Ludwig-Maximilians-Universität München

Department of Statistics

Bachelor Thesis

Topic Modeling: An overview of different approaches and their application

Philipp Lintl

supervised by
Prof. Dr. Christian Heumann

August 21, 2019

Contents

Contents

1 Introduction **1**

2 Foundations of Text Mining **3**

2.1 Data preparation 3

2.2 Vector space representation 6

3 Topic Models **9**

3.1 Non probabilistic approach: Latent Semantic Analysis 9

3.2 Probabilistic approaches 11

3.2.1 Probabilistic Latent Semantic Analysis 12

3.2.2 Latent Dirichlet Allocation 15

3.2.3 Correlated Topic Model 21

3.2.4 Other proceedings 23

4 Inference **26**

4.1 Variational Bayesian Inference 26

4.2 Collapsed Gibbs Sampling 28

5 Topic Model Evaluation **33**

5.1 Probabilistic measures 33

5.1.1 Harmonic mean of log-Likelihood 33

5.1.2 Perplexity 34

5.2 Topic coherence measures 35

5.2.1 Human evaluation of topic coherence 35

5.2.2 Automatic evaluation of topic coherence 36

6 Application **42**

6.1 Data set description 43

6.1.1 Newspaper article data 43

6.2 Choosing a suitable number of topics 44

6.3 Means of visualization 46

6.4 Results 48

6.4.1 Breitbart articles 49

6.4.2 NY Times / CNN articles 49

6.4.3 Coherence measures 49

6.4.4 Comparison of topics between news datasets 53

6.5 Concluding remarks of application 55

7 Conclusion and outlook **58**

References **59**

A Inference **65**
A.1 Basic derivation of Variational Inference for unsmoothed LDA 65
A.2 Update equations of the Variational Inference algorithm 67

B Application **68**
B.1 Preprocessing 68
B.1.1 NY Times description 68
B.1.2 Document lengths 69
B.2 Finding suitable K 70
B.3 Results 72
B.3.1 NY Times / CNN collection 73

C Supplementary Material **77**

Abstract

The following thesis covers the subject of topic modeling, a powerful unsupervised technique to analyze and structure large document collections. Topic models conceive latent topics in text using hidden random variables, and discover that structure with posterior inference. This thesis starts with a comprehensive theoretical overview together with respective advantages and limitations of mentioned approaches. Then, two possible inference procedures, as well as means of model evaluation are shown. The latter part deals with possibilities to determine a suitable number of topics, as well as current developments that try to give special consideration to interpretability for humans, rather than statistical properties.

Afterwards, one specific topic model, namely the *Latent Dirichlet Allocation* is applied to two data situations with two respective data sets. The first two text collections consist of rather short questionnaire answers of an external projectpartner of the mobility sector both in English and German. It can be shown that some meaningful topics can be extracted, though the nature of texts limits overall capabilities of topic modeling. Given texts are simply too short, too few and also tend not to be thematically separable enough to reveal clear structures.

The second collection contains newspaper articles of two different American news outlets. They were chosen to illustrate that topic modeling indeed can work well even for smaller text collections. Due to a large number of topics, their analysis turned out to be rather confusing, which is why additional means of visualization are shown. According results were used to demonstrate *coherence measures*, an approach to consider interpretability of obtained topics. With their help, most and least coherent topics were identified and another way to conduct model selection was shown. The ending is marked by a comparison of topics between the two news article collections.

1 Introduction

The ever-increasing digitization of our society goes hand in hand with ever larger amounts of data. Be it in the form of news, Web pages, scientific articles, images or sound. Much of this information is stored in unstructured formats, such as text, which are not organized automatically. The size and nature of such text collections is often overwhelming for users, making it difficult to find what is looked for. Thus, computational tools to aid organization, search, and understanding of these vast amounts of information are needed.

That is, when a set of statistical methods called topic models comes into play. Topic models are a statistical framework, that allow users not just to find individual documents, but to understand the general themes present in the collection. Previously, documents could be found only with the help of keyword search. Now, it is possible to first identify the theme that is of interest, in order to then inspect the documents related to that theme. [Blei, 2012]

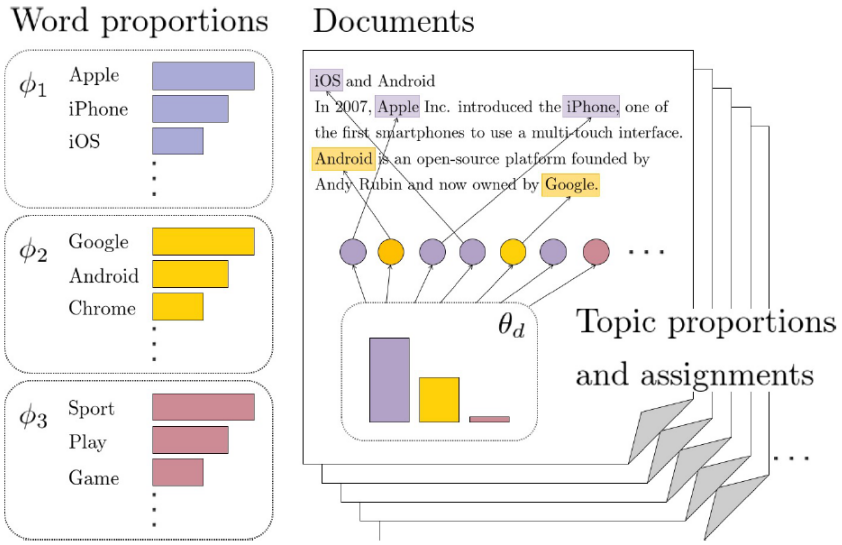


Figure 1: Illustration of LDA, the most prevalent topic model. [Imai, 2016]

The rough intuition of topic models can be indicated as follows. For a given document collection, topic models identify topics with the help of unobserveable latent variables. In that sense, topics are probability distributions over all appearing words. Frequently co-occurring words within the collection are assigned a high probability. Additionally, each document is represented as a distribution over topics. Figure 1 illustrates that idea for one specific topic model called LDA. Each topic ϕ is described by its most probable terms on the left. Each document θ is assigned highest probability to the ϕ it contains most words of. The example shows a document, which mostly shares words that are important to topics 1 and 2. Hence, the according θ values are higher than the one of the third topic. That way, documents

can be organized and allow the identification of structures within large document collections.
A general overview on the topic can be found in Blei [2012]. Furthermore, Boyd-Graber et al. [2017] offer a compilation of different application examples, that illustrate the various benefits of topic models.

This thesis should, as the title suggests, above all give an overview of the broad subject of topic modeling. A solid theoretical foundation from the first few chapters will furtherly be used to develop guidelines and tips, that are important to a user confronted with the task of topic modeling.

At the beginning of this thesis, some basics concerning text analysis and especially the preprocessing of raw textual data are elaborated on.
After that, four topic models are mentioned in a chronological order. The focus is on the Latent Dirichlet Allocation model, which proved to be the cornerstone of all other topic models. The mentioned methods are classified in each case, together with advantages and limitations. Looking ahead, important further developments of the basic model are mentioned together with possible applications.
Chapter 4 then particularly deals with two possibilities of estimation namely Variational Inference and Collapsed Gibbs sampling. Again, limitations, benefits and potential advancements will be discussed.
The last theoretical chapter faces the possibilities of evaluating topic models. It deals separately with probabilistic procedures and methods, that are specifically designed to guarantee human interpretability. Especially since it has been proven that the former do not necessarily lead to results that are helpful to humans, the so-called coherence measures are considered more closely.
The work is then completed with an application chapter. In it, texts of an external project partner, as well as news articles are to be analyzed. On the one hand, the focus is on problems with the external texts as well as on the application of the coherence measures with regards to the news data.

2 Foundations of Text Mining

In this chapter, a quick overview of text mining, as well as specific tasks relevant to topic models shall be given. The former characterizes techniques in data mining that process text documents. Identifying and exploring insightful patterns from unstructured textual data in order to gain useful information, forms a general aim of text mining tasks. Typical techniques, such as text classification, text clustering, text summarization or sentiment analysis usually do not involve deep linguistic analysis, but rather rely on simple text representations. In order to obtain text representations that are suitable for algorithms to work with, several processing strategies exist and shall be introduced. [Sarkar, 2016, p. 49/59] This section shall only be a quick review of specific document processing fundamentals, that are especially vital to topic modeling.

2.1 Data preparation

In general, the *data preparation* or *preprocessing* phase is crucial to the success of text mining practices and applications. Moreover it marks the first step when dealing with textual data. For the sake of simplicity, we assume that the text to be analyzed does not follow a special format. An example for special formats are website contents, which require certain other approaches to obtain analyzable content. Topic models rely on frequencies and occurencies as a basis for probability calculations, rather than deeper linguistical insights and semantic relations. Therefore, the following section covers the basics of text analysis, that focus on text corpora. A collection of documents is called a corpus and Natural Language Processing (NLP) tasks are usually analyzed corpus-based. Suppose for instance, a topic modeling issue regarding newspaper articles: Each article itself is an aggregation of sentences and respectively words and many articles then form a so-called corpus. As this type of data raises difficulties to be analyzed automatically, it is subject to extensive preprocessing. This chapter introduces some issues when coping with raw text data, as well as mentions several low-level preprocessing tasks to deal with possible complications. Further readings into linguistical background can be found in Manning and Schütze [2001], Manning et al. [2008] and Sarkar [2016].

Tokenization

First of all, the text to be analyzed is partitioned into units called tokens. Those units can be words, characters or numbers, that grouped together form a semantic entity suitable for processing. In the example of newspaper articles, one token could be one word, which is part of one document. In this instance, the document would be an entire article. The step, also referred to as *tokenization*, already incorporates a first preprocessing feature, namely the removal of punctuations and whitespaces between words. So, then usually the documents are splitted in order to get tokens.

One possible way of obtaining the tokens is to split at whitespaces after removing punctuation characters. That is, where an issue in context of tokenization arises. As a matter of fact, this step is language specific, meaning that several punctuation characters bear language specific meanings. For instance, hyphens in english can imply splitting of words as within *co-author* or joining nouns as names (*Hewlett-Packard*). Also, the apostrophe character has different meanings depending on the respective language. Another issue with splitting at whitespace in order to obtain word tokenization, appears when dealing with common names such as *New York* or *San Francisco*. Using words as tokens then disregards the information and splits the term into two seperate words *New* and *York*. Furthermore, abbreviations can raise problems. if for instance, the American state of Washington is abbreviated by Wash. and then after removal of punctuation can be mixed with the verb wash. [Manning et al., 2008, p. 22-26]
One alternative to consider compound words is to draw on the concept of n-grams or collocations. In general, an n-gram depicts a sequence of one or more words in a single token. Tokens with one word are called uni-grams (as in Table 2), two words bi-grams (e.g. 'donald trump'), three words tri-grams (e.g. 'donald j trump') and so on. Choosing the maximum size of n-grams depends on the dataset and the use case. In terms of topic models, usually only uni-grams are used. [Jurafsky and Martin, 2008, p. 36]

Stopword removal

Another step within preprocessing large collections of possibly long documents, is the dropping of so called *stop words*. Often, very common words such as *and, it, the* or *how*, lead to an increase in vocabulary size while bearing little information. That is, why strategies exist that exclude these words from the vocabulary, in order to save storage space, as well as speeding up the process. The removal of stop words is usually not at the expense of information, as these words mostly do not impact the final results of most text mining tasks and algorithms. In this case, vocabulary corresponds to all appearing words over all documents available. One strategy could be that the most frequent words are displayed and then those are removed, that do not appear to be valuable to further analysis. Suppose a dataset consisting of emails, then the words *madam or sir* will be frequent but without any contentual information. Another strategy is to match one's text file with existing stopword-lists in order to easily drop those. [Manning et al., 2008, p. 27]

Normalization

Another prevalent preprocessing step includes the lowering of all capital letters. Often it helps to collapse words that are capitally written due to their position in a sentence with their lower written counterpart. However, it can also cause complication. Common names, such as *The New York Times* might be disregarded. That is

why, a compromise is composed by lowering only words at the beginning of sentences and leaving those appearing within sentences. Nevertheless, it is often sufficient to conduct simple text normalization, implying that the entire text is converted to lowercase. [Manning et al., 2008, p. 28-32] For illustrational purpose, suppose an example dataset as in Table 1. Performing word tokenization and normalization

Document	Text
d_1	Trump makes a better deal than Merkel
d_2	Merkel makes a better deal than Trump
d_3	The art of the deal

Table 1: Document examples with respective texts

yields Table 2: This forms one basic step to reaching computer analyzable content.

Document	Vector
d_1	['trump', 'makes', 'a', 'better', 'deal', 'than', 'merkel']
d_2	['merkel', 'makes', 'a', 'better', 'deal', 'than', 'trump']
d_3	['the, 'art', 'of', 'the', 'deal']

Table 2: Documents after tokenization and normalization

Stemming and lemmatization

Another task rises, when dealing with different forms of words of the same origin. Due to grammatical reasons, synonymous words are not represented in the same form and thus would not be captured jointly. In some contexts, it might be of value to merge words like *management, manage, manager or managers* to yield higher frequencies and thus clearer information retrieval.

Stemming refers to cutting of the ends of words, which often implies removing derivational or plural affixes, in the best case leaving only word stems. Though it might improve recall for some queries, empirical research suggests, that stemming does not overhwhelmingly improve the performance of classic Information Retrieval systems, but sometimes even worsens performance due to decreasing accuracy. Manning and Schütze [2001, p. 132/133] elaborate on several reasons for this rather counterintuitive finding.

Lemmatization on the other hand, is based on operations regarding the vocabulary and morphological analysis of words. Rather than solely chopping of endings, it also takes inflectional endings into account and as such leads to base or dictionary forms of words also referred to as *lemma*. An example would be the procession of the words *run, ran, running*. A stemmer might return *run, ran, runn*, whereas a lemmatizer would turn it into *run, run ,run*.

Concluding, the two differ in the way that stemmers typically collapse derivationally related words, whereas lemmatizers usually merge different inflectional forms of a lemma. In Manning et al. [2008, p. 32-34], a few approaches and their basic

functionality are stated.

Most preprocessing tools have in common that it is subjetct to the nature of the texts, as well as the analyzers considerations for a specific task. Several strategies to almost any of those scenarios exist, which can be found in Manning and Schütze [2001] and Manning et al. [2008]. Depending on the size and number of documents, every step can be useful, but needs to be contemplated thoroughly.

2.2 Vector space representation

After preprocessing is finished, the individual word tokens need to be transformed into a format suitable for input into text mining algorithms. Therefore, the so called *bag of words model* is introduced. The name corresponds to the notion, that the exact ordering of words within a sentence is disregarded and only the occurrence and respectively the weighting of a certain word matters. That way, semantic differences, caused by the position of a word, might not be sustained. [Manning et al., 2008, p. 117] Documents d_1 and d_2 of Table 1 for instance, yield the same bag of words representation (Table 3), eventhough meaning the exact opposite. However, as the aim of topic modeling is to extract topics and to allocate documents to topics, such nuances can be neglected. Especially, as both sentences or in general two similar bag of word representations would be matched to the same, or at least a similar topic, which is exactly the goal of topic modeling.
To cover the title of this section in terms of content, the data is transformed into a vector space. To be more precise, documents and words can be seen as vectors, which represent the weights of all included terms of the entire vocabulary. The so called *document-term matrix* was originally introduced as a matter of document *information retrieval*. More generally speaking, a document-term matrix X has $|V|$ (V for Vocabulary and thus representing the number of all distinctive words) columns and M rows (M for number of documents). Looking at the size of prevalent corpora, both M and $|V|$ will likely exceed the thousands, thus leading to computationally expensive calculations. On the other hand, most of the vectors are sparse, meaning that they contain lots of zeros. This can be utilized for more efficient ways of storing and computing such tasks. [Jurafsky and Martin, 2008, p. 271-273] In terms of the aforementioned example, the corresponding *document-term matrix* representing the vectors can be seen in Table 3.
Now, the question arises, whether all words in a document are equally important. As seen with stopwords, there is clearly a distinction between more and less important words. In order to determine importance, several approaches exist and shall be outlined thereafter.

Term frequency

For a collection of documents d, this means that each term t is assigned a certain *weight*. This weight depends on the number of occurrences of each respective term t in the document d. The goal is to somewhat score each term per document, based on the weight of t in d. The first and most intuitive approach is to assign the weight according to the number of occurrences of t within document d. Therefore, this weighing scheme is referred to as *term frequency* and is denoted by $tf_{t,d}$ (Table 3).

		term								
		art	better	deal	makes	merkel	of	than	the	trump
document	d_1	0	1	1	1	1	0	1	0	1
	d_2	0	1	1	1	1	0	1	0	1
	d_3	1	0	1	0	0	1	0	2	0

Table 3: Document-term matrix with term frequency weighting for d_1, d_2, d_3

This representation forms a good basis for models, but entails issues. Depending on the context of a corpus, it will consist of highly frequent common words, which appear in nearly every document. For instance, a document collection of football articles will contain lots of documents with words like *football, club* or *goal* in it and thus leading to higher frequencies. However, those words do not hold as much discriminating power as other, possibly less frequent words. [Manning et al., 2008, p. 117/118]

Inverse document frequency

That is why, a measure is introduced, that takes the *document frequency*, or more specifically speaking, the frequency of terms across documents into account. It aims to decrease the weight of a word with growing document frequency df_t, which represents the number of documents that contain a term t. In order to use it for weighting terms, the so called *inverse document frequency* is denoted by

$$idf_t = log\frac{M}{df_t}, \tag{2.1}$$

where M corresponds to the total number of documents within the corpus. This guarantees a low idf_t for frequent and rather high for less frequent terms across documents. [Manning et al., 2008, p. 118]

Tf-idf weighting

Next up, the preceeding two frequencies are multiplied and hence combined in order to form a joint weight for each term in each document. Thus, the problem of high weights for frequent terms across documents can be solved. The so called *term-*

frequency-inverse-document-frequency weight is denoted by

$$tf\text{-}idf_{t,d} = tf_{t,d} \cdot idf_t. \tag{2.2}$$

To summarize, $tf\text{-}idf_{t,d}$ is lowest for terms practically appearing in every single document and lower for terms less frequent or spread over many documents. To the contrary, it is higher for terms that appear frequently within a small amount of documents. [Manning et al., 2008, p. 117-119] If applied to example documents d_1, d_2 and d_3, one obtains Table 4.

Furthermore, this weighting scheme is often used to identify key words, which can

		term								
		art	better	deal	makes	merkel	of	than	the	trump
document	d_1	0	0.41	0	0.41	0.41	0	0.41	0	0.41
	d_2	0	0.41	0	0.41	0.41	0	0.41	0	0.41
	d_3	1.09	0	0	0	0	1.09	0	0.81	0

Table 4: Document-term matrix with $tf\text{-}idf$ weighting for documents d_1, d_2, d_3

be used for exploratory reasons or to shrink a corpus by deleting low-scoring words. The latter comes in handy, when a large text corpus is subject to analysis. The $tf\text{-}idf_{t,d}$ then helps to prune meaningless words and thus prevents noise. Also, the obtained smaller corpus then yields more efficient calculations, as well as possibly more distinctive results. [Blei and Lafferty, 2009, p. 11]

The vector space model of text with stated weighting schemes is the foundation for several NLP tasks including *named entity extraction* or the computation of *semantic similarity*. The latter results from representing either words or documents as vectors and then computing similarity measures. With that, respective words or documents can be compared for similarity, which is essential to pratical applications like question answering, document classification or clustering. [Jurafsky and Martin, 2008, p. 271] The vector space model also forms the basis for topic modeling, which is why it was introduced thoroughly.

3 Topic Models

Topic models most importantly aid the exploration and browsing of ever growing unstructured data sets in an efficient way, that overstrains our human capabilities. In particular, they provide an automated algorithmic method to identify and discover useful semantic themes in a collection of documents. [Blei, 2012, p. 77/78] They represent a text mining technique for discovering hidden semantic structures, as these are only presumed to run through the documents, but can not actually be observed. More formally speaking, topic models are "[probabilistic] latent variable models of documents that exploit the correlations among the words and latent semantic themes". [Blei and Lafferty, 2007, p. 2] Fundamental to probabilistic models is the idea, that a document is conceived as a mixture of topics, which is a probability distribution over words. The related statistical assumptions slightly differ among the mentioned topic models. In general, documents can be viewed as *generated* by a stochastic process, that is subject to estimation in order to gain estimates of the latent variables. These estimates then serve information retrieval or text mining tasks on a document corpus. [Blei and Lafferty, 2009, p. 2-3] More details on the probabilistic background can be found in section 3.2. Before that, an algebraic procedure is introduced, which more or less formed the foundation for topic models.

3.1 Non probabilistic approach: Latent Semantic Analysis

It is about a method called *Latent Semantic Indexing* or *Latent Semantic Analysis* (LSA). It originates in an algebraic procedure called *Singular Value Decomposition* (SVD). This method is usually utilized to find dominating dimensions of a data set, possibly resulting in a dimension reduction by an approximation. In the NLP setting, SVD helps to reduce large vector spaces, like the vector space representation (see section 2.2), in order to obtain meaningful word representations. The first mention of LSA as a mean of information retrieval can be found in [Deerwester et al., 1990].

Suppose a term-document matrix (transpose of document-term matrix) X with $|V|$ rows, embodying words and M columns for the total number of documents and usually tf-idf weighting. The aim is to approximate X with a lower rank matrix using SVD. Thus, creating a new representation for each document in the corpus to be evaluated. The decomposition allows any such rectangular matrix X to be represented as a product of three matrices U, Σ and V^T (Figure 2). Additional detail on how this is achieved exactly, can be found in [Manning et al., 2008, p. 410-417]. Given X with rank m, the $|V| \times m$ matrix U still row-wise shows a word, but the columns do not represent documents anymore. The columns now respectively represent one of m dimensions in a latent space. Those m column vectors are characterized as being orthogonal to each other. The diagonal matrix Σ consists of all m singular values σ_i, which are ordered descendingly and thus express each dimensions

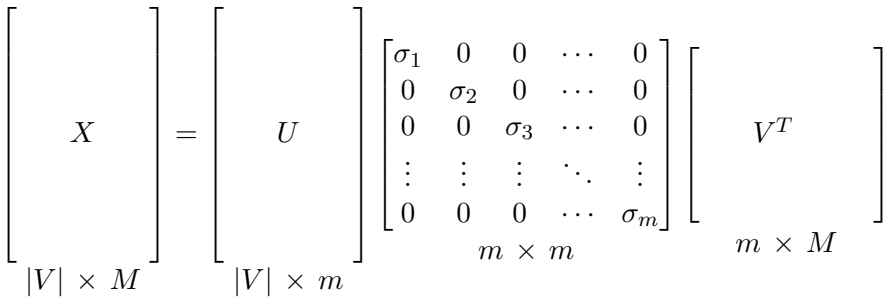


Figure 2: SVD of a term-document matrix visualized

importance. Similar to U, the $m \times M$ matrix V^T still stands for documents in the new latent space dimensions with orthogonal row vectors.
Now, approximation is accomplished by keeping only the top k ($k \leq m$) singular values in Σ by setting all remaining $\sigma_{k+1}, \cdots, \sigma_m$ equal to zero after applying SVD. The

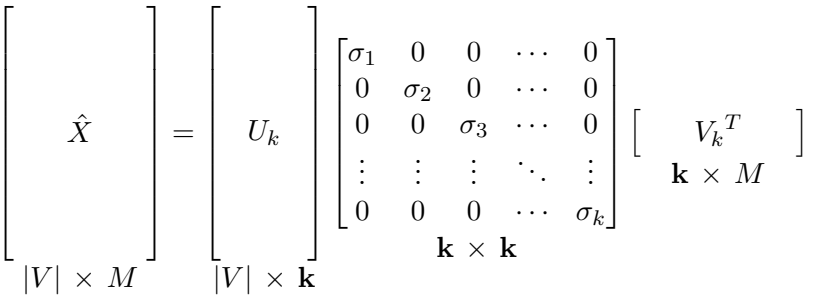


Figure 3: Lower-rank approximation

choice of the top k dimensions results in a reduced matrix U_k with k-dimensional vectors per word and row. The product of reduced matrices then yields $\hat{X}$, the approximation of original X by only looking at the k largest singular values of Σ. [Deerwester et al., 1990, p. 10-14]

LSA for topic extraction

If the goal is to extract topics within a collection of documents, applying SVD results in two sets of factor loadings U_k for terms and V_k^T for documents. The obtained latent space represents these loadings (Figure 3). More precisely, the columns of U_k are the factors with loadings for terms, whereas V_k^T row-wise represents loadings of documents on the new latent dimensions. Therefore, high loading terms of each factor represent a word usage pattern, in this case referred to as topic. Multiplying U_k and Σ_k yields the weight of each term on each latent dimension (topic). Thus, depending on the chosen number of topics K, the strongest weighting terms per topic can be extracted and used to identify their meaning. Equally, multiplying Σ_k and

V_k^t denotes the document representation in the latent space. [Sidorova et al., 2008, p. 471; Sarkar, 2016, p. 235-240]
Even if topic extraction is only a minor application of LSA, it forms the basis for the first probabilistic topic models that proceeded in the following years.

Characteristics of LSA

Contrary to the common usage of SVD as a mean of dimension reduction, LSA does not intend to shrink dimensions as much as possible to allow appropriate visualization. It rather aims to accomplish sufficient power and minimizing the degree of distortion caused by the sparsity of the underlying term-document matrix. [Deerwester et al., 1990, p. 8] The idea is that the low-rank approximation within LSA conflates terms with similar co-occurrences. Thus, quality of information retrieval improves, even if original information is disregarded by the dimension reduction. [Manning et al., 2008, p. 415]
Bearing in mind the features of the introduced vector space representation, several issues arise. It can not handle two classic problems often encountered in natural languages. First, the case of two different words sharing the same meaning (as with *car* and *automobile*), called *synonymy*. The vector space representation views them as distinctive words, thus allocating both to one separate dimension. Secondly, *polysemy* describes words, which have several meanings like *python*, thus leading to inconsistencies regarding similarity tasks. [Manning et al., 2008, p. 413] At least for synonymy, it can be shown, that given enough data, LSA helps to overcome the issues going along with it. [Deerwester et al., 1990, p. 21] However, throughout the literature it is claimed, that the results of LSA are somewhat restricted. Papadimitriou et al. [2000] investigates appropriate conditions for applying LSA.

3.2 Probabilistic approaches

The preceeding method relies on three assertions. First, that a word-document co-occurrence matrix (general for document-term-matrix) generates semantic information, second that dimensionality reduction is a vital result of this and third that words and documents can be illustrated as points in the Euclidean space. The proceeding probabilistic models comply with the first two. The third is not met, as *probabilistic topics* form semantic properties of words and documents. [Steyvers and Griffiths, 2007, p. 2]
In generative probabilistic modeling, data is treated as arising from a generative process, that includes hidden variables. [Blei, 2012, p. 4] More formally speaking, probabilistic topic models map this concept into a *hidden variable model* of documents. Hidden variable models allow observed data to interact with random variables and thus form structured distributions. With these distributions, hidden structures are presumed in the observed data. Applying posterior probabilistic inference on the generated results then yields the learned structures. In terms of observed text data,

it is looked for latent topical structures, that describe the observed data the best. [Blei and Lafferty, 2009, p. 3]

Key assumption: exchangeability

Most probabilistic topic models are based on the already adressed *bag-of-words* assumption, which disregards the order of words in a document. In terms of probability theory, this assumption is referred to as *exchangeability*. The same was elaborated on by Blei et al. [2003] with regards to de Finettis theorem. There is proof that mixture distributions represent a collection of exchangeable random variables. Thus, the present exchangeability of words and documents, is accounted for by mixture models. Suppose a finite set of random variables $\{z_1, \cdots, z_N\}$. If the corresponding joint distribution is invariant to permutation,

$$p(z_1, \cdots, z_N) = p(z_{\pi(1)}, \cdots, z_{\pi(N)}) \tag{3.1}$$

holds. Whereas π denotes a permutation of integers from 1 to N. "De Finetti's representation theorem states that the joint distribution of an infinitely exchangeable sequence of random variables is as if a random parameter were drawn from some distribution and then the random variables in question were independent and identically distributed, conditioned on that parameter." [Blei et al., 2003, p. 998]
This leads to a simple and factored *conditional* joint distribution of the random variables. The assumption is vital to simplifying the process of finding computationally efficient inference procedures. [Blei et al., 2003, p. 994/995; 998]
This explanation of the intuition behind probabilistic topic models is followed by several models of this class.

3.2.1 Probabilistic Latent Semantic Analysis

The beginning is marked by the *Probabilistic Latent Semantic Analysis* (pLSA) model, also referred to as *aspect model*. As the name hints, it is based on the idea of LSA, but enhances it to a proper generative model. In fact, pLSA models each word in a document as a sample of a mixture model, which is a set of topics in the form of multinomial random variables. Provided K topics are given, the aim is to find the probability distribution of words in a topic and the probability distribution of topics in a document. The topics can not be observed and thus are latent variables, words the observed variables.

Notation pLSA

The fact that each author uses a different notation, caused confusion. Therefore, each model more or less is presented in the authors notation. According to Hofmann [1999], notation results as:

- document collection D with documents d_i, $i \in \{1, \cdots, M\}$

- latent variable (topic) z_k, $k \in \{1, \cdots, K\}$
- words within a document $w_i \in \{1, \ldots, N_j\}$
- $p(z_k)$ marks the distribution of topics $\boldsymbol{z}$ in a particular document
- $p(w_i|z_k)$ represents the probability distribution over word w_i given a topic z_k
- $p(d_j)$ denotes the probability for a word occurring in a specific document d_j
- lastly, $p(z_k|d_j)$ describes a document-specific probability distribution over the latent variable space

Data generating process

The generative pLSA model for word/document co-occurrences can be stepwise derived as:

1. Select a document d_j with respective probability $p(d_j)$
2. For each word w_i in the document d_j:
    - Choose a latent class z_k from a multinomial distribution conditioned on the given document with probability $p(z_k|d_j)$
    - Generate a word w_i from a multinomial conditioned on the previous chosen latent class with probability $p(w_i|z_k)$

Those steps lead to an observation pair (d_j, w_i). This data generating process then is used to formulate a joint probability model given by:

$$p(d_j, w_i) = p(d_j)p(w_i|d_j), \qquad p(w_i|d_j) = \sum_{k=1}^{K} p(w_i|z_k)p(z_k|d_j). \tag{3.2}$$

In order to receive equation 3.2, all possible states of latent class z_k need to be considered, or mathematically speaking, summed over. [Hofmann, 1999, p. 180/181]
A graphical representation of the pLSA/aspect model can be seen in Figure 4. It

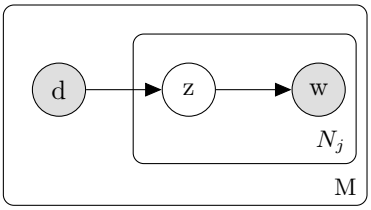


Figure 4: Graphical model representation of pLSA/aspect model. [Blei et al., 2003, p. 1000].

shows the pLSA model as a probabilistic graphical model, which is a common visualization for probabilistic generative models with repeated sampling steps. Each node

corresponds to a random variable, whereas shaded and unshaded variables indicate observed and latent variables respectively. Conditional dependencies are illustrated by arrows between variables. Plates (the boxes in the figure) depict sampling steps with respective number of repetitions in the lower right corner. [Jordan, 2004]
Two important assumptions are made. First, the bag-of-words approach determines that the joint variable (d_j, w_i) is sampled independently. That way, the joint distribution of observed data is factorized as a product. Second, conditional independence of d_j and w_i on the state of the latent class z_k is presumed. These assumptions are essential to the postulation of a maximum-likelihood function for the predictive probability of the observed word occurrences $p(w_i|z_k)$. Finding the parameters, that maximize the joint probability $p(w_i, d_j)$ and thus learn unobservable probabilities to infer the hidden aspects (topics), is the goal. [Hofmann, 1999, p. 180/181]. Exact formulations and means of model fitting can be found in Hofmann [1999].

Characteristics of pLSA

In comparison to LSA, it incorporates some improvements. As it is a purely data-driven unsupervised learning technique to uncover hidden aspects, it allows words to appear in different contexts. That way, it counteracts polysemy, as well as synonymy. The former, because pLSA considers a latent context variable for every word occurrence. The latter due to the fact, that it allocates words with similar meaning to similar topics. Moreover, the probabilistic generative framework itself carries advantages. It allows fundamental statistical methods, such as model fitting, model selection or the quality evaluation in terms of predictive performance, to be applied. [Hofmann, 1999, p. 178]
Though it formed a first step towards probabilistic modelling of text, a few issues were raised. These are mainly due to the fact, that it is not a probabilistic model at the level of documents. The mixture components are multinomial random variables, which does not hold for documents. Those are depicted as a list of numbers not being part of a probabilistic generation. One of the issues regards the number of parameters of the mixture components $p(w_i|z_k)$ and $p(z_k|d_j)$. They grow linearly with corpus size, which makes the model prone to overfitting. Furthermore, pLSA only learns topic mixtures $p(z_k|d_j)$ for documents it has previously been trained on, as d_j is a multinomial random variable with as many possible values. Where it is fundamental to generative models to assign probabilities to yet unseen documents, there does not exist a natural way for this model. Also, parameter estimation is rather slow, and the fitting process does not guarantee global optima. [Blei et al., 2003, p.1000/1001]

3.2.2 Latent Dirichlet Allocation

PLSA suffered from several shortcomings, which motivated the most prevalent topic model called *Latent Dirichlet Allocation* (LDA). It relies on the notion that characteristics of topics and documents are drawn from Dirichlet distributions. In this particular probabilistic model, a presumed hidden topical structure within the documents is modelled with several latent random variables. With the addition of those, the model allows documents to display more than one topic. [Blei et al., 2003]

Notational aspects of LDA

Symbol	Description
K	number of topics
V	number of unique words over all documents
D	number of documents
N_j	number of words within document j
$\boldsymbol{\theta}_j$	proportion of topics specific to document j
$\boldsymbol{\phi}_k$	proportion of words specific to topic k
$z_{j,i}$	identity of the topic of the ith word in document j
$w_{j,i}$	identity of the ith in document j
$\boldsymbol{\alpha}, \boldsymbol{\beta}$	parameters of the Dirichlet distributon

Table 5: Notation of LDA

Specifically for LDA, there are three latent variables together with prior parameters, which require notational reference. There are two hyperparameters for the two Dirichlet distributions: A positive K-vector $\boldsymbol{\alpha}$ and a positive V-vector $\boldsymbol{\beta}$. The latent variable representing the topic assignment of the i-th word in the j-th document is called $z_{j,i} \in \{1, \ldots, K\}$. The topic distribution for document j is denoted by $\boldsymbol{\theta}_j \in [0,1]^K$ and the third latent variable $\boldsymbol{\phi}_k \in [0,1]^V$ describes the word distribution for topic k. If the Dirichlets are chosen to be symmetric, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can also be scalars. Another notation, particularly useful for more thorough derivations, displays frequencies. Particularly, $n_{k,j,v}$ shall stand for the number of times word v is assigned to topic k in document j. Some instances also require the concentration over k, j or v. Specifically, $n_{k,j,\cdot}$ is the number of words in documnet j assigned to topic k, $n_{k,\cdot,v}$ is the frequency of word v being assigned to topic k in all documents. Finally, $n_{k,\cdot,\cdot}$ illustrates the number of words in all documents assigned to topic k. [Carpenter, 2010; Blei et al., 2003]

Dirichlet distribution

Drawing samples from the Dirichlet distribution is essential to LDA. This is the reason to briefly introduce this rather uncommon distribution.
The Dirichlet distribution forms the multivariate generalization of the beta distribution, which itself is heavily drawn on in hierarchical Bayesian models. Suppose

a k-dimensional Dirichlet random variable $\mathbf{x}$, which is defined in the (k-1)-simplex. Thus, $x_i \in [0, 1]$ and $\sum_{i=1}^{k} x_i = 1$ must hold. Expressed differently, parameter vectors for a Multinomial distribution can be obtained by the Dirichlet. In Latent Dirichlet Allocation, this fact is taken advantage of to come up with a principled way of generating the multinomial distributions, that cover the latent variables. The according density function of $\mathbf{x}$ then emerges as:

$$P(\boldsymbol{x}|\alpha_1, \ldots, \alpha_k) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}, \tag{3.3}$$

where Γ denotes the Gamma function and $\boldsymbol{\alpha} > 0$ represents a K-vector [Blei and Lafferty, 2009, p. 4]. The Gamma function is a generalization of the factorial function to real values. The Dirichlet furtherly occupies two beneficial features. For once, it is a *conjugate prior* to the Multinomial. Using it as a prior over the parameters of a Multinomial distribution thus yields a posterior distribution, which again is a Dirichlet distribution. On the other hand, Dirichlet is a member of so-called *exponential family*. In the exact derivations of several inference methods, both instances come in handy. [Geigle, 2016, p. 1-3]
Depending on the nature of parameter vector $\boldsymbol{\alpha}$, there is a *symmetric* and an *asymmetric* Dirichlet version. Former is the case, when each component of the parameter is equal. Each component in the random vector $\mathbf{x}$ is the probability of drawing the item associated with that component. [Blei and Lafferty, 2009, p. 4] In order to get a notion on the influence of hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the density of various examples on the two-simplex is drawn. On the left, Figure 5 depicts the symmetric and on the right the asymmetric case. It can be seen that smaller parameter values lead to higher density at the extremes/edges of the simplex. Larger values by contrast entail a peakier density.

In general, suitable choices for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ depend on the number of topics K and vocabulary size V. Steyvers and Griffiths [2007] suggest that $\alpha = \frac{50}{K}$ and $\beta = 0.1$ tend to work well for several text collections. Hence, in the application section 6, these values are chosen as a matter of extent and convenience.

Data generating process

Sampling a mixture of these topics and then sampling words from that mixture generates a document $\boldsymbol{w}_j = \{w_{j,i}\}_{i=1}^{N_j}$.
The total generative LDA model can be stepwise described as:

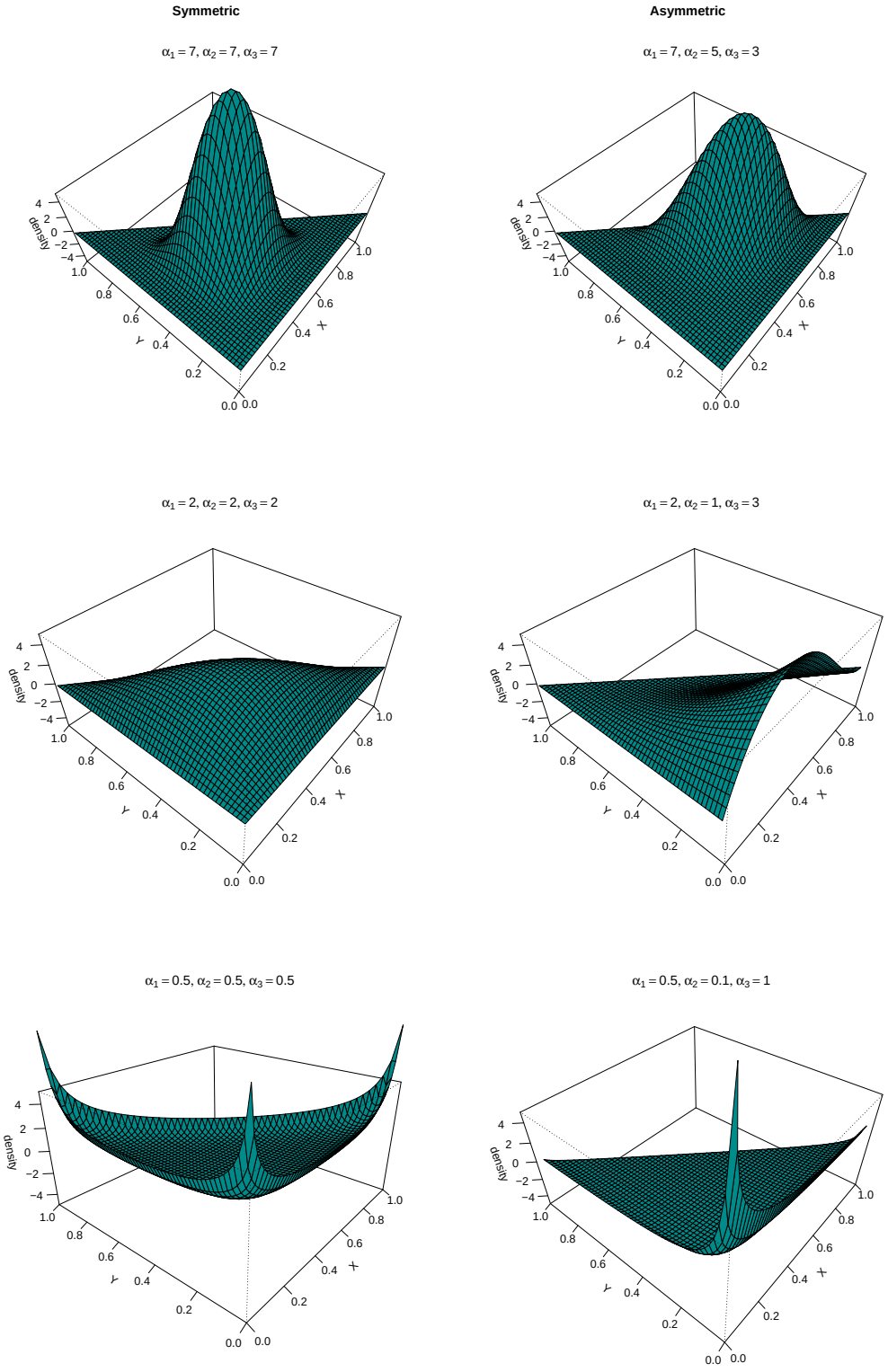


Figure 5: Symmetric and asymmetric Dirichlet density of order K=3

1. For each topic $k = 1, \ldots, K$ choose likely words:
   (a) Draw a distribution over words $\boldsymbol{\phi}_k \sim Dir(\boldsymbol{\beta})$
2. For each document $j = 1, \ldots, M$:
   (a) Decide what proportions of topics should be in the document:
   Draw a vector of topic proportions $\boldsymbol{\theta}_j \sim Dir(\boldsymbol{\alpha})$
   (b) For each word $i = 1, \ldots, N_j$:
      (i) Draw a topic assignment $z_{j,i} \sim Mult(\boldsymbol{\theta}_j)$, $z_{j,i} \in \{1, \ldots, K\}$
      (ii) Given this topic, choose a likely word (generated in step 1):
      Draw a word $w_{j,i} \sim Mult(\phi_{z_{j,i}})$, $w_{j,i} \in \{1, \ldots, V\}$

Following notation of Table 5, $Dir(.)$ displays the Dirichlet distribution and $Mult(.)$ the Multinomial distribution. Topic proportions $\boldsymbol{\theta}_j$ and word proportions $\boldsymbol{\phi}_k$ are the parameters of multinomial distributions. They are drawn from Dirichlet distributions, leading to dependencies among variables seen in Figure 6. Given an observed set of documents $\boldsymbol{W} = \{\boldsymbol{w}_j\}_{j=1}^{M}$, it is aimed to infer the topics $\boldsymbol{Z} = \{\{z_{j,i}\}_{i=1}^{N_j}\}_{j=1}^{M}$. A joint probability distribution over both the observed and hidden random variables is given by the generative process. In order to obtain information, the joint distribution is analyzed by computing the conditional distribution of the hidden variables given the observed variables. This conditional distribution is then called the *posterior distribution*.

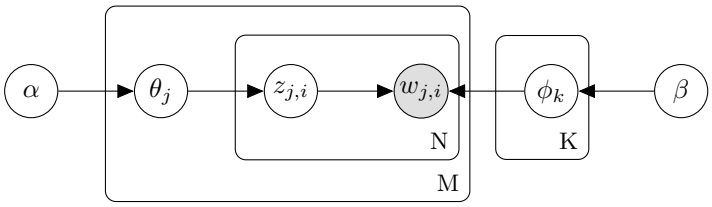


Figure 6: Graphical model representation of *smoothed* LDA with plates notation [Blei et al., 2003, p. 1006].

The corresponding graphical model (Figure 6) incorporates the conditional independence relationships, which can then be used to obtain a joint probability distribution of a set of topic mixtures $\Theta = \{\boldsymbol{\theta}_j\}_{j=1}^{M}$, a set of topic assignments $\boldsymbol{Z}$, a set of the corpus words $\boldsymbol{W}$ and word proportions $\Phi = \{\boldsymbol{\phi}_k\}_{k=1}^{K}$, given the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$:

$$p(\boldsymbol{W}, \boldsymbol{Z}, \Theta, \Phi | \boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\Phi|\boldsymbol{\beta}) p(\Theta|\boldsymbol{\alpha}) p(\boldsymbol{Z}|\Theta) p(\boldsymbol{W}|\Phi, \boldsymbol{Z}) \tag{3.4}$$

$$= \prod_{k=1}^{K} p(\boldsymbol{\phi}_k|\boldsymbol{\beta}) \prod_{j=1}^{M} p(\boldsymbol{\theta}_j|\boldsymbol{\alpha}) \prod_{i=1}^{N_j} p(z_{j,i}|\boldsymbol{\theta}_j) p(w_{j,i}|\boldsymbol{\phi}_{z_{j,i}}) \tag{3.5}$$

The probabilities of interest incorporated in the model can now be determined by marginalizing over the joint distribution. The right side and its particular factors can be specified more exactly, which is important for using a fitted topic model later on, as well as better understanding its components. [Geigle, 2016, p. 5]

The first term $p(\Phi|\boldsymbol{\beta})$ represents the term distributions per topic for the entire corpus. As $\boldsymbol{\phi}_k$ was drawn from a Dirichlet distribution, its hyperparamter $\boldsymbol{\beta} > 0$ is a V-vector appropriate to the properties of a Dirichlet distribution. The according probability of Φ for all topics and the entire vocabulary is given by:

$$p(\Phi|\boldsymbol{\beta}) = \prod_{k=1}^{K} \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v - 1} \tag{3.6}$$

This means, that provided a topic k, the term v is drawn with probability $\phi_{k,v}$.
The second term $p(\boldsymbol{\theta}_j|\boldsymbol{\alpha})$ denotes the topic distribution per document j.

$$p(\boldsymbol{\theta}_j|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{j,k}^{\alpha_k - 1} \tag{3.7}$$

The third term, $p(z_{j,i}|\boldsymbol{\theta}_j)$ displays the probability of picking topic z for word i in document j given the topic proportion is $\boldsymbol{\theta}_j$. In a more general sense, it denotes the distribution of the topic to words assignments. Thus, yielding the following equation for all documents and topics:

$$p(\boldsymbol{Z}|\Theta) = \prod_{j=1}^{M} \prod_{k=1}^{K} \theta_{j,k}^{n_{k,j,\cdot}}. \tag{3.8}$$

The fourth term describes the probability of a corpus $\boldsymbol{W}$, when $\boldsymbol{Z}$ and Φ are given:

$$p(\boldsymbol{W}|\boldsymbol{Z}, \Phi) = \prod_{k=1}^{K} \prod_{v=1}^{V} \phi_{k,v}^{n_{\cdot,k,v}} \tag{3.9}$$

In total, all those terms can be inserted into the joint distribution (equation 3.5). [Ponweiser, 2012, p. 19-21]
Essential to the usage of LDA is the solution to the posterior distribution of the hidden variables given the parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}$ and the observed corpus words $\boldsymbol{W}$. It can be formulated as:

$$p(\boldsymbol{Z}, \Phi, \Theta|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{p(\boldsymbol{W}, \boldsymbol{Z}, \Theta, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})}{p(\boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta})} \tag{3.10}$$

Looking at the form of the denominator more thoroughly, reveals difficulties for

exact posterior inference.

$$p(\boldsymbol{W}|\boldsymbol{\alpha},\boldsymbol{\beta}) = \int_{\Phi}\int_{\Theta}\sum_{Z} p(\boldsymbol{W},\boldsymbol{Z},\Phi,\Theta|\boldsymbol{\alpha},\boldsymbol{\beta})d\Theta d\Phi$$

$$= \int_{\Phi} p(\Phi|\boldsymbol{\beta})\int_{\Theta} p(\Theta|\boldsymbol{\alpha})\sum_{Z} p(\boldsymbol{Z}|\Theta)p(\boldsymbol{W}|\boldsymbol{Z},\Phi)d\Theta d\Phi \tag{3.11}$$

The sum over the interdependent latent topics assignments incorporates the coupling of Φ and Θ, which is intractable to compute. [Blei et al., 2003, p. 1003; Geigle, 2016, p. 6]
Learning the adressed distributions is a problem of Bayesian inference. Thus, efficient methods for approximating the posterior distribution are needed. Two approaches, including Variational Bayesian inference and Gibbs Sampling are covered in section 4.

Topic extraction with LDA

Now, as the latent variables are inferred, the question of how to use these results to better grasp the inherent structures of the corpus arises. The posterior expectations of the latent variables are required to do so. They may be summarized by:

- Topic probability of a term: $\widehat{\phi_{k,i}} = \mathbb{E}[\phi_{k,i}|\mathbf{W}]$.
- Topic proportions of a document: $\widehat{\theta_{j,k}} = \mathbb{E}[\theta_{j,k}|\mathbf{W}]$.
- Topic assignment of a word $\widehat{z_{j,i,k}} = \mathbb{E}[z_{j,i} = k|\mathbf{W}]$.

The first term is used to determine the obtained topics by ordering terms descendingly according to their probabilities for each topic. Thus, the topic can be identified by looking at the semantic information of the top words.
The remaining terms are used to visualize documents with respect to their topic decomposition. The posterior topic proportions $\widehat{\theta_{j,k}}$ provide insight into a documents context. They also allow the grouping of documents regarding to their topical structure. The possibility of one document being composed of multiple topics distinguishes LDA from usual clustering methods. Similar documents can be identified by comparing the difference of $\widehat{\theta}$, rather than computing distance measures as in LSA. [Blei and Lafferty, 2009, p. 5/6]

Characteristics of LDA

By introducing a latent variable on the topic mixtures, LDA overcomes the shortcomings of pLSA. In particular, LDA can be used on unseen documents, as well as it does not grow in parameters with increasing corpus size (K + KV parameters). Moreover, overfitting issues of pLSA seem to be combatted. [Blei et al., 2003,

p. 1001] However, its main adavantage is the use as a basic module for more sophisticated goals. A quick overview of developments is given in section 3.2.4, whereas one specific advancement is introduced more thoroughly.
A central limitation of LDA is the inability to model correlations between topics, even though topics of a text corpus most likely will be highly related. Take for instance, the case of a current newspaper article collection. Texts about Donald Trump will presumably also be about immigration or foreign affairs, but rather less likely adress environmental or sports issues. The *Correlated Topic Model* (CTM) attends to the matter and therefore is introduced more thoroughly in the following. [Blei and Lafferty, 2007, p. 3]

3.2.3 Correlated Topic Model

The Correlated Topic Model counteracts LDAs incapability to allow inter-topic correlation. This issue originates in the Dirichlet distribution on topic proportions. In CTM, the logistic normal distribution is favored over the Dirichlet, as it integrates a covariance structure between components. Whereas the Dirichlet and its nearly implied independence among proportions does not allow one topic to be correlated with another. Theoretically, this should improve language modelling performance, as text corpora tend to inherit correlations among topics. [Blei and Lafferty, 2007, p. 3/4]

Notational aspects of CTM

Symbol	Description
$\mathcal{N}_K()$	K-dimensional Gaussian distribution
$\boldsymbol{\Sigma}$	$K \times K$ covariance matrix of Gaussian
$\boldsymbol{\mu}$	mean K-vector of Gaussian

Table 6: Additional notation of CTM

This follows the notation of Table 5 with addition of Table 6. Due to the logistic normal prior, the only difference between LDA and CTM emerges with topic proportions $\boldsymbol{\theta}_j$. Let $\boldsymbol{\Sigma}$ be a $K \times K$ covariance matrix and $\boldsymbol{\mu}$ the respective mean K-vector of a multivariate Gaussian distribution. Then, the resulting multivariate random variable $\boldsymbol{\eta}_j$ is transformed in order to serve as parameter $\boldsymbol{\theta}_j$ for the Multinomial of $z_{j,i}$. [Blei and Lafferty, 2007, p. 4]

Data generating proceess

The generative process underlying the CTM model is assumed to yield a document with N_j words as follows.

Given topics Φ, for each $j = 1, \ldots, M$:

1. Draw $\boldsymbol{\eta}_j | \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\} \sim \mathcal{N}_K(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
2. For $i = 1, \ldots, N_j$:
   (a) Draw topic assignment $z_{j,i} | \boldsymbol{\eta}_j$ from $Mult(f(\boldsymbol{\eta}_j))$
   (b) Draw word $w_{j,i} | \{z_{j,i}, \boldsymbol{\phi}\}$ from $Mult(\boldsymbol{\phi}_{z_{j,i}})$,

whereas the real-vector $\boldsymbol{\eta}_j$ is depicted into the simplex by $f(\boldsymbol{\eta}_j)$:

$$\boldsymbol{\theta_j} \;=\; f(\boldsymbol{\eta_j}) \;=\; \frac{exp(\boldsymbol{\eta_j})}{\sum_i exp(\boldsymbol{\eta_i})} \tag{3.12}$$

This is necessary, in order to obtain a multinomial parameter, which is needed for the the Multinomial distribution of $z_{j,i}$. As seen in the directed graphical model

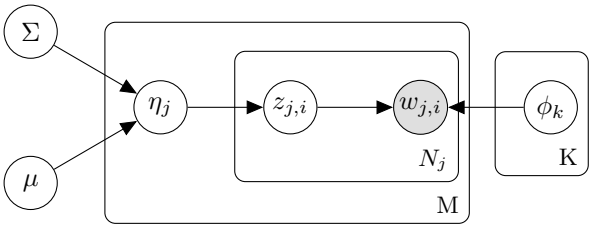


Figure 7: Graphical model representation of CTM. [Blei and Lafferty, 2007, p. 5]

(Figure 7), the generative process is similar to LDA, with the exemption of the topic proportions being drawn from a logistic normal distribution rather than a Dirichlet. The logistic normal distribution maps a multivariate draw into the simplex via normalization. Thus, dependencies among the components of the obtained simplicial random vector are caused by the covariance of the Gaussian. The allowed variability coming along with it represents the inter topic correlation, or in other words capture possible relationships among topics. [Blei and Lafferty, 2007, p. 4]
Overcoming the unrealistic assumption of uncorrelated topics within LDA and thus becoming a more expressive document model does not only impose advantages. With the abundance of the Dirichlet prior for topic proportions, goes astray a favourable characteristic of the Dirichlet distribution. Namely, that contrary to Dirichlet, the logistic normal distribution is not a conjugate prior to the Multinomial. This raises complications for the approximation of the respective posterior inference procedure. [Blei and Lafferty, 2007, p. 4-6]
In CTM, the posterior distribution of the latent variables of one document with N words can be formulated as such:

$$p(\boldsymbol{\eta}, \boldsymbol{Z} | \boldsymbol{W}, \Phi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \;=\; \frac{p(\boldsymbol{\eta} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{i=1}^{N} p(z_i | \boldsymbol{\eta}) p(w_i | z_i, \Phi)}{\int p(\boldsymbol{\eta} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{i=1}^{N} \sum_{z_i=1}^{K} p(z_i | \boldsymbol{\eta}) p(w_i | z_i, \boldsymbol{\Phi}) d\boldsymbol{\eta}} \tag{3.13}$$

This equation, similar to equation 3.10 of LDA, is intractable to compute because of the denominator. Firstly, the coupling of K values of the latent variable over words causes a combinatorial number of terms. Additionally, the nonconjugacy of $p(\boldsymbol{\eta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to $p(z_i|\boldsymbol{\eta})$ leads to incomputability of the exact integrals of each term. [Blei and Lafferty, 2007, p. 6/7]. This feature prevents the usage of many MCMC methods, which were specifically built for Dirichlet-mixed membership models. Therefore, deterministic variational methods are applied. The exact derivation is given in [Blei and Lafferty, 2007, p. 8-10]

Characteristics of CTM

Topic extraction works simililar to LDA, with a handy additional feature. Not only results the change of prior in a better fit for document collections, but the covariance matrix of the logistic normal enables the relationships among topics to be visualized. Thus, an additional tool for explanatory analysis is given. The obtained relationships can be illustrated with a graph, as seen in Blei and Lafferty [2007, p. 11/12]. The direct comparison to LDA in the original paper revealed the support for more topics and a better performance according to performance measures. [Blei and Lafferty, 2007, p.14/15]
Putting aside advantages, there are also restrictions emerging from the additional flexibility of the CTM, particularly regarding computational efficiency. As a matter of fact, time complexity reaches $\mathcal{O}(K^3)$, which is primarily caused by slower inference procedures. These involve frequent matrix inversion operations on a covariance matrix, that quadratically grows in parameters with increasing number of topics. This constitutes a substantial restriction to the model, especially as it's potential in practical real-world tasks involving massive amounts of documents and topics can not be reached without efficient inference procedures. [He et al., 2017, p. 2] However, there has been research combatting this problem. Speeding up the variational inference and taking advantage of a newly learnt low-dimensional topic space, one approach claims to reach linear complexity with topic size. [He et al., 2017]

Concluding, the CTM model facilitates a richer extraction, leading to better predictive performance, as well as an additional way to descriptively analyze text corpora. In order to scale up to large corpora, efficient implementations need to be deployed.

3.2.4 Other proceedings

If generally speaking, more complex goals than solely uncovering the semantic structure of a collection of texts are to be achieved, assumptions made by LDA need to be scrutinized. For instance, the bag-of-words assumption seems out of touch, if more sophisticated goals, such as language generation, are sought. Models, that consider words nonexchangeable thus conditioning word generation on previous words, improve language modelling performance. An example would be the *Bigram Language*

Model introduced by Wallach [2005], which combines bigram-based and topic-based approaches.
Another extension responds to data situations, in which the order of documents might be of interest. This is the case, when documents underly a temporal structure and are collected over a long period of time. Again, suppose a collection of newspaper articles as an example. One can assume, that a shift in topics over the years will be noticeable. The so called *Dynamic Topic Model* proposed by Blei and Lafferty [2006], bears in mind the order of documents and thus allows to elaborate on the change of topics over time.
Furthermore, considering additional information, such as author, geographic location or title, into the topic model can be of advantage. The *Author-Topic Model* [Rosen-Zvi et al., 2004] specifically allows for analysis about authors and documents. It can be useful, if only a certain number of authors is available and for instance, it is of interest, which authors write about similar topics. A more general approach is called the *Structural Topic Model*. It provides a better alternative to post-hoc comparisons by incorporating corpus structure or all kinds of document metadata into the standard Topic Model. [Roberts et al., 2013]
Another, more general issue with LDA, as with practically all other unsupervised learning methods, is the choice of number of latent classes K. In section 5, several model selection approaches to aid the choosing are given. Nevertheless, there is an extension to LDA called *Bayesian nonparametric Topic Model* or *Hierarchical Dirichlet Process* [Teh et al., 2004], which incorporates this step. This particular mixture model is a natural nonparametric generalization of Latent Dirichlet Allocation, where the number of topics can be unbounded and learned from data during posterior inference. [Blei, 2012, p. 83]
Finally, a specific current approach is adressed as an outlook for possible further developments. It is the so called *lda2vec* model. [Moody, 2016] It forms a hybrid approach combining word embeddings with topic models. As a reminder, word embeddings (e.g. *word2vec*) form powerful word representations in form of continuous word vectors, which are obtained by several types of neural networks. These vectors are characterized in particular by the fact that the position in the sentence is taken into account, quite contrary to the bag of words approach. Word vectors are positioned in the vector space such that words, that share common contexts in the corpus are located in close proximity to one another in the space. [Mikolov et al., 2013] If trained on a sufficient amount of data, meaning hundreds of millions of words, the obtained high-quality word vectors along with their similarity representation can be illustrated as follows. The original lda2vec publication, for instance, revealed a linear relationship among the respective word vectors: "NLP - text + image = computer vision" [Moody, 2016, p. 7]. The lda2vec model is now characterized by the fact that it learns dense word vectors together with topic and document vectors. [Moody, 2016, p. 1] Sparing the details, it forms a novel and interesting approach to

topic extraction. The fact that those trained word vectors along with their inherited linguistic contexts are available, could improve topic model tasks and thus lead a new direction towards more complex models with even greater computational efforts.

To conclude, there have been many alterations to the basic form of LDA designed for specific tasks. Not only do several extensions for various text data situations exist, but it can also be benefitial to other kinds of data. For instance, there has been substantial research in applying LDA alike models to audio, computer code or video data. Especially within computer vision it is appreciated, as pictures tend to share characteristics of text documents. Meaning, that pictures might contain several visual patterns even throughout a collection of pictures. It thus helps for classification or similarity tasks. [Blei, 2012, p. 82/83] In general, the user is asked once again to find the best model according to the available data. To accomplish that, being aware of all kinds of alterations of LDA is necessary. As seen in the application section, proper knwoledge of the nature of texts facilitates the topic extraction considerably.

4 Inference

In this chapter, two approaches for approximating the posterior distributions of unobserved variables within LDA and especially equation 3.10 are introduced. As seen, exact inference is intractable, which is why efficient procedures of approximation are necessary. Over the years, many approximate inference algorithms have been applied and especially altered for LDA. This chapter shall give an overview by elaborating on two of the most prevalent and further developments.

4.1 Variational Bayesian Inference

First and originally deployed in Blei et al. [2003], is the so called *Variational Inference* algorithm. The basic underlying principle involves a simpler distribution called variational distribution q with *free variational parameters*. Those are then fit to approximate an intractable posterior distribution over hidden variables as accurately as possible. [Blei and Lafferty, 2009, p. 8]

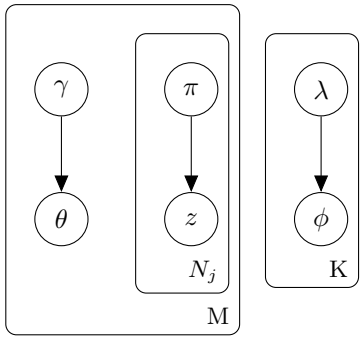


Figure 8: Graphical representation of *smoothed* parameterized distribution q. [Geigle, 2016, p. 13]

The variational distribution for LDA is distuingished by *independent* variables (Figure 8). This is contrary to the true posterior, which had interdependencies coupled through the observed documents. In the respective variational distribution, each parameter is controlled by a different variational parameter, resulting in the following distribution.

$$q(\Theta, \boldsymbol{Z}, \Phi|\boldsymbol{\lambda}, \boldsymbol{\gamma}, \boldsymbol{\pi}) = \prod_{k=1}^{K} q(\boldsymbol{\phi}_k|\boldsymbol{\lambda}_k) \prod_{j=1}^{M} \left(q(\boldsymbol{\theta}_j|\boldsymbol{\gamma}_j) \prod_{i=1}^{N_j} q(z_{j,i}|\boldsymbol{\pi}_{j,i}) \right) \tag{4.1}$$

The variational parameters are described by distributions over their types as follows:

- Topics Φ by a V-Dirichlet distribution $\boldsymbol{\lambda}_k$.
- Topic proportions Θ by a K-Dirichlet distribution $\boldsymbol{\gamma}_j$.
- Topic assignment $z_{j,i}$ by a K-multinomial distribution $\boldsymbol{\pi}_{j,i}$.

[Blei and Lafferty, 2009, p. 8] Those parameters are obtained by solving an optimization procedure, that tries to achieve similarity of q towards the true posterior p. In terms of dissimilarity function, Variational Bayes usually relies on minimizing Kullback-Leibler (KL) divergence between the two distributions.

$$(\boldsymbol{\lambda}^{\star}, \boldsymbol{\gamma}^{\star}, \boldsymbol{\pi}^{\star}) = \underset{(\lambda,\gamma,\pi)}{\operatorname{argmin}} KL(q(\Theta, \boldsymbol{Z}, \Phi|\boldsymbol{\lambda}, \boldsymbol{\gamma}, \boldsymbol{\pi}) \,\|\, p(\Theta, \boldsymbol{Z}, \Phi|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})) \tag{4.2}$$

Using Jensen's inequality, a lower bound on the log likelihood for any q can be determined. Rearranging the formulas shows the equivalency of minimizing KL divergence and maximizing the lower bound $\mathcal{L}$ with respect to λ, γ and π. Appendix A.1 illustrates this idea more thoroughly for unsmoothed LDA. According to Blei and Lafferty [2009, p. 9], the objective function for smoothed LDA of this optimization problem can be specified as such:

$$\begin{aligned}\mathcal{L} = &\sum_{k=1}^{K} \mathbb{E}_q(log\, p(\boldsymbol{\phi}_k|\boldsymbol{\beta})) + \sum_{j=1}^{M} \mathbb{E}_q(log\, p(\boldsymbol{\theta}_j|\boldsymbol{\alpha})) + \sum_{j=1}^{M}\sum_{i=1}^{N_j} \mathbb{E}_q(log\, p(z_{j,i}|\boldsymbol{\theta}_j)) \\ &+ \sum_{j=1}^{M}\sum_{i=1}^{N_j} \mathbb{E}_q(log\, p(w_{j,i}|z_{j,i}, \Phi)) + H(q)\end{aligned} \tag{4.3}$$

$\mathcal{L}$ turns out to be the sum of the expectation of the log probabilities of the posterior with respect to the variational distribution in equation 4.1 and the entropy H of q. As hinted in A.1, computing the derivatives of the exact form of $\mathcal{L}$ with respect to the variational parameters and setting them equal to zero, yields update equations. Since these equations depend on each other, optimization is now achieved by iteratively performing so called coordinate ascent updates to increase the objective. As the variational parameters are independent, the expectations in equation 4.3 will not depend on the parameter being updated. That is why, one iteration of the mean field variational inference algorithm updates each parameter holding the others fixed. The exact updates are given in Appendix Figure 19. [Blei and Lafferty, 2009, p. 9]
These updates are repeated until some criteria of convergence of the objective function $\mathcal{L}$ is reached. As the goal was to approximate the true posterior, the latent variables can be expressed in terms of the variational distribution given by:

- The topic probabilities of term v are

$$\widehat{\phi_{k,v}} = \frac{\lambda_{k,v}}{\sum_{l=1}^{V} \lambda_{k,l}}. \tag{4.4}$$

- The topic proportions of document j are

$$\widehat{\theta_{j,k}} = \frac{\gamma_{j,k}}{\sum_{m=1}^{K} \gamma_{j,m}}. \tag{4.5}$$

- The per-word topic assignment expectation is

$$\widehat{z_{j,i,k}} = \pi_{j,i,k}. \tag{4.6}$$

[Blei and Lafferty, 2009, p. 10] Modelparameters α, β can also be estimated with the help of maximum likelihood estimates based on the marginal likelihood. In this case, the algorithm is extended to a variational EM algorithm. [Blei et al., 2003, p. 1005/1006] More detailed derivations of the exact optimization steps introduced here, are given in Blei et al. 2003, p.1003-1005/Appendix] and Geigle [2016, p. 6-13]. A more general overview of Variational inference can be found in Blei et al. [2017].

Characteristics of Variational Inference

Variational inference is computationally intensive, as it is a *batch learning algorithm* and thus requires solving an optimization problem for **every** document in order to then update dataset-wise variational parameters. [Hoffman et al., 2010, p. 1] Moreover, often computing $\Psi(\cdot)$ function, which is needed in the update equations, has shown to affect performance, as well as memory consumption. However, strategies to moderate computational issues exist. A few practical considerations involve restarting the algorithm multiple times, as finding local maxima tends to depend heavily on the initialization of the parameters. Also, the independence assumption imposed by the variational distribution disregards the strong dependence between latent variables $\boldsymbol{Z}, \Theta, \Phi$. [Blei and Lafferty, 2009, p. 11] According to Teh et al. [2006], this instance can lead to inaccurate estimates of the posterior. To tackle that shortcoming, they introduced a procedure called *Collapsed Variational Bayesian Inference* for LDA. Contrary to usual Variational Bayes, it models dependence of the parameters on the latent variables. Similar to the algorithm in the next section, it takes advantage of the conjugate priors, which allows Φ and Θ to be integrated out. This step implies a weaker independence assumption, which leads to better approximations to the target posterior distribution, as well as faster computation. [Teh et al., 2006] The batch nature of Variational Bayes makes it almost impossible to efficiently analyze large text collections. Therefore an *online Variational Bayes* algorithm has been introduced, too. [Hoffman et al., 2010] Due to an enormous increase in speed, massive text collections do not pose a problem anymore. Also, real time analysis can be handled, as estimation happens on a document basis, after which the document can be discarded without influencing further parameter estimation. [Hoffman et al., 2010, p. 2]

4.2 Collapsed Gibbs Sampling

Additional to deterministic Variational methods, also sampling-based algorithms are drawn on to infer high-dimensional models such as LDA. These algorithms are characterized by the idea of drawing samples from the intractable posterior. The respec-

tive empirical distribution of the samples then approximates the target distribution. In order to get the samples, a Markov chain is used to simulate a high dimensional probability distribution. [Blei, 2012, p. 81] Generally speaking, a Markov chain is a stochastic model describing a sequence of possible events. Whereas the probability of each event solely depends on the state obtained in the previous event. [Brooks et al., 2011, p. 4-6]

When the Markov chain reaches stationarity, meaning that it's statistical properties such as mean, variance and autocorrelation are all constant over time, each transition yields a sample. As the Markov Chain usually strongly depends on the starting parameters, the period before convergence (*burn-in*) is neglected and the respective samples are not considered. In particular, a Markov Chain Monte Carlo method called *Gibbs Sampling* is applied. It stems from the idea, that the distribution to sample from is intractable, but the conditional distribution of a certain variable given all the others can be calculated. Broadly speaking, it iteratively samples each latent variable in turn conditioned on the current values of all of the other latent variables and the data. Thus, a new sample is influenced by the current assignments of all other latent variables. [Steyvers and Griffiths, 2007, p. 7/8]

In terms of LDA, the Gibbs sampler is built over the latent topic assignments $\boldsymbol{Z}$. This is motivated by the fact, that Φ and Θ are *conjugate* and thus can be integrated out of the joint distribution. That way, $p(\boldsymbol{W}, \boldsymbol{Z}|\boldsymbol{\alpha}, \boldsymbol{\beta})$ is used for Gibbs sampling over $\boldsymbol{Z}$, which allows faster convergence of the samples than for the entire joint distribution. With the obtained posterior distribution of the latent topic assignments $\boldsymbol{Z}$, distributions for Φ and Θ can be estimated. Genreally, integrating out the priors is also referred to as *collapsing*, giving the name *Collapsed Gibbs Sampling*. Within the collapsed joint distribution, $\boldsymbol{W}$ is observed. Thus, sampling over each $z_{j,i}$ happens in turn, based on the full conditional containing all other topic assignments $\boldsymbol{z}_{-(j,i)}$. By the definition of conditional probability,

$$p(z_{j,i}|\boldsymbol{z}_{-(j,i)}, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{p(z_{j,i}, \boldsymbol{z}_{-(j,i)}, \boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta})}{p(\boldsymbol{z}_{-(j,i)}, \boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta})} \tag{4.7}$$

$$\propto p(z_{j,i}, \boldsymbol{z}_{-(j,i)}, \boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{W}, \boldsymbol{Z}|\boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{4.8}$$

holds. [Griffiths and Steyvers, 2004, p. 5229/5230; Steyvers and Griffiths, 2007, p. 7/8]

Defining $p(\boldsymbol{W}, \boldsymbol{Z}|\boldsymbol{\alpha}, \boldsymbol{\beta})$ forms the basis for the derivation of an iterative Gibbs sampler over the topic assignments $z_{j,i}$. In order to obtain $p(\boldsymbol{W}, \boldsymbol{Z}|\boldsymbol{\alpha}, \boldsymbol{\beta})$, integrating out Φ and Θ needs to be carried out. Based on the full joint distribution (equation 3.5), this is formulated as such:

$$p(\boldsymbol{W}, \boldsymbol{Z}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_{\Theta} \int_{\Phi} \prod_{k=1}^{K} p(\boldsymbol{\phi}_k|\boldsymbol{\beta}) \prod_{j=1}^{M} p(\boldsymbol{\theta}_j|\boldsymbol{\alpha}) \prod_{i=1}^{N_j} p(z_{j,i}|\boldsymbol{\theta}_j) p(w_{j,i}|\phi_{z_{j,i}}) d\Phi d\Theta \tag{4.9}$$

The first step is to separate the integrals according to dependence on the integration variable,

$$= \int_{\Theta} \prod_{j=1}^{M} p(z_j|\boldsymbol{\theta}_j) p(\boldsymbol{\theta}_j|\boldsymbol{\alpha}) d\Theta \times \int_{\Phi} \prod_{k=1}^{K} p(\boldsymbol{\phi}_k|\boldsymbol{\beta}) \prod_{j=1}^{M} \prod_{i=1}^{N_j} p(w_{j,i}|\phi_{z_{j,i}}) d\Phi \tag{4.10}$$

The next form can be obtained by expanding the terms regarding the Dirichlet priors and using the conjugacy of the multinomial terms.

$$= \left(\frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \right)^{M} \prod_{j=1}^{M} \frac{\prod_{k=1}^{K} \Gamma(n_{k,j,\cdot} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} n_{k,j,\cdot} + \alpha_k)} \times \left(\frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \right)^{K} \prod_{k=1}^{K} \frac{\prod_{v=1}^{V} \Gamma(n_{k,\cdot,v} + \beta_v)}{\Gamma(\sum_{v=1}^{V} n_{k,\cdot,v} + \beta_v)} \tag{4.11}$$

That way, Φ and Θ are gotten rid of by integration and the Gibbs sampler according to equation 4.8 can be derived. Now, the terms only dependent on the hyperparameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and not $z_{j,i}$ can be dropped.

$$p(z_{j,i}, \boldsymbol{z}_{-(j,i)}, \boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta}) \propto \prod_{j=1}^{M} \frac{\prod_{k=1}^{K} \Gamma(n_{k,j,\cdot} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} n_{k,j,\cdot} + \alpha_k)} \times \prod_{k=1}^{K} \frac{\prod_{v=1}^{V} \Gamma(n_{k,\cdot,v} + \beta_v)}{\Gamma(\sum_{v=1}^{V} n_{k,\cdot,v} + \beta_v)} \tag{4.12}$$

In the following steps, terms that do not depend on the current position (j,i) to be estimated, are dropped. Sparing the exact formulations at this point, yields the unnormalized conditional probability for the case of **symmetric priors** α and β:

$$p(z_{j,i} = k|\boldsymbol{z}_{-(j,i)}, \boldsymbol{W}, \alpha, \beta) \propto \frac{n_{k,j,\cdot}^{-(j,i)} + \alpha}{n_{k,j,\cdot}^{-(j,\cdot)} + K\alpha} \times \frac{n_{k,\cdot,w_{j,i}}^{-(j,i)} + \beta}{n_{k,\cdot,\cdot}^{-(j,i)} + V\beta}, \tag{4.13}$$

whereas terms like $n_{z_{j,i},j,\cdot}^{-(j,i)}$ work just as the $n_{\cdot,\cdot,\cdot}$ notation. With the exception, that the counts do not include the current assignment of (j,i). So, for equation 4.13, the first ratio of counts expresses the probability of topic k in document j. The second ratio constitutes the probability of $w_{j,i}$. For the sake of extent, the exact equation transformations and expansions for the preceeding part are omitted, but can be found in Geigle [2016, p. 15-18], Heinrich [2005, p. 19-22] and Carpenter [2010].
Following Steyvers and Griffiths [2007, p. 7-9], the Gibbs sampling algorithm can be summarized as such:

1. At first, the topic to word assignments $\boldsymbol{Z}$ are initialized randomly from $\{1, \ldots, K\}$, which marks the starting state of the Markov chain.

2. For each Gibbs sample (contains the topic assignments for all V words):

   (a) The count matrices incorporating $n_{\cdot,\cdot,\cdot}^{-(j,i)}$ are reduced one for the entries that match the current topic assignment.

   (b) A new topic assignment is drawn from equation 4.13.

(c) According to the new topic assignment, the count matrices are increased by one respectively.

3. Samples drawn during burn-in are thrown away, as they strongly depend on the starting parameters.

4. The subsequent samples approximate the posterior distribution of topic assignments. In order to prevent possible autocorrelation among consecutive samples, they are saved at frequent intervals.

The other latent variables Φ and Θ are subject to many further analyses, therefore requiring estimation from the direct estimates of $\boldsymbol{Z}$ obtained by the Gibbs samples. These estimates can be gained by deploying a point estimate based on the current state of $\boldsymbol{Z}$ in the Markov chain.

$$\widehat{\theta_{j,k}} = \frac{\alpha + n_{k,j,\cdot}}{K\alpha + n_{\cdot,j,\cdot}} \tag{4.14}$$

$$\widehat{\phi_{k,v}} = \frac{\beta + n_{k,\cdot,v}}{V\beta + n_{k,\cdot,\cdot}} \tag{4.15}$$

Conditioned on $\boldsymbol{W}$ and $\boldsymbol{Z}$, both terms in 4.14 and 4.15 equate to the predictive distributions over new (yet unseen) words v and new topics k. [Steyvers and Griffiths, 2007, p. 7/8]

Characteristics of collapsed Gibbs sampling

The algorithm is easily, as well as efficiently implemented, since only count matrices need to be maintained for computing the full conditional distribution. It has also been proven to be competitive to other methods regarding speed and performance. [Griffiths and Steyvers, 2004, p. 5229/5230] Compared to Variational Bayes, it is non deterministic, as sampling is involed. That is, where the issues associated with collapsed Gibbs sampling arise. As hinted above, convergence of the Markov chain needs to be assessed according to some chosen criteria. Then, a suitable choice for the number of burn-in samples needs to be found. After then discarding those burn-in samples, enough iterations should be performed and afterwards thinned out, in order to reduce sampling noise. [Steyvers and Griffiths, 2007, p. 8; Geigle, 2016, p. 19] A trade-off is formed by averaging over many samples, which decreases the noise and improves performance. [Asuncion et al., 2009, p. 32/33] However, if enough iterations are performed, a suitable burn-in is chosen and as described above, autocorrelation is prevented, the algorithm proves to work well. A more thorough comparison concerning speed, complexity and performance of several algorithms can be found in [Asuncion et al., 2009].

To conclude, Asuncion et al. [2009] show that given the right hyperparameter specification, no substantial difference in results among the mentioned algorithms can be observed. Instead, especially with large text collections, characteristics like memory consumption and speed seem more important.

5 Topic Model Evaluation

Up to this point, topic models have been introduced and estimated. However, in order to compare several possible models and their specifications, means of model evaluation are needed. If for instance, the number of topics K is not given beforehand or determined by data, as in Teh et al. [2004], one best value can be found by drawing on evaluation metrics. In general, goals and resources can vary and thus require several performance metrics. Therefore, this chapter points out strategies, according to which topic models can be evaluated. Moreover, newer approaches based on the coherence of the revealed topic words are shown. These are especially justified, as original probabilistic measures tend not to consider the interpretability of the obtained topic words for humans, but rather rely on statistical properties. [Blei, 2012, p. 83] To conclude, model evaluation is essential to choosing one specific model over several possible setups.

5.1 Probabilistic measures

As Blei [2012, p. 83/84] points out, evaluation is usually accomplished by measuring generalization performance of the model. Therefore, a model is learned on a training set and subsequently evaluated on hold-out data. Then, the model reaching best model fit on the hold-out data, according to some defined measure, is chosen. The following metrics are of probabilistic nature, which means that they depend on the underlying model and not on the obtained topic words. An extensive comparison study of held-out evaluation methods specifically for LDA can be found in Wallach et al. [2009].
The first step is to divide the dataset into one training $\mathbf{W_{train}}$ and one test (hold-out) set $\mathbf{W_{test}}$. The test set contains documents $\boldsymbol{w}_j^\star \in \mathbf{W_{test}}$, which have not been used for training the model. Then, performance is assessed by estimating the probability of those hold-out documents. According to this logic, higher probability implies a better underlying model, due to a higher capability to predict new observations, on average. [Wallach et al., 2009, p. 2]

5.1.1 Harmonic mean of log-Likelihood

The first metric was originally used in Griffiths and Steyvers [2004] and involves the log-likelihood in a test set. In terms of LDA, per-document topic distribution Θ is determined while training and thus refers only to training documents. With the help of inference methods, including the ones of section 4, per-word topic distribution Φ and the hyperparameter α can be inferred and used to estimate $p(\boldsymbol{W_{test}}|\Phi, \alpha)$. The resulting probability $p(\boldsymbol{W_{test}}|\Phi, \alpha)$ with given samples for $p(\Phi, \alpha|\boldsymbol{W_{train}})$ hence formulates as

$$p(\boldsymbol{W_{test}}|\Phi, \alpha) = \prod_j p(\boldsymbol{w}_j^\star|\Phi, \alpha). \tag{5.1}$$

Due to independence of the topic assignments of one doccument from all other documents, in the following only one document $\boldsymbol{w}$ is looked at. Carrying on, the probability of hold-out document $\boldsymbol{w}$, also called evaluation probability $p(\boldsymbol{w}|\Phi, \alpha)$, is to be estimated. With the help of Bayes' rule, this probability can be expressed as the normalizing constant to following posterior distribution regarding latent variable $\boldsymbol{Z}$:

$$p(\boldsymbol{Z}|\boldsymbol{w}, \Phi, \alpha) \;=\; \frac{p(\boldsymbol{Z}, \boldsymbol{w}|\Phi, \alpha)}{p(\boldsymbol{w}|\Phi, \alpha)} \tag{5.2}$$

[Wallach et al., 2009, p. 2] In order to estimate $p(\boldsymbol{w}|\Phi, \alpha)$, the harmonic mean method comes into play. [Newton and Raftery, 1994] The estimation step, hinted in equation 5.2, requires samples for topic assignments $\boldsymbol{Z}$ of document $\boldsymbol{w}$. Similar to section 4.2, these can be obtained by performing Gibbs sampling sequentually on the conditional posterior, given $\boldsymbol{w}, \Phi, \alpha$ and the current remaining topic assignments $\boldsymbol{z}_{-s}$. After a reasonable burn-in, S of those samples yield the approximated marginal likelihood

$$p(\boldsymbol{w}|\Phi, \alpha) \approx \frac{1}{\frac{1}{S}\sum_s \frac{1}{p(\boldsymbol{w}|\boldsymbol{z}_s, \Phi)}}. \tag{5.3}$$

[Wallach et al., 2009, p. 2/3]
This is repeated for all held out documents $\boldsymbol{w}_j^\star$ and possibly averaged per documents, if Gibbs sampling was run several times. Thus, the harmonic mean log-probability of all hold-out documents is given by

$$log\, p(\boldsymbol{W}_{test}|\Phi, \alpha) \;=\; \sum_j log\, p(\boldsymbol{w}_j^\star|\Phi, \alpha), \tag{5.4}$$

which is basis for a possible comparison of models. [Wallach et al., 2009, p. 7]
Griffiths and Steyvers [2004] originally conducted this method on the entire corpus, which is helpful if predictive performance is not the main task. For this case, collapsed Gibbs sampling already provides the topic assignments $\boldsymbol{Z}$ and thus is even more practicable to conduct.

All in all, easy implementation, as well as efficiency justify the detailed mention in this thesis. Nevertheless, it proves to be unstable, which is why Wallach et al. [2009] elaborate on further estimation procedures. The author concludes that though most of the presented methods tend to be inaccurate, they are suitable for ranking different models. [Wallach et al., 2009, p. 8]

5.1.2 Perplexity

One of the most prevalent metrics is called *perplexity*. In general, it measures how well a probability distribution or probability model predicts a sample and was originally used for language models. Incorporating the bag-of-words assumption, LDA models probability distributions over sequences of words and thus can be regarded as a unigram language model. [Jurafsky and Martin, 2008, p. 954/955] It describes

the likelihood of held-out documents regarding the trained topic model and with the help of equation 5.4, is formulated as such:

$$perplexity(\boldsymbol{W_{test}}) = exp\left(-\frac{\sum_j log\, p(\boldsymbol{w}_j^{\star}|\Phi,\alpha)}{\sum_j N_j}\right) \tag{5.5}$$

As seen in Equation 5.5, lower values of perplexity indicate high log-likelihood and thus a better representation of the words of the test data. [Blei et al., 2003, p. 16] Similar to harmonic mean, perplexity does not capture the quality of the topics. Consequently, a good perplexity value does not necessarily indicate that the detected topic words are interpretable for humans. Quite contrary, Chang et al. [2009] showed a discrepancy between perplexity and human judgment. That is why, additional evaluation metrics, that consider information of the semantic context between words, are introduced in the following.

5.2 Topic coherence measures

The preceeding methods pose the question, whether most of users really want the model to generalize well in order to apply it to new, unseen data. Since arguably the most prevalent utilization of topic models is the exploration of large text collections in order to gain information on internal structures, this approach might not always be the right choice. In fact, interpretability of the learned topics "could have a far greater impact on the overall value of topic modeling for end-user applications" [Newman et al., 2010a, p. 101]. Therefore, several strategies that focus on human-interpretable topics were developed.

5.2.1 Human evaluation of topic coherence

Chang et al. [2009] were the first to propose an approach to evaluate the obtained latent space quality based on human judgment, in which they rely on two different tasks.

First, the interpretability of topic words can be assessed by a method called *word intrusion.* The l most probable words per topic originating in Φ are mixed with one intruder word, that was not assigned a high probability in that topic. Then, the participating subjects are asked to identify the intruder word. If the original topic words are semantically similar, the intruder should be easily detectable. For instance, a topic on cyber security with the five most probable topic words {*email, security, internet, encryption, hacker*} will likely not include the word {*soccer*} and thus, the intruder is easily found. If however, the topic words are not that coherently compounded, this detection might not, or hardly be possible.

Second, *topic intrusion* evaluates the document-topic decomposition in a similar manner. For one document, the title and a cutout of the entire text is shown. With the help of Θ, the three most probable topics with respective highest probability topic words are given in addition to one intruder topic. Thus, a general way to eval-

uate the quality of the models topic-document assignments is formulated. [Chang et al., 2009, p. 3/4]
They measured precision according to the fraction of correct intruder detections by the subjects. Their key finding was a negative correlation between traditional measures, such as perplexity and their proposed measures of topic quality. So they proved that perplexity can be contradictory to human evaluations in terms of interpretability of the learned topics. [Chang et al., 2009, p. 8] Their work motivated many researchers to develop alternative methods for topic quality evaluation.

5.2.2 Automatic evaluation of topic coherence

Theoretically, human judgment based methods form a gold standard solution to semantic evaluation of topics. However, especially for large corpora, these procedures involve tremendous efforts, as many humans need to be recruited to evaluate possibly hundreds of topics. [Röder et al., 2015, p. 2] In other words, they are usually not practicable, which is why automatic evaluation methods are needed.
Several approaches, that try to measure semantic similarity of high scoring words in a topic, were developed and shall be introduced here. They commonly average a score to measure the degree of relatedness for each pair of words within the top-l words per topic. The basic idea is given by

$$C_{oherence}(T) = \sum_{(w_i,w_j)\in T} score(w_i, w_j), \tag{5.6}$$

whereas $T = \{w_1, \ldots, w_l\}$ denotes a, according to assigned probabilities, ordered set of top words for a particular topic and score represents a coherence score of two words w_i, w_j. A few possible alternatives are mentioned in the following.

UCI measure

The UCI coherence measure was firstly introduced by Newman et al. [2010b] and is based on the so called *Pointwise Mutual Information* (PMI) to score word association. The corresponding UCI score for a set of top-l words of topic T then is:

$$C_{UCI}(T) = \frac{2}{l(l-1)} \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} PMI(w_i, w_j), \tag{5.7}$$

with PMI formulated as such:

$$PMI(w_i, w_j) = log \frac{p(w_i, w_j) + \epsilon}{p(w_i)p(w_j)} \tag{5.8}$$

[Röder et al., 2015, p. 2] The probabilities are based on word frequencies and co-occurrence counts. For instance, $p(w_i)$ represents the probability that word w_i occurs within a document, while $p(w_i, w_j)$ denotes the probability for both w_i and w_j to commonly occur within a document. With the addition of ϵ, log of zero

is prevented. These probabilities can either stem from the text corpus the topic model was estimated on (*intrinsic)*, or another, possibly much larger text corpus (*extrinsic*). In order to prevent the influence of noise or unusual word statistics, the authors recommend to use an external reference corpus. One common choice for this external source is the English version of Wikipedia with all its articles. A window of ten words is moved over all texts, whereas each window position forms a document. These documents then are used to build the (co-)occurence counts of the words. [Newman et al., 2010b, p. 4/5]
For the case of the Wikipedia reference corpus, the required probabilities could be expressed as

$$p(w_i) = \frac{D_{Wikipedia}(w_i)}{D_{Wikipedia}} \qquad \text{and} \qquad p(w_i, w_j) = \frac{D_{Wikipedia}(w_i, w_j)}{D_{Wikipedia}}, \tag{5.9}$$

where $D_{Wikipedia}$ depicts the total number of documents, $D_{Wikipedia}(w_i)$ the frequency of documents containing w_i and $D_{Wikipedia}(w_i, w_j)$ the co-occurence counts for w_i and w_j.

UMass measure

In the following, Mimno et al. [2011] introduced the so called *UMass* metric. It is asymmetric, which means that the order and synonymically the probabilities of the top words, is accounted for. The basic idea is that the occurence of the higher words benefits the occurence of the lower top words. Thus, the probability of any top word to appear should be higher, if a document already contains a higher order top word of the same topic. Therefore, for every word the logarithm of its conditional probability is calculated using every other top word, that has a higher order in the ranking of top words as condition. The probabilities are derived using document co-occurrence counts. [Mimno et al., 2011, p. 265]

$$C_{UMass}(T) = \frac{2}{l(l-1)} \sum_{i=2}^{l} \sum_{j=1}^{i-1} log \frac{p(w_i, w_j) + \epsilon}{p(w_j)}, \tag{5.10}$$

This notation is from [Röder et al., 2015, p. 2], whereas the original publication only performed sums, not means.
Stevens et al. [2012] explored UCI and UMass on an LSA, a non negative matrix factorization and an LDA topic model. According to them, the choice of parameter ϵ influences performance of both measures. They suggest an ϵ value smaller than one, which was originally used in the publications. [Stevens et al., 2012, p. 960]

NPMI

Both Aletras and Stevenson [2013, p. 4] and Röder et al. [2015, p. 7] found that replacing PMI within the UCI measure with its normalized version called *NPMI*, leads

to an increase in performance. The increase was measured in a stronger correlation to human judgment on several datasets. The corresponding coherence measure and respectively the NPMI score of two words then formulatse as

$$C_{NPMI}(T) = \frac{2}{l(l-1)} \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} NPMI(w_i, w_j), \tag{5.11}$$

$$NPMI(w_i, w_j) = \frac{PMI(w_i, w_j)}{-log\left(p(w_i, w_j) + \epsilon\right)}. \tag{5.12}$$

[Aletras and Stevenson, 2013, p. 4]
They also proposed the so called *distributional methods*, which are based on a representation of the top-l words as context vectors. In fact, their method constructs a semantic space for each topic word based on an external or the original data source. One word is then represented as a vector, that consists of the confirmation measures of the word itself along with all the others. Then, the constructed vectors for each topic word can be compared according to some vector similarity measure in order to determine coherence. [Aletras and Stevenson, 2013, p. 3/4] To illustrate this concept, the example of Röder et al. [2015] is adopted. They described a set of top topic words $T = \{game,\ sport,\ ball,\ team\}$. The appropriate context vector for word {game}, according to confirmation measure NPMI, then emerges as

$$\vec{v}_{game} = \{NPMI(game, game), NPMI(game, sport), \tag{5.13}$$

$$NPMI(game, ball), NPMI(game, team)\}$$

After constructing such a vector for every word, coherence is obtained by pairwise calculating a vector similarity measure, like cosine or jaccard similarity and then averaging those scores. [Röder et al., 2015, p. 2/3; Aletras and Stevenson, 2013, p. 3/4]

Framework of coherence measures

Röder et al. [2015] firstly proposed a general framework, that allows the formal description and construction of existing, as well as new coherence measures. The definition of several parts with respective configurations enables the construction of different measures. As seen in Figure 9, there are four major parts. At first, the incoming topic word set t is divided into smaller portions in a step called *Segmentation*. Subsequently, those portions are compared against each other. An example would be the segmentation into word pairs like in the previous methods.
In the next step, word probabilities P are calculated with the help of a reference corpus. Just as in equation 5.9, probabilities depend on (co-)occurence counts at a document level. The document level itself is given by a sliding window. The length

of the window specifies the number of words, by which documents are created when moving over the reference corpus. For instance, in the UCI measure section, documents were created by a window of length ten, that was moved in one-word steps over the entire Wikipedia corpus. These documents then can be used to calculate probabilities, such as $p(w_i)$ or $p(w_i, w_j)$.
The *confirmation measure* step takes the set of word subsets S, as well as the probabilities P, in order to calculate confirmation values φ for respective word pairs, such as the PMI or NPMI score.
Finally, a single coherence score c is aggregated over those values φ. Hence, the coherence of one topic t with respective top words is calculated. [Röder et al., 2015, p. 3-6] All possible specifications and combinations are found in Röder et al. [2015]. Their extensive study additionally revealed two combinations, which outperformed any other measure proposed at that time. The novelty of the two methods, as well as their results are the reason they are mentioned here explicitly.

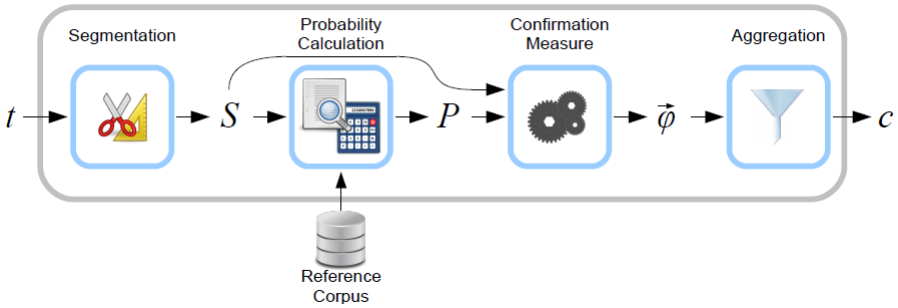


Figure 9: Coherence measure framework according to Röder et al. [2015, p. 4].

C_v measure

They did not show the particular formulation, but rather named its components according to Figure 9. The composition of their proposed C_v measure is specified as such:

- Subsets S are of type S^{one}_{set}, which in their notation, denotes the context vector segmentation as in equation 5.13.
- Probabilities are of type $\mathcal{P}_{sw(110)}$, which means documents underlying the probabilities were obtained by a 110 word long window over the respective reference corpus.
- $\tilde{m}_{cos(nlr,1)}$ describes the confirmation measure used in construction of the vectors, as well as the vector similarity specification. In this case nlr, stands for NPMI confirmation measure in the context vectors. The similarity measure is tied down to the cosine similarity.

In terms of the aforementioned example, there is a context vector for each top word:

$$\{\vec{v}_{game}, \vec{v}_{sport}, \vec{v}_{ball}, \vec{v}_{team}\}. \tag{5.14}$$

The coherence measure C_v is then calculated as:

$$C_v \;=\; \frac{1}{6} \cdot (cos(\vec{v}_{game}, \vec{v}_{sport}) + \cdots + cos(\vec{v}_{ball}, \vec{v}_{team})), \tag{5.15}$$

with the definition of cosine similarity for vectors of length k:

$$cos(\vec{u}, \vec{w}) \;=\; \frac{\sum_{i=1}^{k} u_i \cdot w_i}{\|\vec{u}\|_2 \cdot \|\vec{w}\|_2} \tag{5.16}$$

[Röder et al., 2015, p. 4-7]

$\mathbf{C_p}$ measure

Their second proposed measure is called C_p and incorporates the following specifications:

- Subsets S are of type S^{one}_{pre}, which denotes the same segmentation as in *UMass*. More precisely, an ordered setting, in which words are only compared to preceeding or more likely words.
- Probabilities are of type $\mathcal{P}_{sw(70)}$, which speaks for document creation by a 70 word long window over the respective reference corpus.
- m_f describes a method based on Fitelson [2003]. In it, a single word w_i is evalutated according to all possible subsets of the remaining words, which is denoted by $S(i)$. The pairwise comparison then happens between w_i and with every single subset in $S(i)$. So for instance, pairwise coherence between subset $S(i)_j$ and w_i is then calculated as

  $$m_f(w_i, S(i)_j) \;=\; \frac{p(w_i|S(i)_j) - p(w_i|\neg\, S(i)_j)}{p(w_i|S(i)_j) + p(w_i|\neg\, S(i)_j)} \tag{5.17}$$

  This is repeated for the entire subsets $S(i)$ of every word $w_i \in T$. The coherence is then averaged over all words with respective numbers of comparisons with subsets. [Röder et al., 2015, p. 3-7]

They furtherly found *extrinsic* versions, as well as using the top 10 topic words and not less, to benefit the agreement with human judgment. [Röder et al., 2015, p. 7]

All in all, several ways of evaluating topic models were shown. These can be used for diverse purposes. For instance, a good choice of topic number K can be achieved by performing crossvalidation over perplexity or calculation of coherence measures

for respective models. Also, several different models can be compared. As always in statistical modeling, there does not exist the one and only solution. Hence, depending on the users goals and the data situation, it might be useful to apply several methods to get a hint at which model with respective specification might fit the data accordingly. Topic models tend to produce some useless topics anyway, which is why, it might simply be of interest, whether 50 or 300 topics represent the data more accurately. [Blei, 2012, p.83]

In the following section, possible strategies for both model selection and evaluation are applied on two different data sets.

6 Application

This thesis was originally motivated by an internship at a Munich software startup called functionHR. It specializes in applying people analytics tasks in form of employee questionnaires. Possible tasks involve determination of positive and negative factors for motivation, staffing or employee fluctuation. Analysis happens in a live software tool, that incorporates a wide field of specifications. On the one hand global trends can be detected, but also on the other hand very detailed scrutiny on department or location level is enabled. Open questions are a vital part to these questionnaires. Those involve questions like:

- *What do we have to do better as a company to become faster, more successful and more effective?*
- *What does your team need to work together even better in the future?*
- *What do you wish from your direct manager?*

Topic modeling then comes into play, when an oversight of important themes within the answers is sought. Hence, ideally documents (individual answers) can be assigned to different topics and thus the exploration of the dataset is much clearer than browsing possibly thousands of text answers by hand. More precisely speaking, a visualized overview of prevalent themes is the goal. Afterwards, users can specifically search for all answers regarding one specific topic and subsequently accumulate information. As these answers tend to be rather short and contentually overlapping, possible issues arise. It has been found that traditional topic models need at least around hundreds of words per document in order to yield reasonable results. [Tang et al., 2014] This proved to be true to some extent, as first results of topic models did not really display useful topics. To add a more successful example of topic model application, a text collection of recent newspaper articles of American media outlets was chosen. Due to the fact that coherence measure implementations only work for English corpora yet, German articles were neglected.
So much for the general setting of the application section. It shall illustrate following points:

- Description of data sets
- Procedure to determine a suitable number of topics K
- Means of visualization for both LDA and CTM
- Topic model results for questionnaire data
- Topic model results for newspaper data
- Application of coherence measures in order to detect more and less interpretable topics

- An extra: Comparison of topics regarding two different news outlets

6.1 Data set description

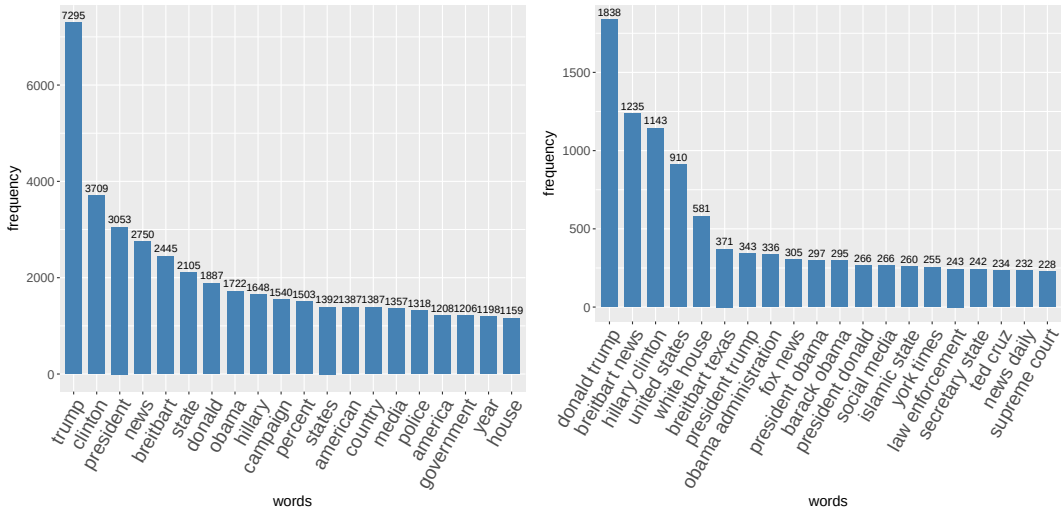
At first, both of the collections require description.

REDACTION

6.1.1 Newspaper article data

The entire newspaper article collection contains 143.000 articles originating from a total of 15 American publication outlets. [Thompson, 2017] This means tremendous computational efforts. Therefore, the dataset was split into one rather liberal subset consisting of New York Times and CNN articles and one rather conservative sourced subset, which is solely based on Breitbart articles. The former amounted to 19.291, the latter to 23.781 articles spanning from 2012 to 2017 with most articles being written in the years 2016 and 2017. Most topic model related publications are based on hundreds of thousands of documents. In order to illustrate that topic models in fact also work for moderately sized collections, as well as due to computational costs, smaller subsets were chosen. For Breitbart, a total of 2.643 and for NY Times / CNN 2.412 articles were analyzed separately.

Breitbart description



(a) 20 most frequent words

(b) 20 most frequent bigrams

Figure 10: Word frequencies for Breitbart newspaper articles

Those newspaper articles are written by journalists and the document length is on average around 350 words for Breitbart and around 750 for NY Times articles (Appendix Figure 22). That is why, not much preprocessing is neccessary. The

steps were the same as with the questionnaire data, but other words needed to be deleted. It seemed, that every article ended with an invitation to follow the author on twitter. Therefore, specific author names along with other words (e.g. *follow, twitter, pamela, key*) were erased. The years of origin become apparant in Figure 10a/10b. In fact, frequencies as well as keywords (Figure 11) seem to be dominated by the 2016 presidential campaign. Especially *Donald Trump*, and terms related to the president are highly frequent, whereas content related terms like (*law enforcement*), (*islamic state*) or (*supreme court*) are in the minority. This is a major difference to the NY Times / CNN data, which holds more content-related words than names. Due to matters of extent, it is found in Appendix B.1.1. The description

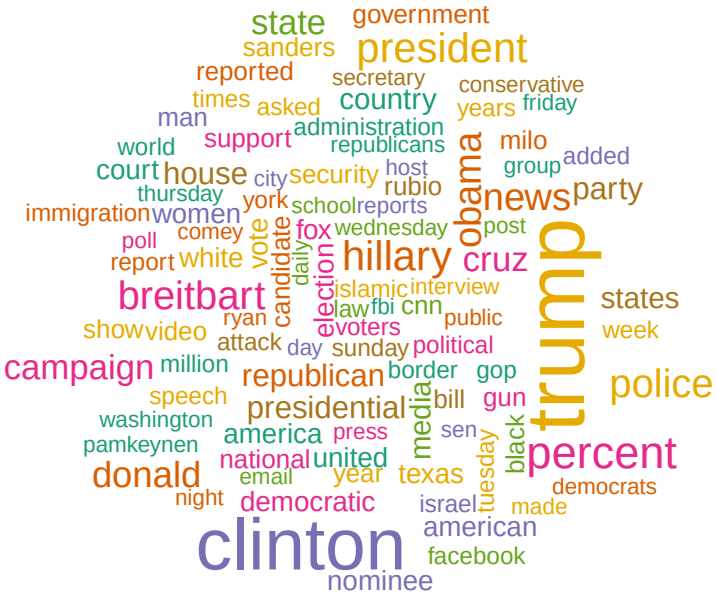


Figure 11: Wordcloud of 50 keywords according to tf-idf weighting of the Breitbart articles

of text collections is vital to topic modeling, as the user obtains a thorough first view on the data. This simplifies preprocessing, as well as presents a first idea of possible structures within the data.

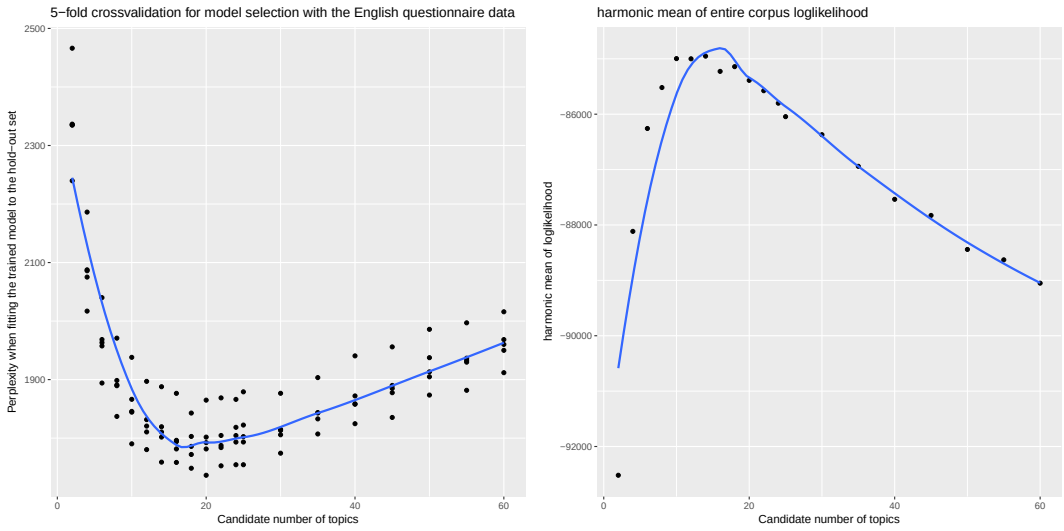
6.2 Choosing a suitable number of topics

The method of Teh et al. [2004] was not applied, thus requiring a strategy to determine a possible hint at which number of topics K conforms with the respective data.

Section 5 entailed a few possible approaches. The original sources often rely on crossvalidation on hold-out data according to perplexity. Therefore, for appropriate candidates of K, LDA models were fitted. Evaluation was performed by a 5-fold crossvalidation according to perplexity [Grün and Hornik, 2011, p. 13] with the addition of the harmonic mean log-likelihood on the entire corpus. [Griffiths and Steyvers, 2004])
Both plots (Figure 12) show the according measure for a number of K different values. The gaps on the right shall display, that larger values are less fitting. Meaning, that perplexity increases and log-likelihood decreases. Especially for large text collections, these gaps save computational efforts, because topic models with large K values tend to take longer.

Exemplary for English answers

The results are visualized in Figure 12. Both perplexity and harmonic mean coincide in the notion that around 10-15 topics should be suitable for the English questionnaire answers.



(a) Perplexity of hold-out data (b) Harmonic mean of the entire corpus

Figure 12: Two strategies to choose number of topics K

The underlying Gibbs sampler was specified with a burnin of 1500, 2000 iterations and possible K values ranged from 1 to 60 with concentration up to K=25.
To conclude, K=12 seems a reasonable choice for the number of topics available for the collection of English questionnaire answers.
Both calculations were done rather rudimentary based on descriptions made in the official publications. Another way to determine optimal number of topics can be achieved with the help of the `ldatuning` package [Nikita, 2016], which incorporates the harmonic mean method, as well as three additional metrics according to which an optimal K can be found. It is not thoroughly mentioned here, as the underlying

metrics are too complex to simply name them. Two of those metrics, are given in [Arun et al., 2010] and [Cao et al., 2009]. However, the results were consistent with the findings of Figure 12.
The same procedure was applied to the German answers and both newspaper article sources and are found in Appendix (Figure 23/24/25). They revealed suitable K values shown in Table 7.

According to it, both questionnaire and newspaper article collections share similar

Dataset	**suitable number of topics K**
English answers	12
German answers	12
Breitbart	150
NY Times/CNN	150

Table 7: Oversight on suitable K values for remaining datasets

suitable values for K. This information is now used to apply topic models to the adressed corpora.

6.3 Means of visualization

In the conclusion, Blei [2012, p. 84] named a few central points, that needed to be worked on in the future. One of them concerns proper means of visualization. When topic extraction is done by analyzing the most probable topic words, especially large corpora and the potentially hundreds of topics imply confusion. One practicable solution is enabled by the `LDAvis` package [Sievert and Shirley, 2015] and the corresponding paper Sievert and Shirley [2014]. It forms an interactive way of discovering the results of the computed models by browsing through topics and their corresponding most probable words. That way, important conclusions can be drawn early on and possibly allow further steps, such as deletion of potential noise words or changes in the models specification. Also, the user gets a quick and easy overview on important themes. An example is given in Figure 13, which illustrates results of a 150 topic LDA model on the NY Times collection. The topics with respective numbers in circles are depicted on the left. On the right, the appropriate most probable words are shown. Furthermore, the number of topic assignments to documents is incorporated by the diameter of the circles. Thus, when browsing through the topics (clicking a circle marks it red), one quickly gets a notion of important words and topics. The two-dimensional visualization furtherly considers similarity of topics concerning the occurence of similar words by multidimensional scaling. In this example, topic 2 is chosen and most frequently contains the terms (*trump, campaign, donald, president, political, washington, presidential, election, america*). Similarly, remaining topics can be browsed and might reveal relevant insights for further analysis.

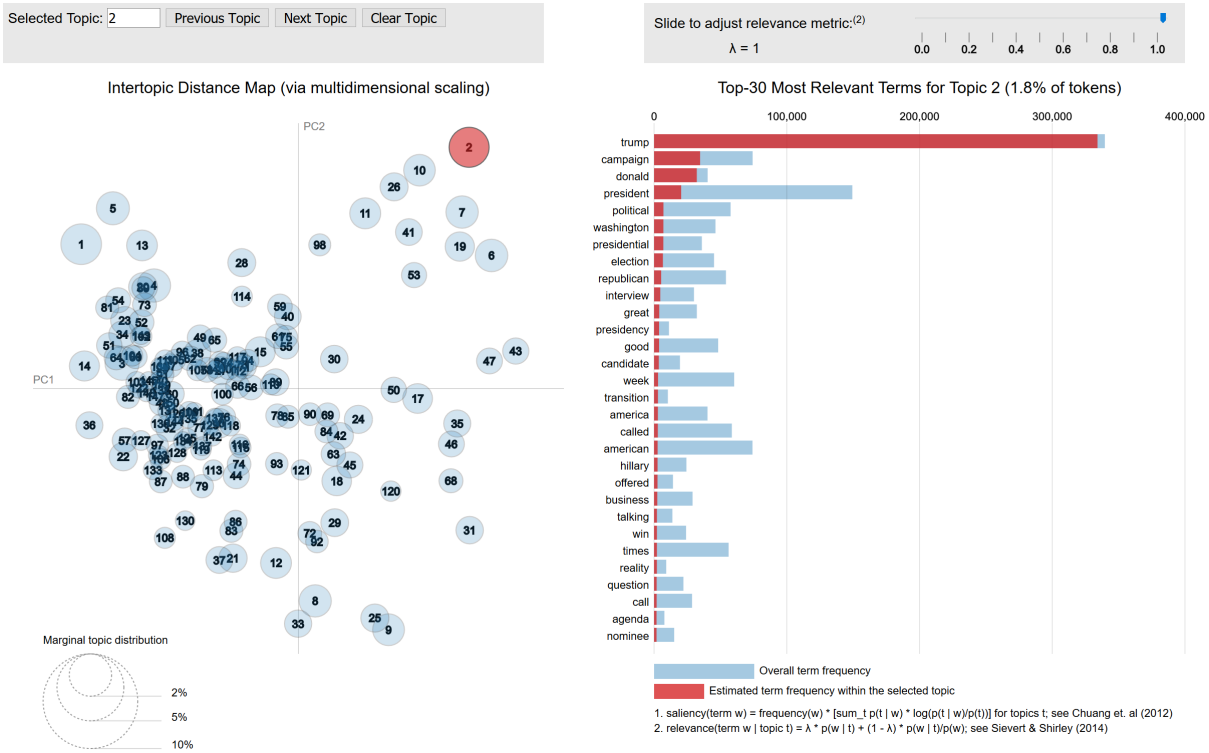


Figure 13: Screenshot of LDA model on the NY Times / CNN data, which is actually an interactive LDAvis interface.

All in all, it forms a great opportunity to get to know the dataset, the topic model results, as well as the underlying structures. Especially for very large text collections, tools for browsing the topic model results are necessary, as printing hundreds of word lists is confusing and thus complicates clear analysis. Also, it can serve as a tool for result presentations due to easy handling.
Another way to visualize results can happen in the form of topic-wordclouds. They are especially helpful, if K is rather small. If not, simply more clouds can be plotted. An example of 12 randomly chosen topics of an 150 topic LDA model on the Breitbart dataset is given in the application section Figure 15. It forms an illustrative way of visualizing the obtained topics, which in fact lacks in interactivity, but is better suited for presentations of results than simple word lists. Also, they consider the estimated word probabilities and therefore enable to visualize importance of words within topics. Thus, it forms a way of displaying otherwise confusing results in an easily understandable graphic. Though there is no real source to topic-wordclouds, the idea is represented in Li and Chua [2017].

Apart from these, researchers have neither settled on a consistent way of visualizing results, nor formulated a coherent workflow for topic models. Much more is the user asked to choose suitable ways to display results.

6.4 Results

This section covers various applications. Though, originally the primary goal was to compare topic models according to different data sets, it changed. Due to unforeseen problems with the CTM model on the larger news data, the outline of this chapter changed to smaller tasks. Those do not only focus on the different topic models, but also consider different aspects of coherence measures. This section begins with a more thorough analysis of the questionnaire data, as it was part of an external project. Afterwards, the news data and especially its LDA models are analyzed with regards to the coherence measures. The underlying text corpora can be characterized by Table 8. The smaller number of ducuments compared to Figure **??** originates in setting the minimum document length. Apparently many answers are simply a few words long and thus not usable.

Dataset	**Number of documents**	**Size of vocabulary**
German answers	709	4080
English answers	893	3432
Breitbart	2643	7728
Breitbart sparse	2643	36810
NY Times/CNN	2410	8161
NY Times/CNN sparse	2411	49028

Table 8: Oversight on document and vocabulary sizes

REDACTION

Concluding, it can be said that some structures can be identified, but the nature of the text collection presumably prevents clearer results. Clear in this context means that many answers share similar themes with only a few words, which are not as contextually dividable, as for instance newspaper articles to separate topics. Furthermore, answers simply are too varying in length to lead to interpretable results.

Similar text collections regarding document length, like twitter corpora, have been researched separately and found not to be too suited for topic models like LDA. [Hong and Davison, 2010; Zhao et al., 2011] The main reason is given by the fact that sparse and short texts do not entail comprehensive word co-occurrence information as longer documents, but inherit imbalance. [Quan et al., 2015] Eventhough some structures can be found, deeper insights require slightly different topic models. Solutions for the projectpartner could be to apply such models especially suited for short texts or to aggregate texts in order to obtain larger documents. Suggestions to the former can be found in Zuo et al. [2016b] or Zuo et al. [2016a], to the latter in Quan et al. [2015]. Also, in terms of the survey framework, a minimum length of texts, spelling correction or answering in one language to increase the number of documents, could be required. Thus, longer documents with possibly more words

about certain topics should lead to clearer structures and that way improve topic extraction. The influence of document length and number of documents is more thoroughly investigated in Tang et al. [2014].

6.4.1 Breitbart articles

As seen in Table 8 and Figure 22, the Breitbart article collection provides an adequate number of documents with sufficient length. Theoretically, those should be contextually dividable, as one article usually is about a small number of themes.
Displaying the entire top topic word list would go beyond the scope of interpretability and not provide clear results. In order to still get an impression about topic quality, two topic-wordclouds for respectively 12 sampled topics are plotted (Figure 14/15). Two different sets of topics are shown to prove that the obtained latent topical structures are representative and not only the best were chosen by hand. Both topic wordclouds reveal that almost all topics seem strongly coherent and meaningful. Only topics 94 in Figure 14 and topic 90 in Figure 15 do not seem to be too related. Looking at the low number of documents compared to most publications, this result is quite impressive.

6.4.2 NY Times / CNN articles

The according LDA model of the NY Times/CNN dataset with its 150 topics is displayed by topic-wordclouds, too. As a matter of extent, those are listed in the Appendix (Figure 26/27). Apart from topic 43 in Figure 26, every top word list seems meaningful. However, it can be observed that some topics mix and then no longer fit together very clearly. For example, topic 40 blends *nuclear war, Philippines* together with *Iran*.
Concluding, both LDA models for news data yield useful topics, even with a manageable amount of documents compared to publications. If more thorough investigations of the results are of interest, the resprective `LDAvis` interfaces can be called up. They are found in the *Interfaces* folder in the supplementary material, which contains the html versions to browse the topic model results.

6.4.3 Coherence measures

This subsection shall demonstrate, how useful coherence measures can be to assess the interpretability of the obtained results. For this purpose, various smaller investigations are made.

Most and least coherent topics within Breibart LDA model

Having seen that topic models on the news data set work well, coherence measures shall be applied to the Breitbart LDA model in order to determine the best and worst topics according to them. With that, their functionality can be evaluated.

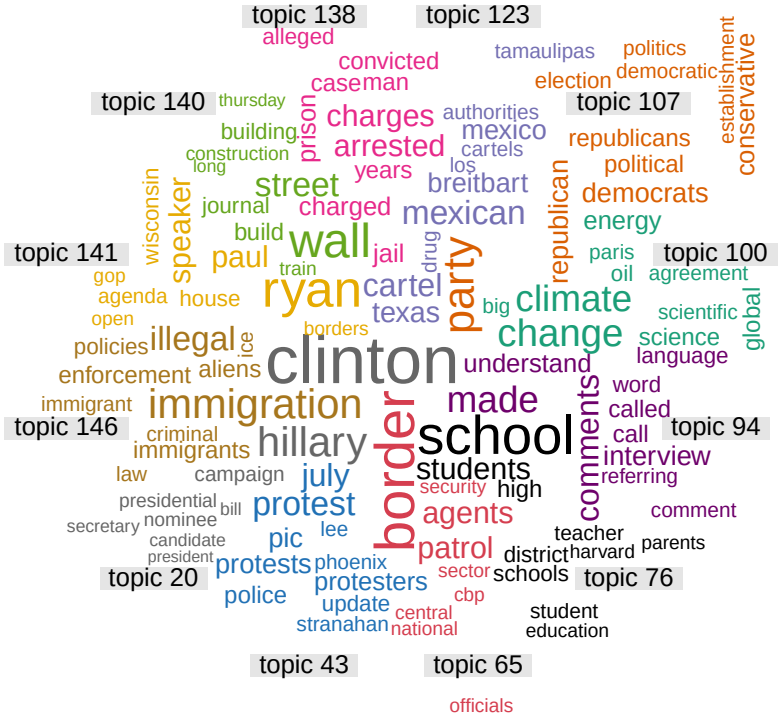


Figure 14: Topic-wordcloud of 12 sample topics of Breitbart LDA model.

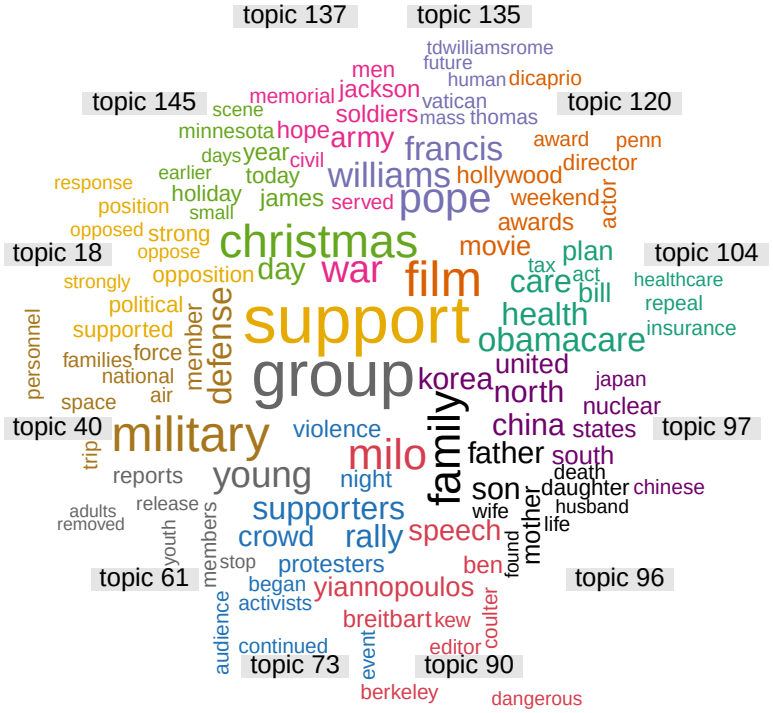


Figure 15: Topic-wordcloud of 12 different sample topics of Breitbart LDA model.

Presenting coherence of 150 topics would not guarantee clarity, so only the 15 highest and lowest scoring topics shall be displayed. First of all, the C_p measure was regarded strongly correlated to human judgment by Röder et al. [2015]. Therefore, its 15 highest and 15 lowest scoring topics are displayed in order to evaluate the performance.
According to Table 9, the most coherent topic within the Breitbart LDA model is described by the following words: *church, christian, christians, religious, god, faith, catholic, religion, freedom, christianity.* This topic in particular and all of the remaining high scoring topics seem strongly coherent, proving that C_p captures coherence. This becomes even clearer when only the lowest-scoring topics are looked at (Table 10). According to it, topic 81 with top words *members, academy, rules, voting, diversity, membership, rule, years, active, straight* should be least cohesive. Those words do not belong together semantically, just as most of the remaining topics in that list. However, there are exceptions like topic 138, 110 or 142 (Figure 10), which at least seem partially useful to humans. Their low score could stem from the external reference corpus. For instance, topic 142 is about *sanders, clinton, bernie, obama, asked, asks, street, points, wall, support*, which probably does not appear often in the Wikipedia reference corpus, as *bernie sanders* only started to appear nationally in 2015 and the puclication of Röder et al. [2015] is dated in that year. Furthermore, many topics contain a few strongly related words mixed with a few random words. All in all, C_p captures coherence quite well.

Comparison of coherence measures on Breitbart LDA model

In order to compare several coherence measures, C_p, C_v, *NPMI, UCI* and *UMass* are plotted against each other. Choosing the entire set of 150 topics would be too confusing, therefore a random sequence of topics is chosen. Before plotting, they undergo scaling to guarantee comparability. The goal is to check for conformity among them and thus assess, whether one coherence score is sufficient or not.
As seen in Figure 16, some simlilarity can be noted, though also differences appear. Especially the measures *NPMI, UCI* and C_p are often resembling. *UMass* is directed similarly, but sometimes differs in amplitude. In total, there are many topics (20, 35, 43, 65, 100, 136, 138, 140, 141), for which most coherence measures are almost equal. It can furtherly be noticed that C_v differs in several topics, sometimes even being directed oppositely. That is surprising, as Röder et al. [2015] found it to be highly correlated to human judgment.
To conclude, this random sequence shows that eventhough most measures conform results, they do not render each other redundant. In fact, it might be useful to consider all of them together in order to derive decisions about model selection or evaluation.
Highest and lowest scoring topics, as well as coherence comparison have also been applied to the NY Times data set and can be found in Appendix B.3.1.

ID	Topic words	C_p measure
46	*church, christian, christians, religious, god, faith, catholic, religion, freedom, christianity*	0.7655783
20	*clinton, hillary, campaign, democratic, presidential, nominee, bill, candidate, secretary, president*	0.7334503
22	*economy, financial, economic, debt, market, growth, bank, crisis, trillion, treasury*	0.7127545
136	*senate, democrats, senator, sen, leader, republicans, president, majority, confirmation, democratic*	0.7108816
104	*obamacare, health, care, bill, plan, insurance, tax, act, repeal, healthcare*	0.5945510
58	*house, bill, president, rep, caucus, congress, committee, news, congressional, leadership*	0.5602333
23	*court, supreme, justice, scalia, president, constitution, gorsuch, judge, law, antonin*	0.5365399
120	*film, movie, awards, hollywood, actor, weekend, director, award, dicaprio, penn*	0.5269883
4	*fbi, investigation, comey, director, general, president, attorney, justice, james, department*	0.5061416
96	*family, father, son, mother, daughter, wife, life, death, found, husband*	0.5031805
66	*court, case, law, judge, decision, order, federal, legal, judicial, courts*	0.4941217
121	*attacks, attack, police, terror, terrorist, security, authorities, islamic, bomb, suicide*	0.4799119
9	*turkey, minister, turkish, party, prime, erdogan, president, government, leader, dutch*	0.4697872
28	*obama, president, administration, barack, national, office, american, congress, presidency, legacy*	0.4655454
107	*party, republican, democrats, conservative, political, republicans, election, democratic, establishment, politics*	0.3987558

Table 9: 15 most coherent topics according to C_p coherence measure

Model selection & evaluation

An example of how to use coherence measures in order to determine a suitable choice for K is given in Moody [2016, p. 4]. The author calculates average C_{NPMI} for each model with different number of topics. Subsequently, the parameter leading to highest coherence is chosen for model specification. In order to illustrate that, the Breitbart LDA models were again calculated separately for various candidates of K and compared for average coherence. According to Table 11, most coherent topics for the Breitbart LDA model are obtained, when using K equal 50. This is not in accordance with the optimal K value obtained by perplexity and log-likelihood. However, it has been shown that probabilistic evaluation measures not necessarily capture coherence. Also, only a limited number of candidates were available.

ID	Topic words	C_p measure
81	*members, academy, rules, voting, diversity, membership, rule, years, active, straight*	-0.4645665
90	*milo, speech, yiannopoulos, ben, breitbart, kew, coulter, editor, berkeley, dangerous*	-0.4601864
138	*arrested, charges, charged, prison, jail, years, man, case, convicted, alleged*	-0.4244879
110	*vote, election, votes, voters, voting, states, electoral, michigan, won, elections*	-0.3996319
109	*pence, mike, vice, flynn, running, kaine, speech, tim, spoke, mate*	-0.3960861
5	*video, shows, head, posted, youtube, released, kill, called, heard, footage*	-0.3676350
43	*protest, july, pic, protests, protesters, police, update, stranahan, lee, phoenix*	-0.3551818
52	*stated, added, ian, ianhanchett, hanchett, msnbc, host, broadcast, argued, responded*	-0.3525861
142	*sanders, clinton, bernie, obama, asked, asks, street, points, wall, support*	-0.3446006
54	*rights, human, letter, wrote, statement, group, act, citizens, respect, international*	-0.3423163
111	*years, year, ago, added, country, weeks, past, level, months, leave*	-0.3385999
38	*pic, https, voters, trump, campaign, news, endorsement, realdonaldtrump, doesn, indiana*	-0.3369365
11	*saudi, arabia, abedin, huma, world, government, years, organization, bin, including*	-0.3239786
61	*group, young, reports, members, release, youth, stop, adults, found, removed*	-0.3227195
29	*drug, medical, war, drugs, cancer, times, good, industry, years, long*	-0.3015580

Table 10: 15 least coherent topics according to C_p coherence measure

Comparison sparse/non sparse collections

Using non sparse data stands for pruning the vocabulary before topic models are applied. Thus, infrequent words with little discriminative power are deleted. Especially for a small number of documents, those lead to noise.
It can be noticed that, except for C_v measure, the sparse versions are less coherent than the pruned versions (Table 12). The reason lies within the low scoring topics. Looking at the according top words reveals that they tend to consist of very uncommon words. That probably leads to very low coherence scores, as the reference corpus does not share those. The high scoring topics on the other hand, do not change noticeably. Respective topic words are not shared in this thesis, but can be reproduced in the according code file. If it is overall of interest to obtain more coherent topics, a manageable extent of pruning might be useful.

6.4.4 Comparison of topics between news datasets

Last but not least, a content comparison is made by presenting the 20 most common topics in both article collections. Taking into account the political orientations of both sources, there might well be differences in relevant topics. According to the

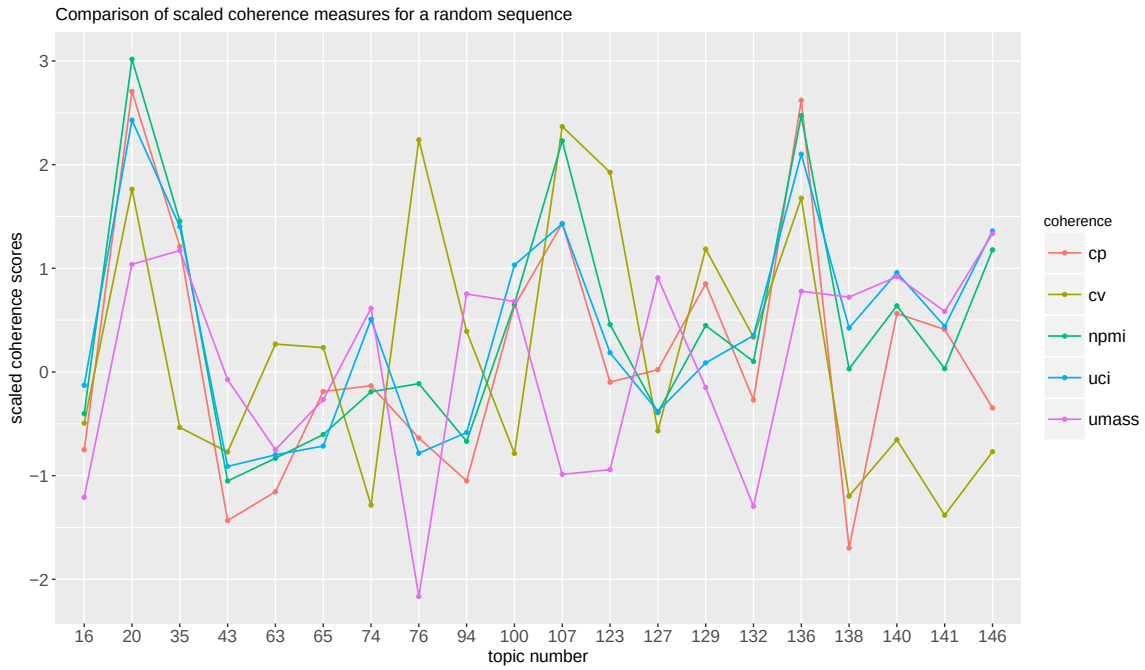


Figure 16: Coherence measures for a random sequence of topics within the Breitbart LDA model.

Measure	**50**	**150**	**250**	**350**	**450**	**550**
C_p	**0.1304**	0.0166	-0.0513	-0.1103	-0.1585	-0.1799
C_v	**0.4084**	0.4025	0.3963	0.3870	0.3851	0.3825
NPMI	**0.0241**	-0.0039	-0.0202	-0.0326	-0.0413	-0.0467
UCI	**-0.5100**	-0.9584	-1.2237	-1.3883	-1.5082	-1.5762
UMASS	**-3.2193**	-3.5646	-3.7349	-3.9412	-3.9657	-3.9499

Table 11: Average coherence scores for top words obtained by Breitbart LDA models for $K \in \{50, 150, 250, 350, 450, 550\}$.

previous chapter, the truncated vocabulary leads to more coherent topics. That is why they underlie the following comparison. Crucial to the topic assignment per document is the highest probability within Θ. In this comparison, one document can only be assigned to one topic according to the highest probability. Therefore, if there are many articles, that are actually assigned to several topics, the comparison is not really representative, but rather a rough overview.

First, one can observe that in both datasets the overall topic assignments are rather broad and not just concentrated on a few. This can be stated by the fact that with a document number of around 2.500, the most frequent topic was assigned only 60 times. In addition to these 20 topics, there are another 130, to which also allocations fall.

The Breitbart collection includes political debates, that have shaped the public debate in the US in recent years. For example, the second most frequent topic

Dataset	$\overline{C_p}$	$\overline{C_v}$	$\overline{NPMI}$	$\overline{UCI}$	$\overline{UMASS}$
Breitbart non sparse	**0.0218**	0.4104	**-0.0065**	**-1.0693**	**-3.7020**
Breitbart sparse	-0.1982	**0.4414**	-0.0635	-2.3316	-5.1134
NY Times non sparse	**0.0618**	0.4141	**0.0169**	**-0.6052**	**-3.4424**
NY Times sparse	-0.1713	**0.4665**	-0.0570	-2.3665	-5.3428

Table 12: Average coherence scores for K=150 LDA models for once sparse once non sparse data with the higher score in bold.

revolves around the protests of black football players, the fourth most common about the second amendment on firearms posture. The refugee situation and especially the dealing in Germany seems to raise tempers in America, too. In addition, themes on the topic of terrorist attacks, Israel and Judaism or the civil war in Syria can be found. The findings from the description are also confirmed, as there are many smaller, more precise issues surrounding the 2016 presidential campaigns.
The results are similar in many ways, which is likely to be related to news being written on a day to day basis. Apart from that, it can be said that the respective political opponent is the most frequently discussed topic. For example, Hillary Clinton and her role as presidential candidate of the democrats was most frequently discussed in the conservative Breitbart collection. The liberals dealt with the role of Republicans in Congress regarding Obamacare.
All in all, most topics with similar topic words can be found in both collections. Since topic models are purely based on frequency relations through the bag-of-words approach, one can not draw any deeper conclusions of content. If it is of interest, to what extent subjects differ between different newspapers, more documents would have to be considered. Also, the period of origin could be limited, to establish a clearer framework.

6.5 Concluding remarks of application

What are the main conclusions that can be drawn from the application section?
First of all, the questionnaire replies of the project partner are not optimally suited to be analyzed with standard topic models. Although some structures can be found and assigned to documents meaningfully, the overall result is not convincing. Possible solutions through special models for short texts as well as possibilities to change the type of texts through alterations within the questionnaire are given in the chapter.
Second, it was confirmed that news articles are very well suited for topic extraction. It is also noteworthy that a manageable number of articles compared to known publications, was enough to deliver highly meaningful topics.
In addition, possibilities to simplify otherwise confusing results by suitable means of visualization were shown. Especially with regard to large data sets, these are

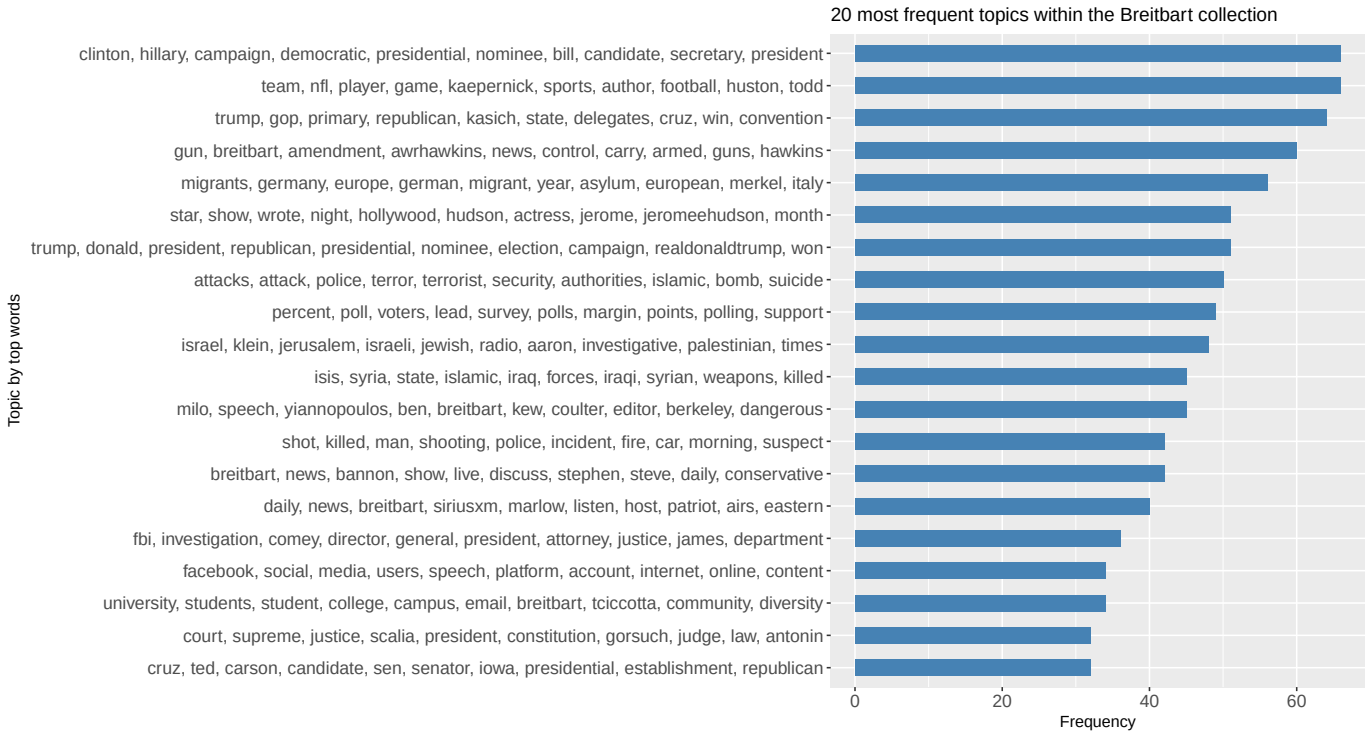


Figure 17: 20 most frequent topics within the Breitbart LDA model.

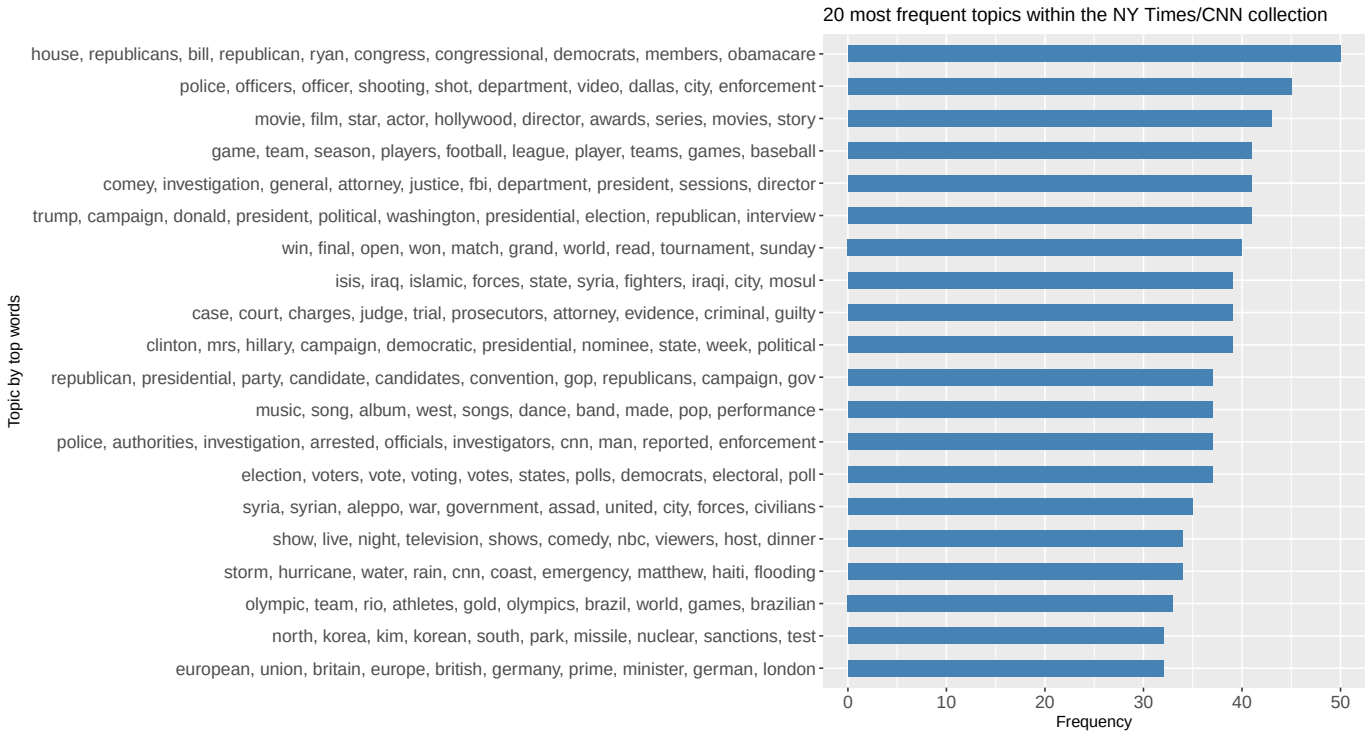


Figure 18: 20 most frequent topics within the NY Times/CNN LDA model.

particularly necessary for the user.
At last, coherence measures have shown that they can be used for obtaining interpretable results as well as for model evaluation. However, since the measures mentioned do not always match, they should be used in combination with each other. In order to achieve more coherent topics, it may also be helpful to trim the vocabulary. Also, esential differences in content between the two news sources could not be observed. Much more, similar topics were often described with the same words and discussed in about the same amount.

7 Conclusion and outlook

A general conclusion in the form of tips and advice for users derived from insights of this thesis shall be briefly summarized here.
Starting with the data preparation, it can be said that there is no uniformly best solution. Rather, it depends on the type and number of texts, whether, for example stemming or other tools should be consulted. In addition, the user must try different specifications. For instance, after a detailed description, various words, that make a clear analysis otherwise difficult, can be removed. This first step is essential to further topic model analysis, especially given small text collections.
Model wise, it can be said that it can be started with the basic LDA model in order to examine then further special models. It helps to be familiar with as many models and their specifications as possible. If it is necessary to determine the number of topics K before the calculation, considerating probabilistic and coherence-based methods is advisable.
If larger collections are available, especially the estimation methods and their efficiency come to the foreground. Thus, in real world applications often speed and memory consumption are important in addition to accuracy. Looking ahead, it can be said that, above all, more complex procedures will probably become the focus of attention. It can be assumed that data sets will continue to become larger and more confusing, thereby requiring finer tools to find more complicated structures. In this regard, there are also already first approaches, which are aimed primarily in the field of deep learning.
Nonetheless, it could be shown that even smaller text collections allow interpretable topic extractions with simpler methods.

References

N. Aletras and M. Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, pages 13–22. Association for Computational Linguistics, 2013.

J. Allaire, K. Ushey, and Y. Tang. *reticulate: Interface to 'Python'*, 2018. URL `https://CRAN.R-project.org/package=reticulate`. R package version 1.7.

R. Arun, V. Suresh, C. E. V. Madhavan, and M. N. N. Murthy. On finding the natural number of topics with latent dirichlet allocation: Some observations. In *Advances in Knowledge Discovery and Data Mining*, pages 391–402. Springer Berlin Heidelberg, 2010.

A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, 2009.

D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012. doi: 10.1145/2133806.2133826.

D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning - ICML 06*. ACM Press, 2006.

D. M. Blei and J. D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35, 2007.

D. M. Blei and J. D. Lafferty. Topic models. In M. Sahami and A. N. Srivastava, editors, *Text Mining: Theory and Applications*, chapter 4, pages 71–93. Taylor and Francis, 2009.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

J. Boyd-Graber, Y. Hu, and D. Mimno. Applications of topic models. *Foundations and Trends in Information Retrieval*, 11(2-3):143–296, 2017.

S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

J. Cao, T. Xia, J. Li, Y. Zhang, and S. Tang. A density-based method for adaptive LDA model selection. *Neurocomputing*, 72(7-9):1775–1781, 2009.

B. Carpenter. Integrating out multinomial parameters in latent dirichlet allocation and naive bayes for collapsed gibbs sampling, 2010. URL https://lingpipe.files.wordpress.com/2010/07/lda1.pdf. [accessed on: 2018-06-07].

J. Chang, S. Gerrish, C. Wang, J. L. Boyd-graber, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc., 2009.

G. Csardi, T. Wormer, M. Ceglowski, J. R. Rideout, and K. S. Johnson. *franc: Detect the Language of Text*, 2015. URL https://CRAN.R-project.org/package=franc. R package version 1.1.1.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

M. Dowle and A. Srinivasan. *data.table: Extension of 'data.frame'*, 2018. URL https://CRAN.R-project.org/package=data.table. R package version 1.11.0.

I. Feinerer, K. Hornik, and D. Meyer. Text mining infrastructure in r. *Journal of Statistical Software*, 25(5):1–54, March 2008. URL http://www.jstatsoft.org/v25/i05/.

I. Fellows. *wordcloud: Word Clouds*, 2014. URL https://CRAN.R-project.org/package=wordcloud. R package version 2.5.

B. Fitelson. A probabilistic theory of coherence. *Analysis*, 63(3):194–199, 2003.

C. Geigle. Inference methods for latent dirichlet allocation, 2016. URL http://times.cs.uiuc.edu/course/598f16/notes/lda-survey.pdf. [accessed on 2018-07-19].

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.

B. Grün and K. Hornik. topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, 40(13):1–30, 2011.

J. He, Z. Hu, T. Berg-Kirkpatrick, Y. Huang, and E. P. Xing. Efficient correlated topic modeling with topic embedding. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 17*. ACM Press, 2017.

G. Heinrich. Parameter estimation for text analysis, 2005. URL http://www.arbylon.net/publications/text-est.pdf. [accessed on 2018-07-26].

M. D. Hoffman, D. M. Blei, and F. Bach. Online learning for latent dirichlet allocation. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 856–864, 2010.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR 99*. ACM Press, 1999.

L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics - SOMA 10*. ACM Press, 2010.

Y. Imai. Mixture of topic models for analyzing short text documents with user information. Master's thesis, Nara Institute of Technology, 2016. URL `https://library.naist.jp/mylimedio/dllimedio/showpdf2.cgi/DLPDFR012578_P1-47`. [accessed on 2018-07-27].

M. I. Jordan. Graphical models. *Statistical Science*, 19(1):140–155, 2004.

D. Jurafsky and J. H. Martin. *Speech and Language Processing, 2nd Edition*. Prentice Hall, 2008.

S. Li and T.-S. Chua. Document visualization using topic clouds. *ArXiv e-prints*, 2017. URL `https://arxiv.org/abs/1702.01520`. [accessed on 2018-07-31].

C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT University Press Group Ltd, 2001.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119. Curran Associates Inc., 2013.

D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 262–272, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

C. E. Moody. Mixing dirichlet topic models and word embeddings to make lda2vec. *CoRR*, 2016. URL `http://arxiv.org/abs/1605.02019`. [accessed on 2018-07-31].

D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the*

North American Chapter of the Association for Computational Linguistics, pages 100–108. Association for Computational Linguistics, 2010a.

D. Newman, Y. Noh, E. Talley, S. Karimi, and T. Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries - JCDL 10*. ACM Press, 2010b.

M. Newton and A. Raftery. Approximate bayesian inference by the weighted likelihood bootstrap. 56:3 – 48, 1994.

M. Nikita. *ldatuning: Tuning of the Latent Dirichlet Allocation Models Parameters*, 2016. URL `https://CRAN.R-project.org/package=ldatuning`. R package version 0.2.0.

J. Ooms. *cld3: Google's Compact Language Detector 3*, 2017. URL `https://CRAN.R-project.org/package=cld3`. R package version 1.0.

C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, 61 (2):217 – 235, 2000.

M. Ponweiser. Latent dirichlet allocation in r. Master's thesis, Institute for Statistics and Mathematics, WU (Wirtschaftsuniversitat Wien), Austria, 2012.

X. Quan, C. Kit, Y. Ge, and S. J. Pan. Short and sparse text topic modeling via self-aggregation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2270–2276. AAAI Press, 2015.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL `http://www.R-project.org/`.

M. Röder, A. Both, and A. Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM 15*. ACM Press, 2015.

T. Rinker. Topic models learning and r resources, 2016. URL `https://github.com/trinker/topicmodels_learning`. [accessed on 2018-07-22].

M. Roberts, B. Stewart, D. Tingley, and E. Airoldi, editors. *The structural topic model and applied social science*, 2013.

M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494. AUAI Press, 2004.

D. Sarkar. *Text Analytics with Python*. APRESS L.P., 2016.

A. Sidorova, N. Evangelopoulos, J. S. Valacich, and T. Ramakrishnan. Uncovering the intellectual core of the information systems discipline. *MIS Quarterly*, 32(3): 467–482, 2008.

C. Sievert and K. Shirley. *LDAvis: Interactive Visualization of Topic Models*, 2015. URL `https://CRAN.R-project.org/package=LDAvis`. R package version 0.3.2.

C. Sievert and K. E. Shirley. LDAvis: A method for visualizing and interpreting topics. *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 1–8, June 2014.

J. Silge and D. Robinson. tidytext: Text mining and analysis using tidy data principles in r. *JOSS*, 1(3), 2016. URL `http://dx.doi.org/10.21105/joss.00037`.

K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 952–961, 2012.

M. Steyvers and T. Griffiths. *Latent Semantic Analysis: A Road to Meaning*, chapter Probabilistic topic models. Laurence Erlbaum, 2007.

J. Tang, Z. Meng, X. Nguyen, Q. Mei, and M. Zhang. Understanding the limiting factors of topic modeling via posterior contraction analysis. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages I–190–I–198. JMLR.org, 2014.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

Y. W. Teh, D. Newman, and M. Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 1353–1360, 2006.

A. Thompson. 143.000 articles from 15 american publications, 2017. URL `https://www.kaggle.com/snapcrack/all-the-news`. [accessed on 2018-06-22].

G. van Rossum. Python tutorial, Technical Report CS-R9526. Technical report, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, 1995.

H. M. Wallach. Topic modeling: beyond bag-of-words. In *NIPS 2005 Workshop on Bayesian Methods for Natural Language Processing*, 2005.

H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML 09*. ACM Press, 2009.

H. Wickham. *tidyverse: Easily Install and Load the 'Tidyverse'*, 2017. URL `https://CRAN.R-project.org/package=tidyverse`. R package version 1.2.1.

W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Lecture Notes in Computer Science*, pages 338–349. Springer Berlin Heidelberg, 2011.

Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu, and H. Xiong. Topic modeling of short texts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD16*, 2016a.

Y. Zuo, J. Zhao, and K. Xu. Word network topic model: A simple but general solution for short and imbalanced texts. *Knowl. Inf. Syst.*, 48(2):379–398, 2016b.

Appendices

A Inference

A.1 Basic derivation of Variational Inference for unsmoothed LDA

Whereas [Blei and Lafferty, 2009] only mention the smoothed version, [Blei et al., 2003] and [Geigle, 2016] give a deeper derivation of the unsmoothed case. All in all, the distributions of both cases are pretty similar. Due to greater detail in the derivation of the unsmoothed version, it is shown here.
Both variational and true posterior distributions are fully factorized across documents. Optimizing the parameters for each document successively will optimize the distribution as a whole. That is, why for the sake of clarity, it will be focussed on the document level, annotated by index j here.
The KL divergence between q and the true posterior $p(\Theta, \mathbf{Z}|\mathbf{W}, \alpha, \phi)$ is given by

$$KL(q \,\|\, p) = \int_{\theta_j} \sum_{\boldsymbol{z_j}} q(\boldsymbol{z_j}, \theta_j|\gamma_j, \pi_j) \, log \, \frac{q(\boldsymbol{z_j}, \theta_j|\gamma_j, \pi_j)}{p(\boldsymbol{z_j}, \theta_j|\boldsymbol{w}_j, \alpha, \boldsymbol{\phi})} d\theta_j \tag{A.1}$$

$$= \mathbb{E}_q(log \, q(\boldsymbol{z}_j, \theta_j|\gamma_j, \pi_j)) \, - \, \mathbb{E}_q(log \, p(\boldsymbol{z}_j, \theta_j|\boldsymbol{w}_j, \alpha, \boldsymbol{\phi}))$$

$$= \mathbb{E}_q(log \, q(\boldsymbol{z}_j, \theta_j|\gamma_j, \pi_j)) \, - \, \mathbb{E}_q(log \, p(\boldsymbol{w}_j, \boldsymbol{z}_j, \theta_j|\alpha, \boldsymbol{\phi}) \, + \, log \, p(\boldsymbol{w}_j|\alpha, \boldsymbol{\phi}). \tag{A.2}$$

Drawing on Jensen's inequality, the log likelihood of a document $p(\boldsymbol{w}_j|\alpha, \boldsymbol{\phi})$ can be bound.

$$log \, p(\boldsymbol{w}_j|\alpha, \Phi) = log \int \sum_{\boldsymbol{z}_j} p(\boldsymbol{w_j}, \boldsymbol{z_j}, \theta_j|\alpha, \Phi) d\theta_j \tag{A.3}$$

$$= log \int \sum_{\boldsymbol{z_j}} \frac{p(\boldsymbol{w_j}, \boldsymbol{z_j}, \theta_j|\alpha, \Phi) q(\boldsymbol{z}_j, \theta_j|\gamma_j, \pi_j)}{q(\boldsymbol{z}_j, \theta_j|\gamma_j, \pi_j)} d\theta_j \tag{A.4}$$

$$\geq \int \sum_{\boldsymbol{z_j}} q(\boldsymbol{z_j}, \theta_j|\gamma_j, \pi_j) \, log \, \frac{p(\boldsymbol{w_j}, \boldsymbol{z_j}, \theta_j|\alpha, \Phi)}{q(\boldsymbol{z_j}, \theta_j|\gamma_j, \pi_j)} d\theta_j \tag{A.5}$$

$$= \mathbb{E}_q \, log \, p(\boldsymbol{w_j}, \boldsymbol{z_j}, \theta_j|\alpha, \Phi) \, - \, \mathbb{E}_q \, log \, q(\boldsymbol{z_j}, \theta_j|\gamma_j, \pi_j). \tag{A.6}$$

As a matter of fact, for any variational distribution $q(\boldsymbol{z_j}, \theta_j|\gamma_j, \pi_j)$, Jensen's inequality thus provides a lower bound on the log-likelihood. Let

$$\mathcal{L}(\gamma_j, \pi_j|\alpha, \Phi) = \mathbb{E}_q(log \, p(\boldsymbol{w_j}, \boldsymbol{z_j}, \theta_j|\alpha, \Phi)) \, - \, \mathbb{E}_q(log \, (q(\boldsymbol{z_j}, \theta_j|\gamma_j, \pi_j)), \tag{A.7}$$

which results in another representation of $log\, p(\boldsymbol{w_j}|\alpha, \Phi)$ as follows:

$$log\, p(\boldsymbol{w_j}|\alpha, \Phi) \;=\; \mathcal{L}(\gamma_j, \pi_j|\alpha, \Phi) \;+\; KL(q(\boldsymbol{z_j}, \theta_j|\gamma_j, \pi_j)\|p(\boldsymbol{z_j}, \theta_j|\boldsymbol{w_j}, \alpha, \Phi)) \quad \text{(A.8)}$$

According to equation A.8, maximizing the lower bound $\mathcal{L}(\gamma_j, \pi_j|\alpha, \Phi)$ according to γ_j and π_j, is equivalent to minimizing the aforementioned KL divergence. In order to obtain a version of the lower bound suitable for maximization, it needs to be given with model and variational parameters. At first, the factorizations of p and q are used.

$$\mathcal{L}(\gamma_j, \pi_j|\alpha, \Phi) \;=\; \mathbb{E}_q(log\, p(\theta_j|\alpha)) \;+\; \mathbb{E}_q(log\, p(\boldsymbol{z_j}|\theta_j)) \;+\; \mathbb{E}_q(log\, p(\boldsymbol{w_j}|\boldsymbol{z_j}, \Phi))$$

$$-\; \mathbb{E}_q(log\, (q(\theta_j|\gamma_j)) \;-\; \mathbb{E}_q(log\, q(\boldsymbol{z_j}|\pi_j)) \quad \text{(A.9)}$$

The next step involves expanding these expectations under q according to the variational and true parameters. Detailed derivations of these steps are found in [Blei et al., 2003, p. 1019-1021] and [Geigle, 2016, p. 6-11]. After $\mathcal{L}(\gamma_j, \pi_j|\alpha, \Phi)$ is specified exactly, constrained maximation according to the variational parameters can be conducted. Therefore respective terms containing the interested parameters are isolated and the appropriate derivation is set zero. Applying Lagrange multipliers, the variational parameter $\pi_{j,i,k}$ is solved by

$$\pi_{j,i,k} \;\propto\; \phi_{k,w_{j,i}}\, exp\left\{\Psi(\gamma_{j,k}) \;-\; \Psi\left(\sum_{l=1}^{K}\gamma_{j,l}\right)\right\}. \quad \text{(A.10)}$$

Appropriately, differentiating the exact form of $\mathcal{L}(\gamma_j, \pi_j|\alpha, \Phi)$ with respect to $\gamma_{j,i}$ and afterwards setting zero yields

$$\gamma_{j,i} \;=\; \alpha_i \;+\; \sum_{i=1}^{N_j}\pi_{j,i,k}. \quad \text{(A.11)}$$

Iteratively updating both of these terms is done, until convergence is met. [Blei et al., 2003, p. 1019-1022]

A.2 Update equations of the Variational Inference algorithm

One iteration of mean field variational inference algorithm

1. For each topic k and term v:

$$\lambda_{k,v}^{(t+1)} \;=\; \beta \;+\; \sum_{j=1}^{M} \sum_{i=1}^{N_j} \pi_{j,i,k}\, \mathbb{1}(w_{j,i} = v) \pi_{i,k}^{(t)}. \tag{A.12}$$

2. For each document j:

   - Update γ_j:

$$\gamma_{j,k}^{(t+1)} \;=\; \alpha_k \;+\; \sum_{i=1}^{N_j} \pi_{j,i,k}^{(t)}. \tag{A.13}$$

   - For each word i, update $\boldsymbol{\pi}_{j,i}$:

$$\pi_{j,i,k}^{(t+1)} \;\propto\; exp\left\{ \Psi(\gamma_{j,k}^{(t+1)}) \;+\; \Psi(\lambda_{k,w_i}^{(t+1)}) \;-\; \Psi\left(\sum_{v=1}^{V} \lambda_{k,v}^{(t+1)} \right) \right\}, \tag{A.14}$$

     with Ψ denoting the digamma function, the first derivative of the $log\,\Gamma$ function. It occurs, when differentiating $\mathbb{E}_q(log\, p)$, which contains the log of a Dirichlet distribution, in which the Gamma function appears.

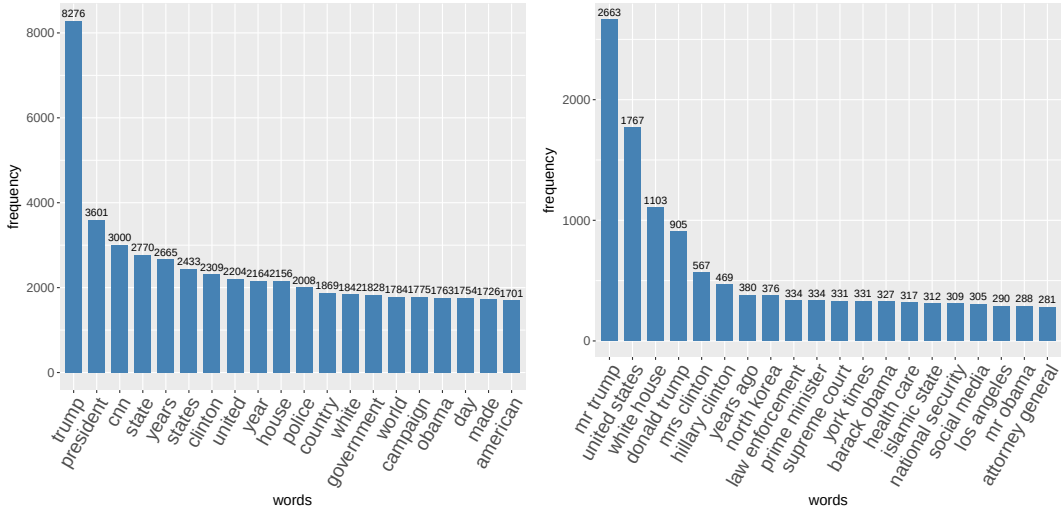
Figure 19: Based on [Blei and Lafferty, 2009, p. 10]

B Application

B.1 Preprocessing

REDACTION

B.1.1 NY Times description



(a) (b)

Figure 20: Word frequencies for NY Times / CNN newspaper articles

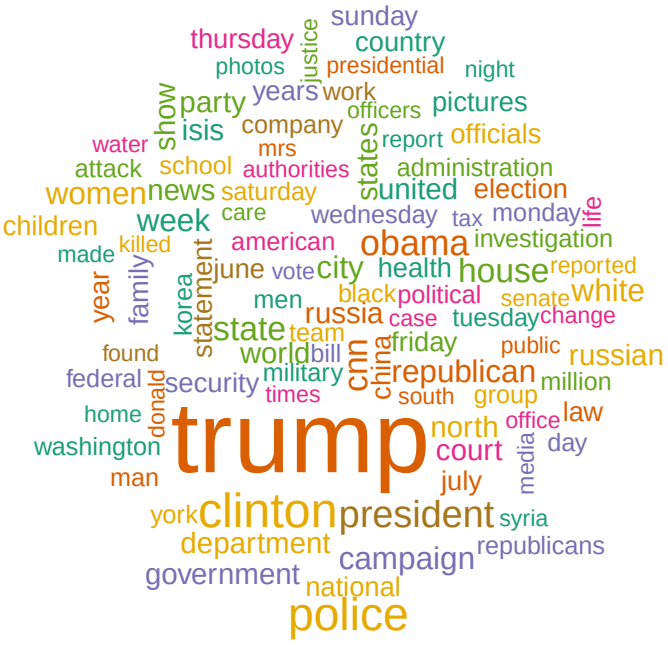


Figure 21: Keyword wordcloud

B.1.2 Document lengths

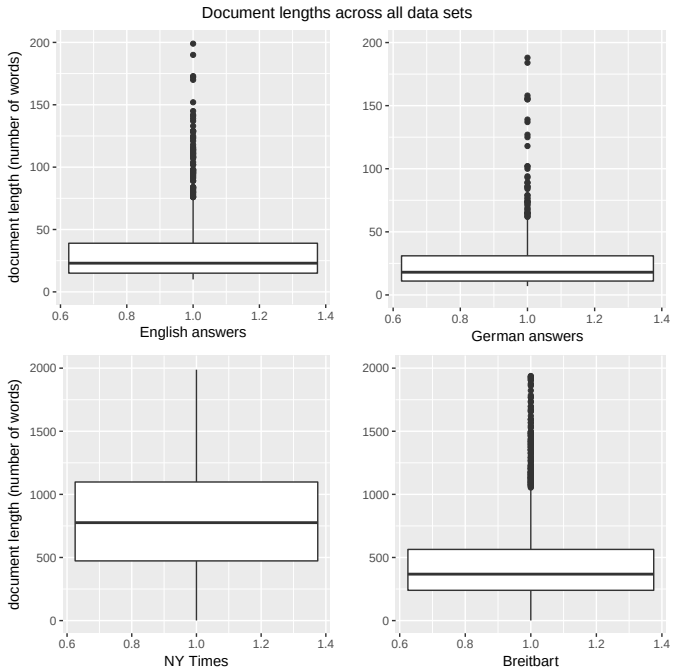
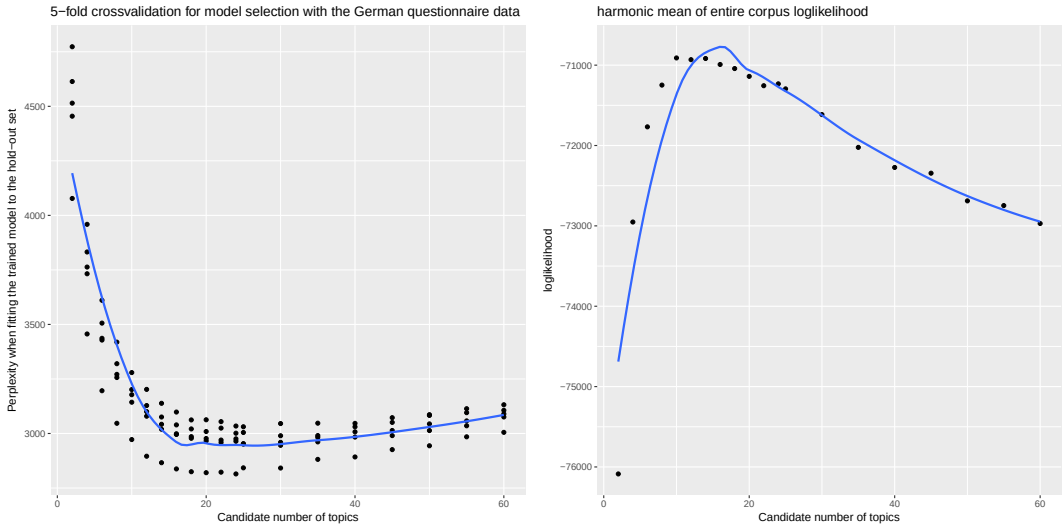


Figure 22: Document lengths across all data sets with different scales (questionnaire: ylim = 200; newspaper: ylim = 2000)

B.2 Finding suitable K

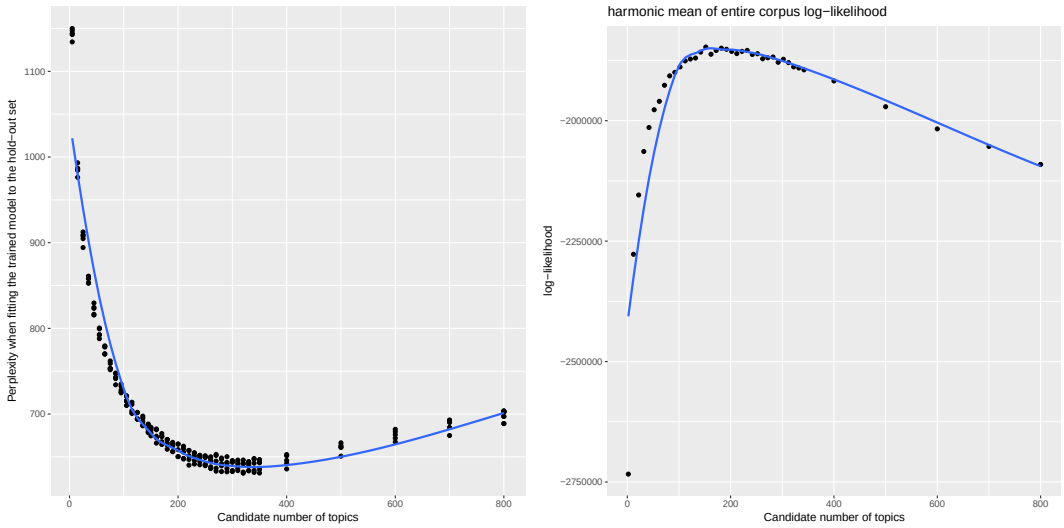
German questionnaire answers



(a) Perplexity of hold-out data (b) Harmonic mean of the entire corpus

Figure 23: Two strategies to choose number of topics K

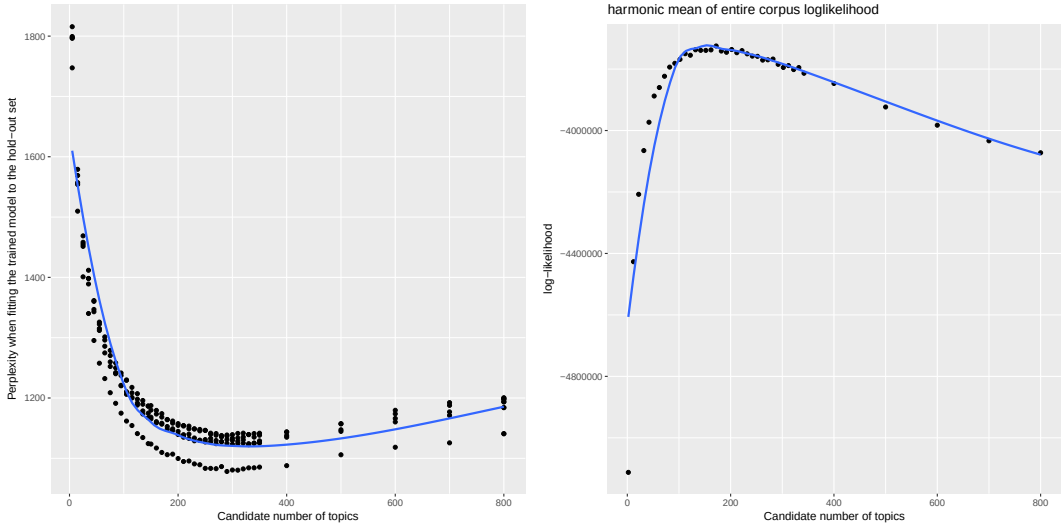
Breitbart articles



(a) Perplexity of hold-out data (b) Harmonic mean of the entire corpus

Figure 24: Two strategies to choose number of topics K

NY Times / CNN articles



(a) Perplexity of hold-out data (b) Harmonic mean of the entire corpus

Figure 25: Two strategies to choose number of topics K

B.3 Results

REDACTION

B.3.1 NY Times / CNN collection

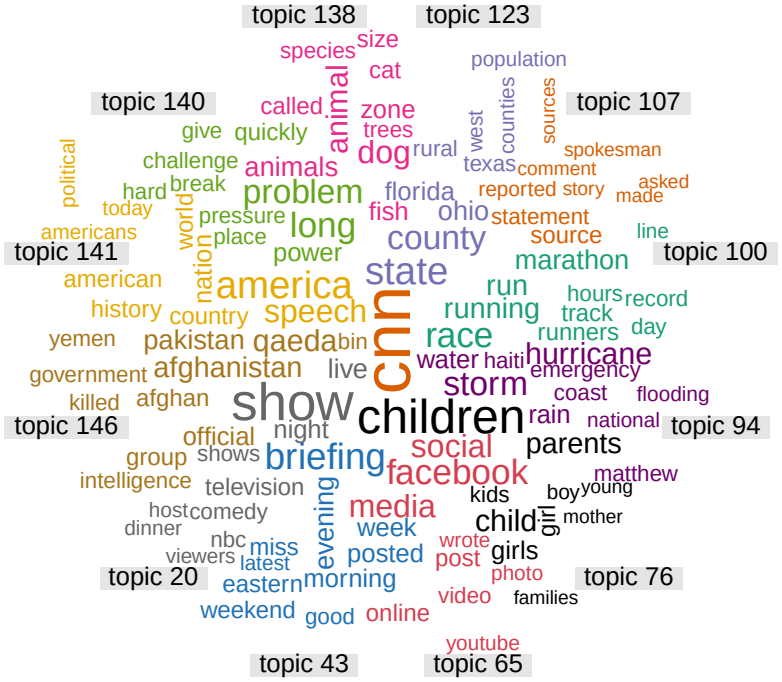


Figure 26: Topic-wordcloud of 12 sample topics of NY Times/CNN LDA model.

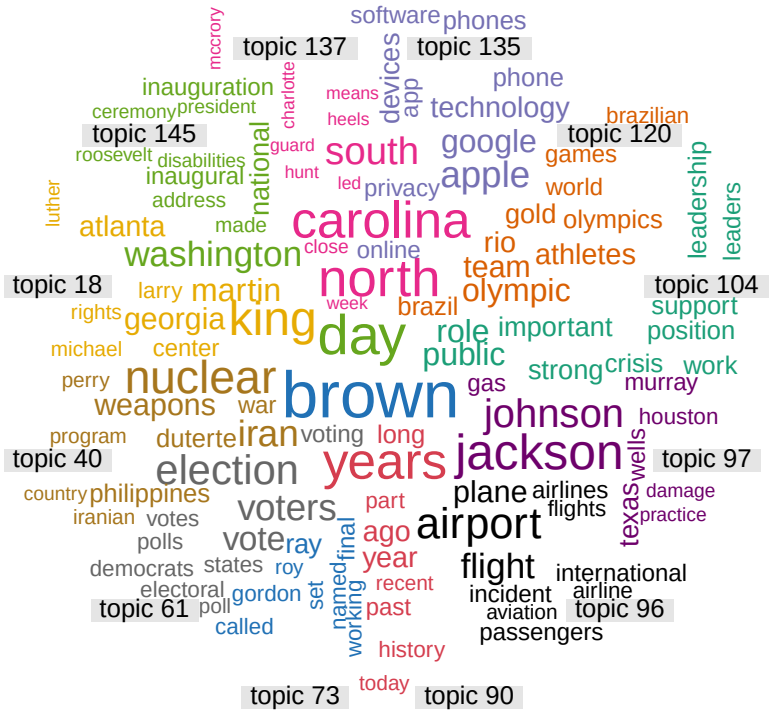


Figure 27: Topic-wordcloud of 12 sample topics of NY Times/CNN LDA model.

ID	Topic words	C_p measure
91	*court, supreme, justice, law, judge, ruling, case, federal, decision, legal*	0.7490736
101	*senate, senator, democrats, sen, secretary, committee, confirmation, republican, democrat, senators*	0.6296885
146	*qaeda, afghanistan, pakistan, official, afghan, group, bin, yemen, government, intelligence*	0.5833359
147	*space, mission, nasa, station, earth, planet, system, crew, solar, rocket*	0.5703616
51	*obama, president, office, barack, bush, george, administration, washington, presidents, white*	0.5703022
31	*family, father, home, mother, life, son, daughter, wife, husband, brother*	0.5502223
5	*muslims, muslim, islamic, islam, mosque, christian, religion, religious, young, center*	0.5497254
117	*republican, presidential, party, candidate, candidates, convention, gop, republicans, campaign, gov*	0.5360158
108	*russia, russian, putin, moscow, russians, election, vladimir, ukraine, president, officials*	0.5221561
132	*june, january, december, april, july, february, november, october, march, august*	0.5116155
130	*game, team, season, players, football, league, player, teams, games, baseball*	0.4484696
77	*church, god, long, pope, francis, catholic, pastor, jesus, christians, give*	0.4440821
72	*clinton, mrs, hillary, campaign, democratic, presidential, nominee, state, week, political*	0.4285030
74	*china, chinese, trade, beijing, india, country, foreign, international, world, economic*	0.4276316
106	*castro, israel, cuba, cuban, israeli, president, netanyahu, minister, prime, peace*	0.4101590

Table 13: 15 most coherent topics according to C_p coherence measure for the NY Times data set

ID	Topic words	C_p measure
112	*wanted, felt, asked, day, made, years, knew, life, room, friends*	-0.7190845
68	*died, cnn, death, family, victims, dead, lost, friends, man, life*	-0.5003572
30	*group, members, groups, called, organization, director, university, organizations, video, member*	-0.4562720
17	*drug, drugs, fda, taking, pills, prescription, heroin, reported, opioid, working*	-0.4399585
71	*comments, asked, wednesday, interview, added, made, remarks, criticism, appeared, statement*	-0.4090203
62	*test, testing, shoes, tests, agency, tested, shoe, move, sound, air*	-0.4033336
80	*questions, answer, question, dylan, answers, names, academy, sweden, prize, nobel*	-0.3819936
131	*documents, wikileaks, document, emails, release, material, released, manafort, published, interview*	-0.3673014
58	*work, job, working, find, good, worked, career, older, years, experience*	-0.3440912
103	*report, marijuana, found, medical, young, colorado, products, cnn, noted, smoking*	-0.3321504
48	*protesters, supporters, crowd, protest, protests, march, movement, event, town, thousands*	-0.3134011
45	*order, ban, executive, travel, countries, refugees, country, trump, united, states*	-0.3047571
15	*moore, account, baker, accounts, confirm, cox, flag, won, haley, issue*	-0.2960242
46	*fight, ali, speed, minor, ring, clay, muhammad, began, boxing, mississippi*	-0.2838021
143	*police, authorities, investigation, arrested, officials, investigators, cnn, man, reported, enforcement*	-0.2785544

Table 14: 15 least coherent topics according to C_p coherence measure for the NY Times data set

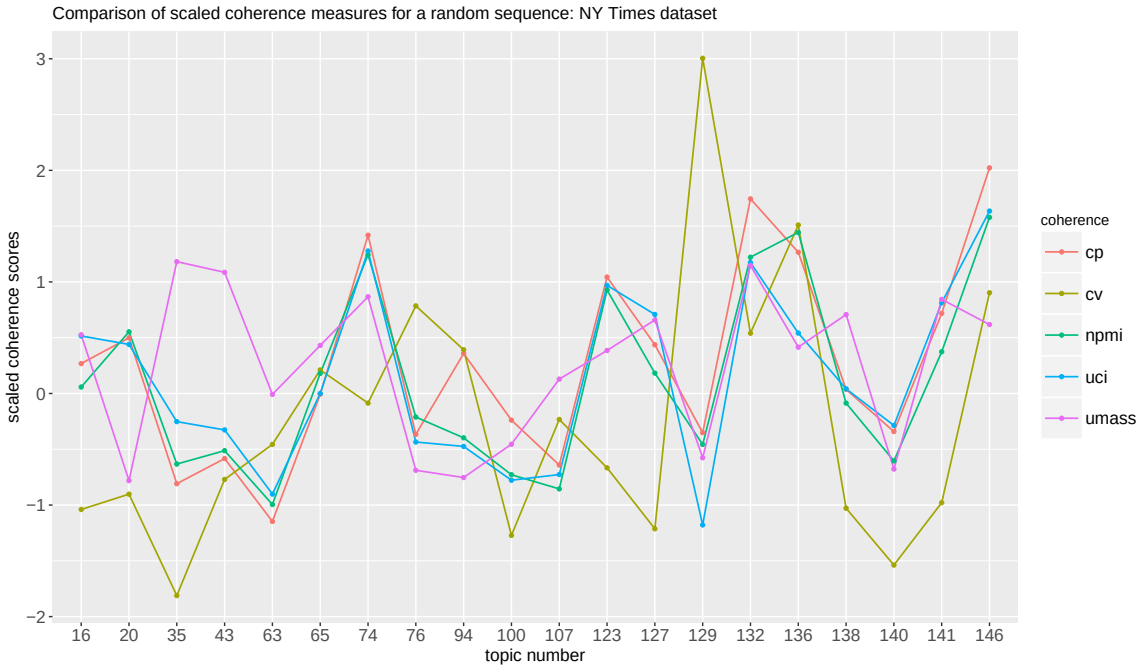


Figure 28: Coherence measures for random sequence of topics within the NY Times LDA model

C Supplementary Material

Analysis mostly happened in the open source software `R` [R Core Team, 2013], as well as some `Python` [van Rossum, 1995] for the coherence measures. In general, `Python` provides a more compound NLP environment, but as preferred by the project partner, `R` was primarily used. The following packages were used: `franc` [Csardi et al., 2015], `cld3` [Ooms, 2017], `tm` [Feinerer et al., 2008], `data.table` [Dowle and Srinivasan, 2018], `wordcloud` [Fellows, 2014], `topicmodels` [Grün and Hornik, 2011], `reticulate` [Allaire et al., 2018] to simulate a `Python` interface within `R`. All kinds of functions out of `tidyverse` [Wickham, 2017] and `tidytext` [Silge and Robinson, 2016]. The `optimal_k()` function was provided by [Rinker, 2016].
Supplementary material is divided by data, general functions, functions related to news and questionnaire data, graphics, the `LDAvis` interfaces and the working environments, with which the calculated results can simply be recalled.

Statement of authorship

I declare that I completed this thesis on my own and that information which has been directly or indirectly taken from other sources has been noted as such.

Neither this nor a similar work has been presented to an examination committee.

Munich, August 21, 2019 .....