context of Bayesian modeling, the structure of LDA can be traced back to a class of models known as *hierarchical models*, which are characterized by a hierarchical structure of their parameters. Notably, these models describe data generation as a stepwise process, involving multiple interdependent levels.

To better understand the concept, let us consider another expression of equation (1.1):

$$p(\theta|y) \propto p(y|\theta) \times p(\theta),$$

which states that the posterior distribution for a parameter is proportional to the likelihood, this being the conditional distribution for data under the parameter, multiplied by the prior, which is the marginal probability over the parameter.

This simple equation reveals a two-level hierarchical structure for the parameter, with the likelihood being on the first level, and the prior occupying the second higher level. Specifically, the hierarchy emerges from the step-by-step procedure involved in the construction of the posterior distribution, which combines the prior beliefs of the parameter with the likelihood, refining the understanding of the parameter itself. In short, the likelihood depends on the prior.

It should be noted that more complex models may display structures that have more than one higher level, as in the case of LDA, which is a three-level hierarchical model.

The levels that can be distinguished for Latent Dirichlet Allocation are the following:

- **Word level** - encompasses the word-level variables z_{dn} and w_{dn}, which are sampled once for each word in the document.
- **Document level** - contains the variables θ_d. These are document-level variables, which are sampled once per document.

[1] This concept will be explained further in section 2.4.1.

- **Corpus level** - includes the corpus-level parameters α and β, which the model assumes to be sampled once in the process of generating a corpus.

Although true that LDA is inherently hierarchical, it would be more appropriate and precise to define it as a *conditionally independent* hierarchical model, where dependencies between variables can be specified through a Bayesian Network. The concepts of conditional independence and Bayesian Networks will be explored in the subsequent sections.

2.4.1 Conditional independence

Conditionally independent hierarchical models are based on the assumption that, given some higher-level parameters, observations are independent of each other. Such an assumption is of utmost importance, since it leads to a simplification of both the structure of these models and of the computations associated with inference.

The concept of conditional independence can be explained as follows: consider three variables A,B and C, and suppose that the conditional distribution of A, given B and C is such that it does not depend on the value of B:

$$p(a|b,c) = p(a|c) \quad \forall\, a,b,c\ .$$

Equivalently we can write

$$A \perp\!\!\!\perp B \mid C.$$

In light of this, when considering the joint distribution between a and b conditioned on c, it is possible to write such distribution as follows:

$$p(a,b \mid c) \;=\; p(a \mid b,c)\; p(b|c) \;=\; p(a|c)\; p(b|c), \quad \forall\, a,b,c\ ,$$

which means that, when conditioned on C, the joint distribution between A and B factorizes into the product of the marginal distribution of A, given C, and the marginal distribution of B, given C. In short, the variables A and B are statistically independent, given C.

In the context of Latent Dirichlet Allocation, it is possible to identify the following conditional independencies:

1. Words are conditionally independent, given the topics and the word-topic distribution. This means that the choice of a word w_n does not depend on any other word in the document, if z_n and β are known.
2. Topic assignments are conditionally independent, given the topic distribution. This implies that the assignment of a word to a specific topic depends only on the document's topic proportions θ, and not on other z_n.
3. The topic proportions of a document are conditionally independent of topic proportions of other documents, given the document-topic distribution. In short, θ depends only on the hyperparameter α.

These dependencies become even more apparent with a compact representation of the LDA model, through Bayesian Networks.

2.4.2 Bayesian Networks

A Bayesian Network is a type of probabilistic graphical model, which represents joint distributions in a compact way through directed acyclic graphs (DAG). Notably, Bayesian Networks enable the factorization of joint distributions into a product of conditional distributions, reducing the complexity of the model.

Given a DAG $\mathcal{G}$ over the random variables $X_1, \ldots, X_n$, the nodes of $\mathcal{G}$ correspond to the random variables of the domain, each associated with a conditional probability distribution. Such a distribution reflects the conditional probability of a random variable given its parents in the graph. On the other hand, the edges of $\mathcal{G}$ encode the dependencies between variables. In other words, edges

reflect the direct influence that one node has on another. If two nodes are not connected, they are conditionally independent given their parents.

This structure allows the joint probability of all variables to be factorized as a product of their conditional probabilities.

Formally, a Bayesian network is a pair $(\mathcal{G}, p)$, where p is a joint probability distribution over $X_1, \ldots, X_n$ that factorizes according to the graph $\mathcal{G}$. A distribution p is said to factorize over $\mathcal{G}$ if it can be expressed as:

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid Parents_{X_i}),$$

where $Parents_{X_i}$ denotes the parents of node X_i in $\mathcal{G}$. This equation is known as the *chain rule* for Bayesian networks.

It is worth noting that this factorization implies certain independence assumptions about the underlying model, known as the *local Markov assumptions*. Specifically, if a joint distribution factorizes over $\mathcal{G}$, then each variable X_i is conditionally independent of its non-descendants given its parents:

$$X_i \perp\!\!\!\perp NonDescendants_{X_i} \mid Parents_{X_i}.$$

Given this property, Bayesian networks can be used to provide a compact graphical representation of models such as Latent Dirichlet Allocation (LDA), clearly highlighting the hierarchical dependencies between variables.

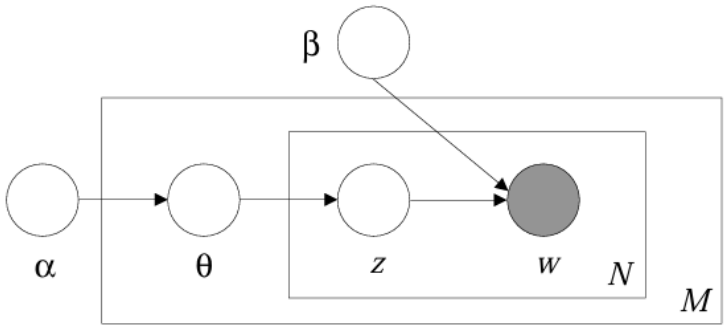

Figure 2.2: Graphical model representation of LDA. The two boxes are known as *plates*, with the outer plate representing documents, and the inner plate containing the repeated selection of words and topics within a document. The node that depicts words is colored in gray to distinguish it from the other variables, which are latent.

2.5 Inference

In section 2.2, Latent Dirichlet Allocation was described as a generative model, with the capability of constructing the parameter space for each document in a corpus, by building a joint distribution between latent and observed variables. However, in practical applications. it is often necessary to reverse this generative process, shifting the focus on inference.

As previously described, LDA's generative process combines the latent variables θ and $\mathbf{z}$ with the hyperparameters α and β to generate a document $\mathbf{w}$. In inference, the procedure is reversed: given the observed document $\mathbf{w}$, the objective is to uncover the hidden variables θ and $\mathbf{z}$, given the prior beliefs encoded by α and β.

In short, the central inferential problem of LDA is computing the posterior distribution of latent variables in a document, or - in other words - the conditional distribution of the topic structure, given the observed documents. Such distribution can be expressed as follows:

$$p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta)}{p(\mathbf{w} \mid \alpha, \beta)},$$

where

- $p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta)$ is the joint distribution of the latent variables θ and $\mathbf{z}$ and the observed document $\mathbf{w}$, conditioned on the hyperparameters α and β.
- $p(\mathbf{w} \mid \alpha, \beta)$ is the marginal probability of the observed document $\mathbf{w}$, obtained by summing over all the possible configurations of θ and $\mathbf{z}$.

As it frequently happens in Bayesian statistics, the denominator of this expression makes the posterior distribution intractable to compute. In fact, to normalize the distribution, it is necessary to marginalize over the latent variables and rewrite equation 2.1 in terms of the model parameters, leading to the following function:

$$p(\mathbf{w} \mid \alpha, \beta) = \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N} \prod_{j=1}^{k} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j} \right) d\theta.$$

This is intractable to compute due to the coupling of θ and β in the summation over the latent topics, which makes the integral over θ non-separable. In addition, the number of possible topic structures is exponentially large, resulting in a significantly increased computational complexity. However, various techniques can be employed to approximate the posterior distributions.

There are two main families of algorithms that are used in the context of topic modeling for approximation purposes: variational algorithms and sampling-based algorithms.

Variational algorithms assume a specific parametrized family of distributions to describe the hidden structure. The goal is to identify the distribution within this family that better approximates the true posterior. Thus, the inferential problem is transformed into an optimization problem, where the parameters of the chosen distribution are adjusted in order to minimize the difference between the distribution itself and the true posterior. An example of such an algorithm will be presented in section 2.5.2.

Sampling-based algorithms, on the other hand, use samples from the posterior in order to approximate it with an empirical distribution. A well-known example of sampling-based algorithm is Gibbs sampling, which will be described in the next section.

2.5.1 Gibbs sampling

The Gibbs sampler is an MCMC sampling algorithm, used to indirectly generate random variables from a marginal distribution, without the need to compute the density. Specifically, given a generic marginal probability density $p(x) = \int p(x, y)\, dy$, it is possible to generate a sample $X_1, \ldots, X_m \sim p(x)$, without the need to calculate $p(x)$. As a result, provided the simulated sample is large enough (i.e. m is large enough), any characteristic of $p(x)$ can be effectively computed with the desired degree of accuracy.

This method is based on the use of univariate conditional distributions (i.e. distributions of a single random variable), given that specific fixed values have been assigned to all other random variables. Such an approach is usually employed when sampling from a multivariate posterior is not feasible, but it is still possible to sample from the conditional distributions of each parameter. The main advantage of using these conditional distributions is that they are easier to simulate, compared with joint distributions, and they tend to have simpler forms. Thus, the computation process is greatly simplified.

In order to better understand how the Gibbs sampler works, consider a simple case: given the bivariate random variable (X, Y), we wish to obtain the marginal distribution $p(x)$. Clearly, the standard approach to obtain $p(x)$ would involve the integration of the joint density $p(x, y)$. However, these computations can be quite cumbersome, and the more variables we consider, the more difficult it becomes to obtain the marginal distribution. The Gibbs sampler circumvents this issue by instead considering a sequence of conditional distributions $p(x \mid y)$ and $p(y \mid x)$, which are used to generate a sample from $p(x)$. Specifically, the algorithm generates a *Gibbs sequence* of random variables

$$Y'_0,\ X'_0,\ Y'_1,\ X'_1,\ Y'_2,\ X'_2,\ \ldots\ Y'_k,\ X'_k.$$

To obtain this sequence of length k, the Gibbs sampler specifies an initial value $Y'_0 = y'_0$, and it obtains the value $X'_0 = x'_0$ by generating a random variable from the conditional distribution $p(x \mid Y = y'_0)$. Then, x'_0 is used to generate a new value $Y'_1 = y'_1$ from the conditional distribution $p(y \mid X' = x'_0)$, and so on for k times. In short, given the initial value $Y'_0 = y'_0$, the algorithm follows the subsequent iterative process, known as *Gibbs sampling*:

$$X'_j \sim p(x \mid Y'_j = y'_j),$$
$$X'_{j+1} \sim p(y \mid X'_j = x'_j).$$

Generally, the distribution of X'_k converges to the true marginal of X - that is $p(x)$ - as k goes to infinity. Hence, provided that the value of k is large enough, the final observation $X'_k = x'_k$ is a sample point from $p(x)$. However, it is not uncommon for the first few samples not to accurately represent the desired distribution, thus being discarded. This phase is known as the *burn-in period*.

If more than two variables are involved, the value of the k-th random variable is drawn from the distribution $p(\theta^{(k)} \mid \Theta^{(-k)})$. In this context, $\Theta^{(-k)}$ is a vector containing all the variables, with the exception of k. As a result, during the i-th iteration, $\theta_i^{(k)}$ is drawn from the following distribution:

$$\theta_i^{(k)} \sim p\left(\theta^{(k)} \mid \theta^{(1)} = \theta_i^{(1)}, \ldots, \theta^{(k-1)} = \theta_i^{(k-1)}, \theta^{(k+1)} = \theta_{i-1}^{(k+1)}, \ldots, \theta^{(n)} = \theta_{i-1}^{(n)}\right).$$

It is worth noting that this algorithm tends to perform poorly when there is a high correlation between the variables involved, due to the misalignment between the distribution's geometry and the sampler's stepping direction. Specifically, because the Gibbs sampler changes one variable at a time, it can only move horizontally or vertically, thus taking a long time to align to a diagonally oriented distribution. Such misalignment causes the algorithm to generate a low effective sample size per iteration, meaning that convergence will be slow.

2.5.2 Variational Inference

Despite being simple and easy to implement, the Gibbs sampler and Monte Carlo algorithms in general tend to converge slowly, and assessment of such convergence can be quite cumbersome. In contrast, variational inference, being based on optimization techniques, converges faster and with fewer iterations.

The approximation procedure employed by variational algorithms is deterministic and it focuses on providing a bound for the probabilities of interest. In this context, the underlying assumption is that even the most complex graphs are actually quite simple from a probabilistic perspective, due to the tendency of their elements not to react strongly to small changes in their neighbors. Specifically, we say that nodes are influenced by an *average effect*, which can be exploited to make simple yet accurate approximations.

For the purposes of our analysis, we will focus on a specific approach to variational inference known as *block approach*, which begins by identifying a substructure of the original graph of interest to approximate its probability distribution. Such approximation is carried out by introducing a family of probability distributions conditioned on a set of *variational parameters*. Then, an optimization procedure is implemented to select the variational parameters associated with the tightest possible lower bound in this family of lower bounds.

Let us now define this procedure in formal terms. Denote by $P(S)$ the joint distribution on the graphical model of interest, where S represents all the nodes in the graph. Define as H and E the two disjoint subsets of S, which represent the hidden nodes and the evidence nodes respectively. To approximate $P(H|E)$ we introduce a family of conditional probability distributions $Q(H|E, \lambda)$, where λ denotes the variational parameters. Then, we select a distribution by minimizing the Kullback-Leiber (KL) divergence - $D(Q||P)$ - with respect to λ. Therefore, we can formally define the minimizing values of the variational parameters as follows:

$$\lambda^* = \arg\min_{\lambda} D(Q(H|E, \lambda)||P(H|E)).$$

where, for any probability distribution $Q(S)$ and $P(S)$, the KL divergence can be expressed as

$$D(Q||P) = \sum_{\{S\}} Q(S) \ln \frac{Q(S)}{P(S)}.$$

Using λ^* we can define a distribution $D(Q(H|E, \lambda^*)$ which is the best approximation of $P(H|E)$ in the family $Q(H|E, \lambda)$.

The reason why we use the KL divergence is that it is the measure that returns the best lower bound on the probability of the evidence nodes ($P(E)$) in the approximating family of distributions. In fact:

$$P(E) = \sum_{\{H\}} P(H \cap E).$$

Taking the logarithm:

$$\ln P(E) = \ln \sum_{\{H\}} P(H \cap E). \tag{2.2}$$

Now, we introduce $Q(H|E)$, a variational distribution over H that we use to approximate the posterior, and we rewrite the sum as an expectation under $Q(H|E)$:

$$\ln P(E) = \ln \sum_{\{H\}} Q(H|E) \frac{P(H \cap E)}{Q(H|E)}.$$

By Jensen's inequality, given that the natural logarithm is a concave function, we have that $f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$. Thus:

$$\ln P(E) \geq \sum_{\{H\}} Q(H|E) \ln \frac{P(H \cap E)}{Q(H|E)}.$$

By rearranging the terms, we obtain the following variational lower bound on $\ln P(E)$:

$$\ln P(E) \geq \sum_{\{H\}} Q(H|E) \ln P(H \cap E) - \sum_{\{H\}} Q(H|E) \ln Q(H|E). \tag{2.3}$$

The difference between the two sides of the equation is simply the KL divergence $D(Q||P)$. In fact, since $P(H \cap E) = P(H|E)P(E)$, we have that:

$$\begin{aligned}
\ln P(E) &\geq \sum_{\{H\}} Q(H|E) \ln P(H|E)P(E) - \sum_{\{H\}} Q(H|E) \ln Q(H|E) \\
&= \sum_{\{H\}} Q(H|E)[\ln P(H|E) + \ln P(E)] - \sum_{\{H\}} Q(H|E) \ln Q(H|E) \\
&= \sum_{\{H\}} Q(H|E) \ln P(H|E) + Q(H|E) \ln P(E) - \sum_{\{H\}} Q(H|E) \ln Q(H|E).
\end{aligned}$$

Since $P(E)$ is independent on H and $\sum_{\{H\}} Q(H|E) = 1$, we can rewrite the equation as follows:

$$\begin{aligned}
\ln P(E) &\geq \ln P(E) + \sum_{\{H\}} Q(H|E) \ln P(H|E) - \sum_{\{H\}} Q(H|E) \ln Q(H|E) \\
&= \ln P(E) \geq \ln P(E) - D(Q||P) \quad \text{where} \quad D(Q||P) \geq 0.
\end{aligned}$$

Therefore, $D(Q||P)$ is a lower bound on $P(E)$ and by computing λ^* we can obtain the tightest lower bound.

Convex duality theory further justifies the choice of KL divergence for the purposes of finding the best lower bound on $Q(H|E, \lambda)$.

The foundational principle of convex analysis is that convex and concave functions $f(x)$ can be represented as follows by means of a conjugate function:

$$f(x) = \min_{\lambda}\{\lambda^T x - f^*(\lambda)\} \quad \text{with} \quad f^*(\lambda) = \min_{\lambda}\{\lambda^T x - f^*(x)\}, \tag{2.4}$$

$$f(x) = \max_{\lambda}\{\lambda^T x - f^*(\lambda)\} \quad \text{with} \quad f^*(\lambda) = \max_{\lambda}\{\lambda^T x - f^*(x)\}. \tag{2.5}$$

It is worth noting that we allow x and λ to be vectors.

Now, let us assume, for the sake of simplicity, that the subset of hidden nodes H is discrete-valued. We can represent the approximating family of distributions $Q(H|E,\lambda)$ as a vector of real numbers, one for each configuration of the variables of H. Such a vector is simply the λ of equations 2.4 and 2.5. Variable x, on the other hand, is another similar vector of real numbers, namely the probability $\ln P(H \cap E)$. Finally, if we define $f(x) = \ln P(E)$, we obtain the following expression:

$$\ln P(E) = \ln \left(\sum_{H} e^{\ln P(H \cap E)}\right),$$

which is convex in the values $\ln P(H \cap E)$. This is simply another way to write equation 2.2 using a logarithmic transformation. Moreover, we have that:

$$f^*(Q) = \min\{\sum_{H} Q(H|E,\lambda) \ln P(H \cap E) - \ln P(E)\}.$$

By minimizing $f^*(Q)$ with respect to the vector $\ln P(H \cap E)$, we get that it is equal to $\sum_H Q(H|E) \ln Q(H|E)$. Hence, by equation 2.4, we can lower bound $\ln P(E)$ as follows:

$$\ln P(E) \geq \sum_{\{H\}} Q(H|E) \ln P(H \cap E) - \sum_{\{H\}} Q(H|E) \ln Q(H|E),$$

which is the same as equation 2.2.

This is a relevant result because it provides a crucial piece of information: in principle, if we were to optimize all the possible probability distributions $Q(H|E)$, we could recover the exact value of $\ln P(E)$. Thus, by ranging over the family $Q(H|E,\lambda)$, we are able to obtain the tightest lower bound available within the family.

This lower bound is of utmost importance in the context of maximum likelihood parameter estimation, which reconciles our discourse with the Latent Dirichlet Allocation model.

In the context of LDA, as previously highlighted, the coupling of θ and β in the summation over the latent topics makes the posterior function intractable. Specifically, issues arise due to the edges between $\theta, \mathbf{z}$ and $\mathbf{w}$. By dropping these edges and the nodes $\mathbf{w}$, it is possible to obtain a simplified graphical model from the original graph of interest. Then, we introduce a family of distributions on the latent parameters conditioned on a set of variational parameters: the Dirichlet parameter γ and the multinomial parameters $(\phi_1, \ldots, \phi_n)$. This family of distribution is characterized by the following variational distribution:

$$q(\theta, \mathbf{z}|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n),$$

which is a conditional distribution that varies as a function of $\mathbf{w}$.

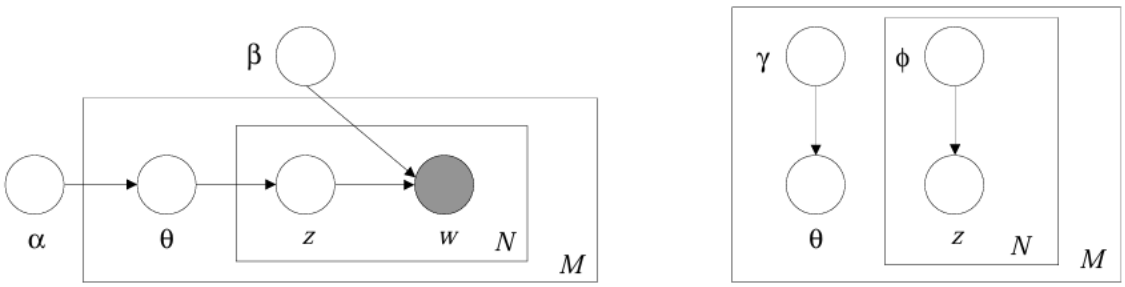

Figure 2.3: On the left side, we have the same graphical model representation of LDA that we have previously observed. On the right side, we can observe a new graphical model representation, which results from the variational distribution used to approximate the posterior of the LDA model.

As before, we need to find a lower bound on the log likelihood of the document using Jensen's inequality. Therefore, by omitting γ and ϕ for simplicity, we have that:

$$\begin{aligned}\ln p(\mathbf{w}|\alpha,\beta) &= \ln \int \sum_z p(\theta,\mathbf{z},\mathbf{w}|\alpha,\beta)\, d\theta \\ &= \ln \int \sum_z \frac{p(\theta,\mathbf{z},\mathbf{w}|\alpha,\beta)q(\theta,\mathbf{z})}{q(\theta,\mathbf{z})}\, d\theta \\ &\geq \int \sum_z q(\theta,\mathbf{z}) \ln p(\theta,\mathbf{z},\mathbf{w}|\alpha,\beta) d\theta - \int \sum_z q(\theta,\mathbf{z}) \ln q(\theta,\mathbf{z}) d\theta.\end{aligned}$$

Since the difference between the two sides of the equation is simply the KL divergence, by denoting the right-hand side of the equation as $L(\gamma,\phi;\alpha,\beta)$, we have that:

$$\ln p(\mathbf{w}|\alpha,\beta) \geq L(\gamma,\phi;\alpha,\beta) - D(q(\theta,\mathbf{z}|\gamma,\phi)||p(\theta,\mathbf{z}|\mathbf{w},\alpha,\beta)).$$

Thus, since the KL divergence is always non-negative, maximizing the lower bound $L(\gamma,\phi;\alpha,\beta)$ with respect to γ and ϕ is simply equivalent to minimizing the KL divergence between the variational distribution and the true posterior. This optimization problem can be formally expressed as follows:

$$(\gamma^*,\phi^*) = \arg\min_{\gamma,\phi} D(q(\theta,\mathbf{z}|\gamma,\phi)||p(\theta,\mathbf{z}|\mathbf{w},\alpha,\beta).$$

It is worth noting that the two optimizing parameters (γ^*,ϕ^*) are document-specific and they both functions of $\mathbf{w}$, which, in the context of the previous equation, is considered as fixed.

A solution to this minimization problem can be provided through fixed-point iteration. Specifically, we write two update equations: a multinomial update and a Dirichlet update, which both maximize the bound $L(\gamma,\phi;\alpha,\beta)$.

$$\phi_{ni} \propto \beta_{iw_n} \exp\{E_q[\log(\theta_i|\gamma)]\} \quad \text{where} \quad E[\log(\theta_i)|\gamma] = \Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{k}\gamma_j\right),$$

$$\gamma_i = \alpha_i + \sum_{n=1}^{N}\phi_{ni}.$$

In this context, ψ is the first derivative of the $\log\Gamma$ function, computed using Taylor Approximation.

This is the first step of a variational expectation-maximization (EM) procedure, aimed at obtaining empirical Bayes estimates of the model's parameters α and β. Specifically, our goal is to find a set of parameters α and β which maximizes the log likelihood of the data. Hence:

$$\ell(\alpha, \beta) = \sum_{d=1}^{M} \log p(\mathbf{w}_d | \alpha, \beta).$$

To reach this objective, the EM algorithm proceeds as follows: first, it maximizes the lower bound with respect to γ and ϕ (E-step); then, considering these two parameters fixed, it maximizes $L(\gamma, \phi; \alpha, \beta)$ with respect to α and β (M-step). Such procedure is performed iteratively until the lower bound converges on the log likelihood.

Chapter 3

LDA Application

This chapter illustrates an implementation of the LDA model on a dataset composed of four novels written by Jane Austen: *Pride and Prejudice*, *Mansfield Park*, *Emma*, and *Sense and Sensibility*. The goal pursued for this application was to identify and analyze the thematic structures present across the four novels by applying LDA.

The dataset employed for this analysis was created from scratch, using the text files of each novel as a source, in order to create a structured and organized corpus suitable for topic modeling. It consists of 214 rows, each representing a chapter from one of the selected novels, and three columns:

- **document_id**: a unique identifier denoting the chapter number of a specific novel (e.g. *Sense_and_Sensibility_Chapter_1*).
- **novel**: the name of the novel from which the chapter is sourced.
- **text**: the content of the chapter.

Python was used for dataset creation, pre-processing and exploratory data analysis, while the implementation of the LDA model was carried out in R.

The following sections will provide details the pre-processing steps, exploratory data analysis, and LDA model implementation and results.

3.1 Data pre-processing

When working with raw text data, pre-processing is a crucial step, aimed at transforming this unstructured information into a structured, normalized format that is suitable for modeling purposes. In short, the goal of this first phase of our application was to reduce the noise in the dataset, to enhance the effectiveness of the LDA model, reduce its complexity and, ultimately, to obtain a set of more meaningful and interpretable topics. For these purposes, six operations were performed on our collection of Jane Austen's novels: tokenization, lower-casing, punctuation and special symbols removal, stopwords removal, lemmatization and de-tokenization.

The first step in the pre-processing process is tokenization, which involves breaking down text data into a set of small, simple semantic units, known as *tokens*. In simple terms, our objective is to reduce the complexity of natural language, by turning it into a format that can easily be processed and analyzed by machines. In addition, this step is of utmost importance for building a vocabulary, which represents the set of unique tokens across the entire dataset. After tokenizing our data, we proceeded with text normalization.

```
# Tokenization
novels_df['text'] = novels_df['text'].apply(word_tokenize)
```

Text normalization involves lower-casing, removal of both punctuation and special symbols (e.g. exclamation points, question marks and commas), and lemmatization.

The reason why it is necessary to convert all text to lowercase is to eliminate any potential discrepancies arising from variations in word capitalization, making it so that identical words are treated as such. Specifically, in natural language, a same word may appear in different forms (e.g. *Day* and *day*) and still maintain the same meaning. However, from a machine's perspective, this difference in form leads to the two versions of the word to be considered as distinct entities.

```
# Lower-casing
novels_df['text'] = novels_df['text'].apply(lambda x: [word.lower() for word in x])
```

Removal of punctuation and special symbols, on the other hand, is performed mainly to reduce complexity and variability. In fact, the information discarded does not usually carry significant semantic meaning, thus adding noise in the context of text analysis. By performing this step, we ensured that only meaningful words were considered during the modeling process.

```
# Punctuation and special symbols removal
novels_df['text'] = novels_df['text'].apply(lambda x: [word for word in x if word.
    isalnum()])
```

Another necessary step aimed at removing noise in the dataset, maintaining only meaningful words, is stopwords removal. Stopwords are common, frequently occurring words in language, which carry little to no semantic value. These include terms such as *and*, *the*, *a* and *is*. By eliminating these words, we reduced the dimensionality of the dataset, improving the performance of the LDA model.

```
# Stopwords removal
stop_words = set(stopwords.words('english'))
novels_df['text'] = novels_df['text'].apply(lambda x: [word for word in x if word
    not in stop_words])
```

To further reduce the dimensionality of the dataset and lower the model complexity, we also implemented a pre-processing technique known as *lemmatization*. Lemmatization is a text normalization technique that reduces words to their root form, or lemma, using linguistic rules and vocabulary to ensure that words are transformed into valid dictionary forms. For example, the words *change*, *changing* and *changer* are all traced back to the lemma *change*. This approach is preferable to stemming, which often truncates words' endings, without considering grammatical correctness. For instance, the same words that we have mentioned before would be reduced to the form *chang*, which does not carry any meaning. By implementing this method, we minimized variation in word forms, leading to better model performance.

```
# Lemmatization
lemmatizer = WordNetLemmatizer()

# Function to get POS tag in WordNet format
def get_wordnet_pos(tag):
    if tag.startswith('J'):
        return wordnet.ADJ
    elif tag.startswith('V'):
        return wordnet.VERB
    elif tag.startswith('N'):
        return wordnet.NOUN
    elif tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN  # Default to noun

# Apply lemmatization with correct POS tagging
novels_df['text'] = novels_df['text'].apply(lambda x:
[lemmatizer.lemmatize(word, get_wordnet_pos(tag)) for word, tag in pos_tag(x)])
```

Since certain exploratory data analysis (EDA) tools require the data to be in its raw, unprocessed form, it was necessary to revert the data back to its raw text form.

```
# De-tokenization
novels_df['text'] = novels_df['text'].apply(lambda x: ' '.join(x))
```

After performing this last step, we proceeded with data analysis and visualization, in order to identify further steps necessary to clean the dataset, before implementing the LDA model.

3.2 Exploratory data analysis

This section will be dedicated to the presentation of the most relevant insights gathered from exploratory data analysis. Since the purpose of this application is topic modeling, EDA was focused on three main aspects: chapter lengths, word frequencies and word importance.

3.2.1 Chapter Lengths

Performing a document-level analysis is of utmost importance in the context of Latent Dirichlet Allocation. In our dataset, the documents of the corpus are represented by chapters. Therefore, we have evaluated their length, in order to get an overview of distribution and to detect possible outliers.

The most relevant insights were gathered from the following boxplots:

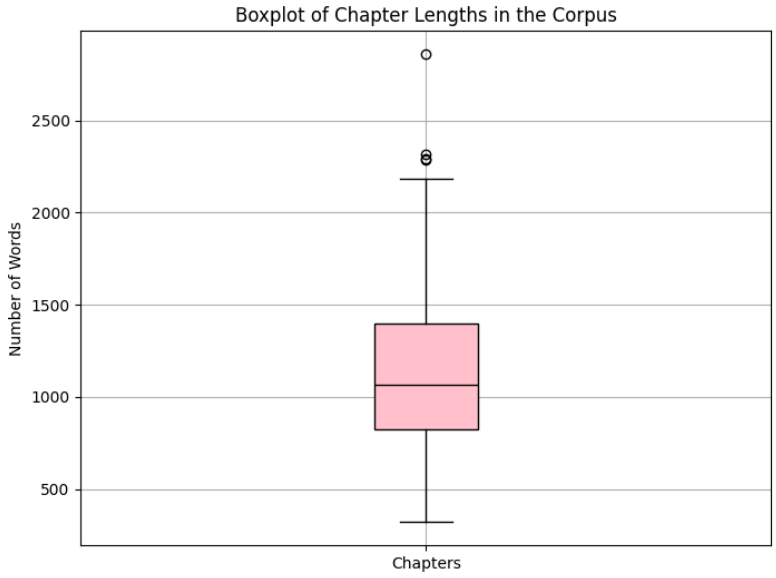

Figure 3.1: Boxplot showing the distribution of chapter lengths across the overall corpus.

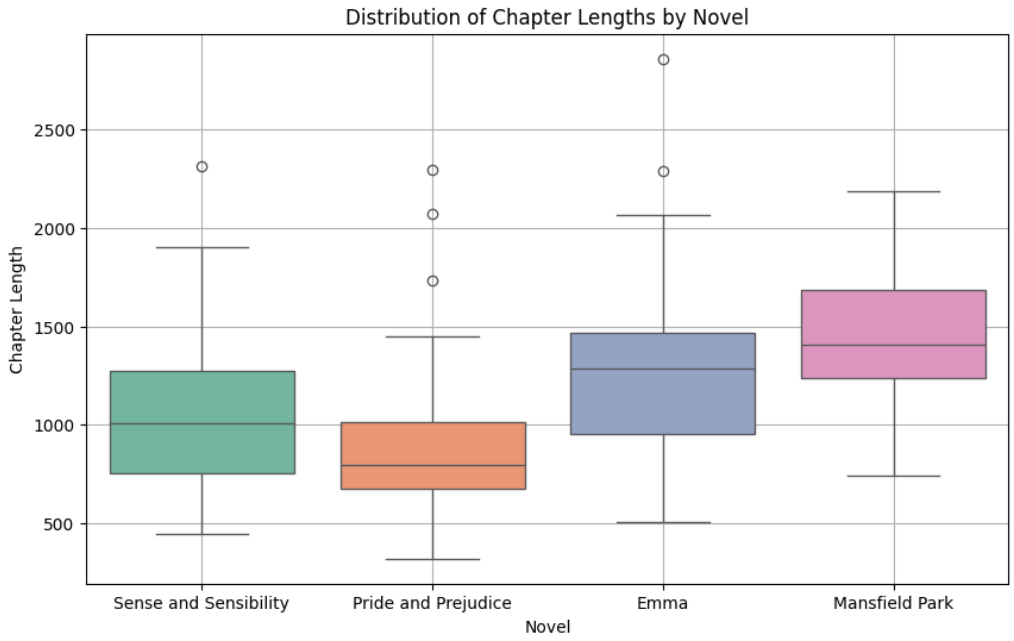

Figure 3.2: Boxplot depicting the distribution of chapter lengths over the four novels that comprise the dataset.

It can be observed that there are important distinctions between the four novels within the dataset, with *Mansfield Park* displaying the highest chapter length - on average - out of all the four novels (about 1380). *Pride and Prejudice*, on the other hand, has the lowest chapter length on average (about 750). In addition, all novels except *Mansfield Park* display some outliers. The most apparent of such outliers is contained within the novel *Emma*, which has a chapter reaching almost 3000 words.

The presence of outliers is also reflected by figure 3.1, which summarizes the document-length distribution across the overall corpus. In addition, this boxplot highlights the fact that we have just few outliers in our dataset, whose removal may not be beneficial for model implementation.

Ultimately, the outliers were not removed. This decision was based on two observations: first, since we are dealing with a small number of irregular documents, it is worth it to preserve information to the detriment of increased complexity. In fact, such chapters may contain relevant information for the purpose of distinguishing topics. Secondly, these outliers are not appendices, summaries, or other documents which would be entirely different compared to the rest. This further supports the first observation, meaning that we cannot exclude that they could be relevant for modeling purposes. In addition, it is safe to assume that the inclusion of these few documents will not significantly hinder the performance of the algorithms employed in this application.

3.2.2 Word Frequencies

The first phase of our word-level analysis was primarily concerned with frequency evaluations. Specifically, we were interested in identifying *hapax-legomena* (i.e. words that appear only once in the overall corpus), and in getting an overview of the most frequently used words. Although finding *hapax-legomena* was necessary to understand which words needed to be removed, as they add unnecessary complexity to the model, the overall frequency analysis was aimed at detecting outlier words, which could add noise.

Figure 3.3 provides a general overview of the results of our analysis: the words that appear the most within the corpus (i.e. the largest words in the wordcloud) seem to be stopword-like terms, such as *would*, *could* and *say*. This is not an unexpected outcome: Jane Austen's novels are permeated by a tone of politeness, which is reflected in the writing. Furthermore, a verb such as *say* is bound to frequently appear in any book in which multiple characters interact.

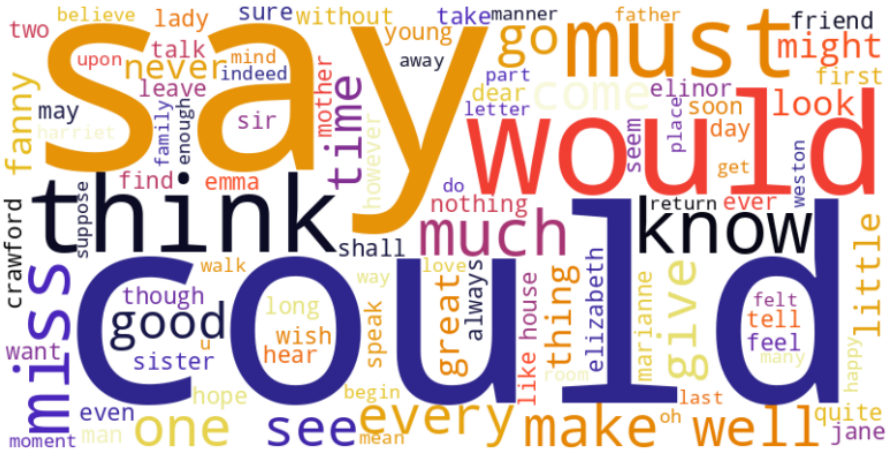

Figure 3.3: Wordcloud showing the top 100 most frequent words in the corpus.

To better understand this high-frequency words in terms of scale, figure 3.4 shows the exact number of times in which some of the most frequent words appear in the corpus. From this histogram, it is clear that there are some outliers: *could*, *say* and *would* have the highest number of occurrences out of all the words in the dataset, surpassing of many points even the fourth most occurring term. In fact, *could* appears more than 2750 times, while *think* does not even occur 2000 times.

Discovering the dominance of terms of little significance is of high importance for our analysis: it suggests that some of these words may be considered as outliers, which should be removed, in

order to reduce noise. Specifically, they may conceal more distinctive terms, leading to poor topic separation. To understand if this is the case, and to take a decision with respect to their omission from the corpus, it was necessary to complete our word-level analysis with a further evaluation: the one of term importance.

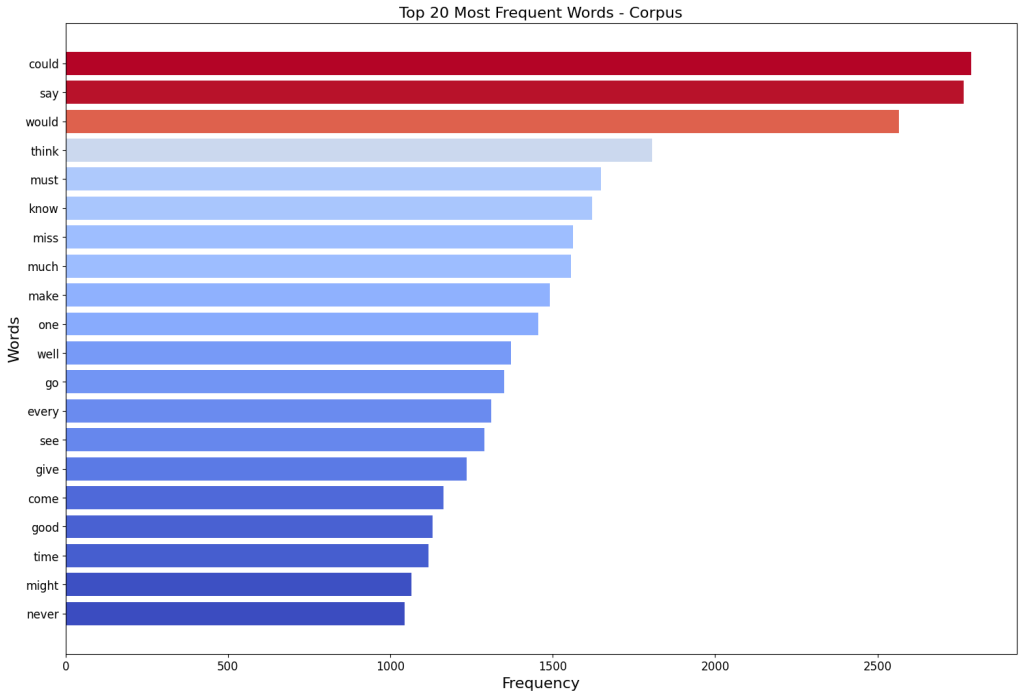

Figure 3.4: Top 20 most frequent words

3.2.3 Word Importance

The identification of outlier words requires to understand whether a term contributes to create noise in the first place. To make this evaluation, we measured the TF-IDF (i.e. term frequency and inverse document frequency) score, which is basically a metric of word importance. Specifically, the higher the score is, the more important a term is with respect to the overall corpus. The analysis of word importance - as we have previously suggested - is strictly related with frequency analysis, since, by connecting the information between frequency and importance, we were able to properly state whether certain words are, in fact, outliers, or if they may distinguish different topics in the corpus.

After computing the TF-IDF score for each word in the vocabulary, and after sorting the resulting values in descending order, we obtain a new wordcloud, displayed in figure 3.5.

Figure 3.5: Wordcloud showing the top 100 most important words in the corpus

Moreover, to get a better sense of scale, a new histogram was created, which captures the 20 words with the highest importance scores:

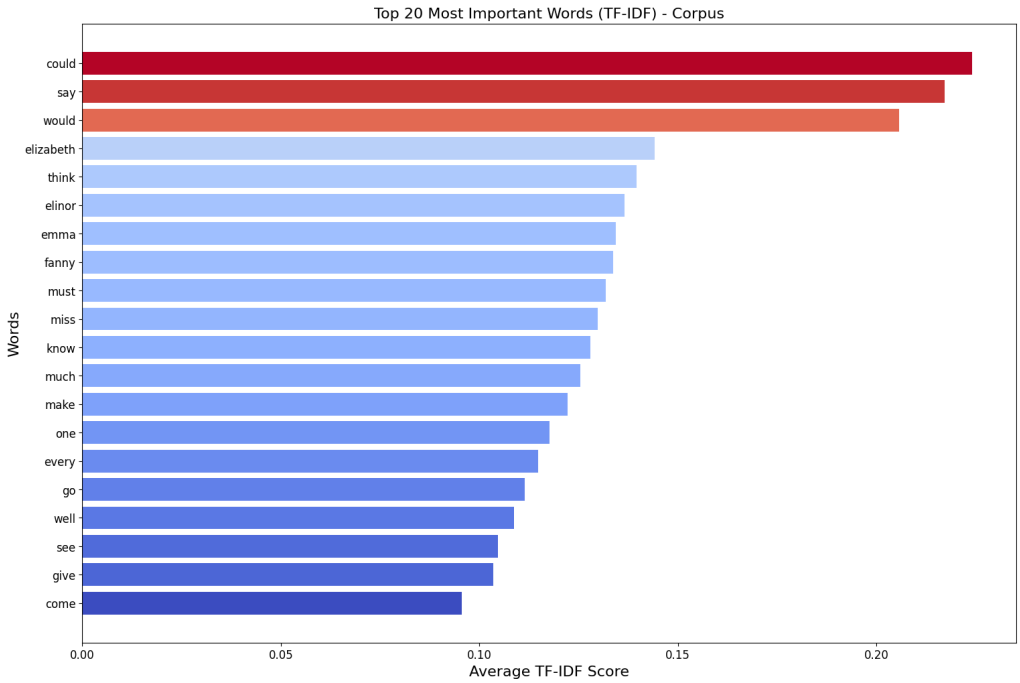

Figure 3.6: Top 20 words with the highest TF-IDF score

As before, *could*, *would* and *say* are the words which display the highest ranking. In addition, there is still a significant gap between their values and that of the fourth-ranked (i.e. *elizabeth*). However, contrary to frequency, main characters' names have a higher rank. In truth, variation in ranking applies to many other words (e.g. *sister*), something which we could infer by checking the results for the 50 most important words and the 50 most frequent words.

From this results, it seems apparent that the TF-IDF score is not penalizing enough the words that appear most frequently, which will undoubtedly increase noise in the results. Therefore, it is necessary to remove these outlier words.

3.2.4 Final pre-processing steps after EDA

The insights gathered from EDA highlighted the necessity of making two further modifications to the dataset, in order to improve the model's performance:

1. ***Hapax-legomena* removal**: words that only appear once add unnecessary complexity, since they do not provide relevant information. Thus, we removed them.

2. **Outliers removal**: from our word-level analysis we inferred that there are words with high frequency and sparsity which dominate over all other terms. Since this could cause poor model performance, such words need to be removed. Therefore, using word importance as a benchmark, words with a sparsity above 0.85 or below 0.05 were removed from the corpus. The reason why low-sparsity words were also excluded was to further reduce complexity.

3.3 Model implementation

The implementation of the Latent Dirichlet Allocation model in R relied mainly on two libraries: *ldatuning*, which facilitated the selection the most optimal number of topics k, and *topicmodels*, which was used to estimate the model itself.

The model's performance was evaluated through the perplexity metric, which quantifies how confidently the model assigns probabilities to a given text sample. Formally, it is defined as the exponential of the average negative log-likelihood of a sequence. The lower the perplexity is, the

lower the uncertainty of the model's predictions, and - as a result - the higher the confidence in predicting the next word in the sequence.

3.3.1 Train-test split and document term matrix

For the purpose of providing a proper evaluation of the model's performance, it was necessary to split the dataset into a train set and a test set. To prevent issues caused by a random split and, more importantly, to preserve data distribution, we performed a stratified split by novel. In this way, we ensured that all novels were properly represented in both the train set and the test set.

```
# Train-test split

# Set seed for reproducible results
set.seed(123)

# Stratification by the 'novel' column
trainIndex <- createDataPartition(clean_novels_df$novel,
                                  p = 0.8,  # 80% train, 20% test
                                  list = FALSE,
                                  times = 1)

# Creation of train and test datasets
train_set <- clean_novels_df[trainIndex, ]
test_set <- clean_novels_df[-trainIndex, ]
```

Furthermore, since we are working with text data, it was necessary to convert our corpus into a format that could be processed by the model. For this purpose, we created a Document-Term-Matrix (DTM), which is a matrix describing the frequency of all the terms that appear in a collection of documents. Specifically, each row denotes a document, while each column represents a word, meaning that the number of columns in a Document-Term-Matrix is exactly equal to the size of the corpus' vocabulary.

The elements of a DTM simply reflect the number of occurrences of a particular term in a specific document. Therefore, if a term does not appear in the document, the entry will be 0, while if the opposite is true the entry will contain the count of the number of times in which the word appears in the document. In short, a Document-Term-Matrix is a simple Bag-of-Words representation.

```
# Document-term matrix creation

# Train set
corpus_train <- Corpus(VectorSource(train_set$text))
dtm_train <- DocumentTermMatrix(corpus_train)

# Test set
corpus_test <- Corpus(VectorSource(test_set$text))
dtm_test <- DocumentTermMatrix(corpus_test, control = list(dictionary = Terms(
    dtm_train)))  # Same dictionary as training set
```

3.3.2 Hyperparameter tuning: the optimal number of topics

At this point, the main challenge was to select the appropriate number of topics k that the LDA model should identify. In fact, like with any other unsupervised learning model, we cannot know *a priori* the value of k that will result in a better performance (i.e. better generalization). As previously stated, the metric that we used to evaluate how well the model generalizes to unseen data was predictive-perplexity. Therefore, we had to perform a cross-validation for the parameter k in order to find a value that minimized the model's perplexity. However, performing such cross-validation for any possible value of k would be cumbersome.

To identify a range of values for k, in which the best value for the performance of the model lies, we used the tools provided by the package *ldatuning*. This R package uses four metrics for an appropriate selection of k:

- **Griffith2004**: a bayesian method based on Gibbs sampling that measures the log-likelihood of the model for each value of k. Therefore, the optimal number of topics is the one that maximizes this metric. It is worth noting that this method is not applicable to a Latent Dirichlet Allocation model using variational inference for parameter estimation.

- **CaoJuan2009**: a measure of topic coherence, based on the computation of the average cosine similarity between topics. Specifically, this method selects the best value of k by evaluating the cluster's density. In this context, clusters are represented by topics, while the average cosine similarity encodes the density of each topic with respect to the others. By using this metric, the best value of k is the one that minimizes the average cosine similarity.

- **Arun2010**: based on the KL divergence, it returns the divergence between document-topic and topic-word distributions. The lower it is, the clearer the document-topic structures are (i.e. the divergence between the two distrubutions is minimized).

- **Deveaud2014**: a measure of topic separation; it identifies the value of k that maximizes the distance between every pair of topics. Topics are considered distinct, and their divergence high, if they present high differences in terms of word distribution. Therefore, the higher this metric is, the more topics are separated, with little overlapping.

The results obtained by the implementation of these metrics are summarized by figures 3.7 and 3.8. These graphs provide some meaningful insights, giving us a hunch of what results we will obtain by performing topic modeling on our dataset using the two different methods for parameter estimation, these being Gibbs sampling and variational inference.

Figure 3.7 suggests that the optimal value of k is likely to be within the range 13-24. Specifically, at 13, topic distinction is maximized, with a high log-likelihood, a low average cosine similarity (i.e. high topic coherence) and a reasonably clear document-topic structure. At 24, on the other hand, the Arun2010 metric is stabilizing and the log-likelihood has significantly improved, while maintaining a good separation between topics, and an approximately equal average cosine similarity. By evaluating these two extrema, and the values in between, we are bound to find a good value of k, which leads to a good model performance in terms of both perplexity and overall modeling of topics.

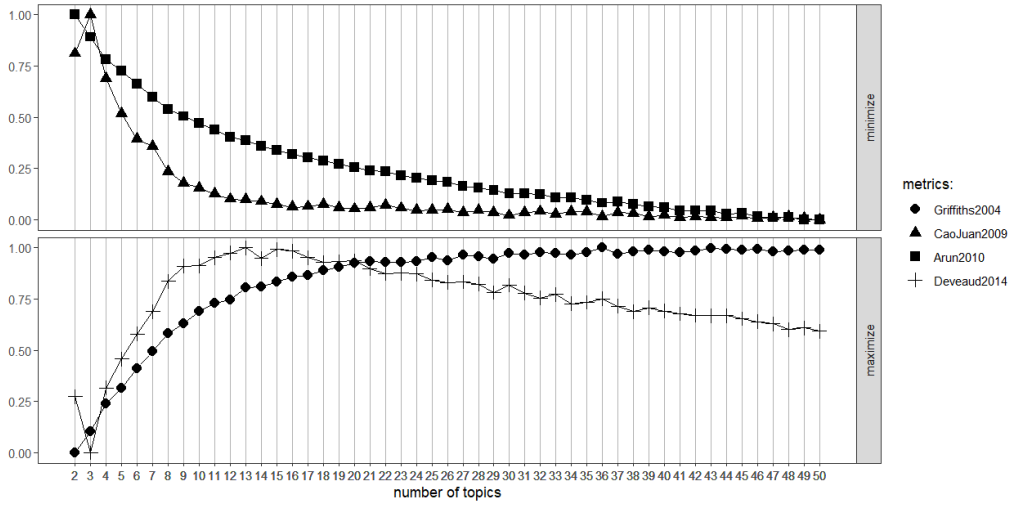

Figure 3.7: Metrics for the selection of the optimal number of topics - LDA using the Gibbs method

Figure 3.8, on the other hand, suggests that the LDA model using variational inference will return poor results in terms of topic separation. In fact, the Deveaud2014 returns very poor results, with a line that is almost constant at zero, with the exception of a small interval around $k = 17$. In reality, these results are quite unsurprising: on average Deveaud2014 for VEM provides worse results compared with Gibbs, due to its approximation of the posterior distribution, which results in smoother topics. In addition, we are working on a dataset in which documents are bound to overlap, since Jane Austen's novels are highly similar with respect to their themes. The combination of these two aspects leads to less distinctive and highly overlapping topics, which will make the interpretation of the generated topics more difficult.

That being said, we can observe that Arun2010 stabilizes in the range 15-20, while CaoJuan2009 returns the best results in the range 12-25. As previously pointed out, the most meaningful result for Deveaud2014 is found around $k = 17$. By combining these observation, we considered the

interval 12-19 as a good range to find the best value of k, since it provides a good compromise between meaningful results and running time of cross-validation.

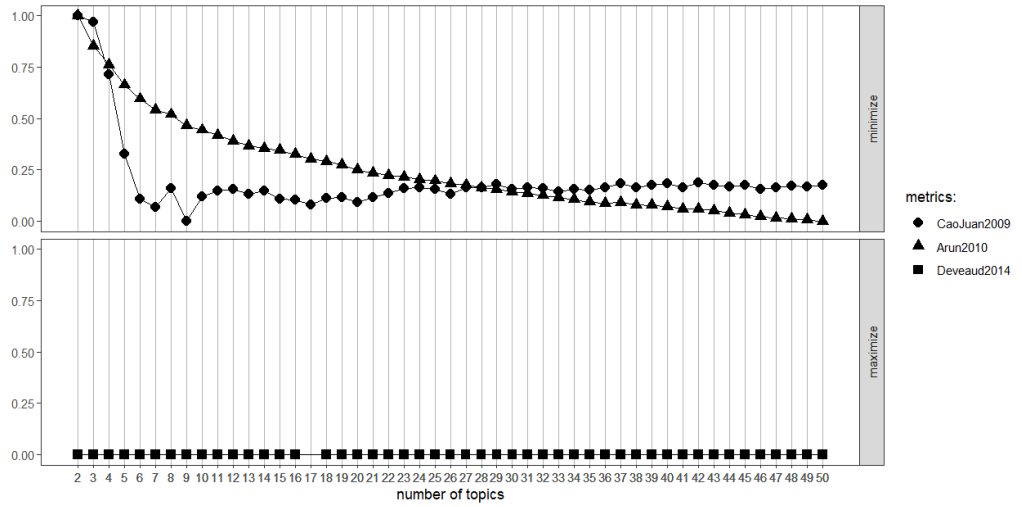

Figure 3.8: Metrics for the selection of the optimal number of topics - LDA using the VEM method

By computing the perplexity of each k within the identified ranges, we obtained the results summarized by figures 3.9 and 3.10. Specifically, we can observe that the value of k which minimizes the model's predictive-perplexity on the test set is 22 in the case in which estimation is performed through Gibbs sampling ($perplexity \approx 3046$), and 12 for estimation with variational inference ($perplexity \approx 1079$). It is worth noting that the perplexity scores have been rounded to improve visualization.

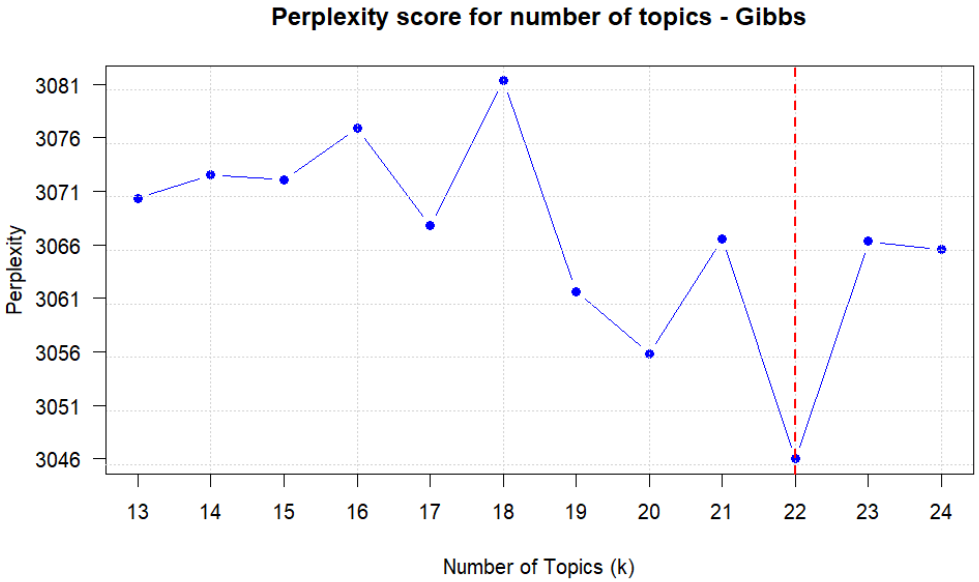

Figure 3.9: Graph depicting the variation in the values of perplexity for the range of k values 13-24 (LDA model using the Gibbs method). The vertical dotted red line highlights the value of k for which perplexity is minimized.

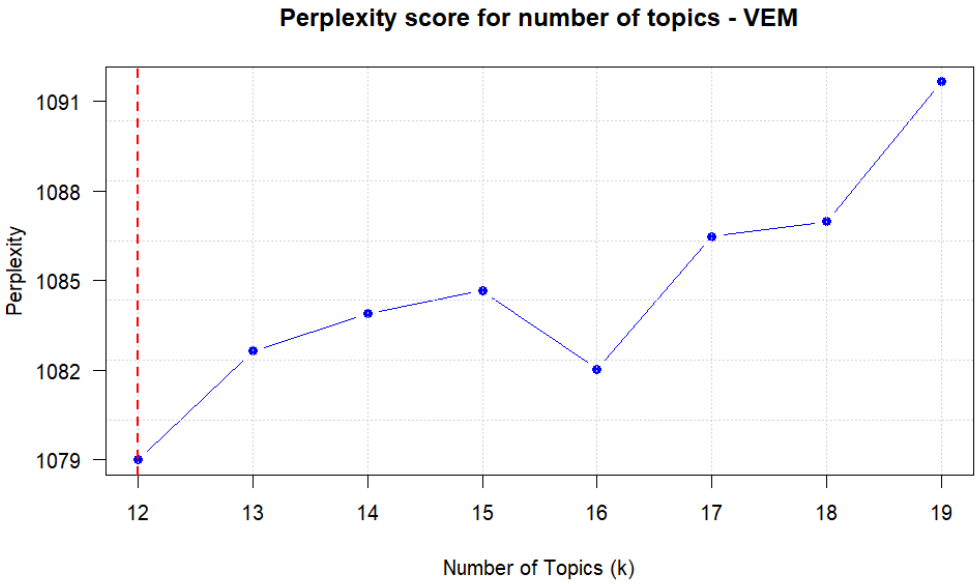

Figure 3.10: Graph depicting the variation in the values of perplexity for the range of k values 12-19 (LDA model using the VEM method). The vertical dotted red line highlights the value of k for which perplexity is minimized.

3.3.3 Training of the model and evaluation of results

After tuning the parameter k, we were finally able to fit the LDA model. Specifically, we fit two versions of the model itself, each using a different method for parameter estimation.

```
# Fit the LDA model with collapsed Gibbs sampling for parameter estimation
lda_Gibbs <- LDA(dtm_train, k = optimal_k_Gibbs_metrics, method = "Gibbs", control
    = list(seed = 1234))

# Fit the LDA model with variational inference for parameter estimation
lda_VEM <- LDA(dtm_train, k = optimal_k_VEM_metrics, method = "VEM", control = list
    (seed = 1234))
```

The perplexity scores obtained by the two LDA models on the test set - which we have already presented in section 3.3.2 - are summarized by the following table:

	LDA Gibbs	LDA VEM
Perplexity score	3045.50	1078.65

Table 3.1: Comparison of Perplexity Scores between LDA Gibbs and LDA VEM

As expected, the LDA model using variational inference for parameter estimation has a better generalization on the test set than the one using Gibbs sampling. Specifically, the perplexity score has a 64.58% decrease in value. However, to get the full picture and properly assess how these two versions of the model behave with respect to the goal of topic modeling, we need to evaluate the results that they generate. For this purpose, we have created two sets of tables (tables 3.2 and 3.3), showing the topic-groups created by the two LDA models. Only the first 10 words for each topic have been displayed.

A quick overview immediately reveals what the metrics used in section 3.3.2 had already suggested: the LDA model using the Gibbs method returns cleaner, more interpretable and better distinguished topics compared to its VEM counterpart, which instead is prone to topic bleeding and redundancy. This will become more apparent by performing a more in depth analysis of the two tables.

In table 3.2, we can observe multiple topics with a high level of distinctiveness. For example, topic 3 is an emotion-heavy topic, which is still quite different from topic 4: although both have a focus on feelings, they consider different aspects of the same subject, avoiding overlaps. Topic 20, on the other hand, is clearly centered around the concept of marriage, one of the core themes of Jane

Austen's novels, while topic 22 is concerned with epistolary communication. However, the most interesting topic groups are undoubtedly the novel-specific ones: topic 5 is about *Mansfield Park*, topic 10 focuses solely on *Sense and Sensibility*. topic 13 on *Emma*, and topic 14 on *Pride and Prejudice*. These groups perfectly capture the key character names and location of each novel, showing that the model has created some text-specific character clusters.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7
1	subject	room	love	heart	fanny	party	bring
2	consider	sit	affection	suffer	crawford	walk	home
3	doubt	minute	happiness	less	edmund	house	father
4	present	walk	heart	poor	thomas	fine	away
5	question	door	happy	distress	sir	shall	hour
6	degree	away	character	believe	bertram	morning	week
7	really	begin	mind	still	rushworth	country	return
8	perhaps	another	felt	talk	norris	home	london
9	circumstance	call	temper	saw	mansfield	turn	comfort
10	case	hear	feeling	almost	aunt	horse	last

	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13	Topic 14
1	year	lady	elinor	moment	part	emma	elizabeth
2	live	sir	marianne	speak	act	harriet	darcy
3	house	young	sister	word	play	weston	bennet
4	daughter	attention	dashwood	tell	many	knightley	bingley
5	child	gentleman	edward	hand	together	elton	sister
6	life	party	mother	felt	work	woodhouse	jane
7	family	house	jennings	towards	whole	churchill	collins
8	brother	observe	willoughby	mean	want	frank	wickham
9	fortune	side	colonel	manner	yates	body	lydia
10	far	family	john	far	maria	hartfield	catherine

	Topic 15	Topic 16	Topic 17	Topic 18	Topic 19	Topic 20	Topic 21	Topic 22
1	shall	dance	dear	place	jane	marry	man	letter
2	yes	pleasure	sure	want	fairfax	suppose	like	write
3	indeed	ball	upon	occasion	hear	woman	young	read
4	sure	happy	tell	present	tell	young	manner	return
5	reply	ask	quite	object	oblige	man	woman	morning
6	really	young	get	together	campbell	marriage	girl	expect
7	something	people	poor	almost	bates	offer	agreeable	next
8	believe	visit	man	general	cole	believe	beauty	since
9	dare	five	suppose	receive	like	manner	please	till
10	ask	many	put	draw	extremely	opinion	acquaintance	ought

Table 3.2: Topics generated by LDA Gibbs

In table 3.3, we can observe a higher similarity between topics, which is mostly attributable to the focus on character clustering. For example, there are now two topics *Mansfield Park* specific (1,2) and three topics *Sense and Sensibility* specific (3,8,10). In addition, topic coherence has overall decreased, resulting in an increased difficulty in topic interpretation. For instance, topic 12 is about epistolary communication, similarly to topic 22 in table 3.2, but contrary to the latter is way more diffuse, containing references to family and emotion. The decreased topic coherence is mostly apparent in topic 7 and 9, which are truly difficult to evaluate.

This is not to say that the model does not provide any meaningful results: topics 6 and 4 for example are novel specific and highly distinguished, the former being about *Pride and Prejudice* and the latter being about *Emma*. In addition, this version of the model seems to be more focused on nuances at the expense of topic separation. This can be inferred by two aspects: firstly, by the overlapping of character names between topics; secondly, by the interesting nature of topic 5. At a first glance, topic 5 may be considered a similar but mixed version of topic 3 in table 3.2: both are emotion-heavy, seemingly focused on feelings alone. However, topic 5 is making an interesting connection, not just mixing a topic with another: it represents emotions with respect

to the complex relationship between men and women, a core theme of Jane Austen's works.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
1	fanny	fanny	edward	emma	love	elizabeth
2	sir	crawford	lucy	harriet	man	darcy
3	thomas	edmund	sure	weston	manner	bingley
4	crawford	bertram	act	knightley	happiness	jane
5	mansfield	sir	part	elton	opinion	bennet
6	rushworth	norris	ferrars	woodhouse	believe	sister
7	edmund	thomas	sister	jane	sense	wickham
8	sister	william	play	quite	woman	lydia
9	susan	lady	elinor	fairfax	felt	dear
10	father	grant	quite	dear	affection	lizzy

	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12
1	letter	elinor	house	elinor	lady	letter
2	yet	marianne	year	dashwood	elizabeth	write
3	love	willoughby	place	marianne	collins	dear
4	mind	jennings	family	john	bennet	sister
5	hour	sister	walk	lady	catherine	believe
6	return	colonel	home	mother	darcy	last
7	doubt	tell	like	sister	young	shall
8	affection	brandon	fine	edward	family	father
9	shall	mother	lady	middleton	charlotte	poor
10	subject	moment	shall	sir	sir	suppose

Table 3.3: Topics generated by LDA VEM

In summary, each method has its strengths and weaknesses and ultimately the best version of the LDA model is simply the one that better suits the specific goals pursued.

Gibbs sampling favors clear-cut topics, with high coherence and distinctiveness, which are of immediate interpretation. As such, it is the best method if the objective is to obtain a clear thematic separation, that properly highlights the fundamental elements which characterize the corpus. However, this high interpretability comes at the expense of nuance and complexity, meaning that subtler themes and relations are lost in topic modeling.

Variational inference, on the other hand, creates overall less coherent topics, characterized by redundancy and opaqueness. In fact, it is often hard to fully grasp the reasoning that resulted in certain words being grouped together. However, by leaning more onto relationships between characters and contextual dynamics, it is better suited to analyze complexity and grasp subtle connections between topics.

Bibliography

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 2003.

[3] George Casella and Edward I. George. Explaining the gibbs sampler. *The American Statistician*, 1992.

[4] Jerome Cornfield. Bayes theorem. *Revue de lInstitut International de Statistique / Review of the International Statistical Institute*, 1967.

[5] Isabella Fornacon-Wood et al. Understanding the differences between bayesian and frequentist statistics. *International Journal of Radiation Oncology, Biology, Physics*, 2022.

[6] Bela a Frigyik, Amol Kapila, and Maya R Gupta. Introduction to the Dirichlet Distribution and Related Processes. Technical report, University of Washington, 2010.

[7] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.

[8] Peter D. Hoff. *A First Course in Bayesian Statistical Methods*. Springer, 2009.

[9] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools Appl.*, 2019.

[10] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. *An introduction to variational methods for graphical models*, page 105161. MIT Press, 1999.

[11] David Kaplan. *Bayesian statistics for the social sciences*. Guilford Publications, 2014.

[12] Robert E. Kass and Duane Steffey. Approximate bayesian inference in conditionally independent hierarchical models (parametric empirical bayes models). *Journal of the American Statistical Association*, 1989.

[13] Pooja Kherwa and Poonam Bansal. Topic modeling: A comprehensive review. *EAI Endorsed Transactions on Scalable Information Systems*, 2019.

[14] Ben Lambert. *A Student's Guide to Bayesian Statistics*. SAGE LTD, 2018.

[15] Julian Gilbey Steve Dobbs, Jane Miller. *Cambridge International AS and A Level Mathematics: Statistics 2 Coursebook*. Cambridge University Press, 2016.

[16] van de Schoot R et al. A gentle introduction to bayesian analysis: applications to developmental research. *Child Development*, 2013.

[17] Wolfgang von der Linden et al. *Bayesian Probability Theory: Applications in the Physical Sciences*. Cambridge University Press, 2014.

[18] C. Walck. *Hand-book on Statistical Distributions for Experimentalists*. Stockholms universitet, 1996.